UM ALGORITMO POLINOMIAL PARA O CASO DE ÁRVORES DO PROBLEMA DE MINIMIZAÇÃO DE TROCAS DE FERRAMENTAS

Horacio Hideki Yanasse
Instituto Nacional de Pesquisas Espaciais - INPE
Avenida dos Astronautas, 1.758 - São José dos Campos, SP, Brasil
horacio@lac.inpe.br

RESUMO

No problema de minimização de trocas de ferramentas procura-se por uma sequência de processamento de um conjunto de tarefas em uma máquina flexível de manufatura de modo que o número total requerido de trocas de ferramentas seja o menor possível. No problema de minimização de pilhas abertas procura-se por uma sequência de corte de um conjunto de padrões em uma serra de modo que o número máximo de pilhas abertas seja o menor possível. Cada tipo de item cortado abre uma pilha que permanece aberta até que o último item daquele tipo seja cortado. Apresenta-se um algoritmo polinomial para o caso especial do problema de minimização de trocas de ferramentas em que seu grafo MOSP correspondente é uma árvore. O novo algoritmo, cuja complexidade computacional é limitado por $O(N^3)$, em que N é o número de tarefas, é uma adaptação do algoritmo para árvores do problema de minimização de pilhas abertas.

PALAVRAS CHAVE. Minimização de troca de ferramentas, Minimização de pilhas abertas, Decomposição.

Área principal: Otimização Combinatória

ABSTRACT

In the minimization of tool switches problem we seek a sequence to process a set of jobs in a flexible manufacturing machine such that the total number of required tool switches is minimized. In the minimization of open stacks problem we seek a sequence to cut a set o patterns in a saw machine such that the maximum number of open stacks is minimized. Each item type cut opens a stack that remains open until the last piece of that type is cut. We present a polynomial algorithm for the special case of the minimization of tool switches problem when the corresponding MOSP graph is a tree. The new algorithm, which has computational complexity of $O(N^3)$,, where N is the number of jobs, is an adaptation of the algorithm for trees of the minimization of open stacks problem.

KEYWORDS. Minimization of tool switches, Minimization of open stacks, Decomposition.

Main area: Combinatorial optimization

1. Introdução

No problema de minimização de trocas de ferramentas (MTSP, do inglês, Minimization of Tool Switches Problem), procura-se por uma sequência de processamento de um conjunto de tarefas em uma máquina flexível de manufatura de modo que o número requerido de trocas de ferramentas seja o menor possível. Cada tarefa requer um conjunto de ferramentas para ser processada. A caixa de ferramentas da máquina tem capacidade limitada, não sendo possível colocar simultaneamente todas as ferramentas necessárias para processar todas as tarefas. Caso haja espaço para colocar todas as ferramentas o problema é trivial de ser resolvido.

Neste trabalho, apresentamos um algoritmo polinomial para o caso especial do MTSP, em que o grafo MOSP (veja Yanasse, 1997a) correspondente é uma árvore. O algoritmo desenvolvido explora o resultado apresentado em Yanasse (2011). Desconhecemos trabalhos na literatura em que casos polinomiais do MTSP tenham sido apresentados e/ou desenvolvidos.

O leitor interessado poderá consultar o trabalho de Yanasse (2011) em que é feita uma revisão de trabalhos sobre o MTSP e sobre alguns trabalhos sobre o problema de minimização de pilhas abertas (MOSP, do inglês, Minimization of Open Stacks Problem). No problema de minimização de pilhas abertas procura-se por uma sequência de corte de um conjunto de padrões em uma serra de modo que o número máximo de pilhas abertas seja o menor possível. Cada tipo de item cortado abre uma pilha que permanece aberta até que o último item daquele tipo seja cortado.

No novo algoritmo desenvolvido é explorada a similaridade do MTSP com o MOSP. Vários resultados e procedimentos apresentados em Yanasse (1997b) também são utilizados.

O MTSP e o MOSP são equivalentes, por exemplo, no caso em que se tem um exemplar do MTSP tendo C como a capacidade da caixa de ferramentas na máquina flexível de manufatura, e o valor ótimo do MOSP correspondente é também igual a C. Então uma sequência ótima deste MTSP é também ótima para o MOSP e vice-versa. Se o valor ótimo para o MOSP correspondente for menor do que C então uma sequência ótima deste MOSP é também ótima para o MTSP mas não necessariamente o inverso. Se o valor ótimo para o MOSP correspondente for maior do que C então nada se pode dizer com relação às sequências ótimas do MOSP e MTSP correspondentes.

Cabe observar que o MTSP também aparece no contexto de pilhas abertas. Por exemplo, se no caso de pilhas abertas tivermos um limite máximo no número de pilhas ao redor da serra, e se tivermos que remover uma pilha aberta para dar espaço a uma outra pilha que está sendo cortada agora e, depois, for necessário trazê-la de volta ao redor da serra para ser completada, então o MTSP surge se quisermos minimizar este remanejamento de pilhas ao redor da serra.

Na próxima seção apresentamos o algoritmo proposto para o MTSP no caso de árvores. Faremos uma apresentação sucinta do algoritmo fazendo referência a partes de algoritmos descritas em Yanasse (1997b). Não reproduziremos os procedimentos para árvores ou casos especiais de árvores para o MOSP neste trabalho pois são extensas e estão publicadas. Na seção 3 mostramos que este algoritmo determina uma solução ótima para o MTSP; na seção 4, a complexidade computacional do algoritmo desenvolvido é calculada e, na seção 5 são apresentados os comentários finais.

2. Algoritmo para resolução do MTSP no caso de árvores

Como já indicado em trabalhos anteriores (veja, por exemplo, Yanasse, 1997c), podemos obter uma solução ótima para o MTSP resolvendo o MOSP correspondente no caso em que o número máximo de pilhas abertas for menor ou igual à capacidade da caixa de ferramentas da máquina. Assim, para todos os casos em que temos uma árvore para o grafo MOSP correspondente ao MTSP, e o número máximo de pilhas abertas for menor ou igual à capacidade da caixa de ferramentas da máquina, o algoritmo para árvores do MOSP resolve também o MTSP em tempo polinomial (limitado por $O(n^2)$, em que n é o número de tarefas, veja Yanasse, 1997b). Usualmente, de início, não sabemos o número máximo de pilhas abertas que serão abertas. Assim, seria preciso resolver o MOSP correspondente antes para só depois saber se a solução encontrada também é ótima para o MTSP.

Desenvolvemos a seguir um algoritmo que resolve o MTSP independentemente de sabermos se o valor ótimo do MOSP correspondente é maior do que a capacidade C da caixa de ferramentas.

Consideremos a seguir alguns casos especiais de árvores que vão "evoluindo" em termos de pilhas abertas.

Caso (a) árvore com um único nó, que denotaremos por ST-1
A resolução deste problema é trivial.
O número 1 do ST-1 indica o número máximo de pilhas abertas para este grafo. Qualquer MTSP tipo ST-1 com $C \geq 1$ pode ser resolvido trivialmente.

Caso (b) árvore em que todos os nós tem grau 2 ou menos, e a árvore não é um ST-1.
Consideraremos dois casos especiais de árvores do caso (b):
(b.1) árvore tipo (b) com 2 nós. Denotaremos esta árvore de ST-2a
(b.2) árvore tipo (b) com 3 ou mais. Denotaremos esta árvore de ST-2b.
O procedimento "WALK" apresentado em Yanasse (1997b) resolve otimamente as árvores tipo ST-2a e ST-2b abrindo 2 pilhas. Qualquer MTSP tipo ST-2a ou tipo ST-2b com $C \geq 2$ pode ser resolvido utilizando o procedimento "WALK". C não pode ser menor do que 2, pois admitimos que a capacidade da máquina é maior ou igual ao número de ferramentas que qualquer tarefa do problema necessita.

Utilizando o pré-processamento 3 apresentado em Yanasse e Senne (2010), para fins de redução do problema a ser resolvido, qualquer árvore do tipo ST-2b com mais de 3 nós pode ser reduzido a uma árvore ST-2b com exatamente 3 nós. Desta forma, para fins de exposição, consideraremos apenas árvores do tipo ST-2b com exatamente 3 nós.

Caso c) árvore tipo estrela, ou seja, uma árvore em que todos os nós tem grau 2 ou menos e exatamente um nó tem grau 3 ou mais.
Sem perda de generalidade, seja k o nó na árvore que tem o grau igual a 3 ou mais. Consideraremos os seguintes casos especiais de árvores do caso (c):
(c.1) árvore tipo (c) em que ao se retirar o nó k e todos os arcos adjacentes a ele, a floresta resultante é composta de árvores do tipo ST-1 e/ou no máximo 2 árvores do tipo ST-2a ou ST-2b. Denotaremos esta árvore de ST-2c.
(c.2) árvore tipo (c) em que ao se retirar o nó k e todos os arcos adjacentes a ele, a floresta resultante é composta de árvores do tipo ST-1 e/ou no mínimo 3 árvores do tipo ST-2a ou ST-2b. Denotaremos esta árvore de ST-3.

O algoritmo "TREE2" apresentado em Yanasse (1997b) resolve otimamente as árvores tipo ST-2c e ST-3 abrindo, respectivamente, 2 pilhas e 3 pilhas.
Qualquer MTSP tipo ST-2c com $C \geq 2$ pode ser resolvido utilizando o procedimento "TREE2". Qualquer MTSP tipo ST-3 com $C \geq 3$ pode ser resolvido utilizando o procedimento "TREE2".
Para resolver o MTSP tipo ST-3 com $C = 2$, seja $m \geq 3$ o número de árvores do tipo ST-2a ou ST-2b na floresta resultante ao se retirar o nó k da árvore tipo (c) e todos os arcos adjacentes a ele. Vamos aplicar agora a idéia apresentada no teorema formulado em Yanasse (2011) e resolver o problema decompondo-o em partes.

Procedimento Decomposição-ST-3

A árvore tipo ST-3 é decomposta da seguinte forma:

subgrafo-parte-1) árvore original retirando dele (m-2) das árvores do tipo ST-2a ou ST-2b da floresta resultante ao se retirar o nó k da árvore tipo (c) e todos os arcos adjacentes a ele, e também os arcos que o unem ao nó k.

subgrafo-parte-2) árvore original retirando dele todas os arcos do subgrafo-parte-1.

Observe que o subgrafo-parte-1 é agora uma árvore do tipo ST-2c, que pode ser resolvido pelo algoritmo "TREE2".
Observe que permanecem no subgrafo-parte-2 as (m-2) árvores do tipo ST-2a ou ST-2b e também os arcos que as unem ao nó k. O subgrafo-parte-2 pode ser do tipo ST-2c ou ST-3. Se for do tipo ST-2c ele pode ser resolvido pelo algoritmo "TREE2". Se for do tipo ST-3, o procedimento Decomposição-ST-3 pode ser usado novamente.

Aplicando-se o procedimento Decomposição-ST-3 de forma recursiva, obtém-se uma solução ótima para o MTSP tipo ST-3.

Observe que nesta decomposição, estamos propondo um sequenciamento em que a ferramenta k que foi utilizada para processar alguma tarefa está saindo da caixa de ferramentas apesar de existirem outras tarefas ainda a processar que a utilizam. Mas, devido às limitações da capacidade da caixa de ferramentas isto é inevitável.

A mesma idéia aplicada para ST-3 com C = 2 é estendida a seguir para o caso mais geral. Seja uma árvore do tipo ST-n com $n > 3$ e seja $C = n - 1$. Uma árvore do tipo ST-n é uma árvore em que temos um nó k com grau 3 ou mais e que ao se retirar este nó k e todos os arcos adjacentes a ele, a floresta resultante é composta de árvores do tipo ST-1, ST-2a, ST-2b, ST-2c, ST-(n-2) e no mínimo 3 árvores do tipo ST-(n-1).
Seja $m \geq 3$ o número de árvores do tipo ST-(n-1) na floresta resultante ao se retirar o nó k da árvore tipo ST-n e todos os arcos adjacentes a ele. A árvore tipo ST-n é decomposta da seguinte forma:

Procedimento Decomposição-ST-n

subgrafo-parte-1) árvore original retirando dele (m-2) das árvores do tipo ST-(n-1) da floresta resultante ao se retirar o nó k da árvore tipo ST-n e todos os arcos adjacentes a ele, e também os arcos que o unem ao nó k.

subgrafo-parte-2) árvore original retirando dele todas os arcos do subgrafo-parte-1.

Observe que o subgrafo-parte-1 é agora uma árvore do tipo ST-(n-1), que pode ser resolvido pelo algoritmo "TREE3" (veja Yanasse, 1997b).

O subgrafo-parte-2 pode ser do tipo ST-(*n*-1) ou do tipo ST-*n*. Se for do tipo ST-(*n*-1) ele pode ser resolvido pelo algoritmo "TREE3". Se for do tipo ST-*n*, o procedimento Decomposição-ST-*n* pode ser usado novamente.

Aplicando-se o procedimento Decomposição-ST-*n* de forma recursiva, obtém-se uma solução ótima para o MTSP tipo ST-*n*.
Baseado nas idéias apresentadas anteriormente, o algoritmo para o MTSP para árvores é apresentado a seguir. Ele é uma adaptação do algoritmo TREE3 de Yanasse (1997b).

Algoritmo MTSP_TREE para árvores.

Passo 0) Aplique os passos do algoritmo TREE3 até que o valor do score de algum nó atinja C.

Passo 1) Continue aplicando os passos do algoritmo TREE3. A cada atualização do score de um nó, digamos k, compare-o com o valor de C.

Passo 2) Se o "score" de k ultrapassar C, aplique a Decomposição-ST-C+1)* (uma variação da Decomposição-ST-(C+1), apresentado a seguir), caso contrário, vá ao passo 1.

Passo 3) Para o subgrafo-parte-1 resultante da decomposição aplique o algoritmo TREE3. No subgrafo-parte-2, atualize o score do nó k e retorne ao Passo 2.

Considere uma árvore, que denotaremos por ST-(*n+1*)*, em que temos um nó k de grau maior ou igual a 3 e ao eliminarmos este nó e todos os arcos incidentes nele, resulta em uma floresta com pelo menos $m \geq 3$ árvores do tipo ST-*n* e, possivelmente, mais algumas outras árvores dos tipos ST-1, ST-2a, ST-2b, ST-2c, ...ST-(*n*-1) e mais algumas árvores cujos tipos não estão identificados (estas, se sequenciadas independentemente, cada uma delas pode resultar em um número de pilhas maior, menor ou igual a *n*).

Introduzimos a seguir uma decomposição para uma árvore do tipo ST-*n**.

Procedimento Decomposição-ST-*n**

subgrafo-parte-1) árvore original retirando dele (*m*-2) das árvores do tipo ST-(*n*-1) da floresta resultante ao se retirar o nó k da árvore tipo ST-*n** e todos os arcos adjacentes a ele, e também os arcos que o unem ao nó k. Retirar também, todas as outras árvores, cujos tipos ainda não estão identificados.

subgrafo-parte-2) árvore original retirando dele todas os arcos do subgrafo-parte-1.

Observe que o subgrafo-parte-1 é agora uma árvore do tipo ST-*n*, que pode ser resolvido pelo algoritmo "TREE3".

3. Algoritmo MTSP_TREE para o caso de árvores é ótimo

Nesta seção damos uma indicação de como se demonstra que o algoritmo MTSP-TREE para o caso de resolução de árvores fornece uma solução ótima para o problema.
No grafo MOSP, arcos correspondem a tarefas e os nós a ferramentas. No que se segue, vamos indistintamente nos referir a arcos como tarefas e a nós como ferramentas. Seguindo-se o sequenciamento das tarefas do MTSP temos em correspondência um percorrimento de arcos no grafo MOSP correspondente. Cada arco percorrido implica que os dois nós (ou ferramentas) os quais ele é adjacente estão necessariamente na caixa de ferramentas da máquina. Ferramentas que são retiradas da máquina podem estar em uma das seguintes condições:
1) todos os arcos incidentes a ele já foram percorridos
2) existe pelo menos um arco incidente a ele que não foi percorrido.
No caso (1), a ferramenta retirada da máquina pode ser excluída do problema (não existe nenhuma outra tarefa que a utiliza), consequentemente, as demais tarefas a serem processadas independem desta ferramenta.
No caso (2), a ferramenta retirada da máquina deve retornar futuramente à caixa de ferramentas pois existe pelo menos alguma outra tarefa que a necessita para ser processada.
Uma ferramenta retirada da máquina no caso (2) equivale a uma duplicação do nó correspondente no grafo MOSP e uma redistribuição dos arcos incidentes ao nó original para os nós duplicados. Seja v a solução MOSP de um problema. Se $v \leq C$, a solução obtida do MOSP é também ótima para o MTSP. Se $v > C$, pela duplicação mínima possível de nós e rearranjos dos arcos incidentes a ele, desejamos encontrar um novo grafo cuja solução ótima para o problema de minimização de pilhas abertas seja C.
Pode-se provar utilizando o resultado apresentado em Yanasse (2009) que para o caso de árvores, a decomposição proposta ST-3, ST-n com $n > 3$ e ST-n* realizam esta duplicação mínima.

4. Complexidade computacional do algoritmo MTSP_TREE

O algoritmo MTSP_TREE segue basicamente os mesmos passos do algoritmo TREE3 cuja ordem é $O(N^2)$ em que N é o numero total de tarefas. O número máximo de aplicações do procedimento de decomposição é limitado por O(N) uma vez que em cada decomposição o número de arcos do subgrafo-parte-1 é no mínimo um número inteiro positivo maior ou igual a 5. Portanto, o algoritmo MTSP-TREE é limitado por $O(N^3)$.

5. Comentários Finais

Neste trabalho apresentamos um algoritmo polinomial para uma classe especial de problemas de minimização de trocas de ferramentas em que o grafo MOSP correspondente é uma árvore. De nosso conhecimento, trata-se do primeiro algoritmo polinomial proposto na literatura para qualquer classe de problemas de minimização de trocas de ferramentas.
O grafo MOSP correspondente a uma árvore pode aparecer, por exemplo, em ambientes de corte em que o número de facas de corte é limitado a 3. É o caso que ocorre, por exemplo, em algumas indústrias de caixas papelão (veja Lins, 1989).
Outros algoritmos polinomiais podem ser resolvidos para outras classes especiais de problemas de minimização de trocas de ferramentas utilizando-se idéias similares às utilizadas neste trabalho e, também, o resultado apresentado em Yanasse (2011).

Agradecimentos: Este trabalho foi parcialmente financiado pelo CNPq e pela FAPESP.

Referências

Garey; M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

Bard, J.F. (1988), A Heuristic for Minimizing the Number of Tool Switches on Flexible Machine. *IIE Transactions*, 20, 382-391.
Crama, Y.; Kolen, A.W.J.; Oelermans, A.G.; Spieksma, F.C.R. (1994), Minimizing the number of tool switches on a flexible machine, *International Journal of Flexible Manufacturing Systems*, 6, 33-54.
Hertz, A.; Laporte, G.; Mittaz, M.; Stecke, K. (1998), Heuristics for minimizing tool switches when scheduling part types on a flexible machine. *IEE Transactions*, 30, 689-694.
Laporte, G.; Salazar, J.J.; Semet, F. Exact Algorithms for the Job Sequencing and Tool Switching Problem. *Les Cahiers du GERAD*, G-2002-40, July 2002.
Linhares, A.; Yanasse, H.H. (2002), Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers and Operations Research*, 29(12):1759-1772.
Lins, S., (1989) Traversing trees and scheduling tasks for duplex corrugator machines, *Pesquisa Operacional* 9(1):40-54.
Tang, C.S.; Denardo, E.V. (1988), Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches, *Operations Research*, 36, 767-777.
Yanasse, H.H. (1996), Minimization of open orders - polynomial algorithms for some special cases, *Pesquisa Operacional*, 16(1):1-26.
Yanasse, H.H. (1997a), A transformation for solving a pattern sequencing problem in the wood cut industry. *Pesquisa Operacional*, 17(1):57-70.
Yanasse, H.H. *An* exact algorithm for the tree case of the minimization of open orders problem. *XXIX Simpósio Brasileiro de Pesquisa Operacional* (SBPO), Salvador, BA, 22 a 24 de outubro de 1997b.
Yanasse, H.H. On a pattern sequencing problem to minimize the maximum number of open orders, *European Journal of Operational Resesarch* 100:454-463, 1997c.
Yanasse, H.H. Um novo limitante inferior para o problema de minimização de trocas de ferramentas – XLI Simpósio Brasileiro de Pesquisa Operacional, Porto Seguro, BA, 01-04 de setembro, 2009. Livro de Resumos, p. 72. Anais em CD, p..2841-2848, ISSN/ISBN: 15181731.
Yanasse, H.H. Sobre o problema de minimização de trocas de ferramentas: um caso especial. XLIII Simpósio Brasileiro de Pesquisa Operacional, Ubatuba, SP, 15 a 18 de agosto de 2011. Livro de resumos, p. 107. Anais do XLIII SBPO, p. 2117-2124, em CD.
Yanasse, H.H.; Lamosa, M.J.P. An application of the generalized travelling salesman problem: the minimization of tool switches problem, *Operations Research 2005 - International Annual Scientific Conference of the German Operations Research Society*, Bremen, Germany, 7-9 September, 2005, Program, p. 90.
Yanasse, H.H.; Rodrigues, R.C.M. Senne, E.L.F. (2009), Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. *Gestão e Produção*, 16(3): 370-381.
Yanasse, H.H.; Senne, E.L.F. (2010), The minimization of open stacks problem: a review of some properties and their use in pre-processing operations. *European Journal of Operational Research*, 203(3): 559-567. Doi: 10.1016/j.ejor.2009.09.017.