

# ORDENAÇÃO DE TAREFAS EM OFICINAS DE MÁQUINAS: COMPARAÇÃO DA EFICIÊNCIA DE ALGUNS ALGORITMOS

Paulo F. Fleury\*

Professor adjunto da COPPEAD-Universidade Federal do Rio de Janeiro. Assessor de Coordenação Científica da CAPES/MEC.

\* O autor agradece o apoio financeiro da CAPES e da FINEP, para a realização da pesquisa que resultou na elaboração deste documento.

---

## INTRODUÇÃO

A ordenação de tarefas em oficinas de máquinas pode ser entendida como um caso particular do problema clássico de "Job Shop"

"Job Shop" pode ser definido como um sistema composto de **M** máquinas destinadas a realizar operações em tarefas que a elas chegam. O processo de chegada e o percurso a ser seguido pelas tarefas determinam, de uma certa maneira, a complexidade de um "job shop"

Em relação ao percurso, dois casos extremos podem ser observados:

— o "job shop" puro, no qual cada uma das tarefas tem a mesma probabilidade de ser operada em qual-

quer máquina, dentro de qualquer sequência;

— o "flow shop" puro, no qual todas as tarefas devem passar pelas **M** máquinas, seguindo uma mesma sequência.

Com respeito ao processo de chegada, dois casos podem ser observados:

— o caso "dinâmico", em que as tarefas chegam à oficina independentemente, de acordo com alguma distribuição de chegada;

— o caso "estático", em que as tarefas chegam à oficina em grupos estanques, para serem operadas.

Da enorme variedade de casos pos-

síveis, o mais complexo é, sem dúvida, o do "job shop" puro com chegada "dinâmica". Para estudo destes casos, a solução adotada tem sido a construção de modelos de simulação digital, a fim de se estudar a eficiência de diversas regras de prioridade.

Diversos autores — SANDERMAN (1961), GERE (1966), EILON (1969), NELSON (1967), CONWAY (1962) — estudaram a performance de regras de prioridade para a sequenciação de tarefas em "job shop" dinâmicos.

No caso mais simples de "flow shop" estático, já se podem formular modelos matemáticos que as representem. Diversos modelos de programação linear inteira e programação linear com resultados arredondados para inteiros já foram utilizados na determinação de uma solução ótima para o problema. Métodos heurísticos otimizantes "branch and bound" e de enumeração também já foram usados.

O problema dos métodos otimizantes se concentra, principalmente, no alto tempo de computação exigido e na sofisticação de cálculo. Tal problema cresce exponencialmente com o número de tarefas e máquinas na oficina.

Com o objetivo de minimizar os problemas acima, vem sendo desenvolvida uma série de algoritmos

heurísticos que, embora não leve sempre à solução ótima, exigem um tempo de computação e uma simplicidade de cálculo significativamente vantajosos em relação aos modelos otimizantes.

### O Problema de Ordenação em Flow Shop Estático

O problema de ordenação ocorre quando se tem  $N$  tarefas para serem processadas em  $M$  máquinas e se deseja determinar a sequência em que essas  $N$  tarefas serão processadas nas  $M$  máquinas. De uma forma mais precisa, o problema de ordenação pode ser definida da seguinte forma:

$N$  tarefas devem ser operadas sequencialmente em cada uma das  $M$  máquinas;

uma tarefa não pode ser processada na máquina  $j$  antes de ser operada na máquina  $j - 1$ , ( $j = 2, m$ );

os tempos de operação de cada tarefa em cada máquina são conhecidos e bem determinados e independentes da tarefa anterior;

todas as  $N$  tarefas estão disponíveis no início das operações;

as máquinas estão sempre disponíveis e nunca falham;

procura-se uma ordenação que minimize o tempo total gasto para

completar a operação de todas as tarefas em todas as máquinas.

Os dados relativos ao problema podem ser representados por uma matriz  $\{X_{ij}\}$  de dimensão  $N \times M$ , que contém os tempos de operação das  $N$  tarefas nas  $M$  máquinas. Desta forma, um elemento  $X_{ij}$  representaria o tempo de operação da tarefa  $i$  na máquina  $j$ .

### Os Diversos Algoritmos Heurísticos

Johnson (1964) foi um dos primeiros autores a formular o problema de ordenação, tendo apresentado um algoritmo que fornece a solução ótima para o caso particular em que o número de máquinas  $M$  é igual a 2, e para quaisquer valores de  $N$ .

Johnson também analisou seu algoritmo para o caso de  $m=3$  máquinas, e concluiu que o mesmo daria soluções ótimas apenas para o caso especial em que o maior tempo de operação na máquina do meio (máquina 2) fosse menor que o tempo de operação nas máquinas 1 e 3.

Merbach (1966) desenvolveu um algoritmo que ele chamou de método aproximado, e que se baseia no princípio de que a minimização do tempo total de término de todas as tarefas é equivalente à minimização dos tempos de ociosidade da última máquina. Este método pode ser aplicado para quaisquer valores de  $M$  e  $N$ .

Smith "et al" (1970) fizeram uma generalização do algoritmo de Johnson para um número  $M$  qualquer de máquinas, dando como resultado um método apenas aproximado.

Palmer (1965) também desenvolveu um algoritmo heurístico não otimizador, generalizado para o caso de  $N$  e  $M$  quaisquer. Este algoritmo é baseado nos chamados índices de tendência, que tentam medir a tendência de aumento nos tempos de operação de uma determinada tarefa, da primeira para a última máquina.

Finalmente, existe o algoritmo desenvolvido por Nicholson (1967), chamado pelo autor de método de sequenciação, que se baseia em permutações efetuadas entre os diversos "jobs"

### Justificativas para o Estudo

Do que foi exposto anteriormente, é possível observar dois fatores predominantes, que deveriam ser levados em consideração quando da escolha de um método para a ordenação de tarefas em uma oficina de máquinas:

- o tempo de computação envolvido na determinação de uma solução;
- o grau de aproximação ao ótimo gerado pelo método utilizado.

Estudos anteriores indicam a existência de um "trade off" entre estes

dois parâmetros de performance.

Os métodos mais sofisticados, que tendem a oferecer uma solução mais próxima do ótimo, têm como desvantagem principal um maior custo de processamento, representado por um maior tempo de computação.

### Um Método de Análise

Se o problema de ordenação for analisado sob o ponto de vista de custos, os seguintes parâmetros deveriam ser considerados:

$C_e$  — custo de espera / unidade de tempo;

$C_c$  — custo de computação / unidade de tempo;

$\Delta t = (t_1 - t_2)$  — diferença entre os tempos de computação dos métodos;

$T_1$  — tempo total de término para a sequência encontrada pelo método 1;

$T_2$  — tempo total de término para a sequência encontrada pelo método 2.

Com base nestes parâmetros, poderíamos então afirmar que o método 1 será mais vantajoso que o método 2 se:

$$(T_2 - T_1) C_e > (t_1 - t_2) C_c$$

Seja  $T_i$  o tempo de término dado pela ordenação gerada pelo método 1. Sabe-se que  $T_i$  é constituído de duas parcelas, que são:

1.  $T_r$  — tempo real de operação na máquina  $M$ , que corresponde à somatória dos tempos de operação de cada uma das  $N$  tarefas na máquina  $M$ :

$$T_r = \sum_{i=1}^n (X_{im})$$

onde  $m$  corresponde à última máquina e  $|X_{ij}|$  é a matriz de operação das  $n$  tarefas nas  $m$  máquinas.

2.  $T_{oc} = T_i - T_r$  — corresponde ao tempo ocioso da última máquina.

Considerando que  $T_{oc} / T_i$  é a percentagem de tempo ocioso na última máquina, temos:

$$p = \frac{T_{oc}}{T_i} \quad T_i - p = T_i - T_r \quad T_i = \frac{T_r}{(1 - p)}$$

Desta forma, se conhecermos a percentagem de tempo ocioso gerado por um determinado método, podemos determinar os tempos totais de operação que serão gerados por aquele método.

Esta percentagem deverá variar de

caso para caso, e deverá também ser uma função do número de tarefas **N** e do número de máquinas **M**. Uma maneira aproximada de se comparar os diversos métodos seria a de se determinarem valores médios de **p** em função de **M** e **N**.

O objetivo deste estudo é, portanto, o de determinar, através de experimentação as funções  $p = f(N, M)$ , juntamente com as funções dos tempos de computação,  $t = f(N, M)$ , para os diversos algoritmos heurísticos mencionados anteriormente (Merbach, Smith "et al" Palmer e Nicholson), a fim de que se possa tomar uma decisão mais racional na escolha entre métodos.

### Projetos de Experimentos

Considerando-se que **M** medirá o tamanho de uma determinada oficina e **N** o grau de ocupação daquela oficina, decidiu-se pesquisar os valores de percentagem de tempo ocioso **p** para os seguintes casos:

- M igual a 2, 3, 5, 7;
- N igual a 2xm, 4xm, 6xm, e 8xm, resultando em um projeto fatorial com 2 fatores e 4 níveis, dando um total de 2<sup>4</sup> experimentos, de acordo com a tabela 1, abaixo:

**TABELA 1 – PROJETO DE EXPERIMENTOS**

M	N	M	N	M	N	M	N
2	4	2	8	2	12	2	16
3	6	3	12	3	18	3	24
5	10	5	20	5	30	5	40
7	14	7	28	7	42	7	56

Para cada um dos 16 experimentos, foi determinado um tamanho de amostra através do teste de ordenação múltipla de Bechhoffer et. al. (1954).

As amostras foram geradas através da utilização da sub-rotina RANDU da IBM, que gerava conjuntos de dados aleatórios para compor a matriz **N** x **M** de tempos de operação. Esses tempos de operação seguem uma distribuição uniforme entre 1 e 25.

O uso de distribuições uniformes para a geração dos tempos de operação se justifica pelo fato de que outros autores (Smith et al., 1961) testaram a performance deste tipo de algoritmo para distribuições normais, exponenciais e uniformes e observaram que, em média, os piores resultados eram obtidos para as sequências provindas das matrizes geradas por distribuições uniformes. Desta forma, como nos interessa um teste de capacidade de melho-

ria, seria mais interessante escolher os casos que apresentassem maiores discrepâncias.

Para a determinação das funções  $t = f(N.M)$ , decidiu-se por uma série mais ampla de experimentos, que correspondiam a combinações de valores de **M** e **N** de acordo com a tabela 2, abaixo:

**Tabela 2**

M	N	M	N	M	N	M	N
2	5	3	5	5	5	7	5
	10		10		10		10
	15		15		15		15
	20		20		20		20
	25		25		25		25
	30		30		30		30
	35		35		35		35
	40		40		40		40
	45		45		45		45
	50		50		50		50
	60		60		60		60
	70		70		70		70
	90		90		90		90

### Apresentação dos Resultados

A primeira parte dos resultados se refere aos tempos de computação necessários à determinação de uma solução, para cada um dos métodos, os quais foram testados para os di-

versos valores de **M** e **N** de acordo com o projeto de experimentos.

Tais tempos foram obtidos para um computador IBM 1130 do Núcleo de Computação Eletrônica da UFRJ e organizados em forma de tabelas(\*).

A partir dos tempos observados, procurou-se ajustar uma curva que melhor se adaptasse aos pontos obtidos. A curva escolhida foi um polinômio da forma  $t = an^b$ , devido ao bom nível de aproximação dada pela mesma (menos de 5% de desvio).

Se aplicarmos logaritmo à função  $t = an^b$  teremos:

$$\log t = \log a + b \log n$$

Considerando-se que a função modificada é um polinômio do 1º grau, podemos, então, fazer uma regressão pelo método dos mínimos quadrados.

Para isso, foi utilizada a rotina APOL da IBM, que fornece como saída os coeficientes **a** e **b**, que são os parâmetros representativos da curva. Esses parâmetros foram determinados independentemente, para valores diferentes de **m**, segundo a tabela 3.

**TABELA 3 – Relação dos Coeficientes da Curva de Tempo de Computação**

	MERBACH		SMITH		PALMER		NICKOLSON	
	A	B	A	B	A	B	A	B
M = 2	0,0133	1,9161	0,0087	1,5983	0,0032	1,6659	0,0030	2,6952
M = 3	0,0221	1,9186	0,0185	1,5786	0,0061	1,6673	0,0044	2,7319
M = 5	0,0375	1,9351	0,0460	1,5416	0,0103	1,6381	0,0071	2,7593
M = 7	0,0541	1,9360	0,0910	1,4850	0,0210	1,5635	0,0102	2,7647

Um exame da tabela 3 e dos tempos de computação dos diversos algoritmos levou-nos às seguintes observações:

O algoritmo de Nicholson apresenta tempos de computação consistentemente maiores que os demais algoritmos. Embora bastante acentuadas em termos relativos, estas diferenças só são significantes em termos absolutos para valores de **N** maiores que 30.

Os algoritmos de Palmer e Smith apresentam tempos de computação com valores bem próximos um do outro, devendo-se notar, no entanto, a maior tendência de crescimento do algoritmo de Palmer representada por maiores valores do parâmetro **b** da curva de regressão.

– O algoritmo de Merbach apresenta tempos de computação que se situam entre os valores apresenta-

dos pelos algoritmos de Palmer e de Nicholson. Por exemplo, para o caso extremo de  $M = 7$  e  $N = 90$ , o algoritmo de Nicholson consumiu 2.871 seg. comparados com os 24 seg. de Palmer, 81 seg. de Smith e 341 seg. de Merbach.

A segunda parte dos resultados se refere aos dados de performance dos diversos algoritmos, em termos de capacidade de gerar soluções mais próximas do ótimo. De acordo com o descrito anteriormente, esta performance é medida através da percentagem de tempo ocioso da última máquina.

Os resultados obtidos foram testados através do teste de ordenação múltipla de Bechhoffer (1954), e todas as diferenças mostraram ser estatisticamente significantes a um nível de confiança de 0,90.

Em função dos resultados, as seguintes observações podem ser feitas:

— Com exceção dos casos em que o número **M** de máquinas é igual a 2 e para os quais o algoritmo de Smith é sempre ótimo (para  $m=2$  o algoritmo de Dudek se torna igual ao de Johnson), os demais casos apresentam sempre os algoritmos, em termos de performance, dentro da seguinte ordem:

1. Nicholson
2. Smith
3. Palmer
4. Merbach

Por exemplo, para o caso de  $M=7$  e  $N=56$ , o algoritmo de Nicholson apresentou um índice de ociosidade de 14.05%, comparado com 15.18% para Smith, 18.00% para Palmer e 18.37% para Merbach.

— À medida em que o número de máquinas vai aumentando, aumentam também as diferenças absolutas entre os algoritmos de Nicholson e os demais.

— Para os casos em que o número de máquinas **M** é igual a 2, e nos quais se tem a solução ótima dada pelo algoritmo de Palmer, é possível observar que o algoritmo de Nicholson apresenta resultados bem próximos do ótimo. Para o caso específico de  $M=2$  e  $N=16$ , em 100 casos testados, o algoritmo de Nicholson gerou soluções ótimas em 89 casos, contra apenas 46 de Palmer e 36 de Merbach. Das sequências geradas aleatoria-

mente, apenas 2 representaram soluções ótimas.

## CONCLUSÕES A RESPEITO DOS ALGORITMOS

### Algoritmo de Merbach

Os resultados obtidos por esse algoritmo mostram, claramente que a minimização por etapas dos tempos ociosos das máquinas não implica absolutamente na minimização do tempo total de operação na última máquina.

O algoritmo se revelou sempre inferior aos demais, no que se refere à capacidade de otimização. Em relação ao tempo de computação, ele também não é vantajoso, apresentando valores significativamente superiores aos apresentados pelos algoritmos de Palmer e Smith, e próximos aos de Nicholson. Sua utilização, portanto, parece não ser interessante em nenhuma circunstância.

### Algoritmo de Smith

O algoritmo de Smith apresentou resultados bastante interessantes. Em primeiro lugar, para os casos em que **M** é igual a 2, ficou demonstrada a sua superioridade sobre os demais algoritmos. Além de apresentar sempre soluções ótimas, ele requer esforços computacionais bastante pequenos e próximos do algoritmo de menor tempo (Palmer).

Para casos onde **M** é maior que 2, ele ainda permanece interessante, principalmente para número de tarefas acima de 30, onde o algoritmo de Nicholson exige tempos de computação relativamente altos. Obviamente, o custo deste esforço computacional vai depender do equipamento disponível em cada situação.

### Algoritmo de Palmer

O melhor aspecto do algoritmo de Palmer é o seu pequeno tempo de computação, o menor dentre os quatro algoritmos. Entretanto, esta vantagem pode ser considerada insignificante, quando comparada ao algoritmo de Smith.

### Algoritmo de Nicholson

Este algoritmo se revelou o mais eficiente, no que diz respeito à capacidade de otimização para todos os casos em que **M** é maior que 2. Sua maior restrição se refere ao elevado tempo de computação requerido, que tende a se tornar mais crítico à medida em que se aumenta o número **N** de tarefas e o número **M** de

máquinas.

Entretanto, para um número de tarefas abaixo de 30 e para um número de máquinas até 7, ele parece ser o mais interessante.

Como comentário final, poderíamos dizer que o estudo comprovou o "trade-off" existente entre esforço computacional e capacidade de otimização, o que significa que uma decisão entre as vantagens de um algoritmo sobre o outro dependerá, na maioria das vezes, do caso particular em estudo.

Parece, entretanto, que os algoritmos de Smith e de Nicholson serão sempre superiores aos outros dois, em razão, principalmente, das pequenas diferenças de tempo de computação e das grandes vantagens na capacidade de otimização.

(\*) Os resultados apresentados nestas tabelas estão à disposição dos interessados que poderão obtê-las com o autor ou na Secretaria Editorial da Revista de administração da USP.

## BIBLIOGRAFIA

- BECHHOFFER, R. E.; DUNNET, C. W.; SOBEL, M — A two sample multiple decision procedure for ranking means of normal populations with a common unknown variance. *Biometrika*, 41: 170-6, 1954.
- CONWAY, R. W. & MAXWELL, W.L. — Network dispatching by the shortest operation discipline. *Operation Research*, 10 (1): 51-73, Jan./fev. 1962.
- EILON, Samuel. — Multi product scheduling in a chemical plant. *Management Science*, 15 (6): 267-79, february 1969.

- GERE JR., W. S.** — Heuristic in job shop scheduling. *Management Science*, **13 (3)**: 167-90, november 1966.
- JOHNSON, S. M.** — Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, **1(1)**: 61-8, 1964.
- MERBACH, H. M.** — Ein verfahren zur losung von reinenfolgung. *Zeitschrift fur Wirtschaftliche Feilitung*, **61 (3)**: 147-52, 1966.
- NELSON, R. T.** — Labor and machine limited production systems. *Management Science*, **13 (9)**: 648-71, may 1967.
- NICHOLSON, T. A. J.** — A sequential method for discrete optimization problem and its applications to the assignment, traveling salesman and three machine scheduling problems. *Journal Institute of Matematics and its Applications*, **(3)**: 362-75, 1967.
- PALMER, D. S.** — Sequencing jobs thorough a multistage process in the minimum total time; a quick method of obtaining a near optimum. *Operational Research Quarterly*, **16 (1)**: 101-7, march 1965.
- SANDERMAN, P.**; Empirical Design of Priority waiting times for Jobbing shop control. *Operation Research*, **9 (4)**: 44655, August 1961.
- SMITH, M. L.; DUDEK, R. A.; CAMPBELL, H. G.** — A heuristic algorithm for the N job M machine sequencing problem. *Management Science*, **16 (10)**: 630-7, june 1970.