

Programação da produção – despacho e sequenciamento

Victor Sequeira Roldão(*)

1. Introdução

A programação da produção com recurso a regras de despacho tem vindo progressivamente a ser cada vez mais utilizada. Com efeito, verifica-se um melhor desempenho quando se desenvolvem regras que orientam a acção do que quando se pretendem definir planos ilusoriamente perfeitos que tudo consideram.

Normalmente as regras de despacho são estudadas em ambientes em que:

- Os recursos em equipamento são pouco abundantes e a mão-de-obra não é um factor condicionante.
- A interrupção de uma operação não é permitida (por exemplo por avaria).
- A duração das operações associada a um lote, bem como os tempos de preparação são independentes da ordem de passagem dos lotes nas máquinas.
- Os tempos de movimentação dos lotes são ignorados.
- Um lote não pode ser fraccionado.

O problema geral da utilização de regras de despacho consiste em programar n tarefas para m máquinas, ou, se as tarefas forem decomponíveis em operações, em programar p operações para m máquinas. O grau de complexidade aumenta se a tarefa além de decomponível em várias operações for decomponível em vários lotes susceptíveis de processamento em várias máquinas, e, se além disso forem consideradas várias sequências alternativas, para o mesmo produto, e se o tempo de preparação (setup) variar com a sequência de operações.

(*) Eng.º Mecânico, Director-Geral de Datinvest, Docente no ISCTE

O equacionamento desta questão pode fazer-se através de uma abordagem de complexidade crescente, considerando as seguintes fases:

- 1) Várias tarefas para 1 máquina;
- 2) Várias tarefas para várias máquinas;
- 3) Várias tarefas com várias operações(1) para várias máquinas;
 - a) As sequências de operações são pré-definidas dentro de cada tarefa;
 - b) Cada operação só pode realizar-se num só posto de trabalho;
 - c) A sequência de operações não tem incidência no tempo de preparação;
 - d) Os tempos de movimentação são ignorados;
 - e) Os lotes não podem ser fraccionados.

A partir deste Modelo de Base o problema ganha maior complexidade:

- 3.1) Se forem considerados os tempos de movimentação e de preparação;
- 3.2) Se as sequências não forem pré-definidas, podendo cada operação ser realizada em diferentes postos de trabalho, com optimização da sequência;
- 3.3) Se existir uma definição de tempos de preparação com eventual melhoria das sequências devidas a setup;
- 3.4) Se a tarefa consistir num lote de produtos e puder ser fraccionada.

Seetharma L. Narasimhan e Srikant Panwalkar (1984), ensaiaram várias regras de despacho heurísticas aplicadas a um problema e concluíram que a configuração do problema (estrutura de fábrica, mix de produção, existência de inventários, número de máquinas), bem como o objectivo (e portanto o critério) em causa, origina diferentes desempenhos – o que conduz a que as regras sejam aprofundadas em consonância com a configuração do problema e com os objectivos. Assim, na programação da produção com regras de despacho a sequência a utilizar é:

- A) Selecção de um ou vários objectivos;
- B) Desenvolvimento de uma heurística;
- C) Avaliação do procedimento.

No desenvolvimento da heurística, são fundamentalmente utilizados 2 tipos de regras:

- Estáticas

- Dinâmicas

• As regras estáticas são decididas «a priori», e aplicadas no início do período, originando períodos fixados. Pressupõem normalmente que o conjunto de tarefas a programar chegam em simultâneo e que esse conjunto não varia até estar concluído;

• As regras dinâmicas são tomadas em tempo real, à medida que as coisas acontecem (têm normalmente melhor desempenho que as estáticas).

Quer se utilizem regras estáticas ou dinâmicas, o recurso a simuladores permite configurar numerosos cenários, cuja comparação é de grande utilidade.

Por outro lado, o tipo de implantação (layout), tem também uma grande influência na programação e no desempenho. Por exemplo, a produção intermitente com layout funcional (por tipo de máquina) ao ser substituída por células do fabrico flexíveis, leva a grandes alterações e melhorias substanciais.

Farzad Mahmood et al (1990) aplicou regras de despacho a um sistema de produção organizado por células de fabrico flexíveis, tendo esta forma de implantação conduzido a:

- Redução do tempo de preparação, uma vez que as células passam a processar componentes semelhantes;
- Redução da dimensão dos lotes;
- Redução do trabalho em curso;
- Redução da movimentação;
- Melhoria de relações entre pessoas.

No entanto, este tipo de layout acarretou como desvantagens a redução da utilização das máquinas e a redução da flexibilidade, podendo no entanto este tipo de dificuldades ser minimizado através de uma utilização de regras de programação heurísticas em 2 fases:

1.^a Fase – Dentro de cada grupo homogéneo

2.^a Fase – Entre grupos homogéneos

(Este problema envolve já alguma complexidade pois considera conjuntamente sequências alternativas, carregamento e despacho.)

2. Regras de despacho básicas

2.1 Objectivos e regras

As regras de despacho mais utilizadas para definição de prioridades têm como objectivos, o prazo de entrega, a margem e os tempos de processamento.

- A) Data de entrega ao mais cedo
(O objectivo é o prazo de entrega)

[EDD-(Earlier due date)] É primeira a tarefa com menor prazo de entrega.

$$EDD = Z_i(t) = d_i - t$$

- B) Margem
(O objectivo é o prazo de entrega através do respeito das margens)

[SLACK] – É primeira a tarefa com menor margem

$$Z_i(t) = S_i(t) = d_i - t - RP_i(t)$$



B1) Margem Estática

SSLACK* – É primeira a tarefa com menor margem estática

$$Z_i(t) = S_i = d_i - R_{i1} - TP_i$$

C) Rácio crítico

(Primordialmente baseado no prazo de entrega)

$$CR = \frac{SLACK}{RP_i(t)}$$

D) Tempo de processamento remanescente mais curto
(Objectivo: minimizar o tempo de processamento)

SRPT – É primeiro a tarefa com menor tempo de processamento remanescente.

$$Z_i(t) = RP_i(t)$$

D1) Tempo de processamento mais curto

SPT* – É primeira a tarefa com menor tempo de processamento.

$$Z_i(t) = TP_i$$

As regras A e B são pouco aplicáveis quando existem tempos de espera curtos entre operações.

As regras A, B, e C cumprem razoavelmente bem com níveis de carga baixos, mas deixam de servir quando a carga de trabalho sobe drasticamente.

A regra D funciona melhor em ambientes congestionados porque quanto mais depressa for concluída a tarefa, mais rapidamente outros lotes poderão entrar. A consequência é uma alta taxa de utilização. Centros operatórios com elevados congestionamentos devem incluir esta regra na sua programação, que no entanto falha para níveis de carga pouco elevados.

À medida que as datas de entrega se tornam mais difíceis de atingir, os critérios baseados na data de entrega tornam-se mais eficientes, uma vez que a média de tempo de fluxo diminui.

Terminologia

* – Regra Estática

T – Horizonte de programação

t – momento de decisão

n – N.º de tarefas na oficina

i – N.º de ordem da tarefa

j – N.º de ordem da operação

j(t) – Operação iminente

P_{ij} – Tempo de processamento para a operação j da tarefa i

TO_i – N.º total de operações da tarefa i

TP_i – Tempo total de processamento da tarefa i

RO_i(t) – N.º de operações remanescentes da tarefa i

RP_i(t) – Tempo de processamento remanescente da tarefa i

R_{ij} – Momento em que a tarefa i fica disponível para a operação j

R_{i1} – Momento em que a tarefa i chega à oficina

C_i – Momento em que a tarefa i é concluída e deixa o sistema

d_i – Data de entrega da tarefa i

L_i – Estado de avanço da tarefa L_i = C_i - d_i

T_i – Atraso da tarefa T_i = máx. (0, L_i)

F_i – Tempo de fluxo = C_i - R_{i1}

S_i(t) – Margem = d_i - t - RP_i(t)

S_i – Margem Estática = d_i - R_{i1} - TP_i

N_{ij}(t) – Conjunto de tarefas na fila de espera no momento da operação j da tarefa i

Z_i(t) – Prioridade da tarefa i no momento t

2.2 Avaliação do desempenho das regras

O desempenho das regras de despacho depende não só do critério escolhido, mas também do tipo do sistema de produção.

Assim, a avaliação da regra decorre sempre do grau de concretização dos objectivos. Avaliações de desempenho possíveis são o atraso das tarefas, a percentagem de tarefas atrasadas, o tempo médio de espera, o tempo de fluxo, a utilização média das máquinas, a utilização de stocks de segurança, o tempo de preparação...

Destes critérios, os mais utilizados são:

- Eficiência de utilização (tempo médio de fluxo) – (a minimizar);
- Atraso médio das Tarefas (a minimizar);
- Percentagem de tarefas atrasadas (a minimizar).

CASO – Para melhor visualização do problema, veja-se o exemplo seguinte envolvendo 4 tarefas a ser processadas numa única máquina (a complexidade do problema cresce com o número de máquinas consideradas).

Tarefas	Prazo entrega	Tempos de processamento	Margem	CR
A	4 dias	3	1	0.33
B	9 dias	6	3	0.5
C	5 dias	5	0	0
D	9 dias	7	2	0.28

a) Imputação segundo a regra da Data de Entrega

Tarefas	Tempo fluxo	Tempo até à data de entrega	Atraso	Nº tarefas atrasadas
A	3	4	–	
C	8	5	– 3	
B	14	9	– 5	
D	21	9	– 12	
Σ	46		Σ 20	3

Tempo Médio Fluxo $46 : 4 = 11,5$	Tempo Médio Atraso $20 : 4 = 5$	% Tarefas Atrasadas 75%
---	---------------------------------------	-------------------------------

b) Imputação segundo Regra de Margem

Tarefas	Tempo fluxo	Tempo até à data de entrega	Atraso	Nº tarefas atrasadas
C	5	4	– 1	
A	8	5	– 3	
D	15	9	– 6	
B	21	9	– 12	
Σ	49		Σ 22	4

Tempo Médio Fluxo $49 : 4 = 12,25$	Tempo Médio Atraso $22 : 4 = 5,5$	% Tarefas Atrasadas 100%
--	---	--------------------------------

c) Imputação segundo Regra do Rácio CR

Tarefas	Tempo fluxo	Tempo até à data de entrega	Atraso	Nº tarefas atrasadas
C	5	5	–	
D	12	9	– 3	
A	15	4	– 11	
B	21	9	– 12	
Σ	53		Σ 25	3

Tempo Médio Fluxo $53 : 4 = 13,25$	Tempo Médio Atraso $25 : 4 = 6,25$	% Tarefas Atrasadas 75%
--	--	-------------------------------

d) Imputação segundo regra do Menor Tempo de Processamento

Tarefas	Tempo fluxo	Tempo até à data de entrega	Atraso	Nº tarefas atrasadas
A	3	4	–	
C	8	5	– 3	
B	14	9	– 5	
D	21	9	– 12	
Σ	46		Σ 20	3

Tempo Médio Fluxo $46 : 4 = 11,5$	Tempo Médio Atraso $20 : 4 = 5$	% Tarefas Atrasadas 75%
---	---------------------------------------	-------------------------------

(Quando o critério é a minimização do tempo de fluxo – A regra SPT domina todas as outras.)

Hershaner e Ebert (1975) testaram várias regras em vários ambientes, tendo concluído que:

a) Regras baseadas em tempo de processamento minimizam normalmente o tempo de fluxo;

b) Regras baseadas no prazo de entrega cumprem normalmente no que diz respeito ao custo.

Blackstone (1982) verificou que as regras baseadas no tempo de processamento cumprem melhor quando não existe controlo sobre as datas de entrega e os prazos começam a deslizar.

Montarezi et al (1990) utilizando como critérios, nomeadamente o tempo médio de espera, a utilização média das máquinas e a utilização média de stocks de segurança verificaram que:

a) Em termos de tempo médio de espera cumprem melhor as regras baseadas no menor tempo de processamento;

b) Em termos de tempo médio de utilização das máquinas cumprem melhor as regras baseadas no maior tempo de processamento de máquina;

c) Em termos de utilização de stock de segurança cumprem melhor as regras baseadas no menor tempo de processamento;

d) Nenhuma regra simples tem melhor comportamento segundo todos os critérios.

2.3 A taxa de escoamento (RUN OUT)

A taxa de escoamento pode considerar-se um caso particular da regra do Rácio Crítico = CR

$$= \frac{\text{Margem}}{\text{Tempo Processamento Remanescente}}$$

O método utiliza-se quando existem vários lotes produzidos na mesma linha de produção, que têm de ser sequenciados.

Assume-se que se está a produzir para stock, considerando-se os níveis de stock existentes e as taxas de procura futuras, sendo os tempos de preparação baixos.

Para cada lote i a taxa de escoamento

$$r_i = \frac{I_i}{d_i} = \frac{\text{n.º de unidades disponíveis}}{\text{n.º unid. consumidas por semana}}$$

O menor r_i é o primeiro a ser produzido.

É portanto um processo dinâmico heurístico que se adapta continuamente à mudança de condições, que não toma em conta:

– Custos de posse, custos de ruptura, variações na procura.

Um exemplo:

Produto	Stock	Produção un/semana	Procura un/semana	Dimensão do lote
A	10 000	3 000	1 000	1 000
B	6 000	1 500	500	1 500
C	9 000	4 000	2 000	2 000

Produto	T.E.	T. Prod. Lote
A	10	0,33
B	12	1,00
C	4,5	0,50

SEMANA 0.5			SEMANA 1.0		SEMANA 1.5		SEMANA 2	
Prod.	Stock	T.E.	Stock	T.E.	Stock	T.E.	Stock	T.E.
A	9 500	9.5	9 000	9.0	8 500	8.5	8 000	8.0
B	5 750	11.5	5 500	11.0	5 250	10.5	5 000	10.0
C	10 000	5.0	11 000	5.5	12 000	6.0	13 000	6.5

SEMANA 2.5			SEMANA 3.0		SEMANA 3.33		SEMANA 3.83	
A	7 500	7.5	7 000	7.0	7 667	7.67	7 167	7.17
B	4 750	9.5	4 500	9.0	4 333	8.67	4 083	8.17
C	14 000	7.0	15 000	7.5	14.333	7.17	15 333	7.67

SEMANA 4.17			SEMANA 4.67		SEMANA 5.67		SEMANA 6.00	
A	7 834	7.84	7 334	7.34	6 334	6.34	7 000	7.00
B	3 917	7.84	3 667	7.34	4 667	9.33	4 500	9.00
C	14 667	7.33	15 667	7.84	13 667	6.83	13 000	6.50

Método do «Período de concretização»

É uma variante da Taxa de Escoamento que pode ser utilizado para determinar o tempo de produção de um grupo de itens que é produzido utilizando o mesmo conjunto de equipamentos.

Período de concretização é o período de tempo para qual a produção previamente programada adicionada dos stocks em posse, satisfaz a procura de um dado produto.

O objectivo básico deste método é utilizar a capacidade de produção de forma que o período de concretização ao mais cedo para todos os itens seja o mesmo. Os esforços da produção são assim equilibrados pelos vários itens, mais do que concentrados em alguns deles.

Ilustremos o procedimento através de um exemplo:

Sabendo-se que existem 96,5 horas máquinas disponíveis durante a semana para um conjunto de produtos (A a F), começa-se por determinar os stocks em posse de cada produto (coluna 1.1) e o

tempo de produção por unidade (coluna 1.2), donde decorre o stock em posse em horas máquina ($1.3 = 1.1 \times 1.2$).

Por outro lado são determinadas as previsões de consumo (coluna 1.4) e por cálculo o consumo semanal previsto em horas máquina ($1.5 = 1.2 \times 1.4$).

No caso do exemplo, os stocks em posse (de A a F) em horas máquina representa 213.95; a capacidade disponível em horas máquina por semana é de 96,5; e as previsões de consumo são de 83,54 em horas máquina.

O período de concretização agregado é dado por:

$$\frac{\text{Horas de máquina} \\ 1.3 + \text{disponíveis} - 1.5}{1.5} = \frac{213.95 + 96.5 - 83.54}{83.54}$$

= 2,72 semanas

Que é o período para o qual a produção previamente programada, mais os stocks em posse satisfazem a procura agregada dos produtos.

Produto	1.1 Stocks em posse (unidades)	1.2 Tempo de produção (Horas M.O. p/ unidade)	1.3 Stocks em posse Hora/Máq. (1.1) x (1.2)	1.4 Previsões de consumo semanal (unidades)	1.5 Consumo semanal previsto Horas/Máq. (1.2) x (1.4)
A	125	0.2	25	60	12
B	250	0.08	20	85	6.8
C	75	0.5	37.5	30	15
D	300	0.09	27	96	8.64
E	239	0.15	35.85	78	11.7
F	98	0.7	68.6	42	29.4
Totais agregados			213.95		83.54

O período de tempo agregado serve de seguida para determinar a produção de cada produto, se cada um tiver um período de tempo de concretização de 2,72.

Caso do Produto A:

Coluna 2.1

• (Previsão do Consumo Semanal (60) x 2.72) = 163 = Previsão de Consumo no Período de Concretização

Coluna 2.2

• Stock Final Desejado (163) + Previsão de Consumo Semanal (160) = 223 = Total de Stocks Necessário

Coluna 2.3

• Stock Necessário (223) + Stock Existente (125) = 98 = Programação da Produção

Coluna 2.4

• Produção na Semana (Horas Máquina) = 98 x 0,2 (Tempo de produção por unidade) = 19.6

ITEM	2.1 Previsão de consumo durante o período de concretização	2.2 Total de itens necessários (1.4) + (2.1)	2.3 Programado (Unidades) (2.2) – (1.1)	2.4 Produção (Horas Máquina) (2.3) x (1.2)
A	163	223	98	19.6
B	231	316	66	5.3
C	82	112	37	18.5
D	261	357	57	5.0
E	212	290	51	7.7
F	114	156	58	40.6
				± 96.5

3. Regras de despacho complexas

3.1 Mais objectivos – mais regras

Como anteriormente exposto, as regras são escolhidas em função dos objectivos e do contexto, por exemplo, as regras de sequenciamento que cumprem bem em regime intermitente, não são necessariamente apropriadas para uma linha de montagem. Assim, a partir da diversidade de objectivos possíveis, há uma grande diversidade de regras, algumas das quais são de seguida apresentadas.

- Margem por Operações Remanescentes
[S/ROP] - Slack per Remaining Operation
A maior prioridade é dada à operação em espera com o máximo rácio: margem (folga) por operação residual
 $Z_i(t) = S_i(t) / RO_i(t)$

- Margem por Volume de Trabalho Remanescente
[S/RPT] - Slack per Remaining Work
A maior prioridade é dada à operação em espera com o máximo rácio: margem por volume de trabalho residual
 $Z_i(t) = S_i(t) / RP_i(t)$

- Primeiro a Chegar
[FIFO] - First in, First out
É primeira a primeira tarefa a chegar
 $Z_i(t) = R_{ij}$

- Minimização do tempo total de processamento
MAKESPAN (Somatório dos tempos de processamento)
 $Z_i(t) = \min \sum TP_i$

- Tempo de Preparação mais Curto

[SST] – Shortest Setup time

É primeira a tarefa com tempo de preparação mais curto.

- Somatório de Sequência dos Tempos de Preparação

[TSS] – Travelling Salesman Sequence

Seleccionar a tarefa com um mais baixo tempo de sequência de somatórios de tempos de preparação.

- Tarefas em Fila de Espera

[JIQ] – Jobs in the Queue

É a primeira a fila de espera com mais tarefas em espera

- Maior Valor Acrescentado – É primeira a tarefa com maior valor acrescentado

- Sorteio – A primeira tarefa é definida por sorteio

- N.º Operações – É primeira a tarefa com menor n.º de operações a executar

- Custo Espera/Duração da Operação – É primeira a tarefa com o mais baixo rácio
Custo de Espera/Duração da Operação

- Minimização do tempo de fluxo total – Tempo de Espera + Tempo de Preparação + Tempo de Movimentação + Tempo de Operação

- Minimização dos inventários médios em curso

3.2 Novas regras combinadas

Quando na programação é utilizada uma só regra heurística, existe rigidez de objectivo, pelo que, podem existir vantagens em utilizar de uma forma flexível mais do que uma regra em simultâneo. Por exemplo, quando se utiliza unicamente a regra SPT com o objectivo de minimizar o tempo de fluxo, pode não se estar a cumprir o objectivo – datas de entrega.

Assim, em virtude da relativa ineficiência destas leis simples, atrás enunciadas, vários autores têm tentado desenvolver leis combinadas. Carrol (1965) desenvolveu uma regra designada COVERT que combina entre si as regras da Margem (SLACK) e do menor tempo de processamento (SPT) – Segundo esta regra, a prioridade é estabelecida a partir de um rácio de custo incremental de atraso:

$$C_i = \frac{hw - S_i(T)}{hw}$$

C_i – custo convertido

hw – tempo de processamento e espera expectável
 $S_i(T)$ – Margem

Esta expressão exige a estimativa de parâmetros complexos, pelo que, Bertrand (1982) e Schultz (1989) desenvolveram uma regra de prioridade não parametrizada a MDD (Modified Due Date). Segundo esta regra, uma tarefa é imputada segundo a EDD sempre que o prazo de entrega pode ser cumprido, e quando isso não acontece através de SPT.

Anderson e Nyirenda (Two New Rules to Minimize Tardiness in a Job Shop – 1992) partindo dos prazos de entrega aplicados a operações induziram uma nova regra, que resulta da combinação das regras S/RPT + SPT em que:

$$d_{ik}^* = \max \left[\frac{S_i(t)}{TP_i}, TP_i \right]$$

Sendo o tempo residual de trabalho igual ao somatório do tempo de todas as operações ainda não realizadas. Em diferentes situações é esco-

lhido o S/RPT ou o SPT. Sendo a tarefa com mais baixo d_{ik}^* , a escolhida.

Tendo estudado o comportamento deste critério, estes autores verificaram que:

- Quando o objectivo é o da minimização do atraso em relação à data de entrega o melhor resultado é conseguido através da regra S/RPT + SPT.
- Quando o objectivo é o da minimização da percentagem de tarefas em atraso, a regra SPT simples funciona melhor para datas de entrega muito apertadas. Mas a regra S/RPT + SPT funciona melhor para prazos de entrega dilatados.

J. Chandra e J. Talavage (1991) apresentaram uma outra alternativa baseada numa estratégia de regras de despacho dinâmica e adaptativa, considerando que uma só regra de despacho em todas as circunstâncias é muito limitada. Esta estratégia ganha maior validade quanto se verifica que o routing das tarefas é flexível, e quando os tempos de operação e as datas de entrega deixam de poder ser cumpridos originando estados diferenciados em relação ao estado original.

Sendo esta metodologia para tomar decisões, são estruturados 4 elementos base de informação:

- Nível de Congestionamento – Avaliado pela propensão a acumular trabalhos em curso;
- Preferência de uma Operação por uma Máquina – Avaliada pelo tempo de processamento em relação a outras máquinas;
- Nível Crítico de uma Operação – Avaliado em termos de margem disponível em relação ao prazo de entrega;
- Objectivo da Programação
 - Maximizar a taxa de progressão do trabalho *Work Prog*
 - Minimizar o atraso das operações *Prev. Tard* (Em cada momento considera-se apenas um objectivo).

A preferência é dada a *Work Prog.*, carregando as operações nas máquinas tão frequentemente quanto possível (Mctold – Machine to Load). No entanto, sempre que existe uma ou mais tarefas

críticas (em relação ao prazo de entrega) escolhem-se essas tarefas (Jtold Job to Load).

3.3 IR

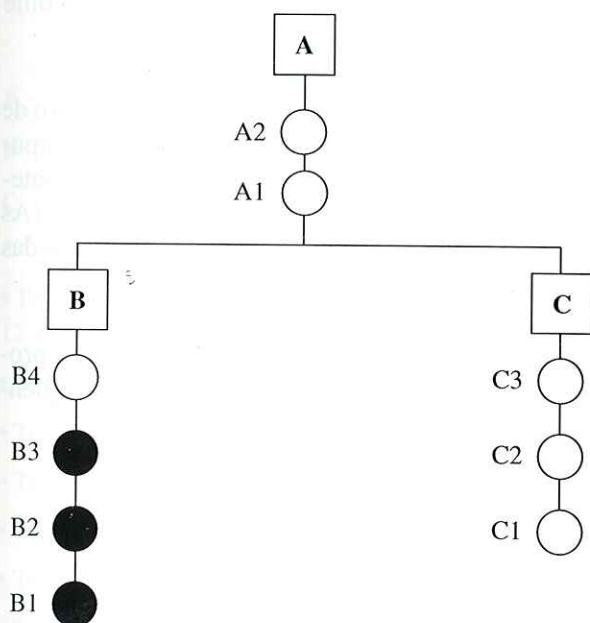
Patrick Philipoom et al (1990) introduziu um novo rácio IR – Important Racio, a seguir definido, que é na sua aplicação combinado com outros rácios relevantes, sendo a sua performance avaliada em função do tempo médio de fluxo, do atraso, e dos stocks em posse.

Para definir o IR, Philipoom recorreu à estrutura do produto, o que parece ter grande impacto na performance de uma linha de montagem.

A estrutura do produto considera o número de componentes da BOM, níveis da BOM, e, a extensão do caminho que um produto segue (n.º de operações) – Tarefas com muitos níveis de montagem devem considerar o caminho crítico como o primeiro a ser realizado. (De referir no entanto que existem 3 tipos de estrutura base).

Assim, o rácio IR considera a estrutura do produto, e o sequenciamento de tarefas por identificação do seu estado crítico em termos de prazo.

O fabrico de B começa em B1



Os círculos a cheio representam operações já realizadas

Máximo n.º de operações pelo Caminho B = 6

Faltam 3 operações para concluir

IR = 3/6

Máximo n.º de operações pelo Caminho C = 5

Faltam 5 operações para concluir

$$IR = \frac{5}{5}$$

A escolha recai sobre o Máx. IR em cada momento de decisão, sendo no entanto combinado com outros rácios, dois a dois, funcionando o 1.º rácio num dado espaço e o 2.º em caso de empate do 1.º, ou quando este deixa de ser aplicável.

Foram realizadas nomeadamente as seguintes combinações:

RPT – RROP	Total remaining work – relative number of remaining operations
RPT – FIFO	Total work remaining – first in first out
RPT – IR	Total work remaining – important racio
IR – S/ROP	Important racio – Remaining operations
IR – RPT	Important racio – Total work remaining
SLACK – IR	Slack – Important racio
SLACK – TWK	Slack – Total work

RESULTADOS

• A combinação do RPT com outras regras RPT-RROP, RPT-IR, não melhoram a performance – O que é de alguma forma surpreendente pois a RPT simples não considera a estrutura do produto.

• As regras combinadas de Margem SLACK-IR e SLACK-RPT não oferecem grandes vantagens em relação à regra mais simples do mesmo tipo S/ROP.

• No entanto as regras IR apresentam resultados francamente melhores para todos os tipos de estrutura e segundo todos os tipos de critério.

Donde resulta que IR – RPT (em particular) é uma regra particularmente apropriada para tarefas com muitos níveis de montagem.

No entanto qualquer regra deve sempre ser testada sob um conjunto variado de circunstâncias, condições de trabalho e estruturas de produto.

3.4 VLA

Stephen Lawrence (1990) desenvolveu regras baseadas numa análise de custos marginais, com o objectivo de maximizar o valor líquido actual. Foram considerados não só os custos afectando a programação da produção por via dos custos inerentes ao atraso na entrega (penalidades, custo das vendas, goodwill), mas também os custos inerentes a uma entrega antecipada (custos de posse e de oportunidade). Este modelo baseia-se numa identificação dos cash flows afectados pelas decisões de programação – O VLA é maximizado quando todos os trabalhos podem ser entregues exactamente nas suas datas de entrega.

Foram criadas 2 regras – A primeira referente à realização do trabalho RDE (Release at delay equilibrium) especifica que se a tarefa é realizada em $t < B_i$ são suportados custos desnecessários de atraso. A segunda referente ao despacho designada por MPT (Marginal tardiness penalty) permite escolher para processamento a tarefa com maior penalidade marginal de atraso.

Comparando este critério MTP.RDE com outros rácios combinados nomeadamente Covert/AQT (Average Queue Time) verificaram que o critério MTP maximiza o VLA mais do que qualquer outro critério.

4. Sequenciamento com Gant

O sequenciamento de tarefas é frequentemente realizado com GANT recorrendo à definição de ordem de processamento das tarefas e das máquinas em que se realizam essas tarefas usando os

gráficos de barras. Nesse referencial podem tomar-se em linha de conta os tempos médios de transporte, de espera e de preparação, ou ignorar esses factores.

Por outro lado, a programação pode ser estática ou dinâmica:

Na Programação Estática • As regras são decididas «a priori», e aplicadas no início do período, pressupondo normalmente que o conjunto de tarefas chegam em simultâneo, e que esse conjunto não varia até estar concluído.

Na Programação Dinâmica • As tarefas chegam continuamente e são continuamente introduzidas à medida que há disponibilidade, e • A sequência de tarefas é definida sempre que existe uma disponibilidade.

Este segundo caso coincide com um maior grau de complexidade.

A programação tem ainda 2 níveis de complexidade diferentes, consoante incida sobre tarefas ou sobre operações:

1 – Sequenciar tarefas constituídas por operações com sequências pré-definidas em que se escolhe a tarefa a imputar (segundo um qualquer critério de despacho) sendo realizadas todas as operações dessa tarefa – E só depois se escolhe outra tarefa.

2 – Sequenciar operações pré-definidas (dentro de cada tarefa) em que se escolhe a operação a imputar (segundo um qualquer critério) independentemente da tarefa a que pertence essa operação. (As operações são escolhidas independentemente das tarefas a que dizem respeito).

Por exemplo, no caso seguinte, pretendem programar-se 4 tarefas, cada uma delas com sequências pré-definidas:

Tarefa 1 – (Posto A-2H) → (Posto B-3H)
→ (Posto C-4H)

Tarefa 2 – (Posto C-6H) → (Posto A-4H)

Tarefa 3 – (Posto B-3H) → (Posto C-2H)
→ (Posto A-1H)

Tarefa 4 – (Posto C-4H) → (Posto B-3H)
→ (Posto A-3H)

Sendo a ordem das tarefas arbitrária e com as seguintes datas de entrega:

Tarefa 1 – 16 Horas
Tarefa 2 – 16 Horas
Tarefa 3 – 8 Horas
Tarefa 4 – 10 Horas

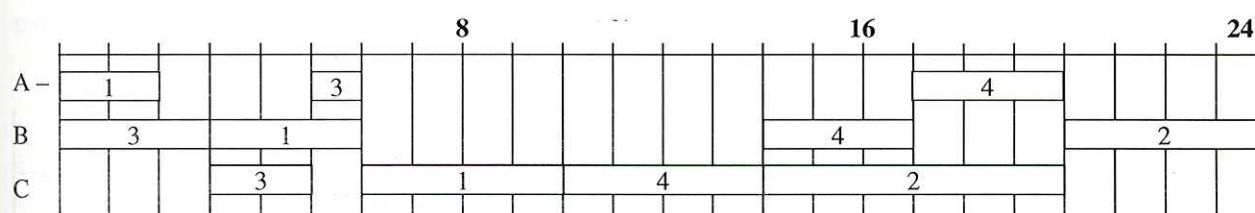
Não considerando os tempos de movimentação e preparação entre operações e tendo como objetivo a minimização dos tempos de processamento do conjunto das máquinas, é conveniente imputar segundo o critério SPT (começando pela tarefa

que cumpre menor tempo de operação, e sequentemente,...)

1 – Resolução Possível Incidindo sobre Tarefas à Luz do Critério SPT

Tarefa 1 – 9 Horas
Tarefa 2 – 10 Horas
Tarefa 3 – 6 Horas
Tarefa 4 – 10 Horas

Escolhe-se em primeiro lugar a Tarefa 3, depois a Tarefa 1, depois as restantes

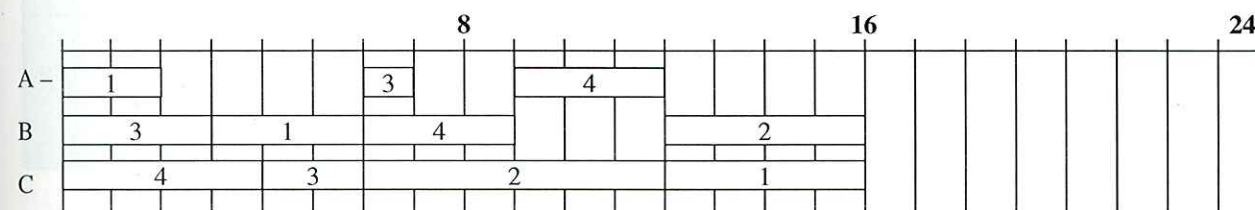


Tempo Total de Processamento (MAKESPAN)... 20 H + 24 H + 20 H = 64 H
Tempo Total Perdido na Máquina A... 14H
Tempo Total Perdido na Máquina B... 11 H
Tempo Total Perdido na Máquina C... 4 H
Tempo Total Perdido na Tarefa 1... 0 H
Tempo Total Perdido na Tarefa 2... 14 H
Tempo Total Perdido na Tarefa 3... 0 H
Tempo Total Perdido na Tarefa 4... 10 H

A tarefa 2 atrasa 8 H em relação ao prazo de entrega.
A tarefa 4 atrasa 10 H em relação ao prazo de entrega.

Perante tão mau desempenho haveria que tentar qualquer outro critério e alternativas de sequência.

2 – Resolução Possível incidindo sobre operações desligadas das tarefas em que estão integradas



- Tempo Total de Processamento (Makespan) 12 + 16 + 16 = 44 H
- Tempo Total Perdido Máq. A... 6 Horas
- Tempo Total Perdido Máq. B... 3 Horas
- Tempo Total Perdido Máq. C... 0 Horas
- Tempo Perdido pela Tarefa 1... 7 Horas
- Tempo Perdido pela Tarefa 2... 6 Horas
- Tempo Perdido pela Tarefa 3... 1 Hora
- Tempo Perdido pela Tarefa 4... 2 Horas

• Apenas a Tarefa 4 se atrasou em relação ao prazo de entrega imposto.

Poderiam de seguida ser introduzidos tempos de espera e movimentação.

4.1 Algoritmo de Johnson (2 x 2)

Neste algoritmo, a tarefa a programar é escolhida segundo o mais baixo tempo de processamento e de seguida imputada, sendo colocado um gráfico à direita ou à esquerda, decorrente de pertencer à Máquina 1 (Esq.) ou à Máquina 2 (Dir).

Por exemplo:

Tarefa	Tempo de Processamento	
	Máquina 1	Máquina 2
A	3	2
B	3	5
C	5	4
D	4	3

1.º Tarefa A: duração 2 → Máquina 2 portanto é colocada à Direita em Baixo

2.º Tarefa B: duração 3 → Máquina 1 colocada à Esquerda em Cima

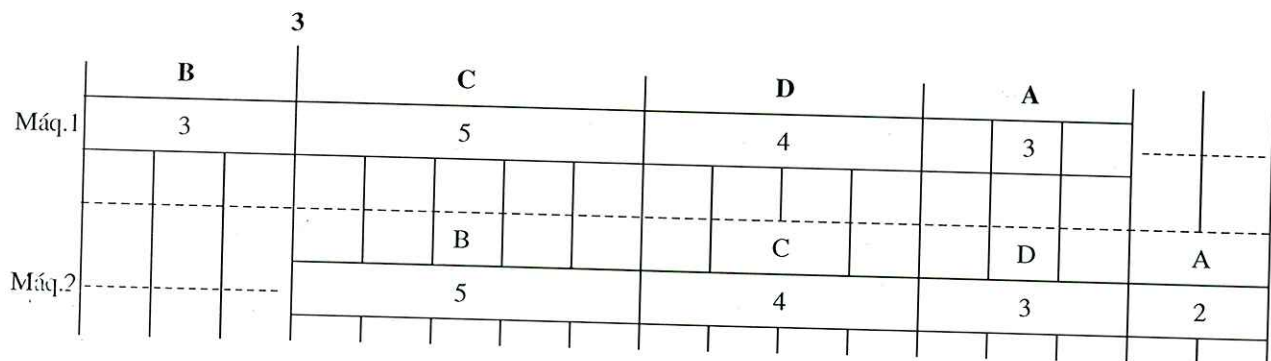
3.º Tarefa D: duração 3 → Máquina 2 colocada à Direita em Baixo

4.º Tarefa C: duração 4 → Máquina 2 colocada à Direita em Cima

Uma vez que os tempos mais curtos no início estão na Máquina 1 à esquerda, a Máquina 2 provavelmente com pequenas alterações não vai ter de esperar para começar a trabalhar.

No final da sequência a situação reverte-se e os tempos mais curtos estão na Máquina 2.

Sequência B → C → D → A



4.2 Método de Giffler e Thompson

O procedimento de «Giffler e Thompson» para a programação resolve-se enfrentando um a um todos os conflitos (de entrada na máquina) de todas as formas possíveis, a partir de uma solução do programa inicial.

Veja-se o seguinte caso:

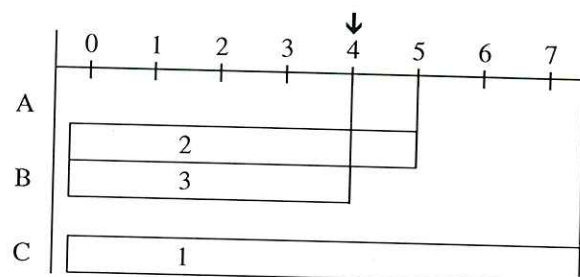
Ordem de Processamento de Tarefas nas Máquinas

Tarefa	Sequência de procedimento		
	1	2	3
1	C	A	B
2	B	C	A
3	B	A	C

Tempo de Operação para Cada Tarefa em Cada Máquina

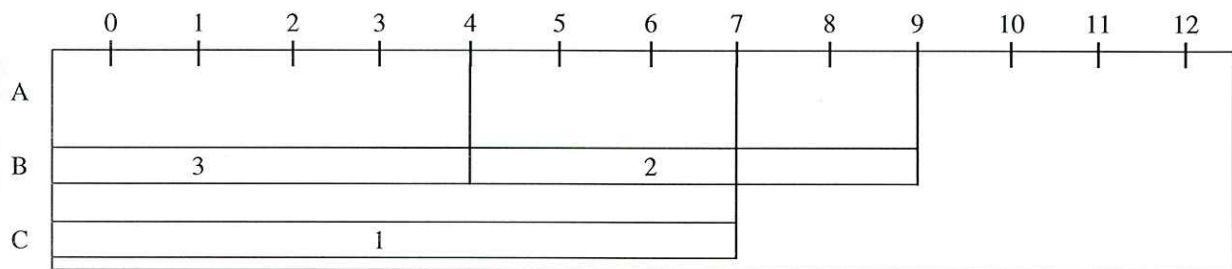
Tarefa	Máquina		
	A	B	C
1	4	2	7
2	3	5	6
3	2	4	3

Considere-se a primeira operação a ser realizada em cada uma das três máquinas e introduza-se isso no quadro da figura seguinte – verifica-se que existe um conflito entre as tarefas 2 e 3 para a máquina B.



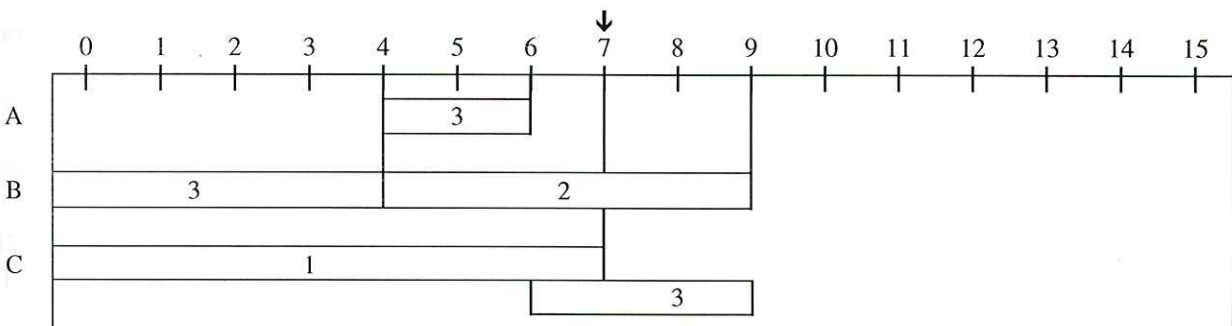
A seta assinala o datum que define o ponto que especifica o tempo ao mais cedo em que numa decisão de imputar a próxima tarefa pode ser tomada (neste caso 4). O datum vai sendo suces-

sivamente deslocado à medida que se tomam decisões. Resolve-se o conflito entre as tarefas 2 e 3 escolhendo arbitrariamente 3.



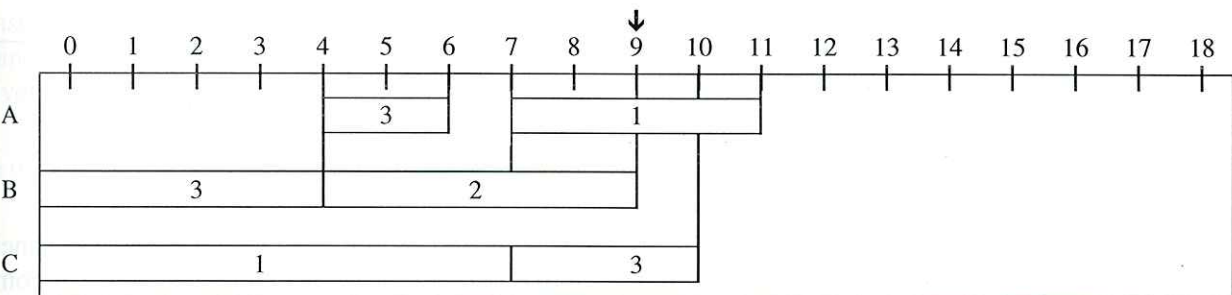
Uma vez concluída a tarefa 3 na Máquina B, pode ser transferida para a Máquina A e o datum passa para 6.

Se 3 pode ser concluída ao tempo 6 pode ser transferida para C, em que surge um conflito entre 1 e 3 – Neste momento o datum é 7.

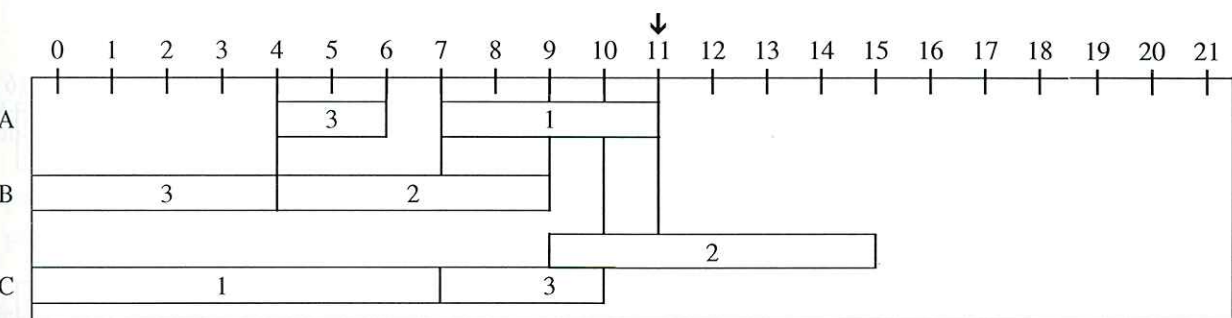


Opte-se por 1 para ser processado em primeiro lugar em C. Logo que 1 esteja concluído em C

pode ser transferido para A. Neste momento o datum é 9 correspondente à tarefa 2.



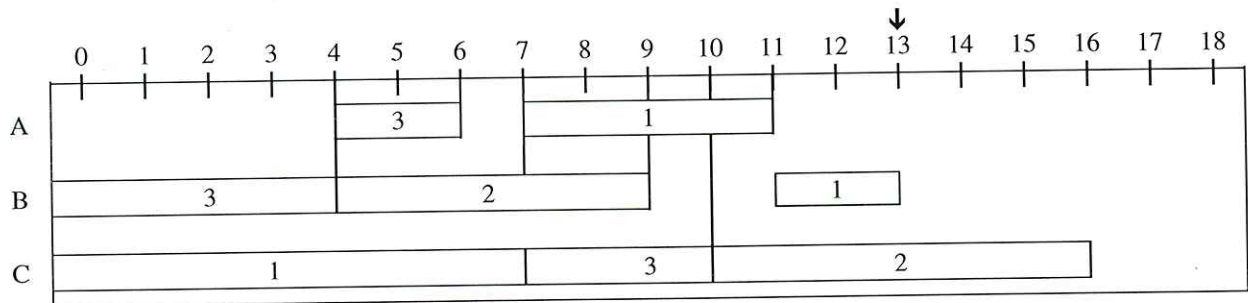
Se a tarefa 2 for completada ao tempo 9 em B pode ser transferida para C, e, o seu tempo de conclusão será 15.



Existe agora um conflito entre 3 e 2 na máquina C.

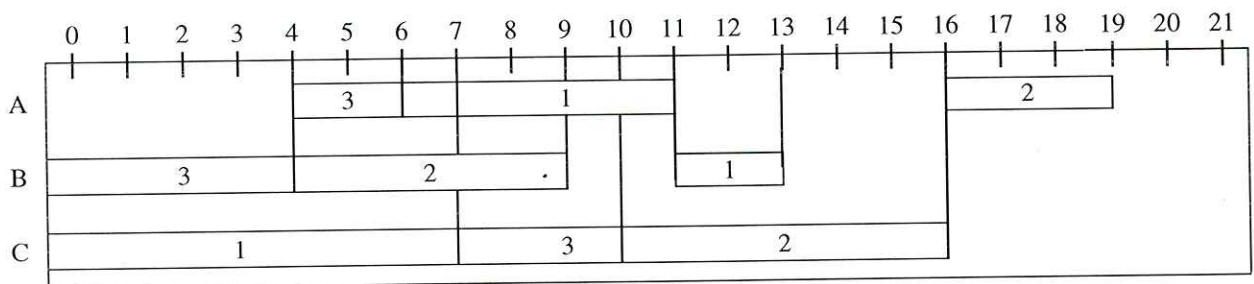
Escolhamos 3.

O datum agora é 11 para 1, de onde decorre que após concluído em A ao tempo 11 pode ser transferido para B.



O datum é agora 13 para 1, mas esta foi a última operação de 1, passemos portanto ao próximo ponto que é 16 para 2.

Logo que 2 esteja concluído em C transita para a máquina A e o seu tempo de conclusão é 19.



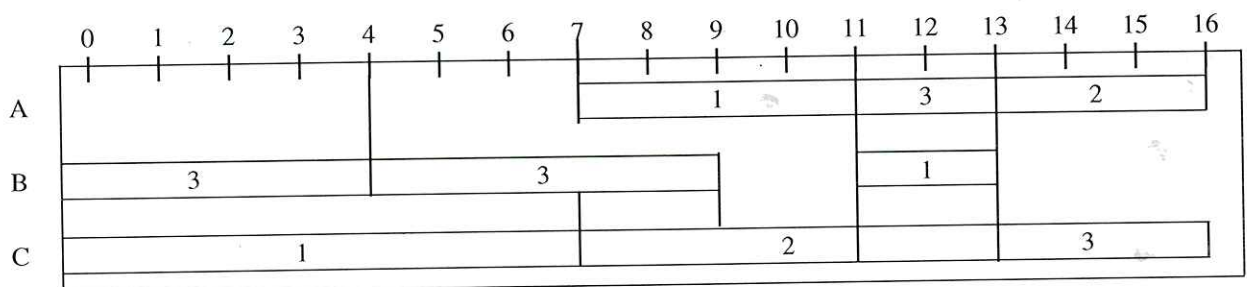
O programa está concluído, e o tempo total de realização é 19.

O programa a que se chegou é um programa viável que resultou das seguintes opções:

Conflito entre 2 e 3 em B a favor de 3
Conflito entre 1 e 3 em C a favor de 1

Conflito entre 3 e 2 em C a favor de 3

Resolvendo os conflitos de várias outras maneiras possíveis (utilizando nomeadamente o computador), atingem-se outros resultados nomeadamente tomando diferentes opções aquando dos conflitos. Uma possibilidade alternativa é apresentada.



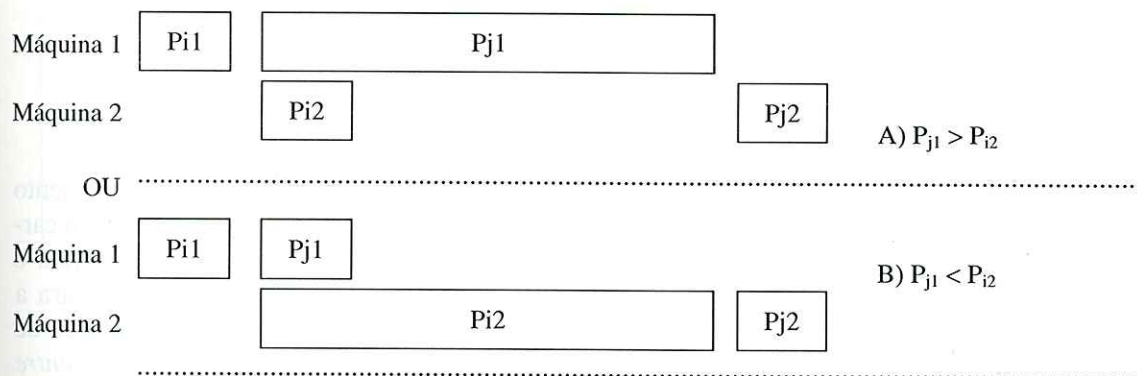
4.2.1 Outros algoritmos de sequenciamento

O sequenciamento utilizando algoritmos tem sido trabalhado por vários autores, de que se destacam Campbell et al (1970) e Gupta (1972); Estes algoritmos são no entanto difíceis de computar em tempo real pela morosidade que envolvem.

Mesmo quando se utiliza programação estática, numa combinação de 4 tarefas para 4 máquinas

em que a duração das operações é bem definida, o n.º de possibilidades de combinação de tarefas com máquinas é: $(4!)^4$.

– A função dos algoritmos é reduzir este tão grande leque de possibilidades, em que o caso mais simples é o de 2 tarefas – 2 máquinas em que apenas existem 4 possibilidades; e, se a sequência for prédefinida 4 possibilidades; e se a sequência for pré-definida 2 possibilidades.



Ding-Yu Chan e David Bedworth desenvolveram um algoritmo de sequenciamento que permite reduzir o n.º de interações de forma a minimizar o tempo de fluxo total. Para a dedução parte do caso mais simples da figura anterior, determinando os tempos de fluxo dos pares de arcos possíveis:

$$F_2(ij) = 2 P_{i1} + P_{i2} + P_{j2} + \underbrace{\text{Max}(P_{j1}, P_{i2})}_{R_2}$$

e

$$F_2(ji) = 2 P_{j1} + P_{j2} + P_{i2} + \underbrace{\text{Max}(P_{i1}, P_{j2})}_{R_2}$$

E, eliminando por simplificação os membros comuns a ambos os arcos

$$\begin{cases} F^*_2(ij) = 2 P_{ij} + \text{Max}(P_{j1}, P_{i2}) \\ F^*_2(ji) = 2 P_{ji} + \text{Max}(P_{i1}, P_{j2}) \end{cases}$$

Generaliza de seguida para um n.º mais elevado de Máquinas e Tarefas:

$$\begin{cases} F^*_{m(ij)} = 2 P_{i1} + \sum_{k=2}^{m-1} P_{ik} + R^*_m \\ R^*_m = \text{Max}(R_{m-1}, \sum_{k=2}^m P_{ik}) \end{cases}$$

A metodologia assenta na resolução de conflitos entre pares comparando tempos de fluxo mínimos, com base nos quais é decidida a sequência, sendo primeiramente sequenciadas as tarefas com maior n.º de pares óptimos.

Por exemplo, para uma situação de 4 tarefas x 4 Máquinas, com Tempos de Processamento indicados no quadro:

Máq. Tarefas	1	2	3	4
1	22	11	19	21
2	9	14	16	2
3	20	19	4	2
4	10	18	6	7

$$F^*_4(1,2) = (2 \times 22) + (11 + 19) + 51 = 25$$

$$R^*_2 = \text{Máx}\{9, 11\} = 11$$

$$R^*_3 = \text{Máx}\{R_2, P_{12} + P_{13}\} = \text{Máx}\{11, 30\} = 30$$

$$R^*_4 = \text{Máx}\{R_3, P_{12} + P_{13} + P_{14}\} = \text{Máx}\{30, 51\} = 51$$

$$F^*_4(2,1) = (2 \times 9) + (14 + 16) + 52 = 115$$

Para o Conjunto de Pares

Pares de Sequências → Valores de Fluxos Temporários

(1-2) (2-1)	(125) (100)
(1-3) (3-1)	(125) (115)
(1-4) (4-1)	(125) (96)
(2-3) (3-2)	(91) (112)
(2-4) (4-2)	(86) (92)
(3-4) (4-3)	(106) (87)

Pares com Sequência Óptima

2* - 1	
3* - 1	
4* - 1	A tarefa mais frequentemente
2* - 3	ótima é a 2
2* - 4	
4* - 3	

A sequência a utilizar é: 2 → 4 → 3 → 1

5. Minimização do somatório dos custos de preparação

Quando o custo de preparação (setup) varia com a sequência do processo que se utiliza, existe todo o interesse em minimizar o somatório dos setups. Por exemplo, o custo de pintura de um dado produto é mais elevado se se realizar em primeiro lugar a pintura branca e depois a pintura preta, pois no 1.º caso tem de haver remoção da pintura.

A matriz da figura seguinte representa os custos de preparação de diferentes produtos quando se evolui de i para j.

	1	2	3	4	5
1	—	60	100	70	50
2	90	—	110	80	30
3	100	65	—	80	40
4	80	70	120	—	50
5	20	75	90	90	—

O problema consiste em minimizar o somatório dos custos de preparação, que pode concretizar-

se mais facilmente em computador, utilizando regras heurísticas ou metodologias de branch and bound. No caso anterior a sequência escolhida utilizando regras heurísticas simples e após 25 interações (em computador com o programa COMSOAL) aconselha a seguinte sequência:

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 1$$

Custo total = 300

6. Carregamento

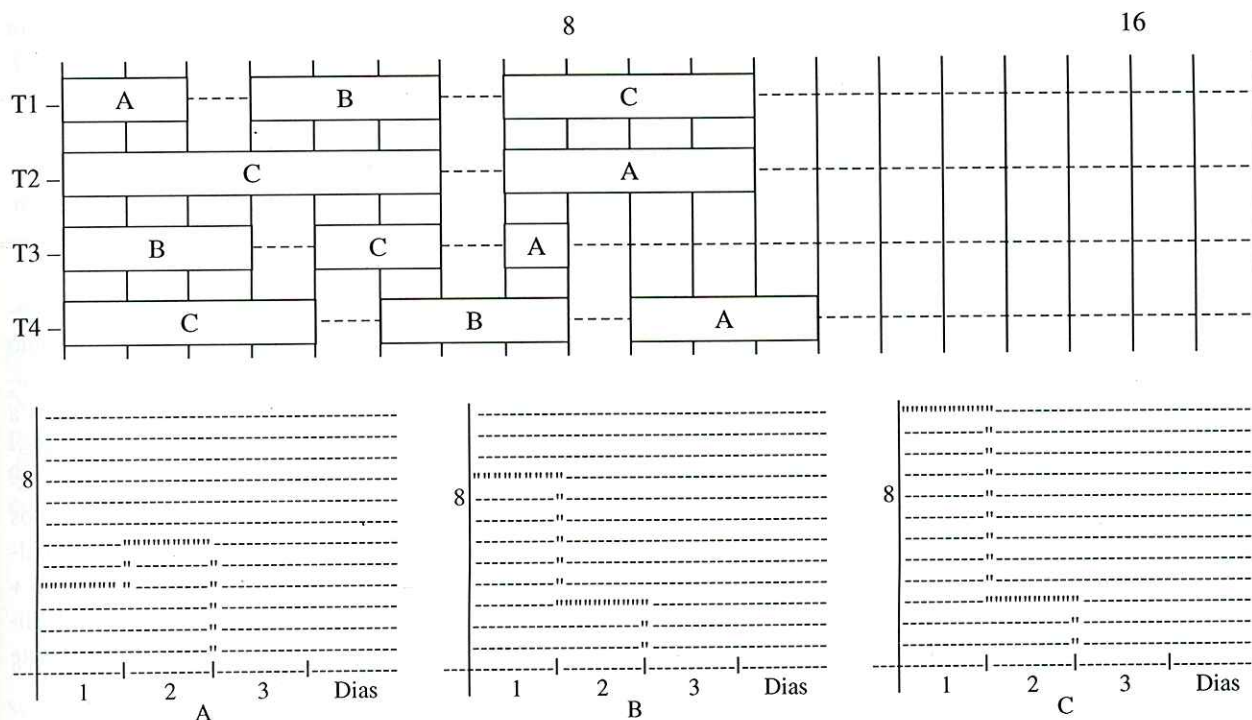
Uma análise complementar do sequenciamento anteriormente exposto, é a análise através do carregamento. Este método parte da data actual e carrega o número de tarefas a realizar para a frente no tempo estruturando-as num gráfico de GANT, não considerando a *interferência entre tarefas*. De seguida, e para cada posto de trabalho, é realizada a acumulação de cargas em tempo, pressupondo-se uma capacidade infinita. Os tempos de espera, preparação e transporte entre actividades depende das circunstâncias de cada actividade.

O objectivo é calcular a carga em cada posto de trabalho em cada momento, como resultado do que podem ser reprogramadas as actividades, testando eventuais melhores alternativas.

A resolução do caso já anteriormente colocado em sequenciamento (ponto 4) é de seguida apresentada utilizando o carregamento para a frente e considerando 1 hora entre operações sempre que há mudança de máquina.

Poderia agora ser testado um carregamento para trás a partir das datas de entrega, e a partir daí equilibrar as cargas existentes.

Uma regra heurística frequentemente usada em carregamento consiste em carregar em primeiro lugar a máquina com maior carga disponível. Esta lógica é no entanto diferente da exposta anteriormente (parágrafo 2) que se baseia nas tarefas e não nas máquinas.



7. Nota final

Um número elevado de trabalhos tem vindo a ser publicado nos últimos anos incidente sobre regras de despacho e sequenciamento.

O problema ganha uma enorme complexidade quando se considera a programação dinâmica de várias tarefas, cada uma delas com várias operações para várias máquinas, considerando tempos de movimentação e de preparação (dependentes da sequência), sem sequências pré-fixadas dentro de cada tarefa, em que os lotes podem ser fraccionados. Para maior desenvolvimento deste aspecto existem vários trabalhos, sendo de referir Pius Egbelu (1991) referenciado na Bibliografia.

Bibliografia

- Anderson, E.J., J.C. Nyirenda, «Two new rules to minimize tardiness in a job shop», pp.2013-2024, Introduction J. Production Research, 1990.
- Chan, Ding-Yu, David D. Bedworth, «Design of a scheduling system for flexible manufacturing cells», pp. 2037-2049, Introduction J. Production Research, 1990.
- Chandra, J., J.Talavage, «Intelligent dispatching for flexible manufacturing», pp.2259-2278, Introduction J. Production Research, 1991.

Chase, Richard B., Nicholas Aquilano, «Production and Operations Management – A Life Cycle Approach» (quinta edição), Irwin 1989

Dogramaci, A., «Production scheduling of independent jobs on parallel identical processors», pp.535-548, Introduction J. Production Research, 1984.

Egbelu, Pius J., «Batch production time in a multi-stage system with material-handling consideration», pp. 739-753, Introduction J. Production Research, 1991.

Garetti, M., A. Pozzetti, A. Bareggi, «On-line loading and dispatching in flexible manufacturing systems», pp.1271-1292, Introduction J. Production Research, 1990.

Huang, Philip Y., «A comparative study of priority dispatching rules in a hybrid assembly/job shop», pp.375-387, Introduction J. Production Research, 1984.

Kim, Yeong-Dae, «A comparison of dispatching rules for job shops with multiple identical jobs and alternative routings», pp. 953-962, Introduction J. Production Research, 1990.

Lawrence, Stephen R., «Scheduling a single machine to maximize net present value», pp.1141-1160, Introduction J. Production Research, 1991.

Mahmoodi, Farzad, Kevin J. Dooley, Patrick J. Starr, «An investigation of dynamic group, scheduling heuristics in a job shop manufacturing cell», pp.1695-1711, Introduction J. Production Research, 1990.

Montazeri, M., L.N. Van Wassenhove, «Analysis of scheduling rules for an FMS», pp.785-802, Introduction J. Production Research, 1990 .

Mosier, C.T., D.A.Elvers, D.Kelly «Analysis of group technology scheduling heuristics», pp.857-875, Introduction J. Production Research, 1984.

Narasimhan, Seetharma L., Srikant S.Panwalkar, «Scheduling in a two-stage manufacturing process», pp.555-564, Introduction J. Production Research, 1984.

Philipoom, Patrick R., Roberta S. Russel, Timothy D. Fry, «A preliminary investigation of multi-attribute based sequencing rules for assembly shops», pp. 739-753, Introduction J. Production Research, 1991.

Ryzin, Garrett J. Van, Sheldon X.C. Lou, Stanley B. Gershwin, «Scheduling job shops with delays», pp.1407-1422, Introduction J. Production Research, 1991.

Schroeder, Roger «Operations Management – Decision making in the operations function» (terceira edição) MacGraw Hill 1989.

Vig, Michelle M., Kevin J. Dooley, «Dynamic rules for due-date assignment», pp.1361-1377, Introduction J. Production Research, 1991.

Zeestraten, M. J., «The Look Ahead Dispatching Procedure», pp. 369-384, Introduction J. Production Research, 1990.

(1) Kanet e Hayya (1982) demonstraram que a performance das regras de despacho, especialmente quando relacionadas aos prazos de entrega, podem ser melhoradas se em vez de aplicadas a tarefas forem aplicadas a operações (normalmente uma tarefa tem várias operações) – Os prazos de entrega de cada operação são estabelecidos imediatamente após o estabelecimento dos prazos de entrega das tarefas, considerando normalmente para cada operação o tempo de processamento + tempo de espera (movimentação + preparação). Utilizando esta variante os atrasos são substancialmente reduzidos.