

Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas

Horacio Hideki Yanasse

Rita de Cássia Meneses Rodrigues

Edson Luiz França Senne



Resumo

No problema de minimização de troca de ferramentas procura-se por uma sequência para processar um conjunto de tarefas de modo que o número requerido de trocas de ferramentas seja o menor possível. Neste trabalho propõe-se um algoritmo para resolver este problema baseado em um ordenamento parcial das tarefas. Uma sequência ótima é obtida expandindo-se as sequências parciais enumeradas. Testes computacionais são apresentados.

Palavras-chave: Problema de minimização de trocas de ferramenta. Ordenamento parcial. Limitante inferior.

1 Introdução

Considere um ambiente produtivo em que se tem um conjunto de tarefas $T = \{1, \dots, N\}$ a serem processadas sequencialmente e sem interrupção em uma única máquina flexível de manufatura e um conjunto de ferramentas $F = \{1, \dots, M\}$. Seja T_f o conjunto de tarefas que requerem a ferramenta $f \in F$. Cada tarefa $t \in T$ requer que um subconjunto de ferramentas F_t seja colocado na máquina e somente quando todo este subconjunto de ferramentas está na máquina é possível processar a tarefa t . Considere que a máquina é capaz de conter no máximo C ferramentas de cada vez, em que $C \geq \max_t |F_t|$. Suponha que a capacidade C de armazenamento de ferramentas da máquina seja inferior ao número total de ferramentas necessário para processar todas as tarefas. Do contrário, o problema é trivial, pois todas as ferramentas podem ser carregadas na máquina e todas as tarefas podem então ser processadas sem que sejam necessárias trocas de ferramentas. Se a capacidade for inferior, trocas de ferramentas serão necessárias. Uma troca de ferramentas consiste em remover uma ferramenta da máquina e incluir outra em seu lugar. A ordem das ferramentas na máquina é irrelevante. O problema de minimizar o número de trocas de ferramentas (*Minimization of Tool Switches Problem – MTSP*) consiste

em determinar uma sequência de processamento das tarefas de modo a minimizar o número de trocas de ferramentas necessárias.

Belady (1966), McGeoch e Sleator (1991) e Privault e Finke (2000) descrevem aplicações do MTSP referentes à administração de memória em sistemas de computação. Este problema ocorre quando páginas (ferramentas) precisam ser transferidas de uma memória secundária para a memória principal (compartimento de ferramentas da máquina) para executar algumas tarefas. Variações e extensões do MTSP são apresentadas por Blazewicz e Finke (1994), Crama (1997), Balakrishnan e Chakravarty (2001), VanHop (2005) e Crama et al. (2007). Oerlemans (1992) e Crama et al. (1994) apresentam uma prova formal de que o MTSP é NP-difícil para $C \geq 2$ (GAREY; JOHNSON, 1979). Bard (1988) e Tang e Denardo (1988) mostram que minimizar o número de trocas de ferramentas para uma dada sequência de tarefas pode ser resolvido em tempo polinomial por uma política conhecida como KTNS (*Keep Tool Needed Soonest*). Esta política estabelece que, quando trocas de ferramentas são necessárias, as primeiras ferramentas requeridas para uma próxima tarefa devem ser as primeiras a serem mantidas na máquina. Blazewicz e Finke (1994)

mencionam que a política KTNS já era conhecida por Belady (1966). Várias heurísticas têm sido propostas para o MTSP (BARD, 1988; KIRAN; KRASONM 1988; TANG; DENARDO, 1988; OERLEMANS, 1992; GRAY et al., 1993; HERTZ; WIDMER, 1993; CRAMA et al., 1994; FOLLOMIER, 1994; SODHI et al, 1994; AVCI; AKTURK, 1996; HERTZ et al., 1998; PRIVAULT; FINKE, 2000; KNUUTILA et al., 2002; ZHOU et al., 2005; SALONEN et al., 2006; AMAYA et al., 2008). Tang e Denardo (1988) propõem uma formulação do MTSP como um modelo de Programação Linear Inteira que, resolvido por um esquema baseado em Programação Linear, levou a resultados computacionais ruins. Laporte et al. (2004) mostram que uma razão para o desempenho ruim do modelo de Tang e Denardo (1988) é o fato de o valor da relaxação linear deste modelo ser sempre igual a zero e propõem uma nova formulação. Laporte et al. (2004) também propõem dois algoritmos exatos para o problema: um deles é um algoritmo do tipo *branch-and-cut*, que resolve a nova formulação, e o outro algoritmo é uma aplicação direta do método *branch-and-bound*, que não usa relaxação de Programação Linear. Yanasse e Lamosa (2005) reescrevem o problema de minimização de trocas de ferramentas como um problema do caixeiro viajante generalizado, em que cada nó representa um possível conjunto de ferramentas carregadas na máquina e os arcos que ligam dois nós neste grafo têm um custo associado igual ao número mínimo de trocas de ferramentas para se ir de uma configuração de ferramentas para a outra. Como existem potencialmente vários nós que contêm as ferramentas necessárias para se processar cada tarefa, basta que se visite apenas uma delas para que a tarefa seja processada. Daí resulta que o problema de minimização de trocas de ferramentas é equivalente a um problema do caixeiro viajante generalizado neste grafo. Ghiani et al. (2007) exploram simetrias existentes no problema e melhoram o algoritmo *branch-and-bound* proposto por Laporte et al. (2004).

De nosso conhecimento, são muito poucos os métodos exatos para resolver o problema de minimização de troca de ferramentas. Além disso, eles têm tido sucesso limitado em problemas pequenos. No trabalho de Laporte et al. (2004) são reportados resultados computacionais em que apenas alguns poucos exemplares de problemas com 25 tarefas puderam ser resolvidos de maneira ótima com um algoritmo do tipo *branch-and-bound*.

A resolução exata de exemplares maiores utilizando esquemas enumerativos é dependente, por exemplo, da determinação de boas soluções para o problema (limitantes superiores) e de bons limitantes inferiores, de modo que se consiga diminuir o espaço de busca da solução. Neste trabalho, apresenta-se um esquema enumerativo que tenta determinar um limitante inferior melhor para o MTSP utilizando exemplares menores obtidos a partir do problema original, considerando-se apenas um subconjunto de

tarefas. Este subconjunto de tarefas é ordenado e, então, utilizado, também, para determinar uma solução ótima para o problema original.

2 Propriedades de uma solução do MTSP

As soluções do MTSP possuem algumas características que podem ser utilizadas no desenvolvimento de métodos mais eficientes para a sua solução.

Lema 1: Seja $S_1 = (s_1, s_2, \dots, s_k)$ uma sequência de tarefas, com $s_i \in T_1 \subseteq T$ ($i = 1, 2, \dots, k$). Considere um conjunto de tarefas T_2 tal que $T_2 \subseteq T_1$. Da sequência S_1 elimine as tarefas s_j ($j = 1, 2, \dots, k$) tais que $s_j \notin T_2$ e denote a nova sequência obtida por S_2 . O número mínimo necessário de trocas de ferramentas para processar as tarefas de T_2 utilizando a sequência S_2 é, no máximo, igual ao número mínimo necessário de trocas de ferramentas para processar as tarefas de T_1 utilizando a sequência S_1 .

Prova: O número mínimo de trocas de ferramentas usando a sequência S_1 não aumenta quando algumas tarefas são retiradas desta sequência. Para demonstrar isto, analisemos o que acontece com a retirada de uma única tarefa qualquer, digamos s_t , da sequência $S_1 = (s_1, s_2, \dots, s_{t-1}, s_t, s_{t+1}, \dots, s_k)$. A nova sequência será $S_1 - \{s_t\} = (s_1, s_2, \dots, s_{t-1}, s_{t+1}, \dots, s_k)$. Considere as trocas de ferramentas necessárias para se obter o mínimo de trocas de ferramentas para processar as tarefas de T_1 utilizando a sequência S_1 . Seguindo estas trocas de ferramentas, a cada tarefa processada s_i da sequência S_1 , teremos, em correspondência, um quadro Q_i de ferramentas na máquina no momento em que a tarefa foi processada. Portanto, correspondente à sequência $S_1 = (s_1, s_2, \dots, s_{t-1}, s_t, s_{t+1}, \dots, s_k)$, temos uma sequência de quadros de ferramentas na máquina $V_1 = (Q_1, Q_2, \dots, Q_{t-1}, Q_t, Q_{t+1}, \dots, Q_k)$, que indicam as ferramentas na máquina no momento em que cada uma das tarefas de S_1 é processada. O número de trocas de ferramentas a cada tarefa processada é obtido comparando-se o quadro de ferramentas correspondente à tarefa em consideração com o quadro de ferramentas imediatamente anterior em V_1 . Observe que para processar as tarefas da nova sequência $S_1 - \{s_t\} = (s_1, s_2, \dots, s_{t-1}, s_{t+1}, \dots, s_k)$, podemos utilizar uma sequência de quadros de ferramentas na máquina igual a $V_1 - \{Q_t\} = (Q_1, Q_2, \dots, Q_{t-1}, Q_{t+1}, \dots, Q_k)$, ou seja, a mesma sequência de quadros de ferramentas para a sequência S_1 , excluindo-se Q_t . Com esta sequência de quadros de ferramentas, as ferramentas necessárias para processar as tarefas da nova sequência $S_1 - \{s_t\}$ estariam todas disponíveis na máquina. Comparando-se as duas sequências V_1 e $V_1 - \{Q_t\}$, a diferença nas trocas de ferramentas ocorre apenas na subsequência de quadros Q_{t-1}, Q_t, Q_{t+1} em V_1 em comparação com a subsequência de quadros Q_{t-1}, Q_{t+1} em $V_1 - \{Q_t\}$. Como as trocas de ferramentas obedecem à desigualdade triangular, o número de trocas de Q_{t-1} para Q_{t+1} é sempre menor ou igual ao número de trocas de Q_{t-1} para Q_t e de Q_t para

Q_{i+1} . Portanto, o número mínimo necessário de trocas de ferramentas para processar as tarefas de $S_1 - \{s_1\}$ é, no máximo, igual ao número mínimo necessário de trocas de ferramentas para processar as tarefas de T_1 utilizando a sequência S_1 . Aplicando-se este resultado sucessivamente, isto é, considerando-se a retirada de uma tarefa por vez, tem-se o resultado do Lema 1.

Lema 2: Seja $S_2 = (s_1, s_2, \dots, s_k)$ uma sequência de tarefas, com $s_i \in T_2 \subseteq T$ ($i = 1, 2, \dots, k$). Insira, na sequência S_2 , tarefas adicionais pertencentes a $T_1 - T_2$, em que $T_2 \subseteq T_1$, resultando em uma sequência S_1 . Então, o número mínimo necessário de trocas de ferramentas usando-se a sequência S_1 é maior ou igual ao número mínimo necessário de trocas de ferramentas usando-se a sequência S_2 .

Prova: Se o número mínimo necessário de trocas de ferramentas usando S_1 fosse menor, então, eliminando-se desta sequência as tarefas que não estão em T_2 , seria obtida a sequência S_2 , cujo número de trocas de ferramentas seria, pelo Lema 1, no máximo igual ao da sequência S_1 . Isto é um absurdo, pois este número seria menor que o número mínimo necessário de trocas de ferramentas usando S_2 .

Teorema 1: Seja E_1 um exemplar do MTSP com um conjunto de tarefas $T_1 \subseteq T$, e E_2 um outro exemplar do MTSP com um conjunto de tarefas $T_2 \subseteq T_1$. Então, o valor ótimo z_1 de E_1 é maior ou igual ao valor ótimo z_2 de E_2 .

Prova: Considere uma solução ótima de E_1 com valor ótimo z_1 . Elimine, desta sequência ótima de tarefas de E_1 , as tarefas que não fazem parte de E_2 . Pelo Lema 1, o número de trocas de ferramentas para esta nova sequência, que é viável para E_2 , é no máximo igual a z_1 .

Usando-se o Teorema 1, pode-se obter limitantes inferiores para qualquer exemplar do MTSP, resolvendo-se de maneira ótima exemplares menores do MTSP obtidos do exemplar original e considerando apenas um subconjunto de suas tarefas. Obviamente, quanto maior o número de tarefas consideradas, melhor a qualidade do limitante obtido.

Teorema 2: Seja $S_1 = (s_1, s_2, \dots, s_k)$ uma sequência de tarefas, com $s_i \in T_1 \subseteq T$ ($i = 1, 2, \dots, k$). Seja S_2 uma outra sequência de tarefas obtida invertendo-se a ordem das tarefas de S_1 , ou seja, $S_2 = (s_k, s_{k-1}, \dots, s_1)$. Então, o número mínimo de trocas de ferramentas obtidas utilizando-se a sequência S_1 é igual ao número mínimo de trocas de ferramentas utilizando-se a sequência S_2 .

Prova: Considerando-se a ordem inversa das trocas de ferramentas para a sequência S_1 , tem-se o mesmo número total de trocas de ferramentas e, seguindo-se esta sequência de trocas, pode-se processar S_2 . Analogamente, considerando-se a ordem inversa de trocas de ferramentas para a sequência S_2 tem-se o mesmo número total de trocas de ferramentas e, seguindo-se esta sequência, pode-se processar S_1 . Para demonstrar isto, similarmente à prova do Lema 1, consideremos uma sequência de quadros de ferramentas $V_1 = (Q_1, Q_2, \dots, Q_k)$, que indicam as ferramentas na máquina no momento em que cada uma das tarefas de S_1 é processada. Como já exposto, o número

de trocas de ferramentas a cada tarefa processada é obtido comparando-se o quadro de ferramentas correspondente à tarefa em consideração com o quadro de ferramentas imediatamente anterior na sequência. Considere agora a seguinte sequência de quadros de ferramentas $V_2 = (Q_k, \dots, Q_2, Q_1)$ obtida invertendo-se a ordem da sequência de quadros de V_1 . Com a sequência de quadros V_2 podemos processar as tarefas da sequência S_2 , pois as ferramentas necessárias para processar cada uma das tarefas estão nos quadros correspondentes de ferramentas. Como o número de trocas de ferramentas é o mesmo se mudarmos de um quadro Q_i para Q_j ou vice-versa, o número total de trocas de ferramentas para as duas sequências é o mesmo e tem-se o resultado do teorema.

Recentemente tomamos conhecimento de que o Teorema 2 foi anteriormente publicado em Ghiani et al. (2007).

Usando-se os Teoremas 1 e 2 e os Lemas 1 e 2, foi desenvolvido o esquema enumerativo apresentado na próxima seção.

3 Um novo esquema enumerativo

A ideia básica do esquema é identificar uma ordem ótima, em uma solução (sequência) ótima, de um subconjunto do conjunto original de tarefas. Uma vez obtida uma ordem ótima deste subconjunto de tarefas, as demais tarefas são inseridas neste subconjunto ordenado de tarefas, antes, entre as tarefas, ou depois delas, de modo que o número de trocas de ferramentas não cresça ou cresça o menos possível.

Para ilustrar, apresenta-se a seguir a aplicação desta ideia a um pequeno exemplo. Seja o exemplar $TESTE_1$ do problema MTSP cujos dados estão apresentados na Tabela 1. Este exemplar tem $M = 5$, $N = 8$ e $C = 3$.

Considere um subconjunto das tarefas de $TESTE_1$, por exemplo, $\{2, 8\}$. Em uma solução ótima, estas tarefas vão ser processadas em alguma ordem. Existem apenas duas possibilidades de sequenciamento destas tarefas: (2, 8) (em que a tarefa 2 é processada antes da tarefa 8) e (8, 2) (em que a tarefa 8 é processada antes da tarefa 2). O número mínimo de trocas de ferramentas nt , quando se processam estas tarefas seguindo-se estas sequências, é o mesmo e igual a 2.

Tabela 1. Dados do exemplar $TESTE_1$.

Tarefa	Ferramentas requeridas
1	1, 4
2	1, 3, 5
3	2, 4
4	3, 4
5	1, 5
6	3, 5
7	4, 5
8	1, 2, 4

Pelo Teorema 1, conclui-se que $nt = 2$ é um limitante inferior para o número de trocas de ferramentas do exemplar $TESTE_1$. Pelo Teorema 2, pode-se considerar apenas uma destas sequências, por exemplo, (2, 8).

Com a sequência parcial (2, 8), observa-se que as ferramentas 1, 3 e 5 vão estar na máquina para processar a tarefa 2. Assim, pode-se processar as tarefas 5 e 6 logo após o processamento da tarefa 2, sem aumentar o número de trocas de ferramentas, resultando em uma nova sequência parcial (2, 5, 6, 8). De maneira similar, as ferramentas 1, 2 e 4 vão estar na máquina para processar a tarefa 8 e, então, pode-se processar as tarefas 3 e 1 logo após o processamento da tarefa 8, sem aumento no número de trocas de ferramentas.

Por outro lado, com a sequência de tarefas (2, 8), observa-se que as tarefas 4 e 7 não podem ser processadas sem que haja alguma troca de ferramentas. Considera-se, então, inserir uma delas, por exemplo, a tarefa 4, que pode ser colocada antes da tarefa 2, entre as tarefas 2 e 8, ou após a tarefa 8. O número mínimo de trocas de ferramentas (NMTF) necessário para processar estas sequências parciais de tarefas está listado na Tabela 2.

A sequência (2, 4, 8) apresenta o menor número de trocas de ferramentas. Esta sequência parcial, portanto, mantém o limitante inferior atual para o número de trocas de ferramentas para o exemplar $TESTE_1$. Repete-se, então, o procedimento inserindo outra tarefa nesta sequência e verificando se este limitante inferior pode ser mantido.

Com a sequência parcial (2, 4, 8), as tarefas 5, 6, 3 e 1 podem ser processadas sem nenhuma troca de ferramentas, porém a tarefa 7 não pode ser sequenciada sem trocas de ferramentas. Assim, considera-se inserir a tarefa 7 antes da tarefa 2, entre as tarefas 2 e 4, entre as tarefas 4 e 8, ou depois da tarefa 8, resultando em novas sequências parciais, como mostra a Tabela 3.

Das Tabelas 2 e 3, o limitante inferior atual para $TESTE_1$ é 3. Das sequências parciais da Tabela 3 pode-se derivar as seguintes soluções para $TESTE_1$ que requerem um total de 3 trocas de ferramentas: (7, 2, 5, 6, 4, 8, 3, 1), (2, 5, 6, 7, 4, 8, 3, 1), (2, 5, 6, 4, 7, 8, 3, 1) e (2, 5, 6, 4, 8, 3, 1, 7). Foram encontradas, portanto, soluções viáveis (limitantes superiores) para $TESTE_1$ com um total de 3 trocas de ferramentas. Como o limitante inferior atual também é igual a 3, pode-se concluir que estas soluções viáveis são ótimas para $TESTE_1$.

No exemplar $TESTE_1$, foram examinadas 8 possibilidades, ao invés das 8! permutações possíveis das tarefas (ou $4! = 24$, uma vez excluídas as tarefas dominadas deste exemplo).

Tabela 2. Número mínimo de trocas de ferramentas para 3 tarefas selecionadas.

Sequência parcial	NMTF
(4, 2, 8)	3
(2, 4, 8)	2
(2, 8, 4)	3

A eficiência computacional deste procedimento enumerativo, que usa um ordenamento parcial de tarefas, depende do subconjunto de tarefas selecionadas, uma vez que elas influenciam diretamente no valor do limitante inferior determinado para o problema original. Deseja-se obter o menor número possível de tarefas que produzam o melhor limitante inferior possível para o problema. Com este intuito, propõem-se as seguintes regras de seleção de tarefas para o algoritmo:

3.1 Escolha inicial de tarefas

Inicie selecionando o menor número possível de tarefas que juntas utilizem todas as ferramentas do problema. Se existem conjuntos de tarefas que utilizam todas as ferramentas com a mesma cardinalidade mínima, escolha o subconjunto cujas tarefas compartilham o maior número de ferramentas em comum.

3.2 Seleção da próxima tarefa a ser inserida

Escolha a tarefa com o maior número mínimo de trocas de ferramentas com as tarefas já selecionadas. Se houver empate, escolha a tarefa que utiliza o maior número de ferramentas. Caso haja novo empate, escolha uma delas arbitrariamente.

Ilustram-se estas regras com o exemplar $TESTE_2$ apresentado na Tabela 4. Neste exemplar tem-se $N = 25$, $M = 15$ e $C = 10$.

Este exemplar pode ser reduzido, uma vez que existem tarefas que são dominadas por outras, pois requerem um subconjunto de suas ferramentas. Seja F_i o conjunto de ferramentas requeridas pela tarefa i . Na Tabela 4, tem-se, por exemplo: $F_1 \subset F_{16}$, $F_2 \subset F_5$. Na Tabela 5 apresenta-se o exemplar reduzido, com $N = 15$ tarefas, obtido após o pré-processamento em que foram eliminadas todas as tarefas dominadas.

Para a escolha inicial de tarefas, pode-se, por exemplo, combinar as tarefas duas a duas e verificar que ferramentas elas utilizam. Ao fazer isto, verifica-se que a única combinação que utiliza todas as 15 ferramentas é o conjunto formado pelas tarefas 24 e 25. Portanto, considerando a sequência (24, 25) (ou, equivalentemente, (25, 24)), tem-se um limitante inferior para o número mínimo de trocas de ferramentas para $TESTE_2$ igual a 5. Observe que este valor é igual ao limitante trivial para o problema, dado por $M - C = 5$, uma vez que todas as M ferramentas devem passar em algum momento pela máquina, que tem capacidade C .

Tabela 3. Número mínimo de trocas de ferramentas para 4 tarefas selecionadas.

Sequência parcial	NMTF
(7, 2, 4, 8)	3
(2, 7, 4, 8)	3
(2, 4, 7, 8)	3
(2, 4, 8, 7)	3

Considere a sequência parcial (24, 25). Pelo Teorema 2, pode-se abandonar a sequência parcial (25, 24). Para selecionar a próxima tarefa a ser inserida, é conveniente manter uma Tabela com o número de trocas de ferramentas que ocorre quando se processa isoladamente quaisquer duas tarefas de $TESTE_2$, uma após a outra, como na Tabela 6.

Pela regra de seleção da tarefa a ser inserida, escolhe-se a tarefa que resulta no maior número mínimo de trocas, ou seja, a tarefa que sequenciada antes ou após qualquer uma das tarefas já presentes na sequência parcial atual resulte no maior número mínimo de trocas de ferramentas. Pela Tabela 7, obtida das duas últimas linhas da Tabela 6, pode-se observar que as tarefas 16, 19 e 20 requerem, pelo menos, 3 trocas de ferramentas para serem processadas juntamente com as tarefas 24 e 25 (em qualquer sequência). Dentre estas 3 tarefas candidatas, pode-se observar que a tarefa 19 requer, no mínimo, 7 trocas de ferramentas para ser processada juntamente com as tarefas 24 e 25 (em qualquer sequência). Assim, esta tarefa deve ser selecionada como a próxima tarefa a ser inserida.

Considera-se inserir a tarefa 19 antes da tarefa 24, entre as tarefas 24 e 25, ou após a tarefa 25 na sequência parcial (24, 25). Para estas possibilidades, tem-se as seguintes soluções parciais:

(19, 24, 25), com NMFT = 8;

(24, 19, 25), com NMFT = 7;

(24, 25, 19), com NMFT = 9.

Portanto, como a sequência (24, 19, 25) corresponde à sequência parcial mais longa com o melhor limitante inferior, considera-se esta sequência como a próxima solução parcial. Note que, desta forma, o limitante inferior atual para o exemplar $TESTE_2$ passa a ser igual a 7.

Pela Tabela 8, obtida da Tabela 6, observa-se que as tarefas 16, 21, 22 e 23 requerem, pelo menos, 2 trocas de ferramentas para serem processadas juntamente com as tarefas 19, 24 e 25 (em qualquer sequência). Dentre estas 4 tarefas candidatas, observa-se que a tarefa 23 requer, no mínimo, 9 trocas de ferramentas para ser processada juntamente com as tarefas 19, 24 e 25 (em qualquer sequência). Assim, esta tarefa deve ser selecionada como a próxima tarefa a ser inserida.

Com isto, tem-se as seguintes sequências parciais possíveis:

(23, 24, 19, 25), com NMFT = 11;

(24, 23, 19, 25), com NMFT = 11;

(24, 19, 23, 25), com NMFT = 8;

(24, 19, 25, 23), com NMFT = 9.

Como a sequência (24, 19, 23, 25) corresponde à sequência parcial mais longa com o melhor limitante inferior, considera-se esta sequência como a próxima solução parcial. Portanto, o limitante inferior atual para o exemplar $TESTE_2$ passa a ser igual a 8.

A partir desta nova solução parcial e considerando os valores da Tabela 6, determina-se a próxima tarefa a ser inserida, como mostra a Tabela 9. Observe que as tarefas 21 e 22 requerem, pelo menos, 2 trocas de ferramentas para serem processadas juntamente com as tarefas 19, 23, 24 e 25 (em qualquer sequência). Dentre estas duas tarefas candidatas, a tarefa 22 é escolhida como a próxima a ser inserida na sequência parcial atual, porque requer 10 trocas de ferramentas e utiliza o maior número de ferramentas (#F = 9).

Com isto, as próximas sequências parciais possíveis são as seguintes:

(22, 24, 19, 23, 25), com NMFT = 11;

(24, 22, 19, 23, 25), com NMFT = 11;

(24, 19, 22, 23, 25), com NMFT = 10;

(24, 19, 23, 22, 25), com NMFT = 10;

(24, 19, 23, 25, 22), com NMFT = 10.

Tabela 4. Dados do exemplar $TESTE_2$.

Tarefa	Ferramentas requeridas	Tarefa	Ferramentas requeridas
1	1, 2	14	2, 3, 5, 7, 11, 13
2	5, 15	15	1, 4, 6, 8, 9, 10
3	4, 7, 11	16	1, 2, 5, 8, 10, 11, 12
4	2, 8, 9	17	3, 4, 5, 6, 8, 9, 11
5	5, 13, 15	18	4, 7, 8, 11, 12, 13, 14
6	1, 7, 9, 13	19	1, 3, 5, 6, 7, 9, 11, 15
7	3, 7, 10, 11	20	1, 4, 5, 6, 7, 8, 9, 10
8	4, 5, 7, 8, 9	21	4, 5, 7, 8, 9, 10, 11, 13
9	2, 7, 9, 10, 15	22	2, 3, 4, 5, 9, 10, 13, 14, 15
10	1, 3, 5, 8, 11	23	1, 2, 5, 8, 9, 10, 12, 14, 15
11	2, 5, 9, 11, 14	24	3, 4, 5, 6, 8, 9, 11, 12, 13, 14
12	7, 8, 9, 11, 13, 14	25	1, 2, 4, 7, 9, 10, 12, 13, 14, 15
13	1, 4, 5, 11, 12, 14		

Tabela 5. Exemplar $TESTE_2$ após eliminação de tarefas dominadas.

Tarefa	Ferramentas requeridas	Tarefa	Ferramentas requeridas
7	3, 7, 10, 11	19	1, 3, 5, 6, 7, 9, 11, 15
10	1, 3, 5, 8, 11	20	1, 4, 5, 6, 7, 8, 9, 10
11	2, 5, 9, 11, 14	21	4, 5, 7, 8, 9, 10, 11, 13
12	7, 8, 9, 11, 13, 14	22	2, 3, 4, 5, 9, 10, 13, 14, 15
13	1, 4, 5, 11, 12, 14	23	1, 2, 5, 8, 9, 10, 12, 14, 15
14	2, 3, 5, 7, 11, 13	24	3, 4, 5, 6, 8, 9, 11, 12, 13, 14
16	1, 2, 5, 8, 10, 11, 12	25	1, 2, 4, 7, 9, 10, 12, 13, 14, 15
18	4, 7, 8, 11, 12, 13, 14		

Tabela 6. Número mínimo de trocas de ferramentas para pares de tarefas não dominadas.

	7	10	11	12	13	14	16	18	19	20	21	22	23	24	25
7	0	0	0	0	0	0	0	0	0	0	0	1	2	2	2
10	0	0	0	0	0	0	0	0	0	0	0	2	1	1	4
11	0	0	0	0	0	0	0	0	0	1	0	1	0	1	2
12	0	0	0	0	0	0	1	0	1	1	0	2	2	1	2
13	0	0	0	0	0	0	0	0	1	1	1	2	1	1	2
14	0	0	0	0	0	0	0	0	0	2	0	1	3	2	3
16	0	0	0	1	0	0	0	1	2	1	1	3	0	3	3
18	0	0	0	0	0	0	1	0	3	2	0	3	3	1	2
19	0	0	0	1	1	0	2	3	0	1	2	3	3	3	4
20	0	0	1	1	1	2	1	2	1	0	0	3	2	3	3
21	0	0	0	0	1	0	1	0	2	0	0	2	3	2	3
22	1	2	1	2	2	1	3	3	3	3	2	0	2	3	2
23	2	1	0	2	1	3	0	3	3	2	3	2	0	4	2
24	2	1	1	1	1	2	3	1	3	3	2	3	4	0	5
25	2	4	2	2	2	3	3	2	4	3	3	2	2	5	0

Tabela 7. Número mínimo de trocas de ferramentas envolvendo as tarefas 24 e 25.

	7	10	11	12	13	14	16	18	19	20	21	22	23	24	25
24	2	1	1	1	1	2	3	1	3	3	2	3	4	0	5
25	2	4	2	2	2	3	3	2	4	3	3	2	2	5	0
NMTF	4	5	3	3	3	5	6	3	7	6	5	5	6		

Tabela 8. Número mínimo de trocas de ferramentas envolvendo as tarefas 19, 24 e 25.

	7	10	11	12	13	14	16	18	19	20	21	22	23	24	25
19	0	0	0	1	1	0	2	3	0	1	2	3	3	3	4
24	2	1	1	1	1	2	3	1	3	3	2	3	4	0	5
25	2	4	2	2	2	3	3	2	4	3	3	2	2	5	0
NMTF	4	5	3	4	4	5	8	6		7	7	8	9		

Tabela 9. Número mínimo de trocas de ferramentas envolvendo as tarefas 19, 23, 24 e 25.

	7	10	11	12	13	14	16	18	19	20	21	22	23	24	25
19	0	0	0	1	1	0	2	3	0	1	2	3	3	3	4
23	2	1	0	2	1	3	0	3	3	2	3	2	0	4	2
24	2	1	1	1	1	2	3	1	3	3	2	3	4	0	5
25	2	4	2	2	2	3	3	2	4	3	3	2	2	5	0
NMTF	6	6	3	6	5	8	8	9		9	10	10			
#F											8	9			

Seguindo o mesmo raciocínio, prossegue-se com a inserção de novas tarefas e atualização do limitante inferior, até que todas as tarefas tenham sido consideradas. Desta forma, chega-se à seguinte sequência ótima de tarefas: (11, 22, 14, 21, 7, 18, 12, 24, 13, 19, 10, 20, 16, 23, 25), que corresponde à sequência (11, 22, 5, 14, 21, 7, 18, 3, 12, 24, 17, 13, 19, 2, 10, 20, 8, 15, 16, 1, 23, 4, 25, 6, 9) para o exemplar original TESTE₂, com 13 trocas de ferramentas.

4 Resultados computacionais

O novo algoritmo proposto foi implementado na linguagem C e os testes computacionais foram realizados em um microcomputador com processador Intel Core 2 Duo, 3GHz, 2GB RAM. Na enumeração implementada utilizou-se a estratégia de busca em profundidade, de modo a tentar evitar uma grande necessidade de memória computacional para armazenar os nós ativos da árvore de enumeração.

Dada uma sequência qualquer de tarefas, para computar o número mínimo de trocas de ferramentas necessário para processar as tarefas nesta sequência foi implementado o algoritmo KTNS (*Keep Tool Needed Soonest*) de Tang e Denardo (TANG; DENARDO, 1988).

Os experimentos computacionais foram conduzidos com exemplares gerados aleatoriamente como indicado em Crama e outros (CRAMA et al., 1994): dadas as quantidades mínima e máxima de ferramentas requeridas por qualquer tarefa, gera-se aleatoriamente um número dentro deste intervalo para estabelecer a quantidade de ferramentas requeridas para cada tarefa. As ferramentas requeridas são, então, estabelecidas aleatoriamente. Em todos os exemplares utilizados para os testes, todas as

ferramentas são utilizadas pelas tarefas e, portanto, não existem tarefas dominadas.

Os resultados computacionais obtidos encontram-se nas Tabelas 10 a 14. Estas tabelas contêm:

- N: Número de tarefas;
- M: Número de ferramentas;
- Min: Quantidade mínima de ferramentas requeridas por uma tarefa;
- Max: Quantidade máxima de ferramentas requeridas por uma tarefa;
- C: Capacidade de carregamento de ferramentas na máquina;
- NP: Número de problemas resolvidos dentro do limite de tempo;

Tabela 10. Resultados computacionais para exemplares com 8 tarefas.

N	M	Min	Max	C	NP	LGS			YRS		
						NG	TC	NP	NG	TC	
8	15	2	5	5	10	5503,9	81,9	10	139,2	0,0	
8	15	2	5	10	10	35409,3	371,2	10	161,4	0,0	
8	15	5	10	10	10	3917,4	72,1	10	32,0	0,0	
8	15	2	10	10	10	7382,9	99,7	10	59,3	0,0	
8	20	2	5	5	10	9564,9	178,2	10	2282,7	0,1	
8	20	2	5	10	10	50006,5	775,7	10	2276,3	0,0	
8	20	5	10	10	10	5790,4	189,7	10	88,1	0,0	
8	20	2	10	10	10	14915,8	325,0	10	86,8	0,0	
8	20	2	5	15	10	36585,4	469,2	10	2276,3	0,0	
8	20	5	10	15	10	23799,9	404,6	10	39,5	0,0	
8	20	10	15	15	10	2131,9	47,0	10	25,9	0,0	
8	20	2	10	15	10	31119,1	476,8	10	106,3	0,0	
8	20	5	15	15	10	9014,8	222,6	10	18,0	0,0	
8	20	2	15	15	10	10206,2	248,3	10	54,9	0,0	
8	25	2	5	5	10	12169,4	252,0	10	2522,6	0,1	
8	25	2	5	10	10	52531,5	1055,6	10	2521,0	0,0	
8	25	5	10	10	10	5447,8	202,0	10	385,4	0,0	
8	25	2	10	10	10	16854,5	580,0	10	919,6	0,1	
8	25	2	5	15	10	52700,2	928,0	10	2521,0	0,0	
8	25	5	10	15	10	32057,8	1057,3	10	417,0	0,0	
8	25	10	15	15	10	1226,2	46,2	10	111,8	0,0	
8	25	2	10	15	10	43993,8	1037,3	10	173,6	0,0	
8	25	5	15	15	10	13533,6	572,3	10	88,0	0,0	
8	25	2	15	15	10	22865,3	816,7	10	113,1	0,0	
8	25	2	5	20	10	13,7	0,8	10	2521,0	0,0	
8	25	5	10	20	10	42120,2	684,9	10	522,5	0,0	
8	25	10	15	20	10	14394,6	443,2	10	130,7	0,0	
8	25	15	20	20	10	614,8	11,4	10	13,3	0,0	
8	25	2	10	20	10	42367,1	693,0	10	1137,7	0,0	
8	25	5	15	20	10	23426,0	502,0	10	146,2	0,0	
8	25	10	20	20	10	7615,7	209,0	10	21,2	0,0	
8	25	2	15	20	10	35652,1	638,1	10	43,6	0,0	
8	25	5	20	20	10	8437,1	235,0	10	22,4	0,0	
8	25	2	20	20	10	16092,0	373,6	10	24,3	0,0	

- NG: Número de nós gerados na árvore de busca;
- TC: Tempo computacional, em segundos.

Em cada linha destas tabelas apresenta-se o resultado da resolução de 10 exemplares, limitando o tempo de execução em 3600 segundos. Os valores das colunas NG e TC destas tabelas referem-se à média dos exemplares resolvidos dentro do limite de tempo estabelecido. Nestas tabelas, a coluna LGS reproduz os resultados de Laporte e outros (LAPORTE et al., 2002). Os tempos computacionais reportados naquele trabalho referem-se a um computador Sunfire 4800 de 900 MHz. Na coluna YRS tem-se os resultados obtidos pelo algoritmo proposto neste trabalho.

Pelos resultados mostrados nas Tabelas 10 a 14 pode-se observar que o algoritmo proposto resolve exemplares

de até 15 tarefas sem dificuldades, deixando de resolver apenas 1 exemplar do problema, dentre os 1090 exemplares considerados. Levando-se em conta somente estes exemplares das Tabelas 10 a 13, o tempo computacional médio para o algoritmo YRS resolver um exemplar do problema é de apenas 7,4 segundos.

A comparação de tempos computacionais para as abordagens LGS e YRS não é perfeita, pois os testes foram realizados em computadores diferentes. Assim, para efeito de comparação destas abordagens, foram determinados o número total de problemas resolvidos (Total NP) e o valor médio para o número de nós gerados na árvore de busca (Média NG), para cada uma das tabelas de resultados (Tabelas 10 a 14). Estes valores estão mostrados na Tabela 15.

Tabela 11. Resultados computacionais para exemplares com 9 tarefas.

N	M	Min	Max	C	NP	LGS		NP	YRS	
						NG	TC		NG	TC
9	15	2	5	5	10	28091,0	767,0	10	304,0	0,0
9	15	2	5	10	9	134725,6	2432,1	10	334,9	0,0
9	15	5	10	10	9	10334,2	383,1	10	50,7	0,0
9	15	2	10	10	10	37044,3	902,5	10	34,4	0,0
9	20	2	5	5	9	40273,4	1283,5	10	2398,5	0,1
9	20	2	5	10	9	126077,2	2825,2	10	1432,0	0,0
9	20	5	10	10	9	21376,3	1214,1	10	234,5	0,0
9	20	2	10	10	6	31548,0	1424,2	10	146,3	0,0
9	20	2	5	15	9	98304,3	1737,7	10	2312,1	0,0
9	20	5	10	15	5	67785,8	2415,1	10	96,9	0,0
9	20	10	15	15	10	9045,2	366,6	10	8,9	0,0
9	20	2	10	15	4	129573,5	2532,0	10	62,8	0,0
9	20	5	15	15	9	43409,3	1886,8	10	50,9	0,0
9	20	2	15	15	8	41242,5	1948,2	10	30,5	0,0
9	25	2	5	5	9	47458,3	1688,5	10	3092,4	0,3
9	25	2	5	10	5	117100,4	2976,3	10	2528,3	0,0
9	25	5	10	10	9	24348,3	1601,9	10	519,1	0,1
9	25	2	10	10	1	50494,0	3178,6	10	510,4	0,0
9	25	2	5	15	9	118109,2	2731,3	10	2528,2	0,0
9	25	10	15	15	10	5397,9	349,4	10	186,9	0,0
9	25	2	10	15	1	120895,0	3459,9	10	746,3	0,1
9	25	5	15	15	4	41433,5	2701,9	10	142,4	0,0
9	25	2	15	15	1	116744,0	3267,5	10	352,8	0,0
9	25	2	5	20	10	42,0	1,7	10	2527,4	0,0
9	25	5	10	20	7	125601,0	2925,8	10	225,7	0,0
9	25	10	15	20	1	50700,0	3095,0	10	346,1	0,0
9	25	15	20	20	10	3078,9	97,7	10	29,3	0,0
9	25	2	10	20	9	111075,9	2526,5	10	713,5	0,0
9	25	5	15	20	3	80084,0	2812,4	10	179,9	0,0
9	25	10	20	20	10	32740,4	1588,9	10	30,1	0,0
9	25	2	15	20	4	125347,8	3086,1	10	97,4	0,0
9	25	5	20	20	8	35628,6	1779,6	10	35,7	0,0
9	25	2	20	20	5	54889,0	2016,8	10	49,1	0,0

Tabela 12. Resultados computacionais para exemplares com 10 e 15 tarefas.

NT	NF	Min	Max	C	LGS			YRS		
					NP	NG	TC	NP	NG	TC
10	10	2	4	4	10	2545,7	0,0	10	113,1	0,0
10	10	2	4	5	10	2397,3	0,0	10	96,8	0,0
10	10	2	4	6	10	1413,5	0,0	10	66,3	0,0
10	10	2	4	7	10	61,3	0,0	10	137,4	0,0
15	20	2	6	6	10	2654120,8	21,8	10	11531,6	2,4
15	20	2	6	8	10	1355251,7	10,7	10	27050,3	6,3
15	20	2	6	10	10	335957,7	2,5	10	124182,1	34,6
15	20	2	6	12	10	214,4	0,0	10	90269,1	22,8

Tabela 13. Resultados computacionais para exemplares com 15 tarefas.

NT	NF	Min	Max	C	LGS			YRS		
					NP	NG	TC	NP	NG	TC
15	15	2	5	5	10	872889,9	6,3	10	8150,5	1,3
15	15	2	5	10	10	203,1	0,0	10	123929,8	23,7
15	15	5	10	10	10	12695066,5	146,5	10	177,5	0,0
15	15	2	10	10	10	3248709,3	38,6	10	604,5	0,1
15	20	2	5	5	10	4015611,7	30,6	10	75428,9	15,6
15	20	2	5	10	10	236,8	0,0	10	62014,9	16,9
15	20	5	10	10	10	6973355,3	94,0	10	5983,7	1,3
15	20	2	10	10	10	54641596,0	647,3	10	8165,2	1,8
15	20	2	5	15	10	120,0	0,0	10	1716,7	0,0
15	20	5	10	15	10	925615,8	10,7	10	287683,3	74,4
15	20	10	15	15	10	1995247,1	41,0	10	4,5	0,0
15	20	2	10	15	10	1494274,5	15,5	10	11288,6	2,7
15	20	5	15	15	10	45984478,8	669,9	10	574,6	0,1
15	20	2	15	15	9	24426805,0	394,4	10	87,6	0,0
15	25	2	5	5	10	2934256,2	23,3	10	266502,5	68,3
15	25	2	5	10	10	140,3	0,0	10	50595,6	16,8
15	25	5	10	10	10	25601323,3	385,1	10	152228,9	43,0
15	25	2	10	10	10	28979040,4	371,2	10	15053,4	4,3
15	25	2	5	15	10	120,0	0,0	10	2589,2	0,0
15	25	5	10	15	10	1324492,4	18,6	10	125520,3	35,4
15	25	10	15	15	10	3376786,9	80,5	10	52130,7	15,0
15	25	2	10	15	10	609674,0	6,9	10	72739,3	23,3
15	25	5	15	15	8	54864983,3	915,5	10	3079,3	0,8
15	25	2	15	15	10	9713799,6	134,4	10	19482,3	5,7
15	25	2	5	20	10	120,0	0,0	10	2587,8	0,0
15	25	5	10	20	10	40618,1	0,4	9	1130986,4	336,8
15	25	10	15	20	10	7822686,2	168,8	10	8954,4	2,0
15	25	15	20	20	10	1933537,9	58,5	10	3,9	0,0
15	25	2	10	20	10	139,1	0,0	10	931,2	0,1
15	25	5	15	20	10	9559616,3	139,0	10	134142,9	41,6
15	25	10	20	20	10	29057947,8	700,5	10	52,7	0,0
15	25	2	15	20	10	9713799,6	134,4	10	24070,0	7,1
15	25	5	20	20	9	52172340,4	1037,8	10	414,0	0,1
15	25	2	20	20	8	68791747,2	1336,9	10	10383,7	2,8

Tabela 14. Resultados computacionais para exemplares com 20 e 25 tarefas.

NT	NF	LGS				YRS				
		Min	Max	C	NP	NG	TC	NP	NG	TC
20	15	2	5	5	8	131899592,2	1375,0	10	773831,3	219,9
20	15	2	5	10	9	1671881,2	14,9	6	2823187,0	946,9
20	15	2	10	10	1	15334759,0	170,4	10	11535,0	3,4
20	20	2	5	5	6	86609305,0	929,8	10	827490,5	273,3
20	20	2	5	10	10	25376623,1	225,0	3	46072,0	19,9
20	20	2	5	15	10	221,2	0,0	9	2644,6	0,4
20	20	5	10	15	4	110670505,0	1771,9	10	1112506,4	418,3
20	20	2	10	15	4	23796204,5	349,3	9	1224997,3	492,7
20	25	2	5	5	3	116304539,3	1226,6	5	3609894,2	1460,3
20	25	2	5	10	10	17861308,4	170,2	1	5603193,0	2539,1
20	25	2	5	15	10	225,5	0,0	10	17395,2	8,4
20	25	5	10	15	1	33106709,0	536,3	5	2662282,6	1149,3
20	25	10	15	15	2	39590854,0	1432,4	6	2364492,7	1119,6
20	25	2	10	15	5	67379248,8	1025,4	8	834895,5	431,2
20	25	2	5	20	10	210,0	0,0	10	2653,5	0,0
20	25	5	10	20	8	16763635,6	274,2	2	981168,5	472,6
20	25	2	10	20	10	42837,6	0,5	6	1550,7	0,4
20	25	5	15	20	1	146490505,0	3094,2	10	52079,5	24,3
25	15	2	5	10	4	9863072,2	84,1	0	–	–
25	20	2	5	10	1	187936788,0	2284,9	0	–	–
25	20	2	10	15	1	7458838,0	102,5	2	2064398,0	1197,0
25	25	2	5	10	3	7320014,0	73,9	0	–	–
25	25	2	5	15	10	895,3	0,0	0	–	–
25	25	2	5	20	10	332,8	0,0	10	2835,0	0,1
25	25	5	10	20	1	688098,0	10,9	0	–	–
25	25	2	10	20	8	4725531,6	70,6	2	423,5	0,1

Tabela 15. Resultados computacionais médios.

Tabela	LGS		YRS	
	Total NP	Média NG	Total NP	Média NG
10	340	20278,3	340	647,3
11	232	63030,3	330	676,9
12	80	543995,3	80	31680,8
13	334	13640334,7	339	78184,1
14	150	40418951,3	144	1191406,0

Pela Tabela 15, observa-se que a abordagem YRS foi capaz de resolver mais problemas (1233, contra 1136 da abordagem LGS), com menor esforço computacional (média de 260519 nós gerados na árvore de busca, contra 10937318 nós gerados na abordagem LGS).

Pelos resultados mostrados na Tabela 14, observa-se que o novo algoritmo apresenta um desempenho pior nos casos em que há “folgas” entre o número máximo de ferramentas utilizado pelas tarefas e a capacidade C da máquina. Quanto maior esta folga, mais tarefas podem

ser processadas conjuntamente sem que trocas sejam realizadas, e esta característica não é explorada pelo novo procedimento proposto.

5 Conclusões

Neste trabalho propõe-se um novo esquema enumerativo, baseado em ordenamentos parciais, para resolver o problema de minimizar o número de trocas de ferramentas em uma máquina flexível de manufatura.

O algoritmo proposto utiliza regras simples de seleção de tarefas a serem inseridas nas sequências parciais que vão sendo construídas. Outras regras podem ser propostas e avaliadas. Os resultados computacionais obtidos indicam um sucesso limitado do novo algoritmo. A nova proposta obtém melhores resultados em exemplares em que o algoritmo de Laporte et al. (2002) falha. No entanto, existem exemplares do problema para os quais o desempenho do algoritmo proposto não é bom. Espera-se, numa investigação futura, o desenvolvimento de novos e melhores limitantes (inferiores) e o desenvolvimento de métodos de redução

do problema pela agregação de tarefas visando superar estas dificuldades do algoritmo proposto.

Com base no esquema de enumeração proposto, várias heurísticas podem ser construídas, visando obter um bom limitante superior para o problema. Dentre as várias possibilidades de métodos heurísticos podemos citar:

Método 1: a cada nó da árvore de enumeração manter apenas os melhores “s” ramos;

Método 2: a cada nó da árvore de enumeração manter somente os melhores “s” ramos, mas, ao efetuar o *backtrack*, escolher aleatoriamente o próximo nó a ser ramificado;

Método 3: a cada nó da árvore de enumeração manter no máximo os “s” ramos com o menor valor para o número de trocas de ferramentas;

Método 4: a cada nó da árvore de enumeração manter no máximo os “s” ramos com o menor valor para o número de

trocas de ferramentas, mas, ao efetuar o *backtrack*, escolher aleatoriamente o próximo nó a ser ramificado;

Método 5: a cada nível da árvore de enumeração, manter no máximo “t” nós com o menor valor para o número de trocas de ferramentas, escolhidos aleatoriamente.

Evidentemente, o esforço computacional e a qualidade da solução obtida em cada um destes métodos heurísticos dependem dos valores de “s” ou “t”. Quanto maiores forem estes valores, melhores serão as soluções em termos de qualidade, mas o esforço computacional também irá aumentar, podendo chegar a ser proibitivo. Alguns resultados obtidos com estes métodos heurísticos, considerando $s = 3$ e $t = 3000$ podem ser encontrados em Yanasse et al., 2008.

A ideia do esquema de enumeração com ordenamento parcial pode ser estendida a outros problemas combinatórios e certamente também será objeto de investigação futura.

An enumeration algorithm based on partial ordering to solve the minimization of tool switches problem

Abstract

In the minimization of tool switches problem we seek to determine a sequence to process a set of jobs so that the number of tool switches required is minimized. In this work we propose an algorithm to solve this problem based on partial ordered sequences of jobs. An optimal sequence is obtained expanding the partial sequences enumerated. Computational test results are presented.

Keywords: *Minimization of tool switches problem. Partial ordering. Lower bound.*

Referências bibliográficas

- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ, A. J. A Memetic Algorithm for the Tool Switching Problem. In: INTERNATIONAL WORKSHOP ON HYBRID METAHEURISTICS, 5., 2008, Málaga. **Proceedings...** Berlin: Springer-Verlag, 2008. p. 190-202. Lecture Notes in Computer Science.
- AVCI, S.; AKTURK, M. S. Tool magazine arrangement and operations sequencing on CNC machines. **Computers and Operations Research**, v. 23, n. 11, p. 1069-1081, 1996.
- BALAKRISHNAN, N.; CHAKRAVARTY, A. K. Opportunistic retooling of a flexible machine subject to failure. **Naval Research Logistics**, v. 48, n. 1, p. 79-97, 2001.
- BARD, J. F. A Heuristic for Minimizing the Number of Tool Switches on Flexible Machine. **IIE Transactions**, v. 20, n. 4, p. 382-391, 1988.
- BELADY, L. A. A study of replacement algorithms for virtual storage computers. **IBM Systems Journal**, v. 5, n. 2, p. 78-101, 1966.
- BLAZEWICZ, J.; FINKE, G. Scheduling with resource management in manufacturing systems. **European Journal of Operational Research**, v. 76, p. 1-14, 1994.
- CRAMA, Y. Combinatorial optimization models for production scheduling in automated manufacturing systems. **European Journal of Operational Research**, v. 99, n. 1, p. 136-153, 1997.
- CRAMA, Y.; KOLEN, A. W. J.; OELERMANS, A. G.; SPIEKSMAN, F. C. R. Minimizing the number of tool switches on a flexible machine. **International Journal of Flexible Manufacturing Systems**, v. 6, p. 33-54, 1994.
- CRAMA, Y.; MOONEN, L. S.; SPIEKSMAN, F. C. R.; TALLOEN, E. The tool switching problem revisited. **European Journal of Operational Research**, v. 182, n. 2, p. 952-957, 2007.
- FOLLONIER, J. P. Minimization of the number of tool switches on a flexible machine. **Belgian Journal of Operations Research, Statistics and Computer Science**, v. 34, p. 55-72, 1994.
- GAREY, M.; JOHNSON, D. S. **Computers and Intractability**. New York: Freeman, 1979.
- GHIANI, G.; GRIECO, A.; GUERRIERO, E. An exact solution to the TLP problem in an NC machine. **Robotics and Computer-Integrated Manufacturing**, v. 23, n. 6, p. 645-649, 2007.

- GRAY, A. E.; SEIDMANN, A.; STECKE, K. E. A synthesis of decision models for tool management in automated manufacturing. **Management Science**, v. 39, n. 5, p. 549-567, 1993.
- HERTZ, A.; WIDMER, M. An improved tabu search approach for solving the job shop scheduling problem with tooling constraints. **Discrete Applied Mathematics**, v. 65, n. 1-3, p. 319-345, 1993.
- HERTZ, A.; LAPORTE, G.; MITTAZ, M.; STECKE, K. Heuristics for minimizing tool switches when scheduling part types on a flexible machine. **IEE Transactions**, v. 30, n. 8, p. 689-694, 1998.
- KIRAN, A. S.; KRASON, R. J. Automated tooling in a flexible manufacturing system. **Industrial Engineering**, v. 20, p. 52-57, 1988.
- KNUUTILA, T.; HIRVIKORPI, M.; JOHNSON, M.; NEVALAINEN, O. **Grouping PCB assembly jobs with typed component feeder units**. Finland: Turku Centre for Computer Science, 2002. Technical report 460.
- LAPORTE, G.; SALAZAR, J. J.; SEMET, F. Exact Algorithms for the Job Sequencing and Tool Switching Problem. **IIE Transactions**, v. 36, n. 1, p. 37-45, 2004.
- McGEOCH, L. A.; SLEATOR, D. D. A strongly competitive randomized paging algorithm. **Algorithmica**, v. 6, n. 1-6, p. 816-825, 1991.
- OERLEMANS, A. G. **Production planning for flexible manufacturing systems**. 1992. Tese (Doutorado) - University of Limburg, Maastricht, Holland.
- PRIVAUULT, C.; FINKE, G. k-server problems with bulk requests: an application to tool switching in manufacturing. **Annals of Operations Research**, v. 96, n. 1-4, p. 255-269, 2000.
- SALONEN, K.; RADULY-BAKA, C.; NEVALAINEN, O. S. A note on the tool switching problem of a flexible machine. **Computers and Industrial Engineering**, v. 50, n. 4, p. 458-465, 2006.
- SODHI, M. S.; ASKIN, R. G.; SEN, S. Multiperiod tool and production assignment in flexible manufacturing systems. **International Journal of Production Research**, v. 32, n. 6, p. 1281-1294, 1994.
- TANG, C. S.; DENARDO, E. V. Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches. **Operations Research**, v. 36, n. 5, p. 767-777, 1988.
- VANHOP, N. The Tool-Switching Problem With Magazine Capacity and Tool Size Constraints. **IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans**, v. 35, n. 5, p. 617-628, 2005.
- YANASSE, H. H.; LAMOSIA, M. J. P. An application of the generalized travelling salesman problem: the minimization of tool switches problem. In: INTERNATIONAL ANNUAL SCIENTIFIC CONFERENCE OF THE GERMAN OPERATIONS RESEARCH SOCIETY, 2005, Bremen. **Proceedings...**
- YANASSE, H. H.; SENNE, E. L. F.; RODRIGUES, R. C. M. Performance evaluation of some heuristics based on partial ordering for solving the minimization of tool switches problem. In: ALIO/EURO WORKSHOP ON APPLIED COMBINATORIAL OPTIMIZATION, 6., 2008, Buenos Aires. **Proceedings...** Buenos Aires: Universidad de Buenos Aires, 2008. p. 1-6.
- ZHOU, B. H.; XI, L. F.; CAO, Y. S. A beam-search-based algorithm for the tool switching problem on a flexible machine. **International Journal of Advanced Manufacturing Technology**, v. 25, n. 9-10, p. 876-882, 2005.

Sobre os autores

Horacio Hideki Yanasse

Instituto Nacional de Pesquisas Espaciais – INPE/LAC
Avenida dos Astronautas, 1758, Jardim da Granja, CEP 12227-010, São José dos Campos, São Paulo, Brasil
e-mail: horacio@lac.inpe.br

Rita de Cássia Meneses Rodrigues

Instituto Nacional de Pesquisas Espaciais – INPE/LAC
Avenida dos Astronautas, 1758, Jardim da Granja, CEP 12227-010, São José dos Campos, São Paulo, Brasil
e-mail: rita@lac.inpe.br

Edson Luiz França Senne

Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista Júlio de Mesquita Filho – UNESP
Av. Ariberto Pereira da Cunha, 333, Portal das Colinas, CEP 12516-410, Guaratinguetá, SP, Brasil
e-mail: elfsenne@feg.unesp.br

Agradecimentos. Agradecemos ao Prof. J.J. Salazar González, que gentilmente disponibilizou o programa gerador de exemplares para o MTSP. Este trabalho contou com financiamentos parciais do CNPq e da FAPESP.

Recebido: 27/5/2008

Aceito: 13/5/2009