

## SEQUENCIAMENTO DE MÁQUINAS PARALELAS NÃO-RELACIONADAS COM TEMPOS DE *SETUP* DEPENDENTES DAS TAREFAS

### SCHEDULING ON UNRELATED PARALLEL MACHINES WITH SEQUENCE DEPENDENT SETUP TIMES

Guilherme V. Etcheverry\* E-mail: [guilherme.etccheverry@epcos.com](mailto:guilherme.etccheverry@epcos.com)

Michel J. Anzanello\*\* E-mail: [anzanello@producao.ufrgs.br](mailto:anzanello@producao.ufrgs.br)

\*EPCOS do Brasil Ltda, Gravataí, RS

\*\* Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS

**Resumo:** Este artigo propõe 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. As funções-objetivo consideradas são a minimização da soma do tempo ponderado de processamento e a minimização do atraso ponderado das tarefas. O método proposto acrescenta uma etapa ao procedimento de duas etapas de Bank e Werner (2001), para solução de problemas NP-hard, e é executado em três etapas: (i) ordenação inicial das tarefas, (ii) alocação das tarefas às máquinas, e (iii) aprimoramento da solução final através da ferramenta de otimização *Tabu Search*. O desempenho da heurística é comparado à solução ótima obtida por enumeração total. Ao ser aplicada em um ambiente de manufatura de componentes eletrônicos composto por duas máquinas paralelas não-relacionadas com tempos de processamento e de *setup*, a heurística proposta resulta em um desvio de 1,1% em relação ao valor ótimo da soma do tempo ponderado de processamento e 4,6% do mínimo atraso ponderado das tarefas em tempo computacional inferior a dez minutos.

**Palavras-chaves:** Sequenciamento. Máquinas paralelas não relacionadas. *Tabu search*.

**Abstract:** This article proposes a scheduling heuristic for unrelated parallel machines with sequence dependent setup times. The minimization of total weighted completion time and the minimization of weighted tardiness are the objective functions of interest. A three phase heuristic, based on Bank and Werner's (2001) NP-hard problems solving procedure, is proposed to (i) perform an initial ordering of the jobs, (ii) allocate the jobs to the machines, and (iii) improve final solution of both objectives with *Tabu Search*. The heuristic performance is evaluated comparing the quality of proposed solution with the optimal solution obtained from total enumeration. The proposed model is applied in an industry of electronic components industry composed of two unrelated parallel machines with deterministic processing and setup times. We obtained a 1.1% deviation for the optimal total weighted completion time and 4.6% of deviation for the weighted tardiness in less than ten minutes of computational time.

**Keywords:** Job scheduling. Unrelated parallel machines. *Tabu search*.

## 1 INTRODUÇÃO

O sequenciamento é um processo gerencial de fundamental importância na maximização do desempenho operacional e financeiro das empresas. Pela sua

grande aplicabilidade, o eficiente sequenciamento se constitui em importante diferencial competitivo, razão pela qual tem sido amplamente pesquisado em diversos ambientes de manufatura.

Utilizado regularmente nas empresas manufatureiras e de serviços, o sequenciamento se constitui no processo de ordenamento das tarefas e escolha do recurso mais adequado para sua execução, considerando restrições de tempo e de relacionamento entre tarefas (PINEDO, 2008). O sequenciamento usualmente visa minimizar o tempo de conclusão de um conjunto de tarefas (*completion time*), o atraso (*tardiness*), a conclusão adiantada das tarefas (*earliness*), o número total de tarefas atrasadas (*tardy jobs*) ou uma combinação destes objetivos. Tais objetivos podem ser integrados a um fator de priorização (*weight*), que assume a forma de um fator de importância ou um fator representativo de custo da tarefa (MORTON & PENTICO, 1993; PINEDO, 2008).

Os ambientes de manufatura podem apresentar configurações diversas, de acordo com o nível de customização do produto em questão. Ambientes constituídos por máquinas não-relacionadas em paralelo (ou seja, cada máquina possui uma velocidade de processamento não associada a outras máquinas) são especialmente atrativos, tanto do ponto de vista teórico como prático. No aspecto teórico, ambientes de manufatura compostos por máquinas em paralelo não-relacionadas são tidos como uma das classes de mais difícil solução dentre os problemas de sequenciamento. No contexto prático, tais organizações representam muitos ambientes de manufatura reais. Arranjos de máquinas em paralelo são gerados pela necessidade de renovação do maquinário existente, ou pelo aumento da capacidade instalada, visto que máquinas de gerações tecnológicas distintas possuem velocidades de operação e tempos de *setup* distintos (MOKOTOFF, 2001; LIAW et al., 2003; PINEDO, 2008; LIN et al., 2011).

Tal relevância tem justificado o elevado número de abordagens focadas no arranjo em questão, principalmente com vistas à minimização do tempo total para conclusão da última tarefa (*makespan*) (MOKOTOFF, 2001; PFUND et al., 2004). Entretanto, a inserção do tempo de *setup* dependente da sequência ainda se constitui em tópico pouco explorado na literatura (PFUND et al., 2004). Tal lacuna apresenta consequências práticas bastante relevantes, como no simples exemplo

que segue. Considere duas tarefas a serem sequenciadas, A e B; se a sequência de produção adotada for  $A \rightarrow B$ , gera-se um *setup* (preparação e ajuste de máquina) igual a 10 minutos; caso opte-se por processar  $B \rightarrow A$ , obtém-se um tempo de preparação igual a 20 minutos. De tal forma, o tempo de *setup* constitui-se em uma variável tão ou mais importante que o próprio tempo de processamento das tarefas em procedimentos de sequenciamento.

Este artigo apresenta uma nova heurística para minimizar duas funções-objetivo de grande interesse prático: (i) o atraso ponderado das tarefas, e (ii) o tempo total ponderado de conclusão de um conjunto de tarefas, ambas com tempos de *setup* dependentes da sequência das tarefas. A heurística acrescenta uma terceira etapa ao procedimento de duas etapas de Bank e Werner (2001) para a solução de problemas de elevada complexidade computacional.

A sistemática proposta é composta por três etapas: (i) ordenação prévia das tarefas de acordo com uma regra alinhada à função-objetivo em análise, (ii) alocação das tarefas às máquinas de maneira a assegurar balanço entre os tempos de processamento alocados e sequenciamento preliminar, e (iii) refino do sequenciamento das tarefas alocadas a cada máquina. O método proposto é validado em um processo de manufatura de componentes eletrônicos composto por duas máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência das tarefas. O desempenho da heurística proposta é comparado ao resultado ótimo gerado através de enumeração total.

O artigo está estruturado como segue: a seção 2 apresenta o referencial teórico sobre sequenciamento em ambiente de manufatura de máquinas paralelas não-relacionadas. As seções 3 e 4 apresentam, respectivamente, o método proposto e discussão dos resultados. Por fim, a seção 5 apresenta a conclusão da pesquisa e direções futuras.

## **2 REFERENCIAL TEÓRICO**

O sequenciamento em máquinas paralelas consiste na alocação de  $n$  tarefas a  $m$  máquinas candidatas, com o objetivo de otimizar uma função-objetivo associada a uma medida de desempenho (MOKOTOFF, 2001). Ambientes de manufatura

constituídos por máquinas em paralelo podem ser categorizados em três classes: (i) máquinas paralelas idênticas (*identical parallel machines*) – tempo de processamento das tarefas é o mesmo em qualquer uma das máquinas, (ii) máquinas paralelas uniformes (*uniform parallel machines*) – tempo de processamento da tarefa é dependente da velocidade da máquina que irá processar a tarefa, e (iii) máquinas paralelas não-relacionadas (*unrelated parallel machines*) – tempo de processamento da tarefa é dependente, simultaneamente, da velocidade da máquina e da tarefa (MOKOTOFF, 2001; PINEDO, 2008; FANJUL-PEYRO & RUIZ, 2011; LIN et al., 2011).

O problema de sequenciamento em ambiente de máquinas paralelas não-relacionadas consiste, inicialmente, na escolha da máquina mais apropriada para cada tarefa, e então na determinação da melhor sequência de processamento das tarefas alocadas a cada máquina (LÓPEZ et al., 1995; MOKOTOFF, 2001; PFUND et al., 2004; PINEDO, 2008; LIN et al., 2011). Esse arranjo de máquinas é detalhado nas seções que seguem.

## 2.1 Máquinas paralelas não-relacionadas

Duas funções-objetivo associadas a máquinas paralelas não-relacionadas têm recebido crescente atenção: (i) minimização do tempo total ponderado de conclusão, e (ii) minimização do atraso ponderado de entrega. Tais problemas são classificados como *NP-hard*, ou seja, requerem um tempo de processamento não polinomial (exponencial) para obtenção da solução ótima à medida que o número de máquinas ou tarefas é aumentado. Por conta disto, diversos algoritmos e heurísticas têm sido propostas para gerar soluções satisfatórias em tempos reduzidos de processamento computacional (LÓPEZ et al., 1995; PFUND et al., 2004; PINEDO, 2008).

Os problemas de sequenciamento são usualmente representados por uma notação de três campos, proposta por Graham et al. (1979), de tal forma que as funções-objetivo descritas no parágrafo anterior são representadas, respectivamente, pelas equações (1) e (2).

$$R_m / S_{jk} / \sum w_j C_j \quad (1)$$

$$Rm / S_{jk} / \sum w_j T_j \quad (2)$$

Nesta representação, o primeiro campo identifica o ambiente das máquinas, assumindo a representação  $R$  para máquinas paralelas não-relacionadas e o índice  $m$  representa o número de máquinas em paralelo. O segundo campo identifica as características das tarefas e as restrições do sistema: *setup* dependente da sequência ( $S_{jk}$ ) (*sequence dependent*) – tempo de *setup* é variável em função da tarefa atual e da tarefa subsequente, ou independente da sequência (*sequence independent*) - tempo de *setup* é fixo, independentemente da sequência de processamento das tarefas (ALLAHVERDI et al., 2008; PINEDO, 2008).

Por fim, o terceiro campo especifica a função-objetivo considerada:  $\sum w_j C_j$  – minimização da soma do tempo ponderado para conclusão de um conjunto de tarefas e  $\sum w_j T_j$  – minimização da soma ponderada do atraso. A presença de um fator de priorização (*weight*) é identificada através da letra  $w$  antes do critério de otimização (GRAHAM et al., 1979). Na sequência são descritas abordagens recentes acerca das funções-objetivo acima descritas, em ambientes de máquinas paralelas não-relacionadas.

### 2.1.1 Minimização do tempo ponderado de conclusão

Heurísticas com vistas à minimização do tempo total de conclusão da última tarefa (*makespan*) envolvendo máquinas paralelas não-relacionadas têm sido amplamente sugeridas na literatura. Grande parte das abordagens, no entanto, não considera tempos de *setup* dependentes da sequência de tarefas.

Dentre os autores que não assumem a presença de tempos de *setup*,  $Rm/\sum w_j C_j$ , Fanjul-Peyro & Ruiz (2011; 2010) utilizaram metaheurísticas de busca iterativa local e redução de tamanho do problema original para conjuntos de tarefas com tempos de processamento distribuídos uniformemente. As heurísticas propostas alcançaram notável desempenho frente a outras proposições, gerando resultados próximos à solução ótima. Lin et al. (2011) propuseram uma sistemática de dois estágios baseada em relaxação de programação linear e procedimentos de redistribuição e intercâmbio. Tais procedimentos geram resultados ótimos para pequeno número de tarefas e aproximações satisfatórias para conjuntos com

elevado número de tarefas.

Para a mesma função-objetivo, Gairing et al. (2005) propuseram uma abordagem combinatória de enumeração parcial baseada em programação inteira com *Branch-and-bound* (B&B) e planos de corte, formulando o problema como um caso geral de fluxo máximo indivisível. Grigoriev et al. (2005) desenvolveram um algoritmo em duas etapas, realizando uma relaxação da programação linear inteira seguida de um procedimento de arredondamento para provar a proximidade do resultado gerado com a solução ótima. Por sua vez, Srivastava (1998) utilizou heurísticas baseadas em *Tabu Search* (TS) para determinar o desvio percentual de ambas em relação à solução ótima.

Valendo-se de ferramentas de otimização, Wu e Ji (2009) propuseram um algoritmo genético para minimizar o tempo total decorrido para completar a última tarefa em uma linha de montagem de placas de circuitos impressos. A utilização de programação quadrática integrada ao algoritmo B&B foi aplicada por Plateau e Rios-Solis (2010) para resolver o problema  $Rm // \sum w_j C_j$ . Os autores alegam que os resultados encontrados se constituem no estado-da-arte para o problema em questão. Focados na mesma função-objetivo, Chen e Powell (1999) propuseram um método utilizando a decomposição de Dantzig-Wolfe, partindo da partição da formulação inicial e resolvendo os nós do algoritmo B&B como um subproblema de máquina simples através de uma sistemática de geração de colunas (*column generation* - CG).

Li e Yang (2009) apresentam uma classificação para modelos baseados em variáveis de atribuição e tempo de liberação das tarefas, relaxação linear e de programação quadrática convexa, algoritmos exatos e heurísticas para os problemas de máquinas paralelas com vistas à minimização do tempo total de conclusão das tarefas, com e sem fator de priorização ( $Rm // \sum w_j C_j$  e  $Rm // \sum C_j$ ). Para o problema  $Rm // \sum w_j C_j$ , Shim (2009) abordou um esquema de ramificação baseado no algoritmo B&B com o objetivo de reduzir o tempo computacional para obtenção da solução ótima. Por fim, Correa et al. (2009) também estudaram a função-objetivo de minimização do tempo total decorrido para completar a última tarefa, sem considerar os tempos de *setup* e permitindo interrupções no processamento das mesmas (*preemption*).

Ao incluir-se o tempo de *setup* em arranjos de máquinas paralelas não-relacionadas com vistas à minimização do tempo de conclusão,  $R_m / S_{jk} / \sum C_{max}$ , destaca-se o estudo de Weng et al (2001), o qual compara sete heurísticas para resolução do problema, onde a melhor heurística utiliza uma lógica de alocação de tarefas com base na menor relação entre o tempo de processamento somado ao de *setup* e ao seu fator de priorização. Com propósitos similares, Hsu et al. (2010) propuseram uma solução em tempo polinomial considerando o tempo de *setup* proporcional ao tempo de processamento total até a tarefa imediatamente anterior.

Buscando resolver o mesmo problema, um algoritmo genético (AG) utilizando intercâmbio adjacente e não adjacente de tarefas foi desenvolvido por Chang e Chen (2011) para aumento da velocidade de convergência da solução, enquanto que Ying et al. (2010) utilizaram um procedimento de *Simulated Annealing* (SA) restrito para eliminar movimentos de busca ineficientes. Rabadi et al. (2006) compararam uma metaheurística de busca prioritária aleatória (MHBPA) com heurística de particionamento (HP). Os três trabalhos obtiveram soluções ótimas para conjuntos compostos por poucas tarefas e superaram os resultados gerados por AG, SA e HP para conjuntos com maior número de tarefas.

Por sua vez, Arnaout et al. (2008; 2010) utilizaram um algoritmo baseado em *Ant Colony Optimisation* (ACO) de dois estágios em comparação com MHBPA, HP e TS para o problema em questão, concluindo que ACO apresentou desempenho superior às demais heurísticas. Por fim, Tavakkoli-Moghaddam et al. (2009) consideraram o problema  $R_m / S_{jk} / \sum C_j$  com datas distintas de liberação e de conclusão entre as tarefas, obtendo resultados satisfatórios através da otimização por AG.

### **2.1.2 Minimização do atraso ponderado das tarefas**

Contrariamente ao vasto número de estudos que tratam do problema  $R_m // \sum w_j C_j$ , percebe-se uma quantidade reduzida de abordagens com vistas à minimização do atraso ponderado das tarefas em máquinas paralelas não-relacionadas ( $R_m // \sum w_j T_j$ ). Lin et al. (2011) e Liaw et al. (2003) abordam o problema propondo, respectivamente, heurísticas de dois estágios baseadas em *apparent tardiness cost*

(ATC) e B&B. A minimização do atraso total das tarefas com tempos de *setup* dependentes das tarefas sem fator de priorização ( $R_m / S_{jk} / \sum T_j$ ) foi estudada por Chen e Chen (2009), que utilizaram uma heurística baseada no *apparent tardiness cost with setup* (ATCS) e SA. A mesma função-objetivo foi analisada por Kim et al. (2002) através de uma metaheurística baseada em AS, e por Silva e Magalhães (2006), que adaptaram heurísticas existentes para o problema de uma empresa produtora de fibras acrílicas.

A minimização do atraso total das tarefas com tempos de *setup* dependentes das tarefas com fator de priorização,  $R_m / S_{jk} / \sum w_j T_j$ , foi abordada por Paula et al. (2010) através de um algoritmo *relax-and-cut* baseado em relaxação Lagrangeana, e por Logendran et al. (2007), que propuseram seis algoritmos baseados em TS. Por sua vez, Kim et al. (2003) acrescentam datas de entrega ao problema em questão, comparando o desempenho de heurísticas baseadas em regras de ordenação e SA.

### 3 MÉTODO

Considere o problema de sequenciamento de  $n$  tarefas em  $m$  máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência de processamento das tarefas. Todas as tarefas possuem datas de liberação e conclusão comuns, e seu processamento, uma vez iniciado, não pode ser interrompido.

Os objetivos considerados consistem em (i) minimizar o tempo total ponderado de conclusão ( $R_m / S_{jk} / \sum w_j C_j$ ) e (ii) minimizar o atraso ponderado de um conjunto de tarefas ( $R_m / S_{jk} / \sum w_j T_j$ ). Em função da complexidade de tais arranjos, Bank e Werner (2001) sugerem um procedimento de duas etapas para a solução do problema: na primeira etapa, as tarefas devem ser ordenadas por uma regra de ordenação para, na segunda etapa, serem alocadas às máquinas.

O método proposto neste artigo adiciona uma terceira etapa ao procedimento de Bank e Werner (2001). As três etapas do método são: (1) ordenação inicial das tarefas, (2) alocação das tarefas às máquinas, e (3) refino da ordenação das tarefas alocadas às máquinas através de uma versão simplificada do algoritmo *Tabu Search*. As duas primeiras etapas são comuns para ambos os objetivos, contudo, na

terceira etapa funções-objetivo distintas promovem ordenamentos específicos às tarefas. Essas etapas são detalhadas na sequência.

### **Etapa 1 – Ordenação inicial das tarefas**

A primeira etapa da heurística realiza a ordenação inicial das tarefas para posterior alocação às máquinas, conforme apresentado no Quadro 1. As tarefas são ordenadas pela regra WSPT (*Weighted Shortest Processing Time*), a qual ordena as tarefas de forma decrescente de  $w_n / p_n$ , onde  $w_n$ , o fator de priorização da tarefa  $n$ , representa a importância de uma tarefa dentro do conjunto de tarefas e  $p_n$  representa o tempo de processo da tarefa  $n$  na máquina  $m$  (PINEDO, 2008). Como o tempo de processamento difere de uma máquina para outra, a ordenação é realizada de forma decrescente de  $w_n / \overline{p}_n$ , fazendo com que sejam priorizadas tarefas com elevado fator de priorização e baixo tempo médio de processamento das tarefas ( $\overline{p}_n$ ).

**Quadro 1** – Passos da etapa 1 da heurística

<b>Passos</b>	<b>Atividade</b>
Passo 1	Calcula $\overline{p}_n$ e $w_n / \overline{p}_n$ de cada tarefa
Passo 2	Aplica WSPT

Fonte: Os autores

### **Etapa 2 – Alocação das tarefas às máquinas**

Nesta etapa as tarefas são alocadas às máquinas com base no tempo total acumulado em cada máquina,  $T_m$ , calculado através da equação (3).  $T_m$  é calculado para cada uma das  $k$  tarefas alocadas à máquina  $m$  através do somatório do tempo de processamento da  $j$ -ésima tarefa ( $p_j$ ) com o tempo de *setup* entre as tarefas  $j-1$  e  $j$ ,  $S_{(j-1)j}$ . Repare que o tempo de *setup* é considerado nulo na alocação da primeira tarefa a cada máquina, e que os tempos de processamento e de *setup* são determinísticos.

$$T_m = \sum_{j=1}^k (p_j + S_{(j-1)j}) \quad (3)$$

As primeiras  $m$  tarefas ordenadas no passo 1 são alocadas às máquinas em ordem crescente do valor de  $T_m$  (ou seja, por se tratar de um problema de *setup* dependente da sequência de tarefas, o tempo de *setup* somente será considerado quando se conhece a primeira tarefa alocada a cada máquina). A próxima tarefa candidata à alocação tem seu tempo de processamento somado ao tempo de *setup*, de acordo com sua tarefa antecessora, e alocada à máquina com o menor  $T_m$ . O ciclo se repete até que todas as tarefas tenham sido alocadas às máquinas. Esses passos são resumidos no Quadro 2.

**Quadro 2** – Passos da etapa 2 da heurística

<b>Passo</b>	<b>Atividade</b>
	$i = 1$ a $n$ (índice de ordenação da tarefa na etapa 1)
Passo 1	Calcule $T_m$
Passo 2	Aloque a tarefa ordenada na etapa 1 à máquina com menor $T_m$ . (Se $T_m = T_{m+1}$ aloque a tarefa à máquina de menor índice).
Passo 3	$i = i + 1$
Passo 4	Se $i \leq n$ , execute o passo 1, senão passe para a Etapa 3 (próxima etapa da heurística).

**Fonte:** Os autores

Um exemplo numérico desta etapa é agora apresentado para quatro tarefas, ordenadas de 1 a 4 pela etapa 1, a serem alocadas em duas máquinas paralelas não-relacionadas com vistas à minimização de  $\sum w_j C_j$ . Os tempos de processamento das tarefas em cada máquina e o tempo de *setup* são mostrados na Tabela 1.

**Tabela 1** – Tempos de processamento e *setup* entre tarefas

<b>i</b>	<b>w<sub>n</sub></b>	<b>p<sub>n1</sub></b>	<b>p<sub>n2</sub></b>	<b>S<sub>jk</sub></b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
1	3	30	20	1	-	0	5	0
2	4	26	30	2	5	-	0	8
3	3	28	32	3	10	4	-	0
4	1	28	36	4	0	3	6	-

Fonte: Os autores

O passo inicial consiste em calcular  $T_1$  e  $T_2$  para as tarefas 1 e 2. A heurística aloca a tarefa 2 à máquina 1 ( $T_1 = 26$ ) e a tarefa 1 à máquina 2 ( $T_2 = 20$ ). Para determinar as máquinas que receberão as tarefas 3 e 4, calculam-se novamente os valores de  $T_1$  e  $T_2$ . Assim, a tarefa 3 é alocada à máquina 1 ( $T_1 = 54$ ) e a tarefa 4 à máquina 2 ( $T_2 = 56$ ).

**Tabela 2** – Cálculos intermediários da etapa 2 da heurística

<b>i</b>	<b>T<sub>1</sub></b>	<b>T<sub>2</sub></b>
3	$26 + (28 + 0) = 54$	$20 + (32 + 5) = 57$
4	<b><math>26 + (28 + 8) = 62</math></b>	<b><math>20 + (36 + 0) = 56</math></b>

Fonte: Os autores

### **Etapa 3 – Ordenação das tarefas alocadas às máquinas via *Tabu Search***

Nesta etapa, as tarefas alocadas a cada máquina têm seu ordenamento refinado através do algoritmo *Tabu Search* (TS) simples de Glover (1989) com vistas à minimização das funções-objetivo  $\sum w_j C_j$  e  $\sum w_j T_j$ . Por fim, a heurística proposta compara a solução refinada com a solução ótima obtida por enumeração total.

O TS é um procedimento para solução de problemas combinatoriais baseado na busca da solução ótima global na vizinhança da solução atual, através do intercâmbio de duas tarefas adjacentes. Partindo de uma solução inicial, o TS realiza o intercâmbio de tarefas adjacentes gerando  $n-1$  vizinhanças adjacentes à solução inicial. Para cada uma das vizinhanças geradas, calcula-se o valor da função-objetivo do problema. A vizinhança com menor (maior) valor, no caso de problemas

de minimização (maximização), é escolhida para prosseguir na busca da solução ótima na próxima iteração do algoritmo (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008).

A cada iteração, o par de tarefas no qual o intercâmbio gerou a vizinhança com menor (maior) valor da função-objetivo é armazenado em uma lista de movimentos proibidos (lista *tabu*) de serem realizados nas iterações seguintes. Evita-se, de tal modo, que a busca retorne a vizinhanças anteriormente visitadas e cujo valor da função-objetivo é superior (inferior) ao atual. A lista *tabu* possui um número fixo de entradas e, a cada novo par de tarefas incluído na lista, o par mais antigo é excluído e seu intercâmbio pode ser realizado novamente nas próximas iterações (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008).

A sistemática de operacionalização do TS é agora ilustrada para a minimização de  $\sum w_j C_j$ . Suponha que as quatro tarefas do exemplo anterior foram alocadas à máquina 1. Partindo de uma solução inicial (3, 4, 2, 1) com função objetivo igual a 840, a primeira iteração do TS realiza os intercâmbios adjacentes das tarefas, gerando as sequências (4, 3, 2, 1), (3, 2, 4, 1), (3, 4, 1, 2). O valor da função objetivo das três sequências é calculado pela equação (4), cujos resultados são, respectivamente, 963, 782 e 846.

$$\sum_{j=1}^n w_j \cdot (p_j + S_{jk}) \quad (4)$$

A sequência (3, 2, 4, 1) substitui a sequência inicial (3, 4, 2, 1) como a solução atual e o par (2, 4), intercambiado na iteração 1, assume a posição superior da lista *tabu*. Na segunda iteração, a sequência (3, 2, 4, 1) tem três novos intercâmbios adjacentes, gerando as sequências (2, 3, 4, 1), (3, 4, 2, 1) e (3, 2, 1, 4). A sequência (3, 4, 2, 1) realiza o intercâmbio proibido entre as tarefas 2 e 4, sendo, portanto, desconsiderada. O valor da função objetivo das sequências (2, 3, 4, 1) e (3, 2, 1, 4) são, respectivamente, 684 e 716. Assim, a sequência (2, 3, 4, 1) assume o lugar da sequência (3, 2, 4, 1) como a solução atual e o par (2,3) ocupa o primeiro lugar da lista *tabu* empurrando o par (2,4), intercambiado na iteração anterior, para a segunda posição da lista.

Por fim, os intercâmbios adjacentes das tarefas da sequência (2, 3, 4, 1) gera

as sequências (3, 2, 4, 1), (2, 4, 3, 1) e (2, 3, 1, 4). A sequência (3, 2, 4, 1) é desconsiderada pois viola a lista tabu. O valor da função-objetivo das sequências (2, 4, 3, 1) e (2, 3, 1, 4) é, respectivamente, 862 e 670, esta última, a sequência (2, 3, 1, 4), se constitui na solução ótima do problema. A Figura 1 mostra a sequência de iterações realizada pelo TS para obter a solução ótima do exemplo.

**Figura 1** – Iterações do TS para obtenção da solução ótima do exemplo

Sequência de tarefas (valor da função objetivo)						
Seq. Inicial	3421 (840)					
Iteração 1	4321 (963)	3412 (846)		3241 (782)		
Iteração 2				2341 (684)	3421 (tabu)	3214 (716)
Iteração 3				3241 (tabu)	2431 (862)	2314 (670)
Iteração 4	3421 (tabu)	4231 (886)	4312 (1032)			

**Fonte:** Os autores

O valor da função objetivo gerada pelo TS é então comparado com a solução ótima obtida através de enumeração total. O Quadro 3 sumariza os passos da terceira etapa da heurística para a minimização do tempo total de execução das tarefas.

**Quadro 3** – Passos da etapa 3 da heurística

Sequência	Atividade
Passo 1	Aplica <i>Tabu Search</i> para obtenção da solução ótima
Passo 2	Compara a solução obtida com a solução ótima obtida por enumeração

**Fonte:** Os autores

Por fim, a minimização da segunda função-objetivo ( $\sum w_j T_j$ ) com uma data de entrega comum para todas as tarefas segue os passos descritos no Quadro 3, valendo-se da equação (5).

$$\sum w_j \cdot T_j = \max(C_j - d, 0) \cdot w_j \quad (5)$$

## 4 RESULTADOS E DISCUSSÃO

A heurística proposta foi aplicada em um ambiente de manufatura composto por 14 tarefas e duas máquinas paralelas não-relacionadas, no processo de metalização de filmes plásticos em alto vácuo. O processo consiste na deposição de uma camada de metal sobre o filme plástico com a finalidade de obter determinadas características elétricas ou de aparência.

Os tempos de processamento e de *setup*, quando existente, são de natureza determinística e correspondem a tempos reais do processo de metalização. A Tabela 3 mostra o tempo de processamento  $p_1$  e  $p_2$  das tarefas nas máquinas 1 e 2.

**Tabela 3** - Tempos de processamento das tarefas

Tarefas	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$p_1$	56	65	89	93	82	85	75	75	88	79	71	62	53	44
$p_2$	52	60	90	93	90	93	82	82	108	97	86	75	64	52

Fonte: Os autores

A primeira etapa da heurística realiza a ordenação das tarefas segundo a regra WSPT. A Tabela 4 mostra o fator de priorização das tarefas ( $w$ ), o tempo médio de processamento ( $\bar{p}_n$ ), oriundo da média aritmética dos tempos de processamento  $p_1$  e  $p_2$ , e, por fim, o resultado dessa etapa, a ordenação realizada pelo WSPT.

**Tabela 4** – Tarefas ordenadas pelo *WSPT*

<b>Tarefas</b>	<b>w</b>	$\bar{p}_n$	$w / \bar{p}_n$
1	212,5	54	3,94
2	244,6	62,5	3,91
4	146,3	93	1,57
3	131,4	89,5	1,47
6	94,4	89	1,06
5	82,1	86	0,95
7	60,8	78,5	0,78
8	52,2	78,5	0,67
14	22,5	48	0,47
9	35,9	98	0,37
10	27,8	88	0,32
13	18,4	58,5	0,31
12	20,5	68,5	0,30
11	22,4	78,5	0,29

Fonte: Os autores

Na sequência, a segunda etapa da heurística aloca as tarefas às máquinas com o menor tempo total acumulado. Inicialmente a tarefa 1 é alocada à máquina 2, que possui menor tempo de processamento do que a máquina 1 (52 x 56). A seguir, a tarefa 2 é alocada à máquina 1, apesar de que seu tempo de processamento é maior na máquina 1 do que na máquina 2, pois a heurística sempre aloca a tarefa à máquina com menor tempo total acumulado.

Assim, temos que a tarefa 2 está alocada à máquina 1, com tempo total de 56 e a tarefa 1 está alocada à máquina 2, com tempo 52. Após a primeira alocação de cada máquina, considera-se o tempo total acumulado somado ao tempo de *setup* (Tabela 5) entre a tarefa anteriormente alocada e a tarefa candidata à alocação, como critério de decisão para alocação da próxima tarefa.

**Tabela 5** - Tempos de *setup* entre tarefas

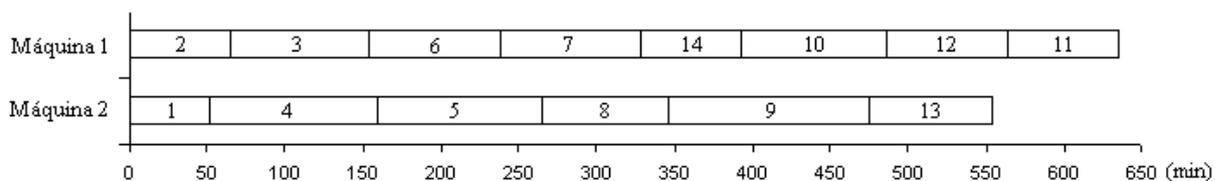
Tarefa	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-	0	0	15	15	0	15	15	20	20	20	20	20	20
2	0	-	0	15	15	0	15	15	20	20	20	20	20	20
3	0	0	-	15	15	0	15	15	20	20	20	20	20	20
4	15	15	15	-	15	15	0	15	20	20	20	20	20	20
5	15	15	15	15	-	15	15	0	20	20	20	20	20	20
6	0	0	0	15	15	-	15	15	20	20	20	20	20	20
7	15	15	15	0	15	15	-	15	20	20	20	20	20	20
8	15	15	15	15	0	15	15	-	20	20	20	20	20	20
9	20	20	20	20	20	20	20	20	-	15	15	15	15	15
10	20	20	20	20	20	20	20	20	15	-	15	15	15	15
11	20	20	20	20	20	20	20	20	15	15	-	0	15	0
12	20	20	20	20	20	20	20	20	15	15	0	-	15	0
13	20	20	20	20	20	20	20	20	15	15	15	15	-	15
14	20	20	20	20	20	20	20	20	15	15	0	0	15	-

Fonte: Os autores

A próxima tarefa candidata a alocação é a tarefa 4. Nesse momento a heurística calcula o tempo total acumulado das duas máquinas, considerando que estas viessem a receber a tarefa 4. O tempo de processamento da tarefa 4 é o mesmo nas duas máquinas, 93; e o tempo de *setup* das tarefas 1 ou 2 para a tarefa 4 é o mesmo, 15. Assim, caso receba a tarefa 4, a máquina 1 ficaria com tempo total de 173 e a máquina 2 com tempo total de 160.

Como resultado a tarefa 4 é alocada à máquina 2. As demais tarefas são alocadas sucessivamente seguindo essa mesma sistemática. A Figura 2 apresenta o resultado da alocação das tarefas às máquinas (o número da tarefa aparece no centro de cada barra do gráfico de Gantt), resultante da etapa 2 da heurística, com  $\sum C_j = 635$  na máquina 1 e  $\sum C_j = 554$  na máquina 2.

**Figura 2** – Sequência de tarefas alocadas às máquinas 1 e 2

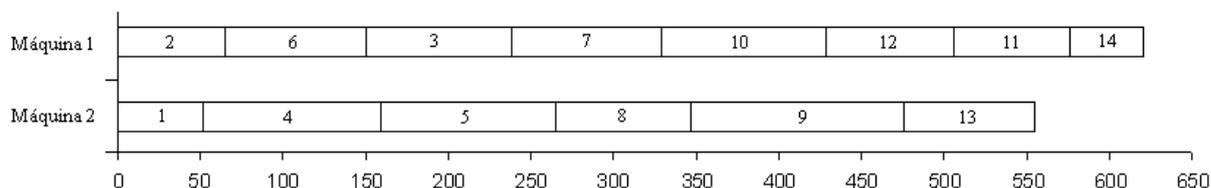


**Fonte:** Os autores

No entanto, esta distribuição de tarefas às máquinas não é ótima, pois a solução ótima do problema, obtida por enumeração, resulta no tempo total de 558 na máquina 1 e 613 na máquina 2.

A etapa 3 da heurística aplica o TS em cada máquina, refinando a solução fornecida pela etapa anterior. A Figura 3 apresenta o novo sequenciamento, gerando  $\Sigma C_j = 620$  na máquina 1 e  $\Sigma C_j = 554$  na máquina 2.

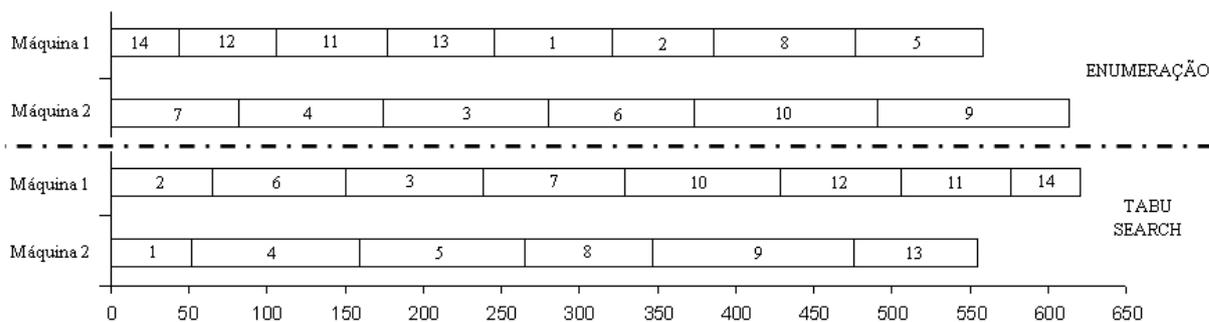
**Figura 3** – Sequência de tarefas alocadas após a execução do *Tabu Search*



**Fonte:** Os autores

A aplicação do TS reduz o  $\Sigma C_j$  da máquina 1 de 635 para 620, o qual representa a melhor solução para o conjunto de tarefas alocadas àquela máquina. Tal melhora decorre do reposicionamento das tarefas 6, 3, 10 e 14, conforme ilustrado nas Figuras 2 e 3. O tempo total da máquina 2,  $\Sigma C_j = 554$ , não foi aprimorado pelo TS, pois o resultado fornecido pela etapa 2 da heurística já se constitui na melhor solução daquele conjunto.

**Figura 4** – Solução ótima obtida por enumeração e solução obtida pela heurística após *Tabu Search*



**Fonte** – Os autores

Por fim, o resultado final da heurística é comparado com a sequência proposta pela enumeração total, na Figura 4. A heurística e a enumeração alocam o mesmo número de tarefas por máquina, contudo, o tempo total entre ambas difere, pois a enumeração não considera o fator de priorização  $w$ , utilizado na primeira etapa da heurística.

**Tabela 6** – Desempenho das etapas da heurística em relação à enumeração total

	2ª Etapa Distribuição de tarefas	3ª Etapa <i>Tabu Search</i>	Enumeração total
Tempo total	635	620	613
Desvio em relação à enumeração total	3,6%	1,1%	--

**Fonte:** Os autores

A Tabela 6 mostra que o desvio do  $\sum C_j$ , da segunda e da terceira etapa da heurística, em relação à enumeração foi de, respectivamente, 3,6% e 1,1%. Observe que as tarefas alocadas às máquinas pelo TS e pela enumeração diferem em 50% na máquina 1 (quatro entre oito tarefas) e 75% na máquina 2 (quatro entre seis tarefas). A função-objetivo de minimização do tempo total na presença de um fator de priorização ( $\sum w_j C_j$ ) é de 130.567 para a máquina 1 e 101.543 para a máquina 2.

A determinação da função-objetivo de minimização do atraso ponderado das

tarefas ( $\sum w_j T_j$ ) segue a mesma sequência de etapas que a função objetivo de minimização do tempo total na presença de um fator de priorização ( $\sum w_j C_j$ ). Os procedimentos diferem entre si, contudo, na execução da terceira etapa da heurística, onde o TS reposiciona as tarefas de acordo com as diferentes funções-objetivo.

Considerando 400 a data de entrega comum de todas as tarefas, a sequência gerada pela segunda etapa da heurística para as máquinas 1 e 2 apresenta, respectivamente, o valor de atraso ponderado de 11.045 e 5.526. Na sequência, a terceira etapa da heurística aplica o TS para aprimorar a solução obtida pela distribuição de tarefas da etapa anterior.

A Tabela 7 apresenta o desvio de  $\sum w_j T_j$  gerado pelas etapas 2 e 3 da heurística em relação à enumeração total (9,7% e 4,6%, respectivamente).

**Tabela 7** – Desempenho das etapas da heurística em relação à enumeração total

	<b>2ª Etapa</b>	<b>3ª Etapa</b>	<b>Enumeração total</b>
	<b>Distribuição de tarefas</b>	<b>Tabu Search</b>	
Atraso total	11.045	10.534	10.066
Desvio em relação à enumeração total	9,7%	4,6%	--

**Fonte:** Os autores

A Tabela 8 apresenta os desvios dos valores das duas funções-objetivo em relação à enumeração.

**Tabela 8** – Desvio da heurística em relação à enumeração total

	<b>2ª Etapa</b>	<b>3ª Etapa</b>
<b>Desvio em relação à enumeração total</b>	<b>Distribuição de tarefas</b>	<b>Tabu Search</b>
Tempo total	3,6%	1,1%
Atraso total	9,7%	4,6%

**Fonte:** Os autores

Concluída a execução da heurística, ambas as funções-objetivo apresentam desvios de seus valores em relação à enumeração inferiores a 5%, tendo sido obtidos em tempo de processamento computacional inferior a dez minutos. Esses resultados foram aprovados pelos especialistas do processo de metalização.

## 5 CONCLUSÃO

O sequenciamento é um processo de decisão estratégico de impacto direto no desempenho operacional e financeiro das empresas, oportunizando àquelas que o utilizam diferenciais competitivos no mercado que atuam. O presente artigo visou minimizar o tempo total ponderado de conclusão ( $\sum w_j C_j$ ) e o atraso ponderado ( $\sum w_j T_j$ ) de um conjunto de tarefas em um ambiente de manufatura de duas máquinas paralelas não-relacionadas com tempos determinísticos de processamento e de *setup*, este último dependente da sequência de processamento das tarefas.

A heurística proposta acrescenta uma etapa ao procedimento de Bank e Werner (2001) e se apoia em três etapas: (i) ordenar as tarefas para alocação com base na regra *WSPT*, (ii) alocar as tarefas às máquinas, e (iii) aprimorar a solução obtida na etapa (ii) através do algoritmo *Tabu Search* (TS) simples de Glover (1989), realizando o intercâmbio adjacente de tarefas de uma mesma máquina. A heurística proposta engloba fatores qualitativos e quantitativos: o fator qualitativo de priorização, representando a agregação financeira por unidade de processamento, é determinante na ordenação das tarefas na etapa 1. As etapas 2 e 3 da heurística utilizam apenas fatores quantitativos do tempo de processamento e tempo de *setup*, quando existente, para obtenção das soluções.

Quando comparada à solução ótima obtida por enumeração total, a segunda etapa da heurística apresenta desvios de 3,6% e 9,7% para as funções  $\sum w_j C_j$  e  $\sum w_j T_j$ , respectivamente. Com a aplicação do TS, o desvio para o tempo de conclusão total é reduzido para 1,1% e o desvio da solução ótima para o atraso total é reduzido para 4,6%. Estes resultados expressam a influência do fator de priorização da etapa 1 no desvio das soluções obtidas pela heurística e foram obtidos em tempo computacional inferior a dez minutos.

O mesmo problema, porém com tempos estocásticos de *setup* dependentes da sequência de processamento, passa a ser de interesse para futuras pesquisas. A utilização de funções-objetivo mais complexas, como a minimização simultânea do atraso e do adiantamento de um conjunto de tarefas sujeitas a um fator de priorização, também constitui-se em tema de interesse para o arranjo de máquinas paralelas não-relacionadas com tempos de *setup* determinísticos e dependentes da sequência de processamento.

## REFERÊNCIAS

ALLAHVERDI, A.; NG, C. T.; CHENG, T. C. E. & KOVALYOV, M. Y. A survey of scheduling problems with setup times or costs. **European Journal of Operational Research** v. 187, p. 985-1032, 2008. <http://dx.doi.org/10.1016/j.ejor.2006.06.060>

ARNAOUT, J. P.; RABADI, G. & MUSA, R. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. **Journal of Intelligent Manufacturing** v. 21, p. 693-701, 2010. <http://dx.doi.org/10.1007/s10845-009-0246-1>

ARNAOUT, J. P.; MUSA, R. & RABADI, G. Ant colony optimization algorithm to parallel machine scheduling problem with setups. **4th IEEE Conference on Automation Science and Engineering**, p. 578-582, 2008.

BANK, J. & WERNER, F. Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. **Mathematical and Computer Modelling** v. 33, p. 363-383, 2001. [http://dx.doi.org/10.1016/S0895-7177\(00\)00250-8](http://dx.doi.org/10.1016/S0895-7177(00)00250-8)

BRUCKER, P. **Scheduling algorithms**. Springer 2007.

BRUCKER, P. & KNUST, S. **Complex Scheduling**. Springer 2006.

CHANG, P. C. & CHEN, S. H. Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times. **Applied Soft Computing**, v. 11, p. 1263-1274, 2011. <http://dx.doi.org/10.1016/j.asoc.2010.03.003>

CHEN, C. & CHEN, C. Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times. **International Journal of Advanced Manufacturing Technology** v. 43, p. 161-169, 2009. <http://dx.doi.org/10.1007/s00170-008-1692-1>

CHEN, Z. & POWELL, W. B. Solving parallel machine scheduling problems by column generation. **INFORMS Journal on Computing**, v.11, p. 78-94, 1999.

<http://dx.doi.org/10.1287/ijoc.11.1.78>

CORREA, J. R.; SKUTELLA, M. & VERSCHAE, J. The power of preemption on unrelated machines and applications to scheduling orders. **LNCS**, 5687, p. 84-97, 2009.

FANJUL-PEYRO, L. & RUIZ, R. Iterated greedy local search methods for unrelated parallel machine scheduling. **European Journal of Operational Research**, v. 207 p.55-69, 2010. <http://dx.doi.org/10.1016/j.ejor.2010.03.030>

FANJUL-PEYRO, L. & RUIZ, R. Size-reduction heuristics for the unrelated parallel machines scheduling problems. **Computers & Operations Research**, v. 38, p.301-309, 2011. <http://dx.doi.org/10.1016/j.cor.2010.05.005>

GAIRING, M.; MONIEN, B. & WOCLAW, A. A faster combinatorial approximation algorithm for scheduling unrelated parallel machines. **ICALP 2005, LNCS 3580**, p. 828-839, 2005.

GLOVER, F. Tabu Search – Part I. **Journal of Computing** v.1 n.3, p. 190-206, 1989.

GLOVER, F. Tabu Search – Part II. **Journal of Computing** v.2 n.1, p. 4-32, 1990a.

GLOVER, F. Tabu Search: a tutorial. **Interfaces**, v. 20, p. 74-94, 1990b.  
<http://dx.doi.org/10.1287/inte.20.4.74>

GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K. & RINNOOY KAN, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of Discrete Mathematics**, v.5, p. 287-326, 1979.  
[http://dx.doi.org/10.1016/S0167-5060\(08\)70356-X](http://dx.doi.org/10.1016/S0167-5060(08)70356-X)

GRIGORIEV, A.; SVIRIDENKO, M. & UETZ, M. Unrelated parallel machine scheduling with resource dependent processing times. **IPCO 2005, LNCS 3509** p. P. 182-195, 2005.

HSU, C.; KUO, W. & YANG, D. Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects. **Applied Mathematical Modelling**, v.35, p. 1492-1496, 2011. <http://dx.doi.org/10.1016/j.apm.2010.09.026>

KIM, D.; KIM, K.; JANG, W. & CHEN, F. F. Unrelated parallel machine scheduling with setup times using simulated annealing. **Robotics and Computer Integrated Manufacturing**, v.18, p. 223-231, 2002. [http://dx.doi.org/10.1016/S0736-5845\(02\)00013-3](http://dx.doi.org/10.1016/S0736-5845(02)00013-3)

KIM, D.; NA, D. & CHEN, F. F. Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. **Robotics and Computer Integrated Manufacturing**, v.19, p. 173-181, 2003. [http://dx.doi.org/10.1016/S0736-5845\(02\)00077-7](http://dx.doi.org/10.1016/S0736-5845(02)00077-7)

LI, K. & YANG, S. Non-identical parallel-machine scheduling research with

minimizing total weighted completion times: Models, relaxation and algorithms. **Applied Mathematical Modelling**, v.33, p. 2145-2158, 2009. <http://dx.doi.org/10.1016/j.apm.2008.05.019>

LIAW, C. F.; LIN, Y. K.; CHENG, C. Y. & CHEN, M. Scheduling unrelated parallel machines to minimize total weighted tardiness. **Computers & Operations Research** v. 30, p. 1777-1789, 2003. [http://dx.doi.org/10.1016/S0305-0548\(02\)00105-3](http://dx.doi.org/10.1016/S0305-0548(02)00105-3)

LIN, Y. K.; PFUND, M. E. & FOWLER, J. W. Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. **Computers & Operations Research**, v.38, p. 901-916, 2011. <http://dx.doi.org/10.1016/j.cor.2010.08.018>

LOGENDRAN, R.; McDONELL, B. & SMUCKER, B. Scheduling unrelated parallel machines with sequence-dependent setups. **Computers & Operations Research**, v. 34, p. 3420-3438, 2007. <http://dx.doi.org/10.1016/j.cor.2006.02.006>

LÓPEZ, O. C.; BARCIA, R. M. & EYADA, O. Problema de programação da produção: um esquema de classificação. **Revista Produção**, v.5, p. 145-168, 1995.

MOKOTOFF, E. Parallel machine scheduling problems: a survey. **Asia-Pacific Journal of Operational Research**, v.18, p. 193-242, 2001.

MORTON, T. E. & PENTICO, D. W. **Heuristic scheduling systems: with applications to production systems and project management**. New York: John Wiley & Sons 1993.

PAULA, M. R.; MATEUS, G. R. & RAVETTI, M. G. A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times. **Computers & Operations Research**, v.37, p. 938-949, 2010. <http://dx.doi.org/10.1016/j.cor.2009.07.006>

PFUND, M.; FOWLER, J. W. & GUPTA, J. N. D. A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. **Journal of the Chinese Institute of Industrial Engineers**, v.21, n 3, p. 230-241, 2004. <http://dx.doi.org/10.1080/10170660409509404>

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

PLATEAU, M. C. & RIOS-SOLIS, Y. A. Optimal solutions for unrelated parallel machines scheduling problems using convex quadratic reformulations. **European Journal of Operational Research**, v.201, p. 729-736, 2010. <http://dx.doi.org/10.1016/j.ejor.2009.03.049>

RABADI, G.; MORAGA, R. J. & AL-SALEM, A. Heuristics for the unrelated parallel machine scheduling problem with setup times. **Journal of Intelligent Manufacturing**, v.17, p. 85-97, 2006. <http://dx.doi.org/10.1007/s10845-005-5514->

## 0

SHIM, S. Generating subproblems in branch and bound algorithms for parallel machines scheduling problems. **Computers & Industrial Engineering**, v.57, p. 1150-1153, 2009. <http://dx.doi.org/10.1016/j.cie.2009.02.013>

SILVA, C. & MAGALHÃES, J. M. Heuristic lot size scheduling on unrelated parallel machines with applications on the textile industry. **Computers & Industrial Engineering**, v.50, p. 76-89, 2006. <http://dx.doi.org/10.1016/j.cie.2006.01.001>

SRIVASTAVA, B. An effective heuristic for minimising makespan on unrelated parallel machines. **Journal of the Operational Research Society**, v.49, p. 886-894, 1998. <http://dx.doi.org/10.1057/palgrave.jors.2600547>

TAVAKKOLI-MOGHADDAM, R.; TAHERI, F.; BAZZAZI, M.; IZADI, M. & SASSANI, F. Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. **Computers & Operations Research** v. 36, p. 3224-3230, 2009. <http://dx.doi.org/10.1016/j.cor.2009.02.012>

WENG, M. X.; LU, J. & REN, H. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. **International Journal of Production Economics**, v.70, p.215-226, 2001. [http://dx.doi.org/10.1016/S0925-5273\(00\)00066-9](http://dx.doi.org/10.1016/S0925-5273(00)00066-9)

WU, Y. & JI, P. A scheduling problem for PCB assembly: a case with multiple lines. **International Journal of Manufacturing Technology**, v.43, p.1189-1201, 2009. <http://dx.doi.org/10.1007/s00170-008-1797-6>

YING, K. C.; LEE, Z. J. & LIN, S. W. Makespan minimization for scheduling unrelated parallel machines with setup times. **Journal of Intelligent Manufacturing**, v.23, p.1795-1803, 2012. <http://dx.doi.org/10.1007/s10845-010-0483-3>



Artigo recebido em 04/12/2012 e aceito para publicação em 22/07/2014  
DOI: <http://dx.doi.org/10.14488/1676-1901.v14i3.1498>