FORMULAÇÕES PARA O PROBLEMA DO *FLOW SHOP* EM DUAS MÁQUINAS COM PENALIDADES POR ATRASO NAS TAREFAS

José Mauricio Brasil Gonçalves
Dep. de Engenharia de Produção - Universidade Federal Fluminense (UFF)
Rua Passo da Pátria 156, Bloco D, São Domingos, Niterói – RJ. 22210-240
josemauriciobrasil@gmail.com

Eduardo Uchoa Barboza
Dep. de Engenharia de Produção - Universidade Federal Fluminense (UFF)
Rua Passo da Pátria 156, Bloco D, São Domingos, Niterói – RJ. 22210-240
uchoa@producao.uff.br

Resumo

Muitos dos problemas de escalonamento situam-se na classe de complexidade NP - difícil, sendo o caso do problema deste estudo. Este trabalho tem por objetivo propor formulações de programação inteira para o problema de escalonamento determinístico *Flow shop* com duas máquinas onde existem penalidades por atraso nas tarefas, problema conhecido na notação de três campos como $F_2 \mid \mid \sum w_j T_j$. São apresentadas três formulações com variáveis contendo índice de tempo. Dentre as três formulações, uma obteve limites duais iniciais mais próximos do valor ótimo da função objetivo, enquanto outra, mais compacta, possibilitou a solução de instâncias em tempos menores.

Palavras-chave: Problemas de Escalonamento. Formulações. Programação Inteira.

Abstract

Many scheduling problems, like the one addressed in this work, belong to the NP – hard complexity class. This paper aims to propose integer programming formulations for the deterministic Flow shop scheduling problem with two machines where there are weighted tardiness penalties. In three-field notation, $F_2 \mid \mid \sum w_j T_j$. Three different time-indexed formulations are proposed. One of the formulations obtained initial dual bounds closer to the optimum value of the objective function, while another formulation, more compact, enabled the solution of instances in shorter times.

Key-words: Scheduling. Formulations. Integer Programming.

1. Introdução

Existem numerosas situações na produção de bens ou prestação de serviços, onde máquinas necessitam ser alocadas a tarefas ao longo de um período de tempo. Resolver um problema de escalonamento é determinar a seqüência ótima de execução das tarefas nas máquinas, em relação a um ou mais objetivos (PINEDO, 2008). Problemas de escalonamento podem ser divididos levando em consideração: o caráter determinístico ou probabilístico das informações a respeito das tarefas; o ambiente de máquinas que pode envolver uma ou mais máquinas em série ou em paralelo; a existência de restrições de processamento, tal como datas de início; e o objetivo a ser minimizado, que pode ser, entre outros, o tempo de finalização da última tarefa a ser processada ou o somatório dos atrasos. Este trabalho apresenta formulações de programação inteira para resolução de problemas no ambiente de máquinas do tipo *Flow shop*, que se caracteriza por possuir *m* máquinas em série, cada tarefa deve ser processada em cada uma das máquinas seguindo a mesma seqüência de execução. O problema deste estudo também considera que cada tarefa a ser executada possui um peso (uma medida de importância) e uma data prometida para sua finalização.

2. O problema $F_2 || \sum w_j T_j$

O problema de escalonamento determinístico *Flow shop* em duas máquinas com penalidades de atraso ponderadas por tarefa, conhecido, na notação de três campos introduzida por Lawler, Lenstra e Rinnooy Kan (1982), como $F_2 || \sum w_j T_j$ é definido da seguinte forma: seja $J = \{1, \dots, n\}$ um conjunto de tarefas que devem ser processadas em duas máquinas 1 e 2, nesta ordem. As máquinas só podem processar uma tarefa por vez e o processamento de uma tarefa na máquina 2 só pode ser iniciado após o término do seu processamento na máquina 1. Os tempos de processamento de uma tarefa $j \in J$ nas máquinas 1 e 2 são dados por p_{1j} e p_{2j}, respectivamente. Cada tarefa $j \in J$ possui uma data prometida de entrega d_j e um peso w_j. O problema consiste em determinar quando cada tarefa deve ser processada em cada máquina de forma a minimizar $\sum_{j \in J} w_j T_j$, onde $T_j = \max\{0, C_j - d_j\}$ é o tempo de atraso da tarefa *j* com respeito a sua data de finalização C_j na segunda máquina.

3. Características do problema $F_2 || \sum w_j T_j$

O problema $F_2 || \sum w_j T_j$ é $\mathcal{NP}$ –difícil. Isto pode ser verificado notando-se que o problema $1 || \sum w_j T_j$, com o mesmo objetivo, mas apenas uma máquina, é sabidamente $\mathcal{NP}$ – difícil, (LENSTRA, RINNOY KAN E BRUCKER, 1977).

Outra característica do problema $F_2 || \sum w_j T_j$ (e que cumpre papel fundamental nas formulações que serão apresentadas mais a frente) é também encontrada num dos

mais clássicos problemas de escalonamento: $F_2||C_{max}$ (*Flow shop* com duas máquinas, tendo como objetivo a minimização do tempo de finalização da última tarefa). Para esse problema é possível encontrar uma solução ótima em tempo polinomial (JOHNSON, 1954). Além disto, é sabido que para este problema, a adição da restrição de que as seqüências das tarefas na primeira e na segunda máquina sejam iguais (ou seja, as tarefas obedecem a uma fila do tipo primeiro que entra é o primeiro que sai, em frente à segunda máquina) não altera a solução ótima do problema. Em outras palavras, a solução ótima do problema $F_2|prmu|C_{max}$ (a notação *prmu* indica a restrição de seqüências iguais nas máquinas) também é uma solução ótima do problema $F_2||C_{max}$. Esta última propriedade, conforme Gonçalves e Uchoa (2009), também é válida para o $F_2||\sum w_j T_j$.

Teorema 1: Para toda solução ótima do problema $F_2||\sum w_j T_j$ onde o escalonamento é não permutacional, é possível encontrar um seqüenciamento permutacional ótimo, a partir da mudança da ordem de execução das tarefas na primeira máquina.

Prova: Sejam $j_{11}, \dots, j_{1n}$ e $j_{21}, \dots, j_{2n}$ as seqüências de tarefas nas máquinas 1 e 2, respectivamente, de um escalonamento ótimo não permutacional. Afirmamos que replicando a seqüência da máquina 2 na máquina 1, sem colocar tempos ociosos entre as tarefas, e mantendo a mesma seqüência e tempos de início na máquina 2, obtemos um outro escalonamento viável e com mesmo valor da função objetivo. Suponha que uma tarefa $k = j_{2k}$ inicie seu processamento na segunda máquina no tempo t_{2k}. A viabilidade do escalonamento original implica em $t_{2k} \geq \sum_{i=1}^{k} p_{1j_{2i}}$, ou seja, todas as tarefas de $j_{21}, \dots, j_{2k}$ já vão ter sido executadas na máquina 1 no novo escalonamento permutacional. Portanto, esse novo escalonamento é viável e possui o mesmo valor da função objetivo. □

O resultado acima implica que uma formulação para o problema $F_2||\sum w_j T_j$ pode ser usada para achar uma solução ótima de uma instância do problema $F_2|prmu|\sum w_j T_j$e vice versa.

4. Formulações

As formulações apresentadas nesta pesquisa utilizaram como ponto de partida o trabalho de Dyer e Wolsey (1990) o qual introduziu na literatura uma formulação que emprega variáveis com índices t de tempo para o problema de escalonamento determinístico com uma máquina, apresentada a seguir:

$$Função\ Objetivo:\ Min\ Z = \sum_{j=1}^{n} \sum_{t=1}^{t_{max}-p_j+1} c_{jt} x_{jt} \qquad (1)$$

$$Sujeito\ a: \sum_{t=1}^{t_{max}-p_j+1} x_{jt} = 1 \quad \forall \quad j = 1, \dots, n \qquad (2)$$

$$\sum_{j=1}^{n} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \quad \forall \quad t = 1, \dots, t_{max} \qquad (3)$$

$$x_{jt} \in \{0,1\} \quad \forall \quad j = 1, \dots, n; \quad \forall \quad t = 1, \dots, t_{max} - p_j + 1 \qquad (4)$$

Define-se a variável x_{jt}, que assume valor igual a 1 se a tarefa *j* é finalizada no tempo *t*, recebendo valor nulo se a tarefa *j* é finalizada em qualquer instante de tempo diferente de *t*. O dado c_{jt} pode representar qualquer custo que seja associado a uma tarefa *j* no tempo de finalização *t*. Por exemplo, pode-se fazer $c_{jt} = w_j.max\ (0\ ,\ t\text{-}d_j)$ o que é equivalente a $w_j.T_j$, caso $c_j = t$. A constante t_{max} consiste num tempo máximo atribuído para execução de todas as tarefas no sistema.

Foram desenvolvidas três formulações com índice de tempo para o problema $F_2||\sum w_j T_j$, denominadas genericamente por Formulação 1, Formulação 2 e Formulação 3. As formulações são apresentadas a seguir.

4.1 Formulação 1

$$Função\ Objetivo: Min\ Z = \sum_{j=1}^{n} \sum_{t=p_{1j}+p_{2j}}^{t_{max}} custo(j,t).x_{j2t} \qquad (5)$$

$$Onde: custo(j,t) = T_j.w_j$$

$$Sujeito\ a: \sum_{t=p_{1j}}^{S_1} x_{j1t} = 1 \quad \forall \quad j = 1, \dots, n \qquad (6)$$

$$\sum_{t=p_{1j}+p_{2j}}^{t_{max}} x_{j2t} = 1 \quad \forall \quad j = 1, \dots, n \qquad (7)$$

$$\sum_{j=1}^{n} \sum_{t=t_1}^{t_1+p_{ij}-1} x_{jit} \leq 1 \quad \forall \quad t_1 = 0, \dots, t_{max}, \quad \forall \quad i = 1,2 \qquad (8)$$

$$\sum_{t=t_1-p_{2j}+1}^{S_1} x_{j1t} + \sum_{t=0}^{t_1} x_{j2t} \leq 1 \quad \forall \ j = 1, \dots, n\ ; \ \forall \ t_1 = 0, \dots, t_{max} \qquad (9)$$

$$x_{jit} \in \{0,1\}, \quad \forall \quad j = 1, \dots, n; \quad \forall \quad t = 0, \dots, t_{max}; \quad \forall \quad i = 1\ e\ 2 \qquad (10)$$

A variável de decisão binária x_{jit} assume valor igual a 1 se a tarefa *j* tem seu processamento finalizado na máquina *i* no tempo igual a *t*. Assume valor igual a zero caso tenha seu processamento finalizado na máquina *i* num instante diferente de *t*. A função objetivo (5) consiste no somatório de todas as variáveis que representam possíveis finalizações de uma tarefa *j* na máquina 2. São incluídas somente variáveis com índices de finalização na máquina 2 pois o custo $T_j.w_j$ é uma função da data de finalização da tarefa *j* no sistema. Isto é o mesmo que o tempo de finalização da tarefa *j* na máquina 2. O coeficiente de cada variável x_{j2t} consiste na multiplicação do atraso T_j, (diferença entre o tempo *t* de conclusão da tarefa e a data de término prometida d_j) pelo peso w_j associado à tarefa. Caso a variável represente a finalização de uma tarefa não atrasada, o coeficiente assume valor nulo.

Os grupos de restrições (6) e (7) fazem com que cada tarefa *j* seja finalizada exatamente uma única vez em cada uma das duas máquinas. Isto é feito determinando que, de todas as variáveis que representam términos possíveis de uma dada tarefa *j* numa dada máquina *i*, apenas uma delas tenha valor não nulo. Especificamente para a restrição (6), voltada à primeira máquina, toma-se como premissa que não há benefício nenhum oriundo da sua ociosidade, antes que esta execute todas as tarefas. Assim, a constante S_1 representa a soma dos tempos de execução na máquina 1 de todas as tarefas designadas. O grupo de restrições (8) impede que uma máquina processe simultaneamente mais de uma tarefa. Isto é feito da seguinte forma: dadas as *n* tarefas e um intervalo de tempo limitado entre um valor de tempo t_1 (valor mínimo) e o tempo t_1 somado aos tempos de processamento de todas as *n* tarefas (valor máximo), é impossível que exista mais de uma variável que represente a finalização da tarefa *j* na máquina *i* nesse intervalo. O grupo (9) de restrições determina que as tarefas sigam a ordem de execução do *flow shop*: todas as tarefas devem ser processadas inicialmente na máquina 1 e, em seguida, na máquina 2, fazendo com que dentre todas as variáveis que representam possíveis finalizações na máquina 2, até um tempo t_1, somada a todas as variáveis que representam eventuais finalizações da tarefa na máquina 1 em tempos de t_1 até t_{max} (ou em tempos onde a tarefa não tivesse sido terminada na máquina 1, quando comparadas as demais variáveis com índice *i* = 2 na mesma equação) apenas uma variável poderá representar a finalização de uma tarefa *j* numa máquina *i* no tempo *t*. As equações do grupo (10) compreendem as restrições de integralidade e não negatividade.

4.2 Formulação 2

A formulação 2 baseia-se no Teorema 1, o qual afirma que os problemas $F_2||\sum w_jT_j$ e $F_2|prmu|\sum w_jT_j$ são equivalente. Neste sentido, na Formulação 2, a preocupação é elaborar um seqüenciamento para a segunda máquina do sistema, e depois utilizar esta mesma seqüência de execução de tarefas para a máquina 1. Procedendo desta maneira, é possível suprimir da variável de decisão o índice de

máquina i. Tem-se com isso a variável de decisão x_{jt}, que representa o tempo t de conclusão da tarefa j na segunda máquina.

$$Função\ Objetivo: Min\ Z = \sum_{j=1}^{n} \sum_{t=p_{1j}+p_{2j}}^{t_{max}} custo(j,t).x_{jt} \qquad (11)$$

$$Onde: custo(j,t) = T_j.w_j$$

$$Sujeito\ a: \sum_{t=p_{1j}+p_{2j}}^{t_{max}} x_{jt} = 1 \quad \forall \quad j = 1, \dots, n \qquad (12)$$

$$\sum_{j=1}^{n} \sum_{t=t_1}^{t_1+p_{2j}-1} x_{jt} \leq 1 \quad \forall \quad t_1 = 0, \dots, t_{max} \qquad (13)$$

$$\sum_{j=1}^{n} \sum_{t=0}^{t_1+p_{2j}} p_{1j}.x_{jt} \leq t_1 \quad \forall \quad t_1 = 0, \dots, t_{max} \qquad (14)$$

$$x_{jt} \in \{0,1\} \quad \forall \quad j = 1, \dots, n. \quad \forall \quad t = 0, \dots, t_{max} \qquad (15)$$

A função objetivo (11) utiliza o mesmo princípio da função objetivo da formulação 1, representando a minimização do somatório ponderado dos tempos de finalização das tarefas na segunda máquina. O primeiro grupo de restrições (12), com propósito semelhante ao primeiro grupo da Formulação 1, equações (6) e (7), faz com que cada tarefa j seja finalizada exatamente uma única vez na última máquina. O segundo grupo de restrições (13) encarrega-se de fazer com que a máquina 2 processe uma tarefa de cada vez. O terceiro grupo de restrições (14) faz com que seja respeitado o seqüenciamento característico do ambiente de *Flow shop.* A montagem das restrições utiliza o raciocínio a seguir: tomado um tempo t_1 o somatório de todos os tempos de processamento na máquina 1 das tarefas que eventualmente já tenham sido finalizadas no sistema deve ser menor ou igual ao tempo t_1. Ou seja, para que a tarefa possa ser finalizada até $t_1 + p_{2j}$, esta deve ter sido finalizada na máquina 1 até o tempo t_1. O grupo (15) apresenta as restrições de integralidade e não negatividade da variável utilizada.

4.3 Formulação 3

A Formulação 3 assemelha-se bastante à Formulação 2, partindo do princípio que os problemas $F_2||\sum w_j T_j$ e $F_2|prmu|\sum w_j T_j$ são equivalentes. Emprega as variáveis x_{jt} que assim como na Formulação 2, representam o tempo t de finalização da tarefa j no sistema. De forma a reduzir a repetição de variáveis nas restrições,

notadamente naquelas restrições que garantem o cumprimento da seqüência correta das tarefas ao longo das máquinas, incluem-se na formulação as variáveis auxiliares y_{jt}. Estas variáveis representam o quanto da tarefa *j* já pode ter sido processada na máquina 1, num dado instante *t*. As variáveis y_{jt} podem assumir valores inteiros (não somente binários).

Função Objetivo: Expressão (11), descrita na Formulação 2.

Sujeito a: O primeiro e segundo grupo de restrições são representadas pelas expressões (12) e (13), respectivos primeiro e segundo grupo de restrições da Formulação 2.

$$(y_{jt_1} - y_{jt_1-1}) - \sum_{t=t_1+p_{2j}}^{t_1+(p_{1j}+p_{2j})-1} x_{jt} = 0 \quad \forall \quad j = 1, \dots, n\,; \quad \forall \quad t_1 = 1, \dots, (t_{max} - (p_{1j} + p_{2j}) + 1) \qquad (16)$$

$$\sum_{j=1}^{n} y_{jt} \leq t \qquad \forall \quad t = 1, \dots, t_{max} \qquad (17)$$

O terceiro grupo de restrições da Formulação 3, representada pela expressão (16) acima, encarrega-se de manter o seqüenciamento correto das *n* tarefas (processamento na máquina 1 e, depois, processamento na máquina 2). Neste grupo, a variável y_{jt} atua no sentido de substituir uma série de variáveis x_{jt} em cada uma das restrições. Assim, diminui-se o número total de variáveis nas restrições. A lógica envolvida em (16) é a seguinte: para um dado tempo t_1 e tarefa *j*, a diferença entre o quanto a tarefa *j* já foi processada na máquina 1 em relação ao tempo *t* imediatamente anterior (esta diferença será igual a 0 ou 1) é igual ao somatório das variáveis x_{jt} que representam possíveis finalizações da tarefa *j*, considerando o tempo t_1.

O quarto grupo de restrições da Formulação 3, representado pela expressão (17) limita o somatório dos valores das variáveis y_{jt}, para um dado tempo *t*. Isto precisa ser feito uma vez que, o somatório dos processamentos na máquina 1 de todas as tarefas, realizados até o tempo *t* não pode ser, de fato, maior do que o próprio tempo *t*.

As restrições de integralidade e não negatividade são representadas pelas expressões (15) (semelhante a restrição de integralidade e não negatividade da Formulação 2) e a expressão (18) abaixo:

$$y_{jt} \geq 0, \quad \forall \quad j = 1, \dots, n. \qquad \forall \qquad t = 0, \dots, t_{max} \qquad (18)$$

5. Resultados Computacionais

Como forma de analisar o desempenho das Formulações 1, 2 e 3, foram utilizadas instâncias de testes com até 50 tarefas. Para cada instância com o mesmo número de tarefas, foi atribuído uma letra (A ou B). As instâncias que contêm a letra A apresentam os tempos de processamento nas máquinas 1 e 2 (p_{1j} e p_{2j}) entre os valores 1 e 5. Já as instâncias com letra B, os valores destes dois parâmetros situam-se entre 1 e 10. Em relação aos pesos associados às tarefas (w_j), em todas as instâncias os valores mínimos e máximos estão entre 1 e 5.

A data de término prometida (d_j) apresenta como valor mínimo a soma dos tempos de processamento da tarefa j nas máquinas 1 e 2 ($p_{1j} + p_{2j}$) e como máximo o maior valor entre a soma dos tempos de processamento nas máquinas 1 e 2 de todas as tarefas.

Todos esses números foram gerados a partir de uma distribuição aleatória uniforme utilizando a fórmula "ALEATÓRIOENTRE" do *MS-Excel 2007*.

O computador utilizado foi um *Intel Core 2 duo* de 2.13 GHz, com 2,00 GB de memória RAM. O tempo limite de execução máximo imposto foi de 14.000 segundos (4 horas), sendo utilizado sempre 1 *thread*. Foi produzida uma tabela com os resultados dos desempenhos das formulações nas instâncias de testes, intitulada "Tabela de Resultados" (Tabela 2). A Tabela 1 (exposta a seguir) contém o significado de cada campo da Tabela de Resultados.

CAMPO	INFORMAÇÃO DO CAMPO
Instância	Informa a instância resolvida
Valor da Relaxação Linear	Informa o valor da função objetivo ao se fazer a relaxação linear do problema original
Valor da Relaxação Linear + Cortes	Informa o valor da função objetivo após a relaxação linear do problema e a aplicação do pré-processamento e dos cortes do CPLEX 11.0
Valor da função objetivo	Transmite o melhor valor da função objetivo até o limite de tempo estabelecido para execução do algoritmo
Valor da função objetivo x Relaxação linear	Registra a diferença percentual entre o valor da relaxação linear do problema original e o valor da função objetivo após o limite de tempo estabelecido.
Tempo (s)	Informa o tempo utilizado na resolução. Caso o tempo seja igual ao valor limite estabelecido, isto significa que o valor do campo "valor da função objetivo" não é provadamente ótimo
Número de nós	Informa o número de nós abertos na árvore de *branch and bound*
Distância máxima do valor ótimo	Para as instâncias onde não foi encontrada solução ótima informa-se, percentualmente, a distância máxima entre a melhor solução encontrada e a solução ótima. Caso o valor deste campo seja igual a 0,00%, a solução do campo "valor da função objetivo" é ótima.

Tabela 1: Significado de cada um dos campos da Tabela de Resultados

Formulação	Instância	Valor da Relaxação Linear	Valor da Relaxação Linear + Cortes	Valor da função objetivo	Valor da função objetivo x Relaxação linear	Tempo (s)	Número de nós	Distância máxima do valor ótimo
1	4ª	26,00	26,00	26	0,00%	0,22	1	0,00%
1	4B	43,00	43,00	43	0,00%	0,02	1	0,00%
1	6ª	57,00	58,00	58	1,72%	0,08	1	0,00%
1	6B	54,00	54,00	54	0,00%	0,09	1	0,00%
1	8ª	71,75	77,00	77	6,82%	0,28	1	0,00%
1	8B	262,50	266,00	266	1,32%	1,19	1	0,00%
1	10ª	99,57	102,00	102	2,38%	1,06	1	0,00%
1	10B	364,05	376,00	376	3,18%	21,58	1	0,00%
1	12ª	137,02	144,00	144	4,85%	53,2	1	0,00%
1	12B	432,77	455,00	455	4,89%	178,33	113	0,00%
1	14ª	190,55	204,00	204	6,59%	25,14	1	0,00%
1	14B	672,29	683,25	693	2,99%	455,98	79	0,00%
1	16ª	232,84	239,42	252	7,60%	106,36	63	0,00%
1	16B	861,50	873,66	886	2,77%	515,56	79	0,00%
1	18ª	259,77	265,36	274	5,19%	153,16	43	0,00%
1	18B	1.094,68	1.105,91	1.134	3,47%	4.955,83	1.062	0,00%
1	20ª	478,25	486,38	493	2,99%	175,81	31	0,00%
1	20B	1.294,84	1.305,91	1.346	3,80%	8.921,28	4.313	0,00%
1	25ª	673,93	684,96	688	2,04%	249,64	3	0,00%
1	25B	1.720,88	1.735,59	1.764	2,44%	13.684,49	835	0,00%
1	30ª	946,61	953,76	968	2,21%	546,06	22	0,00%
1	30B	2.609,26	2.620,95	2.795	6,65%	14.400	26	5,62%
1	35ª	1.221,64	1.230,68	1.278	4,41%	14.400	2.241	2,66%
1	35B	3.662,13	3.673,26	4.094	10,55%	14.400	3.625	9,84%
1	40ª	1.430,40	1.438,34	1.479	3,29%	14.400	1.410	1,83%
1	40B	4.664,20	4.679,12	7.349	36,53%	14.400	7	36,14%
1	45ª	2.093,20	2.102,54	2.302	9,07%	14.400	183	8,21%
1	45B	5.878,39	5.894,69	-	-	14.400	3	-
1	50ª	2.492,77	2.502,04	2.817	11,51%	14.400	30	11,04%
1	50B	6.521,98	6.537,46	7.352	11,29%	14.400	3	10,91%
2	4ª	22,54	26,00	26	13,31%	0,01	1	0,00%
2	4B	43,00	43,00	43	0,00%	0,02	1	0,00%
2	6ª	50,19	58,00	58	13,47%	0,05	1	0,00%
2	6B	54,00	54,00	54	0,00%	0,06	1	0,00%
2	8ª	66,50	77,00	77	13,64%	0,11	1	0,00%
2	8B	242,88	266,00	266	8,69%	0,39	1	0,00%
2	10ª	87,60	102,00	102	14,12%	0,34	1	0,00%
2	10B	338,26	376,00	376	10,04%	8,02	1	0,00%
2	12ª	119,20	144,00	144	17,23%	12,03	1	0,00%
2	12B	405,19	427,49	455	10,95%	74,27	462	0,00%
2	14ª	177,74	204,00	204	12,87%	10,00	1	0,00%
2	14B	622,69	648,19	693	10,15%	264,08	2.712	0,00%
2	16ª	219,85	237,02	252	12,76%	22,63	68	0,00%
2	16B	809,58	838,28	886	8,63%	358,61	2.780	0,00%
2	18ª	243,07	265,72	274	11,29%	31,95	19	0,00%
2	18B	1.028,41	1.064,05	1.134	9,31%	3.355,73	30.708	0,00%
2	20ª	442,48	483,93	493	10,25%	30,19	30	0,00%
2	20B	1.224,42	1.262,36	1.348	9,17%	14.400	118.101	0,37%
2	25ª	630,93	672,18	688	8,29%	56,48	218	0,00%
2	25B	1.617,15	1.672,55	1.764	8,32%	14.400	92.018	0,34%
2	30ª	895,53	930,13	968	7,49%	1.738,66	24.714	0,00%
2	30B	2.502,89	2.546,14	2.732	8,39%	14.400	17.763	4,21%
2	35ª	1.173,29	1.200,38	1.264	7,18%	14.400	189.783	0,63%
2	35B	3.530,73	3.578,42	3.815	7,45%	14.400	6.225	4,38%
2	40ª	1.396,32	1.416,95	1.459	4,30%	10.536,48	56.108	0,00%
2	40B	4.504,20	4.548,74	4.923	8,51%	14.400	3.987	5,53%
2	45ª	2.058,12	2.082,32	2.149	4,23%	14.400	46.389	1,21%
2	45B	5.687,38	5.735,52	6.268	9,26%	14.400	1.491	7,18%
2	50ª	2.452,06	2.468,27	2.582	5,03%	14.400	14.788	3,10%
2	50B	6.337,42	6.380,52	7.048	10,08%	14.400	783	8,61%
3	4ª	26,00	26,00	26	0,00%	0,02	1	0,00%
3	4B	43,00	43,00	43	0,00%	0,05	1	0,00%
3	6ª	55,00	58,00	58	5,17%	0,05	1	0,00%
3	6B	54,00	54,00	54	0,00%	0,09	1	0,00%
3	8ª	70,64	77,00	77	8,26%	0,70	1	0,00%
3	8B	258,80	266,00	266	2,71%	0,39	1	0,00%
3	10ª	98,25	102,00	102	3,68%	1,48	1	0,00%
3	10B	361,81	376,00	376	3,77%	13,36	1	0,00%
3	12ª	135,80	144,00	144	5,69%	14,73	1	0,00%
3	12B	431,15	440,22	455	5,24%	76,17	292	0,00%
3	14ª	190,33	204,00	204	6,70%	25,06	1	0,00%
3	14B	670,24	680,50	693	3,28%	139,17	397	0,00%
3	16ª	232,60	239,13	252	7,70%	29,98	81	0,00%
3	16B	859,87	868,45	886	2,95%	203,09	444	0,00%
3	18ª	258,44	267,89	274	5,68%	28,06	3	0,00%
3	18B	1.091,27	1.105,84	1.134	3,77%	1.007,55	4.400	0,00%
3	20ª	475,62	486,09	493	3,53%	41,67	10	0,00%

Tabela 2: Resultados de desempenho das formulações 1, 2 e 3

Formulação	Instância	Valor da Relaxação Linear	Valor da Relaxação Linear + Cortes	Valor da função objetivo	Valor da função objetivo x Relaxação linear	Tempo (s)	Número de nós	Distância máxima do valor ótimo
3	20B	1.290,59	1.304,24	1.346	4,12%	3.572,88	19.329	0,00%
3	25ª	671,43	680,18	688	2,41%	227,78	3.212	0,00%
3	25B	1.719,24	1.733,06	1.764	2,54%	3.691,86	6.240	0,00%
3	30ª	940,33	951,66	968	2,86%	657,56	2.716	0,00%
3	30B	2.607,43	2.616,84	2.720	4,14%	14.400	6.722	2,35%
3	35ª	1.219,14	1.226,77	1.263	3,47%	10.919,92	56.209	0,00%
3	35B	3.657,94	3.667,87	3.756	2,61%	14.400	3.744	1,44%
3	40ª	1.429,45	1.437,02	1.459	2,03%	1.130,02	9.861	0,00%
3	40B	4.658,93	4.670,73	4.844	3,82%	14.400	2.620	2,46%
3	45ª	2.091,99	2.103,28	2.149	2,65%	14.400	24.347	1,02%
3	45B	5.873,70	5.887,28	6.135	4,26%	14.400	564	3,21%
3	50ª	2.490,81	2.501,02	2.548	2,24%	14.400	11.998	0,98%
3	50B	6.516,84	6.529,47	6.970	6,50%	14.400	490	5,54%

Tabela 2: Resultados de Desempenho das Formulações 1, 2 e 3 (continuação)

A análise do desempenho das Formulações 1, 2 e 3 considerará as três questões a seguir:

- Proximidade do valor da relaxação linear do problema inicial (limite dual inicial) em relação ao valor ótimo da função objetivo;
- Sucesso ou fracasso em encontrar a solução ótima dentro do limite de tempo máximo estabelecido. Para aquelas instâncias onde não foi encontrada a solução ótima, o desempenho será medido em função da distância máxima entre a melhor solução encontrada e a solução ótima; e
- Tempo empregado para encontrar a solução ótima.

No tocante a proximidade do valor da relaxação linear em relação ao valor ótimo, é possível notar nas tabelas de resultados que a Formulação 1 obteve valores mais próximos ao valor ótimo em todas as instâncias. A única exceção foi o fato de que, nas instâncias 4B e 6B, os limites duais iniciais foram exatamente iguais ao obtido pelas demais Formulações. Dentre as Formulações 2 e 3, esta última gerou limites mais próximos do valor ótimo, quando comparadas entre si.

Tomado o segundo aspecto da análise: sucesso ou fracasso em encontrar a solução ótima dentro do limite de tempo máximo estabelecido (ou a distância máxima da solução ótima, para os casos onde não foi encontrada solução comprovadamente ótima dentro do limite de tempo), qual, dentre as três formulações, apresentou o melhor desempenho?

O exame das tabelas de resultados das instâncias transmite que, ao utilizar a Formulação 1, nove instâncias não apresentaram solução ótima. Ao empregar a Formulação 2, dez instâncias não registraram solução ótima. Já o emprego da Formulação 3 fez com que não se chegasse à solução ótima em sete casos. A Tabela 3 sintetiza estes resultados:

Formulação	Nº de instâncias aonde não se chegou ao valor ótimo em 14.400s
1	9
2	10
3	7

Tabela 3 – Número de instâncias onde não se encontrou a solução ótima

Com base na Tabela acima, é possível observar que as três formulações apresentaram números próximos de instâncias onde não foi possível encontrar a solução ótima dentro do limite de tempo. Contudo, levando em consideração a distância máxima entre o valor ótimo e a solução encontrada, nota-se que, no tocante ao sucesso ou fracasso em encontrar a solução ótima dentro do limite de tempo máximo estabelecido, a Formulação 3 apresentou o melhor desempenho. Quando são analisadas estas distâncias, observa-se que a Formulação 3 apresentou as menores diferenças. Ou seja, para aquelas formulações onde não se encontrou a solução ótima, pelo menos a solução encontrada ficou mais próxima da solução comprovadamente ótima. Julgar qual formulação apresentou o segundo melhor desempenho dentre as Formulações 1 e 2 é difícil pois, apesar de a Formulação 1 apresentar menor número de instâncias onde não se encontrou o valor ótimo, as distâncias máximas em relação ao valor ótimo verificadas foram superiores quando comparadas à Formulação 2.

Há que se notar também que, independente da formulação empregada, a dificuldade em encontrar a solução ótima concentra-se na parte inferior das tabelas de resultados. Isto é justificável na medida em que, quanto maior o número de tarefas da instância, maior é o número de variáveis e restrições empregadas.

No tocante ao último ponto de comparação, tempo empregado para encontro da solução ótima, as análises serão feitas tomando por base as instâncias da série A e da série B, separadamente.

Utilizando as instâncias da série A, as Formulações 2 e 3 apresentaram, cada uma, 6 instâncias onde o tempo de execução foi superior a 100 segundos. O número de instâncias não resolvidas em até 100 segundos, com a Formulação 1, foi igual a 7. Apesar do fato de que as Formulações 2 e 3 apresentarem mesmo número de instâncias com tempo de execução superior a 100 segundos, a Formulação 3 foi superior ao lidar com as instâncias de número de tarefas igual ou superior a trinta e cinco. Empregando a Formulação 3, foi possível encontrar a solução ótima da instância de trinta e cinco tarefas, encontrar de maneira mais rápida a solução ótima da instância de quarenta tarefas (além de encontrar soluções mais próximas do valor ótimo da função objetivo nas instâncias de 45 e 50 tarefas, fato abordado anteriormente).

Ao empregar as instâncias da série B, a Formulação 3 possibilitou solucionar o maior número de instâncias com vinte tarefas ou mais. A Formulação 1 conseguiu resolver instâncias com maior número de tarefas do que a Formulação 2 (fato que não ocorreu quando foram utilizadas as instâncias da série A).

É possível sintetizar os critérios analisados e eleger, dentre as três formulações, a melhor? Sem dúvida, trata-se de uma questão de qual critério a ser adotado. Se levado em consideração a questão de quão apertada é a formulação, a Formulação 1 seria eleita

como melhor. Se tomado o critério de velocidade e capacidade de resolver as instâncias propostas, a formulação que mais se destacaria seria a Formulação 3.

Referências Bibliográficas

DYER, M; WOLSEY, L. Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discrete Applied Mathematics*. 26, 1990, pp. 255-270.

GONÇALVES, J.M.B. *Formulações para o problema do Flow shop em duas máquinas com penalidades por atraso nas tarefas*. Niterói: Dissertação de Mestrado, Programa de Pós Graduação em Engenharia de Produção – Universidade Federal Fluminense, 2009.

JOHNSON, S. M. Optimal two- and three-stage Production Schedules with set-up times included, *Naval Research Logistics Quartely*, 1, 1954, pp. 61-68.

LAWLER, E. L; LENSTRA, J. K; RINNOOY KAN, A. H. G. Recent Developments in Deterministic Sequencing and Scheduling: A Survey. *Deterministic and Stochastic Scheduling*, 1982, pp. 35-74.

LENSTRA, J. K; RINNOOY KAN, A. H. G; BRUCKER, P. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, Vol. 1, 1977, pp. 343-362.

PINEDO, M. Scheduling. *Theory, Algorithms, and Systems*. New York: Springer, 2008.

WOLSEY, L. *Integer Programming*. New York: Wiley, 1998.