 ISSN 1984-820X

Formulações de Programação Matemática para o Problema de Seqüenciamento em uma Máquina com Janelas de Entrega Distintas e Tempo de Preparação Dependente da Seqüência de Produção

Bruno F. Rosa, **Sérgio R. de Souza**,
Programa de Pós-graduação em Modelagem Matemática e Computacional, CEFET-MG,
30510-000, Belo Horizonte, MG
E-mail: brunofazmat@dppg.cefetmg.br, sergio@dppg.cefetmg.br

Marcone J. F. Souza
UFOP - Departamento de Computação
Campus Universitário
35.400-000, Ouro Preto, MG
E-mail: marcone@iceb.ufop.br.

Resumo: *Este trabalho trata do problema de seqüenciamento em uma máquina em que o tempo de preparação depende da seqüência de produção. No problema abordado, cada tarefa possui uma janela de tempo, dentro da qual deve ser preferencialmente concluída. O objetivo a ser satisfeito é minimizar a soma ponderada dos atrasos e antecipações na produção de tais tarefas. Duas novas formulações de programação matemática são propostas para representar o problema, sendo a primeira delas um aperfeiçoamento de uma formulação da literatura, e, a outra, uma formulação indexada no tempo. Experimentos computacionais mostram que a formulação indexada no tempo possibilita encontrar soluções de melhor qualidade em menor tempo que as outras duas formulações.*

Palavras-chave: *Seqüenciamento em Uma Máquina, Penalidades por Antecipação e Atraso, Modelagem Matemática Indexada no Tempo.*

1 Introdução

O problema de seqüenciamento em uma máquina com penalidades por antecipação e atraso da produção (PSUMAA) consiste em seqüenciar e determinar o momento em que as tarefas devem ser executadas em uma máquina, com o objetivo de minimizar a soma ponderada das antecipações e dos atrasos na produção de tais tarefas. Neste problema, cada tarefa está associada a um tempo de processamento e a uma data desejada para sua entrega. Além disso, após a conclusão de uma tarefa, é necessário preparar a máquina para a execução da tarefa seguinte.

Segundo [1], o tempo de preparação da máquina, também conhecido como tempo de *setup*, aparece em um grande número de aplicações do PSUMAA. Este tempo de preparação ocorre principalmente em empresas que produzem diversos tipos de produtos, sendo, assim, necessária a troca do tipo de tarefa executada na máquina. Nele são incluídos os tempos gastos para trocar as ferramentas, preparar os materiais, limpar a máquina, dentre outras questões. A maioria dos trabalhos em problemas de seqüenciamento assume que os tempos de *setup* são independentes da seqüência de produção, ou seja, que esses tempos são ou desprezíveis ou podem ser acrescentados aos tempos de processamento das tarefas. Apesar disso, em grande parte das situações práticas, os tempos de *setup* são dependentes da seqüência de produção [6].

As datas de entrega podem ser comuns (*common due date*) ou distintas (*distinct due dates*). Uma abordagem genérica do PSUMAA considera a existência de janelas de entrega distintas, em que há um determinado período de tempo associado a cada tarefa, dentro do qual a tarefa deve ser preferencialmente concluída. De acordo com [9], este último caso ocorre quando existem tolerâncias em torno das datas desejadas para a entrega de cada tarefa. Deste modo, as tarefas concluídas dentro de suas respectivas janelas de entrega não ocasionam penalidades, ao contrário daquelas concluídas fora de suas janelas de entrega.

Existem trabalhos sobre o PSUMAA nos quais a ociosidade da máquina não é permitida. Conforme [7], há situações em que manter a máquina parada é mais custoso que a antecipação de uma tarefa, ou outras, em que a demanda é superior à capacidade da máquina. Por outro lado, [4] afirma que há casos em que manter a máquina inativa é vantajoso, ainda que exista uma tarefa disponível para processamento.

Na literatura, são encontrados muitos trabalhos com o objetivo de resolver o PSUMAA. Isto se deve ao fato de ser um problema com muitas aplicações industriais, de um lado, e, de outro, difícil de ser resolvido na otimalidade, dado que se trata de um problema NP-difícil [1].

Neste trabalho, é estudado o PSUMAA com janelas de entrega distintas, tempo de preparação da máquina dependente da seqüência de produção, sendo permitidos tempos ociosos entre as execuções de tarefas consecutivas. Apesar de ser uma generalização do PSUMAA, esta versão tem recebido pouca atenção. Dentre os trabalhos que tratam desta versão generalizada do problema, a maioria tem seu foco em procedimentos heurísticos para resolvê-lo. Apenas no trabalho de [2] e no trabalho de [5] são propostos métodos exatos para resolver tal problema.

Em [2] é estudado o caso real de uma indústria siderúrgica, em que uma máquina é considerada como sendo uma seqüência de laminadores e cada tarefa representa a produção de um determinado produto (barra chata, cantoneira, vergalhão, etc.). São desenvolvidos dois modelos de programação linear inteira mista (PLIM), os quais, no entanto, exigem que os tempos de *setup* satisfaçam à desigualdade triangular. Apesar de ser considerado o caso com janelas de entrega, os testes computacionais foram realizados em problemas-teste com datas de entrega distintas. Foram resolvidos na otimalidade problemas com até 10 tarefas por meio do *software* de otimização GLPK, versão 4.8.

Já em [5] é proposto um modelo de PLIM baseado em uma das formulações de programação matemática apresentadas por [2]. Porém, este modelo não exige que os tempos de *setup* satisfaçam à desigualdade triangular. Por meio do otimizador CPLEX, versão 9.1, foram encontrados os ótimos em problemas-teste com até 12 tarefas e com tempos de *setup* simétricos. Os autores propõem também um algoritmo heurístico, baseado em GRASP, *Iterated Local Search* e *Variable Neighborhood Descent*. Para cada seqüência de tarefas gerada por esse algoritmo, é acionado um procedimento de complexidade polinomial para determinar a data ótima de início de processamento das tarefas na seqüência dada.

No presente trabalho, é proposta uma formulação matemática indexada no tempo para representar o problema em foco. É apresentado também um novo modelo de PLIM, baseado naquele proposto por [5], com menor número de variáveis e restrições. Estes dois modelos, bem como o modelo proposto por [5], são utilizados para resolver problemas-teste por meio do otimizador CPLEX, versão 10.1, sendo realizada uma comparação entre os resultados encontrados.

Este trabalho está organizado do seguinte modo. Na Seção 2 é feita uma descrição detalhada do problema estudado. O modelo de PLIM proposto por [5], um modelo de PLIM baseado neste último e uma formulação de programação matemática indexada no tempo são apresentados nas Seções 3, 4 e 5, respectivamente. Na Seção 6 são apresentados e discutidos os resultados encontrados, enquanto a Seção 7 conclui o trabalho.

2 Descrição do problema

O PSUMAA abordado neste trabalho possui as seguintes características: (i) Uma máquina deve processar um conjunto I de n tarefas; (ii) A cada tarefa $i \in I$ está associado um tempo de processamento P_i; uma janela de entrega $[E_i, T_i]$, na qual a tarefa i deve ser preferencialmente concluída; um custo α_i por unidade de tempo de antecipação; e um custo β_i por unidade de tempo de atraso. Há antecipação de uma tarefa $i \in I$ quando seu processamento é concluído antes de E_i e há atraso quando seu processamento é concluído depois de T_i; (*iii*) As tarefas que forem concluídas dentro de suas respectivas janelas de entrega não geram custo; (*iv*) A máquina executa no máximo uma tarefa por vez e, uma vez iniciado o processamento de uma tarefa, não é permitida a sua interrupção; (*v*) Todas as tarefas estão disponíveis para processamento na data 0; (*vi*) Entre duas tarefas i e j consecutivas é necessário um tempo S_{ij} de preparação da máquina, chamado tempo de *setup*; (*vii*) Assume-se que o tempo de preparação da máquina para o processamento da primeira tarefa na seqüência é igual a 0; (*viii*) É permitido tempo ocioso entre a execução de duas tarefas consecutivas.

O objetivo é determinar a seqüência de produção e as datas de início de produção das tarefas, de forma a minimizar a soma ponderada das antecipações e dos atrasos.

3 Modelo MPLIM-G

Nesta seção, apresenta-se o modelo de programação linear inteira mista (PLIM) proposto por [5], doravante denotado por MPLIM-G. Para auxiliar na modelagem, são utilizadas duas tarefas fictícias, 0 (zero) e $n+1$, que devem ser seqüenciadas necessariamente na primeira e na última posição, respectivamente. Admite-se que P_0 e P_{n+1} são iguais a zero e que $S_{0i} = S_{i0} = 0$ e $S_{i,n+1} = S_{n+1,i} = 0$, $\forall\, i \in I$. A data de início do processamento da tarefa $j \in I \cup \{0, n+1\}$ e os tempos de antecipação e atraso da tarefa $i \in I$ são representados por s_i, e_i e t_i, respectivamente.

Sejam y_{ij} variáveis que determinam a sequência de produção, sendo $y_{ij} = 1$ se a tarefa j for sequenciada imediatamente após a tarefa i e $y_{ij} = 0$, caso contrário, $\forall\, i, j \in I \cup \{0, n+1\}$. Considere, ainda, uma constante M de valor suficientemente grande. O modelo MPLIM-G é representado pelas equações (1)-(10):

$$\min \quad Z = \sum_{i=1}^{n} (\alpha_i e_i + \beta_i t_i) \tag{1}$$

$$\text{s.a.} \quad s_i + P_i + y_{ij}(M + S_{ij}) - M \leq s_j \qquad \forall\, i \in I \cup \{0\},\ \ \forall\, j \in I \cup \{n+1\} \ \text{ e } \ i \neq j \tag{2}$$

$$\sum_{j=1,\, j\neq i}^{n+1} y_{ij} = 1 \qquad \forall\, i \in I \cup \{0\} \tag{3}$$

$$\sum_{i=0,\, i\neq j}^{n} y_{ij} = 1 \qquad \forall\, j \in I \cup \{n+1\} \tag{4}$$

$$s_i + P_i + e_i \geq E_i \qquad \forall\, i \in I \tag{5}$$

$$s_i + P_i - t_i \leq T_i \qquad \forall\, i \in I \tag{6}$$

$$s_i \geq 0 \qquad \forall\, i \in I \cup \{0, n+1\} \tag{7}$$

$$e_i \geq 0 \qquad \forall\, i \in I \tag{8}$$

$$t_i \geq 0 \qquad \forall\, i \in I \tag{9}$$

$$y_{ij} \in \{0,\, 1\} \quad \forall\, i, j \in I \cup \{0, n+1\} \ \text{ e } \ i \neq j \tag{10}$$

A função objetivo, representada pela equação (1), busca a minimização da soma ponderada das antecipações e atrasos. As restrições (2) garantem que existe tempo suficiente para executar uma tarefa i e preparar a máquina antes do início do processamento da tarefa seguinte j. As restrições (3) e (4) garantem que cada tarefa terá apenas uma tarefa imediatamente sucessora e uma tarefa imediatamente antecessora, respectivamente. As restrições (5) e (6) definem as antecipações e os atrasos de acordo com as respectivas janelas de entrega de cada tarefa. As restrições (7), (8), (9) e (10) dizem respeito ao tipo de variáveis.

4 Modelo MPLIM-BG

O modelo de programação linear inteira mista (PLIM) apresentado a seguir é baseado no proposto por [5] (ver Seção 3) e será denotado por MPLIM-BG. Diferentemente de tais autores, é utilizada apenas uma tarefa fictícia, denominada 0 (zero), para auxiliar na modelagem. Esta tarefa é sequenciada duas vezes, sendo uma na primeira posição e outra na última. Considera-se que $P_0 = 0$ e que $S_{0i} = S_{i0} = 0$, $\forall\, i \in I$.

Considerando as definições de s_i, e_i, t_i e y_{ij} tal como no MPLIM-G, tem-se que as variáveis $y_{n+1,i}$ e $y_{i,n+1}$, $\forall\, i \in I \cup \{0\}$, e s_{n+1} deixam de ser necessárias no MPLIM-BG. Sendo assim, se comparado ao MPLIM-G, no MPLIM-BG há redução de $2n+3$ variáveis e $n+1$ restrições, com n representando o número de tarefas a serem sequenciadas.

Finalmente, o MPLIM-BG é obtido do MPLIM-G substituindo-se as restrições (2), (3), (4), (7) e (10) pelas restrições (11), (12), (13), (14) e (15), respectivamente, apresentadas a seguir:

$$\begin{aligned}
s_i + P_i + y_{ij}(M + S_{ij}) - M &\leq s_j && \forall\, i \in I \cup \{0\}, \quad \forall\, j \in I \quad \text{e} \quad i \neq j && (11)\\
\sum_{j=0,\, j\neq i}^{n} y_{ij} &= 1 && \forall\, i \in I \cup \{0\} && (12)\\
\sum_{i=0,\, i\neq j}^{n} y_{ij} &= 1 && \forall\, j \in I \cup \{0\} && (13)\\
s_i &\geq 0 && \forall\, i \in I \cup \{0\} && (14)\\
y_{ij} &\in \{0, 1\} && \forall\, i, j \in I \cup \{0\} \quad \text{e} \quad i \neq j && (15)
\end{aligned}$$

5 Modelo MPLIM-IT

Propõe-se, a seguir, um modelo de programação linear inteira mista (PLIM) indexado no tempo para representar o PSUMAA abordado. Tal modelo, doravante denotado por MPLIM-IT, é baseado no trabalho de [3], que utilizou a discretização do tempo para modelar o problema de minimização dos atrasos em seqüenciamento de máquinas paralelas com tempos de preparação dependentes da seqüência. O MPLIM-IT é descrito a seguir.

Seja $H = \{h_0, h_1, h_2, \ldots, h_L\}$ o conjunto com as possíveis datas de início de processamento das tarefas e considere x_{ih} variáveis binárias, sendo $x_{ih} = 1$ se a tarefa i é programada para iniciar na data h e $x_{ih} = 0$, caso contrário, $\forall\, i \in I$ e $\forall\, h \in H$.

Se e_i e t_i representam, respectivamente, as unidades de tempo de antecipação e tempo de atraso da tarefa $i \in I$, a formulação MPLIM-IT pode ser escrita como:

$$\min \qquad Z = \sum_{i=1}^{n} (\alpha_i e_i + \beta_i t_i) \tag{16}$$

$$\begin{aligned}
\text{s.a.}\quad x_{ih} + \sum_{u=h}^{\min(h+P_i+S_{ij}-1, H_L)} x_{ju} &\leq 1 && \forall\, i, j \in I, \quad \forall\, h \in H \quad \text{e} \quad i \neq j && (17)\\
\sum_{h=H_0}^{H_L} x_{ih} &= 1 && \forall\, i \in I && (18)\\
\sum_{h=H_0}^{H_L} x_{ih} h + P_i + e_i &\geq E_i && \forall\, i \in I && (19)\\
\sum_{h=H_0}^{H_L} x_{ih} h + P_i - t_i &\leq T_i && \forall\, i \in I && (20)\\
e_i &\geq 0 && \forall\, i \in I && (21)\\
t_i &\geq 0 && \forall\, i \in I && (22)\\
x_{ih} &\in \{0, 1\} && \forall\, i \in I \quad \text{e} \quad \forall\, h \in H && (23)
\end{aligned}$$

A função objetivo representada pela equação (16) tem como critério de otimização a minimização da soma ponderada das antecipações e dos atrasos. As restrições (17) garantem que existe tempo suficiente para executar uma tarefa i e preparar a máquina antes do início do processamento da tarefa seguinte j. As restrições (18) garantem que cada tarefa seja executada uma única vez. As restrições (19) e (20) definem as antecipações e os atrasos de acordo com as respectivas janelas de entrega de cada tarefa. As restrições (21), (22) e (23) determinam os domínios das variáveis do problema.

É importante observar que, diferentemente do MPLIM-G e do MPLIM-BG, o MPLIM-IT somente é válido se a desigualdade triangular for satisfeita pelos dados do problema, ou seja, se as condições (24) forem satisfeitas:

$$S_{ik} \leq S_{ij} + S_{jk} + P_j \qquad \forall\, i,\, j,\, k \in I,\ i \neq j,\ i \neq k \text{ e } j \neq k \tag{24}$$

No PSUMAA abordado neste trabalho, a execução de qualquer tarefa pode ser iniciada em qualquer momento futuro e, portanto, caso tenha-se a garantia de que na seqüência ótima de um determinado problema o início do processamento da primeira tarefa não ocorre antes de h_{inf} e que o início do processamento da última tarefa é no máximo h_{sup}, é conveniente tomar $H = \{h_{inf}, h_{inf}+1, h_{inf}+2, \ldots, h_{sup}\}$. Assim, como o MPLIM-IT é fortemente sensível à cardinalidade de H, quanto menor o intervalo $[h_{inf}, h_{sup}]$, menor o número de variáveis terá o modelo.

6 Experimentos computacionais

Os modelos matemáticos apresentados nas seções 3, 4 e 5 foram implementados usando o modelador AMPL e resolvidos pelo otimizador CPLEX, versão 10.1, da ILOG. Os testes foram realizados em um computador AMD Turion(tm) 64 X2 TL-58 1900 MHz, com 2 GB de RAM, sob plataforma Windows XP.

Foram gerados problemas-teste baseados nos trabalhos de [8] e [9], conforme a seguir se descreve. Dada uma tarefa i, o tempo de processamento (P_i), o custo por unidade de atraso (β_i) e o custo por unidade de antecipação (α_i) são números inteiros selecionados aleatoriamente dentro dos intervalos $[1, 40]$, $[1, 10]$ e $[1, \beta_i]$, respectivamente. O centro da janela de entrega de i é um valor inteiro aleatório dentro do intervalo $[(1-FA-VRJ/2)\times TTP, (1-FA+VRJ/2)\times TTP]$, em que TTP é o tempo total de processamento de todos as tarefas, FA é o fator de atraso e VRJ é a variação relativa da janela de entrega. O tamanho da janela de entrega é um valor inteiro selecionado aleatoriamente no intervalo $[0, TTP/n]$, sendo n o número de tarefas. Para toda tarefa $j \neq i$, o tempo de *setup* (S_{ij}) é um número inteiro aleatório pertencente ao intervalo $[5, 15]$. Assim, diferentemente de [5], os tempos de *setup* não são necessariamente simétricos.

Foram gerados conjuntos de problemas-teste com 6, 7, 8, 9, 10, 11, 12, 13 e 14 tarefas, respectivamente, sendo utilizados os valores 0,1; 0,3; 0,5 e 0,8 para FA e 0,4; 0,7; 1,0 e 1,3 para VRJ. Sendo assim, há 16 problemas-teste em cada conjunto. Durante a geração dos problemas-teste, sempre que um problema gerado não satisfazia a desigualdade triangular, ele era descartado e outro problema com os mesmos valores para FA e VRJ era gerado. Desta forma, todos os problemas da base de dados satisfazem à desigualdade triangular.

Para os modelos MPLIM-G e MPLIM-BG, foi utilizado $M = 1000$. Para cada problema-teste, o conjunto H do MPLIM-IT foi determinado pelas expressões (25) e (26):

$$\begin{aligned} h_{inf} &= \max\Big(0,\, \min\Big\{T_i \mid i \in I\Big\} + \min\Big\{S_{ij} \mid i, j \in I \text{ e } i \neq j\Big\} - \\ &\qquad - \sum_{i \in I} P_i - \sum_{i \in I} \max\Big\{S_{ij} \mid j \in I \text{ e } i \neq j\Big\}\Big) \end{aligned} \tag{25}$$

$$\begin{aligned} h_{sup} &= \max\Big\{E_i \mid i \in I\Big\} + \sum_{i \in I} P_i + \sum_{i \in I} \max\Big\{S_{ij} \mid j \in I \text{ e } i \neq j\Big\} - \\ &\quad - \min\Big\{S_{ij} \mid i, j \in I \text{ e } i \neq j\Big\} - \min\Big\{P_i \mid i \in I\Big\} - \\ &\quad - \min\Big(\min\Big\{P_i \mid i \in I\Big\},\, \max\Big\{E_i \mid i \in I\Big\}\Big) \end{aligned} \tag{26}$$

em que h_{inf} garante que todas as tarefas podem ser concluídas sem atraso (ou o processamento da primeira tarefa é iniciado na data 0) e h_{sup} assegura que todas as tarefas podem ser executadas sem antecipação. Portanto, tem-se a garantia de que as datas de início de processamento das tarefas na seqüência ótima pertencem ao conjunto $H = \{h_{inf}, h_{inf}+1, h_{inf}+2, \cdots, h_{sup}\}$.

Considerou-se, ainda, o limite de 3600 segundos para resolução de cada problema pelo CPLEX. Para os problemas em que este limite de tempo é atingido, a solução retornada não é necessariamente ótima; porém, é retornado, pelo CPLEX, um limite inferior para o ótimo do mesmo. Este limite é utilizado para medir a qualidade da solução retornada, dada pela equação:

$$gap = \frac{(f^{CPLEX} - L)}{L} \times 100\% \tag{27}$$

em que f^{CPLEX} e L representam o valor da solução e o limite inferior encontrados, respectivamente. Se uma solução ótima é encontrada, tem-se $gap = 0$.

Os resultados encontrados para o MPLIM-G, para o MPLIM-BG e para o MPLIM-IT são resumidos na Tabela 1, na qual a primeira coluna indica o número de tarefas existentes em cada conjunto de problemas-teste. Para cada conjunto de problemas, as colunas "Ótimos Encontrados" mostram (em porcentagem) em quantos problemas o CPLEX encontrou uma solução ótima para os respectivos modelos, as colunas "$\overline{gap}$" mostram as médias dos *gaps* para os respectivos modelos e as colunas "Tempo" mostram as médias dos tempos (em segundos) demandados por tal otimizador na resolução dos problemas-teste com os respectivos modelos.

Tabela 1: Comparação MPLIM-G × MPLIM-BG × MPLIM-IT.

# Tarefas	**MPLIM-G** Ótimos Encontrados (%)	$\overline{gap}$ (%)	Tempo (s)	**MPLIM-BG** Ótimos Encontrados (%)	$\overline{gap}$ (%)	Tempo (s)	**MPLIM-IT** Ótimos Encontrados (%)	$\overline{gap}$ (%)	Tempo (s)
06	100,00	0,00	0,18	100,00	0,00	0,16	100,00	0,00	8,77
07	100,00	0,00	1,69	100,00	0,00	1,78	100,00	0,00	20,96
08	100,00	0,00	16,51	100,00	0,00	19,00	100,00	0,00	34,67
09	100,00	0,00	226,25	100,00	0,00	249,67	100,00	0,00	79,24
10	62,50	14,99	1809,11	68,75	14,84	1796,76	100,00	0,00	196,17
11	31,25	41,59	2628,39	31,25	40,61	2658,07	100,00	0,00	453,88
12	6,25	66,88	3378,70	6,25	67,03	3380,08	93,75	0,38	845,25

Pela Tabela 1, observa-se que o MPLIM-G e o MPLIM-BG apresentaram comportamentos bem semelhantes. Usando-se esses dois modelos, o CPLEX foi capaz de resolver na otimalidade somente problemas-teste com até 9 tarefas. Por outro lado, com o modelo MPLIM-IT, o CPLEX conseguiu resolver na otimalidade todos os problemas-teste de até 11 tarefas. Além disso, enquanto as duas primeiras formulações permitiam ao CPLEX resolver na otimalidade apenas 6,25% dos problemas-teste envolvendo 12 tarefas, utilizando-se a formulação MPLIM-IT foi possível solucionar na otimalidade 93,75% do mesmo conjunto de problemas. Esta última formulação também exigiu, na maioria dos casos, um menor tempo computacional para se alcançar a otimalidade. De fato, à exceção dos três primeiros conjuntos de problemas-teste, a formulação MPLIM-IT demandou aproximadamente de 3 a 9 vezes menos tempo que as demais formulações. Ainda na Tabela 1, nota-se que a formulação MPLIM-IT permitiu determinar, para os problemas-teste com 12 tarefas, dentro do limite de tempo adotado, soluções tendo $\overline{gap}$ de 0,38%, em contraste ao valor obtido pelas duas outras formulações avaliadas (MPLIM-G e MPLIM-BG), que obtiveram $\overline{gap}$ de 66,88% e 67,03%, respectivamente.

Apesar de não ser mostrado na Tabela 1, o CPLEX com o MPLIM-IT não conseguiu sequer encontrar uma solução viável, dentro do limite de tempo adotado, em 6,25% dos problemas-teste com 13 tarefas e em 56,25% dos problemas-teste com 14 tarefas. Com os outros dois modelos, o $\overline{gap}$ retornado para estes problemas foi superior a 68%.

7 Conclusões

Este trabalho tratou o problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção (PSUMAA), considerando janelas de entrega distintas e tempo de preparação da máquina dependente da seqüência. Foi apresentada uma nova formulação de programação matemática para o problema (MPLIM-BG), a qual exige um número menor de variáveis e restrições em relação àquela proposta por [5] (MPLIM-G). Propôs-se, também, uma formulação matemática indexada no tempo (MPLIM-IT) para representar o PSUMAA.

O otimizador CPLEX 10.1 foi utilizado para resolver as três formulações matemáticas em tela, aplicadas em problemas-teste com até 14 tarefas. Os resultados computacionais mostram que as formulações MPLIM-G e MPLIM-BG proporcionam resultados semelhantes e que a formulação MPLIM-IT é mais eficiente que estas duas na resolução de problemas com até 12 tarefas, visto que esta última proporcionou ao CPLEX encontrar soluções com garantia de melhor qualidade e com menor tempo médio de processamento que as demais. Para os problemas-teste com mais de 12 tarefas, devido ao limite de tempo estabelecido para o CPLEX retornar uma solução, os resultados obtidos não permitem uma comparação efetiva entre os três modelos. Como trabalhos futuros, sugere-se o estudo de propriedades do PSUMAA que permitam reduzir a cardinalidade do conjunto H da formulação MPLIM-IT, dado que a mesma é fortemente dependente de tal conjunto.

Agradecimentos

O primeiro autor agradece ao CEFET-MG pela bolsa de pesquisa e o terceiro, ao CNPq (processo 474831/2007-8) e à FAPERJ (processo E-26/101.023/2007), pelo apoio ao desenvolvimento da presente pesquisa.

Referências

[1] A. Allahverdi, J. N.D. Gupta e T. Aldowaisan, A review of scheduling research involving setup considerations, *Omega: International Journal of Management Science*, 27 (1999) 219-239.

[2] L. M. Bustamante, “Minimização do custo de antecipação e atraso para o problema de sequenciamento de uma máquina com tempo de preparação dependente da sequência: aplicação em uma usina siderúrgica”, Dissertação de Mestrado, DEP-UFMG, 2007.

[3] M. R. de Paula, “Heurísticas para a minimização dos atrasos em seqüenciamento de máquinas paralelas com tempos de preparação dependentes da seqüência”, Dissertação de Mestrado, DCC-UFMG, 2008.

[4] M. F. França Filho, “GRASP e Busca Tabu aplicados a problemas de programação de tarefas em máquinas paralelas”, Tese de Doutorado, FEEC-Unicamp, 2007.

[5] A. C. Gomes Júnior, C. R. V. Carvalho, P. L. A. Munhoz e M. J. F. Souza, Um método heurístico híbrido para a resolução do problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção, em “Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional”, pp. 1649-1660, SOBRAPO, Fortaleza, 2007.

[6] S. R. Gupta e J. S. Smith, Algorithms for single machine total tardiness scheduling with sequence dependent setups, *European Journal of Operational Research*, 175 (2006) 722-739.

[7] G. Li, Single machine earliness and tardiness scheduling, *European Joumal of Operational Research*, 96 (1997) 546-558.

[8] G. Rabadi, M. Mollaghasemi e G. C. Anagnostopoulos, A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time, *Computers & Operations Research*, 31 (2004) 1727-1751.

[9] G. Wan e B. P.-C. Yen, Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties, *European Journal of Operational Research*, 142 (2002) 271-281.