

# GRASP reativo aplicado ao problema de programação de tarefas em máquinas paralelas com tempos de preparação dependente da seqüência e de recursos

Edmar Hell Kampke<sup>1</sup>, José Elias Cláudio Arroyo<sup>1</sup>, Mauro Nacif Rocha<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal de Viçosa – UFV  
Viçosa - MG

edmar.kampke@ufv.br, jarroyo@dpi.ufv.br, nacif@dpi.ufv.br

**Abstract.** *This work deals with the parallel machine scheduling problem with resource-assignable sequence dependent setup times. The goal of the problem is to minimize the total completion time and the total assigned resources. Due to the combinatorial complexity of this problem, an algorithm based on heuristic GRASP is proposed, in which the randomness parameter used in the construction phase is self-adjusted, according to the quality of the solutions previously found (Reactive GRASP). The obtained results are compared with the best results available in literature and show the good performance of the proposed algorithm.*

**Resumo.** *Este trabalho aborda o problema de seqüenciamento de tarefas em máquinas paralelas, com tempos de preparação das máquinas dependente da seqüência e do número de recursos utilizados. O objetivo do problema é minimizar o tempo total de conclusão das tarefas e número total de recursos utilizados na preparação das máquinas. Dada a complexidade combinatória do problema, propõe-se um algoritmo baseado na heurística GRASP no qual o parâmetro de aleatoriedade utilizado na fase de construção é auto-ajustado de acordo com as soluções previamente encontradas (GRASP reativo). Os resultados obtidos são comparados com os melhores resultados disponíveis na literatura e mostram o bom desempenho do algoritmo proposto.*

## 1. Introdução

As indústrias de manufatura modernas têm sido muito estimuladas a tornar seus processos de produção mais eficientes devido, principalmente, à competitividade crescente imposta pelas transformações que têm afetado a ordem econômica mundial. As principais finalidades da programação da produção são administrar os recursos de uma empresa industrial (equipamentos de produção, espaço de armazenamento, mão-de-obra, etc), seqüenciar de forma eficiente os pedidos nos centros de trabalho e atender aos prazos de entrega dos produtos vendidos. Os problemas de programação de tarefas em máquinas encontram-se intimamente ligados ao planejamento e programação da produção e são muito estudados na literatura científica.

Neste trabalho é abordado um problema de programação de tarefas em máquinas paralelas (PTMP) no qual são considerados tempos de preparação (*setup times*) de máquinas que dependem de recursos disponíveis (por exemplo, mão-de-obra). Um dos primeiros trabalhos propostos para o problema PTMP foi apresentado por Marsh e Montgomery (1973). Eles estudaram o problema considerando a minimização de *setups times*. Guinet (1991) modelou matematicamente o problema e aplicou métodos

heurísticos para minimizar critérios tais como média dos tempos de conclusão (*completions times*) das tarefas, média dos atrasos das tarefas e o tempo máximo de conclusão (*makespan*).

Nos trabalhos mais recentes sobre problemas de PTMP, nota-se uma tendência de aplicação de metaheurísticas, tais como Busca Tabu (Armentano e Yamashita, 2000), *Simulated Annealing* (Kim *et al.*, 2002) e GRASP (Feo *et al.*, 1996; Aiex *et al.*, 2003).

O objetivo do problema abordado neste trabalho é minimizar simultaneamente dois critérios: tempo total de produção e número de recursos utilizados na preparação das máquinas. Para a resolução do problema, é proposto uma abordagem que utiliza a heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) acrescida de um mecanismo reativo para a definição do parâmetro de aleatoriedade utilizado na construção da solução inicial.

## 2. Descrição do problema

O problema de PTMP considerado neste trabalho foi abordado pela primeira vez por Ruiz e Andrés (2006). Este problema, aqui denotado por PTMPSR, consiste em determinar o melhor seqüenciamento de  $n$  tarefas em um conjunto de  $m$  máquinas paralelas diferentes (uma tarefa é processada em uma única máquina). Cada tarefa  $j$  possui um tempo de processamento na máquina  $i$  ( $p_{ij}$ ), se a tarefa  $k$  é processada logo após a tarefa  $j$  na máquina  $i$ , existe um tempo de preparação  $S_{ijk}$  cuja duração depende da quantidade de recurso disponível  $R_{ijk}$ . Cada tempo  $S_{ijk}$  pode variar entre dois valores  $S_{ijk}^-$  (*setup time* mínimo) e  $S_{ijk}^+$  (*setup time* máximo). Similarmente,  $R_{ijk}$  pode variar entre  $R_{ijk}^-$  (recurso mínimo) e  $R_{ijk}^+$  (recurso máximo). Os *setup times* e os recursos são relacionados de forma linear: se é utilizado um número mínimo (máximo) de recursos então a duração do *setup time* será o maior (menor) possível. O tempo de conclusão (*completion time*) da tarefa  $j$  na máquina  $i$  é denotado por  $C_{ij}$ . Conforme foi proposto por Ruiz e Andrés (2006), a função objetivo deste problema é definida como:

$$\text{Min } Z = \lambda \sum_{i=1}^m \sum_{j=1}^n \sum_{\substack{k=1 \\ k \neq j}}^n R_{ijk} + \delta \sum_{i=1}^m \sum_{j=1}^n C_{ij}$$

Onde,  $\lambda$  e  $\delta$  representam as importâncias ou pesos atribuídos aos critérios.  $R_{ijk} = 0$  se a tarefa  $j$  não for precedente da tarefa  $k$  na máquina  $i$ . Similarmente, se a tarefa  $j$  não for processada na máquina  $i$ , então  $C_{ij} = 0$ .

O problema PTMPSR descrito acima é provado ser NP-Difícil (Ruiz e Andrés, 2006). Esta prova consiste em simplesmente reduzir o problema a um problema clássico de PTMP com *setup times* dependentes da seqüência que é NP-Difícil.

O problema PTMPSR é de suma importância para as indústrias de manufatura. Este problema ocorre, por exemplo, na fabricação de cerâmicas, onde existem diferentes máquinas de polimento. Cada máquina inicia o polimento de um tipo de cerâmica, e ao final a máquina precisa ser limpa e preparada para realizar o polimento de outro tipo de cerâmica. Este tempo de limpeza e preparação varia de acordo com a máquina, com o tipo da cerâmica e com o número de recursos utilizados, que neste caso, pode ser o número de pessoas que realizarão a limpeza e ajuste da máquina.

A Figura 1 ilustra um exemplo de solução para um caso do problema, com  $n = 4$  tarefas e  $m = 2$  máquinas. As tarefas 4 e 2 são processadas, nesta ordem, na máquina 1, finalizando nos tempos 43 e 155, respectivamente. O *setup time* entre estas tarefas é 61, sendo utilizados 3 recursos. Da mesma forma, as tarefas 3 e 1 são processadas na máquina 2 e são finalizadas nos tempos 27 e 100, respectivamente. O *setup time* entre estas tarefas é 28 com utilização de 4 recursos. Neste trabalho, uma solução do problema é representada por um arranjo de tamanho  $n+m-1$  contendo as  $n$  tarefas e  $m-1$  elementos (-1) utilizados para dividir as tarefas em  $m$  grupos, um para cada máquina. A solução mostrada na Figura 1 é representada pelo arranjo [4, 2, -1, 3, 1]. Note que as tarefas dos grupos [4,2] e [3,1] são processadas respectivamente nas máquinas 1 e 2.

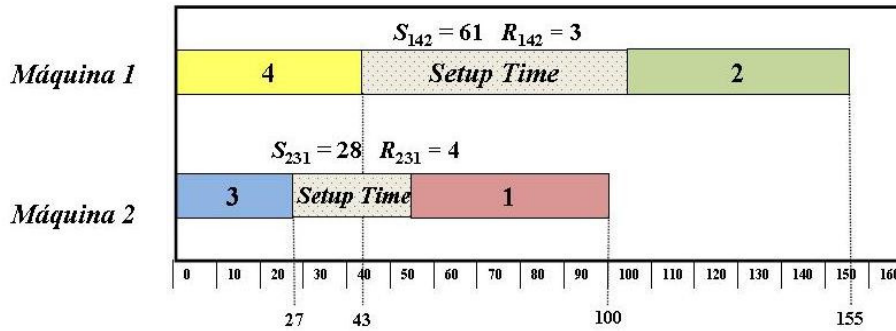


Figura 1. Exemplo de solução.

Ruiz e Andrés (2006) propuseram três heurísticas construtivas para resolver o problema, que se diferenciam basicamente pelo critério de ordenação das tarefas na construção da solução. As heurísticas são denominadas *Shortest Processing Time with Setups Resource Assignment* (SPTSA), *Shortest Processing and Setup Time with Setups Resource Assignment* (SPSTSA) e *Dynamic Job Assignment with Setups Resource Assignment* (DJASA). A heurística SPTSA utiliza o tempo de processamento como critério de ordenação das tarefas, enquanto a heurística SPSTSA considera um valor médio de todos os possíveis valores para o *setup time* além dos tempos de processamento. A heurística DJASA constrói a solução de forma dinâmica, ordenando as tarefas em ordem crescente da função objetivo, se forem incluídas naquele momento na solução parcial. Ruiz e Andrés (2006) verificaram que a heurística DJASA apresenta os melhores resultados.

### 3. GRASP Reativo para o Problema PTMPSR

A metaheurística GRASP (*Greedy Random Adaptive Search Procedure*), proposta inicialmente por Feo e Resende (1995), é um método de múltiplas partidas, na qual cada iteração consiste de duas fases: uma fase de construção de uma solução viável e uma fase de busca local, na qual se procura melhorar a qualidade da solução construída na fase anterior. A melhor solução calculada dentre todas as iterações da heurística é retornada como resultado. Os únicos parâmetros a serem definidos na heurística GRASP são o parâmetro de aleatoriedade  $\alpha$  e o critério de parada que geralmente é o número de iterações da heurística. Na Figura 2, apresenta-se um pseudocódigo genérico da heurística GRASP. Esta heurística tem como entrada o parâmetro  $\alpha$  que é utilizado na fase de construção de soluções. Para valores pequenos de  $\alpha$ , as soluções são construídas com um maior grau de “gulosidade” e para valores maiores de  $\alpha$ , as soluções são

construídas de forma mais aleatória. Da Figura 2, note que a heurística GRASP, repetidamente constrói uma solução  $s_1$  (passo 3) e esta é melhorada por um procedimento de busca local (passo 4). Sempre é armazenada a melhor solução encontrada até o momento (passos 5 a 7).

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Heurística GRASP (<math>\alpha</math>)</b><br/> 1 <math>f_{min} \leftarrow +\infty</math>;<br/> 2 <b>enquanto não</b> <i>CondiçãoParada</i> <b>faça</b><br/> 3 <math>s_1 \leftarrow</math> Construção_de_uma_Solução(<math>\alpha</math>);<br/> 4 <math>s_2 \leftarrow</math> Busca_Local(<math>s_1</math>);<br/> 5 <b>se</b> <math>f(s_2) &lt; f_{min}</math> <b>então</b><br/> 6     <math>s \leftarrow s_2</math>;<br/> 7     <math>f_{min} \leftarrow f(s_2)</math>;<br/> 8 <b>fim-se</b><br/> 9 <b>fim enquanto</b>;<br/> 10 retorne <math>s</math>;</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Figura 2. Pseudocódigo genérico da Heurística GRASP.

Para a resolução do problema PTMPSR abordado neste trabalho é proposta a aplicação da heurística GRASP reativo (Prais e Ribeiro, 2000) que tem como principal característica o auto-ajustamento do parâmetro de aleatoriedade  $\alpha$  utilizado na fase de construção, de acordo com a qualidade das soluções encontradas nas iterações anteriores.

A heurística GRASP reativo funciona da seguinte maneira: define-se um conjunto  $A$  de  $v$  possíveis valores para o parâmetro  $\alpha$ ,  $A = \{\alpha_1, \alpha_2, \dots, \alpha_v\}$ . Inicialmente, todos os valores  $\alpha_k \in A$  tem a mesma probabilidade de serem escolhidos, ou seja,  $p_k = 1/v$ . A cada iteração escolhe-se um valor  $\alpha_k \in A$  (com probabilidade  $p_k$ ) para o parâmetro  $\alpha$ . As probabilidades  $p_k$  são recalculadas a cada  $\gamma$  iterações. Os valores de  $\alpha_k$  que produziram melhores soluções terão maior probabilidade  $p_k$ , conseqüentemente, nas próximas iterações terão maior chance de serem escolhidos.

De acordo com a estrutura do GRASP reativo, a quantidade de soluções construídas utilizando  $\alpha_k$  é armazenada no arranjo *count* e a soma dos valores da função objetivo dessas soluções são armazenadas no arranjo *score*. Se  $f_{min}$  é o menor valor da função objetivo encontrada até o momento, os valores de  $p_k$  serão atualizados utilizando *count*, *score*,  $f_{min}$  e um parâmetro de amplificação  $\theta$ .

Na Figura 3 apresenta-se a estrutura genérica da heurística GRASP reativo. Nos passos 2 a 7, são inicializadas as variáveis e estruturas usadas no ajuste do parâmetro  $\alpha$ . No passo 9 seleciona-se aleatoriamente, com probabilidade  $p_k$ , um  $\alpha_k \in A$  que será utilizado na iteração atual. No passo 16 é incrementado o número de soluções construídas com  $\alpha_k$  e é acumulado o valor da função objetivo da solução encontrada nessa iteração. Os passos 18 a 21, correspondentes à atualização das probabilidades  $p_k$ , são executados a cada  $\gamma$  iterações.

Na implementação do GRASP reativo foi considerado o seguinte conjunto de valores para o parâmetro  $\alpha$ :  $A = \{0,1;0,2;\dots;0,9\}$ . Os valores dos parâmetros  $\gamma$  e  $\theta$  foram calibrados e os melhores resultados foram obtidos para  $\gamma = 20$  e  $\theta = 10$ .

```

Heurística GRASP-Reativo;
1  $f_{min} \leftarrow +\infty;$ 
2  $iter \leftarrow 1;$ 
3 Defina  $A = \{a_1, a_2, \dots, a_v\}$ 
5 para  $k \leftarrow 1$  até  $v$  faça
6    $count[k] \leftarrow 0; score[k] \leftarrow 0; p_k \leftarrow 1/v;$ 
7 fim para;
8 enquanto não CondiçãoParada faça
9   Selecione  $a_k \in A$  com probabilidade de escolha  $p_k;$ 
10   $s_1 \leftarrow$  Construção_de_uma_Solução( $a_k$ );
11   $s_2 \leftarrow$  Busca_Local( $s_1$ );
12  se  $f(s_2) < f_{min}$  então
13     $s \leftarrow s_2;$ 
14     $f_{min} \leftarrow f(s_2);$ 
15  fim-se
16   $count[k] \leftarrow count[k] + 1; score[k] \leftarrow score[k] + f(s_2);$ 
17  se  $iter \bmod \gamma = 0$  então
18     $avg[k] \leftarrow score[k]/count[k]$  para todo  $k \in \{1, 2, \dots, v\};$ 
19     $Q_k \leftarrow (f_{min}/avg[k])^\theta$  para todo  $k \in \{1, 2, \dots, v\};$ 
20     $\sigma \leftarrow \sum Q_k;$ 
21     $p_k \leftarrow Q_k/\sigma$  para todo  $k \in \{1, 2, \dots, v\};$ 
22  fim-se
24 fim enquanto;
25 retorne  $s;$ 

```

Figura 3. Estrutura genérica da heurística GRASP reativo.

### 3.1. Construção de soluções

Na fase de construção, uma solução (seqüência de tarefas) é gerada gradativamente. Iniciando com uma seqüência vazia, a cada iteração, é adicionada uma única tarefa à seqüência parcial. Para escolher a tarefa a ser adicionada, é definida uma Lista de Candidatos (LC) com todas as tarefas ainda não seqüenciadas. Para cada tarefa dessa lista são feitas simulações de inclusão da tarefa nas máquinas da seqüência parcial. As tarefas desta lista são então ordenadas de forma crescente pelo incremento que elas proporcionarão ao valor da função objetivo. Esta forma de ordenação das tarefas também é usada na heurística construtiva DJASA proposta por Ruiz e Andrés (2006). Na heurística DJASA sempre é adicionada a primeira tarefa, ou seja, a tarefa que apresenta o menor incremento ao valor da função objetivo. No algoritmo de construção implementado neste trabalho, ao invés de escolher a primeira tarefa, é escolhida aleatoriamente uma tarefa, dentre as  $\eta$  primeiras tarefas da lista LC, para ser adicionada na seqüência parcial. Estas  $\eta$  primeiras tarefas formam uma Lista Restrita de Candidatas (LRC) cujo tamanho depende do parâmetro  $\alpha$ . O valor de  $\eta$  é definida como  $\text{MAX}(1, \alpha \times |LC|)$ . A Figura 4 mostra o algoritmo de construção de soluções.

```

Procedimento Construção_de_uma_Solução( $a$ );
1.  $LC \leftarrow \{t_1, \dots, t_n\}$ ; // Lista de todas as tarefas
2.  $Seqüência \leftarrow \{ \}$ ;
3. enquanto ( $|Seqüência| < n$ ) faça
4.   Ordena_LC();
5.    $\eta \leftarrow \text{Max}(1, \alpha \times |LC|)$ ;
6.    $i \leftarrow \text{Random}(1, \eta)$ ; //Escolhe aleatoriamente uma posição de LC
7.    $Seqüência \leftarrow Seqüência \cup \{t_i\}$ ; //Adiciona na Seqüência tarefa da posição  $i$  ( $t_i$ )
8.    $LC \leftarrow LC - \{t_i\}$ ; //Remove tarefa  $t_i$  de LC
10. fim enquanto
11. Retorne Seqüência;

```

Figura 4. Algoritmo de construção de soluções.

### 3.2. Busca Local

A Busca Local é um procedimento iterativo que consiste em melhorar uma solução  $s_1$  procurando novas soluções vizinhas dela. Estas soluções vizinhas são obtidas realizando algumas alterações (movimentos) na estrutura da solução atual  $s_1$ . Escolhe-se um vizinho  $s$  dentre todos os vizinhos de  $s_1$ . Se o vizinho escolhido  $s$  é melhor que  $s_1$ , a busca continua a partir de  $s$  (ou seja,  $s_1 \leftarrow s$ ). O procedimento finaliza quando não seja possível melhorar a solução atual  $s_1$ , ou seja, quando  $s_1$  é um ótimo local. Neste trabalho foi utilizado movimentos de inserção para a obtenção de soluções vizinhas. Um vizinho de  $s_1$  é gerado inserindo uma tarefa que está na posição  $i$  da seqüência em outra posição  $j$ , tal que  $1 \leq i, j \leq n$  e  $i \neq j$ . Por exemplo, para a seqüência [4, 2, -1, 3, 1] da Figura 1, as seqüências vizinhas [4, -1, 3, 2, 1] e [2, 4, -1, 3, 1] são gradadas através da inserção da tarefa 2 após a tarefa 3 e antes da tarefa 4, respectivamente. Realizando este tipo de movimento é possível gerar  $k(k-1)$  soluções vizinhas, onde  $k = n+m-1$  (tamanho da seqüência).

### 3.3. Critério de Parada

Neste trabalho adota-se como critério de parada da heurística GRASP reativo o tempo computacional, baseado no número de tarefas ( $n$ ) e número de máquinas ( $m$ ) de cada problema. O tempo computacional utilizado como critério de parada é de  $(n \times m)/2$  segundos.

## 4. Resultados Computacionais

Para testar o desempenho da heurística GRASP na resolução do problema PTMPSR, são utilizados os problemas teste gerados por Ruiz e Andrés (2006) que são disponibilizados na Internet. Eles geraram um total de 720 problemas, divididos em dois grupos: *small* e *large* (cada um com 360 problemas). O grupo *small* contém problemas com número de tarefas  $n \in \{6,8,10\}$  e número de máquinas  $m \in \{3,4,5\}$ . Já no grupo de problemas *large* os valores para o número de tarefas ( $n$ ) e máquinas ( $m$ ) pertencem respectivamente aos seguintes conjuntos:  $\{50,75,100\}$  e  $\{10,15,20\}$ .

Em ambos os grupos de problemas, os tempos de processamento das tarefas são gerados aleatoriamente (com distribuição uniforme) no intervalo [1,99]. Os recursos e *setup times* são também gerados aleatoriamente (com distribuição uniforme) dentro a alguns intervalos predefinidos. Para os recursos mínimos ( $R^-$ ) foram definidos 2 tipos de intervalos [1, 3] e [1, 5]. Para os recursos máximos ( $R^+$ ) os intervalos [3, 5] e [5, 10]. Os

*setup times* mínimos ( $S^-$ ) consideram os intervalos [1, 50] e [50, 100]; e os *setup times* máximos ( $S^+$ ) [50, 100] e [100, 150].

Fazendo as combinações de  $n$ ,  $m$  e os intervalos considerados para os recursos e *setup times*, é possível gerar 36 combinações. Para cada combinação foram gerados 10 problemas. As combinações dos parâmetros de entrada dos problemas são mostradas nas primeiras 4 colunas das Tabelas 1 e 2. Para todos os problemas, os parâmetros  $\lambda$  e  $\delta$ , que representam os pesos dos critérios otimizados, foram os mesmos utilizados por Ruiz e Andrés (2006):  $\lambda=50$  e  $\delta=1$ .

O algoritmo GRASP reativo foi implementado na linguagem de programação Java na versão 1.6 e executado usando o compilador JDK 6.0. Os problemas teste foram resolvidos utilizando um computador com processador Intel® Core™ Quad com 2.4GHz e 3GB de memória RAM.

Os resultados obtidos pela heurística GRASP reativo são comparados com os melhores resultados disponibilizados por Ruiz e Andrés (2006). As soluções disponibilizadas para o grupo de problemas *small* foram obtidas pelo software CPLEX (versão 9.1) através da resolução do Modelo Matemático de Programação Inteira proposto por Ruiz e Andrés (2006). A execução do CPLEX foi limitada a 300 segundos para problemas com  $n = 6$  tarefas e 3600 segundos para problemas com  $n = 8$  e  $n = 10$  tarefas. Para todos os problemas com  $n = 6$  e 8, o CPLEX determinou a solução ótima. Para todos os problemas com  $n = 10$  somente foram obtidas soluções aproximadas, devido ao limite do tempo de execução do CPLEX.

A Tabela 1 exhibe os resultados encontrados pelo Software CPLEX (Ruiz e Andrés, 2006) e pelo GRASP reativo, para os 360 problemas do grupo *small*. Nesta Tabela apresentam-se as médias dos valores da função objetivo  $Z$  a cada 10 problemas que possuem as mesmas dimensões. A coluna *Melhoria\_Média* exhibe os percentuais de melhoria do GRASP reativo com relação aos resultados do CPLEX, onde  $Melhoria\_Média = 100 \times (CPLEX - GRASP\ Reativo) / CPLEX$ .

Na Tabela 1 observa-se quatro valores negativos do percentual de melhoria média da heurística GRASP reativo com relação ao CPLEX. Esses valores negativos indicam que a heurística não conseguiu encontrar a solução ótima em alguns problemas com  $n = 6$  tarefas. Nota-se também que em todos os grupos de problemas com  $n = 8$  tarefas, o GRASP reativo encontrou a solução ótima. Para os problemas com  $n = 10$  tarefas todos os valores do percentual de melhoria média são maiores que zero. Isto indica que para esses problemas o GRASP reativo obteve resultados melhores que os limites superiores obtidos pelo CPLEX. O percentual de melhoria média obtido varia entre 1% e 5,01%.

Os resultados obtidos pela heurística GRASP reativo para os problemas do grupo *small* foram bastante satisfatórios. Comparando com os resultados do CPLEX, do total de 360 problemas, em apenas 6 problemas a heurística obteve resultado inferior, em 90 problemas os resultados foram superiores e nos 264 problemas restantes foram obtidas as mesmas soluções.

**Tabela 1. Comparação de resultados do algoritmo GRASP Reativo x MPI (CPLEX 9.1) detalhado por Ruiz e Andrés (2006)**

| <i>n</i>     | <i>m</i> | Problemas                        |                                  | Médias de Z  |                      | Melhoria Média (%) |
|--------------|----------|----------------------------------|----------------------------------|--------------|----------------------|--------------------|
|              |          | <i>S</i> ; <i>S</i> <sup>+</sup> | <i>R</i> ; <i>R</i> <sup>+</sup> | <i>CPLEX</i> | <i>GRASP Reativo</i> |                    |
| 6            | 3        | 1-50;50-100                      | 1-3;3-5                          | 6.318        | 6.318                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 7.169        | 7.169                | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 7.986        | 7.997                | -0,14              |
|              |          |                                  | 1-5;5-10                         | 8.057        | 8.057                | 0,00               |
|              | 4        | 1-50;50-100                      | 1-3;3-5                          | 4.360        | 4.360                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 4.200        | 4.200                | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 5.168        | 5.168                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 5.238        | 5.238                | 0,00               |
|              | 5        | 1-50;50-100                      | 1-3;3-5                          | 2.738        | 2.745                | -0,26              |
|              |          |                                  | 1-5;5-10                         | 2.473        | 2.473                | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 3.148        | 3.181                | -1,05              |
|              |          |                                  | 1-5;5-10                         | 3.128        | 3.131                | -0,10              |
| 8            | 3        | 1-50;50-100                      | 1-3;3-5                          | 11.029       | 11.029               | 0,00               |
|              |          |                                  | 1-5;5-10                         | 12.236       | 12.236               | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 14.811       | 14.811               | 0,00               |
|              |          |                                  | 1-5;5-10                         | 16.047       | 16.047               | 0,00               |
|              | 4        | 1-50;50-100                      | 1-3;3-5                          | 8.287        | 8.287                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 8.772        | 8.772                | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 9.965        | 9.965                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 10.709       | 10.709               | 0,00               |
|              | 5        | 1-50;50-100                      | 1-3;3-5                          | 5.748        | 5.748                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 6.459        | 6.459                | 0,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 7.193        | 7.193                | 0,00               |
|              |          |                                  | 1-5;5-10                         | 7.916        | 7.916                | 0,00               |
| 10           | 3        | 1-50;50-100                      | 1-3;3-5                          | 16.302       | 16.003               | 1,83               |
|              |          |                                  | 1-5;5-10                         | 18.590       | 18.301               | 1,55               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 23.441       | 22.866               | 2,45               |
|              |          |                                  | 1-5;5-10                         | 26.326       | 25.870               | 1,73               |
|              | 4        | 1-50;50-100                      | 1-3;3-5                          | 12.993       | 12.782               | 1,62               |
|              |          |                                  | 1-5;5-10                         | 13.414       | 13.280               | 1,00               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 16.462       | 16.051               | 2,50               |
|              |          |                                  | 1-5;5-10                         | 17.245       | 16.950               | 1,71               |
|              | 5        | 1-50;50-100                      | 1-3;3-5                          | 9.409        | 9.147                | 2,78               |
|              |          |                                  | 1-5;5-10                         | 10.165       | 9.925                | 2,36               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 12.314       | 11.848               | 3,78               |
|              |          |                                  | 1-5;5-10                         | 12.785       | 12.144               | 5,01               |
| <i>Média</i> |          |                                  |                                  | -            | -                    | <b>0,74</b>        |

A Tabela 2 exibe os melhores resultados obtidos pelas heurísticas propostas por Ruiz e Andrés (2006) e pelo GRASP reativo, para os 360 problemas do grupo *large*. Apresentam-se as médias dos valores da função objetivo Z para cada 10 problemas que possuem as mesmas dimensões. Da mesma forma que na Tabela 1, a coluna



*Melhoria\_Média* exibe o percentual de melhoria do GRASP reativo com relação aos melhores resultados encontrados pelas heurísticas da literatura.

**Tabela 2. Comparação de resultados do algoritmo GRASP reativo x melhor resultado das heurísticas construtivas de Ruiz e Andrés (2006)**

| Problemas    |          |                                  |                                  | Médias de Z               |               | Melhoria Média (%) |
|--------------|----------|----------------------------------|----------------------------------|---------------------------|---------------|--------------------|
| <i>n</i>     | <i>m</i> | <i>S</i> ; <i>S</i> <sup>+</sup> | <i>R</i> ; <i>R</i> <sup>+</sup> | Heurísticas Ruiz e Andrés | GRASP Reativo |                    |
| 50           | 10       | 1-50;50-100                      | 1-3;3-5                          | 92.615                    | 81.917        | 11,55              |
|              |          |                                  | 1-5;5-10                         | 112.832                   | 101.576       | 9,98               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 143.256                   | 132.697       | 7,37               |
|              |          |                                  | 1-5;5-10                         | 164.401                   | 152.668       | 7,14               |
|              | 15       | 1-50;50-100                      | 1-3;3-5                          | 67.979                    | 59.580        | 12,36              |
|              |          |                                  | 1-5;5-10                         | 74.990                    | 68.586        | 8,54               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 97.742                    | 91.287        | 6,60               |
|              |          |                                  | 1-5;5-10                         | 105.534                   | 99.829        | 5,41               |
|              | 20       | 1-50;50-100                      | 1-3;3-5                          | 49.928                    | 45.712        | 8,44               |
|              |          |                                  | 1-5;5-10                         | 53.859                    | 48.534        | 9,89               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 70.002                    | 65.900        | 5,86               |
|              |          |                                  | 1-5;5-10                         | 74.022                    | 69.500        | 6,11               |
| 75           | 10       | 1-50;50-100                      | 1-3;3-5                          | 166.415                   | 148.069       | 11,02              |
|              |          |                                  | 1-5;5-10                         | 236.000                   | 198.327       | 15,96              |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 286.680                   | 270.416       | 5,67               |
|              |          |                                  | 1-5;5-10                         | 354.873                   | 322.741       | 9,05               |
|              | 15       | 1-50;50-100                      | 1-3;3-5                          | 130.347                   | 115.674       | 11,26              |
|              |          |                                  | 1-5;5-10                         | 159.490                   | 144.975       | 9,10               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 205.400                   | 192.547       | 6,26               |
|              |          |                                  | 1-5;5-10                         | 236.376                   | 224.618       | 4,97               |
|              | 20       | 1-50;50-100                      | 1-3;3-5                          | 107.804                   | 94.441        | 12,40              |
|              |          |                                  | 1-5;5-10                         | 120.296                   | 111.389       | 7,40               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 160.067                   | 149.086       | 6,86               |
|              |          |                                  | 1-5;5-10                         | 172.409                   | 165.719       | 3,88               |
| 100          | 10       | 1-50;50-100                      | 1-3;3-5                          | 246.576                   | 224.869       | 8,80               |
|              |          |                                  | 1-5;5-10                         | 359.014                   | 309.431       | 13,81              |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 469.931                   | 450.091       | 4,22               |
|              |          |                                  | 1-5;5-10                         | 585.176                   | 534.074       | 8,73               |
|              | 15       | 1-50;50-100                      | 1-3;3-5                          | 195.793                   | 178.274       | 8,95               |
|              |          |                                  | 1-5;5-10                         | 270.150                   | 237.265       | 12,17              |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 336.943                   | 321.392       | 4,62               |
|              |          |                                  | 1-5;5-10                         | 409.763                   | 383.384       | 6,44               |
|              | 20       | 1-50;50-100                      | 1-3;3-5                          | 167.631                   | 150.767       | 10,06              |
|              |          |                                  | 1-5;5-10                         | 206.661                   | 190.977       | 7,59               |
|              |          | 50-100;100-150                   | 1-3;3-5                          | 267.809                   | 253.632       | 5,29               |
|              |          |                                  | 1-5;5-10                         | 306.226                   | 296.107       | 3,30               |
| <b>Média</b> |          |                                  |                                  | -                         | -             | <b>8,25</b>        |

Para os 360 problemas do grupo *large*, o algoritmo GRASP reativo obteve sempre uma melhoria em relação às heurísticas da literatura, ou seja, o valor de *Melhoria\_Média* sempre foi  $> 0$ . Também foi observado que em 102 problemas a melhoria foi maior que 10% e em apenas 6 problemas a melhoria foi menor que 3,3%.

## 5. Conclusões

Neste trabalho foi proposta uma heurística GRASP reativo para um problema de programação de tarefas em máquinas paralelas, denominado PTMPSR. O problema PTMPSR foi recentemente formulado na literatura e até o momento não existem trabalhos de aplicação da heurística GRASP. O desempenho da heurística proposta foi testado em 720 problemas, incluindo de pequeno e grande porte. As soluções obtidas pela heurística foram comparadas com as melhores soluções encontradas na literatura. Para os problemas de grande porte, a heurística GRASP obteve uma melhoria média de 8,25% com relação às melhores soluções heurísticas disponibilizadas na literatura. Como trabalho futuro sugere-se o uso de outros procedimentos de busca e refinamento, tais como *Path Relinking*, de forma a melhorar ainda mais os resultados obtidos.

**Agradecimentos:** Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq e pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais – FAPEMIG.

## Referências bibliográficas

- Aiex, R.M., Binato, S. e Resende, M.G.C. (2003), “Parallel GRASP with path-relinking for job shop scheduling”. *Parallel Computing*, 29:393–430.
- Armentano, V.A. e Yamashita, D.S. (2000), “Tabu search for scheduling on identical parallel machines to minimize mean tardiness”. *J Intell Manuf.* 11:453–460.
- Feo, T.A. e Resende, M.G.C. (1995), “Greedy randomized adaptive search procedures”. *Journal of Global Optimization*, 6, 109-133.
- Feo, T.A., Sarathy, K. e McGahan, J. (1996), “A GRASP for single machine scheduling with sequence dependent setup costs and linear delay penalties”. *Computers & Operations Research*, 23:881-895.
- Guinet, A. (1991), “Textile production systems: a succession of non-identical parallel processor shops”. *Journal of the Operational Research Society*, 42(8): 655-671.
- Kim, D-W., Kim, K-H, Jang, W. e Chen, F.F.(2002), “Unrelated parallel machine scheduling with setup times using simulated annealing”. *Robotics and Computer Integrated Manufacturing*, 18: 223-231.
- Marsh, J.D. e Montgomery, D.C. (1973), “Optimal procedures for scheduling Jobs with sequence-dependent changeover times on parallel processors”. *AIIE Technical Papers*, 279-286.
- Prais, M. e Ribeiro, C. (2000), “Reactive GRASP: na application to a matrix decomposition problem in TDMA traffic assignment”. *INFORMS Journal on Computing*, 12(3):164-176.
- Ruiz, R. e Andrés, C. (2006), “Unrelated parallel machines scheduling with resource-assignable sequence dependent setup times”. *MISTA Conference*, 3, 439-446.