

Formulações matemáticas e estratégias de resolução para o problema *job shop* clássico

Sergio Gomez Morales^a, Débora Pretti Ronconi^{**}

^aUniversidade de São Paulo, São Paulo, SP, Brasil

*dronconi@usp.br

Resumo

O problema de sequenciamento de tarefas no ambiente de produção *job shop* se caracteriza por conter n tarefas que devem ser processadas por m máquinas, em que cada tarefa a ser realizada é constituída por um roteiro específico de operações com ordem de precedência preestabelecida. O objetivo deste trabalho é realizar uma análise comparativa das formulações matemáticas para este ambiente, minimizando o tempo total de execução de todas as tarefas em todas as máquinas (*makespan*). Modelos conhecidos e um novo modelo são avaliados e comparados através de testes computacionais em problemas-teste da literatura. Adicionalmente, estratégias de resolução são propostas. Experimentos computacionais utilizando um *software* comercial conhecido indicam que as estratégias propostas são eficientes para a redução do *gap* de otimalidade.

Palavras-chave

Job shop. Programação da produção. *Makespan*. Modelos de programação linear inteira mista.

1. Introdução

O modelo conceitual *job shop clássico* é um modelo que representa o ambiente operacional de empresas manufatureiras com as seguintes características: alta variedade de produtos, volume baixo de produção e uma organização dividida em áreas funcionais. O modelo procura simular e determinar a programação da produção (*scheduling*) da empresa com a premissa de que cada produto a ser elaborado surge através de um pedido realizado pelo cliente com especificações e particularidades próprias.

Formalmente, o modelo *job shop* é definido como um conjunto de n tarefas (*jobs*), um conjunto de m máquinas e m operações definidas para cada tarefa j . A série de operações deve ser determinada no instante de início da produção da fábrica, cada operação deve ser realizada numa única máquina por um determinado período de tempo sem interrupção, e cada máquina pode realizar somente uma operação de cada vez.

A Figura 1 ilustra o ambiente descrito. Note que as tarefas possuem diferentes sequências de processamento no ambiente, ao contrário de outros

ambientes como o *flow shop*, em que as tarefas são processadas em estágios sucessivos, formando um fluxo contínuo.

A programação de tarefas no ambiente *job shop* é um problema combinatório de elevada complexidade, em que o sequenciamento de n tarefas em m máquinas envolve (n^m) soluções possíveis. Este problema é classificado por Garey et al. (1976) como *NP-Hard* para a minimização do *makespan* (instante de término de processamento de todas as tarefas a serem processadas no ambiente) e pode ser descrito através da notação $\alpha|\beta|\gamma$, como $Jm/n/C_{max}$.

Neste trabalho, considera-se o caso determinístico estático, em que o tempo de processamento de cada operação e a sequência de operações para cada tarefa são conhecidos e não variantes, e todas as máquinas e tarefas estão disponíveis no começo do processamento. As suposições e conjecturas para facilitar a concepção e definição do problema são citadas a seguir.

Suposições referentes aos produtos ou tarefas:
 1) cada tarefa é determinada no início do período de

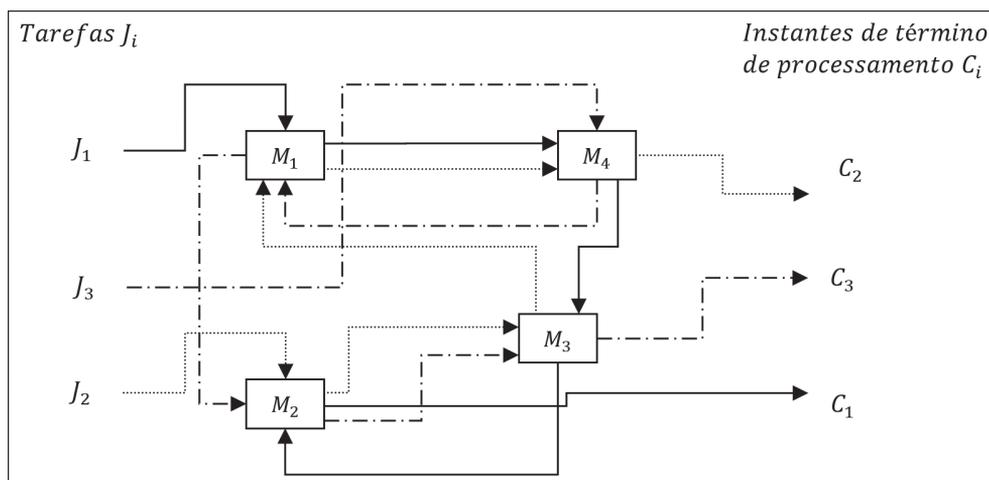


Figura 1. Representação esquemática do problema. Fonte: Elaborada pelos autores.

sequenciamento e deve estar disponível em qualquer momento; *ii*) existe uma relação de precedência entre qualquer par de operações de uma mesma tarefa, mas não entre pares de operações de diferentes tarefas; *iii*) o roteiro de cada tarefa é definido por sua sequência de operações e as máquinas necessárias para seu processamento; *iv*) cada operação possui um tempo de processamento determinístico e contínuo que inclui o tempo de transporte entre máquinas e os tempos de preparação (*setup*); *v*) não há datas de entrega; e *vi*) todas as tarefas devem ser processadas por todas as máquinas. Suposições referentes às máquinas ou processos: *i*) cada máquina processa no máximo uma tarefa de cada vez (ou equivalentemente uma operação de cada vez); *ii*) cada operação, uma vez iniciada numa máquina, deve ser finalizada sem interrupção; e *iii*) existem *buffers* ilimitados antes e depois de cada máquina.

Dentro da literatura, vários modelos de Programação Linear Inteira Mista (PLIM) são propostos para encontrar as soluções ótimas deste problema. Wagner (1958) propôs uma formulação para o problema *job shop* que se caracteriza por designar tarefas em posições na sequência de produção. Manne (1960), por outro lado, desenvolveu uma formulação baseada no uso de restrições disjuntivas para controlar a ordem de precedência das tarefas na sequência de produção.

Uma modificação da formulação de Manne (1960) foi apresentada por Liao & You (1992), em que cada par de restrições disjuntivas é combinado numa única restrição de igualdade. Já Wilson (1989) sugeriu uma alternativa à formulação de Wagner (1958) baseada na relação de precedência entre duas tarefas consecutivas numa máquina.

Buscando comparar as formulações da literatura, Pan (1997) realizou uma análise matemática de

diferentes formulações estabelecidas para os ambientes *job shop*, *flow shop* e *flow shop flexível*, tomando como critério de comparação o número de restrições e o número de variáveis binárias e contínuas envolvidas na modelagem dos problemas. As conclusões deste autor indicam que a formulação de Manne (1960) é a melhor formulação para o caso em estudo pelo menor número de variáveis binárias utilizadas.

Mais recentemente, Pham (2008) também realizou uma análise de diferentes formulações estabelecidas para o ambiente de *job shop*. O autor utilizou como referência as mesmas formulações matemáticas que Pan (1997) e efetuou a comparação através da análise do número de restrições e variáveis de cada modelo, além de fazer testes computacionais com 25 problemas-teste que possuem desde 10 tarefas e 5 máquinas até 30 tarefas e 10 máquinas. As conclusões indicam que a formulação de Manne (1960) é com melhor desempenho para o caso em estudo, pelo número de variáveis binárias utilizadas no processo de resolução e pelos resultados computacionais.

Através da análise da literatura, podemos observar que as comparações efetuadas dos modelos PLIM para o problema de *job shop* foram dispersas no tempo, nem sempre apresentaram testes computacionais e, quando realizaram experimentos, utilizaram um número limitado de problemas-teste para chegar às suas conclusões. O objetivo deste trabalho é fazer uma análise comparativa entre diversas formulações matemáticas do tipo linear inteira mista para o ambiente *job shop clássico*, que permita estabelecer as vantagens e desvantagens de cada uma. Dentre as formulações consideradas, cinco são provenientes da literatura e uma é proposta neste estudo. Além disso, com o objetivo de incrementar o número de problemas-teste da literatura com resultados ótimos

e diminuir o tempo computacional para a sua obtenção, são apresentadas três diferentes estratégias de resolução que permitem a exploração de soluções obtidas através de diferentes metodologias.

Cabe mencionar que a importância do estudo de formulações matemáticas buscando a modelagem e simulação de sistemas de produção advém do fato de esta abordagem garantir a solução ótima do problema para as aplicações teóricas e práticas de pequeno porte. Outro aspecto relevante decorre do fato de o método *branch and bound*, um dos mais populares para a resolução de modelos de programação inteira, permitir a utilização de diversas técnicas que reduzem o esforço computacional, como relaxações lineares, relaxações de Lagrange e cortes de Gomory, que usam como base a formulação do problema.

A próxima seção contém a descrição detalhada das formulações matemáticas da literatura consideradas neste trabalho. A Seção 3 apresenta uma nova proposta de modelagem para o ambiente estudado, e a Seção 4, a ordem de grandeza das formulações, os problemas-teste utilizados nos experimentos e os resultados computacionais obtidos. Na Seção 5 são propostas diversas estratégias de melhoria para a resolução do problema e seus correspondentes resultados. Os principais resultados e conclusões estão resumidos na Seção 6.

2. Modelagem matemática do problema

Nesta seção serão descritas distintas formulações de programação linear inteira mista da literatura propostas pelos seguintes autores: Wagner (1958), Wilson (1989), Manne (1960) e Liao & You (1992). A seguir apresenta-se a notação utilizada.

Índices

n número total de tarefas

m número de máquinas

i tarefa i (*job*)

k máquina k

l operação l

j posição j

Parâmetros

p_{ik} tempo de processamento da tarefa i na máquina k

r_{ilk} indica se a operação l da tarefa i requer a máquina k ; observe que $\sum_{k=1}^m r_{ilk} = 1 \forall i \forall l$

M número suficientemente grande

Variáveis

h_{jk} instante de início do processamento da tarefa na posição j na sequência da máquina k

s_{ik} instante de início da tarefa i na máquina k

I_{1k} tempo ocioso (*Idle time*) da máquina k entre o instante de início da produção e o instante de início da primeira tarefa na primeira posição na máquina k .

I_{jk} tempo ocioso (*Idle time*) da máquina k entre o instante de término da tarefa na posição $(j-1)$ da sequência e o instante de início da tarefa na posição j para $j=2,3,\dots,m$

C_{max} instante de término da última tarefa a ser processada no ambiente (*makespan*)

$$Z_{iuk} \begin{cases} 1 \text{ se a tarefa } i \text{ é sequenciada na posição } j \\ \text{na máquina } k \\ 0 \text{ caso contrário} \end{cases}$$

$$Z_{iik} \begin{cases} 1 \text{ se a tarefa } i \text{ precede a tarefa } \\ \text{(não necessariamente de imediato) na máquina } k \\ 0 \text{ caso contrário} \end{cases}$$

2.1. Modelo de Wagner

O modelo matemático proposto por Wagner (1958) é denominado como formulação do tipo designação e se caracteriza por estabelecer em cada máquina um número finito de posições (normalmente definido como o número total de tarefas n) para fixar cada operação de cada tarefa na posição da máquina correspondente e obter a sequência da produção. A formulação apresenta quatro particularidades importantes que são traduzidas nas restrições e variáveis do modelo: *i*) designação das tarefas nas posições, *ii*) definição do instante de início da produção, *iii*) existência de tempo ocioso entre posições consecutivas em uma máquina e *iv*) definição do instante de início de cada tarefa em cada posição na máquina. O modelo é apresentado a seguir.

$$\text{minimizar } C_{max} \quad (1)$$

s.a.

$$\sum_{j=1}^n X_{ijk} = 1 \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (2)$$

$$\sum_{i=1}^n X_{ijk} = 1 \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (3)$$

$$h_{1k} = I_{1k} \quad k = 1, 2, \dots, m \quad (4)$$

$$h_{jk} = \sum_{i=1}^j I_{ik} + \sum_{i=1}^{j-1} \sum_{k=1}^n p_{ik} X_{ik} \quad j = 2, 3, \dots, n \quad k = 1, 2, \dots, m \quad (5)$$

$$\sum_{k=1}^m r_{ilk} (h_{jk} + p_{ik}) \leq \sum_{k=1}^m r_{i,l+1,k} h_{wk} + M \left(1 - \sum_{k=1}^m r_{ilk} X_{ijk} \right) + M \left(1 - \sum_{k=1}^m r_{i,l+1,k} X_{iwk} \right) \quad i, j, w = 1, 2, \dots, n; l = 1, 2, \dots, m-1 \quad (6)$$

$$h_{nk} + \sum_{i=1}^n p_{ik} X_{ink} \leq C_{max} \quad k = 1, 2, \dots, m \quad (7)$$

$$h_{jk}, I_{jk} \geq 0 \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (8)$$

$$X_{ijk} \in \{0,1\} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (9)$$

O conjunto de restrições do tipo (2) estabelece que cada tarefa pode ser alocada somente uma vez em cada máquina. O conjunto de restrições do tipo (3) garante que cada posição em cada máquina só pode ter uma tarefa. As restrições do tipo (4) asseguram que o instante de início da tarefa na primeira posição numa determinada máquina seja igual ao tempo ocioso (*idle time*) transcorrido entre o instante de início da produção e o instante de início do processamento da primeira tarefa na primeira posição naquela máquina. As restrições do tipo (5) definem que o instante de início de uma posição (diferente da inicial) numa máquina seja igual à soma dos tempos ociosos decorridos na máquina desde o início da produção até a posição avaliada e dos tempos de processamento de todas as tarefas alocadas nas posições anteriores.

As restrições do tipo (6) estabelecem para uma determinada tarefa *i* a sequência dos instantes de início de duas operações consecutivas, ou seja, o início de processamento da operação *l+1* da tarefa *i* na posição *w* da máquina indicada para essa operação só pode começar quando o processamento da operação *l* na posição *j* na máquina correspondente a essa operação seja concluída. Na restrição, pode-se observar que quando $X_{ijk_1} = 1$ e $X_{iwk_2} = 1$ (note que a máquina é diferente para cada operação), temos que $h_{jk_1} + p_{ik_1} \leq h_{wk_2}$, garantindo assim que o processamento de uma tarefa na posição *w* só comece depois do término de processamento da tarefa na posição *j* (operações *l+1* e *l* da tarefa *i* respectivamente), ou em outros termos, garantindo a não sobreposição de operações de uma mesma tarefa. A restrição (7) calcula o instante de término de todas as tarefas em todas as posições de cada máquina e determina o *makespan*. As restrições (8) e (9) estabelecem as variáveis como não negativas e binárias respectivamente.

2.2. Modelo de Wilson

A formulação de Wilson (1989) pertence à família de formulações de Wagner (1958), que estabelece para cada máquina um número finito de posições para programar cada operação de cada tarefa e obter a sequência de produção. Esta formulação apresenta uma alternativa baseada na relação de precedência entre tarefas consecutivas em uma mesma máquina, eliminando o conceito de instante de início de produção e a quantificação do tempo ocioso entre posições consecutivas numa mesma máquina. Assim, no modelo teremos a seguinte restrição:

$$h_{jk} + \sum_{i=1}^n p_{ik} X_{ijk} \leq h_{j+1,k} \quad j = 1, 2, \dots, n-1 \quad k = 1, 2, \dots, m \quad (10)$$

As restrições do tipo (10) impõem que para cada máquina o instante de início da tarefa na posição *j+1* seja maior ou igual ao instante de início da tarefa na posição anterior *j* mais o tempo de processamento da tarefa *i* estabelecida na posição *j*. As restrições do tipo (10) substituem as restrições (4) e (5) estabelecidas no modelo de Wagner. Esta formulação reduz tanto o número de restrições necessárias quanto o número de variáveis contínuas utilizadas no modelo anterior.

2.3. Modelo de Manne

A formulação de Manne (1960) é denominada como de precedência e se caracteriza por utilizar restrições disjuntivas para indicar a precedência entre qualquer par de operações designadas numa mesma máquina. Esta formulação reduz de forma significativa o número de restrições e o número de variáveis utilizadas na modelagem do problema em comparação com as formulações de Wagner (1958) e Wilson (1989).

$$\text{minimizar } C_{max} \quad (11)$$

s.a.

$$\sum_{k=1}^m r_{ilk} (s_{ik} + p_{ik}) \leq \sum_{k=1}^m r_{i,l+1,k} s_{ik} \quad i = 1, 2, \dots, n \quad l = 1, 2, \dots, m-1 \quad (12)$$

$$(M + p_{uk}) Z_{iuk} + (s_{ik} - s_{uk}) \geq p_{uk} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (13)$$

$$(M + p_{ik})(1 - Z_{iuk}) + (s_{uk} - s_{ik}) \geq p_{ik} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (14)$$

$$\sum_{k=1}^m r_{imk} (s_{ik} + p_{ik}) \leq C_{max} \quad i = 1, 2, \dots, n \quad (15)$$

$$s_{ik} \geq 0 \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (16)$$

$$Z_{iuk} \in \{0,1\} \quad i, u = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (17)$$

As restrições do tipo (12) estabelecem para determinada tarefa i a precedência dos instantes de início das operações nas máquinas correspondentes, ou seja, o instante de início da operação $l+1$ da tarefa i numa máquina k deve ser maior ou igual ao instante de início da operação l desta tarefa na máquina correspondente a essa operação mais o seu tempo de processamento.

As restrições (13) e (14) são as denominadas restrições disjuntivas e determinam a precedência das tarefas em uma mesma máquina. As restrições (15) determinam o instante de término da última operação de cada tarefa, o qual é denominado *makespan*. As restrições (16) e (17) estabelecem as variáveis como não negativas e binárias, respectivamente.

Baker (1974), Ravindran et al. (1987) e Liao & You (1992) definem o parâmetro M ou *big-M* como um número muito grande e concluem que a soma do parâmetro p_{ik} não afeta o desempenho e pode ser retirado da formulação. Dessa forma, as restrições (13) e (14) da formulação de Manne (1960) podem ser modificadas da seguinte forma:

$$M Z_{iuk} + (s_{ik} - s_{uk}) \geq p_{uk} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (18)$$

$$M(1 - Z_{iuk}) + (s_{uk} - s_{ik}) \geq p_{ik} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (19)$$

Para efeitos deste estudo, será denominada formulação Adaptada de Manne (AM) a formulação de Manne com a modificação das restrições disjuntivas (18) e (19) apresentadas.

2.4. Modelo de Liao-You

A formulação proposta por Liao & You (1992) pertence à família de formulações que utiliza variáveis que indicam precedência para a resolução do problema, e foi desenvolvida com a finalidade de reduzir o número de restrições e o tempo computacional de resolução do modelo. A formulação se baseia na formulação Adaptada de Manne (1960) e contém uma modificação das restrições detalhada a seguir. Reescrevendo as restrições (18) e (19), temos que:

$$s_{ik} - s_{uk} - p_{uk} + M Z_{iuk} \geq 0$$

$$-s_{ik} + s_{uk} + p_{uk} - M Z_{iuk} \geq p_{uk} + p_{ik} - M$$

Definindo:

$$q_{iuk} = s_{ik} - s_{uk} - p_{uk} + M Z_{iuk} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (20)$$

As desigualdades (18) e (19) se reduzem a:

$$0 \leq q_{iuk} \leq M - p_{ik} - p_{uk} \quad 1 \leq i < u \leq n \quad k = 1, 2, \dots, m \quad (21)$$

As restrições (20) e (21) determinam o processamento de só uma tarefa em uma máquina em qualquer instante de tempo. Neste caso, os autores Liao & You (1992) estabelecem uma variável auxiliar $q_{iuk} \geq 0$ para cada desigualdade encontrada na restrição (18) da formulação Adaptada de Manne, enquanto as restrições (21) estabelecem um limitante inferior e um limitante superior para estas variáveis. Dessa forma, a formulação permite, na investigação da árvore de busca do método *branch and bound*, o uso do método simplex canalizado. Observe que a nova formulação reduz uma restrição para cada par de tarefas em cada máquina, mas incrementa o número de variáveis contínuas na mesma dimensão.

3. Modelo proposto

Com a finalidade de desenvolver uma formulação matemática que auxilie a simulação do ambiente e permita encontrar melhores soluções, propõe-se nesta seção uma nova formulação que pertence ao grupo de formulações que designam operações em posições e utiliza a relação simplificada de relações de posições de Wilson (1989).

A formulação apresenta uma alternativa para o estabelecimento da precedência de operações de uma mesma tarefa, com a ajuda de uma variável real que estabelece o instante de início da operação na máquina. Dessa forma, ao se obter os valores de início de processamento de uma operação numa máquina e os valores de início de cada posição da máquina, a variável de decisão binária deverá decidir qual posição da máquina corresponde ao instante de início da operação, produzindo-se assim uma restrição do tipo “*se-então*”. O modelo é apresentado a seguir.

$$\text{minimizar } C_{\max} \quad (22)$$

s.a.

$$\sum_{j=1}^n X_{ijk} = 1 \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (23)$$

$$\sum_{i=1}^n X_{ijk} = 1 \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (24)$$

$$h_{jk} + \sum_{i=1}^n p_{ik} X_{ijk} \leq h_{j+1,k} \quad j = 1, 2, \dots, n-1 \quad k = 1, 2, \dots, m \quad (25)$$

$$\sum_{k=1}^m r_{i,l+1,k} s_{ik} \geq \sum_{k=1}^m r_{i,l,k} (s_{ik} + p_{ik}) \quad (26)$$

$$i = 1, 2, \dots, n; \quad l = 1, 2, \dots, m-1$$

$$M(1 - \sum_{k=1}^m r_{ilk} X_{ijk}) \geq \sum_{k=1}^m r_{i,l,k} (s_{ik} - h_{jk}) \quad (27)$$

$$i, j = 1, 2, \dots, n; \quad l = 1, 2, \dots, m$$

$$M(1 - \sum_{k=1}^m r_{ilk} X_{ijk}) \geq \sum_{k=1}^m r_{i,l,k} (h_{jk} - s_{ik}) \quad (28)$$

$$i, j = 1, 2, \dots, n; l = 1, 2, \dots, m$$

$$h_{nk} + \sum_{i=1}^n p_{ik} X_{ink} \leq C_{max} \quad k = 1, 2, \dots, m \quad (29)$$

$$h_{jk} \geq 0 \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (30)$$

$$s_{ik} \geq 0 \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (31)$$

$$X_{ijk} \in \{0, 1\} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (32)$$

As restrições do tipo (23) estabelecem que cada tarefa pode ser alocada somente uma vez em cada máquina. As restrições do tipo (24) garantem que cada posição em cada máquina só pode realizar uma tarefa. A restrições do tipo (25) asseguram que para cada máquina o instante de início da posição $j+1$ seja maior ou igual ao instante de início da posição anterior j mais o tempo de processamento da tarefa i estabelecida na posição j .

As restrições do tipo (26) estabelecem para determinada tarefa i a sequência dos instantes de início de duas operações consecutivas, ou seja, o início de processamento da operação $l+1$ da tarefa i na máquina indicada para essa operação só pode acontecer quando for concluído o processamento da operação anterior l da tarefa i na máquina correspondente a essa operação.

As restrições do tipo (27) e (28) estabelecem que para cada tarefa i e cada operação l deve existir uma posição j estabelecida na máquina k que corresponda a essa operação, de tal forma que o instante de início h dessa posição deverá ser igual ao instante de início s da tarefa i na mesma máquina, ou seja: $X_{ijk} = 1 \rightarrow h_{jk} = s_{ik}$. A restrição (29) calcula o instante de término da última posição de cada máquina denominada como *makespan*. As restrições (30), (31) e (32) estabelecem as variáveis como não negativas e binárias.

4. Análise das formulações

A seguir resumem-se as principais características das formulações apresentadas. A Tabela 1 apresenta três parâmetros: número de restrições, número de variáveis binárias e número de variáveis contínuas. As dimensões das formulações da literatura também podem ser encontradas em Pan (1997).

Observa-se na análise da Tabela 1 que as formulações do tipo designação têm um número igual ou maior de restrições, de variáveis binárias e variáveis contínuas que as formulações de precedência. A formulação de Liao-You apresenta um menor número de restrições e um maior número de variáveis contínuas em comparação com as formulações de Manne e sua forma adaptada. A formulação Proposta apresenta o menor número de restrições (supondo $n \geq 5$ e $m \geq 2$) dentre as formulações do tipo designação, seguida das formulações de Wilson & Wagner.

Neste ponto, cabe ressaltar que o número de variáveis e restrições não deve ser o único aspecto considerado para avaliar a dificuldade de resolução dos problemas representados pelas formulações. Pan (1997) alega que, para os ambientes *job shop* e *flow shop*, o fator mais relevante na resolução dos respectivos modelos PLIM é o número de variáveis binárias. Porém, Stafford et al. (2004) e Ronconi & Birgin (2012) concluem que somente este fator não é suficiente para determinar este nível de dificuldade. Estes autores apresentam, para diferentes ambientes do tipo *flow shop*, experimentos computacionais com *softwares* conhecidos de otimização que indicam que o uso das formulações de designação, quando comparadas com formulações de precedência, requer um menor tempo computacional para encontrar soluções ótimas. Adicionalmente, conforme Arenales et al. (2007) e Öncan et al. (2009), a proximidade do poliedro associado a uma formulação PLIM em relação a envoltória convexa do espaço de soluções beneficia

Tabela 1. Dimensões das formulações.

	Formulação	Número de restrições	Número de variáveis binárias	Número de variáveis contínuas
Precedência	Manne	mn^2	$mn(n-1)/2$	$mn+1$
	AM	$mn^2 - n$	$mn(n-1)/2$	$mn+1$
	Liao-You	$mn(n+1)/2$	$mn(n-1)/2$	$mn(n+1)/2+1$
Designação	Wagner	$n^3(m-1)+3mn+m$	mn^2	$2mn+1$
	Wilson	$n^3(m-1)+3mn$	mn^2	$mn+1$
	Proposta	$2mn^2+4mn-n$	mn^2	$2mn+1$

Fonte: Elaborada pelos autores.

a aplicação mais eficiente de métodos como o *branch and cut*, que é utilizado, por exemplo, no *software* de otimização CPLEX.

Com o intuito de determinar a formulação com melhor desempenho para o ambiente *job shop*, a seguir as distintas formulações serão avaliadas através de testes computacionais em problemas extraídos da literatura.

4.1. Experimentos computacionais

Com o objetivo de estabelecer uma comparação entre as formulações matemáticas e determinar a eficiência de cada uma para a resolução de problemas, as diversas formulações apresentadas foram resolvidas através do método *branch and cut* do *software* CPLEX v.12.2 com a interface *OPL Studio IDE Academic Research*. Foram utilizados 45 problemas-teste diferentes extraídos do trabalho de Lawrence (1984) e dos autores Adams et al. (1988), que variam desde 10 tarefas com 5 máquinas até 20 tarefas com 15 máquinas. Os problemas considerados estão disponíveis na *OR-Library* (Beasley, 2012). Os testes foram realizados em um computador *Intel Core i7 2,93 GHz* com 16 Gb de memória RAM e um limite máximo de tempo de resolução de 10 min/problema.

Para cada exemplar, o valor de M ou *big-M* foi definido como $M = \sum_{i=1}^n \sum_{k=1}^m p_{ik}$, dado que satisfaz todas as restrições das diferentes formulações e, ao mesmo tempo, não permite que nós não necessários sejam criados na busca da solução ótima. Deve-se considerar a seguinte notação para todas as tabelas que apresentam os resultados obtidos:

Des_U - Desvio da melhor solução encontrada (UB - *upper bound*) em relação ao valor ótimo:

$$Des_U = \frac{UB - OTM}{OTM} \times 100$$

Des_L - Desvio do limitante inferior (LB - *lower bound*) em relação ao valor ótimo:

$$Des_L = \frac{OTM - B}{OTM} \times 100$$

GAP - *Gap* de otimalidade:

$$GAP = \frac{UB - LB}{LB} \times 100$$

A Tabela 2 apresenta os valores médios das medidas Des_U , Des_L e GAP associados a cada uma das formulações avaliadas, considerando os 45 problemas do conjunto teste, além do número de soluções ótimas e de soluções com *gap* de otimalidade menor que 5%. Adicionalmente, a Tabela 3 exibe o número médio de cada tipo de corte gerado pelo método *branch and cut* durante a resolução das formulações analisadas. A primeira coluna desta tabela contém o tipo de corte conforme a denominação utilizada pelo CPLEX. Mais detalhes podem ser encontrados em IBM (International Business Machines, 2014a).

Na análise da Tabela 2 nota-se que o *software* de otimização, considerando as formulações de precedência, obtém limitantes superiores de melhor qualidade e menores *gaps* de otimalidade, além de fornecer o maior número de soluções ótimas dentro do limite de tempo estabelecido. Por outro lado, através das formulações do tipo designação são alcançados melhores resultados para o limitante inferior. Acredita-se que o baixo desempenho para a obtenção dos limitantes inferiores apresentado pelo *software* utilizando a formulação de precedência baseia-se no fato de as relaxações lineares de modelos com base em restrições disjuntivas normalmente fornecerem limitantes duais (neste caso, limitantes inferiores) de baixa qualidade (veja Koné et al. (2013) e a discussão em Raman & Grossmann (1994) e suas referências).

Dentre as formulações de precedência, a resolução da formulação de Manne apresenta os melhores

Tabela 2. Resultados computacionais das formulações.

	Formulação	Des _U %	Des _L %	GAP %	Nº de soluções ótimas	Nº de soluções com 0% < GAP < 5%
Precedência	Manne	0,83	24,74	46,01	12	0
	AM	1,49	24,74	47,32	12	0
	Liao-You	3,53	26,69	56,34	12	0
Designação	Wagner	351,77	9,36	413,95	1	1
	Wilson	336,17	9,33	395,53	1	2
	Proposta	197,69	8,48	234,88	3	5

Tabela 3. Número médio de cortes gerados para cada formulação.

Tipo de corte	Precedência			Designação		
	Manne	AM	Liao	Wagner	Wilson	Proposta
Implied bound cuts	430,02	675,91	707,82	12,00	18,60	301,80
Flow cuts	72,18	129,60	207,69	3,83	2,35	9,30
Mixed integer rounding cuts	459,93	824,11	1924,16	25,68	17,92	146,73
Zero-half cuts	0,00	16,56	12,05	0,00	0,00	0,00
Lift and project cuts	35,41	4,47	8,07	0,00	1,00	2,13
Gomory fractional cuts	39,60	23,93	83,22	2,40	2,64	21,17
Cover cuts	6,13	15,10	45,16	1,63	1,21	1,14
GUB cover cuts	2,22	3,80	0,00	1,00	0,00	0,00
Clique cuts	0,00	4,40	3,47	0,00	0,00	0,00
Multi commodity flow cuts	0,00	2,00	1,14	0,00	0,00	3,59
Total	1.045,50	1.699,88	2.992,78	46,54	43,72	485,86

Fonte: Elaborada pelos autores.

resultados em todos os indicadores. A utilização da formulação AM obtém piores resultados que a formulação original de Manne, indicando que a adaptação sugerida por Baker (1974) e outros autores nem sempre é favorável para o problema e que, como comentado anteriormente, o valor do parâmetro M pode afetar o processo de resolução do problema. De qualquer forma, embora haja uma diferença relativa considerável nos desvios médios do limitante superior ao valor ótimo encontrado (Des_{μ}) ao resolver cada um dos três modelos de precedência, as diferenças absolutas são pouco significativas e a quantidade de soluções ótimas encontradas em cada caso é a mesma. A análise da quantidade média de cada tipo de corte gerado pelo método *branch and cut* durante a resolução das formulações analisadas (veja a Tabela 3) não mostra uma dominância clara de uma das formulações em relação à quantidade de cortes gerados de cada tipo. Podemos, portanto, concluir que esta análise empírica não nos permite tecer uma relação clara entre a quantidade de cortes gerados e a qualidade dos limitantes obtidos por cada um dos três modelos baseados em precedências.

Considerando as formulações de designação, a Tabela 2 mostra que a resolução das formulações de Wagner e Wilson apresenta resultados muito similares, enquanto a formulação Proposta apresenta os melhores limitantes e, conseqüentemente, o menor *gap*. Analisando a Tabela 3, nota-se que o número médio de cortes gerados durante a resolução das formulações de precedência é significativamente superior ao número de cortes gerados durante a resolução das formulações de designação. Diferentemente da análise envolvendo as formulações de precedência, ao analisar o número de cortes gerados na resolução das formulações de designação, nota-se que em 8 dos 10 tipos de cortes a quantidade de cortes gerados para a formulação Proposta é notavelmente superior

ou igual à quantidade gerada para as outras duas formulações e, em média, esta quantidade é mais do que 10 vezes maior.

5. Estratégias de resolução

Buscando melhorar os resultados obtidos com a resolução dos modelos em estudo, são apresentadas a seguir duas diferentes estratégias de resolução que permitem a integração de soluções obtidas através de diferentes metodologias. Na próxima subseção apresenta-se a primeira estratégia que estabelece para cada problema-teste uma solução heurística inicial no intuito de reduzir o número de combinações a serem avaliadas pelo *software*. A segunda estratégia sugere diferentes processos que utilizam as melhores formulações e/ou procedimentos de forma consecutiva.

5.1. Estratégia 1: Inserção de uma solução inicial

As soluções iniciais foram geradas por meio de heurísticas construtivas que têm como principal característica a capacidade de atingir soluções de boa qualidade com tempos computacionais moderados. Embora não possam garantir encontrar a solução ótima, as heurísticas construtivas são adequadas para problemas de alta complexidade, como é o caso do *job shop*. As heurísticas construtivas mais populares para o problema em estudo são as denominadas regras de despacho, que, segundo French (1982) e Pinedo (2008), são de fácil implementação e reduzem substancialmente o tempo computacional requerido para encontrar soluções de boa qualidade.

Inicialmente a solução factível gerada para acelerar o processo de resolução das formulações foi obtida com a aplicação da regra de despacho SPT (*Shortest Processing Time*). Chang et al. (1996)

avaliaram 42 regras de despacho para o ambiente *job shop* e concluíram que a regra *SPT* obtém resultados consistentes. O procedimento utilizado, denominado HC1, é baseado no algoritmo para a geração de uma solução inicial descrito em Armentano et al. (2013). Neste procedimento, em todo instante T no qual existe uma máquina disponível para processar uma tarefa, programa-se a tarefa disponível com maior prioridade para ser processada naquela máquina, sendo esta prioridade dada por meio da regra de despacho considerada, neste caso a regra *SPT*.

Conforme Baker & Trietsch (2009), as regras de despacho mais bem-sucedidas para a minimização do *makespan* favorecem tarefas com maior tempo de processamento restante no ambiente. Por tal motivo, a segunda heurística, denominada HC2, considera para a alocação de tarefas, além da regra *SPT*, a regra *MWR (Most Work Remaining)*. Outra característica deste procedimento, também presente no trabalho Mainieri & Ronconi (2013), é a consideração de tarefas que estarão disponíveis em instantes de tempo posteriores aos instantes de disponibilidade das máquinas para possível alocação. Nessa heurística, assim como em HC1, em todo instante T , no qual existe uma máquina disponível, programa-se a tarefa disponível com maior prioridade para ser processada naquela máquina, sendo esta prioridade dada pela regra de despacho *MWR*. Caso não existam tarefas disponíveis no instante considerado, estabelece-se uma lista segundo a regra de despacho *MWR* das tarefas que na próxima operação serão processadas nessa máquina e se aloca a tarefa j com maior prioridade. Note que neste caso ocorre a inclusão de tempo ocioso. O procedimento preenche este tempo ocioso com tarefas que possam ser alocadas respeitando o início programado anteriormente da tarefa j . Se existirem tais tarefas, estas devem ser programadas segundo a regra de despacho *SPT*, até que a alocação não seja mais plausível.

As heurísticas propostas foram implementadas no software *Microsoft Visual Studio C++ 2010 Express* e

avaliadas através da resolução dos 45 problemas-teste descritos na Seção 4.1. O tempo de execução da heurística para cada exemplar foi inferior a 2 segundos, e os principais indicadores de desempenho são apresentados a seguir. O desvio médio da solução ótima foi calculado utilizando-se a medida Des_u (veja Seção 4.1), considerando o valor da solução heurística como o limitante superior do valor ótimo. Note que a heurística construtiva 2 apresenta um desempenho superior em todos os indicadores considerados.

- HC1:

- desvio médio da solução ótima: 40,41%,
- maior desvio do ótimo: 76,22%,
- menor desvio do ótimo: 7%,
- número de soluções ótimas encontradas: 0.

- HC2:

- desvio médio da solução ótima: 14,4%,
- maior desvio do ótimo: 34,24%,
- menor desvio do ótimo: 0%,
- número de soluções ótimas encontradas: 4.

A Tabela 4 apresenta os resultados obtidos através da resolução das formulações analisadas utilizando a solução inicial fornecida pelas heurísticas propostas. Este experimento computacional foi realizado por meio da interface *OPL Studio IDE Academic Research*, que possui recursos que permitem a inserção de uma solução como ponto de partida para a resolução de um modelo PLIM pelo software *Cplex*. Inicialmente, a heurística é executada e a solução obtida é armazenada em um arquivo tipo *xls* (associado à planilha eletrônica Excel). A interface *OPL* importa o valor das variáveis inteiras deste arquivo e fornece esta solução como solução inicial para a resolução pelo *Cplex* da formulação avaliada com limite de tempo de processamento de

Tabela 4. Resultados computacionais das formulações com solução inicial de HC1 e HC2.

Designação	Formulação	HC1			HC2		
		Des_u %	Des_l %	GAP %	Des_u %	Des_l %	GAP %
Precedência	Manne	0,71	24,68	45,43	0,59	24,45	44,53
	AM	0,72	24,68	45,02	0,60	24,33	44,23
	Liao-You	3,41	26,33	54,71	1,78	26,23	50,72
Designação	Wagner	28,68	9,33	45,14	12,06	9,28	26,02
	Wilson	27,61	9,33	43,90	11,39	9,28	25,27
	Proposta	19,52	8,17	32,26	7,86	8,01	19,05

10 minutos. Este procedimento é realizado utilizando-se a função *IloOplCplexVectors*. Mais detalhes podem ser encontrados em IBM (International Business Machines, 2014b).

Observa-se na análise da Tabela 4 que a resolução das formulações avaliadas utilizando a estratégia proposta fornece resultados de qualidade superior quando comparados com os resultados obtidos na Seção 4.1 (veja Tabela 2). Além disso, esta estratégia mostra ter um melhor desempenho quanto melhor é a solução inicial inserida.

Note que a associação da solução inicial heurística com as formulações de precedência não aprimora o seu processo de resolução da mesma forma que ocorre com as formulações de designação. Tal fato era esperado dada a qualidade do limitante superior encontrado pelas formulações de precedência nos testes apresentados na Seção 4.1. Vale destacar que, com o uso desta estratégia, a formulação original e a adaptada de Manne e a formulação Proposta continuam obtendo os melhores resultados dentre as formulações de precedência e designação, respectivamente; porém, diferente dos testes anteriores, dentre todas as formulações avaliadas, as formulações do tipo designação obtêm o menor *gap* de otimalidade.

5.2. Estratégia 2: Associação de modelos ou procedimentos

Nesta seção propomos estratégias adicionais que buscam incorporar as melhores características dos procedimentos até agora propostos em um procedimento único que seja capaz de produzir resultados superiores aos alcançados pelos procedimentos originais. A ideia central é utilizar as formulações de melhor desempenho dentro de cada categoria, Manne (Mn) e Proposta (Pr), e a heurística construtiva 2 (HC) de forma consecutiva por um determinado tempo de processamento. Vários testes preliminares foram efetuados e as cinco melhores combinações denominadas Mn10_Pr10, HC_Mn10_Pr10, Mn_Pr10, Mn8_Pr2 e HC_Mn8_Pr2 são descritas a seguir.

A estratégia Mn10_Pr10 utiliza os procedimentos de forma encadeada, ou seja, os resultados obtidos através da formulação de Manne depois de 10 min de processamento no *software* CPLEX são inseridos na formulação Proposta, que também é processada durante 10 minutos. O procedimento HC_Mn10_Pr10 é semelhante à estratégia anterior, porém utiliza os resultados gerados pela heurística construtiva como solução inicial para a formulação de Manne.

As estratégias Mn_Pr10 e Mn8_Pr2 também utilizam de forma consecutiva as formulações de Manne e Proposta, porém sem a utilização da solução inicial obtida pela heurística construtiva e em um tempo

total de execução de aproximadamente 10 minutos. O primeiro procedimento inicialmente processa a formulação de Manne até a obtenção da primeira solução factível, que é inserida como solução inicial para o processamento (10 min) da formulação Proposta. Por outro lado, o procedimento Mn8_Pr2 processa o modelo de Manne durante 8 min para gerar uma solução que é inserida na formulação Proposta, que a seguir é processada durante 2 min. Essa diferença na distribuição dos tempos de processamento é motivada pelo fato de a resolução da formulação de Manne gerar soluções factíveis de boa qualidade que podem favorecer a obtenção de melhores resultados. A estratégia HC_Mn8_Pr2 é similar à Mn8_Pr2, porém neste caso o procedimento utiliza como solução inicial o resultado da heurística construtiva. Os tempos estabelecidos para cada etapa dos procedimentos foram avaliados em testes preliminares.

A Tabela 5 apresenta o valor médio das medidas Des_u , Des_l e GAP para cada uma das estratégias propostas, além do número de soluções ótimas e de soluções com *gap* de otimalidade menor que 5%. Pode-se observar que, de forma geral, todos os procedimentos obtêm valores significativamente menores para o indicador GAP do que as estratégias anteriormente apresentadas, lembrando que foi utilizado um tempo computacional superior nos procedimentos Mn10_Pr10 e HC_Mn10_Pr10. Cabe destacar que por meio de testes adicionais observamos que o procedimento Mn10_Pr10 obtêm resultados melhores que os obtidos através das formulações de Manne e a Proposta processadas durante 20 minutos.

Comparando os resultados obtidos pelo procedimento HC_Mn10_Pr10 com os resultados obtidos pelo procedimento Mn10_Pr10, pode-se notar que a inserção da heurística construtiva foi efetiva para a geração de melhores limitantes e consequentemente para a redução do *gap* de otimalidade.

Dentre as estratégias executadas em um tempo computacional reduzido, algumas considerações podem ser feitas. Embora a estratégia Mn_Pr10 obtenha um limitante superior melhor do que o obtido pela formulação Proposta associada com HC2, este valor não é melhor do que o limitante obtido pela formulação de Manne. Além disso, o limitante inferior e o *gap* de otimalidade obtidos são melhores que os obtidos pela formulação de Manne associada com HC2, porém não superam os valores encontrados pela formulação Proposta inicializada com HC2.

Analisando os resultados da estratégia Mn8_Pr2, observa-se que todos os indicadores mostraram resultados significativamente superiores aos apresentados pela estratégia Mn_Pr10 indicando a relevância da resolução da formulação de Manne por um período de tempo maior. Conforme esperado, a estratégia HC_Mn8_Pr2 obtêm melhores resultados do que os obtidos através

Tabela 5. Resultados computacionais utilizando estratégias combinadas.

Estratégia	Des _U %	Des _L %	GAP %	Nº. de soluções ótimas	Nº. de soluções com 0% < GAP < 5%
Mn10_Pr10	0,83	8,03	10,92	15	5
HC_Mn10_Pr10	0,23	8,02	10,27	16	5
Mn_Pr10	3,77	10,52	20,53	13	3
Mn8_Pr2	1,10	8,40	11,74	15	4
HC_Mn8_Pr2	0,85	8,40	11,58	15	5

Fonte: Elaborada pelos autores.

da estratégia Mn8_Pr2. Este desempenho reforça a validade da inserção da solução inicial fornecida pela heurística construtiva na formulação de Manne. Vale destacar que o procedimento HC_Mn8_Pr2 obtém melhores resultados que o procedimento Mn8_Pr2 aproximadamente no mesmo tempo computacional.

6. Conclusões

Neste trabalho abordou-se o problema de programação de tarefas no ambiente denominado *job shop* clássico. Foram avaliadas seis formulações matemáticas de diversos autores através de experimentos computacionais com 45 problemas-teste extraídos da literatura. Adicionalmente, foram implementadas estratégias de aprimoramento para melhorar o processo de resolução destas formulações.

Os resultados obtidos mostram que o *software* CPLEX utilizando as formulações de precedência obtém melhores resultados em relação ao número de soluções ótimas para o problema *job shop* do que através do uso das formulações do tipo designação, ao contrário do que acontece em outros ambientes com máquina simples (Baker & Keller, 2010) e *flow shop* (Stafford et al., 2004; Ronconi & Birgin, 2012). Além disso, observa-se que o melhor desempenho é obtido utilizando-se a formulação proposta por Manne (1960), pois através da sua resolução encontra-se o maior número de soluções ótimas no tempo estabelecido e o melhor limitante superior nos diferentes testes computacionais. A resolução da formulação de Liao-You obteve piores resultados do que os obtidos através da formulação de Manne, mostrando que a simplificação que os autores realizaram sobre o número de restrições e o incremento das variáveis contínuas não ajudaram ao processo de solução do problema, comportamento semelhante ao observado por Ronconi & Birgin (2012).

A utilização da formulação proposta neste trabalho apresenta os melhores resultados dentro do grupo de formulações do tipo designação, sugerindo que a utilização da restrição do tipo “*se-então*” em conjunto com as relações de precedência permite aprimorar o processo de exploração dentro da árvore de busca. Enquanto a resolução das formulações de Wagner e Wilson apresenta resultados muito similares, a

resolução da formulação Proposta apresenta os melhores limitantes e, conseqüentemente, o menor *gap* de otimalidade. Além disso, observa-se na Tabela 3 que a quantidade de cortes gerados para a formulação Proposta é notavelmente superior ou igual à quantidade gerada para as outras duas formulações. Cabe ressaltar que a resolução das formulações do tipo designação utilizando a estratégia de inserção de soluções iniciais apresenta, em média, o menor *gap* de otimalidade; tal característica pode ser vantajosa para a resolução de problemas com soluções ótimas desconhecidas.

As estratégias de aprimoramento, quando associadas com as formulações analisadas, de forma geral, melhoraram seus indicadores de desempenho. Os testes mostram que os procedimentos propostos produzem resultados satisfatórios. Em especial, o procedimento HC_Mn10_Pr10 encontra o maior número de soluções ótimas, com resultados, em média, a 0,23% das soluções ótimas. Além disso, esta estratégia apresenta o menor *gap* de otimalidade, 10,27%, dentre todos os procedimentos avaliados. Embora este procedimento seja executado em 20 min, a sua aplicação é promissora. O procedimento HC_Mn8_Pr2 é o melhor procedimento desenvolvido neste trabalho para o ambiente considerado no tempo computacional de 10 minutos. Esta estratégia encontra um maior número de soluções ótimas do que as formulações clássicas, obtém na média resultados a 0,85% da solução ótima e um *gap* de otimalidade de 11,58%.

Futuras pesquisas poderiam ser realizadas no desenvolvimento de novas heurísticas que permitam obter melhores resultados e possibilitem aprimorar com maior intensidade o processo de resolução das formulações.

Referências

- Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3), 391-401. <http://dx.doi.org/10.1287/mnsc.34.3.391>.
- Arenales M., Armentano, V. A., Morabito, R., & Yanasse, H. H. (2007). *Pesquisa operacional*. Rio de Janeiro: Elsevier.
- Armentano, V. A., Ronconi, D. P., Scrich, C. R., & Shiguemoto, A. L. (2013). Busca tabu: implementação de estratégias de memórias de curto e longo prazo. In H. S. Lopes, L. C. A. Rodrigues & M. T. A. Steiner (Orgs.), *Meta-heurísticas em pesquisa operacional* (pp. 367-384). Curitiba: Omnipax.

- Baker, K. R. (1974). Introduction to sequencing and scheduling. *Operations Research*, 23, 62-73. <http://dx.doi.org/10.1287/opre.23.1.62>.
- Baker, K. R., & Keller, B. (2010). Solving the single machine sequencing problem using integer programming. *Computers & Industrial Engineering*, 59(4), 730-735. <http://dx.doi.org/10.1016/j.cie.2010.07.028>.
- Baker, K. R., & Trietsch, D. (2009). *Principles of sequencing and scheduling*. New York: John Wiley and Sons.
- Beasley, J. E. (2012). *OR-Library*. Recuperado em 11 de julho de 2014, de <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>
- Chang, Y. L., Sueyoshi, T., & Sullivan, R. S. (1996). Ranking dispatching rules by data envelopment analysis in a job shop environment. *IIE Transactions*, 28, 631-642.
- French, S. (1982). *Sequencing and scheduling: an introduction to the mathematics of the job shop*. Chichester: Horwood.
- Garey, M., Johnson, D., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117-129. <http://dx.doi.org/10.1287/moor.1.2.117>.
- International Business Machines. (2014a). *Branch and cut in CPLEX*. Recuperado em 11 de julho de 2014, de <http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r4/index.jsp?topic=%2Flog.odms.cplex.help%2Frefcpcplex%2Fhtml%2Fbranch.html>
- International Business Machines. (2014b). *Setting the initial solution*. Recuperado em 11 de julho de 2014, de http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r4/index.jsp?topic=%2Flog.odms.ide.help%2FOPL_Studio%2Fopllanguser%2Ftopics%2Fuss_langtut_fc_default_behavior_6.html
- Koné, O., Artigues, C., Lopez, P., & Mongeau, M. (2013). Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. *Flexible Services and Manufacturing Journal*, 25(1-2), 25-47. <http://dx.doi.org/10.1007/s10696-012-9152-5>.
- Lawrence, S. (1984). *Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques* (Relatório técnico). Pittsburgh: Carnegie Mellon University.
- Liao, C. J., & You, C. T. (1992). An improved formulation for the job shop scheduling problem. *Operational Research Society*, 11(11), 1047-1054. <http://dx.doi.org/10.1057/jors.1992.162>.
- Maimieri, G., & Ronconi, D. P. (2013). New heuristics for total tardiness minimization in a flexible flowshop. *Optimization Letters*, 7(4), 665-684. <http://dx.doi.org/10.1007/s11590-012-0448-x>.
- Manne, A. S. (1960). On the job shop scheduling problem. *Operations Research*, 8(2), 219-223. <http://dx.doi.org/10.1287/opre.8.2.219>.
- Öncan, T., Altinel, I. K., & Laportec, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3), 637-654. <http://dx.doi.org/10.1016/j.cor.2007.11.008>.
- Pan, C. H. (1997). A study of integer programming formulations for scheduling problems. *International Journal of System Science*, 28(1), 33-41. <http://dx.doi.org/10.1080/00207729708929360>.
- Pham D.-N. (2008). *Complex job shop scheduling: formulations, algorithms and healthcare application* (Tese de doutorado). University of Fribourg, Suíça.
- Pinedo, M. (2008). *Theory, algorithms and systems*. New York: Springer.
- Raman, R., & Grossmann, I. E. (1994). Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering*, 18(7), 563-578. [http://dx.doi.org/10.1016/0098-1354\(93\)E0010-7](http://dx.doi.org/10.1016/0098-1354(93)E0010-7).
- Ravindran, A., Phillips, D. T., & Solberg, J. J. (1987). *Operations research: principles and practice*. New York: John Wiley & Sons.
- Ronconi, D. P., & Birgin, E. G. (2012). Mixed-Integer programming models for flow shop scheduling problems minimizing the total earliness and tardiness, in *Just-in-Time systems*. In Y. A. Ríos-Solis & R. Z. Ríos-Mercado (Eds.), *Springer series on optimization and its applications* (pp. 91-105). New York: Springer.
- Stafford, E. F., Tseng, F. T., & Gupta, J. N. D. (2004). Comparative evaluation of MILP flow shop models. *The Journal of the Operational Research Society*, 56(1), 88-101. <http://dx.doi.org/10.1057/palgrave.jors.2601805>.
- Wagner, H. M. (1958). An integer linear programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2), 131-140. <http://dx.doi.org/10.1002/nav.3800060205>.
- Wilson, J. M. (1989). Alternative formulations of flow shop scheduling problem. *Operational Research Society*, 4(4), 395-399. <http://dx.doi.org/10.1057/palgrave.jors.0400410>.

Agradecimentos

Os autores são gratos aos revisores anônimos pelos seus úteis comentários e sugestões que permitiram o aprimoramento deste artigo. Esta pesquisa teve o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq – Processo 133659/2010-0) e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP – Processos 2010/10133-0 e 2013/07375-0).

Mathematical models and resolution strategies for the classical job shop problem

Abstract

The scheduling problem in a job shop production environment is characterized by containing n jobs to be processed by m machines, where each job is represented by a specific sequence of operations with an established precedence order. The aim of this work is to perform a comparative analysis of the mathematical formulations for this environment by minimizing the makespan, *i.e.*, the total time to complete all n jobs. Popular models and a proposed model are compared and evaluated through computational tests using cases from the literature. In addition, resolution strategies are proposed. Computational experiments using a well-known commercial software package indicate that the proposal strategies can promote a reduction of the optimality gap.

Keywords:

Job shop. Scheduling. Makespan. Mixed integer linear programming models.