

REGRA DE PRIORIDADE PARA O PROBLEMA DE FLOWSHOP HÍBRIDO COM MÁQUINAS PARALELAS NÃO RELACIONADAS

LEITE, Ítalo Geraldo Donadoni¹

PEREIRA, Ana Amélia de Souza²

Revista
Científica
Fagoc

ISSN: 2525-488X

Multi
disciplinar

RESUMO

Esta pesquisa propõe o estudo de uma regra de prioridade proposta na literatura Priority Rule for Total Tardiness adaptada (PRTTa), aplicada em um ambiente flowshop híbrido com máquinas paralelas não relacionadas. A função objetivo considerada é a minimização das somas dos atrasos em relação às suas datas de entrega. As tarefas representam um único trabalho, e seu processamento não pode ser interrompido; elas estão disponíveis para processamento em diferentes instantes de tempo, de acordo com sua conclusão nos estágios, e seus tempos de processamento são diferentes para cada máquina. Os tempos de preparação são dependentes tanto da tarefa quanto da máquina. Foi adaptado para o problema um modelo de programação inteira (PLI) no software Cplex. A regra em questão é comparada às regras EDD (Earliest Due Date), SPT (Shortest Processing Time) e com os resultados obtidos no Cplex. Verificou-se que a regra PRTTa obteve bom desempenho para o problema e foi superior às outras regras, portanto trata-se de uma boa escolha para programação da produção de um ambiente flowshop híbrido com máquinas paralelas não relacionadas.

PALAVRAS-CHAVE: Regra de Prioridade. Sequenciamento da Produção. Flowshop Híbrido. Máquinas Paralelas Não Relacionadas.

INTRODUÇÃO

Um dos problemas que as empresas enfrentam é a falta de organização de sua produção, o que ocasiona atrasos na entrega dos produtos, sendo prejudicial, pois acarreta um incômodo ao cliente. Existem diferentes técnicas para melhoria dos processos de produção, umas delas são as regras de prioridade para o sequenciamento da produção. O problema de sequenciamento trata da disponibilidade de recursos para execução de um conjunto de tarefas em um dado intervalo de tempo, com o foco de otimizar um ou mais objetivos (PINEDO, 2012). O processo envolve a escolha da melhor sequência em que um lote de tarefas deverá ser executado dado um conjunto de recursos. É necessário controle das atividades que envolvem os processos de produção, de forma a evidenciar as atividades que agreguem valor ao consumidor final.

O problema de programação da produção de um ambiente flowshop híbrido com máquinas paralelas não relacionadas aborda a situação em que várias máquinas operam em paralelo e realizam a mesma função com tempos de processamento diferentes.

O trabalho trata o problema de minimização do atraso total no ambiente flowshop híbrido com máquinas paralelas não relacionadas. Esse ambiente é caracterizado por existirem n tarefas e s estágios em série, cada estágio com um certo número m de máquinas. O

1 Faculdade Governador Ozanam Coelho.

E-mail: idonadoni@hotmail.com

2 Faculdade Governador Ozanam Coelho.

E-mail: aamelia.mg@gmail.com

tempo de processamento de cada tarefa depende da tarefa, da máquina e do estágio. Todas as tarefas devem passar por todos os estágios na mesma ordem, cada tarefa deve ser processada exclusivamente por uma máquina de cada estágio e cada máquina deve processar apenas uma tarefa por vez. Vale ressaltar ainda que os tempos de processamento de cada tarefa são diferentes em cada máquina.

Existem três tipos de máquinas no ambiente flowshop híbrido: máquinas idênticas, que têm a mesma capacidade de produção e desenvolvem os trabalhos na mesma velocidade (OLIVEIRA; TURRIONI, 2012); máquinas uniformes, cada máquina dentro de um estágio escolhido tem uma velocidade relacionada e esta será utilizada como parâmetro para encontrar o novo tempo de processamento da tarefa naquela máquina (OLIVEIRA; TURRIONI, 2012); e máquinas não relacionadas, cujo tempo de processamento depende de cada máquina.

Em Brah (1996) foi realizada uma análise do desempenho de regras de prioridade, considerando atrasos médio e máximo. Neste trabalho foram examinados os efeitos de características do problema, como, por exemplo, número de estágios, número de máquinas em paralelo em cada estágio e número de tarefas. A análise foi feita sobre as regras *Earliest Due Date* (EDD), *Modified Due Date* (MDD), *Shortest Processing Time* (SPT-T), *Shortest Processing Time S/RPT+SPT*, *Cost Over Time* (COVERT) e *Apparent Tardiness Cost* (ATC), conclui-se que as regras EDD e MDD apresentam os melhores desempenhos para atraso máximo e atraso médio.

Ludwig, Anzanello e Vidor (2013) relatam que, em pequenas empresas do ramo de software que usam com frequência programação da produção de suas tarefas de forma manual, é viável enquanto o número de tarefas for reduzido, pois à medida que as tarefas aumentam, implicam dificuldade em gerenciá-las. Foram propostas duas heurísticas para a programação de tarefas, as quais têm como características: i) definição de ordenamento inicial, ii) alocação de tarefas ao time de desenvolvimento e iii) ordenação com

foco na minimização dos atrasos. As heurísticas propostas para resolução do problema de atrasos da produção chegaram ao seu objetivo de minimizar os tempos de atraso.

Existem trabalhos que abordam regras de prioridades a fim de minimizar os tempos de atraso da produção. Mello (2004) utiliza métodos heurísticos conhecidos na literatura (SPT, LPT, EDD e EDD*) que fornecem soluções viáveis e aceitáveis para o problema, entre elas uma idealizada por Ronconi e Armentano (1999), LB, que explora a minimização da somatória dos atrasos totais. Os resultados obtidos no trabalho apresentaram uma diminuição considerável no atraso médio da produção.

Mainieri e Ronconi (2010) utilizaram as regras MDD e PRTT, que foram avaliadas em um grupo de 4.320 problemas teste com o objetivo de minimizar o *makespan* (soma dos tempos de processamento). Os testes e comparações com outras regras de prioridades (SPT, EDD, MDD, MDDa (*Modified Due Date* adaptada), ATC e COVERT) mostram que as duas regras têm um desempenho melhor para o problema.

Em Fuchigami e Moccellin (2015), foi discutida a eficiência da utilização de regras de prioridades para programação de sistemas produtivos conhecidos como *flexible flow line*, com tempos de setup dependentes da sequência de processamento das tarefas. O objetivo foi indicar novas regras de prioridades e avaliar o desempenho delas no sistema produtivo, considerado por meio da porcentagem de sucesso, desvio relativo médio, desvio padrão do desvio relativo e tempo médio de computação.

No artigo de Rocha (2006), é tratado um problema de sequenciamento em máquinas paralelas não relacionadas com tempos de preparação dependentes da máquina e da sequência, datas de entrega e tarefas ponderadas. Foram propostos dois modelos de programação para solução do problema, uma inteira e uma mista, baseadas em estratégias clássicas da literatura. Foi desenvolvido um algoritmo *Branch-and-bound* (B&B) e a solução encontrada pela metaheurísticas GRASP é utilizada como limite

superior.

No trabalho de Etcheverry e Anzanello (2014), foi proposta uma heurística para sequenciamento de tarefas em um ambiente de manufatura composto por máquinas paralelas não relacionadas com tempos de setup dependentes da sequência de processamento das tarefas. Uma heurística de três etapas foi sugerida para fazer a ordenação das tarefas, alocar tarefas à máquina e, através da metaheurísticas Tabu Search, aprimorar a solução final. Foi feita uma experimentação computacional que indicou que as regras que utilizam a ordenação crescente dos tempos de processamento e setup apresentam melhor atuação;

Em Jungwattanakit et al. (2008), considera-se o problema da programação de tarefas independentes no ambiente flowshop híbrido; as tarefas são ordenadas e processadas nas máquinas de cada estágio. O problema considerado foi determinar um cronograma em que se minimiza uma combinação de tarefas no makespan e do número de tarefas atrasadas, então foi desenvolvido um programa com programação inteira mista. Regras de prioridades e heurísticas polinomiais foram empregadas para melhorar o problema de atraso das tarefas.

Objetivos gerais e específicos

O presente trabalho tem como objetivo analisar e aplicar regras de prioridades para o problema de minimização do atraso total no ambiente *flowshop* híbrido com máquinas paralelas não relacionadas. De acordo com Ronconi (2004), regras de prioridade são fáceis de implementar e não exigem grandes testes computacionais para ajustá-las; são valiosas em aplicações práticas em empresas que apresentam o problema de atrasos em sua produção, sendo que é uma das razões que motivaram este estudo. As regras de prioridades selecionadas para os testes foram EDD, SPT e PRTTa e, segundo Pinedo (2012), são regras simples que apresentam bom desempenho para o problema considerado.

As regras citadas terão seus resultados

comparados entre elas e, com os resultados obtidos, em um modelo matemático desenvolvido no software *Cplex*, essas comparações levaram à indagação se a regra PRTTa é melhor que as outras regras e se seu desempenho é favorável para resolver o problema de minimizar o atraso da produção.

Método de desenvolvimento

Para avaliar o desempenho das regras de prioridade, foram realizadas comparações com os resultados obtidos no software *Cplex* em problemas de pequeno, médio e grande porte. Esse tipo de comparação nesse ambiente é pouco utilizado na literatura por apresentar uma dificuldade em lidar com tempo de preparação *setup*, pois a troca de setup acarreta uma perda da sincronia entre o planejamento da produção; buscar, então, reduzir o número de interrupções se faz necessário para a demanda de uma produção. Outra dificuldade é as máquinas terem tempo de processamento diferentes umas das outras.

Organização do trabalho

A próxima seção contém os estudos e tecnologias existentes sobre o tema. A Seção 3 contém a descrição do problema. Na Seção 4, são apresentados métodos e soluções para o problema. Na Seção 5, são expostos os resultados obtidos; e a última Seção apresenta as considerações finais do trabalho.

REFERENCIAL TEÓRICO

Nesta seção, estão relacionados estudos e tecnologias existentes relevantes para este trabalho.

Problema de programação da produção

Sequenciamento de produção é um importante processo de tomada de decisão a

nível operacional usado em muitas indústrias de serviço e manufatura (NOGUEIRA et al., 2014).

O sequenciamento de tarefas é utilizado nas empresas manufatureiras e é constituído pelo processo de ordenação das tarefas com a escolha do recurso adequado para sua execução, sendo sempre consideradas restrições de tempo e de relacionamento entre as tarefas (PINEDO, 2008).

Problemas de sequenciamento lidam com a alocação de recursos para execução de um conjunto de tarefas em um dado intervalo de tempo e seu objetivo é otimizar um ou mais critérios (PINEDO, 2012). A programação da produção pode ser definida genericamente como a alocação de recursos disponíveis para a execução de tarefas em um horizonte de tempo (FUCHIGAMI; MOCCELLIN, 2015).

O problema de programação da produção em uma empresa de manufatura está associado à alocação de recursos a um conjunto de tarefas num dado tempo, em geral podem ser definidas como um conjunto de j tarefas $\{J_1, J_2, \dots, J_n\}$ que devem ser processadas em um conjunto de m máquinas $\{M_1, M_2, \dots, M_n\}$ alocadas em l estágios $\{L_1, L_2, \dots, L_n\}$. O processamento da tarefa J_1 na máquina M_1 por exemplo é denominada operação. Para cada tempo de operação da tarefa na máquina, denominamos seu processamento por tempo de processamento.

Os problemas de produção são classificados em três níveis: estratégico, tático e operacional (ARENALES et al., 2007). O problema tratado se enquadra no nível operacional, pois seu objetivo é se responsabilizar pelas atividades diárias de produção.

Ambiente de produção *flowshop*

No ambiente *flowshop*, todas as tarefas têm o mesmo fluxo de processamento nas máquinas, sendo que cada máquina m está em série, ou seja, as tarefas possuem a mesma sequência de processamento em todas as máquinas e o número de máquinas em cada estágio de produção é igual a um.

Segundo Pinedo (2012):

Em muitas instalações de fabricação e montagem, cada trabalho tem de passar por uma série de operações. Muitas vezes, essas operações têm de ser feitas em todos os postos de trabalho na mesma ordem, o que implica que os trabalhos têm de seguir o mesmo caminho. São assumidas as máquinas em seguida, para ser configurado em série e o ambiente é referido como um *flowshop*.

Existem várias generalizações do *flowshop*, como o *flowshop* flexível e *flowshop* híbrido, em que, em vez de m máquinas em série, existem s estágios em série com um número de máquinas em paralelo em cada estágio.

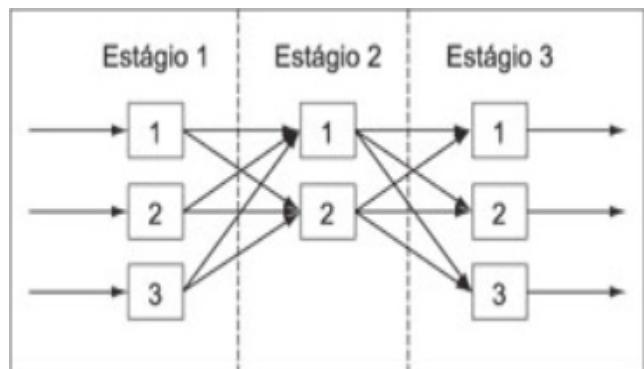
Ambiente de produção *flowshop* flexível

Neste ambiente, o tempo de processamento de uma tarefa depende da tarefa e do estágio, ou seja, é independente da máquina pertencente a um estágio.

Pinedo (2002L) define *flowshop* flexível assim:

O *flowshop* flexível é uma generalização do *flowshop* com máquinas em paralelo. Ao invés de m máquinas em série, existem c centros de trabalho agrupando máquinas idênticas que funcionam em paralelo. Cada trabalho tem que ser processado no centro de trabalho 1, 2 e assim por diante. Um trabalho é processado em uma máquina de um centro de trabalho, mas todas as máquinas deste centro poderiam processá-lo da mesma forma.

Figura 1: Exemplo de *flowshop* flexível



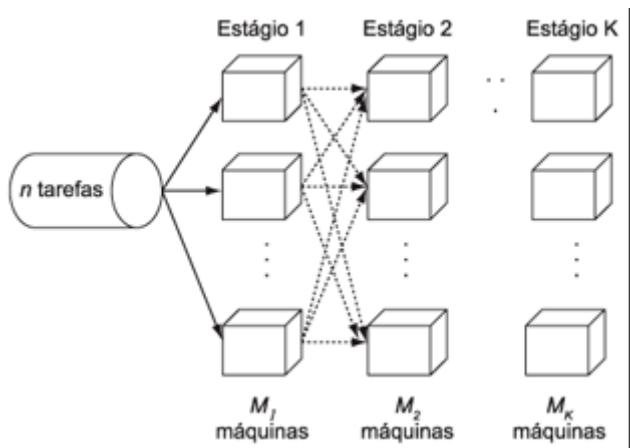
Flowshop híbrido

No ambiente *flowshop* híbrido pelo menos um estágio tem mais de uma máquina.

Segundo Pinedo (2012), a definição para *flowshop* híbrido é:

O *flowshop* híbrido é uma generalização do *flowshop* e os ambientes de máquinas paralelas. Em vez de m máquinas em série existem s estágios em série, em cada fase de um número de máquinas em paralelo. Cada tarefa tem de ser processada primeiro na fase 1, em seguida, na fase 2, e assim por diante.

Figura 2: Exemplo de *flowshop* híbrido



Máquinas paralelas não relacionadas

Segundo Pinedo (2008), a definição de máquinas paralelas não relacionadas é que o tempo de processamento da tarefa é dependente, simultaneamente, da velocidade da máquina e da tarefa. Máquinas paralelas não relacionadas são caracterizadas pelo tempo de processamento de um trabalho que depende da atribuição da máquina (CHEN et al., 1998).

Existem três tipos de máquinas no ambiente *flowshop* híbrido: máquinas idênticas, cuja capacidade de produção é a mesma e desenvolvem os trabalhos na mesma velocidade (OLIVEIRA; TURRIONI, 2012); máquinas uniformes, em que cada máquina dentro de um estágio escolhido tem uma velocidade relacionada e esta será utilizada como parâmetro para encontrar o novo tempo de processamento da tarefa naquela máquina (OLIVEIRA; TURRIONI, 2012); e máquinas não relacionadas onde o tempo de processamento depende de cada máquina.

Escolhido tem uma velocidade relacionada e esta será utilizada como parâmetro para encontrar o novo tempo de processamento da tarefa naquela máquina (OLIVEIRA; TURRIONI, 2012); e máquinas não relacionadas onde o tempo de processamento depende de cada máquina.

Máquinas paralelas não relacionadas

Segundo Pinedo (2008), a definição de máquinas paralelas não relacionadas é que o tempo de processamento da tarefa é dependente, simultaneamente, da velocidade da máquina e da tarefa. Máquinas paralelas não relacionadas são caracterizadas pelo tempo de processamento de um trabalho que depende da atribuição da máquina (CHEN et al., 1998).

Existem três tipos de máquinas no ambiente *flowshop* híbrido: máquinas idênticas, cuja capacidade de produção é a mesma e desenvolvem os trabalhos na mesma velocidade (OLIVEIRA; TURRIONI, 2012); máquinas uniformes, em que cada máquina dentro de um estágio escolhido tem uma velocidade relacionada e esta será utilizada como parâmetro para encontrar o novo tempo de processamento da tarefa naquela máquina (OLIVEIRA; TURRIONI, 2012); e máquinas não relacionadas onde o tempo de processamento depende de cada máquina.

Tempo de preparação (*setup time*)

Tempo de *setup* é o tempo de preparação das máquinas, ou seja, quando a máquina é interrompida para que os equipamentos possam ser alterados e ou ajustados.

No modelo de produção com sistema em lotes (várias tarefas em um certo tempo de produção), o tempo de *setup* está presente devido à necessidade de se produzir grande variedade de produtos, o que é fundamental para que garanta boa produtividade. É necessário que o planejamento da produção e a variedade estejam em sincronia, então existe a necessidade da disponibilidade de lotes na produção, para que

se tenha melhor performance. Busca-se então a redução do número de interrupções nos tempos de preparação das tarefas da produção.

Ao adquirir novos maquinários para os estágios em que as máquinas necessitam de trabalhar em paralelo, são obtidos tempos de *setup* diferentes entre as máquinas que já estavam em uso e as novas.

Para Shingo (2008), o tempo de *setup* se dá desde a preparação, ajustes pós-processamento e verificação de matéria-prima, matrizes, guias entre outros.

Os tempos de preparação das máquinas ou tempos de *setup* são:

- *Setup* dependente da sequência: ocorre quando o tempo depende tanto da tarefa a ser processada como daquela que foi executada imediatamente antes na mesma máquina;
- *Setup* independente da sequência: aquele em que o tempo depende apenas da tarefa a ser processada.

Mesmo com sistemas produtivos híbridos (onde há duas ou mais maneiras em conjunto para proporcionar uma maior eficiência no sistema), ainda existem ambientes de produção com a necessidade de redução do tempo de preparação das tarefas, principalmente quando o tempo de preparação se sobressai sobre o tempo de processamento.

Regras de prioridade

De acordo com Morais, Menegarde e Cantiere (2009), as regras de prioridade são métodos que determinam a ordem de execução das tarefas nas máquinas, ou seja, definem a direção específica em qual máquina as tarefas serão processadas. Existem muitas regras de prioridade aplicáveis aos problemas de produção em sistemas flowshop híbrido, as quais estão fortemente ligadas aos objetivos (RONCONI, 2012).

Algumas regras de prioridade que apresentam um bom desempenho para o problema de minimização do atraso são descritas por Pinedo (2012):

- SPT (*Shortest Processing Time*): dá prioridade à operação com menor tempo de processamento.

- MST (*Minimum Slack Time*): prioriza a tarefa com a menor folga em relação à sua data de entrega. Utiliza os tempos médios de processamento das operações em suas diversas máquinas alternativas.

- SCR (*Smallest Critical Ratio*): prioriza a tarefa com menor relação entre o tempo disponível até sua data de entrega e o tempo ainda demandado de processamento. Esse cálculo também utiliza os tempos médios de processamento das operações.

- EDD: (*Earliest Due Date*): prioriza a tarefa com menor data de entrega.

- MDD: (*Modified Due Date*): faz a modificação da data de entrega da tarefa baseada no tempo de processamento das operações restantes. Como essas operações ainda não têm máquinas definidas, é utilizado o tempo de processamento mínimo entre as máquinas habilitadas.

Cplex

O Cplex é um *software* de otimização utilizado em diversas técnicas para tentar resolver modelos de pesquisa operacional. Ele resolve problemas de programação linear (de grande escala e de fluxo em redes), programação quadrática, programação por restrições quadráticas e programação mista. Possui três tipos de interfaces: ILOG, Concert e Independent modelling systems³.

DESCRIÇÃO DO PROBLEMA

Esta seção do trabalho trata do problema através de um modelo matemático, considerando que os tempos de processamento de cada tarefa não são iguais e cada tarefa processada na máquina em diferentes estágios tem um tempo de 3 IBM ILOG CPLEX Optimization Studio. Disponível na Internet via WWW. <<https://www.ibm.com/bs-en/market-place/ibm-ilog-cplex>>. Acesso em: 05 jul. 2017.

processamento diferente. As tarefas representam um único trabalho, cujo processamento não pode ser interrompido.

Considera-se que existam tempos de preparação dependentes tanto das tarefas quanto da máquina e estágio, ou seja, o tempo de preparação da máquina i para processar a tarefa j após ter executado a tarefa l é diferente do tempo de preparação para processar a tarefa l após ter executado a tarefa j. Diferentes máquinas paralelas, i e i0, também tem tempos de preparação diferentes, por serem não relacionadas. Desta forma $S_{tijk} \neq S_{tilj} \neq S_{ti'jk} \neq S_{ti'lj}$.

O modelo matemático para programação linear inteira, para o problema de programação de máquinas paralelas não relacionadas com tempos de preparação dependentes da tarefa e da máquina é uma versão adaptada de Jungwattanakit et al. (2008):

Modelo matemático

j, k, l índice para tarefas;

t índice para estágio;

i índice para máquina;

Parâmetros:

n número de tarefas;

mt número de máquinas do estágio t;

s número de estágios;

d_j data de entrega da tarefa j;

S_{tijk} tempo de preparação da tarefa j para a tarefa k no estágio t na máquina i;

p_{tji} tempo de processamento da tarefa j no estágio t na máquina i;

Variáveis:

X_{tijk} assume o valor 1 se a tarefa k é designada após a tarefa j na máquina i do estágio t, caso contrário, assume valor zero;

C_{tj} instante de conclusão da tarefa j ao término do processamento no estágio t;

T_j atraso da tarefa j;

$$\text{Minimizar } \sum_{j=1}^n T_j \quad (1)$$

Sujeito a:

$$\sum_{i=1}^{mt} \sum_{j=0}^n X_{tijk}^{\square} = 1, \forall t, k, j \neq k \quad (2)$$

$$\sum_{i=1}^{mt} \sum_{k=1}^n X_{tijk}^{\square} \leq 1, \forall t, j, k \neq k \quad (3)$$

$$\sum_{k=1}^n X_{tik}^{\square} = 1, \forall t, i, k \quad (4)$$

$$\sum_{h=0}^n X_{thkj}^{\square} \geq X_{tijk}^{\square}, \forall t, i, j, h, k; h \neq k \quad (5)$$

$$X_{jjit}^t = 0, \forall t, i, j \quad (6)$$

$$C_{t0} = 0, \forall t, j \quad (7)$$

$$C_{tk} + M * (1 - X_{tijk}^{\square}) \geq C_{tj} + S_{tijk} + p_{tki}, \forall t, s, j, k = 1, \dots, n; j \neq k \quad (8)$$

$$C_{tj} + M * (1 - X_{tij0}^{\square}) \geq C_{t0} + p_{tji}, \forall t, i, j \quad (9)$$

$$C_{tj} - C_{(t+1)j} \geq \sum_{i=1}^{mt} \sum_{k=1}^n X_{tijk}^{\square} * S_{tijk} + \sum_{i=1}^{mt} X_{tik0} + p_{tki}, \forall j = 1, \dots, n; t = 1, \dots, s-1 \quad (10)$$

$$C_{(t+1)j} - C_{tj} \geq p_{(t+1)ik}, \forall t, j, i \quad (11)$$

$$T_j \geq C_{tj} - d_j, \forall j \quad (12)$$

$$T_j \geq 0, \forall j = 1, \dots, n \quad (13)$$

$$C_{tj} \geq 0, \forall j, t \quad (14)$$

A equação 1 representa a minimização da soma dos atrasos das tarefas.

A equação 2 garante que apenas uma tarefa j precede uma tarefa k em alguma máquina do estágio t, cada tarefa j tem uma única tarefa predecessora k imediata em uma única máquina i.

A equação 3 garante que no máximo uma tarefa k sucede uma tarefa j em uma certa máquina i do estágio t.

A equação 4 garante que apenas um trabalho será atribuído à primeira e última posições, respectivamente, em cada máquina em cada etapa.

A equação 5 garante que as tarefas são devidamente ligadas a máquina: se uma dada

tarefa j é processada em uma dada máquina i, um predecessor h deve existir em alguma máquina.

A equação 6 garante que o tempo de conclusão da tarefa 0, tarefa de inicialização das máquinas, seja igual a 0.

A equação 7 garante que se uma tarefa j e outra k estão designadas para a mesma máquina de um certo estágio, então a tarefa j deve ser processada antes ou depois da tarefa k. Caso contrário, não existe restrição de precedência entre as duas tarefas.

A equação 8 aplica-se apenas a empregos que são atribuídos a primeira sequência em cada máquina, ou seja, o trabalho não pode iniciar o seu tratamento antes de a disponibilidade da máquina.

A equação 9 especifica as restrições de precedência conjuntivas para os postos de trabalho, que diz que um trabalho não pode iniciar o seu tratamento no estágio $t + 1$ antes de terminar a fase de t .

A equação 10 especifica as restrições de precedência conjuntivo para os postos de trabalho, que diz que um trabalho não pode iniciar o seu tratamento no estágio $t + 1$ antes de terminar a fase de t .

A equação 11 garante a precedência de uma tarefa entre estágios, ou seja, uma tarefa não pode ser processada no estágio $t+1$ antes de ter sido processada no estágio t .

A equação 12 é o conjunto de restrições que garante, para cada tarefa j, que o atraso seja $\max\{0, C_{tj} - dj\}$.

A equação 13 garante que o atraso não seja negativo.

A equação 14 garante que C será sempre positivo.

REPRESENTAÇÃO E AVALIAÇÃO DE UMA SOLUÇÃO

Uma solução do problema é representada por uma permutação de tarefas, a qual representa a ordem de processamento das tarefas designadas em uma máquina paralela não relacionada

de um determinado estágio. Um exemplo de uma solução é representado pela instância do problema formada por tarefas $n = 5$, máquinas $m = 3$ e estágios $s = 5$. Para designar as tarefas à máquina neste exemplo a sequência é dada por $\pi = [3,4,1,5,2]$, os tempos de processamento são representados por P_{sij} , sendo as tarefas $j = 1, \dots, 5$ e as máquinas $i = 1, 2, 3$, com:

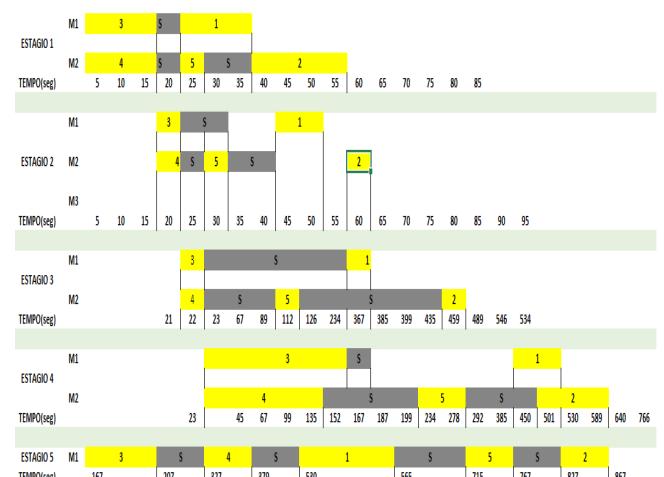
Tabela 1: Tempos de processamento das tarefas

	Processamento									
	Estágio 1		Estágio 2			Estágio 3		Estágio 4		Estágio 5
	Maq1	Maq2	Maq1	Maq2	Maq3	Maq1	Maq2	Maq1	Maq2	Maq1
T1	17	24	12	14	16	2	3	145	150	35
T2	15	18	7	7	9	2	3	148	151	40
T3	15	18	6	7	9	2	3	144	146	40
T4	13	13	5	5	7	3	4	130	130	52
T5	13	13	5	5	7	3	4	130	130	52

os tempos de *setup* são representados por S_{sijk} sendo $S_{1131}=[10]; S_{1245} [3]; S_{1252} [15]; S_{2131} [13]; S_{2245} [3]; S_{2252} [16]; S_{3131} [600]; S_{3245} [100]; S_{3252} [840]; S_{4131} [21]; S_{4245} [10]; S_{4252} [30]; S_{5134} [45]; S_{5141} [150]; S_{5115} [60]; S_{5152} [120]$, as datas de entregas da tarefa $d_j = [502 1000 222 300 990]$ e o tempo total de conclusão ttc ou djl . O atraso é calculado pela diferença entre d_j e ttc , dessa forma $T = d_j - djl$

Na Figura. 3, é apresentado um exemplo do problema com tempos total de conclusão das tarefas $ttc = [3,4,1,5,2]$ sendo, respectivamente, 207,379,565,767e 867.

Figura 3: Uma solução para o problema



Priority Rule for Total Tardiness adaptada (PRTTa)

A regra PRTTa estima datas de entrega para os estágios e antevê estados futuros do sistema, utiliza os tempos das operações para estimar suas datas de entrega, ou seja, a data de entrega da tarefa menos o tempo médio de processamento das operações restantes, a Equação (15) apresenta a formulação geral (MAINIERI; RONCONI, 2010):

$$\text{PRTTa } (j, l, T) = \alpha \cdot \max(r_j, T) + \max[\max(r_j, T) + P_{jl}, d_{jl}] \quad (15)$$

Neste trabalho foi necessário acrescentar o tempo de preparação (setup time), para atender o ambiente tratado que representa uma realidade existente nas empresas. A Equação (16) apresenta a fórmula:

$$\text{PRTTa } (j, l, T) = \alpha \cdot \max(r_j, T) + \max[\max(r_j, T) + P_{jl} + S_{jl}, d_{jl}] \quad (16)$$

Para melhor entendimento as variáveis e parâmetros utilizados são descritos:

j = tarefa considerada;

t= estágio;

T = instante de aplicação da regra (momento em que a tarefa termina seu processamento no estágio t)

α = parâmetro constante que pondera a prioridade do primeiro termo da regra;

d_{ij} = data de entrega

r_j = conclusão da tarefa no estágio t

P_{ji} = processamento da tarefa j no estágio t

S_{jl} = tempo de setup.

Seja j a tarefa a ser considerada, T o instante em que a regra foi aplicada, ou seja o instante que a primeira tarefa começa seu processamento na máquina do estagio, t o estágio, r_{ij} o instante em que a tarefa j processada no estágio t foi liberada e α parâmetro que define a prioridade da primeira tarefa definida na regra.

Neste capítulo, serão apresentados detalhes da regra de prioridade selecionada para minimização do atraso total no ambiente *flowshop* híbrido com máquinas paralelas não relacionadas. As regras de prioridades utilizam o método para a programação de tarefas em ambientes produtivos: o método de programação em ordem direta (do primeiro para o último estágio – *forward*).

Em sistemas complexos (onde existe alto custo na produção) como o considerado neste trabalho, os métodos de programação das operações são muito utilizados. Sempre que uma máquina de um estágio fica disponível, ela é selecionada para processar a tarefa de maior prioridade que está programada, tarefa essa programada por uma regra de prioridade que através de seus recursos estabelece a melhor sequência que as tarefas serão processadas por essas máquinas.

Na literatura existem inúmeras regras de prioridade conhecidas para problemas de sequenciamento de tarefas. As regras utilizadas neste trabalho para comparações foram: SPT e EDD, que segundo Pinedo (2012) são regras simples que apresentam bom desempenho para o problema de minimização do atraso.

Outra regra para minimização do atraso total é a regra PRTT (Priority Rule for Total Tardiness), proposta por Chu e Portmann (1992) para um ambiente com máquina única e tarefas que possuem diferentes instantes de liberação. Uma adaptação da regra foi proposta por Mainieri e Ronconi (2010), chamada de PRTTa (Priority Rule for Total Tardiness adaptada). Esta regra tem o objetivo de estimar datas de entrega para os estágios no ambiente *flowshop* flexível com máquinas paralelas idênticas.

A regra PRTTa neste trabalho foi adaptada para o ambiente *flowshop* híbrido com máquinas paralelas não relacionadas, por apresentar eficiência nos resultados apresentados por Mainieri e Ronconi (2010).

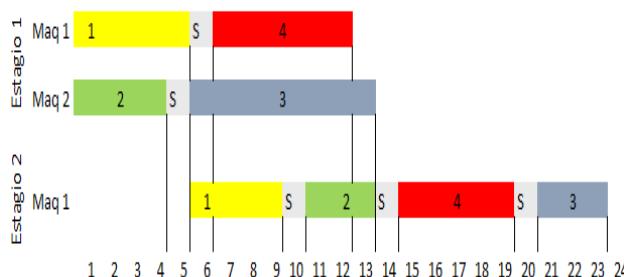
O termo $\alpha \cdot \max(r_j, T)$, pondera uma prioridade maior para a tarefa que não está disponível no instante $t(r_{ij}>T)$.

O método de programação PRTTa tem uma tarefa j programada para todo instante T onde exista uma máquina disponível, esta tarefa ainda não foi programada com menor tempo PRTTa, o desempate do tempo das tarefas é dado pelo menor instante de conclusão da máquina considerada. Se a tarefa que foi programada não estiver disponível no instante T , o tempo ocioso será preenchido pela tarefa gerada que possam ser programadas antes que a tarefa j tenha início.

O tempo ocioso ocorre quando uma tarefa termina seu processamento no estágio atual, mas não pode ser processada no próximo estágio, pois não foi programada pela regra PRTTa, o procedimento para o preenchimento desse tempo é feito da seguinte forma: se existir uma máquina disponível no instante T programe a tarefa j que ainda não foi programada com menor PRTTa, caso ocorra empate (mesmo PRTTa) desempate pelo menor tempo de processamento. Caso a tarefa programada para iniciar seu processamento na máquina não esteja disponível no instante T , o tempo ocioso será preenchido com tarefas que possam ser processadas naquele instante antes do início da tarefa j sem que altere seu início de processamento.

A Figura 4 faz a representação da programação das tarefas por meio da regra PRTTa, onde existem quatro tarefas, dois estágios: o estágio 1 possui duas máquinas; o segundo, apenas uma.

Figura 4: Tarefas ordenadas e processadas com a regra PRTTa



No estágio $t = 1$ o instante inicial será $T = 0$. A Tabela 2 mostra os valores necessários para o cálculo da regra PRTTa para este instante, sendo $\alpha = 1$.

Tabela 2: Dado para o cálculo da regra PRTTa para $t = 1$ e $T = 0$

Tarefa j	r_{j1}	p_{j1}	d_{j1}	$\max(r_{j1}, T)$	$\max(r_{j1}, T) + p_{j1} + S_{j1}$	PRTTa($j, 1, T$)
1	0	5	2	0	5	5
2	0	4	8	0	4	8
3	0	8	-2	0	8	8
4	0	6	6	0	6	6

Observe que, $r_{j1} = 0$ para todas as tarefas pois estão disponíveis, sendo que ainda não houve processamento. A última coluna apresenta o valor calculado para a regra PRTTa($j, 1, T$) por meio da equação (15), por exemplo: iniciamos com a tarefa 1 que têm $r_{j1} = 0$, $p_{j1} = 5$; PRTTa (1,1,0) = $1 * \max(0, 0) + \max(\max(0, 0) + 5, 2) = 5$. Com isto, são escolhidas as duas tarefas com menor valor de PRTTa ($j = 1$ e $j = 2$) para serem programadas nas duas primeiras máquinas do estágio $t = 1$. Seguindo o mesmo procedimento, o algoritmo segue para o próximo instante de disponibilização de máquinas no estágio ($T = 4$) até que todas as tarefas tenham sido programadas no estágio considerado. A programação no segundo estágio começa no primeiro instante de disponibilização de uma tarefa no estágio, ou seja, $T = 4$;

Tabela 3: Dado para o cálculo da regra PRTTa para $t = 1$ e $T = 4$

Tarefa j	r_{j1}	p_{j1}	d_{j1}	$\max(r_{j1}, T)$	$\max(r_{j1}, T) + p_{j1} + S_{j1}$	PRTTa($j, 1, T$)
1	5	6	4	5	11	16
2	4	5	11	4	11	16
3	14	6	2	14	20	34
4	13	5	12	13	18	31

Observa-se que, no instante de conclusão das tarefas, apenas a tarefa 2 está disponível, mas o método programa a tarefa 1 no instante de sua disponibilidade, que é menor que o das outras tarefas seguindo a prioridade colocada pela regra. Com isso, foi inserido um tempo ocioso

de 4 até 5, portanto deve-se procurar por tarefas que possam ser programadas antes da tarefa 1. Como a única tarefa que pode ser processada antes da 1 é a tarefa 2, o processamento segue para o instante de disponibilização de máquina ($T=9$), fazendo com que as demais tarefas sigam o mesmo procedimento.

A próxima seção apresenta os testes preliminares e os resultados computacionais.

EXPERIMENTOS COMPUTACIONAIS

Os algoritmos foram implementados em linguagem de programação C++. Utilizou-se o compilador MinGW e os experimentos foram realizados em um computador Intel Core i5- 3337U CPU @ 1.80 GHz, 6GB de memória RAM, sistema operacional de 64 bits, processador com base em x64 e sistema operacional Windows 10 Home Single Language.

As soluções dos problemas de pequeno porte são obtidas através da resolução do modelo matemático de PLI. Para isto, é utilizado o software de otimização da IBM, ILOG CPLEX Optimization Studio versão 12.5.

Ambiente de testes e instâncias utilizadas

Foram geradas instâncias com as seguintes dimensões expressas pela relação $n \times s \times m$: $5 \times 5 \times 3$, $10 \times 5 \times 3$, $15 \times 5 \times 5$, $20 \times 10 \times 5$, $30 \times 10 \times 5$ e $50 \times 5 \times 5$, sendo que esta geração foi realizada conforme a necessidade de produção de um cenário moveleiro encontrado na cidade de Ub em ambientes reais.

Para cada uma dessas cinco combinações foi gerado um grupo com 100 instâncias do problema, totalizando 600 instâncias. Nestas instâncias o número de máquinas por estágio foi gerado de acordo com uma distribuição (entre 1 e 5). Os tempos de processamento foram gerados através de um intervalo aleatório entre (12 e 20) para o estágio 1, entre (5 e 16) para o estágio 2, entre (2 e 4) para o estágio 3, entre (130 e 151) para o estágio 4 e entre (35 e 52) para o estágio 5. Os tempos de setup foram gerados entre (3 e 12)

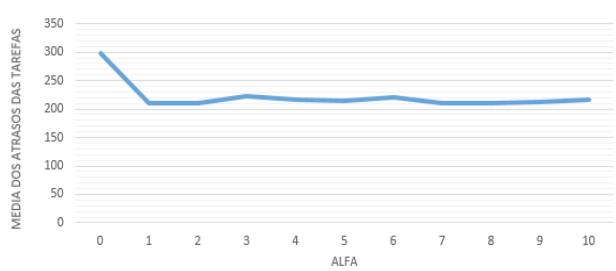
para o estágio 1, entre (5 e 15) para o estágio 2, entre (60 e 780) para o estágio 3, entre (5 e 33) para o estágio 4 e entre (10 e 150) para o estágio 5. As datas de entrega foram geradas através de um intervalo aleatório entre (100 e 500) para instâncias $5 \times 5 \times 3$, (100 e 800) para instâncias $10 \times 5 \times 3$, (100 e 1200) para instâncias $15 \times 5 \times 5$, (100 e 2000) para instâncias $20 \times 5 \times 5$, (100 e 3000) para instâncias $30 \times 5 \times 5$ e (100 e 4500) para instâncias $50 \times 5 \times 5$.

Seleção do parâmetro α

Para selecionar um valor para o parâmetro α , que pondera a importância do primeiro teste da regra PRTTa, foram realizados testes aleatórios para α entre (0 e 10), além desse intervalo os resultados são semelhantes. Estes testes fazem a variação de α e registram o resultado da média do atraso das instâncias para cada valor de α . Com base nos resultados, $\alpha = 1$ foi selecionado por apresentar melhor desempenho na regra.

Pela análise do Gráfico 1, pode-se perceber que a regra apresenta melhor desempenho para $\alpha = 1$, mas é importante observar que resultados similares são obtidos para $\alpha \geq 4$. Os resultados foram obtidos através das medias das instâncias para todas as regras.

Gráfico 1: Comportamento do atraso total em função de α



Resultados computacionais

Nesta seção, considera-se as três regras de prioridades para o problema (EDD, SPT e PRTTa) e também os resultados obtidos pelo Cplex. Todos os métodos são comparados utilizando o desvio relativo percentual (RPD), esta medida

de desempenho é calculada conforme Vallada e Ruiz, (2011) na equação (17), através da seguinte fórmula:

$$RPD\% = \frac{f - f_{best}}{f_{best}} \times 100 \quad (17)$$

onde f corresponde ao valor da função objetivo obtido pela heurística e f_{best} corresponde à melhor solução encontrada a partir da execução da heurística para todos os valores do parâmetro. Os resultados das regras de prioridade são apresentados na Tabela 4, os valores de cada linha equivalem a 100 instâncias e estão estratificados pelas dimensões $s \times n \times m$. Para cada regra é apresentado à média dos atrasos totais no respectivo subconjunto das 100 instâncias. Dando destaque à regra PRTTa que tem sua média abaixo das outras regras.

Verifica-se que o software Cplex se torna inviável para problemas acima de 20 tarefas, por exceder o tempo de 45 minutos de processamento para encontrar o melhor resultado Ronconi (2010).

Tabela 4: Média dos tempos de atraso das instâncias em segundos

Estágio	Maq.	Tarefas	Cplex	EDD	SPT	PRTTa
			Médias			
			Atraso	Atraso	Atraso	Atraso
5	3	5	0,75	9,98	9,88	13,85
5	3	10	53,15	117,5	115,54	99,68
5	5	15	168,24	384,75	463,07	338,94
5	5	20	307	1060,48	1113,51	854,77
5	5	30	30980	3003,63	3100,65	2950,56
5	5	50	-	17259,98	15659,29	12440,37

Pela análise dos resultados apresentados na Tabela 4, pode- se perceber que a regra PRTTa apresenta uma média superior às demais regras, mas os resultados mostraram que 67 das instâncias obteve melhor resultado coma a regra PRTTa, 20 instâncias para a regra EDD e 13 para SPT; com base nisso é possível analisar que a regra PRTTa obteve ótimo desempenho, superior às demais regras apresentadas.

Tabela 5: Média do tempo de CPU das instâncias em segundos

Estágio	Maq.	Tarefas	Cplex	EDD	SPT	PRTTa
			Médias			
5	3	5	2,02	0,91	0,98	0,99
5	3	10	1,77	0,97	1,25	1,92
5	5	15	22,97	1,97	2,24	2,77
5	5	20	144000	2,94	3,85	4,66
5	5	30	144000	3,11	4,06	5,56
5	5	50	-	4,66	4,98	5,67

As tabelas de 6 a 11 mostram média, desvio padrão, mediana e máximo das instâncias testes, levando em consideração resultados obtidos através da aplicação da equação (17).

Tabela 6: Estatísticas para instâncias 5X5X3

Variável	Média	DesvPad	Mediana	Máximo
EDD	797	2250	5	11800
SPT	900	1273	39	5500
PRTTa	1106	2517	9	15100
Cplex	30,0	156,0	1	1100

Tabela 7: Estatísticas para instâncias 10X5X3

Variável	Média	DesvPad	Mediana	Máximo
EDD	3864	7562	105	32700
SPT	3909	7554	105	32700
PRTTa	2604	5046	108	22200
Cplex	854	4108	102	21300

Tabela 8: Estatísticas para instâncias 15X5X5

Variável	Média	DesvPad	Mediana	Máximo
EDD	3510	10321	193	67200
SPT	5154	18077	177	97600
PRTTa	3469	11705	160	72800
Cplex	991	6757	134	52900

Tabela 9: Estatísticas para instâncias 20X5X5

Variável	Média	DesvPad	Mediana	Máximo
EDD	48,92	82,35	27,70	606,30
SPT	57,15	84,15	31,65	624,20
PRTTa	7,97	18,72	23,0	109,20
Cplex	305,3	299,5	206,2	2066,3

Tabela 10: Estatísticas para instâncias 30X5X5

Variável	Média	DesvPad	Mediana	Máximo
EDD	815	9666	37,00	371,60
SPT	1177	11043	14	110500
PRTTa	4066	8969	90	397000
Cplex	-	-	-	-

Tabela 11: Estatísticas para instâncias 50X5X5

Variável	Média	DesvPad	Mediana	Máximo
EDD	49,0	116,2	24,5	1138,8
SPT	104,7	657,1	26,3	6600,0
PRTTa	11615	115515	0	1155200
Cplex	-	-	-	-

Observa-se nas duas tabelas que o CPLEX obteve as melhores soluções para os grupos de problemas entre as heurísticas. Para os grupos de instâncias com 20 tarefas, os tempos de CPU do CPLEX aumentaram muito em relação aos das heurísticas. Nesses grupos, as heurísticas apresentam melhores resultados das médias para os problemas de pequeno porte. Para problema acima, de vinte tarefas, o software Cplex torna-se inviável, pois excede o tempo limite de processamento de quarenta e cinco minutos; para problemas abaixo desse número de tarefas, o software chega a um valor considerado melhor que das heurísticas

propostas. Tomando em consideração empresas de produção da cidade de Ubá problemas que excedem cinquenta tarefas ficaria difícil encontrar a melhor programação dessas tarefas.

É possível notar que, ao utilizar o conhecimento a respeito do problema através da equação (18) que os resultados da regra PRTTa fossem, no geral, melhores que as outras heurísticas. Em todos os testes no software Cplex encontrou-se ótima solução para todos os problemas com até 15 tarefas; já nos problemas com 20 ou mais tarefas, o software não se torna eficiente. O tempo computacional médio dos métodos propostos para cada problema foi inferior a 3,81 segundos enquanto o pacote de otimização apresentou um tempo médio de 2700 segundos para os problemas com até 20 tarefas, limite onde o software consegue chegar para gerar uma solução viável para o problema.

CONSIDERAÇÕES FINAIS

Neste trabalho foi analisada e comparada a eficiência das regras de prioridades EDD, SPT e PRTTa aplicadas em um ambiente de flowshop híbrido com máquinas paralelas não relacionadas; a regra PRTTa foi adaptada da regra PRTT para um ambiente flowshop flexível com máquina única e neste trabalho foi submetida ao ambiente flowshop híbrido com máquinas paralelas não relacionadas, ambos com o mesmo objetivo de minimização do atraso total.

Um modelo matemático de OPL foi formulado para resolver as instâncias dos problemas de pequeno porte. Nas instâncias de pequeno porte (com 5, 10 e 15 tarefas), o software Cplex obteve melhor resultado que as instâncias com vinte tarefas. As comparações entre as heurísticas e através do software Cplex, mostraram que a regra de prioridade PRTTa é eficiente para a resolução do problema, e se mostra superior as outras regras de prioridades analisadas, mostrando que para maioria das instâncias testadas obteve melhor resultado. O desempenho da regra foi favorável ao objetivo de

minimizar o somatório dos tempos de atraso da produção, tendo valores de média menores em relação às outras regras avaliadas, somente nas instâncias pequenas a média foi maior, mais isso se deve a algumas instâncias que apresentaram uma média alta, mas seus resultados foram melhores para 67% dos testes nas instâncias de pequeno porte.

Para o desenvolvimento de futuros trabalhos, sugere-se o estudo das características do ambiente com tempos de setup dependentes da tarefa e da máquina, e a introdução de novas restrições no sistema, tornando-o ainda mais realista. Também poderão ser comparadas com outras regras de prioridade não citadas, além de testar novos ambientes de máquinas com o job shop por se caracterizar por saltar estágios.

REFERÊNCIAS

- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. Pesquisa operacional: para cursos de engenharia. 5^a tiragem. Rio de Janeiro: Elsevier, 2007.
- ARMENTANO, V. A.; RONCONI, D. P. Tabu search for total tardiness minimization in flowshop scheduling problems. *Computers & Operations Research*, v. 26, p. 219-235, 1999.
- BAKER, Kenneth R. Sequencing rules and due-date assignments in a job shop. *Management science*, v. 30, n. 9, p. 1093-1104, 1984.
- BRAH, S. A. A comparative analysis of due date-based job sequencing rules in a flow shop with multiple processors. *Production Planning and Control*, v. 7, n. 4, p. 362-373, 1996.
- CHEN, B.; POTTS, Chris N.; WOEGINGER, Gerhard J. A review of machine scheduling: complexity, algorithms and approximability. In: *Handbook of combinatorial optimization*. Springer US, p. 1493-1641, 1998.
- CHU, C.; PORTMANN, M. C. Some new efficient methods to solve the $n/1/r_i/\sum T_i$ scheduling problem. *European Journal of Operational Research*, v. 58, p. 404-413, 1992.
- ETCHEVERRY, Guilherme; ANZANELLO, Michel J. Sequenciamento de máquinas paralelas não-relacionadas com tempos de setup dependentes das tarefas. *Revista Produção Online*, 2014, 14, n. 3, p. 890-913, 2014.
- FUCHIGAMI, Helio Yochihiro; MOCELLIN, João Vitor. Desempenho relativo de regras de prioridade para programação de flow shop híbrido com tempos de setup.
- Revista Produção Online, v. 15, n. 4, p. 1174, 2015.
- IBM. IBM ILOG CPLEX CP Optimizer. Disponível em: <<http://www-03.ibm.com/software/products/pt/ibmilogcplecpopti>>. Acesso em: 04 jul. 2017.
- JUNGWATTANAKIT, J.; REODECHA, M.; CHAOVALITWONGSE, P.; WERNER, F. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *The International Journal of Advanced Manufacturing Technology*, v. 37, n. 3, p. 354-370, 2008.
- LUDWIG, I.; ANZANELLO, M. J.; VIDOR, G. Minimização dos tempos de atraso na programação de tarefas em uma empresa de desenvolvimento de softwares. *Revista Produção Online*, v. 13, p. 479, 2013.
- MAINIERI, G. B; RONCONI, D. P. Regras de despacho para a minimização do atraso total no ambiente flowshop flexível, São Carlos, v. 17, n. 4, p. 683-692, 2010.
- MELLO, A. R. A. F. Programação de tarefas para uma empresa de manufatura em um ambiente flow-shop flexível. Escola Politécnica da Universidade de São Paulo - SP, 2004.
- MORAIS, M. F.; MENEGARDE, J. K.; CANTIERE, P. C. Regras de prioridade e critérios de desempenho adotados em problemas de programação da produção em ambientes flow shop. IV Encontro de Produção Científica e Tecnológica, p. 1-9, 2009.
- NOGUEIRA, João Paulo de C. M. et al. Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electronic Notes in Theoretical Computer Science*, [s.l.], v. 302, p.53-72, fev. 2014.
- OLIVEIRA, G. S; TURRIONI, J. B. O problema de flow shop híbrido: revisão e classificação. Encontro Nacional de Engenharia de Produção (ENECEP); Encontro Nacional de Engenharia de Produção (ENECEP); XXXII; Bento Gonçalves, 2012.
- PINEDO, M. L. *Scheduling: theory, algorithms, and systems*. Boston, MA: Springer US, 2012.
- PINEDO, M. L. *Scheduling: theory, algorithms, and systems*. New York: Springer, 2008.
- ROCHA, P. L. Um problema de sequenciamento em máquinas paralelas não relacionadas com tempos de preparação dependentes de máquina e da sequência: modelos e algoritmo exato. 2006. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Minas Gerais, Belo Horizonte, 2006.
- RONCONI, D. Minimização do tempo total de atraso no problema de flowshop com buffer zero através de busca tabu. *Computer & Operations Research*, n. 26, p. 219-235, 1999.
- RONCONI, Débora P. A note on constructive heuristics for the flowshop problem with blocking. *International Journal of Production Economics*, v. 87, n. 1, p. 39-48, 2004.
- SHINGO, S. Sistema de troca rápida de ferramenta: uma

revolução nos sistemas produtivos. Porto Alegre: Bookman, 2008.

VALLADA, Eva; RUIZ, Rubén. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. European Journal of Operational Research, v. 211, n. 3, p. 612-622, 2011.