COPPEAD/UFRJ

RELATÓRIO COPPEAD Nº 11

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

PAULO F. FLEURY*

OUTUBRO 1977

* Professor Adjunto da COPPEAD - Universidade Federal do Rio de Janeiro.
O autor agradece o apoio financeiro da CAPES e da FINEP, para a realização da pesquisa que resultou na elaboração deste documento.

SUMÁRIO

Diversos artigos têm sido escritos a respeito do problema de ordenação de tarefas em oficina de máquinas, para o caso do 'flow shop' estático. Alguns métodos otimizantes como programação linear inteira e algoritmos que usam o método do 'branch and bound', foram desenvolvidos, mas têm como restrição o fato de que se tornam impraticáveis para problemas de dimensão um pouco maior, devido ao elevado tempo de computação.

Para solucionar este problema, diversos autores propuseram algoritmos heurísticos que, embora não levem sempre à solução ótima, requerem um tempo de computação sensivelmente menor. Tais algoritmos, no entanto, não foram suficientemente testados para que se possa fazer um julgamento definitivo de suas perfomances.

Neste trabalho, procuramos analisar alguns algoritmos, com relação ao tempo de computação requerido e a capacidade de otimização, buscando com isto fornecer os dados necessários para uma tomada de decisão no momento da escolha entre métodos.

ORDENAÇÃO DE TAREFAS EM OFICINAS DE MÁQUINAS:

COMPARAÇÃO DA EFICIÊNCIA DE ALGUNS ALGORITMOS HEURÍSTICOS

INTRODUÇÃO

A ordenação de tarefas em oficinas de máquinas pode ser entendido 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 uma 'job shop'.

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

i. o 'job shop' puro, no qual cada uma das tarefas tem a mesma probabilidade de ser operada em qualquer máquina, dentro de qualquer seqüência.

ii. o 'flow shop' puro, no qual todas as tarefas devem passar pelas M máquinas, seguindo uma mesma seqüência.

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

i. o caso 'dinâmico', em que as tarefas chegam à oficina independentemente, de acordo com alguma distribuição de chegada.

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

Da enorme variedade de casos possíveis, 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[10], GERE[4], EILON[3], NELSON[7], CONWAY[2], estudaram a performance de regras de prioridades para a seqüenciação de tarefas em 'job shop' dinâmicas.

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 significantemente 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 seqüência em que essas N tarefas serão processadas nas M máquinas. De uma forma mais precisa, o problema de ordenação pode ser definido da seguinte forma:

1. N tarefas devem ser operadas seqüencialmente em cada uma das M máquinas;

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

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

4. todas as N tarefas estão disponíveis no início das operações;

5. as máquinas estão sempre disponíveis e nunca falham;

6. 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 x 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[5] 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[6] 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 qualquer valor de M e N.

Smith et al[11], fizeram uma generalização do algoritmo de Johnson para um número M qualquer de máquinas, dando como resultado um método apenas aproximado.

Palmer[9] também desenvolveu um algoritmo heurístico não otimizante, generalizado para o caso de N e M qualquer. 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[8], chamado pelo autor de método de seqüenciação, 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.

i. O tempo de computação envolvido na determinação de uma solução;

ii. 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:

i. Ce - custo de espera / unidade de tempo

ii. Cc - custo de computação / unidade de tempo

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

iv. T_1 — tempo total de término para a seqüência encontrada pelo método 1.

v. T_2 — tempo total de término para a seqüência encontrada pelo método 2.

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

$$(T_2 - T_1) \cdot Ce > (t_1 - t_2)\, Cc$$

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 ao somatório dos tempos de operação de cada uma das N tarefas na máquina M.

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

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:

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

Desta forma, se conhecemos 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.

Projeto 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:

i) M igual a 2, 3, 5, 7

ii) N igual a 2xm, 4xm, 6xm e 8xm, resultando em um projeto fatorial com 2 fatores e 4 níveis, dando um total de 2^4 experimentos, de acordo com a tabela 1, abaixo:

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

TABELA 1 - PROJETO DE EXPERIMENTOS

Para cada um dos 16 experimentos, foi determinado um tamanho de amostra através do teste de ordenação múltipla de Bechhoffer et.al.[1]

As amostras foram geradas através da utilização da subrotina 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.[10]) 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 seqüências provindas das matrizes geradas por distribuições uniformes. Desta forma, como nos interessa um teste de capacidade de melhoria, 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:

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

TABELA 2

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 diversos 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.

(*) 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.

	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

TABELA 3 - RELAÇÃO DOS COEFICIENTES DAS CURVAS DE TEMPO DE COMPUTAÇÃO

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

i) 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.

ii) 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 representado por maiores valores do parâmetro b da curva de regressão.

iii) O algoritmo de Merbach apresenta tempos de computação que se situam entre os valores apresentados 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[1], 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:

i) 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.

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

iii) 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 seqüências geradas aleatoriamente, apenas 2 representaram soluções ótimas.

Conclusões a respeito dos algoritmos

a) Algoritmo de Merbach

Os resultados obtidos por esse algoritmo mostram, claramente, que a mini

(*) Estes resultados estão à disposição dos interessados que poderão obtê-los com o autor ou na Secretaria Editorial da Revista de Administração da USP.

mização 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.

b) 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.

c) 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.

d) 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 aprticular em estudo.

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

ANEXO I

TEMPOS DE COMPUTAÇÃO PARA OS DIVERSOS ALGORITMOS

N	MERBACH	SMITH	PALMER	NICHOLSON
5	0,310	0,130	0,070	0,280
10	1,080	0,340	0,240	1,470
15	2,320	0,620	0,450	4,130
20	4,110	0,940	0,890	8,870
25	6,070	1,410	1,230	16,340
30	8,740	1,940	1,680	27,140
35	11,950	2,380	2,170	41,880
40	15,730	3,080	2,840	60,870
50	24,380	4,490	4,080	115,280
60	34,800	6,090	5,680	194,980
70	46,120	7,870	8,700	305,280
90	76,180	12,640	11,520	634,930

TABELA I-1

Tempos de Computação em segundos para M = 2

N	MERBACH	SMITH	PALMER	NICHOLSON
5	0,520	0,280	0,110	0,430
10	1,810	0,680	0,280	2,310
15	3,840	1,320	0,510	6,750
20	6,670	1,690	0,860	14,490
25	10,410	2,850	1,290	26,950
30	14,890	3,940	1,830	44,910
35	19,930	4,890	2,460	69,870
40	26,280	6,200	3,180	101,900
50	40,450	8,820	4,870	194,600
60	57,500	12,460	6,930	330,330
70	79,220	16,070	9,360	517,580
90	127,690	24,830	15,290	1.082,090

TABELA I-2

Tempos de Computação em segundos para M = 3

N	MERBACH	SMITH	PALMER	NICHOLSON
5	0,900	0,630	0,160	0,730
10	3,180	1,620	0,500	3,970
15	6,940	2,810	0,760	11,770
20	12,040	4,420	1,400	25,680
25	18,720	6,240	1,890	48,280
30	26,650	8,100	2,530	80,790
35	36,140	10,400	3,140	125,730
40	46,960	12,880	3,940	185,190
50	72,520	18,900	6,510	352,350
60	105,480	26,940	7,950	599,930
70	140,670	34,000	12,050	941,830
90	238,040	53,000	19,300	1.973,500

TABELA I-3

Tempos de Computação em segundos para M = 5

N	MERBACH	SMITH	PALMER	NICHOLSON
5	1,290	1,140	0,230	1,040
10	4,630	2,760	0,490	5,760
15	10,030	4,920	1,030	16,810
20	17,530	7,330	1,624	37,410
25	26,960	10,050	2,460	69,590
30	38,480	13,130	3,470	117,200
35	51,920	16,960	4,390	182,030
40	67,790	21,000	5,650	267,310
50	105,670	30,400	8,790	512,090
60	152,400	41,120	12,250	867,420
70	205,070	52,850	15,010	1.368,490
90	341,670	81,540	23,940	2.871,170

TABELA I-4

Tempos de Computação em segundos para M = 7

ANEXO II

RESULTADOS DAS PERCENTAGENS DE
TEMPO OCIOSO NA ÚLTIMA MÁQUINA
PARA OS DIVERSOS ALGORITMOS

	Original		Merbach		Smith		Palmer		Nicholson	
N	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.
4	26,88	227,3	20,63	295,0	18,10	236,1	19,73	227,9	18,18	234,4
8	20,73	145,9	14,71	193,0	11,41	149,9	13,09	144,3	11,56	149,3
12	17,73	108,9	11,51	130,8	8,90	105,7	10,15	103,1	9,04	104,4
16	14,97	79,2	9,25	101,1	6,72	107,3	7,79	80,3	7,04	80,3

TABELA II-1

Médias e Variâncias das percentagens de tempo ocioso para M = 2

	Original		Merbach		Smith		Palmer		Nicholson	
N	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.
6	39,44	143,2	33,51	240,8	28,68	191,1	30,54	183,7	27,93	182,9
12	27,93	98,7	20,77	167,5	17,09	122,1	19,11	115,6	16,12	121,4
18	23,25	80,8	15,91	128,0	12,85	95,9	14,22	90,3	12,14	93,2
24	20,20	75,0	13,40	94,3	10,65	74,0	11,55	68,6	9,88	71,4

TABELA II-2

Médias e Variâncias das percentagens de tempo ocioso para M = 3

	Original		Merbach		Smith		Palmer		Nicholson	
N	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.
10	45,41	76,2	40,40	119,3	36,16	95,6	37,99	90,3	34,45	93,2
20	33,44	67,4	27,29	103,2	23,71	77,4	24,96	72,5	21,38	73,2
30	28,25	51,8	21,80	79,6	19,07	61,7	19,50	61,1	16,63	63,7
40	23,91	47,7	17,56	91,9	15,63	56,5	16,61	65,0	13,13	66,1

TABELA II-3

Médias e Variâncias das percentagens de tempo ocioso para M = 5

	Original		Merbach		Smith		Palmer		Nicholson	
N	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.	Med.	Var.
14	47,89	70,0	44,57	76,9	39,60	78,2	41,47	76,7	37,46	76,2
28	34,95	38,0	29,45	77,0	26,20	50,1	27,30	52,0	22,46	45,8
42	28,67	19,3	22,62	36,5	19,32	15,0	20,64	21,6	16,45	20,2
56	25,75	22,3	18,37	31,6	15,18	24,6	18,00	23,0	14,05	26,3

TABELA II-4

Médias e Variâncias das percentagens de tempo ocioso para M = 7

BIBLIOGRAFIA

1. 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.

2. CONWAY, R.W. & MAXWELL, W.L.. Network dispatching by the shortest operation discipline. Operation Research, 10 (1): 51-73, Jan./fev. 1962.

3. EILON, Samuel. Multi product scheduling in a chemical plant. Management Science, 15 (6): 267-79, february 1969.

4. GERE JR., W.S., Heuristc in job shop sheduling. Management Science, 13 (3): 167-90, november 1966.

5. JOHNSON, S.M.. Optimal two and three stage production schedules with setup times included. Naval Research Logistics Quartely, 1 (1): 61-8, 1964.

6. MERBACH, H.M. Ein verfahen zur losung von reinenfolgung. Zeitsehrift fur Waitschaftliche Feilitung, 61 (3): 147-52, 1966.

7. NELSON, R.T., Labor and machine limited production systems. Management Science, 13 (9): 648-71, may 1967.

8. NICHOLSON, T.A.J.. A sequential method for discrete optimization problem and its applications to the assgmment, traveling selesman and three machine scheduling problems. Journal Institute of Matematics and its Application, (3): 362-75, 1967.

9. PALMER, D.S.. Sequencing jobs through a multistage process in the minimum total time; a quick method of obtaining a near optimum. Operational Research Quartely, 16 (1): 101-7, march 1965.

10. SANDERMAN, P.; Empirical Design of Priority waiting times for Jobbing shop control. Operation Research, 9 (4): 446-55, August 1961.

11. SMITH, M.L.; DUDEK, R.A.; CAMPBELL, H.G.. A heuristic algorithim for the N job M machine sequencing problem. Management Science, 16 (10): 630-7, june 1970.

/rcrn