

Resolução do Problema de Minimização de Troca de Ferramentas como um Problema de Fluxo em Rede

Maria José Pinto
Instituto Nacional de Pesquisas Espaciais
INPE/LAC
e-mail: maju@lac.inpe.br

Horacio Hideki Yanasse
Instituto Nacional de Pesquisas Espaciais
INPE/LAC
e-mail: horacio@lac.inpe.br

Resumo

Neste trabalho, formulamos o problema de minimização de troca de ferramentas como um problema de fluxo em rede a custo mínimo com algumas restrições adicionais. Discussões sobre a implementação desta formulação e alguns resultados parciais são apresentados.

1. Introdução

O problema de minimização de troca de ferramentas – MTSP (do inglês *Minimization of the Number of Tool Switches Problem*) pode ser definido da seguinte forma: considere que se deseja seqüenciar N tarefas, que necessitam de M ferramentas para o seu processamento, em uma máquina que possui uma caixa de ferramentas com capacidade limitada C . Levando em consideração que, em geral, $M > C$, surge a necessidade de haver trocas de ferramentas para o processamento das tarefas. Com isto, o MTSP consiste em obter um seqüenciamento das tarefas que resulte no menor número de trocas de ferramentas, visto que a cada troca realizada, a produção da máquina fica comprometida devido à interrupção necessária para a realização das trocas.

Tang e Denardo [8] propuseram uma formulação de programação inteira para o MTSP. Devido ao fato da resolução deste modelo, utilizando um código comercial, não ter obtido bons resultados, métodos heurísticos foram sugeridos pelos autores.

Recentemente, Pinto e Yanasse [5] realizaram alguns testes computacionais para resolver o MTSP, utilizando o mesmo modelo, usando o software comercial CPLEX 6.5 em uma estação de trabalho Sun UltraSPARC II modelo Ultra 60. Em um conjunto de vinte exemplos de tamanhos variados, com M e N no intervalo de [8, 10] e C no intervalo de [3,8], o tempo médio para o CPLEX encontrar uma solução foi de 4,87 minutos. Os tempos computacionais observados foram bastante diversificados,

variando de 0,32 minutos ($M = 10$, $N = 9$, $C = 8$) a 32,21 minutos ($M = 9$, $N = 10$, $C = 3$). Esta variação ocorreu também considerando problemas com os mesmos parâmetros. Por exemplo, um outro problema de tamanho $M = 9$, $N = 10$ e $C = 3$ chegou à solução ótima em menos de 2 minutos. Isto parece indicar que o tempo computacional para resolução do MTSP com o CPLEX 6.5 parece ser bastante dependente dos dados do problema. Ao aumentar o tamanho dos exemplos considerados o tempo computacional requerido para sua solução se tornou consideravelmente grande. Devido a este fato, outras formas de resolver o problema foram considerados. Um método testado, que não apresentou resultados satisfatórios, consistiu em utilizar relaxação lagrangiana [4]. O método de Benders [7] foi também implementado e apresentou-se inviável devido ao alto tempo computacional para obtenção da solução.

Considerando que a capacidade da máquina é usualmente fixa na prática, que uma grande variedade no tempo para encontrar uma solução para exemplos de mesmo tamanho não é uma característica desejável, que os tamanhos dos exemplos considerados em Pinto e Yanasse [5] foram pequenos e que a quantidade de esforço computacional necessário para resolvê-los foi razoável, propõe-se, neste trabalho, uma forma alternativa para se resolver o MTSP, que consiste em formulá-lo como um problema de fluxo em rede a custo mínimo, com restrições adicionais. Espera-se, com isto, reduzir os inconvenientes encontrados.

O interesse em resolver o MTSP está relacionado ao trabalho apresentado em Pinto e Yanasse [6] que propõem uma formulação para integrar o problema de corte de estoque e o problema de sequenciamento dos padrões que, geralmente, são tratados de forma independente na literatura. De maneira geral, o problema de corte consiste em obter os padrões de corte (como os objetos são cortados) e suas respectivas frequências para atender a demanda dos itens. Associado a este problema está o problema de sequenciamento que consiste em seqüenciar os padrões gerados pelo problema de corte com o objetivo

de, por exemplo, minimizar o número de pilhas abertas durante o processo de corte (MOSP, do inglês *Minimization of Open Stacks Problem*). Yanasse [9] apresenta uma formulação matemática para o MOSP, baseada na formulação proposta por Tang e Denardo [8] para o MTSP, relacionando tarefas com padrões a serem sequenciados e ferramentas com os itens que compõem o padrão de corte. Neste trabalho é visto que, caso a solução para o MTSP for igual a $M-C$, a solução obtida indicará um sequenciamento dos padrões onde no máximo C pilhas serão abertas. Pinto e Yanasse [6] propuseram, então, resolver o modelo integrado utilizando um método de decomposição que relaxa as restrições que ligam os dois problemas. Com isto, o problema de corte é resolvido e os padrões são sequenciados utilizando a formulação para o MTSP. Caso a solução para o MTSP for maior a $M-C$, o objetivo do sequenciamento não poderá ser satisfeito, sendo necessária a geração de novos padrões de corte. Com isto, a motivação para este estudo consiste em encontrar um método que resolva o MTSP de forma eficiente, visto a possibilidade de ser necessário resolvê-lo um grande número de vezes até se chegar à solução ótima do problema integrado.

A formulação da abordagem proposta utilizando fluxo em rede a custo mínimo para resolução do MTSP será apresentada na seção 2. Os resultados parciais obtidos e discussões sobre a implementação desta formulação estão apresentados na seção 3. Algumas considerações finais são feitas na seção 4.

2. MTSP como problema de fluxo mínimo

Considerando a capacidade da máquina fixa, o número total de combinações das M ferramentas em conjuntos de C ferramentas será dado por $n = \binom{M}{C}$. Considere um grafo

não direcionado onde cada nó representa uma destas combinações. Pensando no MTSP, poderíamos considerar que o custo de ir de um nó i a um nó j deste grafo corresponde ao número de ferramentas que existem no nó i e que não pertencem ao nó j , ou seja, o número de trocas de ferramentas que ocorrerão de um nó para outro do grafo.

Sejam $J_1, J_2, \dots, J_N$ as N tarefas a serem processadas e $S_1, S_2, \dots, S_N$ o conjunto de todos os nós do grafo que possuem todas as ferramentas necessárias para processar as tarefas $J_1, J_2, \dots, J_N$, respectivamente. Considere ainda dois nós adicionais neste grafo: um nó origem O e um nó destino D . Os nós O e D estão conectados a todos os outros nós no grafo, os custos C_{Oj} e C_{jD} são nulos e os custos C_{jO} e C_{Dj} são infinitos, para $j = 1, 2, \dots, N$.

Com isto, o problema de fluxo em rede que desejamos resolver é mandar uma unidade de fluxo do nó O para o nó D a um custo mínimo, com restrições adicionais que

garantam que esta unidade de fluxo deve passar por pelo menos um nó de cada um dos conjuntos $S_1, S_2, \dots, S_N$. O nó visitado define as ferramentas que estarão presentes na máquina.

Observe que, se estamos em algum nó do conjunto S_i , as ferramentas necessárias para processar a tarefa J_i , $i = 1, 2, \dots, N$, estão disponíveis e, portanto, a tarefa J_i pode ser processada. Por esta razão, se ao menos um nó dos conjuntos $S_1, S_2, \dots, S_N$ é visitado, seremos capazes de processar todas as tarefas desejadas. Em outras palavras, qualquer solução factível para este problema de fluxo é uma solução factível para o problema de troca de ferramentas. Da mesma forma, qualquer solução factível para o problema de troca de ferramentas é um caminho factível para o problema de fluxo considerado. Portanto, qualquer solução de custo mínimo para o problema de fluxo fornecerá uma solução ótima para o MTSP, que consiste do nosso objetivo.

Considerando a variável x_{ij} como sendo o fluxo que vai do nó i ao nó j , obtemos o modelo de problema de fluxo em rede com restrições adicionais para o MTSP:

Função objetivo

$$\min \sum \sum C_{ij} x_{ij} \quad (1)$$

Restrições

$$\sum_j x_{ij} - \sum_k x_{ki} = 0 \quad i = 1, \dots, n \quad (2)$$

$$\sum_j x_{Oj} = 1 \quad (3)$$

$$\sum_k x_{kD} = 1 \quad (4)$$

$$\sum_{i \in S_k} \sum_{j \in S_k} x_{ij} = 1 \quad k = 1, 2, \dots, N \quad (5)$$

$$\sum_j \sum_k x_{jk} \leq N+1 \quad (6)$$

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} = 1 \quad \text{para todos os possíveis subconjuntos } Q; \quad (7)$$

$$x_{ij} = 0 \text{ ou } 1 \quad \text{para todos } i \text{ e } j. \quad (8)$$

A função objetivo (1) e as restrições (2), (3), (4) e (8) consistem de um problema tradicional de fluxo em rede a custo mínimo [1].

Para representar o MTSP, vimos a necessidade de adicionar as restrições (5), (6) e (7) ao modelo tradicional. As restrições (5) impõem que uma unidade de fluxo, de algum nó fora do conjunto S_k , deve entrar em algum nó pertencente ao conjunto S_k , $k = 1, 2, \dots, N$. As restrições (6) são consideradas pois, de acordo com (5), o fluxo tem que passar através de N conjuntos de nós e, com isto, deverão existir no máximo $(N+1)$ arcos onde o fluxo deve passar para formar um caminho de O a D passando em pelo menos um dos nós de cada conjunto S_k , $k = 1, 2, \dots, N$.

Infelizmente, as restrições (5) e (6) não são suficientes para prevenir ciclos, sendo necessária a introdução das restrições (7). Os subconjuntos Q são os subconjuntos possíveis que podem ser formados com os nós do grafo. Observe que as restrições (5) estão incluídas nas restrições (7), entretanto, estamos apresentando ambas desta forma por conveniência, devido às discussões apresentadas na próxima seção, onde serão feitas considerações a respeito desta formulação e de sua implementação.

3. Considerações sobre a implementação

A abordagem proposta apresenta algumas dificuldades onde, o primeiro problema que merece atenção é o número de nós n a serem considerados. Como n é o número de combinações possíveis de todas as ferramentas em conjuntos de tamanho C , n será “pequeno” quando C for pequeno ou quando C for relativamente grande em relação a M (o número total de ferramentas).

Em alguns casos podemos reduzir o número total de nós a serem considerados com um pré-processamento dos dados. Considere, por exemplo, que existam C ferramentas na máquina mas estas ferramentas não sejam suficientes para processar qualquer uma das tarefas restantes. Esta combinação de ferramentas nunca será utilizada numa solução prática do problema, não existindo assim, necessidade de ser considerada no problema, isto é, o nó correspondente a esta combinação pode ser eliminado do grafo. Para efeito de ilustração, considere o exemplo apresentado na Tabela 1 com $M = 5$, $N = 8$ e $C = 3$. Nesta tabela, o valor 1 indica as ferramentas necessárias para o processamento de cada tarefa e 0 as ferramentas não necessárias.

Para este exemplo, o número de nós a serem considerados no grafo seriam $\binom{5}{3} = 10$ nós e eles estão

listados na Tabela 2. Dentre estes nós, o nó 1, correspondendo a combinação 123, não necessita ser considerado pois, com estas ferramentas na máquina, nenhuma das 8 tarefas poderia ser processada. Isto pode também ser verificado na Tabela 3 onde os nós contidos nos conjuntos $S_1, S_2, \dots, S_8$ estão indicados.

Tabela 1. Exemplo ilustrativo, $M = 5$, $N = 8$ e $C = 3$

Tarefas	Ferramentas				
	1	2	3	4	5
1	1	0	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	0	0	1	1	0
5	1	0	0	0	1
6	0	0	1	0	1
7	0	0	0	1	1
8	1	1	0	1	0

Tabela 2. Nós no grafo para o exemplo da Tabela 1

Nós	Ferramentas				
	1	2	3	4	5
1	1	1	1	0	0
2	1	1	0	1	0
3	1	1	0	0	1
4	1	0	1	1	0
5	1	0	1	0	1
6	1	0	0	1	1
7	0	1	1	1	0
8	0	1	1	0	1
9	0	1	0	1	1
10	0	0	1	1	1

Tabela 3. Nós contidos nos conjuntos S_i do exemplo

Conjuntos	Nós									
	1	2	3	4	5	6	7	8	9	10
S_1		x		x		x				
S_2					x					
S_3		x					x		x	
S_4				x			x			x
S_5			x		x	x				
S_6					x			x		x
S_7						x			x	x
S_8		x								

Em alguns casos, também é possível reduzir o número de restrições (5). Considere o exemplo dado na Tabela 1. Da Tabela 3, vemos que o nó 2 é o único nó que satisfaz os requisitos para a tarefa J_8 . Portanto, em uma solução ótima, o fluxo necessariamente passará pelo nó 2. Sabendo que este nó fará parte do caminho na solução ótima, não há necessidade de incluir a restrição (5) para $k = 1$ e $k = 3$ pois elas serão satisfeitas automaticamente. Similarmente, o nó 5 é o único nó que satisfaz as condições de requisitos para a tarefa J_2 , portanto, não existe necessidade de incluir a restrição (5) para $k = 5$ e $k = 6$, pois elas serão automaticamente satisfeitas. Assim, podemos eliminar as linhas correspondentes a S_1, S_3, S_5 e S_6 , restando somente 4 conjuntos para serem analisados.

Com isto, o lado direito da restrição (6) se torna 5 (agora o fluxo deve ir do nó 0 ao nó D passando pelos 4 conjuntos S_2, S_4, S_7 e S_8) ao invés de 9. Em consequência, as combinações a serem consideradas nas restrições (7) também são reduzidas.

Vamos discutir agora as dificuldades em se resolver o problema de fluxo proposto. Suponha que o problema seja resolvido por algum método simplex especializado para resolver problema de fluxo com as restrições (8) substituídas pelas restrições de não negatividade das variáveis de decisão. Se a solução obtida for inteira, a solução é também ótima e o problema está resolvido. Entretanto, não existem garantias de que a solução obtida utilizando-se o problema relaxado seja inteira.

Cada solução obtida pelo método simplex determinará um caminho ou caminhos de O a D onde alguma

quantidade positiva de fluxo estará sendo transferida do nó O para o nó D. Se existir um único caminho com fluxo positivo, então o problema está resolvido (a quantidade deverá ser de 1 unidade, conforme as restrições). Caso contrário, valores fracionários de fluxo serão enviados utilizando-se caminhos diferentes de O a D.

Estamos confiantes que, na prática, um método tradicional do tipo *branch-and-bound*, que fixa os valores das variáveis de decisão para 0 ou 1, para algumas das variáveis fracionárias, rapidamente convergirá para uma solução inteira. Seguindo um caminho com fluxo positivo, propomos que a variável que deverá ser fixada em 0 ou 1 corresponda ao primeiro arco fracionário no caminho de O a D. Uma vez que isto é feito, todos os arcos subsequentes no caminho que têm valor fracionário serão forçados a serem inteiros (0 ou 1) pela restrição (2) de manutenção de fluxo em cada nó. Observe que fixar algum outro valor fracionário em 0 ou 1 forçaria apenas os arcos subsequentes no caminho a serem 0 ou 1, mas não necessariamente o mesmo ocorreria com todos os arcos antecedentes.

Infelizmente, por enquanto, não temos suporte teórico para garantir que esta convergência para uma solução inteira ocorrerá de maneira rápida. Alguns testes computacionais limitados serão realizados para verificar empiricamente se esta proposta é eficiente.

Vamos discutir agora as restrições (7) que previnem ciclos. Se n é o número de nós, o número total de possíveis combinações é 2^n . Para $n \leq 10$, $2^{10} \leq 1024$, isto é, o número de restrições a considerar é grande, mas não muito. Para valores maiores de n o número de restrições cresce rapidamente. Propomos resolver o problema sem estas restrições (somente com as restrições (5)) e apenas introduzi-las quando necessário. Se uma solução com um ciclo é encontrada, acrescentamos as restrições (7) correspondentes que evitam a ocorrência deste ciclo e o problema é novamente resolvido. Numa primeira tentativa considere restrições relativas à subconjuntos Q que são uniões dos conjuntos $S_1, S_2, \dots, S_N$. Estas restrições impõem que, para algum subconjunto dos conjuntos $S_1, S_2, \dots, S_N$, deve existir uma unidade de fluxo vindo do nó O ou de nós não pertencentes aos subconjuntos combinados. Estamos confiantes de que o número de restrições a serem consideradas até que uma solução ótima seja obtida não será grande em virtude da existência das restrições (6). Os testes computacionais limitados realizados confirmam este sentimento.

Os mesmos vinte exemplos gerados aleatoriamente com diferentes valores de M, N e C utilizados em Pinto e Yanasse [5], foram utilizados para a realização dos primeiros testes computacionais. Novamente, o software comercial CPLEX 6.5 foi utilizado em uma estação de trabalho Sun UltraSPARC II modelo Ultra 60. Na tabela 4, estão apresentados os resultados obtidos para resolução destes problemas utilizando a formulação por fluxo em

redes proposta, onde: **n1** representa o número de nós iniciais; **n2** o número de nós após o pré-processamento; **r** o número de restrições do tipo (7) que foram adicionadas e **T**, o tempo de resolução do problema (em minutos).

Tabela 4. Resultados parciais obtidos

Ex	M	N	C	n 1	n 2	r	T
1	9	10	3	84	66	4	0.18
2	10	10	5	252	34	4	0.05
3	9	10	6	126	29	7	0.07
4	10	9	5	252	30	4	0.04
5	9	10	4	126	76	10	0.72
6	10	8	6	210	6	2	0.01
7	10	10	4	210	84	14	1.17
8	9	10	3	84	24	6	0.04
9	10	9	6	210	118	7	0.80
10	9	10	4	126	35	6	0.07
11	10	10	4	210	25	9	0.06
12	8	10	4	70	11	5	0.02
13	10	10	7	120	31	2	0.04
14	10	9	3	120	26	7	0.06
15	10	8	3	120	84	37	6.76
16	9	10	5	126	27	4	0.03
17	8	10	6	28	18	6	0.04
18	10	8	5	252	107	13	2.33
19	10	9	8	45	28	1	0.02
20	9	10	3	84	30	6	0.05
Média				142.75	44.45	7.7	0.63

Os tempos computacionais foram reduzidos em 18 dos 20 exemplos (representados em negrito na tabela) e o tempo médio para execução de todos eles, comparativamente ao obtido em Pinto e Yanasse [5], foi reduzido de maneira significativa.

Confirmando as expectativas, o número de restrições (7) adicionadas (para eliminação de ciclos) até a determinação de uma solução ótima foi pequeno. Para os exemplos resolvidos, na média, menos de 8 restrições foram introduzidas por problema, sendo que no máximo, em um exemplo, foi necessária a introdução de 37 restrições. O pré-processamento dos nós também mostrou-se bastante útil reduzindo de maneira significativa o número total de nós a serem considerados nos exemplos.

Pode-se constatar também que os tempos maiores de execução observados se deram em casos onde o número de nós do grafo considerados eram relativamente grandes e, já era de nosso conhecimento que o tempo de resolução do problema está intimamente relacionado ao tamanho do grafo.

4. Considerações Finais

Neste trabalho, apresentamos uma formulação de fluxo em rede a custo mínimo com restrições adicionais

para o MTSP. A formulação proposta parece ser promissora, principalmente para exemplos práticos de tamanho pequeno, ou com caixas de ferramentas de pequena capacidade, ou ainda com caixas de ferramentas com capacidade relativamente grande em relação ao número total de ferramentas requeridas.

Algumas características e estruturas especiais da formulação proposta podem, eventualmente, ser exploradas para se desenvolver algoritmos especializados mais eficientes para sua resolução. Pesquisas também podem ser conduzidas para se obter algum suporte teórico que dê garantias de que o procedimento para obtenção de uma solução inteira, quando relaxamos a condição de integralidade, convergirá rapidamente para uma solução inteira.

Reconhecimento: Este trabalho foi parcialmente financiado pela FAPESP, CNPq e CAPES.

5. Referências

- [1] Bazaraa, M. S.; Jarvis, J. J. & Sherali, H. D., *Linear Programming and Network Flows*, W. H. Freeman and Company, John Wiley & Sons, 1990.
- [2] Gilmore, P. C. & Gomory, R., "A Linear Programming Approach to the Cutting-Stock Problem", *Operations Research*, 1961, **9**, 848-859.
- [3] Gilmore, P. C. & Gomory, R., "A Linear Programming Approach to the Cutting-Stock Problem- Part II", *Operations Research*, 1963, **11**, 863-888.
- [4] Nemhauser, G. L.; Wolsey, L. A., *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.
- [5] Pinto, M. J & Yanasse, H. H., "O Problema de Minimização de Troca de Ferramentas: propostas para sua resolução", Trabalho apresentado no XXXIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), Campos de Jordão, SP, 2001, 1410-1419.
- [6] Pinto, M. J & Yanasse, H. H., "O Problema de Corte e Sequenciamento de Padrões: uma Abordagem Integrada", Trabalho apresentado no I Workshop dos Cursos de Computação Aplicada do INPE (WORCAP), São José dos Campos, SP, 2001, 109-111.
- [7] Shapiro, J. F., *Mathematical Programming Structures and Algorithms*, John Wiley & Sons, 1979.
- [8] Tang, C. S. & Denardo, E. V., "Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches", *Operations Research*, 1988, **36**, 767-777.
- [9] Yanasse, H. H., "On a pattern sequencing problem to minimize the maximum number of open stack", *EJOR*, 1997, **100**, 454-563.