



## **Comparação de Heurísticas para Scheduling utilizando Simulação por Eventos Discretos**

**Rodrigo Cardoso de Jesus**

Centro de Pesquisas Renato Archer  
Laboratório de Tecnologia de Gestão Empresarial  
Rod. D. Pedro I, Km 143,6  
Campinas SP 13083-970  
[Rodrigo.Jesus@cenpra.gov.br](mailto:Rodrigo.Jesus@cenpra.gov.br)

### **Resumo**

Neste trabalho procura-se estudar a minimização do tempo de ciclo de produção. Para isso utiliza-se heurísticas ligadas diretamente ao problema de scheduling em sistemas flow shop. Três são os métodos heurísticos explanados: Minimize Idle Time (MIT); Campbell, Dudek e Smith (CDS); e Palmer. É apresentado um exemplo com dez produtos e cinco máquinas para comparar o resultado dos métodos. Os três métodos heurísticos são implementados e são geradas seqüências dos produtos para o scheduling. Um modelo temporal determinístico do problema de scheduling é desenvolvido no ambiente de simulação A-SIM. A execução do modelo gera relatórios estatísticos que descrevem detalhadamente o comportamento do sistema produtivo permitindo a análise de cada um dos métodos.

***Palavras-chave:***



### Os Métodos Heurísticos

Os métodos heurísticos são utilizados neste trabalho para calcular o melhor scheduling que minimiza o tempo inativo de máquina, e assim minimizando o tempo total de processamento de um produto com  $n$  trabalhos em  $m$  máquinas, como representado abaixo:

*Tabela 1:*

*Representação Matricial de um Sistema Produtivo*

	M1	M2	M3	.	M <sub>m</sub>
j1	a <sub>11</sub>	a <sub>12</sub>	a <sub>13</sub>	.	a <sub>1m</sub>
j2	a <sub>21</sub>	a <sub>22</sub>	a <sub>23</sub>	.	a <sub>2m</sub>
j3	a <sub>31</sub>	a <sub>32</sub>	a <sub>33</sub>	.	a <sub>3m</sub>
.	.	.	.	.	.
j <sub>n</sub>	a <sub>n1</sub>	a <sub>n2</sub>	a <sub>n3</sub>	.	a <sub>nm</sub>

Na tabela acima cada linha representa um produto  $j_i$  que deve ser processado por um determinado número de máquinas  $m$ , aqui representadas pelas colunas, onde  $m_j$  é a  $j$ -ésima máquina do processo de produção. O cruzamento de uma linha com uma coluna, representada por  $a_{ij}$ , equívale ao tempo de processamento de um produto  $j_i$  na máquina  $m_j$ .

A tabela abaixo apresenta um exemplo onde se busca estabelecer o melhor scheduling com o menor tempo de máquina inativo. Este é um exemplo real de produção e será utilizado durante todo o trabalho para estudo das heurísticas.

O objetivo é estabelecer uma seqüência para cada heurística, calculando seu tempo de máquina inativo e o tempo total de fabricação. Sendo assim selecionada a melhor seqüência de cada método.

*Tabela 2:*

*Exemplo de tempos de um processo produtivo*

Produto	M1	M2	M3	M4	M5
1	5	12	15	4	13
2	10	8	6	9	4
3	13	17	5	8	10
4	21	13	18	14	17
5	15	6	9	4	13
6	14	10	3	7	22
7	12	5	6	9	14
8	13	7	4	8	3
9	10	14	13	20	5
10	6	19	14	13	18

Os métodos heurísticos aqui apresentados baseiam-se na regra de Johnson, assim sendo, propõe-se uma breve revisão e estudo do tema.

### A regra de Johnson

A proposta de Johnson para scheduling é relativamente simples, trata-se de posicionar inicialmente os produtos que apresentam menor tempo de processamento na primeira máquina (o que faz com que a máquina seguinte entre em processamento rapidamente) e posteriormente os produtos com menor tempo de processamento na segunda máquina (de forma que a primeira máquina fique o menor tempo possível ociosa, a partir do momento que o último produto passa por ela).

### Scheduling para Duas – Máquinas

A regra de Johnson é considerada ótima para resolução de scheduling em processos que apresentam  $n$  produtos passando por duas máquinas.



Primeiro Passo: Selecionar o grupo de produtos  $U$  com menor tempo de processamento na primeira máquina do que na segunda;  $U = \{j \mid p_{j1} < p_{j2}\}$  como o primeiro grupo de prioridade.

Segundo Passo: Selecionar o grupo de produtos  $V$  com tempo na segunda máquina menor ou igual o da primeira;  $V = \{j \mid p_{j2} \leq p_{j1}\}$  como segundo grupo de prioridade.

Terceiro Passo: Selecionar  $U$  por STP na primeira máquina.

Quarto Passo: Selecionar  $V$  por LPT na Segunda máquina.

Pode-se exemplificar o funcionamento da Regra de Johnson:

Tabela 3:

Problema de scheduling em duas máquinas

Produto $j$ :	1	2	3	4	5	6	7	8
$P_{j1}$ :	5	2	1	7	6	3	7	5
$P_{j2}$ :	2	6	2	5	6	7	2	1

Primeiro Passo: O conjunto  $U$  de tempos menores na primeira máquina é  $U = \{2, 3, 6\}$ ; este será o primeiro bloco de produtos processados.

Segundo Passo: O segundo grupo formado pelos produtos de menor tempo na Segunda máquina será  $V = \{1, 2, 4, 5, 7, 8\}$ .

Terceiro Passo:  $U$  é ordenado de forma crescente em relação aos tempos na primeira máquina: 3 – 2 – 6.

Quarto Passo:  $V$  é ordenado de forma decrescente em relação aos tempos na Segunda máquina: 5 – 4 – 7 – 1 – 8.

Tabela 4:

Solução Ótima para o Problema de Scheduling de 2 máquinas

Produto $j$ :	3	2	6	5	4	7	1	8
$p_{j1}$ :	1	2	3	6	7	7	5	5
$p_{j2}$ :	2	6	7	6	5	2	2	1
Final em M1	1	3	6	12	19	26	31	36
Final em M2	3	9	16	22	27	29	33	37

Note que o tempo total de processamento do conjunto de peças, *makespan*, é 37; O cálculo desse indicador se dá pelo tempo total desde de a entrada do primeiro produto na primeira máquina até a saída do último produto na última máquina. As duas últimas linhas da tabela acima representam a progressão do processo produtivo através dos tempos nas duas máquinas, também ilustrado no gráfico abaixo.

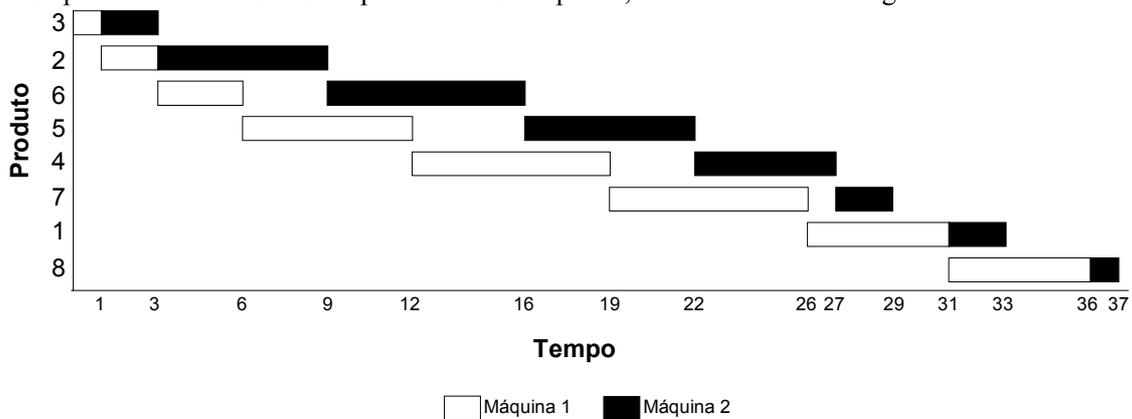


Figura 1: Diagrama de Gantt do exemplo da Tabela 4

**Scheduling para Três – máquinas**

Resultados exatos para o problema com três máquinas existe uma extensão da regra de Johnson.

No caso de três máquinas utiliza-se os vetores:  $P_{i1}$  e  $P_{i2}$ .

Onde,

$$P_{i1} = p_{i1} + p_{i2}; P_{i2} = p_{i2} +$$

Para solucionar o scheduling, basta aplicar o algoritmo  $P_{i1}$  e  $P_{i2}$ .

**Minimize Idle Time (MIT)**

Essa heurística, baseada na regra de Johnson, foi desenvolvida para diminuir o tempo de máquinas inativo durante o processo de funcionamento passo a passo utilizando o exemplo anterior.

Primeiro Passo: Calcular a soma dos tempos de processamento de cada máquina. Este cálculo é apresentado na última coluna.

Produto	M1	M2	M3	M4
1	5	12	15	4
2	10	8	6	9
3	13	17	5	8
4	21	13	18	14
5	15	6	9	4
6	14	10	3	7
7	12	5	12	17

7 12 5 12 17 6 12 40 562 831 579 46 78



	M1	M2	M3	M4	M5	MIT
8	13	20	24	32	35	
2	23	31	37	46	50	(23-20)+(31-24) +(37-32)+(46-35)=26
7	25	30	36	45	58	(25-20)+(30-24)+(36-32)+(45-35)=25
5	28	34	43	47	60	(28-20)+(34-24)+(43-32)+(47-35)=41

Note, nas colunas de M1 até M5, a progressão com que os produtos saem das máquinas num determinado tempo. Na primeira linha, o produto de número 8, por exemplo, deixa a primeira máquina aos treze minutos, dessa forma, o próximo produto sairia da máquina 1 com o seu tempo de processamento acrescido treze minutos em que a máquina estava ocupada, ou seja, dessa forma é calculada o tempo de ciclo dos dois primeiros produtos, sendo o produto 8 o primeiro.

Para o cálculo do MIT, leva-se em conta o tempo em que as máquinas não operam, um exemplo seria, o produto 2 deixa a máquina 1 após 23 minutos, mas o produto 8 (primeiro da seqüência) deixa a máquina 2 aos 20 minutos, conclui-se aqui que a máquina 2 ficou 3 minutos inativa. Aquela tiver menor tempo de máquinas inativo passa a ser escalado.

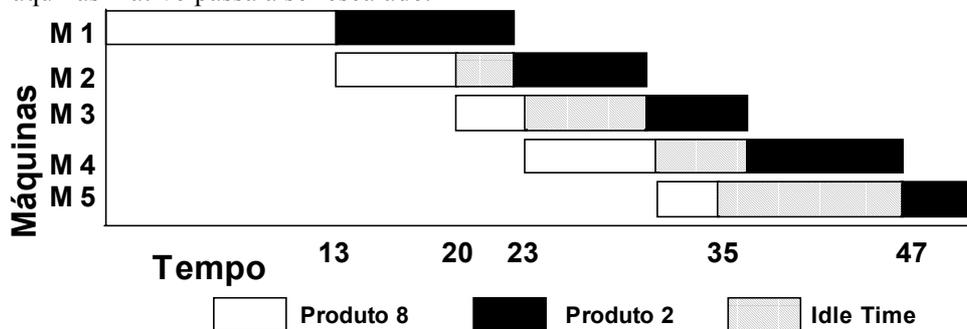


Figura 2: Ilustração do cálculo do MIT

A figura acima mostra a dinâmica do método de cálculo do MIT. Onde os produtos previamente relacionados são comparados com o produto anterior já escalado na seqüência definitiva. O produto que tiver menor MIT, dentre os três previamente escalados, é então selecionado. Na figura, onde existe uma comparação entre os produtos 8 (já selecionado) e 2, o MIT é visualizado, em cada máquina, pelas barras destacadas. O MIT de cada produto é a soma de todo seu Idle Time.

Quarto Passo: Confirmar no scheduling o produto com menor MIT (7); então comparar os próximos 3 produtos com ele, utilizando as mesmas regras e novamente ajustando os tempos até que todos os produtos tenham sido selecionados e a seqüência montada. Chega-se assim, ao seguinte scheduling, {8,7,2,1,5,6,3,9,4,10}

### Campbell, Dudek and Smith (CDS)

Este método é baseado na regra de Johnson. O CDS pressupõe que produzir o scheduling com trabalhos com menores tempos no início e maiores tempos no final, Sule (1997), resulta na otimização da produção. Como na extensão da regra de Johnson, aqui são utilizados dois vetores  $P_{j1}$  e  $P_{j2}$ .

Onde,

$$P_{j1} = \sum_{k=1,i} p_{jk} \text{ e } P_{j2} = \sum_{k=1,i} p_{j, m - k + 1}$$

(i.e.,  $P_{j1}$  é o valor da soma dos tempos do produto j nas primeiras i máquinas e  $P_{j2}$  é a soma para as últimas i máquinas). Basta resolver o scheduling através do algoritmo de Johnson. Naturalmente obtêm-se  $m - 1$  seqüências, a melhor será aquela com menor tempo de ciclo.

Os passos a seguir determinam a maneira de se encontrar uma solução perto do ótimo:

Primeiro Passo: Selecionar 2 auxiliares (A e B) usando a regra de Johnson para converter m máquinas em (m-1) pares. Construindo m-1 cenários para  $A_{ij}$  e  $B_{ij}$ .



Segundo Passo: Aplicar o método de Johnson para cada  $A_{ij}$  e  $B_{ij}$ , este produzirá  $m-1$  schedules como a seguir:

Nº	Seqüência
1	{1,10,9,7,6,4,5,3,2,8}
2	{1,7,6,9,10,4,3,5,2,8}
3	{7,6,1,9,10,4,5,3,2,8}
4	{7,6,1,10,4,9,3,5,2,8}

Terceiro Passo: Calcular o tempo de ciclo para cada seqüência:

Nº	Seqüência	Makespan
1	{1,10,9,7,6,4,5,3,2,8}	175
2	{1,7,6,9,10,4,3,5,2,8}	174
3	{7,6,1,9,10,4,5,3,2,8}	179
4	{7,6,1,10,4,9,3,5,2,8}	166

Quarto Passo: Selecionar o scheduling com menor tempo de ciclo. Neste caso o melhor scheduling seria o número 4.  $S = \{7, 6, 1, 10, 4, 9, 3, 5, 2, 8\}$

### Palmer

Também conhecido como método de diferença, é baseado na regra de Johnson, ou seja, a prioridade seqüencial é dada aos trabalhos que têm forte tendência a progredir começando com de tempos curtos para tempos longos. Essa prioridade é calculada através do índice  $\pi_j$ .

Primeiro Passo: Calcular o índice  $\pi_j$  usando a seguinte equação, onde  $P_{ji}$  é o tempo de processamento do produto  $j$  na máquina  $i$ :

$$\pi_j = (m-1) P_{jm} + (m-3) P_{j(m-1)} + (m-5) P_{j(m-2)} + \dots - (m-5) P_{j3} - (m-3) P_{j2} - (m-1) P_{j1}$$

Segundo Passo: Construir a seqüência baseada na ordem decrescente de magnitude de  $\pi_j$ .

### Implementação das heurísticas na linguagem C

Cada método segue um procedimento que pode ser tratado como uma rotina programável em ambiente computacional, assim as heurísticas são implementadas em linguagem C com a finalidade de maior agilidade no processo de obtenção de scheduling. Uma vez encontrado o scheduling pelos métodos heurísticos poder-se-á utilizar o simulador. O simulador atua na validação e verificação dos resultados gerados pelo uso dos métodos heurísticos. Bem como, implementar o uso de estatísticas para comparar os resultados e analisar o comportamento do processo produtivo. Novamente o scheduling pode variar e ser ou não uma boa opção dependendo das condições e modificações que permite o simulador.

Os programas foram compilados e comportaram-se como previsto agilizando e muito na elaboração do scheduling para cada um dos métodos. Não é apresentado aqui a descrição dos programas.

Os programas geram seqüências diferentes para as três heurísticas, mostradas na tabela abaixo. No entanto, afim de se estudar o desempenho desses métodos, é necessário medir o desempenho de cada uma das seqüências através, por exemplo, de modelos de simulação.

Tabela 6

Seqüências finais de cada heurística

Método	Seqüência Final
MIT	{8,7,2,5,1,6,9,3,10,4}
CDS-1	{1,10,7,6,4,5,3,9,2,8}
CDS-2	{1,7,6,9,10,4,3,5,2,8}
Palmer	{10,1,6,7,2,9,4,5,3,8}



Na tabela acima pode-se notar a presença de duas seqüências calculadas da heurística CDS, isso se deve ao fato da heurística para 5 máquinas, neste caso, gerar 4 seqüências, dessa forma, escolhe-se as duas de menor makespan, que foram denominadas CDS-1 e CDS-2.

### A-SIM

O simulador A-SIM, é um software desenvolvido pelo Laboratório Tecnologia de Gestão Empresarial (LTGE) do CenPRA. Essa linguagem de simulação é formada por símbolos gráficos (ou ícones) que possibilitam a modelagem de eventos discretos que ocorrem em sistemas físicos. O modelo é o conjunto de símbolos interligados que representa o comportamento discreto do sistema físico. Estes modelos utilizam os conceitos de recurso, entidade, fila, operação, entre outros para descrever o comportamento dinâmico de sistemas.

### Resultados

Quando executados os modelos do simulador geram-se relatórios estatísticos dos quais são extraídos dois índices para análise: *final time of collection* e *idle time*.

O primeiro indica o valor total de tempo gasto para processar todas as entidades e o segundo refere-se ao tempo inativo de cada operação (máquina) durante todo o processo. Esses são os indicadores para a análise.

Para efeito de cálculo chama-se *total idle time* a soma dos tempos de máquina inativo em cada método.

Tabela 9:  
Resultados da Simulação

Método	Final Time of Collection	Total Idle Time
MIT	192	422
CDS-1	170	312
CDS-2	167	297
Palmer	169	307

Os valores mostrados acima permitem identificar que, para este sistema estudado, o método heurístico CDS é o mais eficiente. Seguido pelo método de Palmer. Enquanto a heurística MIT, desenvolvida especialmente para este tipo de estudo, não demonstra muita rentabilidade. Porém, caso fossem simuladas várias seqüências aleatoriamente seria esperado que esses três resultados estivessem essencialmente entre os melhores.

Nota-se ainda, que pequenas alterações provocadas no tempo de ciclo do sistema resultam de uma grande diferenciação em relação ao tempo inativo de máquinas. Na tabela acima pode-se identificar as variações de tempo entre as heurísticas MIT, a pior para o sistema descrito, e CDS-2, a melhor. Enquanto o tempo de ciclo varia cerca de 13%, o tempo de obsolescência das máquinas varia em torno de 30%.

Observa-se assim que é possível, através do estudo do scheduling que melhor se adapta a uma sistema, reduzir custos no âmbito operacional uma vez que é possível diminuir o tempo de máquinas inativo. Mesmo que não variando muito o tempo de fabricação.

### Conclusão

A utilização de métodos que permitem a geração de dados referentes a um sistemas produtivo, sem que exista a necessidade da experimentação física, é fator essencial para o planejamento da produção, onde se visa atingir metas em processos e redução de custos.



Neste trabalho pode-se identificar, pelo estudo das diferentes heurísticas, a possibilidade da redução do tempo de máquinas inativo num sistema produtivo flow shop, uma vez implantadas e testadas as diferentes seqüências de produção, que gera uma redução também no tempo de processamento do sistema e nos custos da produção.

Para o sistema estudado observa-se uma maior eficiência da heurística CDS, seguida de perto pela heurística de Palmer. Já o método MIT não se mostra razoável, visto que fora desenvolvido especialmente a esse propósito. Cabe lembrar que estes resultados retratam o comportamento dessas heurísticas apenas no sistema estudado e não podem ser vistos como regras gerais

Este tipo de estudo deve ser feito no nível operacional visando uma melhor utilização dos recursos da produção. Por fim, o A-SIM comportou-se como esperado e mostrou ser uma ferramenta eficaz para a solução desse tipo de problema.

### Referências

Britan, G. B. and Yanasse, H. H. (1982), “Computational complexity of the capacitated lot size problem”, *Management Science* 28/10, 1174-1185.

Domschke, W., Scholl, <sup>a</sup>, and Voß, S. (1993), *Produktionsplanung – Ablauforganisatorische Aspekte*, Springer-Verlag, Berlin.

Ezpeleta, J., Colom, J.M., (1997), Automatic Synthesis of Colored Petri Nets for the Control of FMS, *IEEE Transactions on Robotics and Automation*, vol. 13, no.3, pp.327-337.

Joshi, S. B., and Smith, J. S., (1994), *Computer Control of Flexible Manufacturing Systems*, Chapman & Hall, London, UK.

Kattan, I. A., (1997), Design and Scheduling of Hybrid Multi-Cell Flexible Manufacturing Systems, *International Journal of Production Research*, vol.35, no.5, pp. 1239-1257, Taylor & Francis, London, England.

Kattan, I. A., et al (2001), “Minimizing cycle time of job scheduling using Petri Nets a study of heuristic methods”, *The 6<sup>th</sup> IFAC Workshop on Intelligent Manufacturing Systems IMS 2001*, Poznan, Poland.

Manual do Usuário A-SIM.

Morton, T. E., Pentico, D. W., (1993), *Heuristic Scheduling Systems*, John Wiley & Sons Inc., USA.

Sule, D. R., (1997), *Industrial Scheduling*, PWS Publishing Company, Boston.