Formulações matemáticas para o problema de programação de tarefas com janelas de entrega e tempos de preparação da máquina

Bruno Ferreira Rosa, Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil
brunorosa@div.cefetmg.br, sergio@dppg.cefetmg.br

Philippe Michelon, Zacharie Ales
Laboratoire d'Informatique d'Avignon, Université d'Avignon et des Pays du Vaucluse (UAPV)
339 chemin des Meinajaries, Agroparc BP 1228, 84911, Avignon, cedex 9, França
philippe.michelon@univ-avignon.fr, zacharie.ales@univ-avignon.fr

Marcone Jamilson Freitas Souza
Departamento de Computação - Universidade Federal de Ouro Preto (UFOP)
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto (MG), Brasil
marcone@iceb.ufop.br

RESUMO

Este trabalho trata o problema de programação de tarefas em uma máquina. No problema abordado, cada tarefa possui uma janela de entrega, dentro da qual ela deve ser preferencialmente concluída, e os tempos de preparação da máquina são dependentes da sequência de processamento. O objetivo é minimizar a soma ponderada das antecipações e dos atrasos no processamento das tarefas. Propõe-se um horizonte de planejamento para o processamento das tarefas. Além disso, são propostas duas novas formulações inteiras mistas (sendo uma delas indexada no tempo) e duas formulações binárias indexadas no tempo que representam o problema. O horizonte de planejamento proposto é utilizado para determinar os parâmetros de entrada das formulações matemáticas. Experimentos computacionais com as formulações propostas e com mais duas formulações da literatura mostram que as formulações binárias proporcionam ao otimizador CPLEX melhor desempenho que as demais formulações experimentadas.

PALAVRAS CHAVE. Programação em uma máquina. Janelas de entrega distintas. Formulação matemática.

SE4 - Cooperação Brasil & França.

ABSTRACT

This paper addresses the single-machine job scheduling problem. In the considered problem, each job has a time window, in which the job should preferably be completed. The machine setup times are sequence-dependent. The objective is to minimize the weighted sum of the earliness and tardiness of the jobs. It is proposed a planning horizon for the jobs. In addition, two new mixed integer formulations (where one of them is time-indexed) and two time-indexed binary formulations that represent the problem are proposed. The proposed planning horizon is used to determine the input parameters of the mathematical formulations. Computational experiments with the proposed formulations and two other literature formulations show that the binary formulations provide to CPLEX Optimizer a better performance than the other experimented formulations.

KEYWORDS. Single-machine scheduling. Distinct time windows. Mathematical formulation.

ES4 - Cooperation Brazil & France.

Introdução

O problema de programação de tarefas em uma máquina com penalidades por antecipação e atraso da produção consiste em programar um conjunto de tarefas em uma máquina, com o objetivo de minimizar a soma ponderada das antecipações e dos atrasos dessas tarefas. A resolução desse problema, segundo [Baker e Scudder, 1990], reflete melhores ambientes de produção administrados conforme a filosofia *Just in Time* (JIT).

Neste trabalho trata-se o problema de programação de tarefas em uma máquina considerando a existência de penalidades por antecipação e atraso da produção distintas, de janelas de entrega distintas, de ociosidade de máquina e de tempos de preparação da máquina dependentes da sequência de processamento programada. Doravante, esse problema será denotado por PSUMAA, nomenclatura adotada por [Gomes Júnior et al., 2007]. De acordo com a notação de [Allahverdi, 2015], o problema em foco é representado por $1/ST_{sd}/w_{1i}E_i + w_{2i}T_i$.

O PSUMAA possui muitas aplicações práticas em ambientes de administração JIT, em fabricação de semicondutores, na fabricação de produtos químicos, em programação PERT/CPM, em serviços de vídeo sob demanda, dentre outras [Janiak et al., 2015]. Além do grande número de aplicações associado ao PSUMAA, está a dificuldade de resolvê-lo de forma ótima. Segundo [Allahverdi, 2015], ele é um problema NP-difícil. Essa união entre aplicabilidade e dificuldade de encontrar uma solução ótima para o PSUMAA motiva o seu estudo.

[Bustamante, 2007] desenvolveu duas formulações lineares inteiras mistas para o caso do PSUMAA com datas de entrega distintas. [Gomes Júnior et al., 2007] propuseram uma formulação inteira mista para o caso geral do PSUMAA, sendo tal formulação baseada em uma das formulações de [Bustamante, 2007]. Esses autores também propuseram um algoritmo heurístico baseados em metaheurísticas da literatura para resolvê-lo. Em [Rosa e Souza, 2009] é proposta uma formulação inteira mista indexada no tempo para o problema. O horizonte de planejamento, o qual faz parte dos dados de entrada da formulação proposta, é obtido por um procedimento heurístico e, por isso, não garante a otimalidade da solução obtida pela resolução da formulação. Para mais informações, [Janiak et al., 2015] apresentam uma revisão de trabalhos com janelas de entrega, ao passo que [Allahverdi et al., 1999, 2008] e [Allahverdi, 2015] fazem uma ampla revisão da literatura sobre problemas de programação com tempos de preparação.

Este trabalho propõe um horizonte de planejamento para cada tarefa do PSUMAA. Também são propostas duas novas formulações inteiras mistas (sendo uma delas indexada no tempo) e duas novas formulações binárias indexadas no tempo que representam o problema. O horizonte de planejamento proposto é utilizado para determinar os parâmetros de entrada das formulações matemáticas e garante a otimalidade da solução obtida. O otimizador CPLEX foi aplicado a um conjunto de problemas-teste da literatura para experimentar as formulações propostas, bem como as formulações de [Gomes Júnior et al., 2007] e de [Rosa e Souza, 2009]. Uma síntese dos resultados obtidos é apresentada ao final do trabalho.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2 é feita uma descrição detalhada do PSUMAA. Um horizonte de planejamento para o processamento de cada tarefa é proposto na Seção 3. As formulações matemáticas propostas e as formulações de [Gomes Júnior et al., 2007] e de [Rosa e Souza, 2009] são apresentadas na Seção 4. Na Seção 5 são apresentados e discutidos os resultados computacionais obtidos, enquanto a Seção 6 conclui o trabalho.

Características do problema abordado

O problema de programação de tarefas em uma máquina tratado neste trabalho (PSUMAA) possui as seguintes características: (*i*) Uma máquina deve processar um conjunto I de n tarefas; (*ii*) Cada tarefa $x \in I$ possui um tempo de processamento P_x e uma janela de entrega $[E_x, T_x]$, dentro da qual a tarefa x deve ser preferencialmente concluída. E_x indica o início e T_x indica o final da janela de entrega de x; (*iii*) Uma tarefa $x \in I$ é dita antecipada se é concluída antes de E_x. Nesse caso, tem-se um custo α_x por unidade de tempo de antecipação. Por outro lado, uma tarefa $x \in I$ é dita atrasada se é concluída após T_x. Nesse caso, o custo por unidade de atraso de x é dado por β_x. As tarefas concluídas dentro de suas respectivas janelas de entrega não geram custos; (*iv*) Não é permitida a sobreposição de tarefas, isto é, a máquina pode processar no máximo uma tarefa por vez; (*v*) Uma vez iniciado o processamento de uma tarefa, não é permitido a sua interrupção; (*vi*) Todas as tarefas estão disponíveis para processamento na data 0; (*vii*) Entre o processamento de duas tarefas consecutivas x e $y \in I$, é necessário um tempo de preparação da máquina S_{xy}. Assume-se que o tempo de preparação da máquina para o processamento da primeira tarefa programada é igual a 0 e que os tempos de preparação satisfazem a desigualdade triangular, isto é, $S_{xz} \leq S_{xy} + P_y + S_{yz}$, $\forall\, x,\, y,\, z \in I$, $x \neq y$, $x \neq z$ e $y \neq z$; (*viii*) É permitida a ociosidade de máquina, ou seja, a máquina pode ficar ociosa entre o processamento de duas tarefas consecutivas.

O objetivo é determinar uma programação de I (isto é, uma alocação de tempo para cada tarefa $x \in I$ em uma dada máquina) que minimize o custo oriundo de antecipações e de atrasos. Em outras palavras, o objetivo do PSUMAA é encontrar uma programação X de I que minimize

$$f(X) = \sum_{x \in I} \alpha_x \cdot \max\left(0,\; E_x - c_x^X\right) + \sum_{x \in I} \beta_x \cdot \max\left(0, c_x^X - T_x\right),$$

em que c_x^X representa a data de conclusão da tarefa x na programação X.

Horizonte de planejamento

Um posicionamento ótimo para o PSUMAA não é necessariamente aquele de menor *makespan*, isto é, um posicionamento de menor data de conclusão da última tarefa [Pinedo, 2012; Nogueira et al., 2014]. Esta seção tem o objetivo de estabelecer um horizonte de planejamento para o processamento das tarefas em I.

Seja S^I_{max} o maior tempo possível de preparação da máquina contido em uma programação de I. O problema de determinar S^I_{max} pode ser facilmente reduzido a uma variação do Problema do Caixeiro Viajante, denominada *Maximum Traveling Salesman Problem with General Non-Negative weights* – MAX TSP [Pinedo, 2012; Gutin e Punnen, 2007]. Com esse propósito, seja x_0 uma tarefa fictícia tal que $S_{x,x_0} = S_{x_0,x} = 0, \forall\, x \in I$. Cada tarefa em $I \cup \{x_0\}$ corresponde a uma cidade no MAX TSP e a distância da cidade x para a cidade y é dada pelo tempo de preparação S_{xy}. Observa-se que, neste caso, a distância entre duas cidades não é necessariamente simétrica. Seja γ_x a variável de decisão que diz a ordem em que a cidade x é visitada, $\forall x \in I \cup \{x_0\}$. Assim como em Miller et al. [1960], sejam u_{xy} variáveis de decisão binárias tais que, $\forall\, x,\, y \in I \cup \{x_0\}$ e $x \neq y$,

$$u_{xy} = \begin{cases} 1, & \text{se a cidade } y \text{ é visitada imediatamente após a cidade } x; \\ 0, & \text{caso contrário.} \end{cases}$$

O problema de determinar S^I_{max} pode ser formulado como segue:

$$\begin{aligned}
\max \quad & \sum_{x \in I} \sum_{y \in I \setminus \{x\}} S_{xy} \cdot u_{xy} && && (1)\\
\text{s. a.} \quad & \sum_{y \in I \cup \{x_0\},\, y \neq x} u_{xy} = 1 && \forall\, x \in I \cup \{x_0\} && (2)\\
& \sum_{x \in I \cup \{x_0\},\, x \neq y} u_{xy} = 1 && \forall\, y \in I \cup \{x_0\} && (3)\\
& \gamma_0 = 1 && && (4)\\
& \gamma_x + (n+1)u_{xy} - n \leq \gamma_y && \forall\, x,\, y \in I \cup \{x_0\} \;\text{ e }\; x \neq y && (5)\\
& \gamma_x \geq 0 && \forall\, x \in I && (6)\\
& u_{xy} \in \{0,\, 1\} && \forall\, x, y \in I \cup \{x_0\} && (7)
\end{aligned}$$

A função objetivo, dada pela expressão (1), busca a maximização do tempo total de preparação da máquina. As restrições (2) e (3) garantem que uma preparação é realizada antes e outra depois de cada tarefa em $I \cup \{x_0\}$. A restrição (4) impõe que a tarefa fictícia x_0 seja a primeira a ser processada. As restrições (5) previnem sub-rotas. As restrições (6) e (7) dizem respeito aos domínios das variáveis de decisão.

Dadas uma programação X de I e uma tarefa $x \in I$, considere que s_x^X representa a data de início da tarefa x em X e que c_x^X representa a data de conclusão da tarefa x em X. Além disso, seja $s_I^X = \min_{y \in I} s_y^X$ e seja $c_I^X = \max_{y \in I} c_y^X$. A seguinte definição particiona o conjunto das programações de tarefas em I. Trata-se de uma generalização da definição de programação semi-ativa de [Pinedo, 2012].

Definição 1 (Programação natural) *Uma programação possível X de I é dita natural se, mantendo a ordem de processamento das tarefas, não é possível construir uma programação X' tal que $f(X') \leq f(X)$ e uma das seguintes condições é satisfeita:*

- $c_I^{X'} < c_I^X$;
- $c_I^{X'} - s_I^{X'} < c_I^X - s_I^X$.

Em outras palavras, uma programação X é dita natural se, fixada a sequência de processamento das tarefas, toda programação com custo associado menor ou igual ao custo de X é concluída depois de X e possui tempo de processamento maior que o de X. Para os problemas de programação em que não há penalidades por antecipação, a definição de programação natural proposta é equivalente à definição de programação semi-ativa de [Pinedo, 2012].

A Proposição 1 permite reduzir o espaço de busca do PSUMAA ao conjunto de programações naturais.

Proposição 1 *Existe uma programação ótima de I que é natural.*

Prova: Seja X uma programação ótima de I que não é natural. Logo, existe uma outra programação X' tal que $c_I^{X'} - s_I^{X'} < c_I^X - s_I^X$ ou $c_I^{X'} < c_I^X$. Se X' não é natural, então existe uma outra programação ótima cujo tempo de processamento é menor ou a data de conclusão é mais cedo. Já que tempos ociosos de máquina são eliminados cada vez que esse procedimento é realizado, ele pode ser repetido até que uma programação ótima natural de I seja encontrada. $\square$

Lema 1 *Dada uma programação natural X de I, se a última tarefa em X (x_n) tem o seu processamento iniciado após a data $\max\big(\max_{x \in I}(E_x - P_x), 0\big)$, então não existe ociosidade de máquina entre essa data e a data de início de x_n $\big(s_{x_n}^X\big)$.*

Prova: Suponha que o processamento de x_n inicia após a data $\max\big(\max_{x \in I}(E_x - P_x), 0\big)$ e que existe ociosidade de máquina entre essa data e $s_{x_n}^X$. Toda tarefa y que tem seu processamento iniciado após a data $\max\big(\max_{x \in I}(E_x - P_x), 0\big)$ é tal que $c_y^X = s_y^X + P_y \geq \max_{x \in I}\big(E_x - P_x\big) + P_y \geq \big(E_y - P_y\big) + P_y = E_y$. Logo, essas tarefas não estão antecipadas e podem ser deslocadas para a esquerda, mantendo-se não antecipadas, até que não tenha mais ociosidade de máquina entre $\max\big(\max_{x \in I}(E_x - P_x), 0\big)$ e $s_{x_n}^X$. A programação obtida X' é tal que $f(X') \leq f(X)$ e $c_I^{X'} < c_I^X$, o que contradiz o fato de X ser uma programação natural. $\square$

Lema 2 *Dada uma programação natural X de I, se a primeira tarefa em X (x_1) é concluída antes da data $\min_{x \in I} T_x$, então não existe ociosidade de máquina entre a conclusão de x_1 $\big(c_{x_1}^X\big)$ e $\min_{x \in I} T_x$.*

Prova: Suponha que a tarefa x_1 é concluída antes da data $\min_{x \in I} T_x$ e que existe ociosidade de máquina entre $c^X_{x_1}$ e $\min_{x \in I} T_x$. Uma vez que $c^X_y \leq \min_{x \in I} T_x \leq T_y$ para toda tarefa y que é concluída antes de $\min_{x \in I} T_x$, essas tarefas não estão atrasadas e podem ser deslocadas para a direita, mantendo-se não atrasadas, até que não tenha mais ociosidade de máquina entre $c^X_{x_1}$ e $\min_{x \in I} T_x$. A programação X' obtida após esse deslocamento é tal que $f(X') \leq f(X)$ e $c^{X'}_I - s^{X'}_I < c^X_I - s^X_I$, o que contradiz o fato de X ser uma programação natural. $\square$

Lema 3 *A data mais cedo para o início do processamento de uma dada tarefa x em programações naturais de I ocorre em pelo menos uma programação na qual x é a primeira tarefa processada.*

Prova: Seja X uma programação natural de I em que uma dada tarefa x não é a primeira tarefa processada. Considere a programação X' oriunda de X pela realocação das tarefas processadas antes de x para o final da programação. Caso X' não seja natural, é possível obter uma programação natural X'' de X'. X'' é tal que x é a primeira tarefa processada e $s^{X''}_x \leq s^X_x$. $\square$

Corolário 1 *A data mais tarde para o início do processamento de uma dada tarefa x em programações naturais de I ocorre em pelo menos uma programação na qual x é a última tarefa processada.*

Prova: Seja X uma programação natural de I em que uma dada tarefa x não é a última tarefa processada. A programação X' obtida de X pela realocação das tarefas processadas após x para imediatamente antes da primeira tarefa é uma programação natural tal que x é a última tarefa processada e $s^{X'}_x \geq s^X_x$. Observa-se que, para essa realocação seja possível, pode ser necessário deslocar as demais tarefas para a direita. $\square$

A próxima proposição fornece um horizonte de planejamento para cada tarefa $x \in I$.

Teorema 1 *Se X é uma programação natural de I, então, para qualquer tarefa $x \in I$, tem-se:*

$$s^{LB}_x \leq s^X_x \leq s^{UB}_x,$$

onde

- $s^{LB}_x = \max\left(0, \min\left(\min_{y \in I \setminus \{x\}} T_y - \sum_{y \in I} P_y - S^{I \setminus \{x\}}_{max} - \max_{y \in I \setminus \{x\}} S_{xy},\, T_x - P_x\right)\right)$ *e*
- $s^{UB}_x = \max\left(\max\left(0, \max_{y \in I \setminus \{x\}} (E_y - P_y)\right) + \sum_{y \in I \setminus \{x\}} P_y + S^{I \setminus \{x\}}_{max} + \max_{y \in I \setminus \{x\}} S_{yx},\, E_x - P_x\right)$.

Prova: Dada uma tarefa $x \in I$, de acordo com o Lema 3, a data mais cedo de início de x em programações naturais de I ocorre em uma programação X' na qual x é a primeira tarefa processada. Nesse caso, $s^{X'}_I = s^{X'}_x$ e o maior tempo de preparação da máquina contido em X' é $S^{I \setminus \{x\}}_{max} + \max_{y \in I \setminus \{x\}} S_{xy}$. Se $s^{X'}_x < s^{LB}_x$, então é possível deslocar a tarefa x para a direita, mantendo a sequência de processamento das tarefas, sem aumentar o custo associado a X' (da mesma forma em que foi feito na prova do Lema 2, mas agora sabe-se que x é a primeira tarefa em X'). Isso contradiz o fato de X' ser uma programação natural. Logo, se X é uma programação natural de I, então $s^{LB}_x \leq s^{X'}_x \leq s^X_x$.

Similarmente, com base no Corolário 1, a data mais tarde de início da tarefa x em programações naturais de I ocorre em uma programação X'' na qual x é a última tarefa processada. Nesse caso, $c^{X''}_I = c^{X''}_x = s^{X''}_x + P_x$ e o maior tempo de preparação da máquina contido em X'' é $S^{I \setminus \{x\}}_{max} + \max_{y \in I \setminus \{x\}} S_{yx}$. Se $s^{X''}_x > s^{UB}_x$, então é possível deslocar a tarefa x para a esquerda, mantendo a sequência de processamento das tarefas, sem aumentar o custo associado a X'' (da mesma forma em que foi feito na prova do Lema 1, mas agora sabe-se que x é a última tarefa em X''). Isso contradiz o fato de X'' ser uma programação natural. Logo, se X é uma programação natural de I, então $s^X_x \leq s^{X''}_x \leq s^{UB}_x$.

Formulações matemáticas

A formulação inteira mista de [Gomes Júnior et al., 2007] e a formulação inteira mista indexada no tempo de [Rosa e Souza, 2009] para o PSUMAA são apresentadas nesta seção. Ademais, são propostas duas novas formulações inteiras mistas, sendo uma delas indexada no tempo, e duas novas formulações binárias para o problema.

Formulação inteira mista de [Gomes Júnior et al., 2007] – IMG

Nesta Subseção é apresentada a formulação linear inteira mista de Gomes Júnior et al. [2007] para representar o PSUMAA, denotada por IMG. Para auxiliar na modelagem, utiliza-se uma tarefa fictícia (x_0). Admite-se que $P_{x_0} = 0$ e que $S_{x_0x} = S_{xx_0} = 0,\ \forall\, x \in I$.

Sejam u_{xy} variáveis de decisão que determinam a sequência em que as tarefas serão processadas, sendo $u_{xy} = 1$ se a tarefa y for processada imediatamente após a tarefa x e $u_{xy} = 0$, caso contrário, $\forall\, x,\, y \in I \cup \{x_0\}$. Considere que s_x, e_x e t_x são variáveis de decisão que determinam a data de início e os tempos de antecipação e de atraso, respectivamente, de cada tarefa $x \in I \cup \{x_0\}$.

Se M_{xy} é um valor suficientemente grande, $\forall x,\, y \in I \cup \{x_0\}$ e $x \neq y$, a formulação IMG é representada pelas equações:

$$\min \sum_{x \in I} (\alpha_x e_x + \beta_x t_x) \tag{8}$$

$$\begin{aligned}
\text{s. a.} \quad \sum_{y \in I \cup \{x_0\},\, y \neq x} u_{xy} &= 1 && \forall\, x \in I \cup \{x_0\} && (9)\\
\sum_{x \in I \cup \{x_0\},\, x \neq y} u_{xy} &= 1 && \forall\, y \in I \cup \{x_0\} && (10)\\
s_x + P_x + S_{xy}u_{xy} + M_{xy}(u_{xy} - 1) &\leq s_y && \forall\, x,\, y \in I \cup \{x_0\} \text{ e } x \neq y && (11)\\
s_x + P_x + e_x &\geq E_x && \forall\, x \in I && (12)\\
s_x + P_x - t_x &\leq T_x && \forall\, x \in I && (13)\\
s_x &\geq 0 && \forall\, x \in I \cup \{x_0\} && (14)\\
e_x &\geq 0 && \forall\, x \in I && (15)\\
t_x &\geq 0 && \forall\, x \in I && (16)\\
u_{xy} &\in \{0,\, 1\} && \forall\, x, y \in I \cup \{x_0\} && (17)
\end{aligned}$$

A função objetivo, representada pela expressão (8), busca a minimização da soma ponderada das antecipações e atrasos. As restrições (9) e (10) garantem que cada tarefa terá apenas uma tarefa imediatamente sucessora e uma tarefa imediatamente antecessora, respectivamente. As restrições (11) garantem que existe tempo suficiente para processar uma tarefa antes de iniciar o processamento da tarefa seguinte. As restrições (12) e (13) determinam as antecipações e os atrasos de acordo com as respectivas janelas de entrega de cada tarefa. As restrições (14), (15), (16) e (17) dizem respeito aos domínios das variáveis de decisão.

[Gomes Júnior et al., 2007] consideram $M_{xy} = 1000,\ \forall\, x, y \in I \cup \{x_0\}$ e $x \neq y$. No entanto, esse valor pode ser melhorado com os limites fornecidos na Seção 3. Dados $x,\, y \in I \cup \{x_0\}$ e $x \neq y$, se $u_{xy} = 1$, então a constante M_{xy} não influencia as restrições (11). Caso contrário, isto é, se $u_{xy} = 1$, a restrição que contém M_{xy} é reduzida a $M_{xy} \geq s_x + P_x - s_y$. Logo, se s_x^{UB} é um limite superior para s_x e s_y^{LB} é um limite inferior para s_y, então pode-se utilizar $M_{xy} = s_x^{UB} + P_x - s_y^{LB}$.

Nova formulação inteira mista – IMW

Nesta Subseção é proposta uma nova formulação inteira mista para representar o PSUMAA, a qual é inspirada no trabalho de Wagner [1959]. Sejam s_x, e_x e t_x variáveis de decisão tais como na formulação IMG e sejam v_{xi} e w_{xyi} variáveis de decisão tais que, $\forall\, x,\, y \in I,\ x \neq y$ e $\forall\, i \in \{1,\, 2, \ldots,\, n\}$:

- $v_{xi} = \begin{cases} 1, & \text{se } x \text{ é a } i\text{-ésima tarefa processada;} \\ 0, & \text{caso contrário.} \end{cases}$

- $w_{xyi} = \begin{cases} 1, & \text{se } x \text{ é a } i\text{-ésima tarefa e } y \text{ é a } i{+}1\text{-ésima tarefa processada;} \\ 0, & \text{caso contrário.} \end{cases}$

A formulação matemática dada pelas equações (18)-(29) também representa o PSUMAA.

$$\min \quad \sum_{x \in I} (\alpha_x e_x + \beta_x t_x) \tag{18}$$

$$\begin{aligned}
\text{s.a.} \quad \sum_{x \in I} v_{xi} &= 1 && \forall\, i \in \{1, 2, \ldots, n\} && (19)\\
\sum_{i=1}^{n} v_{xi} &= 1 && \forall\, x \in I && (20)\\
v_{xi} + v_{y,i+1} - w_{xyi} &\leq 1 && \forall\, x, y \in I,\ x \neq y \ \text{ e} && \\
&&& \forall\, i \in \{1, 2, \ldots, n-1\} && (21)\\
\sum_{x \in I} \left(s_x v_{x,i+1} - (s_x + P_x) v_{xi} - \sum_{y \in I \setminus \{x\}} S_{xy} w_{xyi} \right) &\geq 0 && \forall\, i \in \{1, 2, \ldots, n-1\} && (22)\\
s_x + P_x + e_x &\geq E_x && \forall\, x \in I && (23)\\
s_x + P_x - t_x &\leq T_x && \forall\, x \in I && (24)\\
s_x &\geq 0 && \forall\, x \in I && (25)\\
e_x &\geq 0 && \forall\, x \in I && (26)\\
t_x &\geq 0 && \forall\, x \in I && (27)\\
v_{xi} &\in \{0, 1\} && \forall\, x \in I \ \text{ e} && \\
&&& \forall\, i \in \{1, 2, \ldots, n\} && (28)\\
w_{xyi} &\in \{0, 1\} && \forall\, x, y \in I,\ x \neq y \ \text{ e} && \\
&&& \forall\, i \in \{1, 2, \ldots, n\} && (29)
\end{aligned}$$

Assim como na formulação IMG, a equação (18) representa a função objetivo do problema. As restrições (19) e (20) garantem que cada tarefa será processada exatamente uma vez. As restrições (21) e (22) asseguram que existe tempo suficiente para processar uma tarefa antes de iniciar o processamento da tarefa sucessora. As restrições (23) e (24) determinam a antecipação e o atraso de cada tarefa. As restrições (25)-(29) definem os domínios das variáveis de decisão.

Embora as restrições (22) sejam não-lineares, é possível linearizá-las. Para isso, considere a variável auxiliar q_{xi} tal que $q_{xi} = s_x v_{xi},\ \forall\, x \in I$ e $\forall\, i \in \{1, 2, \ldots, n\}$. Para toda tarefa $x \in I$, se s_x^{UB} é um limite superior para s_x, então $0 \leq q_{xi} \leq s_x$ e $s_x + s_x^{UB}(v_{xi} - 1) \leq q_{xi} \leq s_x^{UB} v_{xi}$.

Uma nova formulação inteira mista que representa o PSUMAA, denotada por IMW, é obtida ao substituir as restrições (22) pelas restrições (30)-(34).

$$\sum_{x \in I} \left(q_{x,i+1} - q_{xi} - P_x v_{xi} - \sum_{y \in I \setminus \{x\}} S_{xy} w_{xyi} \right) \geq 0 \quad \forall\, i \in \{1, 2, \ldots, n-1\} \tag{30}$$

$$q_{xi} \geq 0 \quad \forall\, x \in I \text{ e } \forall i \in \{1, 2, \ldots, n\} \tag{31}$$

$$q_{xi} - s_x \leq 0 \quad \forall\, x \in I \text{ e } \forall i \in \{1, 2, \ldots, n\} \tag{32}$$

$$q_{xi} - s_x^{UB} v_{xi} \leq 0 \quad \forall\, x \in I \text{ e } \forall i \in \{1, 2, \ldots, n\} \tag{33}$$

$$s_x + s_x^{UB} v_{xi} - q_{xi} \leq s_x^{UB} \quad \forall\, x \in I \text{ e } \forall i \in \{1, 2, \ldots, n\}. \tag{34}$$

Formulações indexadas no tempo

Nesta Subseção são apresentadas formulações indexadas no tempo que representam o PSUMAA. Para isso, seja $H_x = \left\{ s_x^{LB}, s_x^{LB} + 1, \ldots, s_x^{UB} \right\}$ o conjunto das possíveis datas de início do processamento da tarefa $x \in I$. Sejam l_{xh} variáveis de decisão tais que, $\forall\, x \in I$ e $\forall\, h \in H_x$,

$$l_{xh} = \begin{cases} 1, & \text{se o processamento da tarefa } x \text{ inicia na data } h; \\ 0, & \text{caso contrário.} \end{cases}$$

De agora em diante, $\lfloor \lambda \rfloor_x = \max(s_x^{LB}, \lambda)$ e $\lceil \lambda \rceil_x = \min(\lambda, s_x^{UB})$, $\forall\, x \in I$ e $\forall\, \lambda \in \mathbb{R}$.

Formulação linear inteira mista de [Rosa e Souza, 2009] – IMR

Nesta Subseção é apresentada a formulação linear inteira mista proposta por [Rosa e Souza, 2009] para representar o PSUMAA, denotada por IMR. Sejam e_x e t_x variáveis de decisão tais como na formulação IMG (vide Seção 4.1). A formulação IMR é dada pelas equações (35)-(42).

$$\min \quad \sum_{x \in I} (\alpha_x e_x + \beta_x t_x) \tag{35}$$

$$\text{s.a.} \quad l_{xh} + \sum_{k = \lfloor h \rfloor_y}^{\lceil h + P_x + S_{xy} - 1 \rceil_y} l_{yk} \leq 1 \quad \forall\, x, y \in I,\ x \neq y \text{ e } \forall\, h \in H_x \tag{36}$$

$$\sum_{h \in H_x} l_{xh} = 1 \quad \forall\, x \in I \tag{37}$$

$$e_x + \sum_{h \in H_x} h \cdot l_{xh} \geq E_x - P_x \quad \forall\, x \in I \tag{38}$$

$$-t_x + \sum_{h \in H_x} h \cdot l_{xh} \leq T_x - P_x \quad \forall\, x \in I \tag{39}$$

$$e_x \geq 0 \quad \forall\, x \in I \tag{40}$$

$$t_x \geq 0 \quad \forall\, x \in I \tag{41}$$

$$l_{xh} \in \{0, 1\} \quad \forall\, x \in I \text{ e } \forall\, h \in H_x \tag{42}$$

A função objetivo, representada pela equação (35), busca a minimização da soma ponderada das antecipações e atrasos. As restrições (36) garantem que existe tempo suficiente para processar uma tarefa e preparar a máquina antes de iniciar o processamento da tarefa seguinte. As restrições (37) garantem que cada tarefa será processada uma única vez. As restrições (38) e (39) determinam as antecipações e os atrasos das tarefas. As restrições (40), (41) e (42) dizem respeito aos domínios das variáveis de decisão.

Restrições válidas de [Nogueira et al., 2014] – IMRN

As restrições (36) presentes na formulação IMR podem ser substituídas pelas restrições (43), apresentadas em [Nogueira et al., 2014] e que são mais fortes que as restrições (36).

$$l_{xh} + \sum_{k=\lfloor h-P_y-S_{yx}+1\rfloor_y}^{\lceil h+P_x+S_{xy}-1\rceil_y} l_{yk} \leq 1, \quad \forall\, x, y \in I,\ x \neq y,\ h \in H_x \tag{43}$$

As restrições (43) asseguram que se o processamento da tarefa $x \in I$ é iniciado na data $h \in H_x$, então nenhuma outra tarefa $y \in I \setminus \{x\}$ pode ser iniciada no intervalo $\{h - P_y - S_{yx} + 1, \ldots, h + P_x + S_{xy} - 1\}$. Em outras palavras, as restrições (43) garantem que não haverá tarefas sendo processadas simultaneamente. A formulação obtida da formulação IMR após a substituição das restrições (36) pelas restrições (43) é denotada por IMRN.

Formulações binárias – BR e BRN

A presença das variáveis de decisão não binárias (variáveis e_x e t_x) nas formulações IMR e IMRN podem ser eliminadas dessas formulações. Para isso, assim como em [Tanaka, 2012], $\forall x \in I$, considere a função custo $g_x : H_x \to \mathbb{R}$ dada por:

$$g_x(h) = \alpha_x \cdot \max(E_x - h - P_x, 0) + \beta_x \cdot \max(h + P_x - T_x, 0). \tag{44}$$

O valor $g_x(h)$ corresponde à penalidade associada à tarefa x caso seu processamento seja iniciado na data h. Logo, a função objetivo das formulações IMR e IMRN, dada pela equação (35), pode ser substituída pela equação (45). Essa substituição torna as variáveis e_x e t_x, $\forall x \in I$, bem como as restrições (38)-(41), desnecessárias.

$$\min \quad \sum_{x\in I} \sum_{h\in H_x} g_x(h) \cdot l_{xh} \tag{45}$$

As formulações obtidas das formulações IMR e IMRN pela eliminação das variáveis e_x e t_x são denotadas por BR e BRN, respectivamente.

Resultados computacionais

As formulações matemáticas apresentadas neste trabalho foram implementadas e resolvidas com a C++ Concert Technology e com o otimizador IBM ILOG CPLEX Optimization Studio 12.6.2. Os experimentos foram realizados em um computador Intel® Xeon(R) CPU E5620 @ 2.40GHz × 16, com 48 GB de RAM e sistema operacional CentOS Linux 7. O CPLEX foi configurado para utilizar somente uma *thread* e os demais parâmetros não foram alterados.

Foram utilizados os conjuntos de problemas-teste de [Rosa e Souza, 2009] que possuem de 8 a 15 tarefas, sendo 16 problemas-teste em cada conjunto de problemas de mesmo tamanho. Esses problemas estão disponíveis em `www.decom.ufop.br/prof/marcone/projects/SMSPETP/Instances_SMSPETP.zip`.

Para cada tarefa $x \in I$, os limites s_x^{UB} e s_x^{LB} utilizados para determinar os parâmetros de entrada de cada formulação matemática são aqueles fornecidos pelo Teorema 1. O tempo total demandado para determinar esses limites (incluindo a resolução do subproblema de determinação de S^I_{max} pelo CPLEX) foi sempre inferior a 1 segundo.

O *gap* de uma solução X em relação à solução X^* é dado por

$$gap = \frac{f(X^*) - f(X)}{f(X^*)} \times 100\%,$$

em que $f(X^*)$ e $f(X)$ representam os valores da função objetivo do problema aplicada a X e a X^*, respectivamente. Logo, quanto mais próximo de zero estiver o *gap* de X, mais próximo essa solução estará de X^*. Neste trabalho, foram utilizadas as melhores soluções encontradas por [Rosa e Souza, 2009] como referências para os cálculos dos *gap*'s.

Primeiramente foram realizados experimentos com as relaxações lineares das formulações IMG, IMW, IMR, IMRN, BR e BRN. Um resumo dos resultados obtidos é apresentado na Tabela 1. Nessa tabela, a primeira coluna indica o número de tarefas em cada conjunto de problemas-teste. Para cada conjunto de problemas, as colunas "$\overline{gap}$" e "$\overline{\text{tempo}}$" apresentam, respectivamente, as médias dos *gap*'s obtidos (em porcentagem) e as médias dos tempos demandados (em segundos) ao se utilizar o CPLEX para resolver as relaxações lineares das respectivas formulações matemáticas aplicadas aos problemas desse conjunto. O carácter "–" diz que o CPLEX não foi capaz de resolver a relaxação linear de pelo menos um dos 16 problemas-teste dentro de uma hora de execução.

Tabela 1: Resultados obtidos com as relaxações lineares das formulações apresentadas na Seção 4.

Nº de Tarefas	IMG		IMW		IMR		IMRN		BR		BRN	
	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)	$\overline{gap}$ (%)	$\overline{\text{tempo}}$ (s)
08	90,7	0,1	90,0	0,1	89,7	0,5	88,7	1,3	86,2	0,5	78,6	1,2
09	94,9	0,1	94,8	0,1	94,0	0,9	93,1	2,0	90,4	0,9	82,6	1,8
10	94,8	0,1	94,8	0,2	94,2	1,9	93,6	3,3	91,1	1,8	84,6	3,4
11	96,1	0,2	96,1	0,2	95,8	3,3	95,3	17,5	91,9	2,3	85,6	4,5
12	94,3	0,3	94,3	0,3	93,6	24,0	92,8	5,9	91,2	3,1	86,0	5,3
13	93,3	0,4	93,3	0,4	92,4	34,9	91,5	17,7	89,5	3,9	83,6	8,0
14	94,5	0,5	94,5	0,5	94,0	5,4	93,4	101,1	91,3	6,2	85,6	11,6
15	93,7	0,8	93,7	0,8	95,1	141,0	88,3	–	92,4	6,8	87,5	12,5

De acordo com a Tabela 1, o CPLEX foi capaz de resolver todas as relaxações lineares dos problemas-testes dentro de uma hora, exceto para os problemas com 15 tarefas, que não tiveram todas as relações lineares da formulação IMRN resolvidas dentro desse limite de tempo. As médias dos resultados obtidos pelo CPLEX com as relaxações lineares das formulações IMG e IMW foram sempre próximos. Embora os tempos médios demandados com as relaxações lineares das formulações indexadas no tempo sejam maiores que os respectivos tempos demandados com as relaxações das formulações IMG e IMW, as relaxações lineares das formulações indexadas no tempo obtiveram soluções de menores *gap*'s médios que os respectivos *gap*'s das soluções das relaxações lineares das formulações IMG e IMW. O CPLEX encontrou soluções de menores *gap*'s médios com a relaxação linear da formulação BR do que com a relaxação da formulação IMR em todos os conjuntos de problemas utilizados. Comportamento semelhante pode ser observado ao se comparar as formulações BRN e IMRN (os *gap*'s médios das soluções obtidas com a relaxação linear da formulação BRN foram sempre menores que os respectivos *gap*'s das soluções encontradas com a relaxação linear da formulação IMRN). Exceto para os problemas com 14 tarefas, os tempos médios demandados pelo CPLEX com a relaxação linear da formulação BR são consideravelmente menores que os respectivos tempos demandados ao se utilizar a relaxação linear da formulação IMR. Novamente, resultados semelhantes foram obtidos com as relaxações das formulações BRN e IMRN. Finalmente, para todos os conjuntos de problemas, a relaxação linear da formulação BRN foi a que obteve soluções de menor *gap* médio.

A Tabela 2 resume os resultados obtidos com as formulações IMG, IMW, IMR, IMRN, BR e BRN. A primeira coluna indica o número de tarefas em cada conjunto de problemas-teste. Para cada um desses conjuntos, as colunas "$\overline{\text{tempo}}$" e "$\overline{\text{nós}}$" mostram, respectivamente, as médias dos tempos demandados (em segundos) e as médias dos números de nós investigados pelo *Branch-and-bound* ao se utilizar o CPLEX com as respectivas formulações matemáticas. Novamente, o carácter "–" diz que o CPLEX não foi capaz de resolver pelo menos um dos 16 problemas-teste dentro de uma hora de execução.

A Tabela 2 mostra que o CPLEX foi capaz de resolver todos os problemas-teste com até 9

Tabela 2: Resultados obtidos com as formulações apresentadas na Seção 4.

Nº	IMG		IMW		IMR		IMRN		BR		BRN	
de Tarefas	tempo (s)	$\overline{\text{nós}}$	tempo (s)	$\overline{\text{nós}}$	tempo (s)	$\overline{\text{nós}}$	tempo (s)	$\overline{\text{nós}}$	tempo (s)	$\overline{\text{nós}}$	tempo (s)	$\overline{\text{nós}}$
08	6,5	20527,8	6,4	4659,4	92,4	431,4	339,5	4109,8	15,0	1,0	34,1	1,0
09	66,2	177675,1	27,5	13232,4	298,7	1002,8	460,3	304,4	28,5	3,7	59,3	2,6
10	528,9	1069311,8	95,8	31768,1	481,2	507,4	–	–	46,0	4,8	122,4	4,4
11	–	–	–	–	–	–	–	–	84,8	26,9	–	–
12	–	–	–	–	–	–	–	–	140,4	10,9	–	–
13	–	–	–	–	–	–	–	–	188,6	42,8	–	–
14	–	–	–	–	–	–	–	–	–	–	–	–
15	–	–	–	–	–	–	–	–	–	–	–	–

tarefas usando a formulação IMRN, todos os problemas com até 10 tarefas usando as formulações IMG, IMW, IMR e BRN, bem como todos os problemas com até 13 tarefas usando a formulação BR, dentro de uma hora de execução. O número médio de nós investigados pelo *Branch-and-bound* do CPLEX com as formulações IMG e IMW foram consideravelmente maiores que os respectivos números de nós investigados ao se utilizar as formulações indexadas no tempo. O CPLEX com as formulações BR e BRN foi capaz de resolver todos os problemas com 8 tarefas já na raiz da árvore de busca. As formulações inteiras BR e BRN proporcionaram ao CPLEX resolver os respectivos conjuntos de problemas-teste em menor tempo médio e investigando um menor número de nós, em média, que as respectivas formulações inteiras mistas IMR e IMRN.

Conclusões

Este trabalho tratou o problema de programação de tarefas em uma máquina com penalidades por antecipação e atraso da produção, considerando a existência de janelas de entrega distintas e de tempos de preparação da máquina dependentes da sequência de processamento programada (PSUMAA).

Propôs-se um horizonte de planejamento para o processamento de cada tarefa. Além disso, foram propostas duas novas formulações inteiras mistas (IMW e IMRN, sendo a formulação IMRN indexada no tempo) e duas novas formulações binárias indexadas no tempo (BR e BRN) para representar o PSUMAA. O horizonte de planejamento proposto é utilizado para determinar os parâmetros de entrada das formulações matemáticas.

As formulações propostas foram comparadas com a formulação inteira mista de [Gomes Júnior et al., 2007] (IMG) e com a formulação inteira mista indexada no tempo de [Rosa e Souza, 2009] (IMR). Para isso, utilizou-se o otimizador CPLEX e um conjunto de problemas-teste com até 15 tarefas. Na primeira bateria de testes foram resolvidas as relaxações lineares das formulações apresentadas aplicadas aos problemas-teste. As formulações IMG e IMW obtiveram soluções semelhantes, ao passo que as formulações indexadas no tempo obtiveram soluções, em média, melhores que essas duas. Dentre todas as formulações experimentadas, a relaxação linear da formulação BRN foi a que obteve as melhores soluções.

Finalmente, o CPLEX foi utilizado para resolver as seis formulações apresentadas aplicadas aos problemas-teste. Fixado o limite de tempo de uma hora, as formulações IMG, IMW, IMR e BRN permitiram ao CPLEX resolver todos os problemas com até 10 tarefas. No entanto, as formulações IMG e IMW exigiram que um número consideravelmente maior de nós fossem investigados durante o *Branch-and-Bound*. A formulação IMRN foi a que apresentou pior desempenho (possibilitando ao CPLEX resolver na totalidade somente os problemas com até 9 tarefas) e a formulação BR foi a que obteve melhor desempenho (permitindo ao CPLEX resolver todos os problemas com até 13 tarefas) entre todas as formulações avaliadas.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), ao CEFET-MG, à UFOP e à UAPV pelo apoio.

Referências

Allahverdi, A., Gupta, J. N. D., e Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2):219–239.

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2):345 – 378.

Allahverdi, A., Ng, C., Cheng, T., e Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032.

Baker, K. R. e Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38:22–36.

Bustamante, L. M. (2007). 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, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte.

Gomes Júnior, A. C., Carvalho, C. R. V., Munhoz, P. L. A., e Souza, M. J. F. (2007). Um método heurístico híbrido para a resolução do problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção. In *Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional*, p. 1649–1660. SOBRAPO.

Gutin, G. e Punnen, A. P. (2007). *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media.

Janiak, A., Janiak, W. A., Krysiak, T., e Kwiatkowski, T. (2015). A survey on scheduling problems with due windows. *European Journal of Operational Research*, 242(2):347 – 357.

Miller, C. E., Tucker, A. W., e Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329. ISSN 0004-5411.

Nogueira, T. H., de Carvalho, C. R. V., e Ravetti, M. G. (2014). Analysis of mixed integer programming formulations for single machine scheduling problems with sequence dependent setup times and release dates. *4OR: A Quarterly Journal of Operations Research (submitted)*.

Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*. Springer US, 4 edition.

Rosa, B. F. e Souza, M. J. F. (2009). Uma nova formulação de programação matemática indexada no tempo para uma classe de problemas de seqüenciamento em uma máquina. In *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, p. 2898–2909, Porto Seguro (BA). SOBRAPO.

Tanaka, S. (2012). An exact algorithm for the single-machine earliness-tardiness scheduling problem. In Ríos-Mercado, R. Z. e Ríos-Solís, Y. A., editors, *Just-in-Time Systems*, volume 60 of *Springer Optimization and Its Applications*, p. 21 – 40. Springer New York.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2):131–140.