

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

**SÍNTESE DE CONTROLADORES PARA TRANSITÓRIOS EM
ESCALONAMENTO DE JOB SHOP CÍCLICO**

Autor: Gideon Villar Leandro

Orientador: Prof. Dr. Rafael Santos Mendes

Este exemplar corresponde a redação final da tese defendida por <u>Gideon Villar Leandro</u> e aprovada pela Comissão Julgada em <u>03/06/00</u> <u>Rafael Santos Mendes</u> Orientador
--

Banca Examinadora:

Prof. Dr. Rafael S. Mendes (Orientador) – DCA/FEEC/Unicamp

Prof. Dr. Antônio Marcus Nogueira Lima - DEE/ UFPB

Prof. Dr. Paulo Eigi Miyagi - Poli/USP

Prof. Dr. Fernando A. Campos Gomide – DCA/FEEC/Unicamp

Prof. Dr. Luis Gimeno Latre – DCA/FEEC/Unicamp

Prof. Dr. Márcio Luiz Andrade Netto – DCA/FEEC/Unicamp

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Tese apresentada à Faculdade de Engenharia
Elétrica e de Computação da Universidade
Estadual de Campinas, como parte dos pré-
requisitos para a obtenção do Título de Doutor
em Engenharia Elétrica.

Agosto de 2000

UNIDADE BC
 N.º CHAMADA:
I/UNICAMP
L476s
 V. Ex.
 TOMBO BC/ 43391
 PROC. 16-392/01
 C ☐ D ☒
 PREÇO R\$ 11,00
 DATA 09/01/01
 N.º CPD

CM-00153445-7

FICHA CATALOGRÁFICA ELABORADA PELA
 BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

L476s

Leandro, Gideon Villar

Síntese de controladores para transitórios em escalonamento de job shop cíclico / Gideon Villar Leandro.--Campinas, SP: [s.n.], 2000.

Orientador: Rafael S. Mendes.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Controle da produção. 2. Planejamento da produção. I. Mendes, Rafael S. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação III. Título.

RESUMO

Um escalonamento cíclico é uma seqüência de tarefas realizadas num conjunto de máquinas e repetido indefinidamente. O escalonamento cíclico aparece em problemas de job shop em que vários produtos tem taxas de demanda equivalentes com tempo e custo de *setup* insignificantes. Parte-se da premissa que uma solução cíclica de regime é conhecida, e que perturbações tais como quebra de máquina, falta de matéria prima, etc, desviam o sistema de sua solução de regime. Propõe-se neste contexto, um procedimento sistemático para a síntese de controladores *on-line* que atuem no sistema perturbado com o objetivo de levá-lo de volta a sua solução de regime. A lei de controle é especificada numa álgebra não usual denominada álgebra max-plus, e tem por objetivo determinar os instantes de início dos jobs nos ciclos subsequentes à extinção de uma eventual perturbação. Será mostrado utilizando-se esta lei de controle, que a solução converge para a solução de regime depois de um número finito de iterações.

ABSTRACT

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

A cyclic scheduling is a sequence of tasks, infinitely repeated by a set of machines. Cyclic scheduling arises in job-shop problems whenever several products are regularly produced at the same rate, with setup cost and time negletable. A steady-state solution for this problem is assumed to be known. We also assume that the system is subject to disturbances (e.g. machine failures, lack of material etc.) and their effect drives the system away from the steady-state. This work proposes a systematic procedure for real-time, feedback controller synthesis whose main task is to reject disturbances, that is, to drive the system back to its steady-state operation. The controllers derived by the synthesis procedure act on the system by determining the starting times of the next tasks. The control law is developed within a max-plus algebra framework and is shown to be stable, that is, after a finite number of iterations, the system returns to steady-state.

AGRADECIMENTOS

A DEUS que me tem dado forças quando estou fraco, que me levanta quando estou caído e principalmente pelo dom da vida. Portanto, até aqui tem o Senhor nos ajudado.

À minha esposa Regiane que nestes anos tem suportado com amor e carinho as dificuldades que temos enfrentado no dia a dia, e por ter nos dado nossa maior riqueza; Gabriela.

À minha filha Gabriela que nestes dois últimos anos tem me feito refletir melhor a vida, e deu um significado todo especial a ela.

Ao professor Rafael Santos Mendes por nos ter encorajado, partilhado momentos de incertezas, e pela orientação competente e valiosa, cujo o intuito foi garantir a melhor qualidade ao trabalho.

Aos professores Dr. Fernando A. C. Gomide, Dr. Luis Gimeno Latre e Dr. Márcio L. A. Netto, membros da banca do exame de qualificação pelas sugestões apresentadas que melhoram a apresentação de itens deste trabalho.

Aos professores Dr. Antônio Marcus N. Lima e Dr. Paulo Eigi Miyagi, por terem aceito participar da banca examinadora.

À Denise Sodero Vinhas Portugal pela concessão dos problemas de job shop e suas soluções de regime apresentados neste trabalho.

Aos meus familiares que sempre me apoiaram e incentivaram nesta caminhada.

À UNIJUÍ pelo suporte financeiro.

À CAPES pela concessão de bolsa PICD.

Sumário

Agradecimentos	v
Lista de Figuras	xi
Lista de Tabelas	xiii
Notação	xv
1 – INTRODUÇÃO	1
2 - PROBLEMAS DE ESCALONAMENTO	7
2.1 – Introdução	7
2.2 – Caracterizações de Problemas de Escalonamento	8
2.2.1 – Conjunto de tarefas	8
2.2.2 – Conjunto de máquinas (processadores)	10
2.2.3 – Critérios de desempenho	10
2.2.4 – Suposições adicionais	11
2.3 – Classificação de Problemas de Escalonamento	12
2.3.1 – Problema de escalonamento em função da demanda	12
2.3.2 – Problema de escalonamento em função do conhecimento sobre o tempo de processamento	13
2.3.3 – Problema de escalonamento em função da existência/importância das perturbações	14
3 - O PROBLEMA DO JOB SHOP CÍCLICO	15
3.1 – Introdução	15
3.2 – O Job Shop Cíclico	17
3.2.1 – Modelo matemático	18
3.2.1.1 – Simbologia	18
3.2.1.2 – Formulação matemática	19
3.3 – Soluções de Regime	26
3.4 – Planejamento Reativo de Job Shop Cíclico	27

3.4.1 – Regras de despacho	27
3.4.2 – Leis de controle	28
4 – DIÓIDES	31
4.1 – Introdução	31
4.2 – Teoria Geral de Dióides	32
4.2.1 – Axiomatização e definições básicas	32
4.2.2 – Propriedades de reticulados dos dióides	36
4.2.3 - Dióides de matrizes	38
4.2.4 – Dióides de polinômios e séries de potência	39
4.3 - Álgebra Max-Plus e Teoria dos Grafos	41
4.3.1 - Dióide max-plus ou álgebra max-plus	41
4.3.2 – Teoria dos grafos	42
4.3.3 – Grafos e matrizes em $Z_{\max}$	43
4.3.4 – Produtos generalizados de matrizes	48
4.4 - O Dióide P	50
4.4.1 – O dióide I	50
4.4.2 – O dióide P	53
5 – ESTRATÉGIA REATIVA PARA O TRANSITÓRIO DO JOB SHOP CÍCLICO	55
5.1 – Introdução	55
5.2 – Modelo para a Análise da Factibilidade das Decisões	60
5.3 – Síntese da Lei de Controle	67
6 – ALGORITMOS E RESULTADOS	75
6.1 – Introdução	75
6.2 – Algoritmos de Síntese	76
6.2.1 – Algoritmo de síntese I	77
6.2.2 – Algoritmo de síntese II	79
6.2.3 – Algoritmo de síntese III	79
6.2.4 – Algoritmo de síntese IV	81
6.3 – Resultados Obtidos	82

6.4 – Eliminação das Esperas Fixas	120
6.5 – Análise dos Resultados	129
7 – CONCLUSÕES	131
Referências Bibliográficas	133

LISTA DE FIGURAS

Figura 1.1 – Diagrama de blocos do problema sob o ponto de vista da teoria de controle	4
Figura 2.1 – Grafo de precedência entre tarefas	9
Figura 3.1 – Gráfico de Gantt para o exemplo 3.1	22
Figura 3.2 – Gráfico de Gantt para o exemplo 3.2	25
Figura 4.1 – Elemento típico $m(\tau)$	51
Figura 4.2 – Elemento típico $m(k)$	54
Figura 5.1 – Representação do novo instante de início do job	57
Figura 5.2 – Gráfico típico de um elemento da função s_{ij}	61
Figura 5.3 – Matriz M para o exemplo 5.1	65
Figura 5.4 – Matriz M^* para o exemplo 5.1	65
Figura 5.5 – Matriz $((t'_1 \otimes M^*) \oplus (t'_0 \otimes M))$ para o exemplo 5.1	66
Figura 5.6 – Diagrama esquemático para visualização da factibilidade	66
Figura 6.1 – Instantes de início do primeiro job de cada seqüência	85
Figura 6.2 – Instantes de início do primeiro job de cada seqüência	86
Figura 6.3 – Instantes de início do primeiro job de cada seqüência	87
Figura 6.4 – Instantes de início do primeiro job de cada seqüência	89
Figura 6.5 – Instantes de início do primeiro job de cada seqüência	90
Figura 6.6 – Instantes de início do primeiro job de cada seqüência	92
Figura 6.7 – Instantes de início do primeiro job de cada seqüência	94
Figura 6.8 – Instantes de início do primeiro job de cada seqüência	95
Figura 6.9 – Instantes de início do primeiro job de cada seqüência	96
Figura 6.10 – Instantes de início do primeiro job de cada seqüência	98
Figura 6.11 – Instantes de início do primeiro job de cada seqüência	99
Figura 6.12 – Instantes de início do primeiro job de cada seqüência	100
Figura 6.13 – Instantes de início do primeiro job de cada seqüência	103

Figura 6.14 – Instantes de início do primeiro job de cada seqüência	104
Figura 6.15 – Instantes de início do primeiro job de cada seqüência	105
Figura 6.16 – Instantes de início do primeiro job de cada seqüência	108
Figura 6.17 – Instantes de início do primeiro job de cada seqüência	109
Figura 6.18 – Instantes de início do primeiro job de cada seqüência	110
Figura 6.19 – Instantes de início do primeiro job de cada seqüência	112
Figura 6.20 – Instantes de início do primeiro job de cada seqüência	113
Figura 6.21 – Instantes de início do primeiro job de cada seqüência	114
Figura 6.22 – Instantes de início do primeiro job de cada seqüência	117
Figura 6.23 – Instantes de início do primeiro job de cada seqüência	118
Figura 6.24 – Instantes de início do primeiro job de cada seqüência	119
Figura 6.25 – Instantes de início do primeiro job de cada seqüência	123
Figura 6.26 – Instantes de início do primeiro job de cada seqüência	124
Figura 6.27 – Instantes de início do primeiro job de cada seqüência	125
Figura 6.28 – Instantes de início do primeiro job de cada seqüência	126
Figura 6.29 – Instantes de início do primeiro job de cada seqüência	127
Figura 6.30 – Instantes de início do primeiro job de cada seqüência	128

LISTA DE TABELAS

Tabela 3.1 – Roteiro de fabricação e tempos de processamento dos jobs	21
Tabela 5.1 – Roteiro de fabricação, duração e instante de início das tarefas	64
Tabela 6.1 – Roteiro de fabricação, duração e instante de início das tarefas	84
Tabela 6.2 – Instantes de início do primeiro job de cada seqüência	85
Tabela 6.3 – Instantes de início do primeiro job de cada seqüência	86
Tabela 6.4 – Instantes de início do primeiro job de cada seqüência	87
Tabela 6.5 – Tempos relativos do exemplo 1	88
Tabela 6.6 – Roteiro de fabricação, duração instante de início das tarefas	88
Tabela 6.7 – Instantes de início do primeiro job de cada seqüência	89
Tabela 6.8 – Instantes de início do primeiro job de cada seqüência	90
Tabela 6.9 – Instantes de início do primeiro job de cada seqüência	91
Tabela 6.10 – Tempos relativos do exemplo 2	92
Tabela 6.11 – Roteiro de fabricação, duração instante de início das tarefas	93
Tabela 6.12 – Instantes de início do primeiro job de cada seqüência	94
Tabela 6.13 – Instantes de início do primeiro job de cada seqüência	95
Tabela 6.14 – Instantes de início do primeiro job de cada seqüência	96
Tabela 6.15 – Tempos relativos do exemplo 3	97
Tabela 6.16 – Roteiro de fabricação, duração instante de início das tarefas	97
Tabela 6.17 – Instantes de início do primeiro job de cada seqüência	98
Tabela 6.18 – Instantes de início do primeiro job de cada seqüência	99
Tabela 6.19 – Instantes de início do primeiro job de cada seqüência	100
Tabela 6.20 – Tempos relativos do exemplo 4	101
Tabela 6.21 – Roteiro de fabricação, duração instante de início das tarefas	102
Tabela 6.22 – Instantes de início do primeiro job de cada seqüência	102
Tabela 6.23 – Instantes de início do primeiro job de cada seqüência	104
Tabela 6.24 – Instantes de início do primeiro job de cada seqüência	105

Tabela 6.25 – Tempos relativos do exemplo 5	106
Tabela 6.26 – Roteiro de fabricação, duração instante de início das tarefas	107
Tabela 6.27 – Instantes de início do primeiro job de cada seqüência	107
Tabela 6.28 – Instantes de início do primeiro job de cada seqüência	108
Tabela 6.29 – Instantes de início do primeiro job de cada seqüência	109
Tabela 6.30 – Tempos relativos do exemplo 6	110
Tabela 6.31 – Roteiro de fabricação, duração instante de início das tarefas	111
Tabela 6.32 – Instantes de início do primeiro job de cada seqüência	111
Tabela 6.33 – Instantes de início do primeiro job de cada seqüência	113
Tabela 6.34 – Instantes de início do primeiro job de cada seqüência	114
Tabela 6.35 – Tempos relativos do exemplo 7	115
Tabela 6.36 – Roteiro de fabricação, duração instante de início das tarefas	116
Tabela 6.37 – Instantes de início do primeiro job de cada seqüência	116
Tabela 6.38 – Instantes de início do primeiro job de cada seqüência	118
Tabela 6.39 – Instantes de início do primeiro job de cada seqüência	119
Tabela 6.40 – Tempos relativos do exemplo 8	120
Tabela 6.41 – Instantes de início do primeiro job de cada seqüência	122
Tabela 6.42 – Instantes de início do primeiro job de cada seqüência	124
Tabela 6.43 – Instantes de início do primeiro job de cada seqüência	125
Tabela 6.44 – Instantes de início do primeiro job de cada seqüência	126
Tabela 6.45 – Instantes de início do primeiro job de cada seqüência	127
Tabela 6.46 – Instantes de início do primeiro job de cada seqüência	128

Notação

R – Conjunto dos números reais

Z – Conjunto dos números inteiros

N – Conjunto dos números naturais

t_i – vetor de instante de início dos jobs no i -ésimo ciclo

$[t_i]_j$ – j -ésima componente do vetor de instante de início no i -ésimo ciclo

$[A]_{ij}$ – elemento da i -ésima linha e j -ésima coluna da matriz A

a_{ij} – elemento da i -ésima linha e j -ésima coluna da matriz A

$[a_{ij}]$ – matriz A composta pelos elementos a_{ij}

A^s – resultado do produto da matriz A com a própria matriz A , s vezes (no contexto da álgebra de dióides)

$[a]$ – classe de equivalência definida por a

$t^\#$ – solução de referência

$A^\#$ – matriz de referência

$\lambda^\#$ – autovalor da matriz de referência ($A^\#$)

z – comprimento do período ou ciclo

$\oplus$ - operação interna de um dióide denominada soma ou adição

$\otimes$ - operação interna de um dióide denominada produto ou multiplicação

ε - elemento neutro da operação soma na álgebra de dióides

e - elemento neutro da operação produto na álgebra de dióides

$D[z_1 z_2, \dots, z_n]$ – dióide de polinômial construído a partir do dióide D

$D[[z_1 z_2, \dots, z_n]]$ – dióide em série de potência construído a partir do dióide D

t'_1 – monômio tal que $t'_1 = f([t]_1) \otimes \tau^{t_1}$

$B[[\tau]]$ – dióide em série de potências com coeficientes booleanos

CAPÍTULO 1

INTRODUÇÃO

Nos últimos anos, indústrias e universidades têm cada vez mais investido em pesquisas de técnicas para modelar, analisar e controlar sistemas complexos tais como: sistemas de manufaturas flexíveis, sistemas de controle de tráfego, redes de comunicação, sistemas de processamento paralelo entre muitos outros.

Estes sistemas ficam a cada dia mais complexos e sofisticados, envolvendo muitas formas de tecnologia. Devido à necessidade de agilização do processo produtivo, causada principalmente pelo surgimento de um mercado crescentemente competitivo, tem-se buscado técnicas novas e eficientes para a solução dos mais diversos problemas presentes no cotidiano de atividades produtivas.

No dia a dia de muitas indústrias de manufatura, o escalonamento é uma necessidade real. O escalonamento é, segundo Baker (1974), definido como sendo o problema de alocação de recursos ao longo do tempo, cuja finalidade é a execução de um conjunto de tarefas. O conjunto de tarefas são as atividades que se deseja executar e, para tanto dispõe-se de certas quantidades dos recursos necessários para a sua execução. Portanto, considera-se que o problema de escalonamento parte de um conjunto definido de recursos e de um conjunto de tarefas que são realizáveis com os recursos disponíveis.

Existem diversos critérios na literatura para organizar e classificar os problemas de escalonamento. Um dos problemas de escalonamento bem conhecido é o problema de “Job Shop”, o qual intuitivamente pode ser descrito da seguinte forma: serviços (comumente conhecidos como “jobs”), compostos por um conjunto de tarefas, devem ser processados em

um conjunto de máquinas. O objetivo consiste em otimizar alguma função do tempo de término destes serviços, satisfazendo a diversas restrições.

As diversas técnicas de escalonamento surgiram com o objetivo de se otimizar as plantas industriais, ou seja, encontrar o máximo ou mínimo de uma certa função definida em algum domínio. Num ambiente industrial típico, deseja-se maximizar a produção com a instalação disponível.

No desenvolvimento de novos métodos de solução de problemas de otimização, problemas de escalonamento de produção, tal qual o job shop, têm despertado grande interesse. Tanto métodos exatos, quanto aproximados, vêm sendo objetos de pesquisa na procura por formas convenientes à solução do problema.

Segundo Morton & Pentico (1993), o problema de escalonamento de job shop clássico pode ser formulado da seguinte forma: n jobs devem ser processados em m diferentes máquinas, cada job sendo formado por uma seqüência de tarefas¹ que devem ser processadas em uma ordem fixa, cada tarefa sendo executada em uma máquina previamente estipulada e com uma duração fixa.

Um caso particular de job shop, que tem tido uma crescente importância no meio industrial e acadêmico, e que é o alvo deste trabalho, é o problema de job shop cíclico. O problema de job shop cíclico, pode ser visto como uma seqüência de jobs, cada job tendo uma quantidade fixa de tarefas, que devem ser processados em um conjunto de máquinas, sendo este procedimento repetido indefinidamente.

A maioria dos trabalhos na área de escalonamento de job shop cíclico, preocupam-se em determinar uma solução ótima ou a mais próxima dela para o problema, que neste trabalho passará a ser chamada de solução de regime ou solução de referência. A utilização prática desta solução supõe que o sistema de produção está isento de perturbações. Contudo, isto não é a realidade do meio industrial, que está sujeito a diversas perturbações (falhas imprevisíveis; falhas que não foram incorporadas no modelo), tais como quebra de máquinas, falta de matéria prima, falta de espaço para estoque, etc. Com isto, a solução de regime é perdida se alguma perturbação acontece e nenhuma medida é tomada a tempo.

¹ Em Morton & Pentico (1993), o termo operação é utilizado no lugar da palavra tarefa

De um modo geral a reprogramação do sistema, utilizando-se as mesmas ferramentas matemáticas usadas para o cálculo da solução de regime, é inviável. Isto porque os métodos de programação matemática usados são custosos computacionalmente, não havendo tempo hábil para a sua utilização.

Frente à ocorrência de uma perturbação, uma decisão em geral deve ser tomada em pouco tempo.

Existem alguns caminhos a serem seguidos após a ocorrência de perturbações na linha de produção, cujo intuito é fazer o sistema voltar ao regime sem acarretar grandes prejuízos. A primeira alternativa seria a parada da linha de produção quando da ocorrência da perturbação e, retomá-la quando a perturbação estiver sanada, mas possivelmente isto não evitará eventuais prejuízos à produção. A segunda alternativa seria a utilização de regras de despacho, contudo sem a garantia da volta para a solução de regime, podendo novamente ocorrer prejuízos à produção. A terceira alternativa é utilizar leis de controle que tentem retornar à solução de regime sem a interrupção da produção. Com isso minimiza-se os prejuízos, evitando-se desperdícios de materiais e perda de produtos.

Do exposto anteriormente, suponha-se que se conheça uma solução de regime (referência) do job shop cíclico, isto é, os instantes de início de cada tarefa de cada job. Imagine-se agora uma perturbação na linha de produção. Como retornar ao regime? Neste ponto encontra-se o objetivo deste trabalho. Busca-se, com base em uma lei de controle em tempo real (*on-line*), reprogramar as ordens de instante de início dos jobs, assegurando ao sistema o seu retorno ao regime no menor tempo, sem a interrupção da produção.

A lei de controle proposta é baseada numa álgebra não usual denominada álgebra de dióides, mais especificamente a álgebra max-plus, e consiste na determinação dos próximos instantes de início dos jobs, utilizando-se de informações dos instantes de início precedentes.

Resumidamente, conhecendo-se uma solução de regime (referência) e os instantes de início dos jobs, determinam-se os limitantes superior e inferior para o próximo instante de início dos jobs. Se nenhuma perturbação ocorre, estes limitantes são iguais, e a solução de regime é mantida. Fora da trajetória de regime permanente, ou seja, se os instantes de início dos jobs não são os mesmos da solução de regime, os limitantes necessariamente são

diferentes entre si. Este procedimento é utilizado até que o sistema volte à trajetória de referência e os limitantes voltem novamente a ser iguais.

Esta lei de controle terá como característica principal a determinação de instantes de início do próximo ciclo o mais cedo possível, garantindo assim sempre uma alternativa de início dos jobs. Portanto, uma vantagem da lei de controle apresentada, é a sua rapidez em obter uma nova solução para o problema, em relação ao método de programação matemático utilizado para a obtenção da solução de regime, associada a outra vantagem que é a garantia da estabilidade do sistema, ou seja, a garantia da convergência da solução para a solução de regime.

O problema exposto nos parágrafos precedentes e que será abordado neste trabalho, pode ser apresentado informalmente sob a ótica da teoria de controle, pelo seguinte diagrama de blocos:

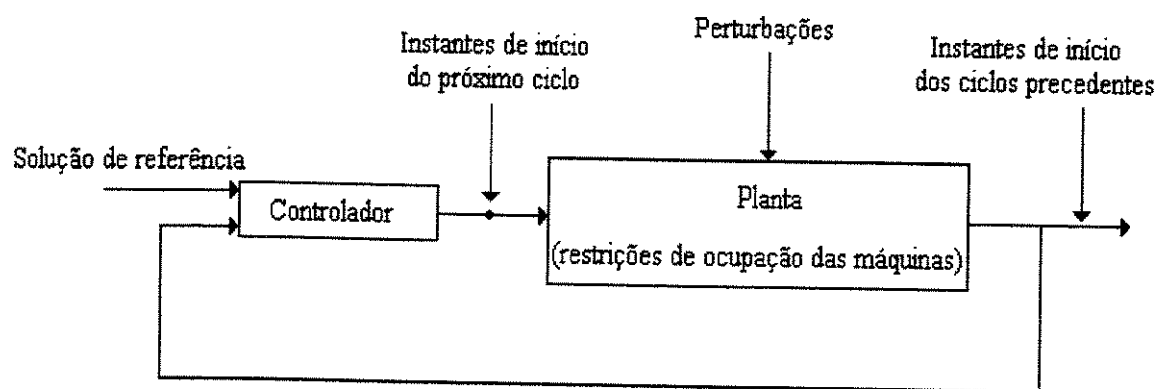


Figura 1.1 – Diagrama de blocos do problema sob o ponto de vista da teoria de controle.

Nesta representação, as entradas do controlador são a solução de regime e a solução do ciclo anterior, neste trabalho supõe-se que a solução de regime é conhecida e que a solução do ciclo anterior está disponível. Com estas informações o controlador decidirá quais são os instantes de início do próximo ciclo. Ou seja, o controlador é um decisor que deve satisfazer duas restrições (a restrição de estabilidade e a restrição de factibilidade). As perturbações que

afetam a planta são consideradas determinísticas, pois somente após a ocorrência de alguma perturbação e consequentemente sua extinção o controlador irá agir.

Este trabalho está organizado em capítulos como dado a seguir. O capítulo 2 apresenta o problema de escalonamento e as grandezas físicas e matemáticas que o caracterizam. O capítulo 3 apresenta o problema de escalonamento do job shop cíclico, assim como a formulação matemática que é utilizada na determinação da solução de referência. Suas características em regime também são abordadas, e algumas estratégias como regras de despacho e leis de controle já implementadas são resumidamente explanadas. Uma revisão dos trabalhos mais relevantes nesta área, já publicados, também é apresentada.

No capítulo 4 é feita uma revisão da álgebra de dióides que é utilizada duplamente. Por um lado, na forma da álgebra max-plus, para a análise e síntese da lei de controle, e para a demonstração de sua estabilidade. Por outro lado, na forma de um dióide polinomial, para os testes de factibilidade do sistema.

No capítulo 5 é apresentado o teste de factibilidade juntamente com um exemplo para sua elucidação, o ferramental matemático necessário a obtenção da lei de controle que será utilizada, assim como uma prova de que esta lei faz o sistema convergir para sua trajetória de referência, ou seja, após um número determinado de ciclos a solução retorna para a solução de referência.

O capítulo 6 consiste da apresentação dos algoritmos sintetizados a partir da lei de controle e do teste de factibilidade, que determina os instantes de início que serão utilizados no sistema quando este está perturbado (regime transitório). Vários exemplos são apresentados com o intuito de mostrar a eficácia dos controladores, em fazer a solução retornar a solução de referência e consequentemente o sistema ao regime permanente.

As conclusões, discussões e sugestões de trabalhos futuros se encontram no capítulo 7.

CAPÍTULO 2

PROBLEMAS DE ESCALONAMENTO

2.1 – Introdução

Problemas de escalonamento de tarefas têm recebido uma grande atenção nas últimas décadas. Parte da atenção dispensada provém do interesse teórico de pesquisadores em áreas como otimização combinatória, análise de complexidade e análises de pior caso, entre outras. Em acréscimo a este interesse teórico, o valor prático a eles atribuído (aplicabilidade a sistemas de manufaturas flexíveis e sistemas multiprocessadores, por exemplo) tem representado uma motivação adicional ao desenvolvimento de métodos satisfatórios para solução de tais problemas (Tzafestas & Triantafyllakis, 1993).

Em geral, pode-se entender problemas de escalonamento como problemas de alocação de recursos ao longo do tempo, cuja finalidade é a execução de um conjunto de tarefas (Baker, 1974 e Morton & Pentico, 1993).

Os elementos básicos de problemas de escalonamento são:

- um conjunto J de n jobs: $J = \{J_1, J_2, \dots, J_n\}$, com J_i possuindo k tarefas;
- um conjunto M de m máquinas (procesadores): $M = \{M_1, M_2, \dots, M_m\}$.

Além destes, na maioria dos modelos gerais de escalonamento também é considerado um conjunto R de recursos adicionais (outros além das máquinas).

Há duas restrições gerais na teoria clássica de escalonamento:

- i) cada tarefa pode ser processada por no máximo uma máquina a cada instante
- ii) cada máquina é capaz de executar no máximo uma tarefa por vez.

Diferentes critérios que medem a qualidade do desempenho de um conjunto de jobs são também levados em consideração. Assim, para que problemas de escalonamento de tarefas sejam formulados completamente, são necessárias algumas informações específicas pertinentes ao conjunto de jobs J , ao conjunto de máquinas (processadores) M e ao critério de desempenho que se deseja otimizar (conhecido como função objetivo).

Na seção 2.2 são apresentados os elementos básicos e representações padrões para o problema de escalonamento, assim como são exemplificados alguns critérios de desempenho freqüentemente usados na avaliação da qualidade de soluções. Na seção 2.3 são resumidos os principais tipos de problemas de escalonamento em sistemas de manufatura.

2.2 – Caracterizações de Problemas de Escalonamento

São apresentadas a seguir algumas caracterizações dos elementos que compõem problemas de escalonamento de tarefas. Estas caracterizações se fazem necessárias a fim de que a estrutura destes problemas seja bem compreendida.

2.2.1 – Conjunto de tarefas

Cada job $J_j \in J$ pode ser caracterizado pelos seguintes dados:

- i) um vetor de tempo de processamento $p_{jk} = \{p_{jk1}, p_{jk2}, \dots, p_{jkn}\}$, onde p_{jki} é o tempo requerido pela máquina (processador) M_i , $i=1, 2, \dots, m$, para executar a tarefa k do job J_j ,
- ii) um *ready time* $r_{jk} \geq 0$ que indica o instante de tempo após o qual a tarefa k do job J_j encontra-se disponível para ser executada em alguma máquina,
- iii) uma *due date* d_{jk} , que declara o instante de tempo desejado como limite para o término de execução da tarefa k do job J_j ,
- iv) uma ponderação (prioridade) w_{jk} que retrata a importância relativa da tarefa k do job J_j , em relação as demais tarefas dos outros jobs.

Observe que os dados acima constituem atributos de cada tarefa individualmente. Em acréscimo a estes, para o conjunto de tarefas como um todo são especificadas as seguintes questões:

- o modo de processamento das tarefas pode ser “preemptivo” ou “não preemptivo”. Um escalonamento é dito preemptivo se qualquer tarefa pode ser interrompida a qualquer momento, ou seja, pode ter sua execução suspensa por um montante limitado de tempo, e sua continuação ocorrer num instante posterior (na mesma ou em outra máquina) sem custo adicional.
- as tarefas de $J_j \in J$ são ditas dependentes caso a ordem de execução de pelo menos duas tarefas de J_j esteja restrita por relações de precedência. A notação $i < k$ indica que a execução da tarefa i deve ser finalizada antes que a tarefa k inicie sua execução. Caso inexista a relação de precedência entre tarefas, estas são ditas independentes.

Restrições de precedência são comumente representadas por meio de grafos dirigidos acíclicos (GDA). Os grafos devem ser acíclicos a fim de que impasses (*deadlocks*) na execução das tarefas sejam evitados. A figura 2.1 ilustra um grafo de precedência simples de um job composto de seis tarefas, onde os nós do grafo representam as tarefas e os arcs dirigidos representam as restrições de precedência. Observe que, por exemplo, a tarefa T_6 só pode ser iniciada após a conclusão de todas as demais tarefas, excetuando-se a tarefa T_4 , dado que esta não guarda nenhuma relação de precedência com aquela.

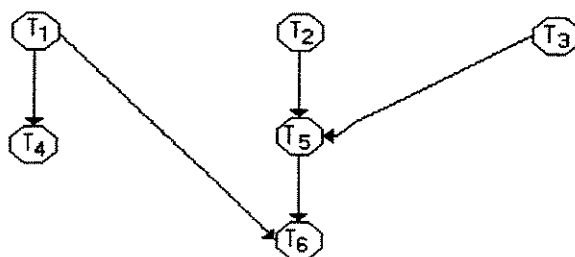


Figura 2.1 – Grafo de precedência entre tarefas de um mesmo job J_j .

2.2.2 – Conjunto de máquinas (processadores)

Máquinas ou processadores que podem executar qualquer tarefa são designados “multipropósitos”, enquanto que, se cada processador é especializado na execução de determinadas tarefas, eles são ditos “dedicados”.

Os processadores multipropósitos podem ser classificados, de acordo com suas velocidades, em três tipos:

- i) idênticos: todos os processadores de M têm a mesma velocidade de processamento. A velocidade de cada processador é independente da tarefa que está sendo executada.
- ii) uniformes: os processadores de M têm velocidades distintas mas constantes. A velocidade de cada processador é independente da tarefa que está sendo realizada.
- iii) Não relacionados: a velocidade de cada processador é variável, e depende da tarefa que está sendo executada.

Assim se para toda tarefa $k \in J_j$, $j=1, 2, \dots, n$ é mantida a relação $p_{jki} = p_{jk}$, $i=1, 2, \dots, m$, então temos processadores idênticos. Enquanto se $p_{jki} = p_{jk}b_i$, $i=1, 2, \dots, m$, onde b_i^{-1} é uma medida de velocidade de M , independente da tarefa k do job J_j , então temos processadores uniformes.

Processadores dedicados podem operar em um dos seguintes modos:

- i) Flow Shop: cada conjunto de tarefas pertencente ao job J_j deve ser executado por todos os processadores, na mesma ordem, sendo esta previamente definida.
- ii) Open Shop: difere do Flow Shop quanto á ausência de uma ordem de execução predeterminada das tarefas dos jobs.
- iii) Job Shop: cada job J_j deve ser executado por um subconjunto de processadores de M . A ordem de execução das tarefas do job é arbitrária e, claro, pode ser diferente de job para job. Contudo, esta ordem deve ser dada como entrada para o problema.

2.2.3 – Critérios de desempenho

Toda seqüência que representa a solução de algum problema de escalonamento, é caracterizada por algum critério ou medida de desempenho. Tal medida consiste em um

qualificador do escalonamento baseado na função objetivo associada ao problema. O escalonamento que minimiza (ou maximiza) determinado critério é dito ser ótimo com respeito a este critério. Alguns dos critérios mais utilizados como função objetivo para problemas de escalonamento são:

- i) tamanho do escalonamento (*makespan*): $C_{\max} = \max_j \{C_j\}$

onde: C_j = tempo de término da última tarefa do job J_j .

- ii) tempo médio de fluxo : $F = \frac{1}{n} \left(\sum_1^n f_j \right)$

onde: f_j = tempo que o job J_j permanece no sistema.

- iii) máximo atraso: $L_{\max} = \max_j \{L_j\}$

onde: L_j = tempo em que excede o instante de tempo desejado para o término do job J_j .

Em Baker (1974) é ressaltado que o critério i) é usado quando se deseja uma utilização eficiente de recursos, ii) está associada a datas e prazos de prescrições (*deadlines*) e iii) é um bom critério quando se deseja avaliar demanda por resposta rápida.

2.2.4 – Suposições adicionais

Algumas suposições adicionais adotadas neste trabalho são as seguintes:

- máquinas estão sempre disponíveis e nunca quebram.
- *buffers* entre processadores têm capacidade ilimitada. Embora existam problemas em que a capacidade de estocagem é uma restrição importante, neste trabalho considera-se o caso que esta capacidade é tão grande quanto o necessário.
- uma tarefa concluída em uma máquina pode esperar até que seu processamento seja iniciado na próxima máquina.
- *ready times*, isto é, os momentos em que as tarefas estão disponíveis para serem processadas, são iguais a zero. Logo, todas as tarefas encontram-se disponíveis no começo do processamento, observando-se que as restrições de precedência devem ser obedecidas.

- tempos gastos na configuração de máquinas são independentes do escalonamento e estão incluídos no tempo de processamento.
- tempos de processamento e restrições tecnológicas são determinísticos e previamente conhecidos.

2.3 – Classificação dos Problemas de Escalonamento

Nesta seção será feita uma classificação dos problemas de escalonamento, segundo as características apresentadas por eles e denominações encontradas na literatura (Zhang & Graves, 1997; Graves et al, 1983). Vale ressaltar que esta classificação não é uma classificação genérica e, sim uma classificação cujo objetivo é situar o problema que será estudado neste trabalho. Esta classificação abordará três aspectos, o aspecto da demanda, o aspecto do tipo de conhecimento sobre o tempo de processamento e o aspecto da existência/importância das perturbações.

2.3.1 – Problema de escalonamento em função da demanda

Problema de escalonamento não-cíclico. Neste problema de escalonamento, o conjunto de jobs a serem executados no conjunto de máquinas é realizado apenas uma vez. Há numerosos resultados sobre escalonamento não-cíclico publicados, os quais apresentam as mais variadas metodologias e principalmente heurísticas para a obtenção de uma solução ótima.

Problema de escalonamento cíclico ou periódico. Neste problema de escalonamento, o conjunto de jobs a serem executados no conjunto de máquinas é realizado indefinidamente, mantendo sempre a mesma sequência das tarefas nas máquinas. Por ser o escalonamento cíclico um caso particular do escalonamento não-cíclico, existe na literatura um número bem menor de estudos para a obtenção de uma solução para este problema. Neste caso, busca-se

principalmente determinar uma sequência de processamento que minimize o ciclo. No capítulo subsequente uma metodologia para a obtenção de soluções para este problema será apresentado, haja vista a necessidade de tais soluções para a obtenção da lei de controle, objetivo maior deste trabalho.

2.3.2 – Problema de escalonamento em função do conhecimento sobre o tempo de processamento.

Problema de escalonamento determinístico. Neste problema de escalonamento, o conjunto de jobs a serem executados no conjunto de máquinas, quer seja realizado apenas uma vez ou indefinidamente, os tempos da duração do processamento das tarefas são todos eles conhecidos *a priori*. A grande maioria dos trabalhos disponíveis na literatura utilizam este procedimento, pois não levam em consideração perturbações que possam ocorrer no sistema.

Problema de escalonamento não-determinístico. Neste problema de escalonamento, o conjunto de jobs a serem executados no conjunto de máquinas, quer seja realizado apenas uma vez ou indefinidamente, não tem os tempos da duração de processamento das tarefas conhecidos *a priori*. Para este tipo de problema pode-se não ter nenhuma informação sobre os tempos da duração de processamentos das tarefas, ou pode-se conhecer regiões que possivelmente eles poderão pertencer (teoria *fuzzy*). Há também problemas que é conhecida alguma distribuição (Poisson, Uniforme, Gaussiana, etc) que representa a probabilidade dos tempos de processamento das tarefas. Com isto, somente para aqueles jobs correntemente sendo processados nas máquinas, o tempo de início podem ser determinados. Os poucos trabalhos disponíveis na literatura que utilizam esta metodologia, levam em consideração perturbações que ocorrem no sistema tais como quebra de máquinas, falta de matéria prima, etc. E através de regras de prioridade ou de despacho obtêm-se uma solução.

2.3.3 – Problema de escalonamento em função existência/importância das perturbações.

Problema de escalonamento estático. Neste tipo de problema de escalonamento, a solução para o conjunto de jobs a serem executados no conjunto de máquinas, quer seja realizado apenas uma vez ou indefinidamente, é obtida integralmente após um único procedimento de cálculo. Em outras palavras supõe-se a não existência de perturbações no sistema que provoquem a necessidade de um replanejamento para o escalonamento. Para este caso, há na literatura um farto material com algoritmos específicos para a obtenção da solução.

Problema de escalonamento dinâmico. Neste problema de escalonamento, o conjunto de jobs a serem executados no conjunto de máquinas, quer seja realizado apenas uma vez ou indefinidamente, não tem os seus instantes de início conhecidos *a priori* nos ciclos subsequentes, pois eles são modificados ao longo do tempo. Problemas de escalonamento estocástico, serão sempre dinâmicos devido à sua própria natureza. Já o escalonamento determinístico será dinâmico, quando regras de prioridade são usadas para determinar o início das tarefas ao longo do tempo.

Do exposto anteriormente, utilizar-se-á no restante deste trabalho os seguintes tipos de escalonamentos: a solução de referência será obtida utilizando-se o escalonamento de um job shop cíclico determinístico estático. No regime transitório pelo qual o job shop cíclico irá passar, estará sendo utilizado o escalonamento determinístico dinâmico, pois os tempos de início dos jobs serão determinados pela lei de controle, os quais não são conhecidos *a priori*.

CAPÍTULO 3

O PROBLEMA DO JOB SHOP CÍCLICO

3.1- Introdução

Escalonamento cíclico é uma técnica que vem gradualmente sendo incorporada nas ultimas décadas aos mais diversos processos industriais. Esta técnica é mais vantajosa do que o escalonamento clássico, quando o tempo e custo de *setup* são considerados insignificantes (Lee & Posner, 1997).

Dauscha *et al.* (1985) descrevem o problema de escalonamento cíclico como: “um dado conjunto finito de tarefas que deve ser escalonado em um fluxo contínuo e periódico, isto é, todas as tarefas devem ser repetidamente executadas de forma que o intervalo de tempo entre dois instantes de início sucessivos de uma mesma tarefa é uma constante z para todas as tarefas”. Esta constante z é chamada de comprimento do período (ou ciclo).

Vários estudos teóricos, a respeito do problema de escalonamento cíclico em um ambiente de manufatura de produção periódica têm sido realizados. Com respeito ao problema de job shop, uma primeira tentativa de escalonamento cíclico foi proposta por Graves *et al.* (1983) no qual se tinha o problema de produzir vários produtos em uma proporção fixa por um longo período de tempo, sendo que um job poderia passar em uma máquina uma ou mais vezes. Roundy (1992) estudou a estrutura combinatorial do problema e provou que sua complexidade é NP-difícil. Também estabeleceu uma caracterização da “estrutura de precedência cíclica” que é usada para desenvolver um procedimento de busca *branch & bound*. Rao (1992) em seu estudo, propôs uma formulação em programação inteira mista para

o problema de jobs idênticos. Para isto ele combina as restrições do problema de escalonamento básico e máquinas especializadas, isto é, máquinas cujas tarefas estão associadas exclusivamente a elas. O objetivo é achar um escalonamento periódico com um fluxo máximo, o que equivale a achar um escalonamento periódico com um período mínimo. Os tempos de processamento das tarefas são fixos, não existindo sobreposição de tarefas em uma mesma máquina e também cada tarefa é associada a uma única máquina. Hanen (1994) propõe em seu estudo uma formulação em programação inteira mista para o problema de escalonamento cíclico em que considera que qualquer seqüência de processamento das tarefas associadas a um recurso é factível o qual denominou job shop recorrente. Neste caso também o objetivo é achar um escalonamento periódico com um fluxo máximo. Nesta formulação as tarefas são não-reentrantes, isto é, duas ocorrências de uma tarefa genérica não se sobrepõem, e os tempos de processamento das tarefas são fixos. Portugal (1999) propõe em seu trabalho adaptações nos trabalhos de Rao (1992) e Hanen (1994) para obter soluções do problema de job shop cíclico com recursos disjuntivos e múltiplos jobs, o qual denominou de job shop cíclico com jobs distintos. Para isto uma formulação em programação inteira mista é proposta.

Existem ainda na literatura vários outros trabalhos que abordam o tema escalonamento de job shop cíclico. A seguir alguns deles serão brevemente citados. Yura(1989) mostrou a necessidade de uma capacidade de *buffer* para haver um escalonamento sem ociosidade; Song e Lee (1996) desenvolveram um procedimento de busca tabu; Song e Lee (1998) utilizaram uma rede de Petri para solucionar um job shop periódico com bloqueio e propuseram também uma modelagem em programação inteira mista para achar um escalonamento livre de impasse (*deadlock*) e com ciclo mínimo; Zhang e Graves (1997) estudaram o comportamento de um escalonamento cíclico em um ambiente estocástico caracterizado por quebras aleatórias de máquinas, que poderiam atrasar a execução de tarefas e como consequência atrapalhar o escalonamento especificado; Lee (1994), Groth e Santos-Mendes (1997) e Nawijn (1998) utilizam procedimentos baseado em uma álgebra não convencional chamada de dióide para obterem uma solução para o problema ; Hanen (1994), Munier (1996) e Levner e Kats (1998) utilizam a teoria dos grafos para obterem um procedimento de resolução para o problema.

Na seção 3.2 é apresentado o problema do job shop cíclico juntamente com a formulação utilizada para o cálculo da solução de referência levando-se em conta sua natureza

determinística. Esta solução é de fundamental importância para o restante do trabalho. Na seção 3.3 são realçadas as características do job shop cíclico quando ele encontra-se em regime permanente. E na seção 3.4 são resumidas as estratégias utilizadas, quando o job shop cíclico está perturbado (regime transitório), encontradas na literatura.

3.2 – O Job Shop Cíclico

O problema do job-shop cíclico, pode ser caracterizado resumidamente como sendo o problema de escalonar n jobs, com cada job tendo um conjunto de tarefas não necessariamente iguais a dos outros jobs, que utilizam m máquinas em uma determinada sequência para cada job, infinitas vezes. O objetivo é determinar períodos (ciclos), que compreendem a inicialização de um job até o próximo disparo deste mesmo job, sendo que, deverá ocorrer neste intervalo o disparo de todos os jobs restantes. Ao resolver este problema busca-se maximizar o número médio de iterações por unidade de tempo ou minimizar o tempo de ciclo. Vale ressaltar que estas tarefas são não reentrantes, isto é, duas ocorrências de uma mesma tarefa não podem se sobrepor. Além disso, uma vez que uma tarefa comece a ser processada, não poderá ser interrompida para a execução de uma outra tarefa qualquer.

O problema de escalonamento de um job-shop cíclico, é alvo de análise e pesquisa de vários estudiosos da área da Pesquisa Operacional, existindo várias vertentes de estudo deste problema. Portanto, dos trabalhos brevemente apresentados na seção anterior, utiliza-se neste trabalho a estratégia proposta no trabalho de Portugal (1999), o qual determina a solução ótima ou o ciclo ótimo de job-shop cíclico com jobs distintos.

A solução obtida pela estratégia desenvolvida por Portugal, conforme descrito anteriormente será chamada de solução de referência. Esta solução consiste nos instantes de início dos jobs, se nenhuma perturbação ocorrer. A seguir será apresentada a formulação proposta no referido trabalho.

3.2.1 – Modelo matemático

Para a estratégia utilizada por Portugal (1999), define-se o job-shop cíclico como um problema de escalonamento cíclico com jobs distintos e fluxo não reentrante. Cada job consiste de um conjunto de tarefas que devem ser realizadas em alguma ordem. Assume-se que cada tarefa deve ser realizada em apenas uma máquina, tempo de processamento fixo, somente um único sucessor no job, uma vez que a tarefa começou a ser processada não pode ser interrompida para execução de outra tarefa.

Assim o problema de job shop cíclico de jobs distintos pode ser modelado através de adaptações feitas às formulações do problema de job shop recorrente (Hanen, 1994) e, do problema de escalonamento cíclico de jobs idênticos (Rao, 1992). Estes dois novos modelos têm como objetivo minimizar o período z o que equivale a maximizar o fluxo τ (onde $\tau = 1/z$).

3.2.1.1 – Simbologia

Os símbolos a seguir são utilizados por Portugal (1999), para a formulação de seus dois métodos.

Z = conjunto dos números inteiros;

N = conjunto dos números naturais;

M = conjunto de máquinas;

m, s = índice de máquina $1 \leq m, s \leq \text{card}(M)$;

O_m = conjunto de tarefas associadas a uma máquina;

J = conjunto de jobs;

J, Q = índices relativos a jobs: $1 \leq J, Q \leq \text{card}(J)$;

O_J = conjunto de tarefas do job J ;

i, j = índices de tarefas associadas a um job J : $1 \leq i, j \leq \text{card}(O_J)$;

m_{ji} = única máquina capaz de processar a tarefa i do job J ;

σ_{ji} = sucessor imediato da tarefa i do job J na máquina m_i ;

s_{ji} = único sucessor da tarefa i no job J ;

p_{ji} = tempo de processamento da tarefa i do job J ;

h_{ji} = altura de recorrência da tarefa i do job J ;

$$h_{ji} = \begin{cases} H & \text{se } i = \text{card}(J), H \in N \\ 0 & \text{caso contrário.} \end{cases}$$

t_{ji} = tempo de início de processamento da tarefa i do job J ;

z = comprimento do ciclo

$T_{ji} = t_{ji} \bmod z$ (resto da divisão de t_{ji} por z)

$x_{ji,Qj}$ = variável binária, indica a seqüência das tarefas de uma mesma máquina;

$$x_{ji,Qj} = \begin{cases} 1 & \text{se } m_{ji} = m_{Qj} \text{ e } i \text{ precede imediatamente } j \text{ na máquina;} \\ 0 & \text{caso contrário.} \end{cases}$$

O_{ji} = distância em ciclos entre as tarefas i e s_{ji} do job J ;

$$O_{ji} = \begin{cases} 0 & \text{se } s_{ji} \text{ é disparado no mesmo ciclo que a tarefa } i \text{ do job } J; \\ 1 & \text{se } s_{ji} \text{ é disparado no ciclo seguinte relativo ao início da tarefa } i \text{ do job } J; \\ 2 & \text{caso contrário.} \end{cases}$$

v_{ji} = constante relativa a tarefa i do job J , para evitar-se sub-ciclos (similar a regra dos nós potenciais em uma rede(Miller et al., 1960));

L_{ji} = indica a última tarefa a ser inicializada no ciclo em um máquina;

$$L_{ji} = \begin{cases} 1 & \text{se } ji \text{ é a última tarefa da máquina } m_{ji} \text{ no ciclo} \\ 0 & \text{caso contrário} \end{cases}$$

$k(ji,Qj)$ = variável inteira que garante a não sobreposição de tarefas.

3.2.1.2 – Formulação matemática

• Job Shop Recorrente Adaptado (JSRA)

O modelo que será apresentado a seguir é uma adaptação feita por Portugal (1999) ao trabalho de Hanen (1994).

- Cada job consiste de um conjunto de tarefas que devem ser realizadas em alguma ordem. Assim estabelecida esta ordem pode-se sugerir um escalonamento cíclico, estabelecendo-se os tempos de inicialização de cada tarefa, de acordo com esta ordem de precedência;

então, com a periodicidade surge uma relação de pseudo-precedência entre a primeira e a última tarefa do job. Pode-se explicitar esta pseudo-precedência determinando que a k -ésima iteração da última tarefa do job J , seja finalizada até a inicialização da $(k+h)$ -ésima iteração da primeira tarefa deste job, onde J é um job qualquer e h é um número inteiro. Estas duas relações, precedência e pseudo- precedência, podem ser representadas pela seguinte equação:

$$t_{J_i} - t_{J_i} \geq p_{J_i} - h_{J_i} \cdot z \quad \forall i, j \in O_J, \quad \forall J \in J \quad (3.1)$$

Esta relação de pseudo-precedência é chamada de altura de recorrência do job.

- O conjunto de tarefas associadas a uma determinada máquina obedece a um certo escalonamento parcial gerado na resolução do problema, porém é necessário garantir a não sobreposição de tarefas na máquina e para isso as seguintes restrições serão utilizadas:

$$t_{Q_j} - t_{J_i} \geq p_{J_i} - k(J_i, Q_j) \cdot z \quad \forall J_i, Q_j \in O_s, \quad \forall s \in M \quad (3.2)$$

$$k(J_i, Q_j) + k(Q_j, J_i) = 1 \quad \forall J_i, Q_j \in O_s, \quad \forall s \in M \quad (3.3)$$

$$k(J_i, Q_j) \in Z \quad \forall J_i, Q_j \in O_s, \quad \forall s \in M \quad (3.4)$$

- Para se ter um equilíbrio na produção dos jobs distintos ao longo do tempo, estipulou-se uma altura de recorrência entre as primeiras tarefas dos jobs distintos e que é mostrada a seguir:

$$t_{J_1} - t_{Q_1} \geq 1 - z \quad \forall J, Q \in J \quad (3.5)$$

- Limites das variáveis

$$t_{J_i} \geq 0 \quad \forall J_i \in O_J, \quad \forall J \in J \quad (3.6)$$

$$z \geq 0 \quad (3.7)$$

A formulação JSRA pode ser resumida como segue:

$$\begin{aligned}
 &\min z \\
 &t_{Jj} - t_{Ji} \geq p_{Ji} - h_{Ji} \cdot z \quad \forall i, j \in O_J, \quad \forall J \in J \\
 &t_{Qj} - t_{Ji} \geq p_{Ji} - k(J_i, Q_J) \cdot z \quad \forall J_i, Q_J \in O_s, \quad \forall s \in M \\
 &k(J_i, Q_J) + k(Q_J, J_i) = 1 \quad \forall J_i, Q_J \in O_s, \quad \forall s \in M \\
 &t_{J1} - t_{Q1} \geq 1 - z \quad \forall J, Q \in J \\
 &t_{Ji} \geq 0 \quad \forall J_i \in O_J, \quad \forall J \in J \\
 &z \geq 0 \\
 &k(J_i, Q_J) \in \mathbb{Z} \quad \forall J_i, Q_J \in O_m, \quad \forall m \in M
 \end{aligned}$$

Exemplo 3.1

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Tempo de processamento das tarefas nas máquinas
A	5	1 3 1 4 1	11 6 15 2 11
B	10	1 5 3 4 3 1 5 3 2 5	12 11 9 3 10 12 15 8 3 14
C	9	3 5 3 4 3 4 2 1 2	6 13 6 3 9 1 4 15 4

Tabela 3.1 – Roteiro de fabricação e tempos de processamento dos jobs.

Utilizando a formulação apresentada anteriormente, tem-se o seguinte diagrama de Gantt para o exemplo 3.1.

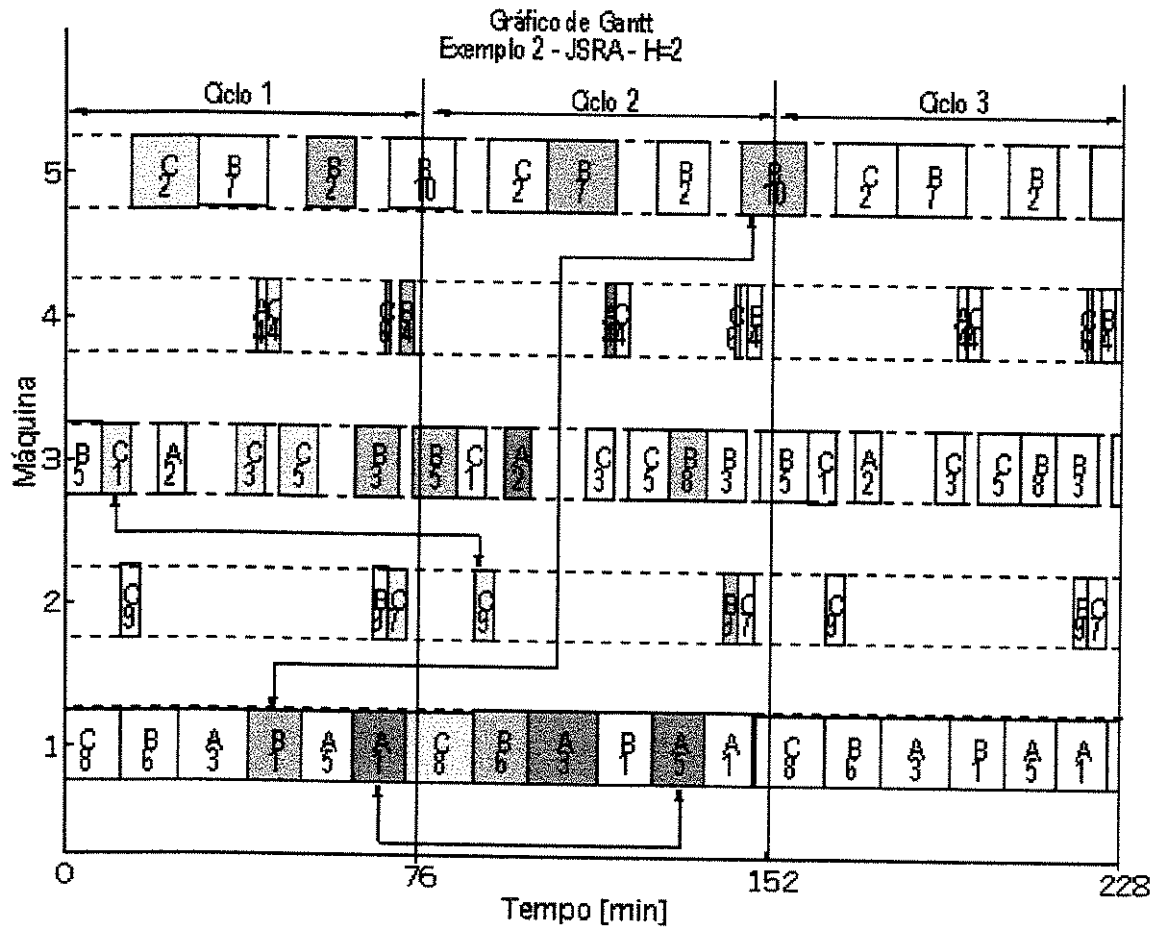


Figura 3.1 – Gráfico de Gantt para o exemplo 3.1.

• Escalonamento Cíclico de Jobs Distintos (ECJD)

O modelo que será apresentado a seguir é uma adaptação feita por Portugal (1999) ao trabalho de Rao (1992).

- O conjunto de tarefas associado a um determinado job deve obedecer a uma certa ordem denominada de precedência no job:

$$T_{SJ_i} - T_{J_i} \geq p_{J_i} - O_{J_i} \cdot z \quad \forall J_i \in O_J, \quad \forall J \in J \quad (3.8)$$

- O conjunto de tarefas associadas a uma determinada máquina obedece a um certo escalonamento parcial gerado na resolução do problema, mas para garantir a não sobreposição de tarefas na máquina a seguinte restrição é utilizada:

$$T_{\sigma J_i} - T_{J_i} \geq p_{J_i} - L_{J_i} \cdot z \quad \forall J_i \in O_s, \quad \forall s \in M \quad (3.9)$$

$$\sum_{J_i \in O_s} L_{J_i} = 1 \quad \forall s \in M \quad (3.10)$$

- Para garantir a não existência de subciclos nas máquinas as seguintes restrições são utilizadas:

$$v_{J_i} - v_{Q_j} + n \cdot x_{J_i, Q_j} \leq (n - 1) + n \cdot (L_{J_i} + L_{Q_j}) \quad n = \text{card}(O_J), \quad \forall J_i, Q_j \in O_s, \quad \forall s \in M \quad (3.11)$$

$$\sum_{J_i \in O_s} x_{J_i, Q_j} = 1 \quad \forall Q_j \in O_s, \quad \forall s \in M \quad (3.12)$$

$$\sum_{Q_j \in O_s} x_{J_i, Q_j} = 1 \quad \forall J_i \in O_s, \quad \forall s \in M \quad (3.13)$$

- Para se controlar o fluxo do conjunto de jobs, será feito o controle de largura de cada job através da altura de recorrência, usando-se a relação de pseudo-precedência discutida anteriormente:

$$T_{J_l} - T_{J_n} \geq p_{J_n} - O_{J_n} \cdot z \quad n = \text{card}(O_J) \quad \forall J \in J \quad (3.14)$$

$$\sum_{i=1, i \in O_J}^n O_{J_i} \leq H \quad \forall J \in J \quad (3.15)$$

$$\sum_{i=1, i \in O_J}^n O_{J_i} \leq 1 \quad \forall J \in J \quad (3.16)$$

onde $n = \text{card}(J)$ e $H \in \mathbb{N}$.

- Para se ter um equilíbrio na produção dos jobs ao longo do tempo propõe-se uma altura de recorrência entre as primeiras tarefas dos jobs distintos:

$$T_{J_l} - T_{Q_l} \geq 1 - z \quad \forall J, Q \in J \quad (3.17)$$

- Limites das variáveis:

$$T_{J_i} \geq 0 \quad \forall i \in O_J, \quad \forall J \in J \quad (3.18)$$

$$T_{Ji} < z \quad \forall i \in O_J, \quad \forall J \in J \quad (3.19)$$

$$O_{Ji} \leq 2 \quad \forall i \in O_J, \quad \forall J \in J \quad (3.20)$$

$$O_{Ji} \in \mathbf{Z} \quad \forall i \in O_J, \quad \forall J \in J \quad (3.21)$$

$$O_{Ji} \in \mathbf{N} \quad \forall i \in O_J, \quad \forall J \in J \quad (3.22)$$

$$z \geq 0 \quad (3.23)$$

E a formulação do ECJD será então:

$$\min z$$

$$T_{S_{Ji}} - T_{Ji} \geq p_{Ji} - O_{Ji} \cdot z \quad \forall J_i \in O_J, \quad \forall J \in J$$

$$T_{\sigma_{Ji}} - T_{Ji} \geq p_{Ji} - L_{Ji} \cdot z \quad \forall J_i \in O_s, \quad \forall s \in M$$

$$\sum_{J_i \in O_s} L_{Ji} = 1 \quad \forall s \in M$$

$$v_{Ji} - v_{Qj} + n \cdot x_{Ji,Qj} \leq (n-1) + n \cdot (L_{Ji} + L_{Qj}) \quad n=\text{card}(O_J), \quad \forall J_i, Q_j \in O_s, \quad \forall s \in M$$

$$\sum_{J_i \in O_s} x_{Ji,Qj} = 1 \quad \forall Q_j \in O_s, \quad \forall s \in M$$

$$\sum_{Q_j \in O_s} x_{Ji,Qj} = 1 \quad \forall J_i \in O_s, \quad \forall s \in M$$

$$T_{J1} - T_{Jn} \geq p_{Jn} - O_{Jn} \cdot z \quad n=\text{card}(O_J) \quad \forall J \in J$$

$$\sum_{i=1, i \in O_J}^n O_{Ji} \leq H \quad \forall J \in J$$

$$\sum_{i=1, i \in O_J}^n O_{Ji} \leq 1 \quad \forall J \in J$$

$$T_{J1} - T_{Q1} \geq 1 - z \quad \forall J, Q \in J$$

$$T_{Ji} \geq 0 \quad \forall i \in O_J, \quad \forall J \in J$$

$$T_{Ji} < z \quad \forall i \in O_J, \quad \forall J \in J$$

$$O_{Ji} \leq 2 \quad \forall i \in O_J, \quad \forall J \in J$$

$$O_{Ji} \in \mathbf{Z} \quad \forall i \in O_J, \quad \forall J \in J$$

$$O_{Ji} \in \mathbf{N} \quad \forall i \in O_J, \quad \forall J \in J$$

$$z \geq 0$$

Exemplo 3.2

Seja o problema de job shop como o apresentado no exemplo 3.1. Aplicando agora a metodologia do ECJD, tem-se o seguinte diagrama de Gantt.

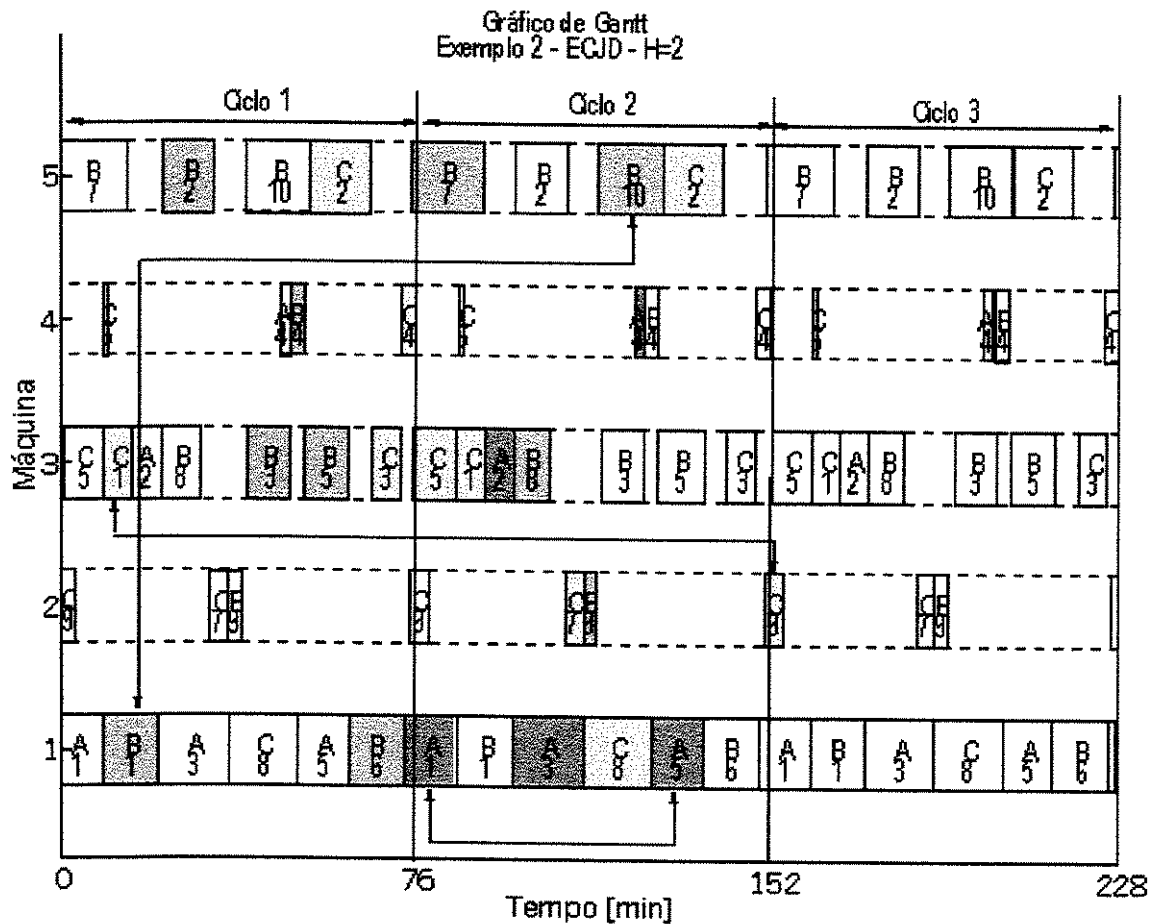


Figura 3.2 - Gráfico de Gantt para o exemplo 3.2.

No trabalho realizado por Portugal (1999), demonstra-se que os dois métodos apresentados anteriormente são equivalentes, isto é, suas restrições definem o mesmo conjunto de soluções factíveis e suas funções objetivo são idênticas. Através desses métodos determinam-se escalonamentos equivalentes com o mesmo período.

3.3 – Soluções de Regime

Considere-se uma planta com características de job shop cíclico com tarefa definida pelos métodos apresentados anteriormente. Dir-se-á que o sistema está em regime permanente quando a solução obtida vem do sistema isento de perturbações e, seus ciclos posteriores ocorrem perfeitamente (constante para todas as tarefas). Uma formulação matemática capaz de obter uma solução factível foi apresentada na seção 3.2.

Conhecer uma solução de regime implica o conhecimento dos tempos de início e fim de cada tarefa de todos os jobs a serem escalonados, assim como o comprimento do período (ciclo). Nos problemas dos capítulos posteriores, será assumido que uma solução de regime para o problema dado é conhecida. Ela poderá ser obtida por qualquer método disponível na literatura para o job shop cíclico com tempos determinísticos. Esta solução de regime também será chamada de solução de referência. Vale ressaltar que a solução de referência necessariamente não é uma solução ótima, isto dependerá da complexidade do problema e do método utilizado, mas esta solução deve ser factível.

Uma solução de referência (regime) assume, que se um job shop cíclico está no j -ésimo ciclo, os jobs do $(j+1)$ -ésimo ciclo só poderão iniciar se todos os jobs do ciclo anterior já foram iniciados e assim sucessivamente para os próximos ciclos.

O job-shop cíclico definido nos parágrafos anteriores, é uma idealização do problema prático. No contexto das operações reais podem ocorrer perturbações de diversos tipos. Imagine que no $(j+k)$ -ésimo ciclo ocorre uma perturbação em seu ciclo de referência; tal perturbação poderia ser a quebra de uma máquina, a falta de alguma matéria prima, falta de espaço para estocagem, etc. A ocorrência de uma destas perturbações, implicará em um ciclo que não é o ciclo de referência, pois haverá espera até que tal perturbação seja corrigida. Imediatamente após a extinção da perturbação, para que se continue a operar dentro do escalonamento de referência, é necessário atrasar de um período igual ao período da perturbação, todas as máquinas da planta. Contudo, esta alternativa pode conduzir a um tempo de ociosidade excessiva no sistema. Contrariamente mantendo-se as outras máquinas funcionando após a extinção da perturbação será, em geral impossível garantir que a ocupação das diversas máquinas se dê conforme previsto no escalonamento cíclico de referência². Neste

ponto é interessante a introdução de alguma metodologia, que interfira no sistema de tal forma que a produção não seja interrompida totalmente, o que garante a continuidade da produção, e que o sistema retorne a solução de referência algum tempo depois de ocorrido a perturbação. O planejamento reativo é, por definição, a tomada de decisões relativas aos ciclos subsequentes a uma ocorrência imprevista, de modo a satisfazer os quesitos expostos anteriormente.

3.4 – Planejamento Reativo de Job Shop Cíclico

Planejamento reativo é a tomada de decisões em problemas de escalonamento com o objetivo de garantir a continuidade da produção após ocorrências imprevistas. Nesta seção restringir-se-á a aplicação do planejamento reativo a problemas de job shop cíclicos determinísticos.

Para o problema de job shop cíclicos determinísticos, pode-se citar dois ramos do planejamento reativo: a utilização de regras de despacho, e a utilização de leis de controle.

3.4.1 – Regras de despacho

São diversos processos heurísticos de decisão, desenvolvidos para a escolha da próxima tarefa a ser processada numa máquina, baseadas no estado das tarefas e planta. Exemplos de regras de despacho utilizadas na literatura segundo Kim (1990), Holthaus e Ziegler (1997), Morton & Pentico(1993) são:

- FIFO (first in, first out): os jobs são processados na ordem em que eles chegam à máquina.
- SPT (shortest processing time): maior prioridade é dada a tarefa com o menor tempo de processamento.
- SLACK(minimum slack time): maior prioridade é dada à tarefa com o menor tempo de folga, sendo o tempo de folga, a diferença entre a data de entrega e o tempo em que a tarefa poderia ser completada não existir atraso.

- S/RMOP(slack per remaining operation): maior prioridade é dada à tarefa que apresenta maior razão entre o tempo de folga e o número de tarefas sucessoras.
- S/OPN(slack per number of operation remaining): maior prioridade é dada à tarefa que apresenta menor razão entre o tempo de folga e o número de tarefas sucessoras.
- S/RMWK(slack per remaining work): maior prioridade é dada à tarefa com a maior razão entre o tempo de folga e o montante de trabalho restante (soma dos tempos de processamento das tarefas sucessoras).
- MDD(modified due date): maior prioridade é dada à tarefa com a menor data de entrega modificada. A data de entrega modificada é o menor tempo entre a data de entrega e o tempo de conclusão da tarefa se não houver atraso.
- MOD(modified operation due date): maior prioridade é dada a tarefa com a menor data de entrega de tarefa modificada. A data de entrega de tarefa modificada é o máximo entre a data de entrega de tarefa e o tempo de conclusão da tarefa.
- NINQ(number of jobs in the queue of its next operation): o job, cuja próxima tarefa será processada pela máquina com a menor fila de jobs, é o selecionado.
- COVERT(cost over time): a maior prioridade é dada ao job que apresenta a maior razão entre o custo estimado por atraso e o tempo de processamento.
- ATC(apparent tardiness cost): similar a COVERT exceto que uma função exponencial do tempo de folga é usada para estimar os custos de atraso.

3.4.2 – Leis de controle

O problema de escalonamento cíclico pode ser visto como um problema de controle onde as variáveis de controle são as decisões de seqüenciamento e alocação; as perturbações são falhas imprevisíveis em máquinas, falta de matéria prima, etc; o estado da planta é definido pelos estoques, ocupação das máquinas e disponibilidade de recursos; o objetivo é determinar as entradas controladas que levam o sistema a ter a evolução desejada no tempo satisfazendo as restrições. Nesse caso trata-se de um sistema a eventos discretos conforme o tratamento dado no capítulo seguinte e em Cassandras (1993).

Bowman e Muckstadt (1993) modelam o escalonamento cíclico como uma cadeia de Markov e usam a teoria ergódica para analisar o desempenho do escalonamento cíclico em ambientes com quebra de máquinas, perda de produtividade e outras falhas imprevisíveis. Eles desenvolvem 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 o gerenciamento de escalonamentos cíclicos.

Zhang e Graves (1997) estudam o escalonamento cíclico em um ambiente estocástico com quebra de máquinas que podem atrasar a execução das tarefas e consequentemente causar desvio do escalonamento cíclico preestabelecido. Eles tentam encontrar escalonamentos cíclicos que sejam menos interrompidos pela quebra de máquinas. Para isto duas hipóteses são consideradas: 1) indiferente do histórico da quebras da máquinas e os atrasos consequentes, as tarefas são sempre processadas na ordem descrita; 2) uma tarefa nunca se inicia antes do tempo de escalonamento prefixado, ainda que a máquina possa estar ociosa.

Groth e Santos-Mendes (1997) modelam e controlam uma classe restrita de problema de escalonamento de job shop, onde assumem uma estrutura de processamento “zero-wait”. Eles propõem um modelo matemático baseado na teoria de dióides, com isto, escrevem equações dinâmicas para o sistema. 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.

Noronha (1999) estuda o problema de escalonamento cíclico quando ele é perturbado com quebra de máquina, falta de matéria, etc, e partindo de uma solução conhecida para o escalonamento cíclico estático determinístico, propõe uma lei de controle baseada na teoria de dióides que deve satisfazer a determinadas propriedades. Esta lei de controle consiste em determinar os instantes de início das tarefas a partir dos instantes de início das tarefas precedentes, com isto, sendo capaz de reestabelecer e manter a solução de referência.

Portanto, tendo-se um problema de job shop cíclico determinístico, do qual obtem-se uma solução de referência, o planejamento reativo oferece alternativas de controle, quando ocorre uma perturbação, objetivando restabelecer a solução de referência.

Neste trabalho a estratégia reativa utilizada para restabelecer a solução de referência, será baseada em uma lei de controle que garante a convergência das soluções, esta lei de controle será introduzida no capítulo 5.

CAPÍTULO 4

DIÓIDES

4.1 - Introdução

Modelos de sistemas dinâmicos a eventos discretos (“DEDS”) têm tido sua utilização gradativamente aumentada na descrição, análise, e controle de processos em ambientes tais como redes de computadores e sistemas de manufatura. O que caracteriza esses processos é o fato de que suas dinâmicas são descritas pela ocorrência de eventos discretos. Portanto, enquanto não ocorrer nenhum evento, o estado de um DEDS permanecerá inalterado. Embora seja conceitualmente simples a descrição da evolução do estado de um DEDS, ele é de difícil análise na presença de incertezas e muitas das ferramentas matemáticas tradicionais têm provado ser inadequadas ou impróprias para resolver muitos problemas de interesse prático (Cassandras & Strickland, 1989).

Embora os DEDS tendam a serem não-lineares quando descritos na álgebra tradicional, existe uma classe de DEDS (sistemas que não apresentam problemas de concorrência) que pode ser descrita por um modelo que é linear na álgebra de dióides.

Dióides são estruturas algébricas cujas propriedades fazem-se presentes na álgebra linear convencional e nos reticulados (“lattices”) dotados com duas operações internas, as quais compartilham propriedades tais como associatividade, comutatividade e distributividade, assim como possuem os elementos identidade e nulo (zero) (Baccelli et al., 1992).

Existe uma gama razoável de dióides na literatura, os quais são definidos com o objetivo de modelar determinados problemas. Um dos objetivos deste capítulo é apresentar um

dióide utilizando uma série de potências com coeficientes booleanos, e as operações internas soma e multiplicação serão as utilizadas normalmente na álgebra de Boole.

Também será utilizada no decorrer deste capítulo a teoria dos grafos. Tal teoria será introduzida com o objetivo de auxiliar a obtenção de resultados, que serão de suma importância na definição de uma lei de controle.

Na seção 4.2 são apresentados os fundamentos matemáticos que caracterizam um dióide, abordando definições, propriedades e teoremas. Na seção 4.3 é apresentado o dióide $Z_{\max}$, assim como a teoria dos grafos, mais especificamente a teoria relacionada com grafos direcionados, pois estas duas teorias possuem características comuns as quais serão importantes no desenvolvimento deste trabalho. Nesta seção também serão introduzidos resultados obtidos por Noronha (1999), que generalizam alguns conceitos clássicos da álgebra max-plus os quais serão também utilizados para a lei de controle que será apresentada no capítulo 5. Na seção 4.4 será proposto um dióide utilizando-se de série de potências, o qual será utilizado neste trabalho para modelar as condições de factibilidade, que farão parte das restrições da lei de controle do capítulo 5.

4.2 – Teoria Geral de Dióides

A seguir são apresentadas definições e propriedades que caracterizam um dióide. Nem todas características são apresentadas, apenas algumas julgadas mais relevantes, maiores detalhes podendo ser vistos em (Cohen et al., 1989; Baccelli et al., 1992).

4.2.1 – Axiomatização e definições básicas

Nesta subseção, será apresentada a teoria axiomática dos dióides, e algumas de suas propriedades.

Definição 4.1 - Um dióide é uma tripla $(D, \oplus, \otimes)$, sendo D um conjunto não vazio, e $\oplus$ e $\otimes$ operações internas chamadas de soma ou adição e produto ou multiplicação, as quais obedecem os seguintes axiomas:

Axioma 1 - Associatividade

$$\begin{aligned} \forall a, b, c \in D; \quad & (a \oplus b) \oplus c = a \oplus (b \oplus c) \\ & (a \otimes b) \otimes c = a \otimes (b \otimes c) \end{aligned}$$

Axioma 2 - Comutatividade da Adição

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

Axioma 3 - Distributividade do produto em relação à soma

$$\begin{aligned} \forall a, b, c \in D; \quad & (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \\ & c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b) \end{aligned}$$

Axioma 4 - Existência do Elemento Nulo (zero) e Elemento Identidade

$$\begin{aligned} \exists \varepsilon \in D: \quad \forall a \in D; \quad & a \oplus \varepsilon = a \\ \exists e \in D: \quad \forall a \in D; \quad & a \otimes e = e \otimes a = a \end{aligned}$$

Axioma 5 - Absorção pelo Elemento Nulo (zero)

$$\forall a \in D; \quad a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$$

Axioma 6 - Idempotência da Adição

$$\forall a \in D; \quad a \oplus a = a$$

Dos axiomas apresentados acima o axioma 6 é o que mais se destaca na estrutura dos dióides, pois devido a ele tem-se que a adição é não cancelativa, ou seja, se $a \oplus b = a \oplus c$ isto não implica que $b = c$ em geral, senão D se reduziria a ε . Este axioma também é a base da introdução de uma relação de ordem, e de sua estrutura de reticulado.

Definição 4.2 - Um dióide é comutativo se sua multiplicação for comutativa, isto é:

$$\forall a, b \in D; \quad a \otimes b = b \otimes a$$

Definição 4.3 - Um dióide é inteiro se $a \otimes b = \varepsilon \Rightarrow a = \varepsilon$ ou $b = \varepsilon$.

Se $a \neq \varepsilon, b \neq \varepsilon$ e $a \otimes b = \varepsilon$ então a e b são chamados de divisores de zero. Portanto, um dióide inteiro é um dióide que não contém divisores de zero.

Se a multiplicação é cancelativa então o dióide é inteiro. Seja, $a \otimes b = \varepsilon$ então $a \otimes b = a \otimes \varepsilon$ o que implica que $b = \varepsilon$ se $a \neq \varepsilon$ pelo cancelamento de a .

Exemplo 4.1

A seguir, expõe-se alguns exemplos de dióides os quais satisfazem os axiomas 1 à 6.

- i) A tripla $(\mathbf{R} \cup \{-\infty\}, \max, +)$ é um dióide. Observa-se que neste caso, $\varepsilon = -\infty$; este fato motiva a inclusão do elemento $-\infty$ no conjunto que define o dióide.
- ii) A tripla $(\mathbf{R} \cup \{-\infty\} \cup \{+\infty\}, \min, +)$ é um dióide. Observa-se que neste caso, $\varepsilon = -\infty$, e $T = +\infty$. Sendo T a soma de todos os elementos do dióide (limitante superior). Estes fatos motivam a inclusão dos elementos $-\infty$ e $+\infty$ no conjunto que define o dióide.
- iii) Sejam $D = \{0,1\}$, $\oplus = \max$ e $\otimes = \min$. Em outras palavras para qualquer $a, b \in D$, tem-se $a \oplus b = \max\{a,b\}$ e $a \otimes b = \min\{a,b\}$. Com isto a tripla $(D, \oplus, \otimes)$ é um dióide.

Os dióides apresentados anteriormente são todos dióides comutativos pois os mesmos satisfazem a condição imposta na definição 4.2.

Definição 4.4 – Um subconjunto S de um dióide D é um sub-dióide de D se:

$$\varepsilon \in S \quad e \quad e \in S$$

S é fechado para soma ($\oplus$) e multiplicação ($\otimes$)

A última declaração significa que:

$$\forall a, b \in S; \quad (a \oplus b) \in S \quad \text{e} \quad (a \otimes b) \in S$$

Exemplo 4.2

O dióide $(N \cup \{-\infty\}, \max, +)$ é um subdióide do dióide apresentado no item i) do exemplo 4.1.

Definição 4.5 – Uma função f de um dióide D para um dióide S é um homomorfismo se e somente se:

$$\forall a, b \in D; \quad f(a \oplus b) = f(a) \oplus f(b) \quad \text{e} \quad f(\varepsilon) = \varepsilon \quad (4.1)$$

$$f(a \otimes b) = f(a) \otimes f(b) \quad \text{e} \quad f(e) = e \quad (4.2)$$

onde:

$\oplus, \otimes, \varepsilon$ e e no lado esquerdo das equações são definidas no dióide D

$\oplus, \otimes, \varepsilon$ e e no lado direito das equações são definidas no dióide S

Definição 4.6 - Uma função f de um dióide D para um dióide S é um isomorfismo se e somente se:

- i) a inversa de f for definida no dióide S ;
- ii) f e f^{-1} forem homomorfismos.

Lema 4.7 – Se f é um homomorfismo de D em S e é uma bijeção, então ela é um isomorfismo.

Demonstração: A demonstração deste lema encontra-se em Bacelli et al. (1992).

Definição 4.8 – Uma relação de equivalência (reflexiva, simétrica e transitiva) definida num dióide D é chamada de congruência se e somente se ela for compatível com $\oplus$ e $\otimes$, isto é:

$$\forall a, b, c \in D; \quad a \equiv b \Rightarrow a \oplus c \equiv b \oplus c \quad (4.3)$$

$$\forall a, b, c \in D; \quad a \equiv b \Rightarrow a \otimes c \equiv b \otimes c \quad (4.4)$$

Lema 4.9 – O quociente de um dióide D por uma congruência (isto é, o conjunto das classes de equivalência) é um dióide onde:

$$[a] \oplus [b] = [a \oplus b] \quad \text{e} \quad [a] \otimes [b] = [a \otimes b]$$

Obs: a notação $[a]$ é a classe de equivalência definida por a

Demonstração: A demonstração deste lema encontra-se em Bacelli et al. (1992).

Definição 4.10 – Um monóide é uma dupla $(D, \oplus)$, sendo D um conjunto não vazio, e $\oplus$ uma operação interna. A qual satisfaz os axiomas da associatividade, comutatividade existência do elemento nulo e idempotência da adição, da definição 4.1.

4.2.2 – Propriedades de reticulados dos dióides

Nesta subseção são apresentadas as características de reticulados que também estão presentes nos dióides, assim como outras propriedades neste contexto.

Teorema 4.11 (*Relação de Ordem*) - Em um dióide D para todo $a, b \in D$, a seguinte equivalência é satisfeita:

$$a = a \oplus b \Leftrightarrow \exists c \text{ tal que } a = b \oplus c.$$

Estas afirmações definem uma ordem parcial denotada por $\geq$ como segue:

$$a \geq b \Leftrightarrow a = a \oplus b \quad (4.5)$$

Essa relação de ordem é compatível com a adição

$$a \geq b \Rightarrow \{\forall c, a \oplus c \geq b \oplus c\}$$

e com a multiplicação;

$$a \geq b \Rightarrow \{\forall c, a \otimes c \geq b \otimes c\}$$

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Lema 4.12 (*Ordem total*) - A relação de ordem $\geq$ definida anteriormente é total se e somente se:

$$\forall a, b \in D; \quad a \oplus b = a \quad \text{ou} \quad a \oplus b = b \quad (4.6)$$

Demonstração: A demonstração deste lema encontra-se em Bacelli et al. (1992).

Definição 4.13 – Um dióide é completo se ele é fechado para somas infinitas e a distributividade (Axioma 3) estende-se para somas infinitas.

Da relação de ordem em dióides, tem-se que um limitante superior de qualquer subconjunto é a soma de todos os seus elementos.

Em um dióide completo o maior elemento do dióide, denotado por ∞ , existe e é igual a soma de todos os elementos do dióide. O maior elemento é sempre absorvido pela adição, desde que $\forall a, \infty \oplus a = \infty$. Também $\infty \otimes \varepsilon = \varepsilon$, devido ao Axioma 5.

Exemplo 4.3

O dióide apresentado no exemplo 4.1 item ii) é um dióide completo, já o apresentado no exemplo 4.1 item i) não o é.

Definição 4.14 – Um dióide é Arquimediano se

$$\forall a \neq \varepsilon, \forall b \in D, \exists c \text{ e } d \in D \quad \text{tais que} \quad a \otimes c \geq b \text{ e } d \otimes a \geq b. \quad (4.7)$$

Teorema 4.15 - Em um dióide completo e Arquimediano a propriedade da absorção é verdadeira, ou seja:

$$\forall a \in D, a \neq \varepsilon: \quad \infty \otimes a = a \otimes \infty = \infty.$$

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Teorema 4.16 - Se um dióide é completo, Arquimediano, e se ele tem uma multiplicação cancelativa, então o dióide é isomórfico a álgebra de Boole.

Demonstração: Se D é completo e arquimediano então:

$$\forall a \neq \varepsilon \quad \infty \otimes a = a \otimes \infty = \infty = b \otimes \infty$$

mas D é cancelativo; logo $a = b$, em particular $a = b = \infty$.

Definição 4.17 – Em um dióide completo, levando-se em conta a existência de um elemento mínimo ε , é possível construir uma nova operação interna ($\wedge$) tal que:

$$\forall a, b \in D; \quad a \geq b \Leftrightarrow a = a \oplus b \Leftrightarrow b = a \wedge b \quad (4.8)$$

Esta operação é associativa, comutativa, idempotente e tem ∞ como elemento neutro, pois:

$$\forall a \in D; \quad \infty \wedge a = a$$

Ela também satisfaz a lei de absorção, pois:

$$\forall a, b \in D; \quad a \wedge (a \oplus b) = a \oplus (a \wedge b) = a$$

4.2.3 – Dióides de matrizes

Partindo de um dióide escalar D , considere-se as matrizes quadradas $n \times n$, cujos elementos pertençam a este dióide. Define-se a soma e o produto de matrizes do modo convencional, respeitando as operações $\oplus$, $\otimes$ definidas no dióide.

Definição 4.18 - O conjunto de matrizes $n \times n$ juntamente com as operações soma e produto constitui um dióide. Assim $D^{n \times n}$ é o dióide das matrizes quadradas.

Definição 4.19 - Uma relação de ordem em $D^{n \times n}$, para quaisquer matrizes A e $B \in D^{n \times n}$ é especificada por:

$$A \geq B \Leftrightarrow [A]_{ij} \geq [B]_{ij} \quad \forall i, j \quad (4.10)$$

Além disso:

$$[A \wedge B]_{ij} = [A]_{ij} \wedge [B]_{ij} \quad \forall i, j \quad (4.11)$$

Definição 4.20 - Uma matriz em $D^{n \times n}$, é dita matriz identidade se e somente se os elementos da diagonal principal forem todos iguais a e , e os demais forem iguais a ε .

Definição 4.21 - Uma matriz em $D^{n \times n}$, é dita matriz nula se e somente se todos os seus elementos forem iguais a ε .

Observa-se do exposto que:

- Se D , é um dióide comutativo então $D^{n \times n}$ não o é necessariamente.
- Se D , é um dióide arquimediano então $D^{n \times n}$ não o é necessariamente
- Se D , é um dióide distributivo então $D^{n \times n}$ também é distributivo.
- Se D , é um dióide completo então $D^{n \times n}$ também é completo.

4.2.4 – Dióides de polinômios e séries de potências

Similarmente ao dióide de matrizes, partir-se-á de um dióide escalar D , visando a obtenção de novos dióides constituídos pelo conjunto de polinômios e séries de potências em uma ou várias variáveis com coeficientes em D . Se várias variáveis estão envolvidas será considerada a situação de variáveis comutativas, ou seja, $z_1 \otimes z_2 = z_2 \otimes z_1$. Os expoentes das variáveis são de tal forma que eles pertencem ao conjunto N ou Z .

Definição 4.22 – Uma série de potências formal em n variáveis (comutativas) com coeficientes em D é uma função $f: N^n \rightarrow D$ ou $f: Z^n \rightarrow D$ tal que $\forall k = (k_1, k_2, \dots, k_n) \in N^n$ ou Z^n , $f(k)$ representa o coeficiente de $z_1^{k_1} \dots z_n^{k_n}$. Uma representação equivalente é:

$$f = \bigoplus_{k \in N^n \text{ ou } Z^n} f(k_1 \dots k_n) z_1^{k_1} \dots z_n^{k_n} \quad (4.12)$$

Definição 4.23 – O suporte de uma série de potências f em n variáveis é o conjunto:

$$\text{Supp}(f) = \{k \in Z^n / f(k) \neq \varepsilon\} \quad (4.13)$$

Definição 4.24 – O grau de uma série de potências f em n variáveis é:

$$\deg(f) = \text{limitante superior de } \text{supp}(f) \text{ em } \mathbb{Z}^n \cup \{+\infty\} \cup \{-\infty\} \quad (4.14)$$

Definição 4.25 – A valoração de uma série de potências f em n variáveis é:

$$\text{val}(f) = \text{limitante inferior de } \text{supp}(f) \text{ em } \mathbb{Z}^n \cup \{+\infty\} \cup \{-\infty\} \quad (4.15)$$

Definição 4.26 – Um polinômio é uma série de potências com suporte finito.

Observa-se que se f é um polinômio monovariável ($n = 1$) a definição de grau de polinômio coincide com a definição clássica, isto é, o maior expoente em z .

Definição 4.27 – O conjunto de séries de potências juntamente com as operações soma e multiplicação definidas a seguir:

$$\begin{aligned} f \oplus g: (f \oplus g)(k) &= f(k) \oplus g(k) \\ f \otimes g: (f \otimes g)(k) &= \bigoplus_{i+j=k} f(i) \otimes g(j) \end{aligned}$$

constitui um dióide. Assim $D[[z_1, \dots, z_n]]$, é dito dióide de série de potências.

Definição 4.28 - O conjunto de polinômios juntamente com as operações soma e multiplicação definidas a seguir:

$$\begin{aligned} f \oplus g: (f \oplus g)(k) &= f(k) \oplus g(k) \\ f \otimes g: (f \otimes g)(k) &= \bigoplus_{i+j=k} f(i) \otimes g(j) \end{aligned}$$

constitui um dióide. Assim $D[z_1, \dots, z_n]$, é dito dióide de polinômios.

Definição 4.29 - Sejam $D[z, \dots, z_n]$, e $D[[z_1, \dots, z_n]]$, dióides. Uma relação de ordem em $D[z_1, \dots, z_n]$, e $D[[z_1, \dots, z_n]]$, para quaisquer dois polinômios. P e $Q \in D[z_1, \dots, z_n]$ e quaisquer duas séries de potências S e $R \in D[[z_1, \dots, z_n]]$ é especificada por:

$$P > Q \Leftrightarrow \{p(k) \geq q(k), \forall k\} \quad (4.16)$$

$$S > R \Leftrightarrow \{s(k) \geq r(k), \forall k\} \quad (4.17)$$

Observa-se do exposto que:

- Se D , é um dióide comutativo então $D[[z_1, \dots, z_n]]$ também é comutativo.
- Se D , é um dióide distributivo então $D[[z_1, \dots, z_n]]$ também é distributivo.
- Se D , é um dióide completo então $D[[z_1, \dots, z_n]]$ também é completo, mas $D[z_1, \dots, z_n]$ não o é.
- Se D , é um dióide arquimediano então $D[z_1, \dots, z_n]$, e $D[[z_1, \dots, z_n]]$ necessariamente não são, isto só é verdadeiro se os expoentes pertencem a N .

4.3 - Álgebra Max-Plus e Teoria dos Grafos

A utilização da álgebra max-plus, na descrição de certos sistemas a eventos discretos, possui uma característica muito desejada nos modelos matemáticos, que é a linearidade. Além disto, vários outros conceitos conhecidos na teoria de sistemas lineares tradicional também se fazem presentes nesta álgebra não usual, tais como autovalor, autovetor, etc.

A teoria dos grafos direcionados permite inúmeras interpretações de conceitos da álgebra max-plus, facilitando a sua compreensão. Nesta seção serão apresentados alguns conceitos da teoria de grafos e sua conexão com a álgebra max-plus. Também serão apresentados os resultados obtidos por Noronha (1999), que correspondem a generalização de conceitos clássicos da álgebra max-plus.

4.3.1 - Dióide max-plus ou álgebra max-plus

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

A álgebra max-plus é talvez dentro da estrutura chamada dióide, a instância mais utilizada e conhecida, pois ela apresenta características semelhantes a álgebra linear convencional. Como qualquer dióide possui algumas características diferentes tais como: a não existência do elemento inverso em relação à adição, a idempotência da adição, etc. A seguir serão brevemente apresentadas as principais características desta álgebra.

Definição 4.30 – Seja o conjunto $D = \mathbb{Z} \cup \{-\infty\}$, juntamente com as seguintes operações;

$$\forall a, b \in D; \quad a \oplus b = \max(a, b) \quad (4.18)$$

$$\forall a, b \in D; \quad a \otimes b = a + b \quad (4.19)$$

esta estrutura com $\varepsilon = -\infty$ e $e = 0$, constitui o dióide max-plus que daqui para frente será denotado por $\mathbb{Z}_{\max}$.

As operações $\oplus$ e $\otimes$ podem ser estendidas a matrizes como a seguir:

$$\forall A, B \in \mathbb{Z}_{\max}^{n \times p}; \quad [A \oplus B]_{ij} = [A]_{ij} \oplus [B]_{ij} \quad (4.20)$$

$$\forall A \in \mathbb{Z}_{\max}^{n \times p} \text{ e } \forall B \in \mathbb{Z}_{\max}^{p \times q} \quad [A \otimes B]_{ij} = \bigoplus_{k=1}^p ([A]_{ik} \otimes [B]_{kj}) \quad (4.21)$$

4.3.2 – Teoria dos grafos

Utilizando-se a teoria dos grafos é possível modelar várias estruturas de tal forma que estas apresentem modelos simples e de fácil manuseio. Portanto, neste parágrafo serão apresentados conceitos de grafos que posteriormente serão combinados com matrizes escritas na álgebra max-plus.

Definição 4.31 – Um grafo direcionado G é um par (N, E) onde N é um conjunto de nós (vértices) e E é um conjunto de pares ordenados de nós, chamados arcos direcionados.

Definição 4.32 – Um caminho p é uma sequência de nós $(i_1, i_2, \dots, i_p)$, $p > 1$, tais que entre cada par de nós sucessivos o arco (i_j, i_{j+1}) existe. Um caminho elementar é um caminho no qual nenhum nó aparece mais que uma vez.

Definição 4.33 – Um grafo G é dito ser conectado se para todos os pares de nós (vértices) i e j existe um caminho ligando i e j ou existe um caminho ligando j e i .

Definição 4.34 – Um grafo G é dito ser fortemente conectado se para qualquer dois diferentes nós (vértices) i e j existe um caminho de i para j .

Definição 4.35 – Os subgrafos $G_i = (N_i, E_i)$ correspondendo a classes de equivalência determinadas por “existe um caminho nos dois sentidos” são os subgrafos maximais fortemente conectados de G (s.m.f.c).

Definição 4.36 – Considere-se um grafo $G = (N, E)$ e associe-se um elemento $a_{ij} \in Z_{\max}$ a cada arco $(j, i) \in E$; então G é chamado um grafo valorado. O valor a_{ij} é chamado de peso do arco (j, i) .

Definição 4.37 – O grafo de precedência de uma matriz quadrada $A \in Z^{n \times n}$, com elementos em $Z_{\max}$, é um grafo direcionado valorado com n nós e arcos (j, i) se $a_{ij} \neq \varepsilon$. Neste caso o peso deste arco corresponde ao valor numérico a_{ij} . O grafo de precedência é denotado por $G(A)$.

4.3.3 – Grafos e matrizes em $Z_{\max}$

Existe uma rica relação entre grafos e matrizes, sendo possível utilizar matrizes para representar grafos, onde os elementos das matrizes correspondem aos arcos dos grafos, e vice-versa. A seguir serão apresentados algumas relações entre grafos direcionados e matrizes. O objetivo principal nesta seção é o de introduzir conceitos tais como autovalor, autovetor e a convergência do limite de A^k quando k tende a infinito. Em seguida será apresentada uma generalização deste resultado clássico proposto no trabalho de Noronha (1999).

Definição 4.38 – A matriz $A \in Z^{n \times n}$, é chamada irredutível se não existe nenhuma matriz P tal que a matriz $\tilde{A}$, definida por $\tilde{A} = P^T A P$, tenha uma estrutura triangular superior.

Para determinar se A é irredutível, pode-se calcular A^+ que é definida pela equação a seguir:

$$A^+ \stackrel{\text{def}}{=} A \oplus A^2 \oplus \dots \oplus A^n \oplus A^{n+1} \oplus \dots \quad (4.22)$$

Se A^+ possuir todos os elementos diferentes de ε , então A é irredutível (Bacelli et al. 1992).

A matriz A^+ pode ser escrita como $A^+ = A \otimes A^*$, onde A^* é definida pela expressão a seguir:

$$A^* \stackrel{\text{def}}{=} e \oplus A \oplus A^2 \oplus \dots \oplus A^n \oplus A^{n+1} \oplus \dots \quad (4.23)$$

onde:

e = a matriz identidade no contexto max-plus.

Teorema 4.39 – Uma condição necessária e suficiente para que uma matriz $A \in \mathbb{Z}^{n \times n}$, seja irredutível é que seu grafo de precedência seja fortemente conectado.

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Definição 4.40 – Uma matriz $A \in \mathbb{Z}^{n \times n}$, é aperiódica se existe um inteiro r tal que para todo $s \geq r$ e para todo i, j , $[A^s]_{ij} \neq \varepsilon$.

Definição 4.41 – Seja $G = (N, E)$ um grafo valorado com n nós. Seja uma matriz $A \in \mathbb{Z}^{n \times n}$. O valor numérico de a_{ij} é o peso do arco (j, i) . Se tal arco não existe $a_{ij} = \varepsilon$. O peso de um caminho entre dois nós é a soma dos pesos dos arcos que compõem o caminho. O caminho crítico entre dois nós é o caminho sem repetição de nós com o maior peso.

Vale ressaltar que o caminho crítico pode ser determinado, como a seguir:

→ Caminho com apenas um ramo: a_{ij}

→ Caminho com dois ramos: $\max_{k=1, \dots, n} (a_{ik} + a_{kj}) = \bigoplus_{k=1}^n (a_{ik} \otimes a_{kj}) = [A^2]_{ij}$

→ Caminho com n ramos: $[A^*]_{ij} \quad n \rightarrow +\infty$

Definição 4.42 – O peso médio de um caminho é definido como a soma dos pesos dos arcos individuais deste caminho, dividido pelo comprimento deste caminho, algebricamente:

$$\rho = \frac{\text{soma dos pesos dos arcos}}{\text{comprimento do caminho}} \quad (4.24)$$

Se tal caminho é um circuito (caminho fechado) seu peso médio (ρ) é chamado de ciclo médio.

O máximo ciclo médio de todos os circuitos de comprimento j de um grafo G , é dado pela seguinte expressão (Bacelli et al., 1992):

$$\lambda = \bigoplus_{j=1}^n (\text{traço}(A^j))^{1/j} \quad (4.25)$$

onde, para uma exemplificação considere $j = 2$, então:

$$(A^2)^{1/2} = \sqrt[2]{A \otimes A} = \frac{A \otimes A}{2}$$

Todos os circuitos com ciclo médio igual ao máximo ciclo médio são chamados circuitos críticos. O teorema a seguir (teorema de Karp) fornece uma expressão para o cálculo do máximo ciclo médio.

Teorema 4.43 (Teorema de Karp) – Dada uma matriz $A \in \mathbb{Z}^{n \times n}$ com grafo de precedência $G(N,E)$, o máximo ciclo médio é dado por:

$$\lambda = \max_{i=1, \dots, n} \min_{k=0, \dots, n-1} \frac{[A^n]_{ij} - [A^k]_{ij}}{n - k}, \quad \forall j \quad (4.26)$$

Nesta equação, A^n e A^k são utilizadas no contexto $\mathbb{Z}_{\max}$; as demais operações são as convencionais.

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Definição 4.44 – Um autovalor (λ) de uma matriz A em $Z_{\max}$ é o valor não nulo tal que:

$$A \otimes y = \lambda \otimes y \quad (4.27)$$

para algum y também não nulo.

Teorema 4.45 – Se a matriz A é irredutível, ou equivalentemente, se $G(A)$ é fortemente conectado, existe um e somente um autovalor. Este autovalor é igual ao máximo ciclo médio do grafo.

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Definição 4.46 – Uma matriz $A \in Z^{n \times n}$ é dita normalizada se e somente se seu ciclo médio máximo é igual a e . Normalizar uma matriz A qualquer, significa dividir todos os seus elementos pelo máximo ciclo médio.

Definição 4.47 – Um circuito ξ do grafo de precedência $G(A)$ é chamado de circuito crítico se seu ciclo médio é igual ao máximo ciclo médio.

Definição 4.48 – O grafo crítico $G^c(A)$ consiste dos nós e arcos de $G(A)$ que pertencem a algum circuito crítico de A .

Definição 4.49 – A ciclicidade de um subgrafo maximal fortemente conectado (s.m.f.c) é o máximo divisor comum (mdc) dos comprimentos de todos os seus circuitos. A ciclicidade $c(G)$ de um grafo G é o mínimo múltiplo comum (mmc) das ciclicidades de todos os seus s.m.f.c.

Definição 4.50 – Uma matriz Q satisfazendo $A \otimes Q = Q \otimes A = Q^2$ é chamada projetor espectral de A associado ao autovalor e .

Proposição 4.51 – Se uma matriz possui um único autovetor, consequentemente o projetor espectral desta também possui um único autovetor.

Demonstração: Seja y_1 autovetor de A . Suponha por absurdo que o projetor espectral de A tenha outro autovetor não proporcional ao autovetor de A , ou seja: $Q \otimes y_2 = y_2$ porém $A \otimes y_2 \neq y_2$ e $y_2 \neq \delta \otimes y_1$. Pela definição de projetor espectral tem-se:

$$\begin{aligned} Q \otimes A &= A \otimes Q = Q \otimes Q \Rightarrow Q \otimes A \otimes y_2 = A \otimes Q \otimes y_2 = Q \otimes Q \otimes y_2 \Rightarrow \\ &Q \otimes A \otimes y_2 = A \otimes y_2 = y_2 \end{aligned}$$

o que é uma contradição (Bacelli et al., 1992).

Proposição 4.52 – Qualquer vetor x , multiplicado pelo projetor espectral de uma matriz A com um único autovetor y , tem como resultado um vetor proporcional ao autovetor desta matriz, ou seja, $Q \otimes x = \delta \otimes y$, onde Q é projetor espectral de A .

Demonstração: Seja A uma matriz com um único autovetor y . Tem-se que $Q_i = A^+_{\cdot i} \otimes A^+_{i \cdot}$, com $A^+_{\cdot i}$ ($A^+_{i \cdot}$) representando a i -ésima coluna (linha) de A^+ , projetor espectral de A . Sendo $A^+_{\cdot i}$ um autovetor de A tem-se que $Q_i \otimes x = A^+_{\cdot i} \otimes A^+_{i \cdot} \otimes x$. O produto $A^+_{\cdot i} \otimes x$ tem como resultado um escalar, isto é $A^+_{\cdot i} \otimes x = \delta_i$. Tem-se então que $Q_i \otimes x = A^+_{\cdot i} \otimes \delta_i = \delta_i \otimes y$ (Bacelli et al., 1992).

Teorema 4.53 – Uma condição necessária e suficiente para termos $\lim_{k \rightarrow +\infty} A^k = Q$ é que a ciclicidade de cada subgrafo maximal fortemente conectado (s.m.f.c) de $G^c(A)$ seja igual a 1.

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Teorema 4.54 – Suponha que $G(A)$ seja fortemente conectado. Então existe L tal que:

$$\forall k \geq L; \quad A^k = Q$$

se e somente se a ciclicidade de cada subgrafo maximal fortemente conectado (s.m.f.c) de $G^c(A)$ for igual a 1.

Demonstração: A demonstração deste teorema encontra-se em Bacelli et al. (1992).

Da proposição 4.51 e do teorema 4.52, pode-se inferir o seguinte resultado:

$$\lim_{k \rightarrow +\infty} A^k \otimes x = Q \otimes x = \delta \otimes y \quad (4.28)$$

onde:

δ é um escalar

y é o autovetor de A .

4.3.4 – Produtos generalizados de matrizes

Uma generalização do resultado $\lim_{k \rightarrow +\infty} A^k \otimes x = \delta \otimes y$ é proposta no trabalho de Noronha(1999). Neste trabalho supõe-se a existência de um conjunto de matrizes $B_l, l \in \{1, \dots, m\}$, tal que se o conjunto de matrizes satisfizer a determinadas condições então:

$$\left(\lim_{k \rightarrow +\infty} \bigotimes_{i=1}^k (B_{l(i)}) \right) \otimes t_0 = \delta \otimes t^\# \quad (4.29)$$

Deve-se observar que, no produtório acima, a seqüência de matrizes B_l utilizadas é arbitrária, ou seja, a função $l(i)$ é qualquer. E que $\delta \otimes t^\#$ corresponde a um deslocamento de δ no autovetor $t^\#$.

Definição 4.55 – Dado um vetor $t^\# \in \mathcal{Z}^n$, um conjunto finito de matrizes é chamado $t^\#$ -consistente desde que seus elementos e toda matriz resultante de qualquer produto de seus elementos tenham autovalor igual a e e possuam um único autovetor $t^\#$.

Teorema 4.56 – Seja um conjunto $t^\#$ -consistente de matrizes $B_l, l \in \{1, \dots, m\}$. Então:

$$\left(\lim_{k \rightarrow +\infty} \bigotimes_{i=1}^k (B_{l(i)}) \right) \otimes t_0 = \delta \otimes t^\# \quad (4.30)$$

onde:

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

δ é um escalar;

t_0 é um vetor arbitrário $\in \mathbb{Z}^n$.

Demonstração: A demonstração deste teorema encontra-se em Noronha (1999).

Definição 4.57 – Dado um vetor $t^\# \in \mathbb{Z}^n$, define-se sua matriz associada como segue:

$$B^\# = -t^\# \otimes t^{\#T} \quad (4.31)$$

onde:

a multiplicação ($\otimes$) por $-t^\#$ corresponde a subtração na álgebra convencional.

Esta matriz possui algumas propriedades importantes para o estudo da convergência do limite definido na equação (4.30). Particularmente é imediato concluir que seu autovalor é e e seu autovetor é $t^\# = [[t^\#]_1 \ [t^\#]_2 \ \dots \ [t^\#]_n]^T$.

Proposição 4.58 – Para a matriz $B^\#$ definida na equação (4.31) as seguintes asserções são válidas:

- i) $B^\# = B^* = B^+$;
- ii) $B^\#$ é projetor espectral dele próprio, isto é, $B^\# \otimes Q = Q \otimes B^\# = Q^2$;
- iii) $\forall t_0, \exists \delta$, tal que: $B^\# \otimes t_0 = \delta \otimes t^\#$.

Demonstração: A demonstração desta proposição encontra-se em Noronha (1999).

Definição 4.59 – Um conjunto de matrizes $\mathcal{B} = \{B_1, \dots, B_m\}$ é dito sub-crítico em relação $t^\#$ se para todo $B_l \in \mathcal{B}$:

- i) autovalor de cada B_l igual a e ;
- ii) autovetor de B_l igual a $t^\#$ e é único;
- iii) não existe nenhum ciclo crítico no grafo de precedência $G(B_l)$ passando por i ou seja, $[B_l]_{ij} < [B^\#]_{ij}, \forall j = 1, \dots, n$;

- iv) existe um ciclo crítico no grafo de precedência $G(B_i)$ passando por i ou seja, $[B_i]_{ij} = [B^\#]_{ij}, \forall j = 1, \dots, n$.

O conjunto de matrizes anteriormente definido coincide com o conjunto das matrizes possuindo pelo menos uma linha igual a uma linha da matriz $B^\#$ e os demais valores estritamente menores que os respectivos de $B^\#$.

Teorema 4.60 – Qualquer conjunto sub-crítico em relação a $t^\#$, é $t^\#$ -consistente.

Demonstração: A demonstração desta proposição encontra-se em Noronha (1999).

4.4 - O Dióide $\mathcal{P}$

Esta seção tem por objetivo apresentar o dióide que será utilizado no capítulo seguinte para análise de factibilidade de soluções.

A análise de factibilidade de soluções em regime transitório pode ser feita utilizando um dióide (denotado por $\mathcal{J}$) proposto inicialmente por Groth e Santos-Mendes (1997). Contudo, visando uma padronização da notação, é mais conveniente a utilização de um dióide escrito em forma de série de potências a partir dos resultados apresentados na seção 4.2. Portanto, no que se segue, apresentar-se-á inicialmente o dióide $\mathcal{J}$ seguido do dióide $\mathcal{P}$, isomórfico ao primeiro, e que pode ser obtido como uma generalização do dióide booleano.

4.4.1 – O dióide $\mathcal{J}$

Para a construção do dióide proposto em Groth e Santos-Mendes (1997), define-se inicialmente um monóide $(\mathcal{X})$ e uma classe de operadores $(\mathcal{P})$ definida sobre os elementos de $\mathcal{X}$.

Para obter o dióide, então define-se a multiplicação em $\mathcal{F}$ como a composição de operadores e adição em $\mathcal{F}$ como a operação induzida pela operação definida em $\mathcal{X}$.

Definição 4.61 – Sendo $\mathcal{X}$ a classe de todas as funções $m: Z \rightarrow \{0, 1\}$ do tipo descritas a seguir.

$$m(\tau) = \begin{cases} 1 & \text{se } \tau \geq \tau^* \\ 0 & \text{se } \tau \leq \tau^0 \\ 0 \text{ ou } 1 & \text{se } \tau^0 < \tau < \tau^* \end{cases} \quad (4.32)$$

onde:

$$0 \leq \tau^* - \tau^0 \leq \Delta$$

O gráfico a seguir ilustra um elemento típico pertencente a $m(\tau)$.

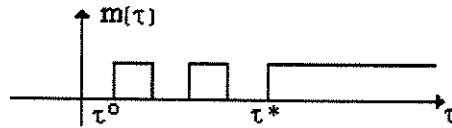


Figura 4.1 – Elemento típico de $m(\tau)$.

Sejam m_1 e m_2 dois elementos de $\mathcal{X}$. Define-se adição neste conjunto como a seguinte operação interna:

$$(m_1 \oplus m_2)(\tau) = m_1(\tau) \cdot m_2(\tau) \quad (4.33)$$

onde a multiplicação no termo a direita é a multiplicação booleana tradicional.

Sejam $m_e(\tau)$ e $m_\infty(\tau)$ elementos particulares de $\mathcal{X}$ definidos como a seguir:

$$m_e(\tau) = 1, \quad \text{para todo } \tau \in Z \quad (4.34)$$

$$m_\infty(\tau) = 0, \quad \text{para todo } \tau \in Z \quad (4.35)$$

$(\mathcal{X}, \oplus)$ é um monóide comutativo completo, m_e é o elemento neutro com respeito a adição, m_∞ é a soma de todos os elementos de $\mathcal{X}$. A operação definida pela equação (4.33) é associativa e comutativa, e:

$$m \oplus m_e = m \quad \text{para todo } m \in \mathcal{X} \quad (4.36)$$

Além disso, a adição em $\mathcal{X}$ é idempotente, pois para todo $m \in \mathcal{X}$:

$$m \oplus m = m \quad (4.37)$$

Se $\mathcal{F}$ é a classe de todos operadores $f: \mathcal{X} \rightarrow \mathcal{X}$. Se f_1 e f_2 são elementos de $\mathcal{F}$ e se m é qualquer elemento de $\mathcal{X}$. Define-se o produto entre elementos de $\mathcal{F}$ como:

$$(f_1 \otimes f_2)(m(\tau)) = f_1(f_2(m(\tau))) \quad (4.38)$$

Para qualquer $f \in \mathcal{F}$ e $m \in \mathcal{X}$, a seguinte notação será utilizada, desde que ela não traga ambiguidade:

$$f(m(\tau)) = f \otimes m \quad (4.39)$$

Deve-se prestar atenção especial aos elementos de $\mathcal{F}$ que correspondem a uma translação da função m . Para esses elementos será introduzida uma notação especial. Se $\alpha \in \mathbb{Z} \cup \{-\infty, \infty\}$; então:

$$\alpha \otimes m(\tau) = m(\tau - \alpha) \quad (4.40)$$

Seguindo esta notação o operador f_e , associado ao número real zero, é o elemento neutro com respeito ao produto (elemento unitário), isto é, para qualquer $f \in \mathcal{F}$:

$$f_e(m) = 0 \otimes m = m \quad (4.41)$$

$$f \otimes 0 \otimes m = 0 \otimes f \otimes m = f \otimes m \quad (4.42)$$

Define-se a adição em $\mathcal{F}$, a partir da definição que já tinha sido proposta para $\mathcal{X}$ na equação (4.33). Em outras palavras a adição em $\mathcal{F}$ é induzida de $\mathcal{X}$, assim pela definição:

$$(f_1 \oplus f_2)(m) = f_1(m) \oplus f_2(m) \quad (4.43)$$

O elemento neutro com respeito a esta operação (elemento nulo) é definido como a seguir:

$$f_e(m) = m_e \quad \text{para todo } m \in \mathcal{X} \quad (4.44)$$

onde m_e é definido pela equação (4.34). Observe que o operador f_e é do tipo descrito pela equação (4.40), com $\alpha = -\infty$, desde que para algum $m \in \mathcal{X}$ e para k suficientemente grande, $m(k) = 1$.

Também pode-se definir outro elemento especial de $\mathcal{F}$ como:

$$f_\infty(m) = m_\infty \quad \text{para todo } m \in \mathcal{X} \quad (4.45)$$

Agora, se $\mathcal{J}$ é o subconjunto de $\mathcal{F}$ definido como a seguir: $f_e \in \mathcal{J}$; $f_\infty \in \mathcal{J}$; se f é do tipo definido na equação (4.40) e $f \in \mathcal{J}$; se $f_1 \in \mathcal{J}$ e $f_2 \in \mathcal{J}$ então $f_1 \oplus f_2 \in \mathcal{J}$.

Portanto, o conjunto $\mathcal{S}$ com as operações $\oplus$ e $\otimes$ definidas respectivamente pelas equações (4.43) e (4.38) é um dióide completo. Os elementos nulo e unitário são respectivamente f_∞ (equação 4.44) e f_e (equação 4.41). A soma de todos os elementos em $\mathcal{S}$ é o elemento f_∞ (equação 4.44).

4.4.2 – O dióide $\mathcal{P}$

Na seção 4.4.1 foi mostrada uma maneira de se obter um dióide a partir de um monóide. O procedimento a seguir utilizará a técnica de construção de dióides, a partir de séries de potências, para obter um resultado similar.

Definição 4.62 – Define-se o dióide Booleano B , como sendo $\langle B, \oplus, \otimes \rangle$ onde $B = \{0, 1\}$, $\oplus = \max$ e $\otimes = \min$. Em outras palavras para qualquer $a, b \in B$, tem-se $a \oplus b = \max\{a, b\}$ e $a \otimes b = \min\{a, b\}$.

É possível definir um novo dióide de acordo com os resultados apresentados na seção 4.2.4.

Denotar-se-á por $\mathcal{P}$ o dióide em série de potências construído a partir de B , com uma única variável.

$$\mathcal{P} = B[[\tau]] \quad (4.46)$$

Definição 4.63 – Seja $f = \bigoplus_{k \in Z} f(k_1 \dots k_n) \tau_1^{k_1} \dots \tau_n^{k_n}$ um polinômio de $\mathcal{P}$. Define-se seu complemento (notação $\bar{f}$), como sendo o polinômio:

$$\bar{f} = \bigoplus_{k \in Z} \bar{f}(k_1 \dots k_n) \tau_1^{k_1} \dots \tau_n^{k_n}$$

onde: $\bar{f}(k) = \begin{cases} 1 & \text{se } f(k) = 0 \\ 0 & \text{se } f(k) = 1 \end{cases}$

Definição 4.64 – A valoração complementar (ou c-valoração) de um polinômio da classe $\mathcal{P}$ é a valoração de seu complemento.

Definição 4.65 – Dado um número $\Delta \in \mathbb{N}$. Define-se $\mathcal{S}_\Delta$ como a classe de todas as séries de potências, cujos coeficientes das variáveis são do tipo descritos a seguir.

$$m(k) = \begin{cases} 0 & \text{se } k \geq K^* \\ 1 & \text{se } k \leq K^0 \\ 0 \text{ ou } 1 & \text{se } K^0 < k < K^* \end{cases} \quad (4.47)$$

onde:

$$0 < K^* - K^0 \leq \Delta$$

O gráfico a seguir ilustra um elemento típico pertencente a $m(k)$.

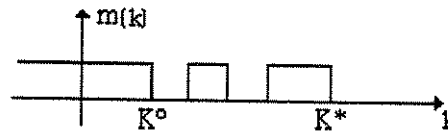


Figura 4.2 – Elemento típico de $m(k)$.

Para a classe de série de potências $\mathcal{S}_\Delta$, o elemento neutro desta classe é definido como a seguir:

$$m_e(k) = 0, \quad \text{para todo } k \in \mathbb{Z}; \text{ observe que neste caso } K^0 = K^* = -\infty \quad (4.48)$$

Outro elemento particular de $\mathcal{S}_\Delta$ é definido como a seguir:

$$m_\infty(k) = 1, \quad \text{para todo } k \in \mathbb{Z}; \text{ observe que neste caso } K^0 = K^* = +\infty \quad (4.49)$$

Observa-se que m_e é o elemento neutro com respeito a adição, e que m_∞ é a soma de todos os elementos de $\mathcal{S}_\Delta$. E $\mathcal{S}_\Delta$ é fechada em relação à adição e em relação à multiplicação. Porém tem-se que na definição do conjunto de série de potências $\mathcal{S}_\Delta$, o elemento identidade da multiplicação não pertence a $\mathcal{S}_\Delta$. Por isto, este subconjunto não pode ser definido como um sub-dióide do dióide mais geral de série de potências.

É possível entretanto definir o monóide $(\mathcal{S}_\Delta, \oplus)$ que tem propriedades idênticas às do monóide $\mathcal{R}$. Além disso é possível utilizar este monóide para induzir um dióide análogo ao dióide $\mathcal{R}$. O resultado é o próprio dióide $\mathcal{P}$.

CAPÍTULO 5

ESTRATÉGIA REATIVA PARA O TRANSITÓRIO DO JOB SHOP CÍCLICO

5.1- Introdução

O objetivo deste capítulo é apresentar um modelo matemático para a análise e síntese de controladores para o problema de job shop cíclico quando ele está em regime transitório e conhece-se uma solução de referência.

Vale ressaltar que na obtenção da solução de referência, esperas entre tarefas de um mesmo job são introduzidas com o propósito de satisfazer a função objetivo. Se tais esperas são mantidas na análise do regime transitório, acaba-se restringindo o espaço de soluções impondo uma estrutura rígida a cada job. Nesta estrutura rígida as soluções são obtidas se e somente se, os jobs com esperas entre tarefas definidas na solução de referência são mantidas fixas e satisfazem à restrição de factibilidade. Há porém uma alternativa que possibilita um maior espaço de soluções. Esta alternativa consiste em considerar cada tarefa como um job, com isto eliminam-se as esperas introduzidas na obtenção da solução de referência. Isto possibilita uma maior flexibilidade das tarefas nos jobs, ressaltando-se que as restrições de precedência entre as tarefas de um mesmo job devem ser respeitadas.

Na abordagem proposta neste trabalho são consideradas as duas possibilidades. Por um lado é possível tratar o problema considerando as esperas fixas, deixando ao controlador a

tarefa de determinar os instantes de início dos jobs. Nesse caso, uma vez determinado o início de um job, todas as tarefas deste job estarão automaticamente definidas. Por outro lado é possível tratar cada tarefa individualmente (respeitando obviamente a estrutura de precedência no job) desse modo as esperas entre tarefas de um mesmo job deixam de ser fixas durante o transitório. A principal vantagem de se utilizar a primeira das abordagens acima é a redução do esforço computacional, resultante da simplicidade do modelo. Ao se utilizar a segunda abordagem ganha-se em precisão e qualidade da solução devido ao maior espaço de alternativas. Há portanto um compromisso entre esforço computacional (favorável na primeira abordagem) e qualidade da solução (favorável na segunda abordagem). A abordagem que considera as esperas entre tarefas de um mesmo job fixas, é mais geral do ponto de vista da formulação do problema. Por esse motivo, neste capítulo este será o caso considerado.

A alternativa de se tratar cada tarefa individualmente é, na prática, um caso particular do cenário anterior e será tratado em exemplos específicos no capítulo 6.

Como já discutido, um job shop cíclico está em regime transitório, quando está afastado de sua trajetória regular (regime permanente). Com isto são necessários controladores que no menor tempo possível levem o sistema de volta à condição anterior, e assim consigam evitar prejuízos e mesmo a parada ou perda da produção.

Objetivando-se atender tais necessidades, e baseando-se em uma lei de controle similar à proposta por Noronha(1999), é apresentada no decorrer deste capítulo uma metodologia para a obtenção iterativa dos instantes de início dos ciclos.

Seja o problema de job shop cíclico como apresentado no capítulo 3, sendo $t_j \in \mathbb{Z}^n$ o vetor cujos componentes sejam os instantes de início do j-ésimo ciclo do job shop. Conforme técnicas desenvolvidas por Portugal (1999) e descritas no capítulo 3, $t^\#$ representará a solução de referência. Em outras palavras $t^\#$ é o vetor cujas componentes representam os instantes de início dos jobs no primeiro ciclo do sistema. O escalar $\lambda^\#$ representará o período do ciclo de referência. Se o sistema estiver operando em regime, os instantes de início dos jobs do n-ésimo ciclo são portanto dados por:

$$t_n = (\lambda^\#)^n \otimes t^\#$$

onde as operações “produto” e “exponenciação” acima devem ser compreendidas no contexto max-plus.

Ocorrendo uma perturbação no sistema de produção, tais como quebra de máquina, falta de matéria prima, falta de espaço para estoque, etc, uma máquina ou um conjunto de máquinas terão seus funcionamentos suspensos ao passo que as outras máquinas poderão dar prosseguimento à produção. Quando o problema é sanado, o sistema volta à sua capacidade produtiva original, porém os instantes de início das diversas tarefas não estão relacionados entre si da mesma maneira que no regime de funcionamento regular. A questão central em análise neste trabalho é como, a partir deste momento, iniciar as diversas tarefas dos diversos jobs de modo a não interromper o funcionamento da planta e além disso garantir o retorno ao regime regular de produção. Para isso, vai-se considerar que, após o saneamento da falha ocorrida, os instantes de início dos jobs correspondem às componentes de um vetor arbitrário $t_0 \in \mathbb{Z}^n$. Este vetor deve ser considerado como um “resumo” da situação da planta após o saneamento da falha, isto é, tudo se passa como se os jobs tivessem se iniciado naqueles instantes. Deve-se observar que os instantes de início do último ciclo observado podem corresponder virtualmente aos componentes do vetor t_0 . Este fato é consequência da possibilidade de se considerar um job como uma unidade indivisível. A figura 5.1 ilustra este efeito.

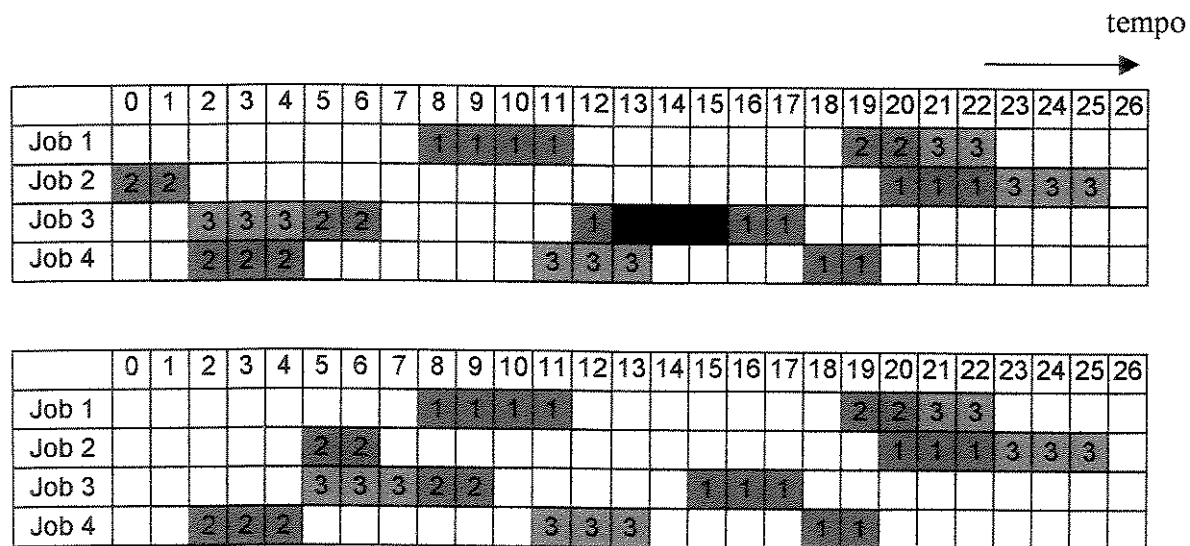


Figura 5.1 – Representação do novo instante de início do jobs.

No exemplo mostrado na figura 5.1, os números que aparecem dentro dos quadros coloridos são as máquinas que estão sendo utilizadas, observe que no instante de tempo igual a

13 ocorreu a quebra da máquina 1, quando ela realizava a tarefa 3 do job 3, e que ela voltou a estar disponível para uso no instante de tempo igual a 16, com isto houve um atraso de 3 unidades de tempo no término da tarefa 3 do job3. Esta quebra teve influência no job 2, pois o término da tarefa 3 ocorreu 5 unidades de tempo atrasada em relação ao término desta tarefa quando o sistema está em regime. Os instantes de início t_0 são obtidos do diagrama que está embaixo, neste diagrama os tempos de atraso que ocorrerão no diagrama anterior são incorporados no instante de início da primeira tarefa dos respectivos jobs.

Em síntese, para todos os efeitos, o problema de tratar o transitório do sistema de job shop cíclico consiste em definir os instantes de início dos próximos ciclos a partir de uma condição inicial t_0 de modo que se retorne ao funcionamento de regime definido pelo vetor $t^\#$.

A lei de controle proposta no trabalho de Noronha(1999) supõe a existência de um conjunto de matrizes $A_l, l \in \{1, \dots, n\}$ que serão utilizadas para a determinação dos instantes de início dos jobs nos ciclos subsequentes, com o objetivo de restabelecer o ciclo anteriormente determinado.

Com isto, o controle do sistema é feito da seguinte forma:

$$t_i^T = t_{i-1}^T \otimes A_l \quad (5.1)$$

onde:

t_i é o vetor de instante de início dos jobs no próximo ciclo;

t_{i-1} é o vetor de instante de início dos jobs no ciclo atual;

A_l é uma matriz de decisão, escolhida no conjunto citado acima.

O conjunto de matrizes $A_l, l = \{1, \dots, m\}$ deve satisfazer a duas condições. Em primeiro lugar o conjunto das matrizes normalizadas $B_l = -\lambda_l \otimes A_l, l = \{1, \dots, m\}$ deve ser $t^\#$ -consistente. Em segundo lugar deve-se garantir para qualquer vetor t_{i-1} a existência de pelo menos uma matriz A_l tal que t_i seja factível.

As matrizes referidas na lei de controle proposta anteriormente, podem ser obtidas conforme procedimentos apresentados na seção 4.3.4 do capítulo 4.

A lei de controle proposta neste trabalho, deve satisfazer as seguintes restrições:

$$\beta \otimes t^\# \leq t_i < \alpha \otimes t^\# \quad (5.2)$$

e

$$t_i^T \notin \text{supp}\{(t_i^T \otimes M^*) \oplus (t_{i-1}^T \otimes M)\} \quad (5.3)$$

onde:

$t^\# \in \mathbb{Z}^n$ é a solução de referência;

$t_i \in \mathbb{Z}^n$ é o vetor de instante de início dos jobs no ciclo i ;

t_i^* é um vetor $\in \mathcal{P}^n$ cujo significado será especificado na seção 5.2;

$t_0 \in \mathbb{Z}^n$ é o vetor de instantes de inícios dos jobs da primeira solução após a perturbação;

β e α são escalares que dependem de t_{i-1} (a serem definidos na seção 5.3);

M é uma matriz quadrada no domínio $\mathcal{P}_\Delta^{n \times n}$, que será definida na seção 5.2;

M^* é uma matriz quadrada no domínio $\mathcal{P}_\Delta^{n \times n}$, que será definida na seção 5.2.

Satisfazer estas restrições, em geral é possível escolher os instantes de início dos jobs do ciclo subsequente, dentro de um conjunto de soluções. São portanto necessárias heurísticas para a tomada final de decisão. Estas heurísticas serão apresentadas no capítulo 6.

Vai-se mostrar que esta solução é equivalente à soluções do tipo proposto pela equação (5.1), usufruindo portanto das mesmas propriedades, e apresentando vantagens do ponto de vista do projeto da lei de controle. A principal vantagem é que o controlador é obtido *on-line* baseado em informações disponíveis. A equação (5.3), verifica se t_i é factível em relação ao próprio ciclo e em relação ao ciclo anterior.

Se a equação (5.3) for verificada, após alguns ciclos o job shop retorna ao regime permanente, isto é para a solução de referência deslocada no tempo.

O que vai se mostrar neste capítulo é que as equações (5.2) e (5.3) se comportam da seguinte forma. Não havendo perturbação no sistema de produção, ou seja, não havendo perturbação na solução de referência cíclica, a lei de controle deve manter o sistema em regime permanente. Ocorrendo uma perturbação, a lei de controle deve reconduzir o sistema à sua solução de referência, eventualmente deslocada no tempo.

Na seção 5.2 apresenta-se um modelo para verificação de factibilidade dos instantes de início de cada job nos ciclos subsequentes à perturbação detectada. Tal modelo é similar ao modelo proposto por Groth e Santos-Mendes (1997) e se baseia no monóide $\mathcal{P}_\Delta$ apresentado

na seção 4.4. Na seção 5.3, utilizando-se dos resultados apresentados na seção 4.3, mostra-se que se os próximos instantes de disparo pertencem à região determinada pela equação 5.2, então a solução converge para a solução de referência.

5.2 – Modelo para a Análise da Factibilidade das Decisões

Conforme discutido na introdução deste capítulo, a lei de controle proposta neste trabalho é tal que os instantes de início de cada job nos ciclos subsequentes sejam dados pela expressão:

$$\beta \otimes t^{\#} \leq t_i \leq \alpha \otimes t^{\#}$$

desde que t_i seja factível.

Nesta seção serão discutidos os modelos matemáticos para a análise de factibilidade de uma solução dada. Em outras palavras considera-se o seguinte problema: dada uma sequência de instantes de início dos jobs nos ciclos precedentes e dado um vetor, candidato a ser o próximo vetor de instante de início, pergunta-se se os instantes propostos provocarão ou não algum conflito (uso de uma mesma máquina no mesmo período) no sistema. Dir-se-á que uma solução (um vetor de instante de início) é factível se a sua utilização não provocar nenhum conflito. Levando em consideração o problema de job shop apresentado no capítulo 3, serão caracterizadas as situações que levam a conflitos entre tarefas.

Iniciar-se-á o estudo considerando-se quaisquer dois jobs $i, j \in \{1, \dots, n\}$, e a situação em que apenas o job i está em curso tendo sido disparado em t_0 , desejando-se disparar o job j em algum instante posterior. Observando e analisando os diagramas temporais de cada job, é possível determinar em que instantes o job j pode ser disparado sem que nenhum conflito ocorra entre operações de dois jobs. Seja s_{ij} uma função booleana, tal que:

$$s_{ij}(k) = \begin{cases} 0 & \text{se não houver conflito quando o job } j \text{ for disparado em } t_1 = t_0 + k \\ 1 & \text{caso contrário} \end{cases} \quad (5.4)$$

De uma maneira geral, considerando-se que a duração de qualquer job é finita, é possível afirmar que:

$$s_{ij}(k) = 0 \quad \forall k > K^* \quad (5.5)$$

Além disso, supõe-se que não é possível disparar o job j antes de disparar o job i , com isto tem-se:

$$s_{ij}(k) = 1 \quad \forall k < K^0 \quad (5.6)$$

Um gráfico típico da função $s_{ij}(k)$ é mostrado a seguir

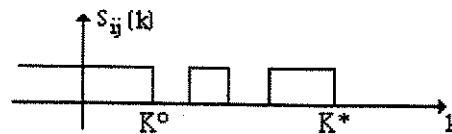


Figura 5.2 – Gráfico típico de um elemento da função s_{ij}

Observa-se no gráfico da figura 5.2, que $K^0 < s_{ij}(k) = 0 < K^*$ ocorre devido o fato do job ser considerado uma “unidade indivisível”, ou seja, as tarefas mantêm fixas as esperas entre elas determinadas na solução de referência.

Para cada par $\langle i, j \rangle$ a função s_{ij} dá os instantes onde o job j pode ser disparado, supondo que o job i tenha sido disparado em $t_0 = 0$.

Do exposto até o momento, é possível portanto associar ao par de jobs i e j , um elemento do monóide $\mathcal{S}_\Delta$ anteriormente definido da seguinte forma:

$$s_{ij} = \bigoplus_{k \in \mathbb{Z}} s_{ij}(k) \tau^k \quad (5.7)$$

Define-se para um job shop a matriz $M \in \mathbb{Z}^{n \times n}$, cujos elementos pertençam ao monóide $\mathcal{S}_\Delta$ como a seguir:

$$[M]_{ij} = \bigoplus_{k \in \mathbb{Z}} s_{ij}(k) \tau^k \quad (5.8)$$

Esta matriz tem seu tamanho definido pelo número de jobs que compõem o job shop, e cada elemento dá os instantes que um job pode ser disparado em relação ao instante de disparo de outro. Observe que é possível, saber quando se pode disparar o job i em relação a ele próprio.

Como se verá a seguir a matriz M constitui um modelo para a análise de factibilidade de decisões.

Em geral, 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 que deseja-se conhecer em que intervalos de tempo o job j pode ser disparado, sem qualquer conflito como os jobs já disparados. Observe que:

- i) os instantes de tempo nos quais o job j pode ser disparado sem conflito com o job i_m são os valores dos expoentes de τ no polinômio $\tau^k \otimes [M]_{mj}$ cujos coeficientes assumem valor zero (0);
- ii) os instantes de tempo nos quais o job j pode ser disparado sem conflitos com o job i_1 , com o job i_2 e os restantes, são os valores dos expoentes de τ no polinômio $g_j(\tau)$ cujos coeficientes assumem valor zero (0), sendo:

$$g_j(\tau) = ((t'_1 \otimes [M]_{i_1,j}) \oplus (t'_2 \otimes [M]_{i_2,j}) \oplus \dots \oplus (t'_p \otimes [M]_{i_p,j})) \quad (5.9)$$

onde:

as operações $\oplus$ e $\otimes$ são aquelas definidas para o dióide $B[[\tau]]$;

$$t'_i = \tau^{t_i}.$$

Generalizando esta última equação para n jobs, e escrevendo-a para um job shop qualquer, em notação matricial vem:

$$g^T(\tau) = t'^T \otimes M \quad (5.10)$$

onde:

$$g^T(\tau) = [g_1(\tau) \ g_2(\tau) \ \dots \ g_n(\tau)], \quad g_i(\tau) \in \mathcal{S}_\Delta$$

$$t'^T = [t'_1 \ t'_2 \ \dots \ t'_n], \quad t'_i = \tau^{t_i}$$

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix}, \quad m_{ij} \in \mathcal{S}_{\Delta}^{n \times n}$$

Conclui-se portanto que se p jobs foram disparados (em ciclos anteriores ou no atual), então o job j poderá ser disparado sem conflito no instante t'_j se e somente se $t'_j \notin \text{supp}(g_j(\tau))$.

Observa-se que na metodologia proposta, a análise da factibilidade é feita job a job, ou seja, conhecidos os jobs já disparados, $g_j(\tau)$ fornece os instantes de início possíveis ao job j .

Considere-se agora o problema de analisar a factibilidade de todo um ciclo, isto é os disparos de todos n próximos jobs que compõem o sistema. Neste caso deve-se separar o problema em duas partes. Por um lado deve-se analisar as interferências de jobs disparados em ciclos precedentes. Por outro lado também deve-se garantir a não existência de conflitos entre os jobs disparados no mesmo ciclo. Portanto, nenhum job do ciclo em teste deve estar em conflito com nenhum job disparado em ciclos precedentes, assim como não estar em conflito com nenhum job do mesmo ciclo disparado anteriormente a ele.

Para a análise dos jobs disparados em ciclos precedentes o procedimento não é diferente do discutido anteriormente. Para verificar se há conflito entre os jobs a serem disparados e os jobs já disparados no mesmo ciclo, uma outra matriz pertencente a $\mathcal{S}_{\Delta}^{n \times n}$ será introduzida para tal objetivo.

Define-se a nova matriz denominada M^* , da seguinte forma:

$$m^*_{ij} = \begin{cases} m_{ij} & \text{se } [t]_i < [t]_j \\ \varepsilon & \text{caso contrário} \end{cases}$$

com isto a expressão a ser verificada torna-se:

$$t_i^T \notin \text{supp}\{(t_i^T \otimes M^*) \oplus (t_{i-1}^T \otimes M)\} \quad (5.11)$$

A seguir um exemplo é apresentado para ilustrar a verificação de factibilidade.

Exemplo 5.1.

A tabela 5.1, fornece os dados relativos a um job shop cíclico, com quatro jobs, três máquinas e cada job tendo três operações.

Número do Job	Número de Tarefas	Roteiro de Fabricação	Instante de Início das Tarefas (regime)		
			Duração das Tarefas nas Máquinas		
1	3	1 2 3	8	19	21
			4	2	2
2	3	2 1 3	0	15	18
			2	3	3
3	3	3 2 1	2	5	12
			3	2	3
4	3	2 3 1	2	11	18
			3	3	2

Tabela 5.1 – Roteiro de fabricação, duração e instante de início das tarefas.

A solução em regime para o job shop cíclico, segundo técnicas de Portugal (1999), corresponde a:

$$t^{\#} = [8 \ 0 \ 2 \ 2]^T \quad \text{e} \quad \lambda^{\#} = 12$$

Considere $t_0 = [0 \ 1 \ 0 \ 7]^T$ como sendo os instantes de início dos jobs em um determinado ciclo, após as seguintes perturbações: quebra da máquina 2 quando está realizando a tarefa 1 do job 4 em $t = 4$ voltando a estar disponível em $t = 5$; quebra da máquina 3 quando está realizando a tarefa 2 do job 4 em $t = 12$ voltando a estar disponível em $t = 19$; quebra da máquina 1 quando está realizando a tarefa 3 do job 3 em $t = 14$ voltando a estar disponível em $t = 19$; quebra da máquina 3 quando está realizando a tarefa 3 do job 2 em $t = 24$ voltando a estar disponível em $t = 29$; quebra da máquina 1 quando está realizando a tarefa 3 do job 4 em $t = 26$ voltando a estar disponível em $t = 30$.

De posse do vetor t_0 pode-se calcular os instantes de início dos jobs do próximo ciclo pela equação (5.2).

$$\beta \otimes t^{\#} \leq t_i \leq \alpha \otimes t^{\#}$$

da qual obtêm-se o seguinte vetor de instantes de início:

$$t_1 = [25 \ 17 \ 10 \ 19]^T$$

Deve-se agora verificar se t_1 é factível, para isto ele deve satisfazer a equação (5.11).

A matriz M para este exemplo é apresentada na figura 5.3 através de diagramas que representam os coeficientes de τ^{ij} em função de t_i .

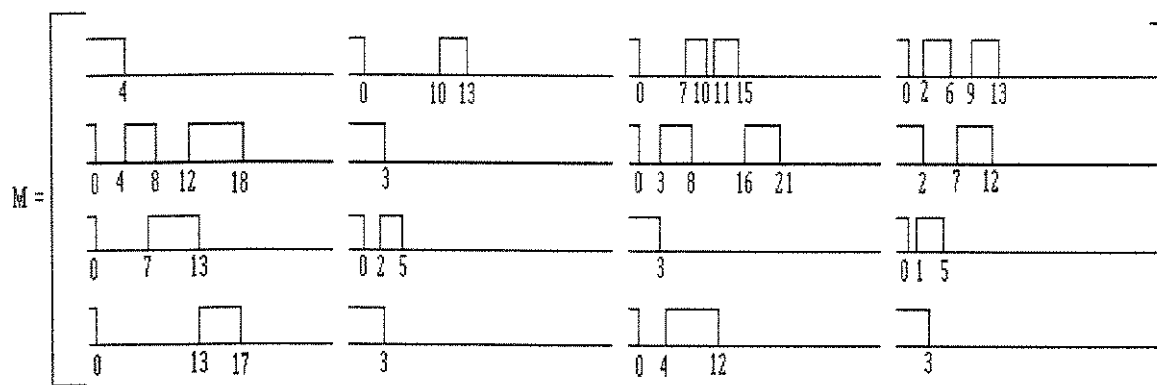


Figura 5.3 – Matriz M para o exemplo 5.1

Utilizando-se a mesma representação, a matriz M^* é apresentada na figura 5.4.

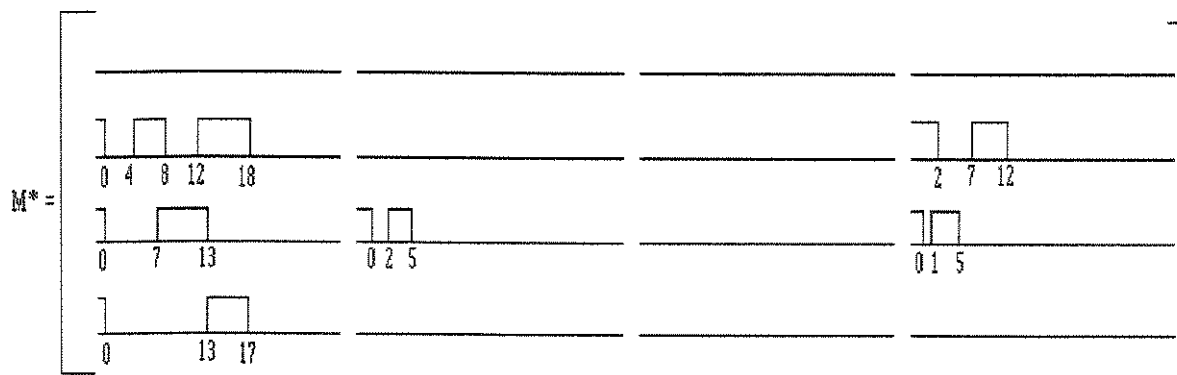
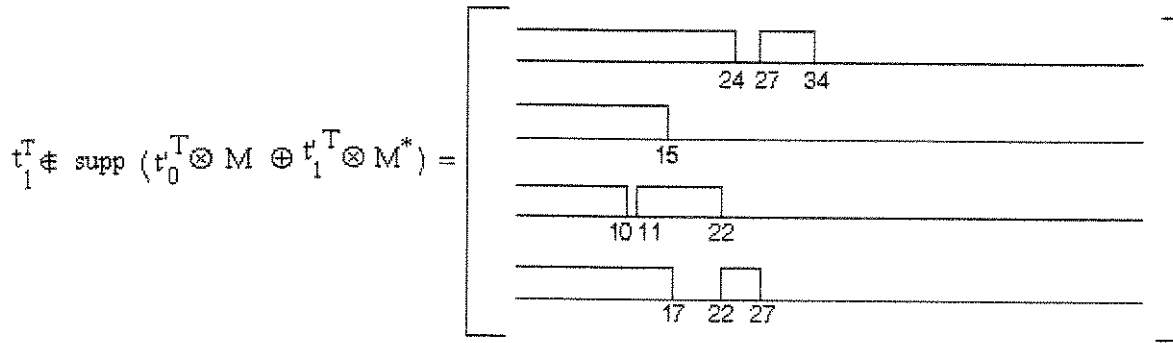


Figura 5.4 – Matriz M^* para o exemplo 5.1.

Utilizando-se a equação (5.11), obtêm-se o vetor $g(\tau)$ representado na figura 5.5.


 Figura 5.5 - Matriz $(t_1^T \otimes M^*) \oplus (t_0^T \otimes M)$ para o exemplo 5.1.

como os valores das componentes de t_1 não pertencem ao suporte de $(t_1^T \otimes M^* \oplus t_0^T \otimes M)$ conclui-se que ele (t_1) é factível.

Como observação, os elementos das matrizes representadas nas figuras 5.3, 5.4 e 5.5, são elementos pertencentes ao monóide $\mathcal{P}_A^{nm}$, cujos eixos possuem valores pertencentes ao conjunto de números inteiros.

O diagrama da figura 5.6, ilustra graficamente o exposto acima.

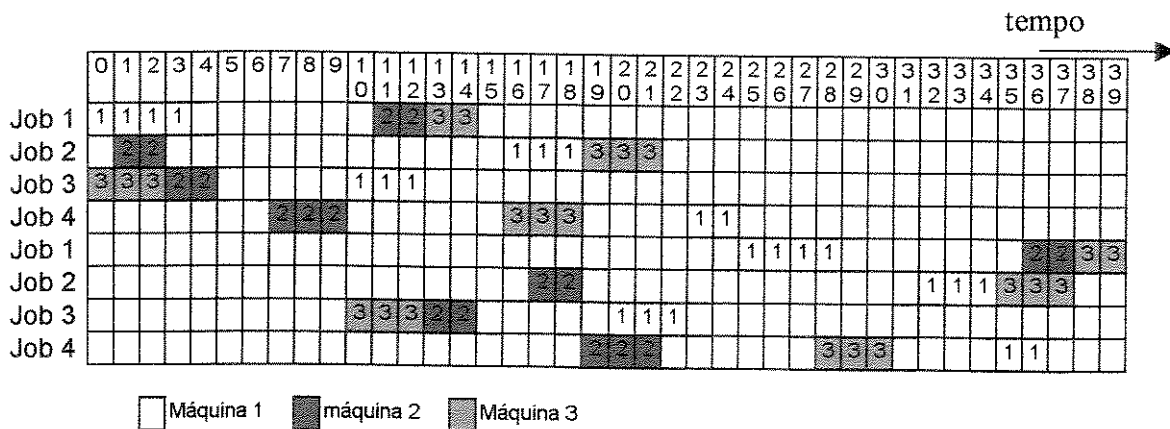


Figura 5.6 – Diagrama esquemático para visualização da factibilidade.

5.3 – Síntese da Lei de Controle

Nesta seção será proposta uma metodologia de determinação da lei de controle para o transitório num sistema de job shop cíclico. A metodologia proposta neste trabalho fundamenta-se em alguns resultados originalmente propostos em (Noronha,1999) cujo resumo será apresentado a seguir.

Considere-se um conjunto de matrizes $\{A_1, \dots, A_n\}$, tal que seus autovalores sejam únicos e dados por $\{\lambda_1, \dots, \lambda_n\}$.

Seja $\{B_1, \dots, B_n\}$ a normalização do conjunto anterior. A partir das definições e teoremas 4.55 a 4.60 pode-se inferir que, se o conjunto $\{B_1, \dots, B_n\}$ for sub-crítico em relação a $t^\#$ então a seguinte lei de controle pode ser utilizada com garantia de convergência para a solução de regime ($\delta \otimes t^\#$):

$$t_i^T = t_{i-1}^T \otimes A_i$$

onde:

t_i é o vetor de instante de início dos jobs no próximo ciclo;

t_{i-1} é o vetor de instante de início dos jobs no ciclo atual;

A_i é qualquer matriz do conjunto $\{A_1, \dots, A_n\}$ desde que t_i seja factível.

Para que esta estratégia de controle funcione efetivamente é necessário garantir que sempre exista uma matriz no conjunto $\{A_1, \dots, A_n\}$ de modo que o vetor t_i resultante seja factível.

O exemplo a seguir ilustra o método apresentado .

Exemplo 5.2.

Utilizando-se o exemplo 5.1, será apresentada a metodologia desenvolvida em Noronha (1999), para a obtenção das soluções até que elas converjam.

Tem-se através das técnicas desenvolvidas por Portugal (1999) e Noronha(1999), os seguintes dados:

$$t^\# = [8 \quad 0 \quad 2 \quad 2]^T$$

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

$$A^{\#} = \begin{bmatrix} 12 & 4 & 6 & 6 \\ 20 & 12 & 14 & 14 \\ 18 & 10 & 12 & 12 \\ 18 & 10 & 12 & 12 \end{bmatrix}$$

$$\lambda^{\#} = 12$$

onde:

$t^{\#}$ é a solução de referência

$A^{\#}$ é a matriz de decisão de referência, ou seja, é a matriz que será sempre utilizada caso não haja nenhuma perturbação

$\lambda^{\#}$ é o autovalor de $A^{\#}$

Considere uma perturbação na solução de referência, e seja t_0 a primeira solução determinada imediatamente após tal perturbação, isto é:

$$t_0 = [0 \ 1 \ 0 \ 7]^T$$

A partir dos dados do problema é fácil verificar que as matrizes dadas a seguir têm suas normalizações pertencentes ao conjunto sub-crítico em relação a $t^{\#}$.

$$A_1 = \begin{bmatrix} 13 & 5 & 7 & 7 \\ 19 & 10 & 9 & 14 \\ 17 & 9 & 10 & 11 \\ 18 & 10 & 2 & 12 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 11 & 3 & 5 & 5 \\ 19 & 11 & 13 & 13 \\ 17 & 9 & 11 & 11 \\ 18 & 10 & 12 & 12 \end{bmatrix}$$

Utilizando-se uma matriz do conjunto de matrizes de decisão, tem-se:

$$t_1 = [25 \ 17 \ 10 \ 19] = t_0^T \otimes A_1$$

Utilizando-se novamente uma matriz do conjunto de matrizes de decisão, tem-se:

$$t_2 = [37 \ 29 \ 31 \ 31] = t_1^T \otimes A_2$$

A partir deste instante deve-se utilizar a matriz $A^{\#} = \lambda^{\#} \otimes B^{\#}$, que manterá o sistema em regime no ciclos subsequentes, pois neste instante ocorreu a convergência para a solução de referência.

Uma dificuldade inerente a este método é a síntese das matrizes $\{A_1, \dots, A_n\}$ de modo a satisfazer a todas as restrições. Sempre é possível obter uma matriz sub-crítica em relação a $t^\#$ que resulte numa solução factível. Contudo esta solução pode ser excessivamente conservativa no sentido de provocar atrasos importantes no sistema. A tentativa de calcular matrizes cujas normalizações sejam sub-críticas em relação a $t^\#$ de modo que, para condições iniciais arbitrárias, tenha-se uma solução não-conservativa (isto é, que resulte em atrasos importantes) é computacionalmente custosa. Os resultados apresentados a seguir representam um esforço para superar estas dificuldades.

Definição 5.1 – Seja A uma matriz $m \times n$, e seja B uma matriz $n \times p$.

Define-se a seguinte operação:

$$[A \otimes' B]_{ij} = \min_k ([A]_{ik} \otimes [B]_{kj})$$

A operação apresentada na definição 5.1 é análoga à definição da multiplicação de matrizes (Eq. (4.21))

$$[A \otimes B]_{ij} = \max_k ([A]_{ik} \otimes [B]_{kj})$$

Essas operações apresentam algumas características comuns. Uma delas é a associatividade, haja vista que:

$$[(A \otimes' B) \otimes' C]_{ij} = [A \otimes' (B \otimes' C)]_{ij}$$

pois,

$$[(A \otimes' B) \otimes' C]_{ij} = \min_l [\min_k ([A]_{ik} \otimes [B]_{kl}) \otimes [C]_{lj}] = \min_l [\min_k ([A]_{ik} \otimes [B]_{kl} \otimes [C]_{lj})]$$

e

$$[A \otimes' (B \otimes' C)]_{ij} = \min_k [[A]_{ik} \otimes \min_l ([B]_{kl} \otimes [C]_{lj})] = \min_k [\min_l ([A]_{ik} \otimes [B]_{kl} \otimes [C]_{lj})]$$

o que prova a igualdade anterior.

Exemplo 5.3

Considere-se as seguintes matrizes:

$$A = \begin{bmatrix} 12 & 6 & -6 & 7 \\ 19 & 5 & 19 & 1 \\ 17 & 3 & 1 & 4 \\ 17 & 4 & -3 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -8 & -6 & -6 \\ 8 & 0 & 2 & 2 \\ 6 & -2 & 0 & 0 \\ 6 & -2 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 20 & 5 & 16 & 9 \\ -8 & -10 & 13 & -4 \\ 3 & 1 & 30 & -6 \\ 7 & 2 & 5 & 13 \end{bmatrix}$$

então tem-se:

$$A \otimes' B = \begin{bmatrix} 0 & -8 & -6 & -6 \\ 7 & -1 & 1 & 1 \\ 7 & -5 & 1 & 1 \\ 3 & -5 & -3 & -3 \end{bmatrix} \quad e \quad (A \otimes' B) \otimes' C = \begin{bmatrix} -16 & -18 & -1 & -12 \\ -9 & -11 & 6 & -5 \\ -13 & -15 & 6 & -9 \\ -13 & -15 & 2 & -9 \end{bmatrix}$$

$$B \otimes' C = \begin{bmatrix} -16 & -18 & -1 & -12 \\ -8 & -10 & 7 & -4 \\ -10 & -12 & 5 & -6 \\ -10 & -12 & 5 & -6 \end{bmatrix} \quad e \quad A \otimes' (B \otimes' C) = \begin{bmatrix} -16 & -18 & -1 & -12 \\ -9 & -11 & 6 & -5 \\ -13 & -15 & 6 & -9 \\ -13 & -15 & 2 & -9 \end{bmatrix}$$

Teorema 5.3 – Sejam $t_{i-1}, t^\# \in \mathbb{Z}^n$ vetores quaisquer.

Sejam:

$$\alpha = t_{i-1}^T \otimes (-t^\#)$$

$$\beta = t_{i-1}^T \otimes' (-t^\#)$$

B = conjunto de matrizes sub-críticas em relação a $t^\#$ definido em 4.59.

Tem-se:

$$\beta \otimes t^{\#T} \leq t_i^T < \alpha \otimes t^{\#T} \Leftrightarrow \exists B \in B / t_i^T = t_{i-1}^T \otimes B$$

Demonstração:

$$i) \beta \otimes t^\# \leq t_i < \alpha \otimes t^\# \Rightarrow \exists B \in B / t_i^T = t_{i-1}^T \otimes B$$

Tem-se:

$$\beta \otimes t^{\#T} \leq t_i^T < \alpha \otimes t^{\#T}$$

substituindo β e α na inequação anterior por suas definições vem:

$$t_{i-1}^T \otimes' (-t^\#) \otimes t^{\#T} \leq t_i^T < t_{i-1}^T \otimes (-t^\#) \otimes t^{\#T}$$

Sejam j^* e k^* tais que:

$$t_{i-1}^T \otimes' (-t^\#) = \min_j ([t_{i-1}]_j \otimes [-t^\#]_j) = [t_{i-1}]_{j^*} \otimes [-t^\#]_{j^*}$$

$$t_{i-1}^T \otimes (-t^\#) = \max_k ([t_{i-1}]_k \otimes [-t^\#]_k) = [t_{i-1}]_{k^*} \otimes [-t^\#]_{k^*}$$

Seja B tal como a seguir:

$$[B]_{nm} = \begin{cases} \varepsilon & \text{se } n \neq k^* \text{ e } n \neq j^* \\ [-t_{i-1}]_{k^*} \otimes [t_i]_m & \text{se } n = k^* \\ [-t^\#]_{j^*} \otimes [t^\#]_m & \text{se } n = j^* \end{cases}$$

sendo que:

$$[-t_{i-1}]_{k^*} \otimes [t_i]_m < [-t^\#]_{k^*} \otimes [t^\#]_m$$

A multiplicação de t_{i-1}^T por B resulta em:

$$[t_{i-1}^T \otimes B]_m = \max\{([t_{i-1}]_{k^*} \otimes [-t_{i-1}]_{k^*} \otimes [t_i]_m), ([t_{i-1}]_{j^*} \otimes [-t^\#]_{j^*} \otimes [t^\#]_m)\} = [t_i]_m$$

observe que:

$$[t_{i-1}]_{k^*} \otimes [-t_{i-1}]_{k^*} \otimes [t_i]_m < [t_{i-1}]_{k^*} \otimes [-t^\#]_{k^*} \otimes [t^\#]_m$$

$$[t_{i-1}]_{k^*} \otimes [-t_{i-1}]_{k^*} \otimes [t_i]_m < \alpha \otimes [t^\#]_m$$

e

$$[t_{i-1}]_{j^*} \otimes [-t^\#]_{j^*} \otimes [t^\#]_m = \beta \otimes [t^\#]_m$$

Portanto,

$$t_i^T = t_{i-1}^T \otimes B$$

Além disso, por construção:

$$\forall m; \quad [B^\#]_{nm} > [B]_{nm} \quad \text{para } n \neq j^* \text{ e } n \neq k^*$$

$$\forall m; \quad [B^\#]_{nm} > [B]_{nm} \quad \text{para } n = k^*$$

$$\text{pois } [-t_{i-1}]_{k^*} \otimes [t_i]_m < [-t^\#]_{k^*} \otimes [t^\#]_m$$

$$\forall m; \quad [B^\#]_{nm} = [B]_{nm} \quad \text{para } n = j^*$$

Estas três condições garantem que $B \in \mathcal{B}$.

$$\text{ii) } t_i^T = t_{i-1}^T \otimes B \Rightarrow \beta \otimes t^\# \leq t_i < \alpha \otimes t^\#$$

Dado uma matriz $B \in \mathcal{B}$ qualquer, então as componentes do vetor t_i satisfazem a:

$$[t_i]_m = \max_k ([t_{i-1}]_k \otimes [B]_{km})$$

Suponha que o máximo ocorrerá para $k = p$, com isto tem-se;

$$[t_i]_m = [t_{i-1}]_p \otimes [B]_{pm}$$

Seja $[t_i]_m$ escrito como a seguir:

$$[t_{i-1}]_{j^*} \otimes [-t^{\#}]_{j^*} \otimes [t^{\#}]_m \leq [t_i]_m < [t_{i-1}]_{k^*} \otimes [-t^{\#}]_{k^*} \otimes [t^{\#}]_m$$

ou

$$[t_{i-1}]_{j^*} \otimes [-t^{\#}]_{j^*} \otimes [t^{\#}]_m \leq [t_{i-1}]_p \otimes [B]_{pm} < [t_{i-1}]_{k^*} \otimes [-t^{\#}]_{k^*} \otimes [t^{\#}]_m$$

$$[t_{i-1}]_{j^*} \otimes [B]_{j^*m} \leq [t_{i-1}]_p \otimes [B]_{pm} < [t_{i-1}]_{k^*} \otimes [B]_{k^*m}$$

mas, é obvio que:

$$[t_{i-1}]_{j^*} \otimes [B]_{j^*m} = \beta \otimes [t^{\#}]_m$$

$$[t_{i-1}]_{k^*} \otimes [B]_{k^*m} = \alpha \otimes [t^{\#}]_m$$

Portanto:

$$\beta \otimes [t^{\#}]_m \leq [t_{i-1}]_p \otimes [B]_{pm} < \alpha \otimes [t^{\#}]_m$$

consequentemente:

$$\beta \otimes [t^{\#}]_m \leq [t_i]_m < \alpha \otimes [t^{\#}]_m$$

Corolário 5.4 – Dada a lei de controle:

$$\beta \otimes t^{\#} \leq t_i < \alpha \otimes t^{\#}$$

então $\lim_{i \rightarrow \infty} t_i = \delta \otimes t^{\#}$, onde α e β são definidos no teorema 5.3.

Demonstração: Do teorema 5.3, tem-se que:

$$\beta \otimes t^{\#T} \leq t_i^T < \alpha \otimes t^{\#T} \Leftrightarrow \exists B \in B / t_i^T = t_{i-1}^T \otimes B$$

como B é sub-crítico em relação a $t^{\#}$, então ele é também $t^{\#}$ -consistente. Com isto o teorema 4.56 garante a convergência da solução para a solução de referência.

Na resolução de um problema de job shop cíclico de dimensão n cuja solução de referência é conhecida e é dada pelo vetor $t^{\#} \in \mathbb{Z}^n$. O teorema 5.3 estabelece um procedimento para o cálculo dos instantes de início de um ciclo i a partir dos instantes de início do ciclo

precedente (i-1). Deve-se observar que este procedimento garante a convergência para a solução de regime, mas não trata a questão da factibilidade abordada na seção 5.2.

É importante observar que a multiplicação do vetor t_i por um escalar λ , não altera os resultados relativos à convergência do resultado para a solução de regime. Assim sendo é lícito, a cada passo, escolher arbitrariamente um escalar para deslocar o vetor t_i . Este procedimento é utilizado para produzir resultados factíveis segundo a análise apresentada na seção 5.2. A proposição a seguir estabelece um limite para este escalar.

Proposição 5.5 – Para um dado vetor de instantes de início t_i , o menor autovalor que pode ser utilizado deve satisfazer à seguinte condição:

$$\min_k (\beta \otimes [t^\#]_k) = \text{c-valorção de } ([t'_{i-1}]_k \otimes [M]_{kj})$$

Demonstração: Sejam $[t_i]_m = \beta \otimes [t^\#]_m$ os instantes de início do próximo ciclo, neste momento deve ser observado se as componentes $[t_i]_m$ são menores ou maiores do que as c-valorções de $(t'_{i-1}{}^T \otimes M)$ correspondentes. Ressalta-se que neste momento $\lambda = 0$. Então pode-se observar dois casos:

Caso 1: $t_i = \beta \otimes t^\# >$ que as c-valorções de $(t'_{i-1}{}^T \otimes M)$.

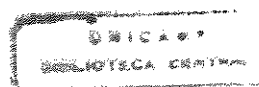
Neste caso λ deve ser feito menor do zero(0), até que $\min_k (\beta \otimes [t^\#]_k)$ se iguale a c-valorção de $([t'_{i-1}]_k \otimes [M]_{kj})$ correspondente.

Caso 2: $t_i = \beta \otimes t^\# <$ que as c-valorções de $(t'_{i-1}{}^T \otimes M)$.

Neste caso λ deve ser feito maior do zero(0), até que $\min_k (\beta \otimes [t^\#]_k)$ se iguale a c-valorção de $([t'_{i-1}]_k \otimes [M]_{kj})$ correspondente.

Exemplo 5.4

Utilizando-se o exemplo 5.1, será apresentada a metodologia desenvolvida neste trabalho, para a obtenção das soluções até que elas converjam.



Tem-se através das técnicas desenvolvidas por Portugal (1999), os seguintes dados:

$$t^{\#} = [8 \quad 0 \quad 2 \quad 2]^T$$
$$\lambda^{\#} = 12$$

Considere uma perturbação na solução de referência, e seja t_0 a primeira solução determinada imediatamente após tal perturbação, isto é:

$$t_0 = [0 \quad 1 \quad 0 \quad 7]^T$$

Utilizando a lei de controle, obtém-se:

$$\beta = -8 \text{ e } \alpha = 5$$

e consequentemente,

$$t_1 = [25 \quad 17 \quad 10 \quad 19]$$

Utilizando-se novamente a lei de controle, obtém-se:

$$\beta = 8 \text{ e } \alpha = 17$$

e consequentemente,

$$t_2 = [37 \quad 29 \quad 31 \quad 31]$$

Utilizando-se novamente a lei de controle, obtém-se:

$$\beta = 29 \text{ e } \alpha = 29$$

e consequentemente,

$$t_3 = [49 \quad 41 \quad 43 \quad 43]$$

Resultados iguais ao obtidos pelo método de Noronha (1999).

Portanto, observa-se que as leis de controle deste trabalho e do trabalho de Noronha (1999) são semelhantes em relação a convergência, mas elas diferem quanto ao modo de obterem a solução, sendo o deste trabalho de menor esforço computacional.

No capítulo seguinte são analisadas algumas heurísticas para determinação dos instantes de início dos jobs de modo a satisfazer as restrições impostas neste capítulo. Alguns exemplos são analisados visando ilustrar a eficácia da metodologia.

CAPÍTULO 6

ALGORITMOS E RESULTADOS

6.1 – Introdução

Nos capítulos precedentes foram apresentados o problema de job shop cíclico, uma formulação para suas trajetórias de referência, as restrições que devem ser satisfeitas por uma lei de controle de modo a garantir a convergência das soluções para a trajetória de referência. Neste capítulo serão apresentados algoritmos, que implementam tal lei de controle, determinando os instantes de início do próximo ciclo a partir dos instantes de início do ciclo anterior.

Para que um determinado vetor de instantes de tempo t_i seja aceitável como uma solução, ele deve satisfazer à restrição de estabilidade e também satisfazer às restrições de factibilidade que foram apresentadas no capítulo anterior. Portanto partindo de uma condição inicial e com os algoritmos que serão apresentados, chegar-se-á a uma formulação para a obtenção de uma solução.

É importante observar que as inequações 5.2 e 5.3 representam restrições para a lei de controle a ser implementada. Satisfeitas estas restrições, em geral é possível escolher os instantes de início dos jobs do ciclo subsequente, dentro de um conjunto de soluções. São portanto necessárias heurísticas para a tomada final de decisão. Os algoritmos apresentados neste capítulo propõem diferentes maneiras de tratar esta questão.

Os algoritmos propostos serão utilizados para resolver várias instâncias de vários problemas de job shop cíclico, cujas soluções de referência são obtidas empregando as técnicas desenvolvidas por Portugal (1999) que resumidamente foram apresentadas no

capítulo 3. Para a obtenção dos problemas de job shop cíclico Portugal (1999), usou procedimentos aleatórios. Inicialmente foi definido o número de jobs, então foi escolhido aleatoriamente a quantidade de tarefa dos jobs, depois foi associada, aleatoriamente, a cada tarefa i uma máquina m_i , dependendo da especialidade da máquina, gera-se aleatoriamente o tempo de processamento.

Os problemas que serão abordados a seguir, são alguns dos exemplos gerados pela metodologia apresentada no parágrafo anterior, acrescidos de perturbações que podem ocorrer nas linhas de produção. Portanto os algoritmos que serão utilizados para a resolução destes, têm claramente espaço em aplicações industriais.

As soluções obtidas serão apresentadas em gráficos, juntamente com as soluções de referência (regime), e as soluções obtidas simplesmente retardando a solução de regime de modo que o critério de factibilidade fosse satisfeito.

Na seção 6.2 serão apresentados os algoritmos que implementam a lei de controle para os problemas em questão. Na seção 6.3 serão apresentados os problemas de job shop cíclico, assim como os resultados obtidos utilizando-se os algoritmos da seção precedente. Conforme observado na introdução do capítulo 5, na seção 6.4, será abordado o caso em que cada tarefa é tratada como um job. Mostra-se que aumentando a dimensão do problema e mantendo as relações de precedência no job, é possível formular o problema de modo que cada tarefa seja tratada independentemente. Com isto embora o custo computacional do algoritmo aumente elimina-se as esperas fixas introduzidas na obtenção da solução de regime e melhora-se o desempenho global do algoritmo. Na seção 6.5 será feita uma análise dos resultados obtidos com os algoritmos propostos para os vários exemplos das seções anteriores.

6.2 – Algoritmos de Síntese

Nesta seção apresentar-se-ão detalhadamente os algoritmos, para o cálculo das soluções para os problemas de job shop cíclico, em regime transitório.

Para os algoritmos a serem apresentados, são admitidas como entradas os seguintes dados:

- a) o número de jobs, e o número de tarefas de cada job;
- b) a rota das tarefas no conjunto de máquinas de cada job, e suas respectivas durações;
- c) a solução de referência, isto é, os instantes de início das tarefas em cada ciclo e a duração do ciclo.

Observa-se que as informações relacionadas no item c) anterior são obtidas utilizando-se a metodologia descrita no capítulo 3. Os instantes de início das primeiras tarefas de cada job no primeiro ciclo constituem as componentes do vetor $t^\#$. A duração do ciclo é o escalar $\lambda^\#$.

6.2.1 – Algoritmo de síntese I

Conforme descrito na seção 5.3. A restrição de estabilidade admite a multiplicação (no contexto max-plus) por um escalar qualquer solução que satisfaça a inequação 5.2. Assim sendo, esta estratégia consiste em calcular vários vetores de instantes de início factíveis, variando-se o escalar λ . Para obter em cada λ um vetor instante de início (se possível), escolhe-se como o primeiro job a ser disparo, aquele job que possui a menor c-valorização em $(t'_{i-1} \otimes M)$, os demais jobs seguem a mesma linha de raciocínio. De posse dos vários instantes de início, é selecionado como o próximo instante de início (t_i) aquele que possui a menor c-valorização em $(t'_i \otimes M)$, ou seja, aquele que fornece o job que disparará o mais cedo possível no próximo ciclo.

A seguir o algoritmo é apresentado.

Dados:

Condição inicial: t_0

Solução de regime: $t^\#, \lambda^\#, M$

- 1) $t = t_0$;
- 2) Enquanto a condição de parada não é satisfeita;

- 2.1) $\lambda = \lambda^\#; i = 0; \beta = t \otimes' (-t^\#); \alpha = t \otimes (-t^\#);$
- 2.2) Enquanto $\lambda < (\lambda^\#)^2$
 - 2.2.1) $t_+ = \lambda \otimes \beta \otimes t^\#;$
 - 2.2.2) Se t_+ não é factível então
 - 2.2.2.1) $t_+ = \text{c-val}(t'^T \otimes M)$
 - 2.2.2.2) Enquanto t_+ não for factível: alterar t_+
 - 2.2.3) Se $t_+ < \lambda \otimes \alpha \otimes t^\#$
 - 2.2.3.1) $i = i + 1$
 - 2.2.3.2) $tc(i) = t_+$
 - 2.2.4) $\lambda = \lambda + 1$ e volta para 2.2)
- 2.3) $i^* = \arg \min_i \min_j (\text{c-val}(tc'_j(i) \otimes M));$ onde $tc'_j(i)$ é a j -ésima componente do i -ésimo vetor da lista tc
- 2.4) $t = tc(i^*)$ e volta para 2)

Detalhamento do algoritmo:

Linha 2.2) → O “loop” definido pela instrução 2.2) busca, dentro de limites para λ um conjunto de soluções factíveis.

Linha 2.2.1) → A primeira solução a ser analisada é o limitante inferior da restrição de estabilidade.

Linha 2.2.2.1) → Se o limitante não é satisfeito a próxima tentativa é aquela que se inicia mais cedo, porém compatível como os ciclos precedentes.

Linha 2.2.2.2) → Se a solução anterior não for factível, t_+ pode ser alterado sistematicamente até se tornar factível (de acordo com a inequação 5.3).

Linha 2.2.3) → Se o resultado obtido satisfaz à restrição de estabilidade (5.2), o resultado é acrescentado à lista de soluções tc .

Linha 2.2.4) → Incrementa-se o deslocamento da solução e busca-se um novo candidato para a lista tc .

Linha 2.3) → Calcula-se, na lista de candidatos tc , a solução que permitirá no próximo ciclo, que o primeiro job a ser disparado seja mínimo.

Linha 2.4) → Esta solução é implementada.

6.2.2 – Algoritmo de síntese II

Esta estratégia é similar ao algoritmo de síntese I, sendo que a diferença entre elas está na forma da escolha do próximo instante de início (t_i). Nesta estratégia, de posse dos vários instantes de início, é selecionado como o próximo instante de início (t_i) aquele que fornece o menor valor de instante de início para o job que será o último a ser disparado no próximo ciclo, ou seja, o mínimo das máximas c-valorções de $(t'_i \otimes M)$.

A seguir o algoritmo é apresentado.

Este algoritmo tem suas linhas de instruções iguais ao do algoritmo I. Exceto a linha 2.3, que é escrita como a seguir:

2.3) $i^* = \arg \min_i \max_j (c\text{-val}(tc'_j(i) \otimes M))$; onde $tc'_j(i)$ é a j -ésima componente do i -ésimo vetor da lista tc

Detalhamento do algoritmo:

Linha 2.3) → Calcula-se, na lista de candidatos tc , a solução que permitirá no próximo ciclo, que o último job a ser disparado seja mínimo.

6.2.3 – Algoritmo de síntese III

Esta estratégia consiste em escolher os componentes do vetor de instante de início do próximo ciclo como a seguir. Seja $x = \min (c\text{-valorção de } ((t'_{i-1} \otimes M) \oplus (t'_i \otimes M^*)))$, com $t'_i = [0 \ 0 \ \dots \ 0]$. O primeiro job a ser disparado é o job que possui o coeficiente que determinou x , o segundo job a ser disparado é o que tem a menor c-valorção de $((t'_{i-1} \otimes M) \oplus (t'_i \otimes M^*))$, com $t'_i = [0 \ 0 \ \dots \ x \ \dots \ 0]$ desde que não tenha sido disparado no ciclo atual, assim sucessivamente até que t'_i esteja completo.

A seguir o algoritmo é apresentado.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Dados

Condição inicial: t_0

Solução de regime: $t^\#, \lambda^\#, M$

- 1) $t = t_0$;
- 2) Enquanto a condição de parada não é satisfeita;
 - 2.1) $\beta = t \otimes' (-t^\#)$; $\alpha = t \otimes (-t^\#)$; $\lambda = \lambda^\#$.
 - 2.2) $t_+ = \lambda \otimes \beta \otimes t^\#$;
 - 2.2.1) Se t_+ não é factível então
 - 2.2.1.1) $t_+ = [0 \ 0 \ \dots \ 0]$
 - 2.2.1.2) Enquanto todas as componentes do vetor t_+ não forem determinadas
 - 2.2.1.2.1) $x = \min(\text{c-val de } ((t' \otimes M) \oplus (t'_+ \otimes M^*)))$
 - 2.2.1.2.2) atualiza-se t_+ e volta para 2.2.1.2)
 - 2.2.2) Se $t_+ > \lambda \otimes \alpha \otimes t^\#$ então
 - 2.2.2.1) $\lambda = \lambda + 1$ e volta para 2.2)
 - 2.3) $t = t_+$ e volta para 2)

Detalhamento do algoritmo:

Linha 2.2) → A primeira solução a ser analisada é o limitante inferior da restrição de estabilidade.

Linha 2.2.1.1) → Se o limitante não é satisfeito o vetor de instante de início do próximo ciclo é construído passo a passo. O primeiro passo é determinar o job que se inicia mais cedo, porém compatível como os ciclos precedentes.

Linha 2.2.1.2) → O “loop” definido pela instrução 2.2.1.2) busca, determinar todas as componentes de t_+ .

Linha 2.2.1.2.1) → As componentes do vetor t_+ são definidas como sendo os instantes de início mais cedo que cada job pode ser disparado, levando-se em conta os jobs

já disparados. Observe-se que segundo o tratamento na seção 5.2, $t' = \tau^l$ e permite utilizar a formulação de dióides em série de potências.

Linha 2.2.1.2.2) → Atualizar t_+ implica em colocar a componente determinada no passo 2.2.1.2.1) no vetor

Linha 2.2.3) → Se o resultado obtido não satisfaz à restrição de estabilidade (5.2). Incrementa-se o deslocamento da solução e volta para 2.2).

Linha 2.3) → Esta solução é implementada.

6.2.4 – Algoritmo de síntese IV

Nesta estratégia, o próximo job a ser disparado é escolhido como aquele para o qual se estima maior perda, no caso dele não ser o próximo job disparado. Define-se a “função perda” para cada job j representado pela componente $g_j(k)$ do vetor $\mathbf{g}$ como a seguir. Suponha que o job j não é o próximo job disparado, mas ele será disparado posteriormente. Considere que os instantes quando o job j poderá ser disparado é uma variável aleatória T . No futuro, os valores de k para os quais $g_j(k) = 0$ são, a priori, infactíveis, e aqueles para os quais $g_j(k) = 1$ serão considerados equiprovavelmente possíveis. Entretanto a probabilidade que $T > t+H$ é zero, onde H é um horizonte determinado em função de considerações práticas. Pode-se portanto propor uma função densidade de probabilidade $p_j(k)$ para a variável aleatória T . Primeiramente calcula-se a “área A ” sob a função $g_j(k)$ de t até $t+H$. Com isto $p_j(k)$ é dado como a seguir:

$$p_j(k) = \begin{cases} 0 & \text{se } k \geq t+H \text{ ou } k \leq t \\ 0 & \text{se } g_j(k) = 1 \\ \frac{1}{A} & \text{se } g_j(k) = 0 \text{ e } t \leq k \leq t+H \end{cases}$$

onde:

t é igual ao valor do instante de início do último job disparado no ciclo anterior;

H é um horizonte de tempo que representa um limitante superior para a expectativa de disparo do job, neste trabalho ele é igual ao tamanho do maior job, ou seja, a maior

diferença entre a ultima tarefa mais seu tempo de processamento com a primeira tarefa do job.

A é igual a área sob a função $g_j(k)$

Pode-se agora calcular $E[T]$, a esperança (valor esperado) da variável aleatória T que é uma estimativa do instante de tempo no qual o job j estará habilitado para disparar no futuro. Para um job j , a função perda é definida como:

$$L' = E[T] - c\text{-val}(t' \otimes M)$$

Assim, o próximo job a ser disparado é aquele cuja componente de L' é máxima. Repete-se este procedimento até que todos os jobs sejam disparados, lembrando sempre em atualizar $g = (t' \otimes M) \oplus (t'_i \otimes M^*)$ para o cálculo da esperança matemática e da valoração.

A seguir o algoritmo é apresentado.

Este algoritmo tem suas linhas de instruções iguais ao do algoritmo III. Exceto a linha 2.2.1.2.1, que é escrita como a seguir:

$$2.2.1.2.1) \ x = \min\{-E[T] + c\text{-val}[(t' \otimes M) \oplus (t_+ \otimes M)]\}$$

Detalhamento do algoritmo:

Linha 2.2.1.2.1) $\rightarrow$ As componentes do vetor t_+ são definidas como sendo os instantes de início cuja função perda apresenta maior valor, levando-se em conta os jobs já disparados.

6.3 – Resultados Obtidos

Nesta seção serão apresentados os problemas de job shop cíclico, cujas soluções são previamente conhecidas quando ele está em regime, assim como os resultados obtidos empregando-se os algoritmos da seção anterior.

Cabe neste instante ressaltar algumas condições que serão adotadas no restante deste capítulo. O primeiro vetor com os instantes de início, será chamado de condição inicial e será denotado por t_0 . Portanto, o vetor t_0 conterá uma solução factível porém diferente da solução de regime, que representará os instantes de início de cada job, após a extinção da falha, sem que haja conflito entre os jobs.

As soluções obtidas simplesmente retardando a solução de referência, até que ela possa ser novamente utilizada, será chamada de solução de regime atrasada, e ela é determinada como descrito a seguir. Faz-se o produto de t'_0 por M , com isto obtém-se um vetor de dimensão igual ao número de jobs, estes elementos pertencem ao monóide S_Δ e os instantes de início do próximo ciclo não deverão pertencer ao suporte destes elementos. Portanto deslocando-se a solução de regime em algum momento todas as componentes do vetor t pertencerão simultaneamente a regiões factíveis dos elementos de $t'_0 \otimes M \in S_\Delta$.

Utilizar-se-á para a apresentação dos problemas e suas respectivas soluções, as seguintes tabelas:

- i) uma tabela contendo os dados do problema incluindo a solução de regime;
- ii) uma tabela contendo as soluções, ou seja, os instantes de início da primeira tarefa de cada job nos ciclos posteriores a falha, utilizando-se a seguinte nomenclatura:

Regime – solução de regime;

Reg. Atras. – solução de regime atrasada;

Algoritmo 1 – solução obtida com o algoritmo de síntese I;

Algoritmo 2 – solução obtida com o algoritmo de síntese II;

Algoritmo 3 – solução obtida com o algoritmo de síntese III;

Algoritmo 4 – solução obtida com o algoritmo de síntese IV.

As tabelas que contém os resultados (instantes de início) dos exemplos que serão apresentados nesta seção, foram construídas da seguinte forma. A primeira coluna informa o número do ciclo. A segunda coluna informa os instantes de início da solução de regime, sendo portanto informado sempre o instante de início de um mesmo job. A terceira coluna informa os instantes de início da solução de regime atrasada, como a primeira coluna informa sempre o

instante de início de um mesmo job. Já as demais colunas informam os instantes de início que foram obtidos utilizando-se os algoritmos propostos. Vale salientar que nem sempre o instante de início apresentado é de um mesmo job, pois dependerá da sequência determinada. É claro que a partir de um certo número de ciclos, considerando a convergência à solução de regime os instantes de início que demarcam o início do ciclo passam a ser de um mesmo job.

Nas colunas da tabela que contém os instantes de início do primeiro job determinados pelos algoritmos propostos, os valores coloridos correspondem ao ciclo em que a solução convergiu para a solução de referência.

Nos gráficos serão plotados os instantes de início que estão presentes nas tabelas de cada exemplo, valendo ressaltar que nem sempre aparecerá nestes gráficos todas as cores que representam cada coluna das tabelas. Este fato ocorre quando dois ou mais algoritmos apresentam a mesma solução e somente a cor do último aparece.

Exemplo 1

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)
			Duração das Tarefas nas Máquinas
1	7	2 3 4 3 5 4 3	72 84 133 260 308 340 363
			12 2 11 2 15 13 3
2	8	3 4 3 4 1 4 1 4	66 110 158 182 196 219 250 255
			3 11 4 14 5 12 5 13
3	6	4 1 5 2 5 4	0 84 98 112 181 268
			12 5 14 13 15 12

Tabela 6.1 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 98$, e $t^{\#} = [72 \quad 66 \quad 0]$

i) – Para a condição inicial $t_0 = [0 \quad 34 \quad 124]$, e utilizando-se a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	98	236	132	139	132	130
Ciclo 3	196	334	262	297	262	303
Ciclo 4	294	432	360	351	360	352
Ciclo 5	392	530	458	459	458	483
Ciclo 6	490	628	556	557	556	581
Ciclo 7	588	726	654	655	654	679
Ciclo 8	686	824	752	753	752	777
Ciclo 9	784	922	850	851	850	875
Ciclo 10	882	1020	948	949	948	973

Tabela 6.2 – Instantes de início do primeiro job de cada seqüência.

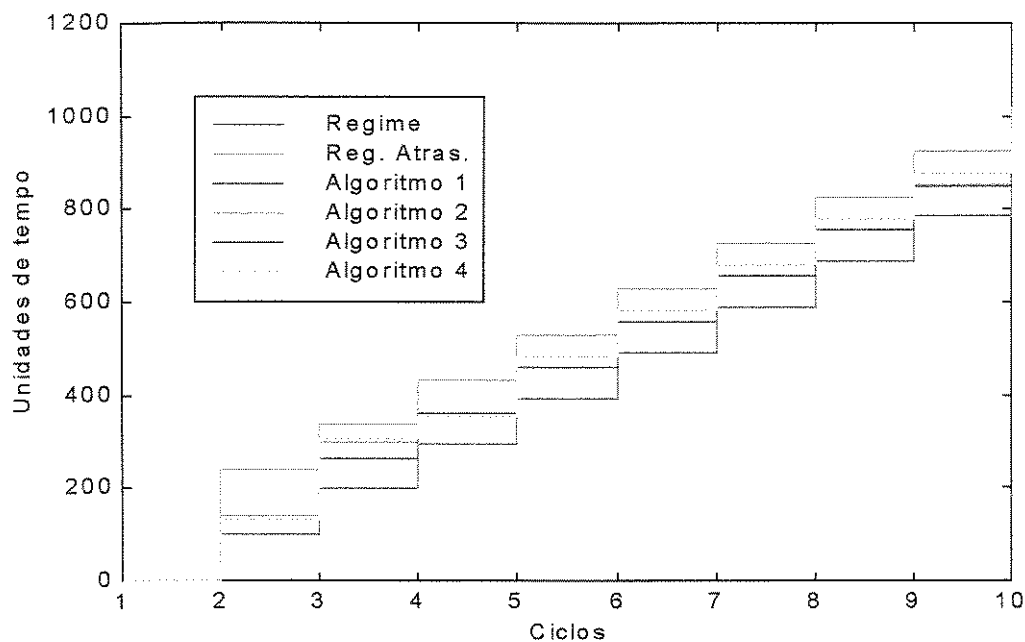


Figura 6.1 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.2 e a figura 6.1, observa-se que para esta condição inicial todos os algoritmos propostos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 3, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no quarto ciclo, e a solução obtida como o algoritmo 4 converge no quinto ciclo.

ii) – Para a condição inicial $t_0 = [44 \quad 0 \quad 83]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	98	230	98	98	108	86
Ciclo 3	196	328	228	228	304	254
Ciclo 4	294	426	326	326	402	352
Ciclo 5	392	524	424	424	500	450
Ciclo 6	490	622	522	522	598	548
Ciclo 7	588	720	620	620	696	646
Ciclo 8	686	818	718	718	794	744
Ciclo 9	784	916	816	816	892	842
Ciclo 10	882	1014	914	914	990	940

Tabela 6.3 – Instantes de início do primeiro job de cada sequência.

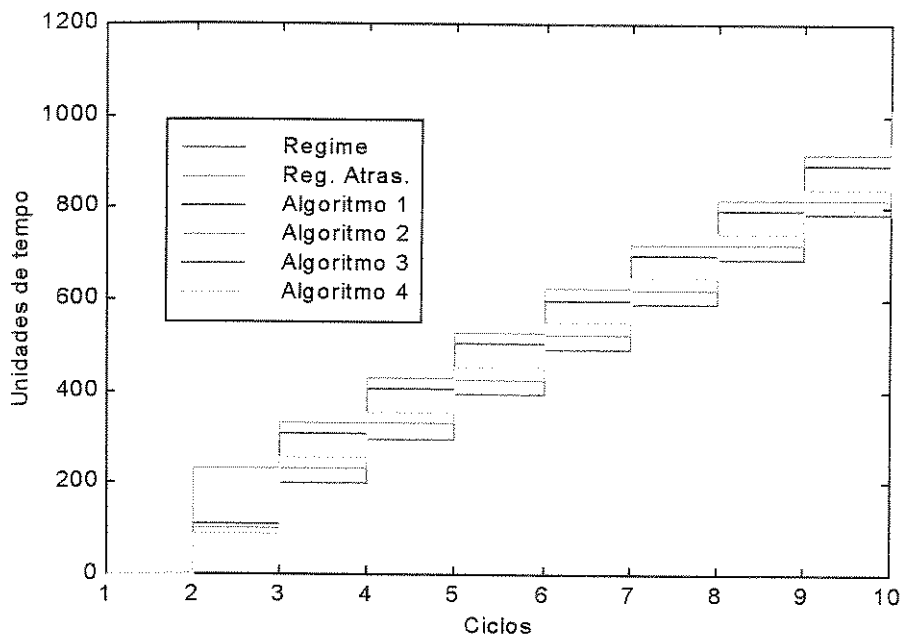


Figura 6.2 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.3 e a figura 6.2, observa-se que para esta condição inicial todos os algoritmos propostos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2, 3 e 4, ocorrem no terceiro ciclo, contudo estas soluções não apresentam valores iguais.

iii) – Para a condição inicial $t_0 = [40 \ 85 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	98	215	112	215	112	99
Ciclo 3	196	313	301	373	272	301
Ciclo 4	294	411	399	555	447	399
Ciclo 5	392	509	497	686	549	497
Ciclo 6	490	607	595	872	720	595
Ciclo 7	588	705	693	1059	950	693
Ciclo 8	686	803	791	1157	1048	791
Ciclo 9	784	901	889	1255	1244	889
Ciclo 10	882	999	987	1353	1437	987

Tabela 6.4 – Instantes de início do primeiro job de cada seqüência.

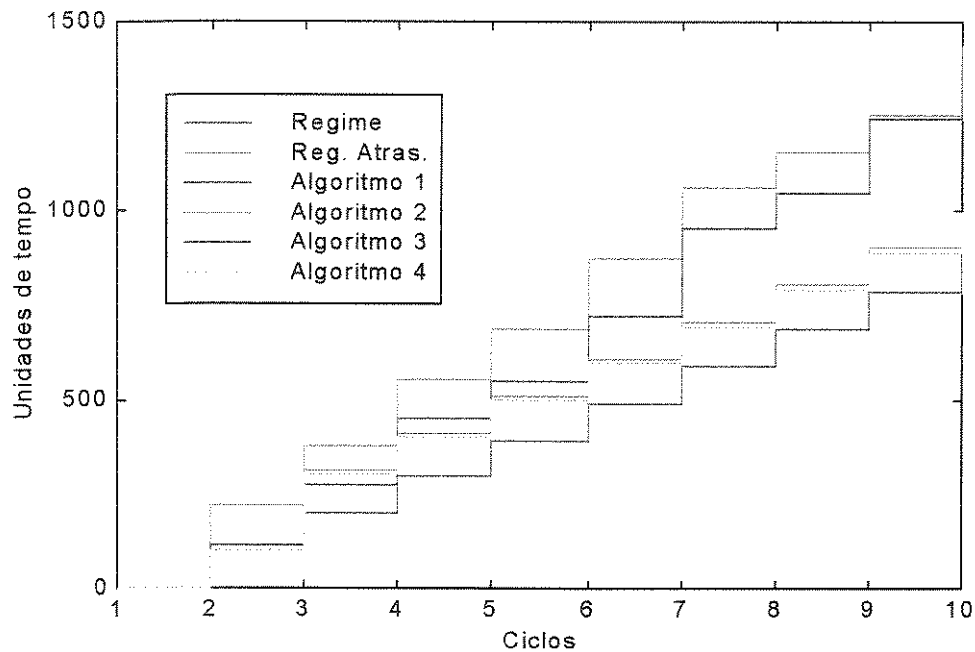


Figura 6.3 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.4 e a figura 6.3, observa-se que para esta condição inicial os algoritmos 1 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Já o algoritmo 2 tem a sua solução convergida para um valor superior à solução de regime atrasada, portanto não caracterizando melhora na produção. O algoritmo 3 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a

convergência das soluções obtidas com os algoritmos 1 e 4, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 2 converge no sétimo ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.2, 6.3 e 6.4 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	138	66	67	66	91
Caso ii)	0	132	32	32	108	58
Caso iii)	0	117	105	471	555	105

Tabela 6.5 – Tempos relativos do exemplo 1

Observa-se da tabela 6.5 que para este exemplo, o algoritmo 1 é o que apresenta os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram valores menores. O algoritmo 4 também tem suas soluções convergindo para a solução de referência mas apresenta atrasos maiores em relação a solução de regime quando comparados com o algoritmo 1. O algoritmo 2 embora tenha atrasos em relação a solução de regime menores do que o algoritmo 4 para os casos i) e ii), apresenta para o caso iii) um atraso alto em relação a solução de referência ficando acima do solução de regime atrasada. O algoritmo 3 é que apresenta o pior resultado global, pois para o caso ii) é que apresenta o maior atraso e no caso iii) após dez ciclos não havia ainda sua solução convergida para a solução de referência.

Exemplo 2

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (solução de regime)
			Duração das Tarefas nas Máquinas
1	5	1 3 1 4 1	0 114 125 146 190
			11 6 15 2 11
2	10	1 5 3 4 3 1 5 3 2 5	26 39 50 72 82 140 163 211 222 225
			12 11 9 3 10 12 15 8 3 14
3	9	3 5 3 4 3 4 2 1 2	67 102 120 143 149 158 159 163 218
			6 13 6 3 9 1 4 15 4

Tabela 6.6 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 76$, e $t^{\#} = [0 \quad 26 \quad 67]$

i) – Para a condição inicial $t_0 = [0 \quad 49 \quad 14]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	76	175	76	84	84	76
Ciclo 3	152	251	176	198	198	291
Ciclo 4	228	327	290	274	274	379
Ciclo 5	304	403	366	350	350	455
Ciclo 6	380	479	442	426	426	668
Ciclo 7	456	555	518	502	502	773
Ciclo 8	532	631	594	578	578	917
Ciclo 9	608	707	670	654	654	1024
Ciclo 10	684	783	746	730	730	1191

Tabela 6.7 – Instantes de início do primeiro job de cada seqüência.

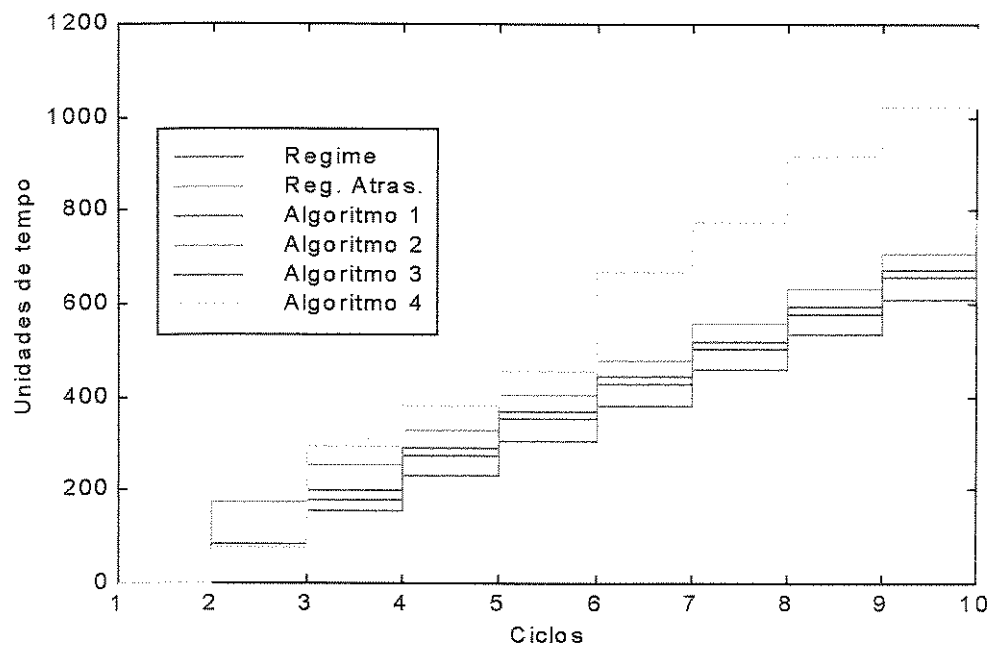


Figura 6.4 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.7 e a figura 6.4, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. O algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 3, ocorrem no quarto ciclo, sendo que os algoritmos 2 e 3 apresentam resultados iguais.

ii) – Para a condição inicial $t_0 = [43 \quad 0 \quad 33]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	76	183	88	81	81	49
Ciclo 3	152	259	202	236	221	231
Ciclo 4	228	335	278	438	398	339
Ciclo 5	304	411	354	547	599	418
Ciclo 6	380	487	430	748	800	500
Ciclo 7	456	563	506	949	978	640
Ciclo 8	532	639	582	1150	1054	808
Ciclo 9	608	715	658	1328	1130	917
Ciclo 10	684	791	734	1404	1206	1057

Tabela 6.8 – Instantes de início do primeiro job de cada seqüência.

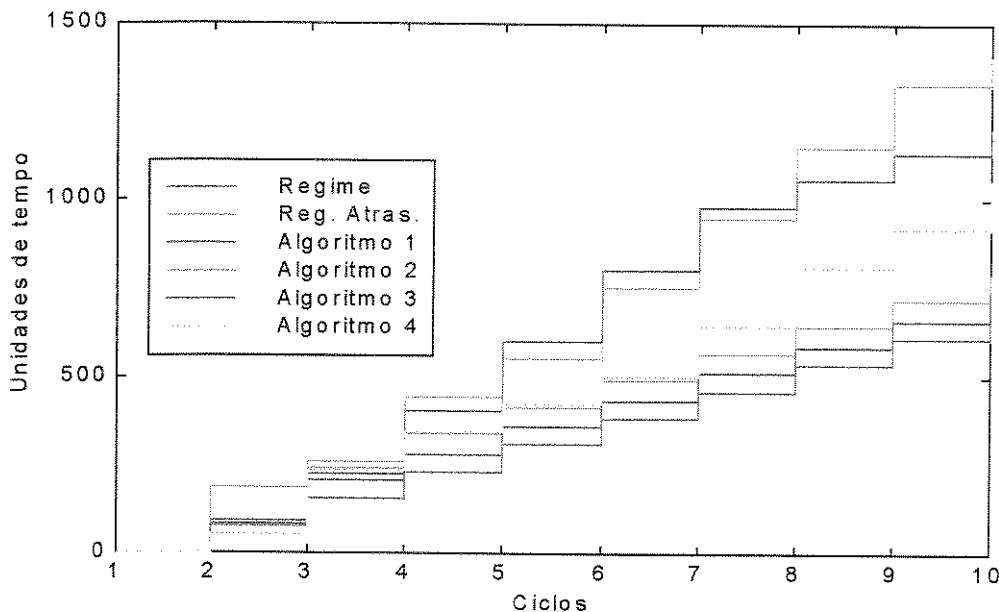


Figura 6.5 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.8 e a figura 6.5, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores superiores à solução de regime, porém somente o algoritmo 2 apresenta valores menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já os algoritmos 2 e 3 têm suas soluções convergidas para valores superiores a solução de regime atrasada, portanto não caracterizando melhora na produção. O algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência da solução obtida com o algoritmo 1, ocorre no terceiro ciclo, enquanto as soluções obtidas com os algoritmos 2 e 3 convergem no nono e no sétimo ciclo respectivamente.

iii) – Para a condição inicial $t_0 = [69 \quad 35 \quad 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	76	183	76	76	112	69
Ciclo 3	152	259	161	151	248	247
Ciclo 4	228	335	237	237	353	355
Ciclo 5	304	411	313	313	430	434
Ciclo 6	380	487	389	389	507	516
Ciclo 7	456	563	465	465	581	656
Ciclo 8	532	639	541	541	661	824
Ciclo 9	608	715	617	617	742	933
Ciclo 10	684	791	693	693	818	1073

Tabela 6.9– Instantes de início do primeiro job de cada sequência.

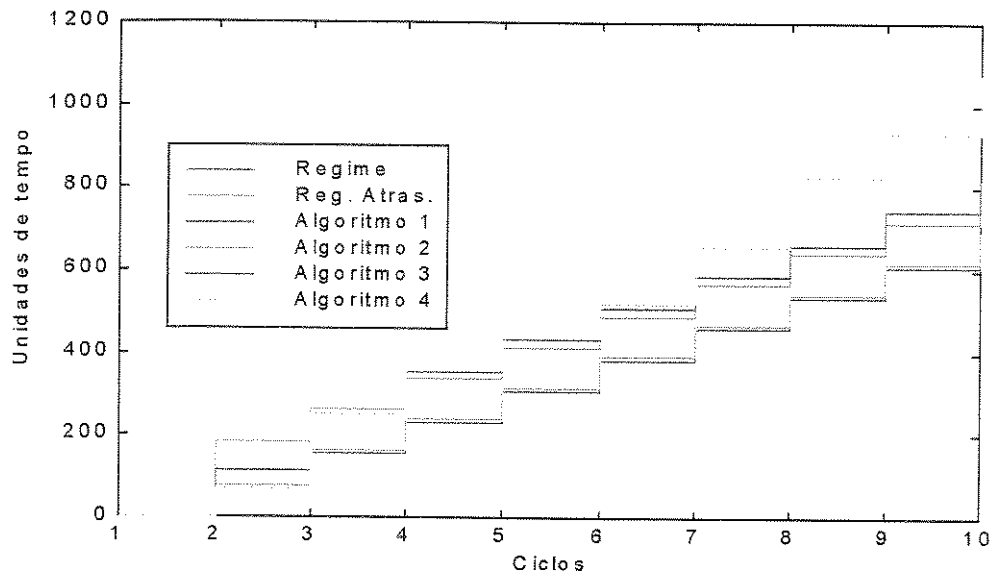


Figura 6.6 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.9 e a figura 6.6, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores superiores à solução de regime, porém apenas os algoritmos 1 e 2 apresentam valores menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. O algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida, esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no nono ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.7, 6.8 e 6.9 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	99	62	46	46	507
Caso ii)	0	107	50	720	522	373
Caso iii)	0	107	9	9	134	389

Tabela 6.10 – Tempos relativos do exemplo 2

Observa-se da tabela 6.10 que para este exemplo, o algoritmo 1 é o que apresenta os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram menores valores, exceto no caso i) . O algoritmo 2 também tem suas soluções convergindo para a solução de referência. Embora apresente um melhor resultado que o algoritmo 1 no caso i) e igual no caso iii), no caso ii) apresenta um atraso alto em relação a solução de referência ficando acima do solução de regime atrasada. O algoritmo 3 tem apenas no caso i) um resultado melhor do que a solução de regime atrasada, nos demais casos apresenta atrasos maiores do que ela. O algoritmo 4 é o que apresenta o pior resultado global, pois para todos os casos até o décimo ciclo não havia ainda sua solução convergida para a solução de referência.

Exemplo 3

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)				
			Duração das Tarefas nas Máquinas				
1	5	1 2 3 4 5	0	2	6	32	43
			2	4	15	11	8
2	5	4 2 4 1 2	19	57	94	150	167
			13	5	12	3	1
3	5	2 3 4 3 2	11	21	55	87	98
			3	15	15	11	4

Tabela 6.11 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 51$, e $t^{\#} = [0 \quad 19 \quad 11]$

i) – Para a condição inicial $t_0 = [0 \quad 43 \quad 56]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	51	137	98	98	98	56
Ciclo 3	102	188	173	173	173	180
Ciclo 4	153	239	224	224	224	231
Ciclo 5	204	290	275	275	275	282
Ciclo 6	255	341	326	326	326	333
Ciclo 7	306	392	377	377	377	384
Ciclo 8	357	443	428	428	428	435
Ciclo 9	408	494	479	479	479	486
Ciclo 10	459	545	530	530	530	537

Tabela 6.12 – Instantes de início do primeiro job de cada sequência.

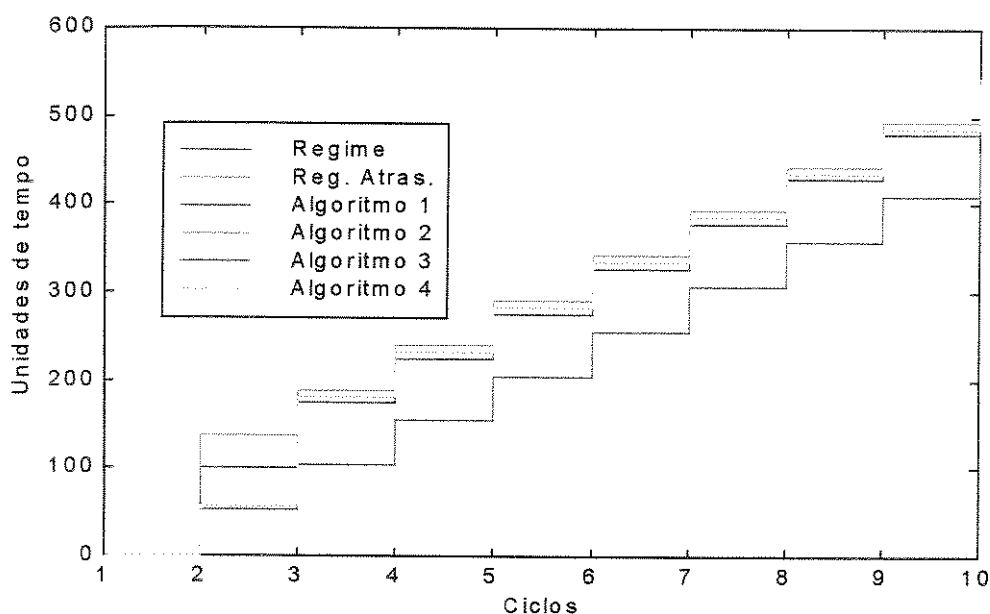


Figura 6.7 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.12 e a figura 6.7, observa-se que para esta condição inicial todos os algoritmos propostos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com todos os algoritmos ocorrem no terceiro ciclo.

ii) – Para a condição inicial $t_0 = [28 \ 0 \ 50]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	51	139	77	77	77	65
Ciclo 3	102	190	179	179	188	167
Ciclo 4	153	241	230	230	239	218
Ciclo 5	204	292	281	281	290	269
Ciclo 6	255	343	332	332	341	320
Ciclo 7	306	394	383	383	392	371
Ciclo 8	357	445	434	434	443	422
Ciclo 9	408	496	485	485	494	473
Ciclo 10	459	547	536	536	545	524

Tabela 6.13 – Instantes de início do primeiro job de cada sequência.

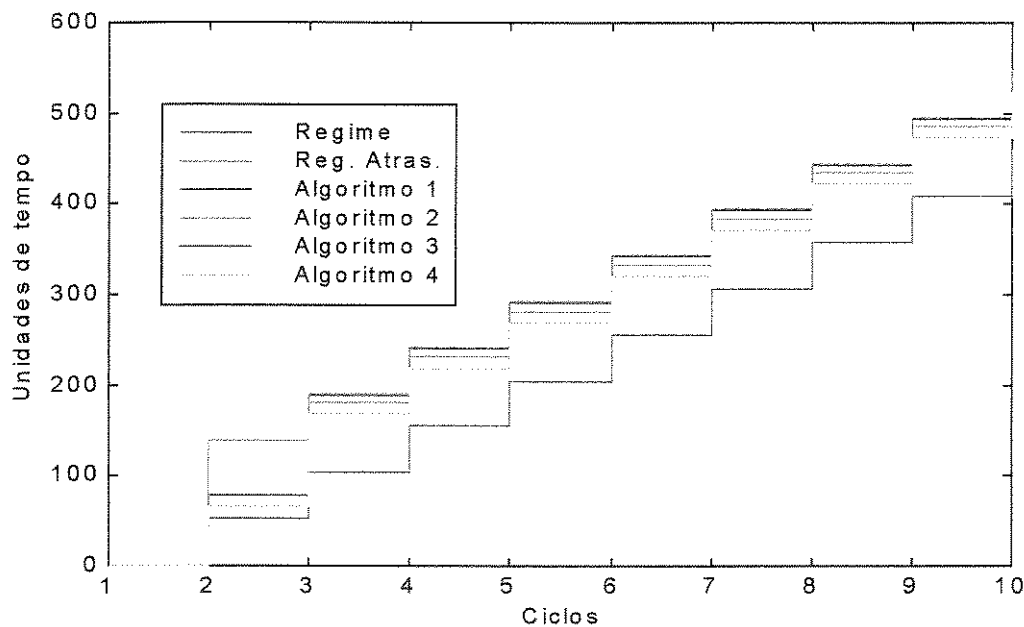


Figura 6.8 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.13 e a figura 6.8, observa-se que para esta condição inicial todos os algoritmos propostos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com todos os algoritmos ocorrem no terceiro ciclo.

iii) – Para a condição inicial $t_0 = [27 \ 26 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	51	94	42	42	42	42
Ciclo 3	102	145	117	117	119	100
Ciclo 4	153	196	168	168	206	187
Ciclo 5	204	247	219	219	312	230
Ciclo 6	255	298	270	270	404	311
Ciclo 7	306	349	321	321	496	338
Ciclo 8	357	400	372	372	588	396
Ciclo 9	408	451	423	423	680	503
Ciclo 10	459	502	474	474	772	530

Tabela 6.14 – Instantes de início do primeiro job de cada sequência.

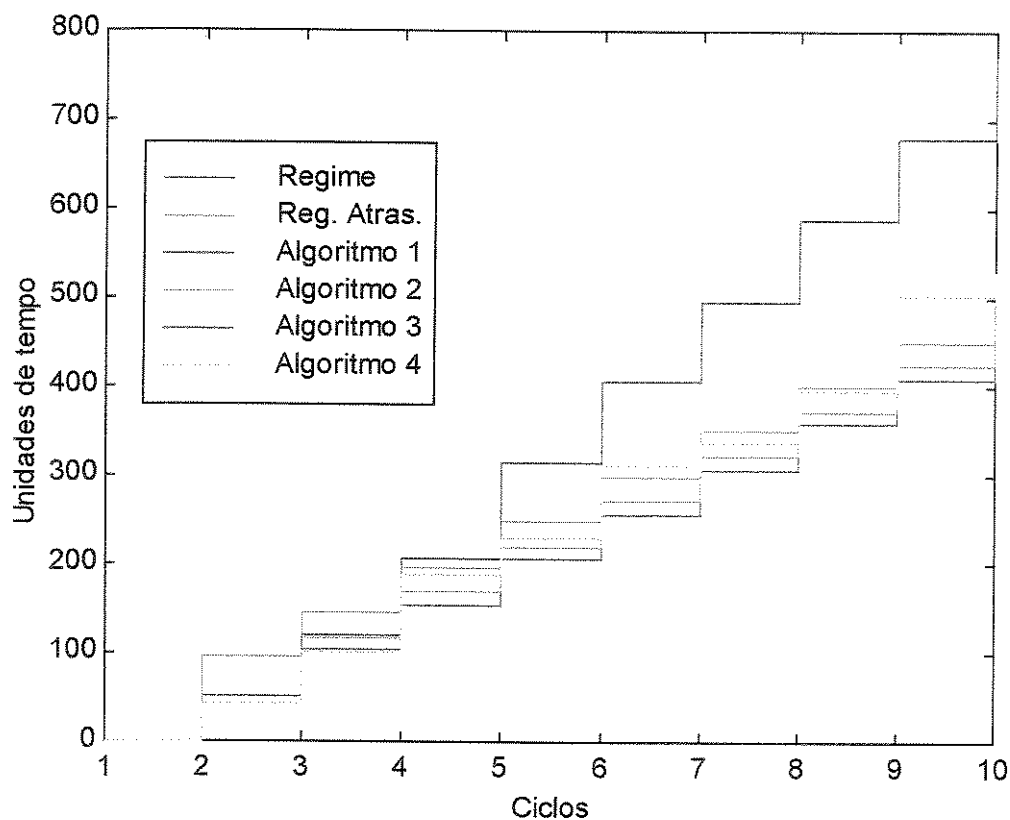


Figura 6.9 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.14 e a figura 6.9, observa-se que para esta condição inicial os algoritmos 1 e 2 tiveram suas soluções convergindo para valores superiores à solução de regime, porém apresentam valores menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já os

algoritmos 3 e 4 até o décimo ciclo ainda não tinham suas soluções convergidas. Estas convergências ocorrerão nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo, e possuem valores iguais.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.12, 6.13 e 6.14 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	86	71	71	71	78
Caso ii)	0	88	77	77	86	65
Caso iii)	0	43	15	15	313	71

Tabela 6.15 – Tempos relativos do exemplo 3

Observa-se da tabela 6.15 que para este exemplo, os algoritmos 1 e 2 são os que apresentam os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram valores menores em relação a solução de referência atrasada. O algoritmo 4 apresenta um melhor resultado que os algoritmos 1 e 2 no caso ii), mas para os outros casos apresenta atrasos maiores sendo que no caso iii) sua solução fica acima do solução de regime atrasada. O algoritmo 3 é o que apresenta o pior resultado global.

Exemplo 4

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)					
			Duração das Tarefas nas Máquinas					
1	5	1 3 5 3 5	65	121	126	169	243	
			12	5	12	4	15	
2	10	4 5 3 1 3 2 4 5 4 3	0	6	60	92	107	113
			6	14	1	15	3	5
3	7	5 2 3 4 1 3 4	33	45	111	138	147	176
			12	2	5	9	11	1

Tabela 6.16 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 66$, e $t^{\#} = [65 \quad 0 \quad 33]$

i) – Para a condição inicial $t_0 = [0 \quad 34 \quad 124]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	66	118	76	76	76	52
Ciclo 3	132	184	175	175	175	131
Ciclo 4	198	250	241	241	241	247
Ciclo 5	264	316	307	307	307	313
Ciclo 6	330	382	373	373	373	379
Ciclo 7	396	448	439	439	439	445
Ciclo 8	462	514	505	505	505	511
Ciclo 9	528	580	571	571	571	577
Ciclo 10	594	646	637	637	637	643

Tabela 6.17 – Instantes de início do primeiro job de cada sequência.

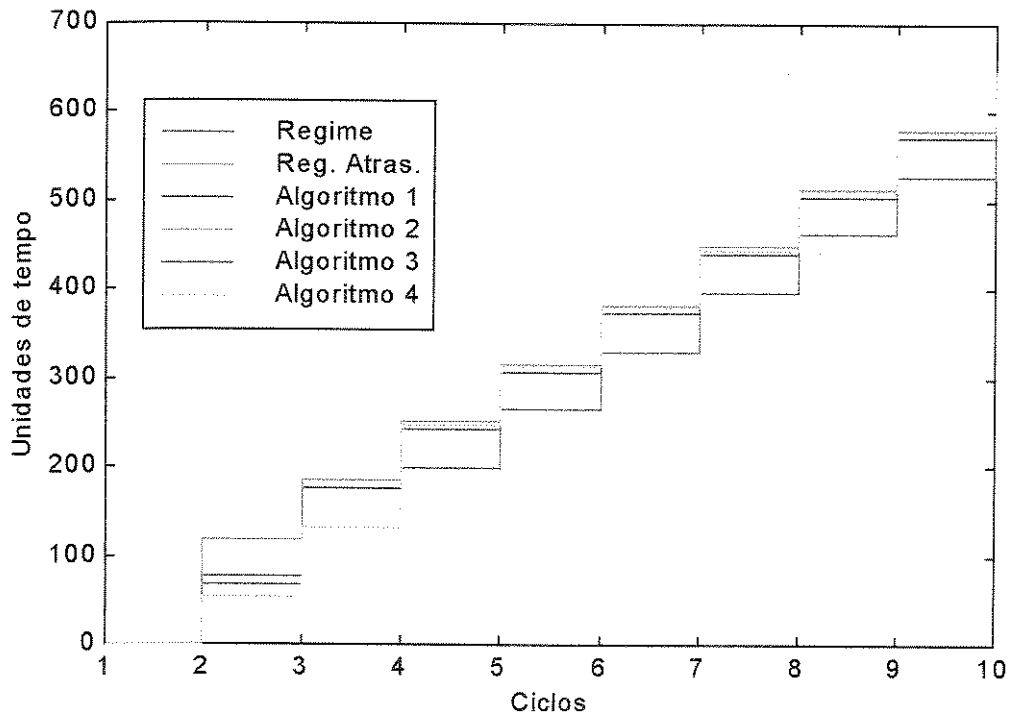


Figura 6.10 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.17 e a figura 6.10, observa-se que para esta condição inicial todos os algoritmos propostos tiveram suas soluções convergindo para valores superiores à

solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com todos os algoritmos 1,2 e 3 ocorrem no terceiro ciclo, já com o algoritmo 4 ocorre no quarto ciclo.

ii) – Para a condição inicial $t_0 = [69 \quad 0 \quad 23]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	66	159	71	70	159	70
Ciclo 3	132	225	203	202	287	239
Ciclo 4	198	291	269	268	393	291
Ciclo 5	264	357	335	334	458	367
Ciclo 6	330	423	401	400	617	408
Ciclo 7	396	489	467	466	683	493
Ciclo 8	462	555	533	532	749	535
Ciclo 9	528	621	599	598	815	618
Ciclo 10	594	687	665	664	881	660

Tabela 6.18 – Instantes de início do primeiro job de cada sequência.

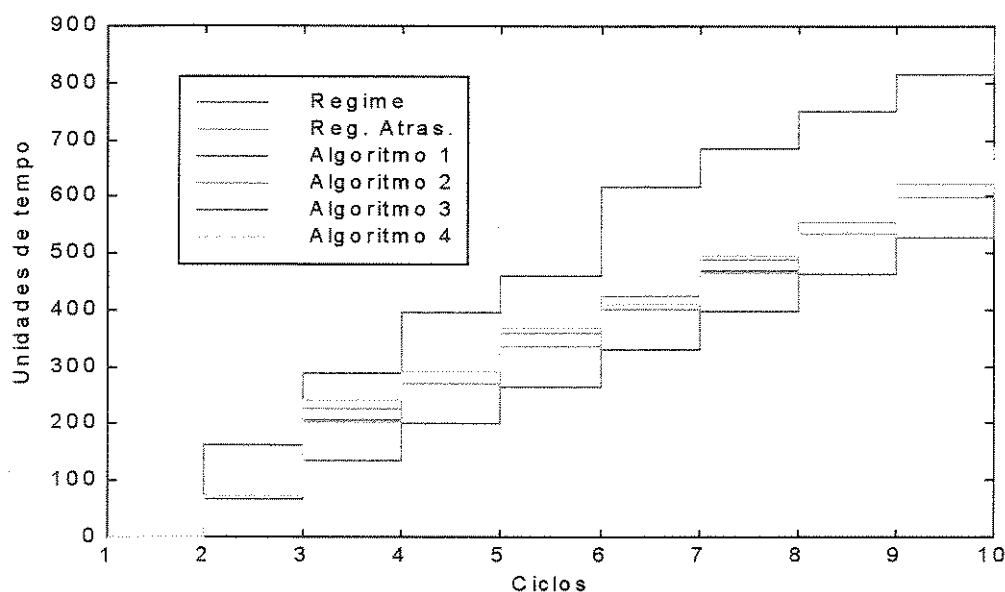


Figura 6.11 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.18 e a figura 6.11, observa-se que para esta condição inicial os algoritmos 1 e 2 tiveram suas soluções convergindo para valores superiores à solução de regime, porém apresentam valores menores que a solução de regime atrasada. Evidenciando

um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 4, embora apresente valores menores que a solução de regime atrasada até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo, já com o algoritmo 3 ocorre no sexto ciclo.

iii) – Para a condição inicial $t_0 = [27 \ 34 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	66	114	66	66	66	42
Ciclo 3	132	180	165	165	206	172
Ciclo 4	198	246	231	231	307	238
Ciclo 5	264	312	297	297	404	304
Ciclo 6	330	378	363	363	469	370
Ciclo 7	396	444	429	429	628	436
Ciclo 8	462	510	495	495	694	502
Ciclo 9	528	576	561	561	760	568
Ciclo 10	594	642	627	627	826	634

Tabela 6.19 – Instantes de início do primeiro job de cada sequência.

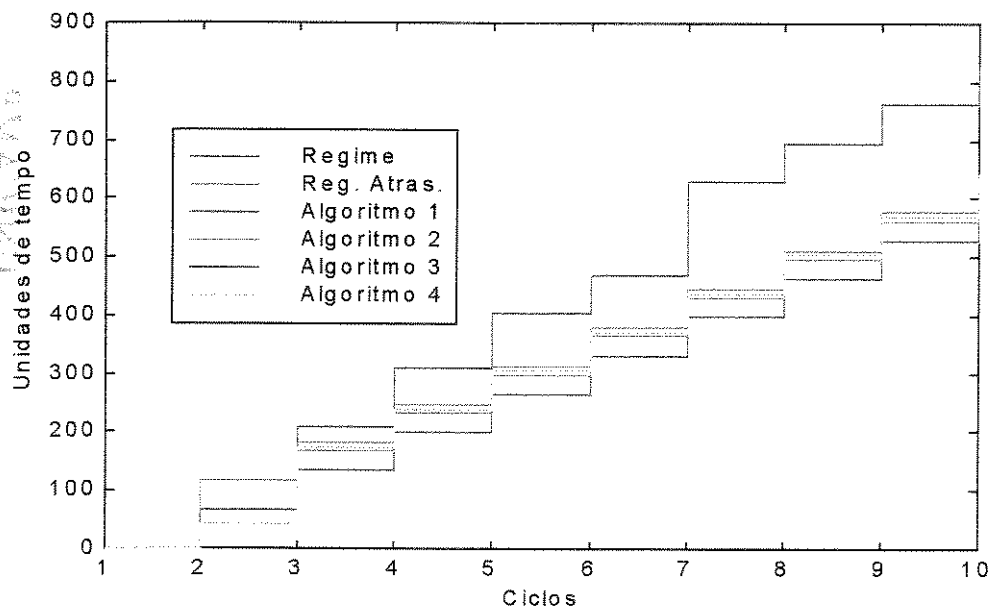


Figura 6.12 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.19 e a figura 6.12, observa-se que para esta condição inicial os algoritmos 1, 2 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 4, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no sétimo ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.17, 6.18 e 6.19 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	52	43	43	43	49
Caso ii)	0	93	71	70	287	66
Caso iii)	0	48	33	33	184	40

Tabela 6.20 – Tempos relativos do exemplo 4.

Observa-se da tabela 6.20 que para este exemplo, os algoritmos 1 e 2 são os que apresentam os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram valores menores em relação a solução de referência atrasada. O algoritmo 4 apresenta um melhor resultado que os algoritmos 1 e 2 no caso ii), mas para os outros casos apresenta atrasos maiores. O algoritmo 3 é o que apresenta o pior resultado global.

Exemplo 5

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)				
			Duração das Tarefas nas Máquinas				
1	5	2 1 3 5 2	22	29	87	108	116
			7	4	15	8	8
2	5	2 5 3 2 4	53	63	75	102	108
			10	10	12	6	15
3	7	3 5 3 2 5 4 5	25	37	48	67	73 80 95
			12	9	13	6	7 15 7
4	6	3 1 5 1 5 3	0	11	16	28	29 37
			11	5	9	1	8 11
5	6	5 3 4 2 3 2	0	11	31	46	61 94
			6	14	15	7	14 8

Tabela 6.21 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 102$, e $t^{\#} = [22 \ 53 \ 25 \ 0 \ 0]$

i)- Para a condição inicial $t_0 = [14 \ 121 \ 0 \ 94 \ 160]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	102	240	160	160	160	160
Ciclo 3	204	342	340	340	273	326
Ciclo 4	306	444	442	442	411	428
Ciclo 5	408	546	544	544	586	530
Ciclo 6	510	648	646	646	688	632
Ciclo 7	612	750	748	748	790	734
Ciclo 8	714	852	850	850	892	836
Ciclo 9	816	954	952	952	994	938
Ciclo 10	918	1056	1054	1054	1096	1040

Tabela 6.22 – Instantes de início do primeiro job de cada sequência.

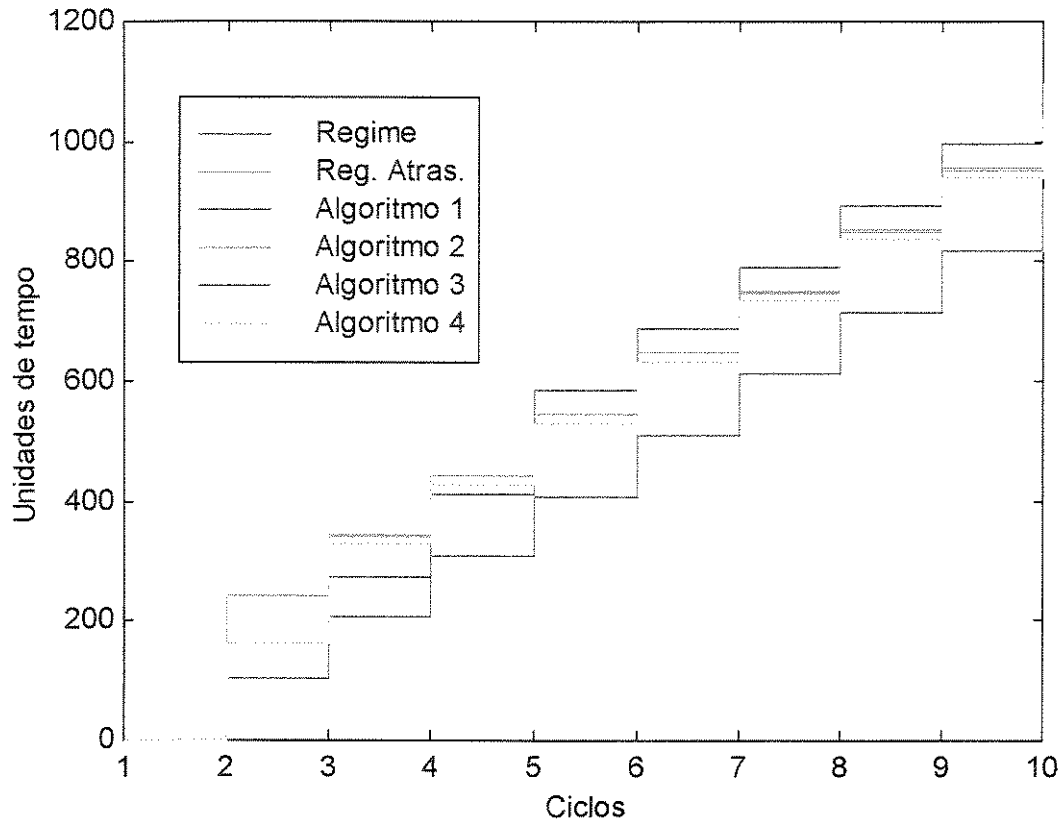


Figura 6.13 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.22 e a figura 6.13, observa-se que para esta condição inicial os algoritmos 1, 2 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 4, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no quinto ciclo.

ii) – Para a condição inicial $t_0 = [22 \ 130 \ 25 \ 0 \ 169]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	102	249	169	169	169	169
Ciclo 3	204	351	349	349	282	335
Ciclo 4	306	453	451	451	420	437
Ciclo 5	408	555	553	553	595	539
Ciclo 6	510	657	655	655	697	641
Ciclo 7	612	759	757	757	799	743
Ciclo 8	714	861	859	859	901	845
Ciclo 9	816	963	961	961	1003	947
Ciclo 10	918	1065	1063	1063	1105	1049

Tabela 6.23 – Instantes de início do primeiro job de cada sequência.

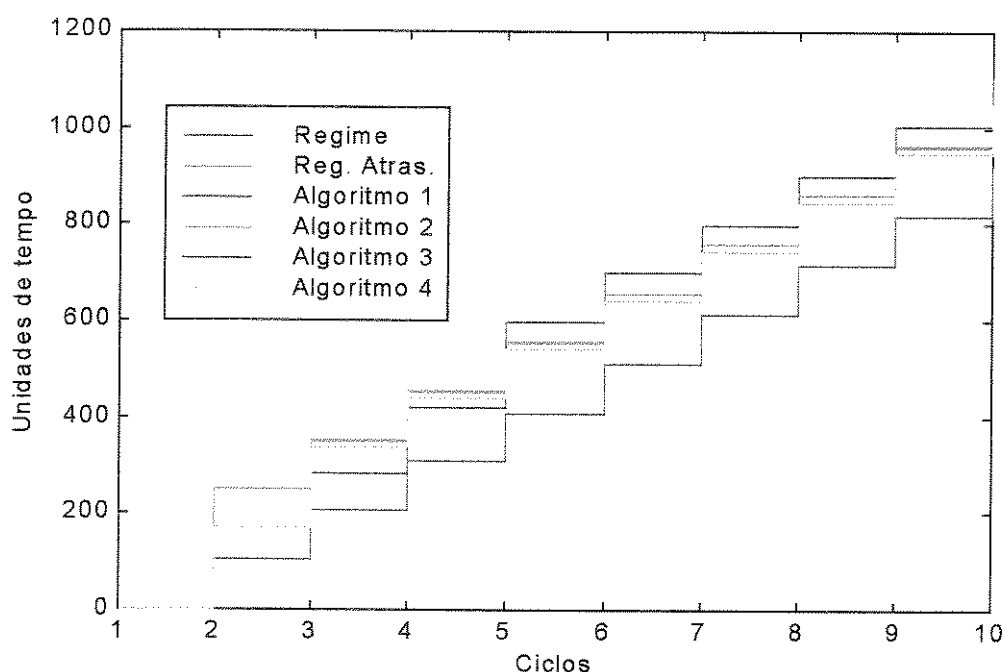


Figura 6.14 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.23 e a figura 6.14, observa-se que para esta condição inicial os algoritmos 1, 2 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com

os algoritmos 1, 2 e 4, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no quinto ciclo.

iii) – Para a condição inicial $t_0 = [0 \ 102 \ 29 \ 4 \ 141]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	102	221	141	141	141	141
Ciclo 3	204	323	321	321	254	307
Ciclo 4	306	425	423	423	392	409
Ciclo 5	408	527	525	525	567	511
Ciclo 6	510	629	627	627	669	613
Ciclo 7	612	731	729	729	771	715
Ciclo 8	714	833	831	831	873	817
Ciclo 9	816	935	933	933	975	919
Ciclo 10	918	1037	1035	1035	1077	1021

Tabela 6.24 – Instantes de início do primeiro job de cada seqüência.

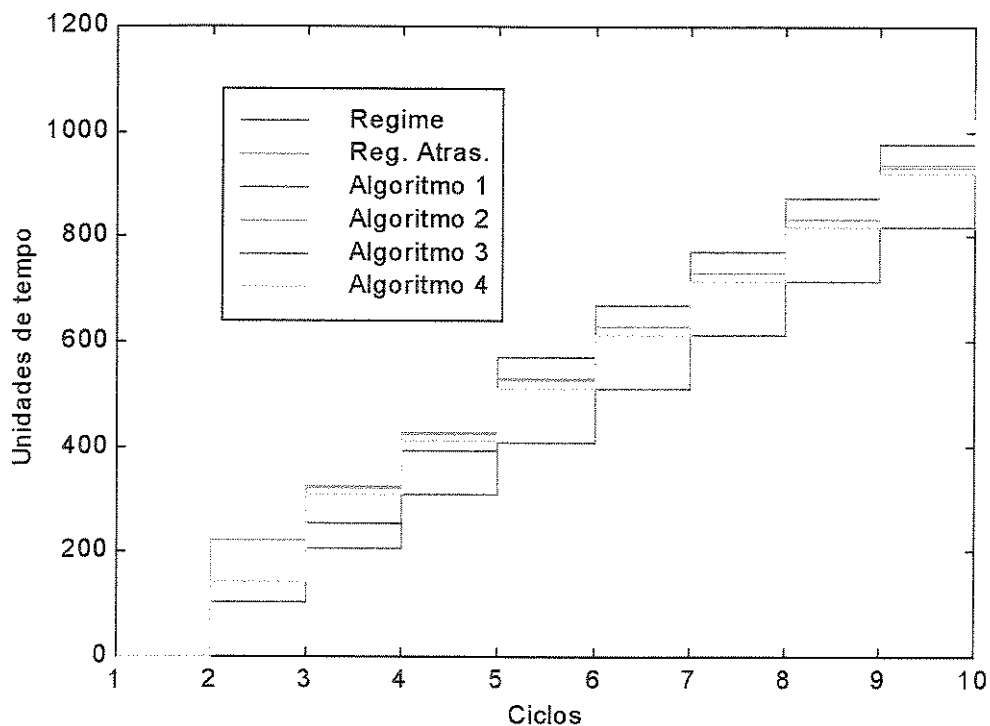


Figura 6.15 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.24 e a figura 6.15, observa-se que para esta condição inicial os algoritmos 1, 2 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 4, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no quinto ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.22, 6.23 e 6.24 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	138	136	136	178	122
Caso ii)	0	147	145	145	187	131
Caso iii)	0	119	117	117	159	103

Tabela 6.25 – Tempos relativos do exemplo 5.

Observa-se da tabela 6.25 que para este exemplo, o algoritmo 4 é o que apresenta os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram valores menores em relação a solução de referência atrasada. Os algoritmos 1 e 2 apresentam pequenas melhoras em relação aos resultados da solução de regime atrasada. O algoritmo 3 é o que apresenta o pior resultado global, sendo seus resultados sempre acima da solução de regime atrasada.

Exemplo 6

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)	
			Duração das Tarefas nas Máquinas	
1	2	1 2	8	10
			2	3
2	3	1 2 3	5	8 10
			3	2 3
3	3	1 2 4	4	5 12
			1	2 3
4	3	1 3 4	0	7 11
			4	3 1
5	2	3 4	3	7
			4	4

Tabela 6.26 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 10$, e $t^{\#} = [8 \ 5 \ 4 \ 0 \ 3]$

i) - Para a condição inicial $t_0 = [0 \ 5 \ 4 \ 10 \ 3]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	10	17	13	13	13	13
Ciclo 3	20	27	26	26	26	26
Ciclo 4	30	37	36	36	36	36
Ciclo 5	40	47	46	46	46	46
Ciclo 6	50	57	56	56	56	56
Ciclo 7	60	67	66	66	66	66
Ciclo 8	70	77	76	76	76	76
Ciclo 9	80	87	86	86	86	86
Ciclo 10	90	97	96	96	96	96

Tabela 6.27 – Instantes de início do primeiro job de cada sequência.

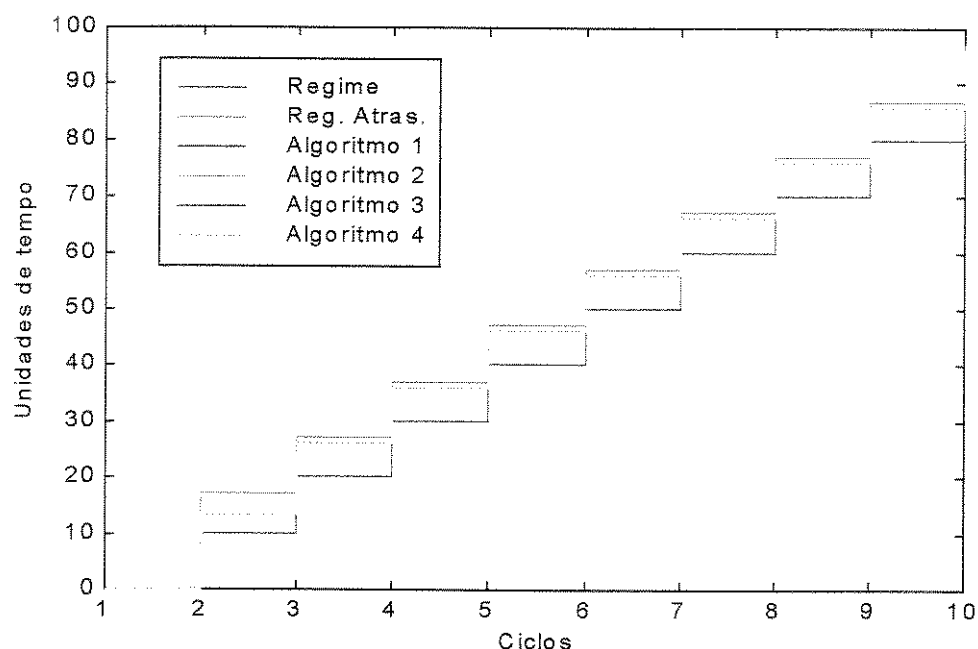


Figura 6.16 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.27 e a figura 6.16, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso, com os mesmos valores.

ii) – Para a condição inicial $t_0 = [8 \ 5 \ 0 \ 12 \ 5]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	10	19	13	13	13	13
Ciclo 3	20	29	28	28	28	28
Ciclo 4	30	39	38	38	38	38
Ciclo 5	40	49	48	48	48	48
Ciclo 6	50	59	58	58	58	58
Ciclo 7	60	69	68	68	68	68
Ciclo 8	70	79	78	78	78	78
Ciclo 9	80	89	88	88	88	88
Ciclo 10	90	99	98	98	98	98

Tabela 6.28 – Instantes de início do primeiro job de cada sequência.

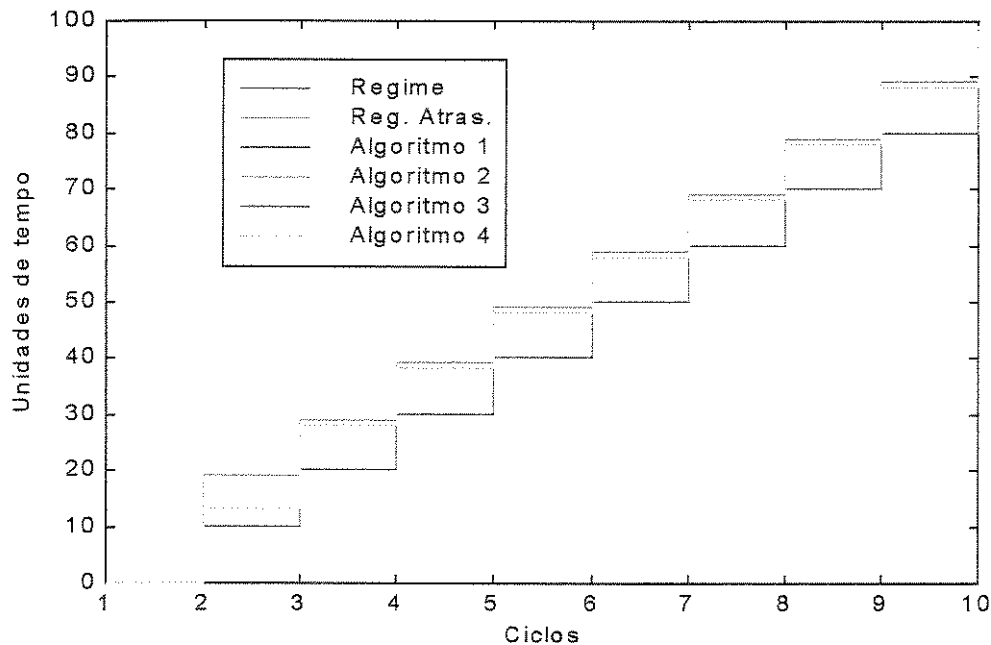


Figura 6.17 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.28 e a figura 6.17, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

iii) – Para a condição inicial $t_0 = [6 \ 0 \ 4 \ 8 \ 1]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	10	15	10	10	10	10
Ciclo 3	20	25	24	24	21	24
Ciclo 4	30	35	34	34	35	34
Ciclo 5	40	45	44	44	45	44
Ciclo 6	50	55	54	54	55	54
Ciclo 7	60	65	64	64	65	64
Ciclo 8	70	75	74	74	75	74
Ciclo 9	80	85	84	84	85	84
Ciclo 10	90	95	94	94	95	94

Tabela 6.29 – Instantes de início do primeiro job de cada sequência.

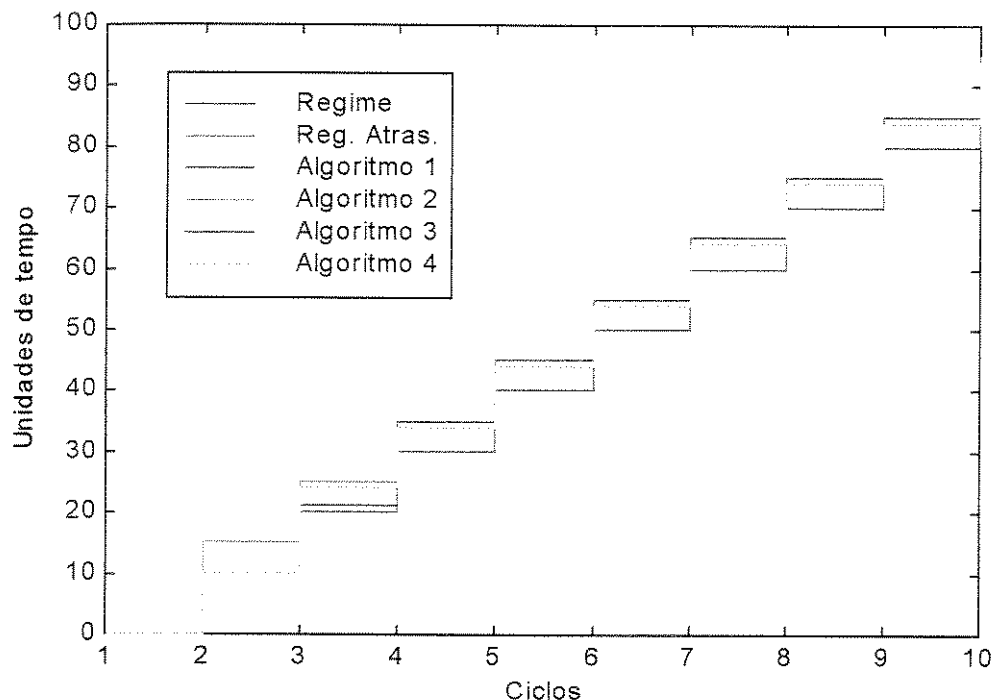


Figura 6.18 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.29 e a figura 6.18, observa-se que para esta condição inicial os algoritmos 1, 2 e 4 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor igual a solução de regime atrasada, portanto não caracterizando melhora na produção. Observa-se que a convergência das soluções obtidas com os algoritmos, ocorrem no terceiro ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.27, 6.28 e 6.29 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	7	6	6	6	6
Caso ii)	0	9	8	8	8	8
Caso iii)	0	5	4	4	5	4

Tabela 6.30 – Tempos relativos do exemplo 6.

Observa-se da tabela 6.30 que para este exemplo, a única diferença que ocorre entre os algoritmos é caso iii) com o algoritmo 3 apresentando resultado igual a solução de regime atrasada. Portanto o algoritmo 3 é o que apresenta o pior resultado global.

Exemplo 7

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)					
			Duração das Tarefas nas Máquinas					
1	6	2 1 5 2 3 4	54	67	72	80	97	111
			13	5	8	15	12	3
2	5	1 3 4 2 3	35	40	91	95	109	
			5	11	4	14	15	
3	7	1 5 4 3 1 3 2	30	35	48	51	65	109
			5	7	3	14	2	13 14
4	6	4 2 5 2 1 5	33	39	57	67	100	105
			5	15	10	13	5	6
5	5	3 1 4 1 4	0	21	22	27	30	
			13	1	5	3	3	

Tabela 6.31 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 84$, e $t^{\#} = [54 \ 35 \ 30 \ 33 \ 0]$

i) Para a condição inicial $t_0 = [93 \ 45 \ 0 \ 0 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	84	148	97	97	97	97
Ciclo 3	168	232	221	221	221	241
Ciclo 4	252	316	305	305	305	373
Ciclo 5	336	400	389	389	389	555
Ciclo 6	420	484	473	473	473	708
Ciclo 7	504	568	557	557	557	825
Ciclo 8	588	652	641	641	641	1007
Ciclo 9	672	736	725	725	725	1160
Ciclo 10	756	820	809	809	809	1277

Tabela 6.32 – Instantes de início do primeiro job de cada sequência.

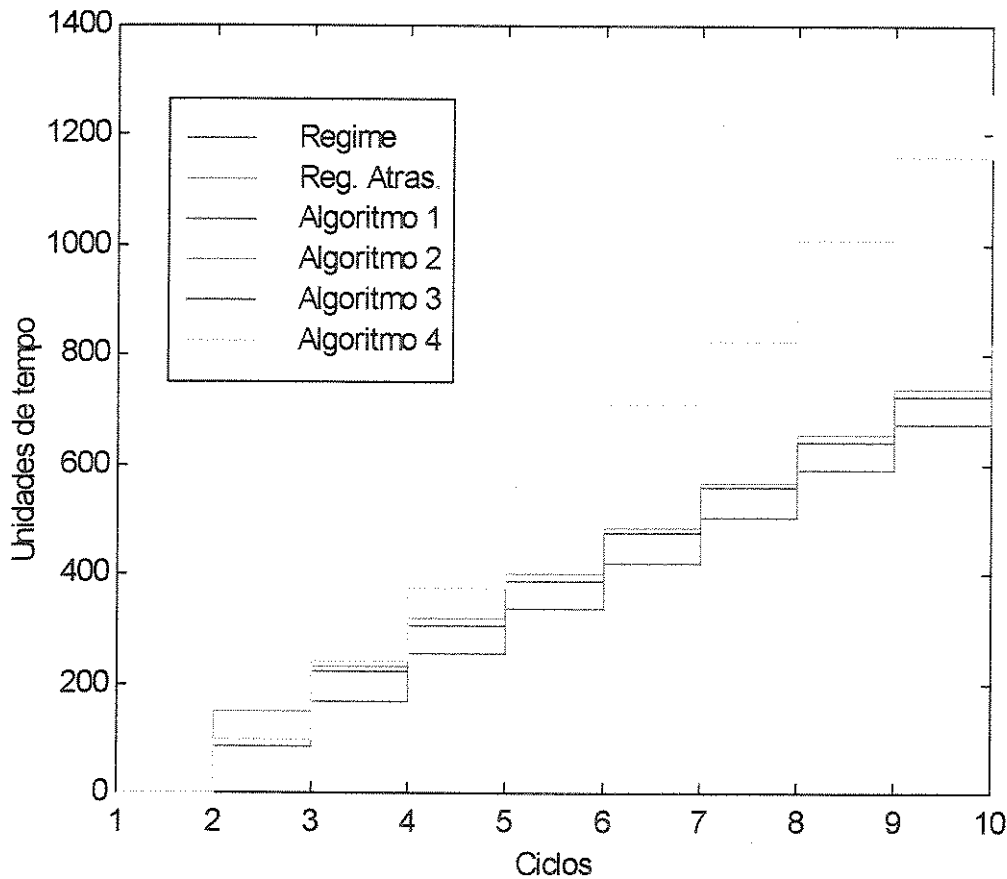


Figura 6.19 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.32 e a figura 6.19, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores superiores à solução de regime, porém apresentam valores menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 3 ocorrem no terceiro ciclo, sendo elas iguais nos três algoritmos.

ii) – Para a condição inicial $t_0 = [24 \ 5 \ 0 \ 3 \ 49]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	84	94	90	94	90	79
Ciclo 3	168	178	184	157	184	243
Ciclo 4	252	262	268	241	268	370
Ciclo 5	336	346	352	325	352	487
Ciclo 6	420	430	436	409	436	669
Ciclo 7	504	514	520	493	520	822
Ciclo 8	588	598	604	577	604	939
Ciclo 9	672	682	688	661	688	1121
Ciclo 10	756	766	772	745	772	1274

Tabela 6.33 – Instantes de início do primeiro job de cada sequência.

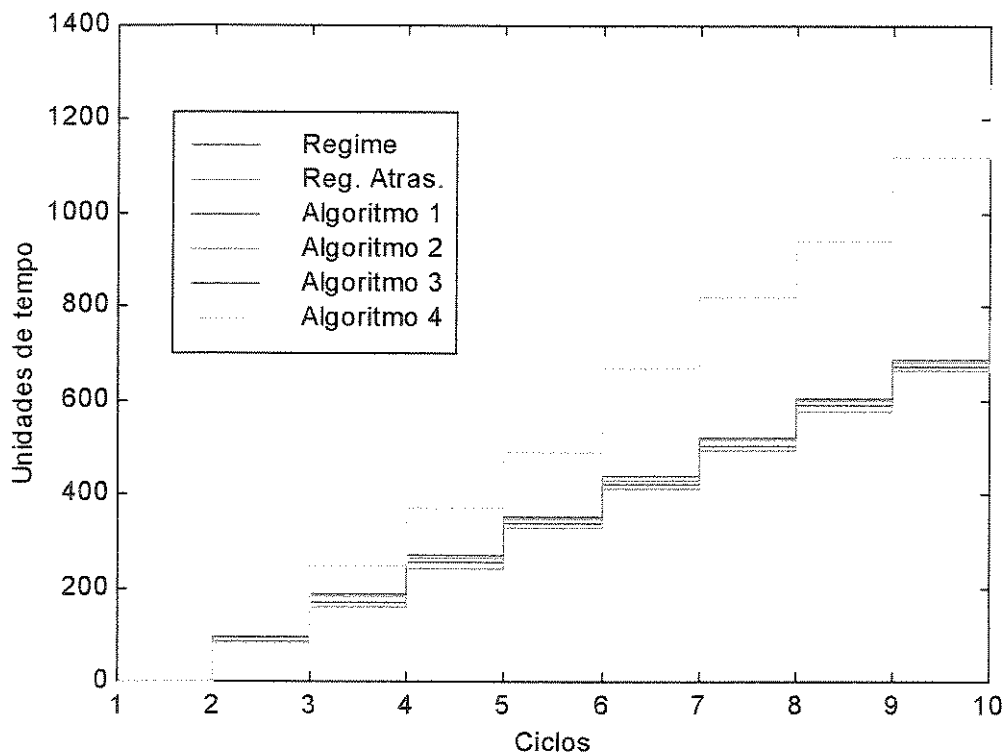


Figura 6.20 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.33 e a figura 6.20, observa-se que para esta condição inicial o algoritmo 2 tem sua solução convergindo para um valores inferior à solução de regime. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já os algoritmos 1 e 3 tem suas soluções convergidas para valores superiores a

solução de regime atrasada, portanto não caracterizando melhora na produção. Já o algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 3 ocorrem no terceiro ciclo.

iii) – Para a condição inicial $t_0 = [0 \ 27 \ 34 \ 40 \ 85]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	84	116	116	116	100	100
Ciclo 3	168	200	179	179	234	266
Ciclo 4	252	284	263	263	361	383
Ciclo 5	336	368	347	347	488	565
Ciclo 6	420	452	431	431	612	718
Ciclo 7	504	536	515	515	696	835
Ciclo 8	588	620	599	599	780	1017
Ciclo 9	672	704	683	683	864	1170
Ciclo 10	756	788	767	767	948	1287

Tabela 6.34 – Instantes de início do primeiro job de cada sequência.

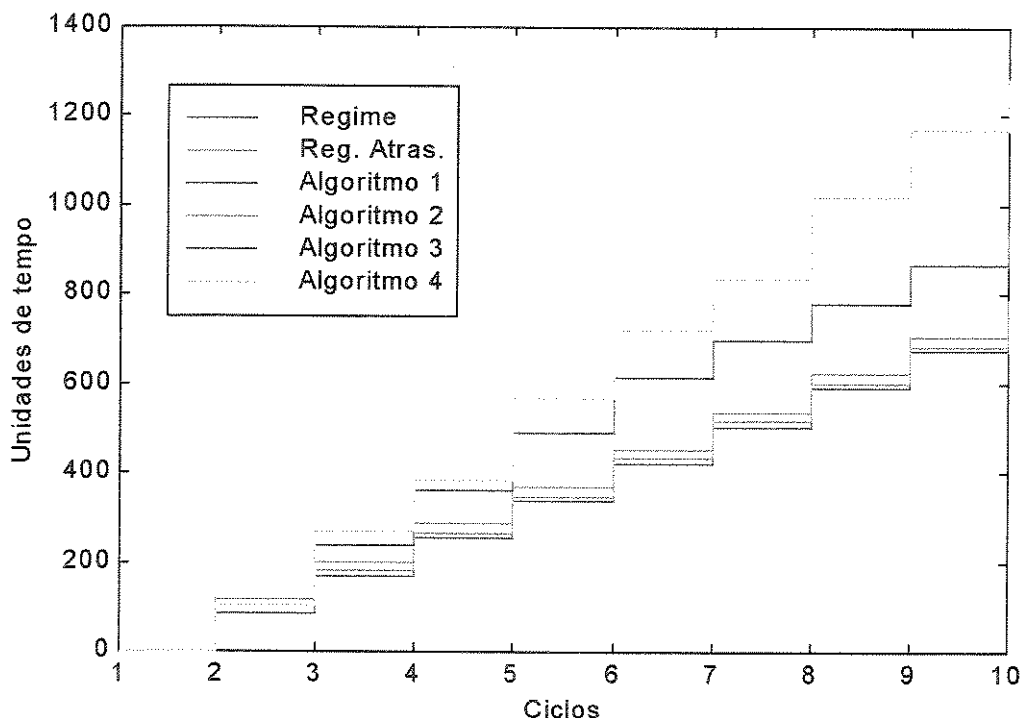


Figura 6.21 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.34 e a figura 6.21, observa-se que para esta condição inicial os algoritmos 1 e 2 tiveram suas soluções convergindo para valores superiores à solução de regime, porém apresentam valores menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 3 tem a sua solução convergida para um valor superior a solução de regime atrasada, portanto não caracterizando melhora na produção. Já o algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo, já com o algoritmo 3 ocorre no sexto ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.32, 6.33 e 6.34 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	64	53	53	53	521
Caso ii)	0	10	16	-11	16	518
Caso iii)	0	32	11	11	192	531

Tabela 6.35 – Tempos relativos do exemplo 7.

Observa-se da tabela 6.35 que para este exemplo, o algoritmo 2 é o que apresenta os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram menores valores. O algoritmo 1 tem resultados piores do que algoritmo 2, mas resultados melhores do que o algoritmo 3 e 4, sendo que apenas no caso ii) apresenta um atraso em relação a solução de referência atrasada. O algoritmo 3 tem apenas no caso i) um resultado melhor do que a solução de regime atrasada, nos demais casos apresenta atrasos maiores do que ela. O algoritmo 4 é o que apresenta o pior resultado global, pois para todos os casos até o décimo ciclo não havia ainda sua solução convergida para a solução de referência.

Exemplo 8

Seja o problema de job shop cíclico como o apresentado na tabela a seguir.

Número do Job	Número de Tarefas	Roteiro de Fabricação (máquinas)	Instante de Início das Tarefas (regime)						
			Duração das Tarefas nas Máquinas						
1	7	1 2 1 5 2 3 4	0	13	102	110	163	187	219
			1	4	1	7	4	1	1
2	7	5 1 5 2 1 5 4	25	78	93	158	166	220	244
			8	3	9	3	1	8	3
3	6	1 4 2 3 4 1	7	16	19	95	96	167	
			5	3	4	1	3	4	
4	5	2 5 2 5 3	3	74	109	127	173		
			1	7	4	8	1		
5	8	2 3 4 1 2 1 2 1	7	34	35	94	100	176	180 184
			2	1	2	3	3	3	4 2
6	9	1 5 2 4 5 4 2 5 3	4	87	94	114	135	143	148 194 204
			3	6	2	2	8	4	5 10 1
7	6	2 5 2 3 1 4	30	81	153	205	251	253	
			2	6	4	1	2	4	

Tabela 6.36 – Roteiro de fabricação, duração e instante de início das tarefas.

Para este problema tem-se os seguintes dados: $\lambda^{\#} = 77$, e $t^{\#} = [0 \ 25 \ 7 \ 3 \ 7 \ 4 \ 30]$

i) - Para a condição inicial $t_0 = [0 \ 150 \ 7 \ 3 \ 135 \ 4 \ 125]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	77	310	155	154	153	154
Ciclo 3	154	387	371	375	255	245
Ciclo 4	231	464	448	452	453	358
Ciclo 5	308	541	525	529	530	459
Ciclo 6	385	618	602	606	607	632
Ciclo 7	462	695	679	683	684	724
Ciclo 8	539	772	756	760	761	832
Ciclo 9	616	849	833	837	838	967
Ciclo 10	693	926	910	914	915	1108

Tabela 6.37 – Instantes de início do primeiro job de cada sequência.

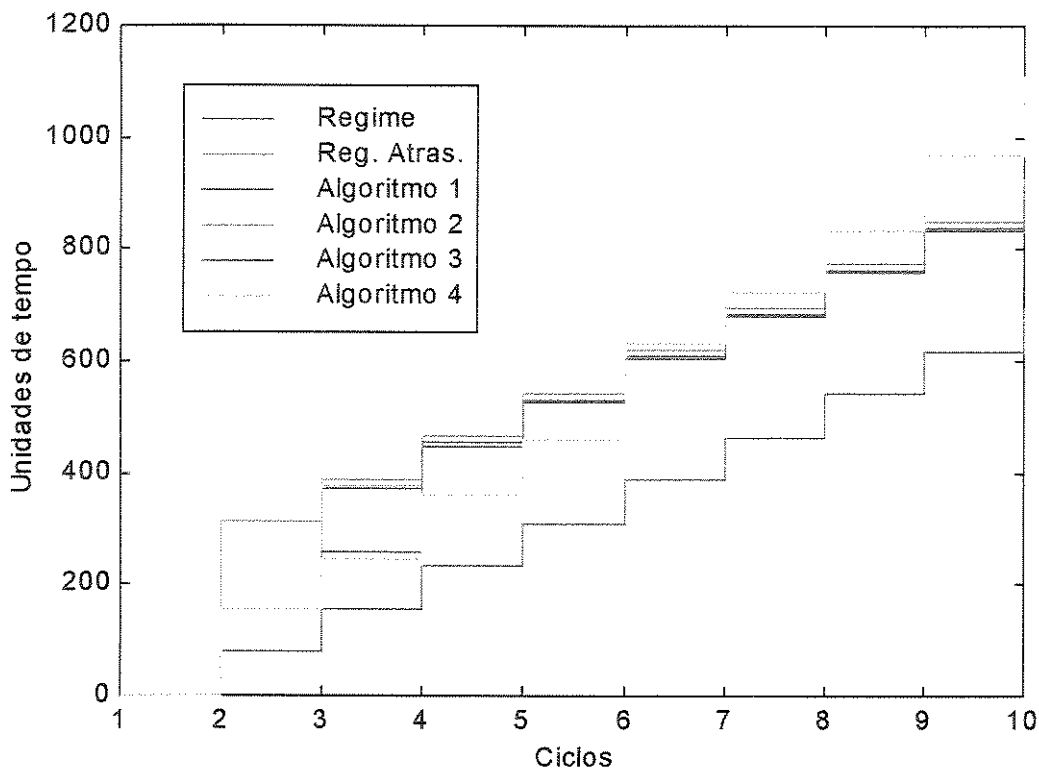


Figura 6.22 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.37 e a figura 6.22, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já o algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo, enquanto a solução obtida com o algoritmo 3 converge no quarto ciclo.

ii) - Para a condição inicial $t_0 = [10 \ 160 \ 0 \ 14 \ 7 \ 18 \ 130]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	77	289	167	167	167	186
Ciclo 3	154	366	282	282	257	278
Ciclo 4	231	443	356	356	376	372
Ciclo 5	308	520	433	433	473	423
Ciclo 6	385	597	510	510	578	524
Ciclo 7	462	674	587	587	651	700
Ciclo 8	539	751	664	664	792	810
Ciclo 9	616	828	741	741	950	930
Ciclo 10	693	905	818	818	1132	1091

Tabela 6.38 – Instantes de início do primeiro job de cada sequência.

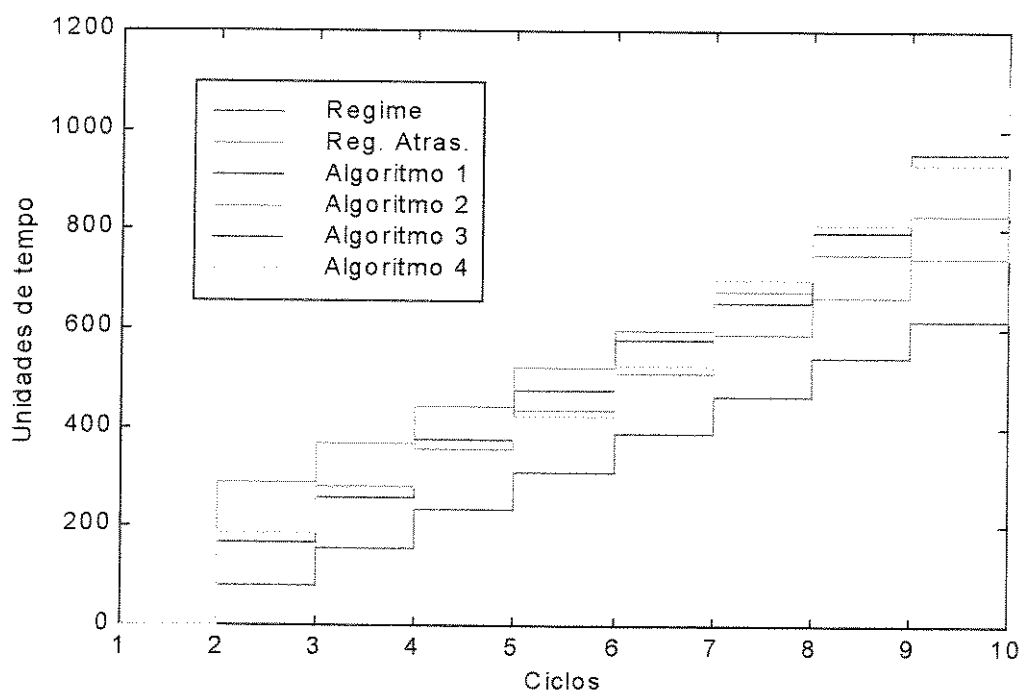


Figura 6.23 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.38 e a figura 6.23, observa-se que para esta condição inicial os algoritmos 1 e 2 tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Já os algoritmos 3 e 4 até o décimo ciclo ainda não tinham suas soluções convergidas. Estas convergências ocorrerão

nos ciclos subsequentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1 e 2, ocorrem no terceiro ciclo.

iii) – Para a condição inicial $t_0 = [0 \ 22 \ 25 \ 0 \ 4 \ 1 \ 48]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	77	200	200	200	200	140
Ciclo 3	154	277	277	277	277	212
Ciclo 4	231	354	354	354	354	342
Ciclo 5	308	431	431	431	431	471
Ciclo 6	385	508	508	508	508	600
Ciclo 7	462	585	585	585	585	653
Ciclo 8	539	662	662	662	662	728
Ciclo 9	616	739	739	739	739	829
Ciclo 10	693	816	816	816	816	992

Tabela 6.39 – Instantes de início do primeiro job de cada sequência.

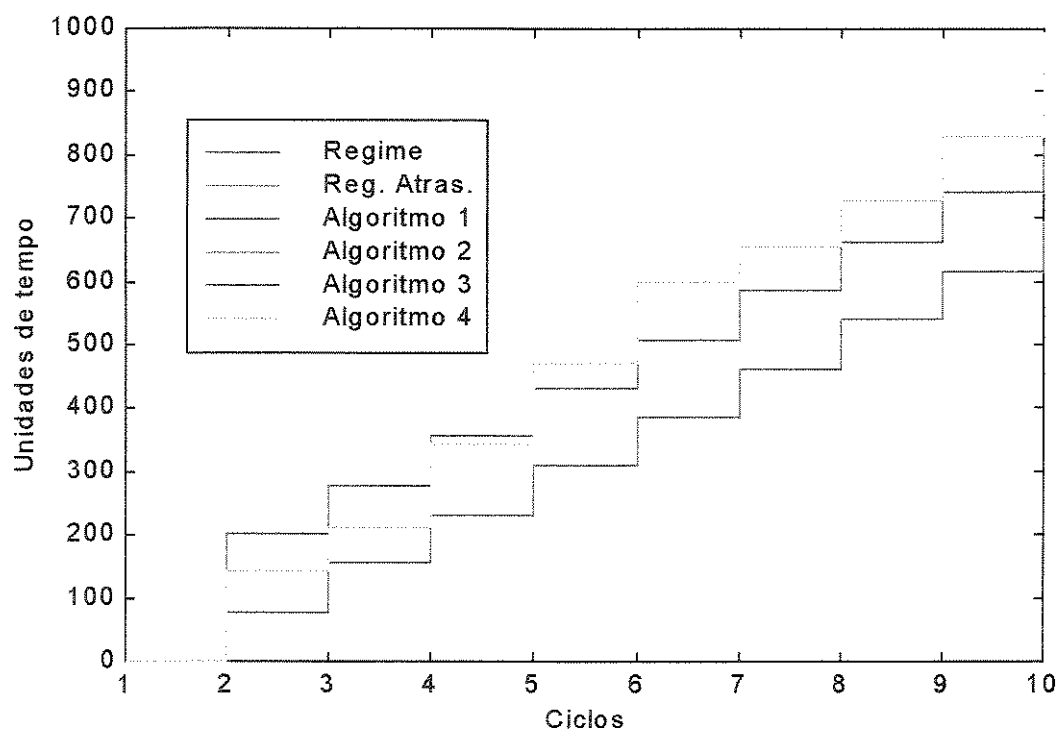


Figura 6.24 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.39 e a figura 6.24, observa-se que para esta condição inicial os algoritmos 1, 2 e 3 tiveram suas soluções convergindo para valores iguais à solução de regime atrasado. Isto significa que os resultados obtidos se igualam ao pior caso, considerado neste trabalho. Já o algoritmo 4 até o décimo ciclo ainda não tinha sua solução convergida. Esta convergência ocorrerá nos ciclos subseqüentes. Observa-se que a convergência das soluções obtidas com os algoritmos 1, 2 e 3 ocorrem no segundo ciclo.

A tabela que será apresentada a seguir, é simplesmente a última linha das tabelas 6.37, 6.38 e 6.39 com todas colunas subtraídas do valor de regime no ciclo 10.

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Caso i)	0	233	217	221	222	415
Caso ii)	0	212	125	125	439	398
Caso iii)	0	123	123	123	123	299

Tabela 6.40 – Tempos relativos do exemplo 8.

Observa-se da tabela 6.40 que para este exemplo, o algoritmo 1 é o que apresenta os melhores resultados globais. Pois, sempre suas soluções convergiram para a solução de regime e seus atrasos apresentaram menores valores. O algoritmo 2 tem resultados semelhantes ao do algoritmo 1, perdendo somente no caso i). O algoritmo 3 tem apenas no caso i) um resultado melhor do que a solução de regime atrasada, nos demais casos apresenta atrasos maiores do que ela. O algoritmo 4 é o que apresenta o pior resultado global, pois para todos os casos até o décimo ciclo não havia ainda sua solução convergida para a solução de referência.

6.4 – Eliminação das Esperas Fixas

Conforme observado na introdução do capítulo 5, o desenvolvimento formal apresentado neste trabalho trata os jobs como sendo unidades indivisíveis, isto é, os tempos de espera entre tarefas de um mesmo job são considerados fixos e iguais àqueles calculados para a solução de regime. Contudo, conforme também observado na introdução do capítulo 5, é

possível na formulação proposta neste trabalho, tratar as tarefas de maneira independente eliminando a rigidez de esperas entre tarefas de um mesmo job, mas aumentando a dimensão do problema.

Em qualquer problema que envolve modelos matemáticos há um certo compromisso entre a precisão de resposta e a complexidade deste modelo. O modelo que passa agora a ser utilizado apresenta um grau de detalhamento superior ao modelo que até agora vinha sendo utilizado. Em consequência disto, espera-se melhoria nas soluções dos problemas. Em contrapartida maiores esforços computacionais serão empregados na obtenção destas soluções.

Na obtenção das soluções para o problema de job shop cíclico utilizando-se a estratégia com esperas fixas, a matriz M e a matriz M^* possuem dimensão igual ao número de jobs. Na estratégia que agora passa a ser utilizada a dimensão destas matrizes passam a ser iguais à soma de todas as tarefas de todos os jobs.

A seguir será mostrado como obter os elementos da nova matriz M , de acordo com a relação entre as operações.

- i) Se t_j for tarefa inicial de algum job e ambas tarefa t_j e t_k utilizam a mesma máquina, então os elementos m_{jk} da matriz M são:

$$p_{tjtk} = \begin{cases} p_{tj} & \text{if } k \leq K_{tj} \\ 0 & \text{if } k > K_{tj} \end{cases}$$

- ii) Se t_j for tarefa inicial de algum job, porém as tarefa t_j e t_k não utilizam a mesma máquina, então os elementos m_{ik} da matriz M são:

$$p_{tjtk} = \begin{cases} 1 & t \leq 0 \\ 0 & t > 0 \end{cases}$$

- iii) Se as tarefas t_j e t_k utilizam a mesma máquina e além disso t_j não é tarefa inicial de nenhum job, então os elementos m_{jk} da matriz M são:

$$p_{tjtk} = \begin{array}{c} \text{---} \\ \lambda^{\#} \quad (p_{tk}-1) p_{ti} \end{array}$$

- iv) Se as tarefas t_j e t_k pertencem ao mesmo job e t_j antecede t_k (não necessariamente imediatamente), então os elementos m_{jk} da matriz M são:

$$p_{t_j t_k} = \overline{\hspace{1.5cm}} \begin{array}{c} \text{---} \end{array} s_{jk}$$

onde s_{jk} é a somatória das durações das tarefas do job i entre t_j e t_k , inclusive a duração de t_j .

- v) Se nenhuma das situações descritas anteriormente entre t_j e t_k ocorrem, então os elementos m_{jk} da matriz M são iguais ao elemento nulo (zero).

Nos exemplos a seguir a estratégia proposta nos parágrafos anteriores é utilizada para que se possa analisar seu desempenho.

Exemplo 9

Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.1 (exemplo 1).

Conforme visto no exemplo 1, tem-se os seguintes dados: $\lambda^\# = 98$ e $t^\# = [72 \ 66 \ 0]$.

Para a condição inicial $t_0 = [40 \ 85 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	98	215	85	85	85	85
Ciclo 3	196	313	237	237	237	237
Ciclo 4	294	411	335	335	335	335
Ciclo 5	392	509	433	433	433	433

Tabela 6.41 – Instantes de início do primeiro job de cada seqüência.

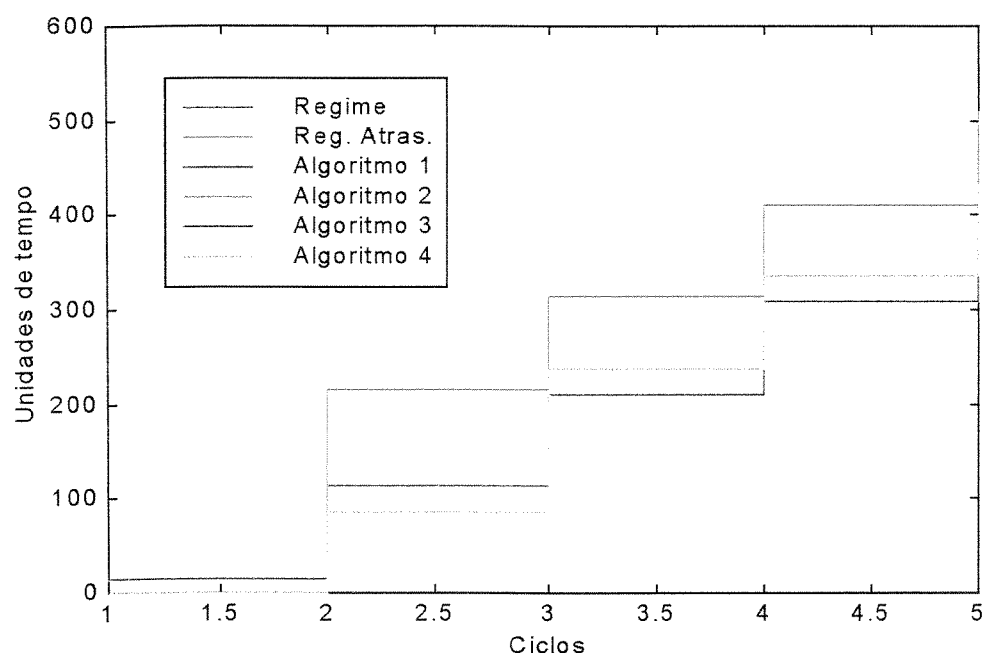


Figura 6.25 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.41 e a figura 6.25, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

Exemplo 10

Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.6 (exemplo 2).

Conforme visto no exemplo 2, tem-se os seguintes dados: $\lambda^{\#} = 76$ e $t^{\#} = [0 \ 26 \ 67]$.

Para a condição inicial $t_0 = [43 \ 0 \ 33]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	76	183	54	54	54	54
Ciclo 3	152	259	255	255	255	255
Ciclo 4	228	335	331	331	331	331
Ciclo 5	304	411	407	407	407	407

Tabela 6.42 – Instantes de início do primeiro job de cada seqüência.

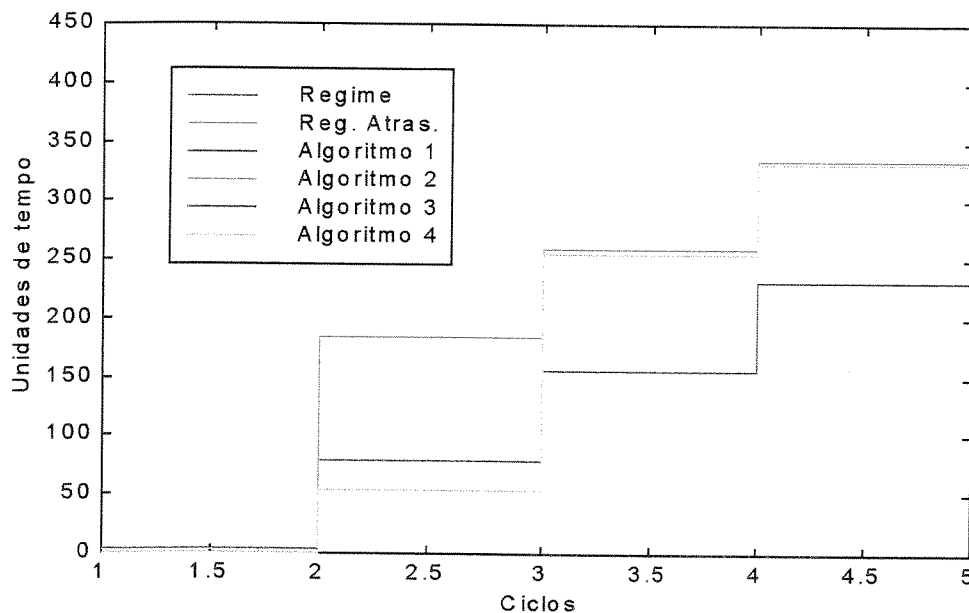


Figura 6.26 – Instantes de início do primeiro job de cada seqüência.

Analisando a tabela 6.42 e a figura 6.26, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

Exemplo 11

Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.11 (exemplo 3).

Conforme visto no exemplo 3, tem-se os seguintes dados: $\lambda^\# = 51$ e $t^\# = [0 \ 19 \ 11]$.

Para a condição inicial $t_0 = [27 \ 26 \ 0]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	51	94	40	40	40	40
Ciclo 3	102	145	133	133	133	133
Ciclo 4	153	196	184	184	184	184
Ciclo 5	204	247	235	235	235	235

Tabela 6.43 – Instantes de início do primeiro job de cada sequência.

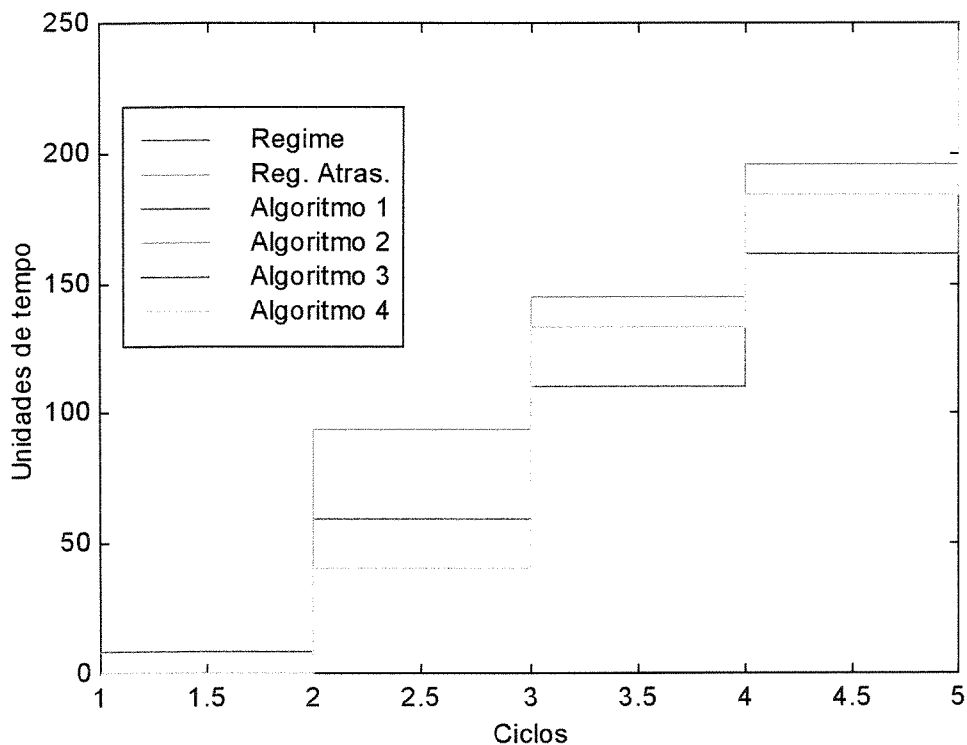


Figura 6.27 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.43 e a figura 6.27, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores superiores à solução de regime, porém menores que a solução de regime atrasada. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a

convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

Exemplo 12

Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.16 (exemplo 4).

Conforme visto no exemplo 4, tem-se os seguintes dados: $\lambda^{\#} = 66$ e $t^{\#} = [65 \ 0 \ 33]$.

Para a condição inicial $t_0 = [69 \ 0 \ 23]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	66	159	69	69	69	69
Ciclo 3	132	225	122	122	122	122
Ciclo 4	198	291	188	188	188	188
Ciclo 5	264	357	254	254	254	254

Tabela 6.44 – Instantes de início do primeiro job de cada sequência.

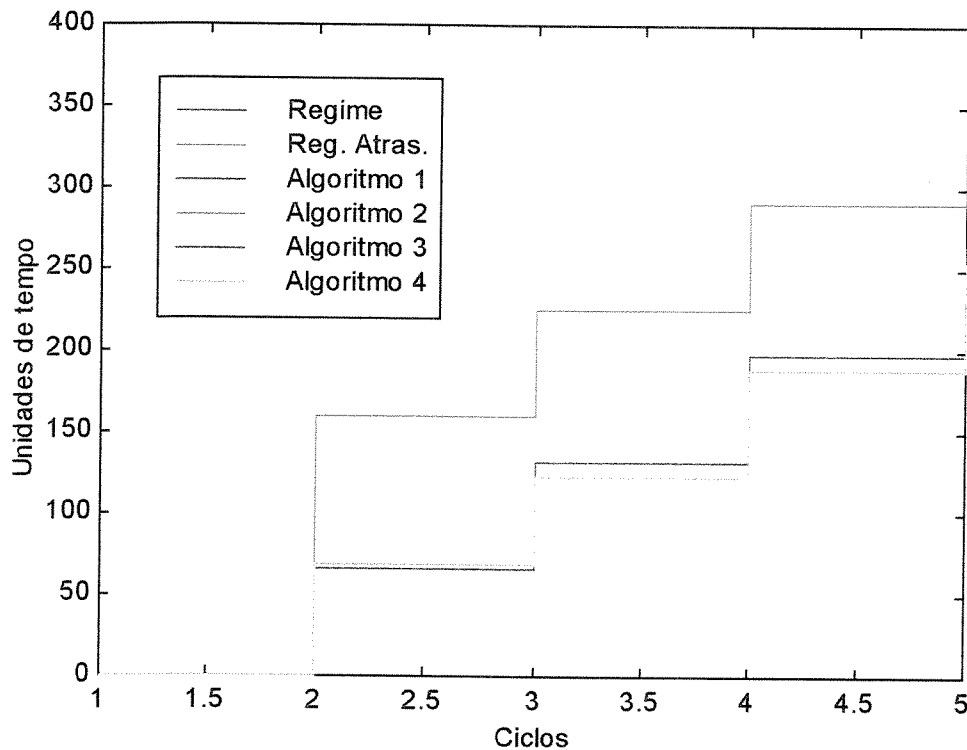


Figura 6.28 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.44 e a figura 6.28, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores menores do que à solução de regime. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

Exemplo 13.

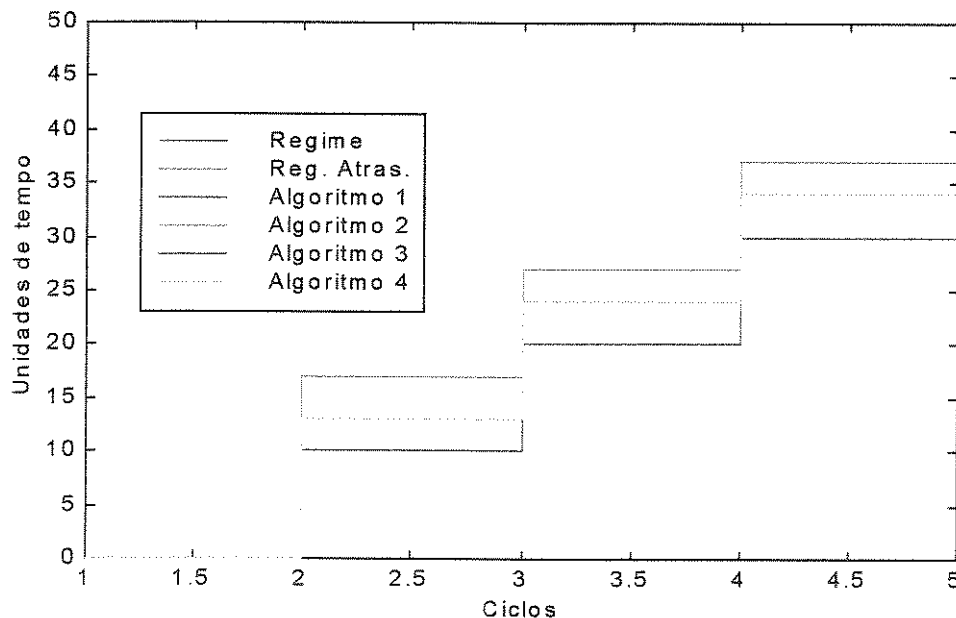
Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.26 (exemplo 6).

Conforme visto no exemplo 6, tem-se os seguintes dados: $\lambda^{\#} = 10$ e $t^{\#} = [8 \ 5 \ 4 \ 0 \ 3]$.

Para a condição inicial $t_0 = [0 \ 5 \ 4 \ 10 \ 3]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	10	17	13	13	13	13
Ciclo 3	20	27	24	24	24	24
Ciclo 4	30	37	34	34	34	34
Ciclo 5	40	47	44	44	44	44

Tabela 6.45 – Instantes de início do primeiro job de cada sequência.



UNICAMP
 BIBLIOTECA CENTRAL
 SEÇÃO CIRCULANTE

Figura 6.29 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.45 e a figura 6.29, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores menores do que à solução de regime. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a sequência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

Exemplo 14.

Os dados referentes ao problema de job shop cíclico para este exemplo são os que se encontram na tabela 6.31 (exemplo 7).

Conforme visto no exemplo 7, tem-se os seguintes dados: $\lambda^{\#} = 84$ e $t^{\#} = [54 \ 35 \ 30 \ 33 \ 0]$.

Para a condição inicial $t_0 = [0 \ 27 \ 34 \ 40 \ 85]$, e utilizando a lei de controle obtém-se:

	Regime	Reg. Atras.	Algoritmo 1	Algoritmo 2	Algoritmo 3	Algoritmo 4
Ciclo 1	0	0	0	0	0	0
Ciclo 2	84	116	85	85	85	85
Ciclo 3	168	200	194	194	194	194
Ciclo 4	252	284	278	278	278	278
Ciclo 5	336	368	362	362	362	362

Tabela 6.46 – Instantes de início do primeiro job de cada sequência.

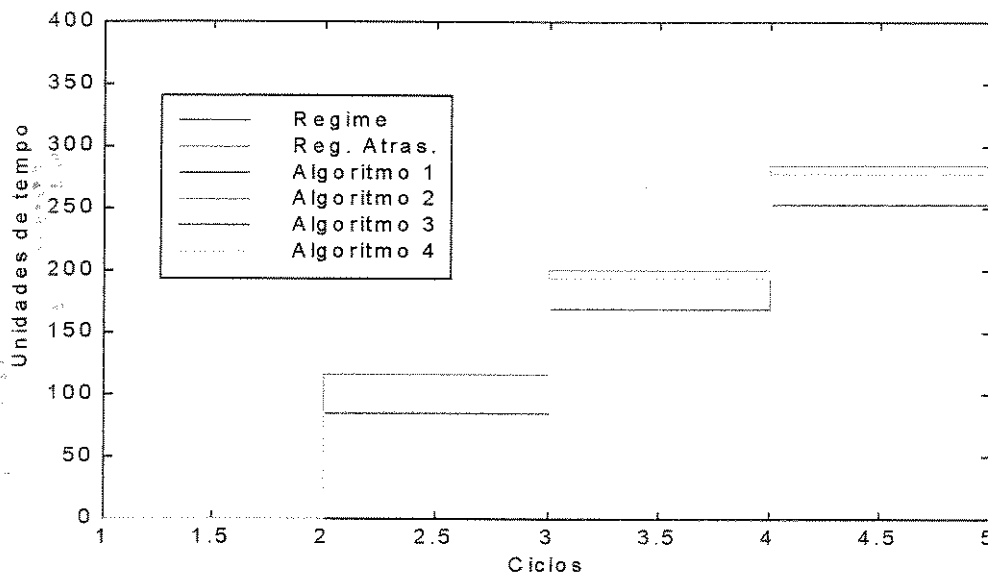


Figura 6.30 – Instantes de início do primeiro job de cada sequência.

Analisando a tabela 6.46 e a figura 6.30, observa-se que para esta condição inicial todos os algoritmos tiveram suas soluções convergindo para valores menores do que à solução de regime. Evidenciando um ganho na produção e a volta dos disparos dos jobs para a seqüência de referência. Observa-se que a convergência das soluções obtidas com os algoritmos ocorrem no terceiro ciclo, além disso com os mesmos valores.

6.5 – Análise de Resultados

Observando-se os resultados obtidos nos exemplos da seção 6.3 e 6.4, fica claro que a utilização de metodologia proposta neste trabalho, garante um desempenho melhor do sistema comparado com a solução de regime atrasada no tempo (deslocada no tempo).

No vários exemplos da seção 6.3, foi sempre possível com pelo menos um dos algoritmos conseguir resultados com economia de unidades de tempo com relação à solução de regime atrasada. Numa análise global o algoritmo de síntese 1 foi o que melhores resultados obteve, pois somente em um caso dos vários exemplos teve resultado acima do que a solução de regime atrasada. Já o algoritmo de síntese 4 foi o pior globalmente, pois em muitos casos sua solução ainda não havia convergido e seus valores já eram superiores a solução de regime atrasada, com isto sua convergência só pode ocorrer para valores iguais ou maiores do que até então apresentados. O algoritmo de síntese 2 também apresentou um desempenho global bom, ficando atrás do algoritmo de síntese 1, por ter mais vezes apresentado soluções cujos valores ficaram acima da solução de regime atrasada, porém sempre conseguiu resultados melhores do que os algoritmos de síntese 3 e 4.

Quanto aos exemplos da seção 6.4, que são repetições dos problemas da seção 6.3, foi sempre possível obter resultados melhores do que a solução de regime atrasada com todos os algoritmos. Observa-se que retirando a restrição de esperas fixas entre tarefas de um mesmo job, os resultados apresentados por todos os algoritmos foram iguais. Não foi alvo de uma análise mais profunda, mas a repetição dos valores nas soluções obtidas em todos os

algoritmos, sinaliza que a matriz M construída sem as esperas fixas apresenta seus elementos muito semelhante entre si, proporcionando aos algoritmos escolhas iguais nas decisões.

Com estes exemplos pode-se observar que a metodologia proposta sempre foi superior em relação a simplesmente atrasar (retardar) a solução de regime.

CAPÍTULO 7

CONCLUSÕES

O estudo e desenvolvimento de estratégias para atuarem em transitórios de problemas de escalonamento cíclico, tal qual o apresentado neste trabalho, vem tendo sua importância aumentada a cada dia, em função dos benefícios advindo da sua utilização.

Ressalta-se que a lei de controle proposta neste trabalho garante a convergência das soluções para a solução de regime, desde que as restrições impostas no capítulo 5 sejam satisfeitas. Nos resultados de simulação do capítulo 6, pode ser observado que a maior parte dos desempenhos obtidos com a metodologia proposta são melhores que o simples deslocamento da solução de regime até que ela possa novamente ser utilizada. Há entretanto algoritmos cujos resultados são piores do que a solução de regime deslocada no tempo. Isto ocorre porque as heurísticas utilizadas são “miopes”, isto é não são capazes de determinar se uma certa alternativa, boa no momento da decisão, pode vir a levar o sistema a uma pior situação no futuro. Deve-se ressaltar, contudo, que para as simulações geradas aleatoriamente no capítulo 6, a incidência destes casos foi pequena.

Os resultados obtidos nas simulações foram viabilizados pelos resultados teóricos apresentados no capítulo 5. Os resultados apresentados por Noronha (1999) já apontavam para a possibilidade de tratar os transitórios em job shop cíclicos com garantia de estabilidade em relação a uma solução de referência conhecida. Contudo as dificuldades em obter as matrizes de decisão referidas naquele trabalho levaram à proposição de alternativas para a formulação da lei de controle. Nesse trabalho a lei de controle proposta baseia-se na satisfação de restrições de estabilidade (equação 5.2) associadas a restrições de factibilidade (equação 5.3), que a tornam computacionalmente menos custosa. O resultado teórico que fundamenta esta

abordagem é resumido pelo teorema 5.3. A análise de factibilidade, determinada pela equação (5.3), embora não originalmente proposta neste trabalho foi refinada ao ser expressa em termos de série de potências. Esse avanço em relação aos trabalhos precedentes, constitui um passo para a representação padronizada do problema segundo a literatura de teoria de dióides aplicada aos sistemas a eventos discretos.

Quanto aos resultados obtidos quando a espera entre tarefas de um mesmo job é fixa, e quando estas esperas estão livres desta restrição, observa-se que a segunda alternativa é mais eficaz globalmente, pois todos os algoritmos desta alternativa apresentam soluções melhores do que o caso da solução defasada. Há porém casos em que a solução com esperas fixas é melhor do que com esperas livres. Isto é compreensível devido ao fato de que com esperas livres as opções de escolha do próximo instante de início são muito maiores e nem todas elas são testadas. Caso todas as alternativas fossem testadas recairíamos num problema combinatório. É conveniente ressaltar que o custo computacional da alternativa que não considera as esperas fixas é muito mais elevado do que a outra alternativa, por isso resultados melhores.

Embora o objetivo principal deste trabalho tenha sido alcançado, ou seja, provar uma lei de controle que garante a convergência das soluções para a solução de regime, foram utilizadas heurísticas na síntese desta lei para a obtenção das soluções. Uma desvantagem destas heurísticas é que elas são míopes, visualizando apenas o próximo ciclo. Portanto, há um vasto campo de pesquisa para a determinação analítica, ou até mesmo através de meta-heurísticas que ampliem o horizonte de visualização dos ciclos subsequentes objetivando uma melhor solução para o problema. Outro caminho a ser explorado é a tentativa de se escrever as duas inequações num mesmo dióide e, então analisar o problema sob a ótica do controle ótimo.

Quanto à possibilidade deste problema vir a ser linear em função da utilização de novos dióides, é muito remota, pois os dióides que fornecem modelos lineares não possuem a característica de modelarem concorrência (Baccelli et al., 1992), uma característica muito forte deste problema.

Em síntese, com a metodologia proposta é possível tomar decisões em escalonamentos cíclicos, quando uma solução de regime (conhecida) é perturbada.

REFERÊNCIAS BIBLIOGRÁFICAS

- Baccelli, F. L., G. Cohen, G. J. Olsder and J. P. Quadrat (1992). Synchronization and Linearity: An Algebra for Discrete Event Systems, John Wiley & Sons.
- Baker, K. R. (1974). Introduction to Sequencing and Scheduling, John Wiley & Sons.
- Blackstone, J. H., D. T. Phillips and L. G. Hogg (1982). A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations. International Journal of Production Research, Vol. 20, pp. 27-45.
- Bowman, R. A. and J. A. Muckstand (1993). Stochastic Analysis of Cyclic Schedules. Operations Research, Vol. 41, n° 5, pp. 947-958.
- Cassandras, C. G. (1993). Discrete Event Systems: Modeling and Performance Analysis. Aksen Associates Incorporated Publishers.
- Cassandras, C. G. and S. G. Strickland (1989). Sample Path Properties of Timed Discrete Event Systems. Proceeding IEEE, Vol. 77, n° 1, pp. 59-71.
- Cofer, D. D. and V. K. Garg (1992). A Timed Model for the Control of Discrete Event Systems Involving Decisions in the Max/Plus Algebra. Proc. Of the 31st Conf. On Decision and Control, pp3363-3368.
- Cofer, D. D. and V. K. Garg (1993). A Generalized Max-Algebra Model for Performance Analysis of Timed and Untimed Discrete Event Systems. Proc. of the American Control Conference, San Francisco, pp. 2288-2292.
- Cohen, G., M. Pierre, J. P. Quadrat and M. Voit (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. Proceedings of the IEEE, Vol. 77, n° 1, pp. 39-58.
- Conway, R. W., L. Maxwell and L. W. Miller (1967). Theory of Scheduling. Addison-Wesley.
- Cuninghame-Green, R. (1979). Minimax Algebra. Springer-Verlag.
- Dauscha, W., H. D. Modrow and A. Meumann (1985). On Cyclic Sequence Types for Constructing Cyclic Schedules. Zeitschrift für Operations Research, 29, pp. 1-30.
- Gondran, M. and M. Minoux (1985). Graphes et Algorithmes, Eyrolles, Paris.

- Graves, S. C., H. C. Meal, D. Stefek and A. H. Zeghmi (1983). Scheduling of Re-Entrant Flow Shops. *Journal Operation Management*.
- Groth, A. F. and R. S. Mendes (1997). A Min-Max Model for Reative Job Shop Scheduling. 4th European Control Conference, Bruxelas.
- Gunawardena, J. (1996). An Introduction to Idempotency. HP Technical Report.
- Hanen, C. (1994). Study of a NP-hard Cyclic Scheduling Problem: The Recurrent Job-Shop. *European Journal of Operation Research*, Vol. 72, pp. 82-101.
- Hillion, H. P. and P. Jean-Marie (1989). Performance Evaluation of Job-Shop Systems Using Timed Event-Graphs. *IEEE Transactions on Automatic Control*, Vol. 34, n° 1, pp. 3-9.
- Hothaus, O. and H. Ziegler (1997). Improving Job-Shop Performance by Coordinating Dispatching Rules. *International Journal of Production Research*, Vol. 35, n° 2, pp. 539-549.
- Kim, Y. D. (1990). Dispatching Rules for Job Shops. *International Journal of Production Research*, Vol. 28, n° 5, pp. 963-980.
- Lee, T. E. and M. E. Posner (1997). Performance Measures and Schedules in Periodic Job Shops. *Operations Research*, Vol. 45, n° 1, pp. 72-91.
- Lee, T. E. (1994). Stable Earliest Schedules for Periodic Job Shops: A Linear System Approach. In *Proceedings of the 11th International Conference on Analysis and Optmization of systems*. Lecture Notes in Control and Information Science, 199, Spring Verlag, pp. 290-296.
- Levner, E. and V. Kats (1998). A Parametric Critical Path Problem and an Aplication for Cyclic Scheduling. *Discrete Applied Mathematics*, 87, pp. 149-158.
- Morton, T. E. and D. W. Pentico (1993). *Heuristic Scheduling Systems: with Applications to Production Systems and Project Management*. John Wiley & Sons.
- Munier, A. (1996). The Basic Cyclic Scheduling Problem with Linear Precedence Constraints. *Discrete Applied Mathematics*, 64, pp. 219-238.
- Nawijn, W. M. (1998). A Cyclic Prodution Problem: An Application of Max-Algebra. *European Journal of Operation Research*, Vol. 104, pp. 187-200.
- Noronha, A. B. (1999). *Controle de Transitórios em Escalonamento de Job Shop Cíclico*. Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp.

- 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.
- Rao, U. S. (1992). Multi-Stage, Identical Job Cyclic Scheduling for Repetitive Manufacturing. Technical Report 1029, SORIE, Cornell University, Ithaca, NY, USA.
- Round, R. (1992). Cyclic Scheduling for Job Shops with Identical Jobs. *Mathematics of Operations Research*, Vol. 17, nº 4, pp. 842-865.
- Schutter, B. D. and B. D. Moor (1997). On the Sequence of Consecutive Matrix Powers of Boolean Matrices in the Max-Plus Algebra. Technical Report, ESAT-SISTA/TR 97-67.
- Song, J. S. and T. E. Lee (1996). A Tabu Search Procedure for Periodic Job Shop Scheduling. *Computers Industrial Engineering*, Vol. 30, nº 3, pp. 433-447.
- Song, J. S. and T. E. Lee (1998). Petri Net Modeling and Scheduling for Cyclic Job Shops with Blocking. *Computers Industrial Engineering*, Vol. 34, nº 2, pp. 281-295.
- Tzafestas, S. and A. Triantafyllakis (1993). Deterministic Scheduling in Computing and Manufacturing Systems: a Survey of Models and Algorithms. *Mathematics and Computers in Simulation*, Vol. 35, pp. 397-434.
- Yura, K. (1998). Multi Job Cyclic Scheduling na Automated Job-Shop-Type Manufacturing System (Relationship Between Job Outup Interval and Buffer Capacity). *JSME International Journal*, Series III, Vol 32, nº 3, pp. 498-503.
- Zhang, H. and S. C. Graves (1997). Cyclic Scheduling in a Stochastic Enviroment. *Operations Research*, Vol. 45, nº 6, pp. 894-903.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE