

Um algoritmo exato com ordenamento parcial para solução de um problema de programação da produção: experimentos computacionais



Horacio Hideki Yanasse

José Carlos Becceneri

Nei Yoshihiro Soma

Resumo

Nesta nota técnica apresentamos os resultados de testes computacionais da implementação feita de um algoritmo exato proposto na literatura para resolver um problema de seqüenciamento que surge em alguns ambientes produtivos em que desejamos minimizar pedidos em aberto dos clientes. Dos resultados computacionais observamos que os critérios de dominância incorporados no processo enumerativo deste algoritmo reduzem o espaço de busca, tornando-o mais eficiente em termos de tempo de execução.

Palavras-chave: Seqüenciamento de tarefas. Enumeração.

1 Introdução

Consideremos uma fábrica que produz diversos tipos de itens. Devido às suas características peculiares, a fábrica é capaz de produzir apenas um tipo de item de cada vez. Uma vez decidido qual item produzir, toda a demanda do mês daquele tipo de item é fabricada devido aos altos custos de preparação. Assim, apenas um lote de cada tipo de item é produzido no mês.

Consideremos que esta fábrica tem em cada mês uma carteira de pedidos de vários clientes. Na Tabela 1 apresentamos uma carteira hipotética destes pedidos.

Da Tabela 1 constatamos a existência de 11 tipos de itens (a, b, c, d, e, f, g, h, i, j, k) produzidos nesta fábrica. Cada cliente solicitou um subconjunto destes produtos. Por exemplo, o cliente 7 solicitou os itens b, d e g, o cliente 13 solicitou os itens a, b e h, e o cliente 12 solicitou os itens j e k.

Se a fábrica produzir os itens a, b, c, d, e, f, g, h, i, j, k, nesta seqüência, os pedidos dos clientes vão sendo parcialmente atendidos segundo a Tabela 2.

Com a produção dos itens na seqüência indicada anteriormente, o primeiro cliente que teve seu pedido completado foi o cliente 1, depois o cliente 11, 10, 2, 6, e assim por diante (veja a terceira coluna da Tabela 2). Na

segunda coluna da Tabela 2, observamos a evolução de como os pedidos dos clientes estão sendo completados. Por exemplo, após a produção do item tipo b, temos nove pedidos parcialmente atendidos dos clientes 1, 2, 3, 5, 7, 9, 10, 11 e 13. Os demais clientes ainda não tiveram nenhum item ordenado fabricado.

Pedidos incompletos podem ser inconvenientes para a fábrica, por exemplo, devido à dificuldade no manuseio dos itens fabricados, dificuldade de armazenagem devido ao espaço físico limitado, recursos financeiros empatados em estoque, etc. Por outro lado, um pedido completo pode ser entregue de imediato ao cliente e faturado.

Neste contexto, há interesse em se determinar uma seqüência de produção dos itens que evite este número grande de pedidos parcialmente atendidos, ou seja, desejamos minimizar o número máximo de pedidos parcialmente atendidos durante a fabricação dos itens.

Vários trabalhos abordam este problema de seqüenciamento descrito anteriormente, muitos deles em contextos distintos do apresentado. No contexto de cortes, este problema é conhecido como o problema de minimização de pilhas abertas, ou MOSP, acressemia de *Minimization of Open Stack Problem*.

Tabela 1. Carteira hipotética de pedidos.

Cliente	Pedido	Cliente	Pedido	Cliente	Pedido	Cliente	Pedido
1	a, b, c	2	b, e, f	3	b, f, g	4	d, e, g
5	b, c, h	6	d, e, f	7	b, d, g	8	e, j
9	a, c, h	10	b, d, e	11	b, c, d	12	j, k
13	a, b, h	14	i, k				

Tabela 2. Pedidos sendo atendidos conforme seqüência de produção.

Seqüência de produção	Evolução dos pedidos parcialmente atendidos dos clientes	Clientes cujos pedidos não sendo completados
a	1, 9, 13	
b	1, 2, 3, 5, 7, 9, 10, 11, 13	
c	1, 2, 3, 5, 7, 9, 10, 11, 13	1
d	2, 3, 4, 5, 6, 7, 9, 10, 11, 13	11
e	2, 3, 4, 5, 6, 7, 8, 9, 13	10
f	2, 3, 4, 5, 6, 7, 8, 9, 13	2, 6
g	3, 4, 5, 7, 8, 9, 13	3, 4, 7
h	5, 8, 9, 13	5, 9, 13
i	8, 14	
j	8, 12, 14	8
k	12, 14	12, 14

MOSP é um problema NP - árduo (LINHARES; YANASSE, 2002) e foi abordado em Yuen (1991), Yuen (1995), Yuen e Richardson (1995), Yanasse (1996, 1997a, 1997b, 1997c, 1998), Limeira (1998), Foerster e Wäscher (1998), Faggioli e Bentivoglio (1998), Yanasse e Limeira (1998, 2004a, 2004b), Yanasse et al. (1998, 1999, 2002a, 2002b), Becceneri (1999), Ashikaga (2001), Becceneri et al. (2002, 2004) apenas para citar alguns. Relações deste problema com outros problemas de otimização combinatória podem ser encontrados em Linhares e Yanasse (2002).

Esta nota técnica é uma complementação do artigo de Yanasse et al. (2002a), em que o algoritmo de ordenamento parcial para resolução do MOSP foi sugerido. O algoritmo de ordenação parcial modifica a enumeração proposta no algoritmo apresentado em Becceneri et al. (2004), impedindo que seqüências de fabricação de itens que não obedeçam a uma ordem pré-determinada de alguns deles, sejam geradas e analisadas.

Desde 2002, quando foi proposto, este algoritmo de ordenação parcial não foi implementado e, portanto, não se tem conhecimento de como é seu desempenho computacional. O objetivo deste trabalho é verificar o desempenho computacional deste algoritmo.

No presente trabalho apresentamos os resultados de testes computacionais realizados com a implementação feita do algoritmo de ordenação parcial. Dos resultados

obtidos com a aplicação deste algoritmo a instâncias da literatura e outros, gerados aleatoriamente, observamos que este algoritmo é computacionalmente mais eficiente do que o algoritmo proposto em Becceneri et al. (2004), que é pelo nosso conhecimento, aquele que apresenta o melhor desempenho computacional dentre os algoritmos exatos propostos na literatura para resolução do MOSP.

Com os resultados deste trabalho, o algoritmo de ordenação parcial proposto em Yanasse et al. (2002a) passa a ser uma referência relevante para os futuros algoritmos a serem desenvolvidos para o MOSP. Sem os resultados computacionais dos testes realizados, apresentados neste trabalho, isto não poderia ser afirmado.

Na próxima seção revisamos alguns conceitos desenvolvidos anteriormente para o MOSP e o trabalho de Yanasse et al. (2002a), apresentando os ordenamentos parciais da produção dos itens em uma solução ótima. O esquema de enumeração baseado no espaço solução ordenado parcialmente de Yanasse et al. (2002a) é revisado na seção 3. Tais reproduções visam manter este texto auto contido, facilitando o entendimento do leitor. Apresentamos os resultados dos testes computacionais na seção 4 e fazemos algumas considerações finais na seção 5.

2 Ordenamentos parciais

Seja $G(N,A)$ um grafo MOSP, em que N é o conjunto de nós do grafo e A o conjunto de arcos.

Um grafo MOSP é construído da seguinte forma (YANASSE, 1997b): cada cliente corresponde a um nó no grafo e existe uma aresta entre dois nós neste grafo, ou seja, são nós adjacentes se existe pelo menos um tipo de item comum no pedido dos dois clientes.

O MOSP pode ser colocado como um problema de percorrimento de arcos no grafo MOSP (YANASSE, 1997b). Procuramos uma seqüência de percorrimento dos arcos de modo que o número máximo de nós abertos é minimizado durante o percorrimento.

Um nó é dito aberto se ele já foi visitado, mas não foi ainda rotulado. Um nó é dito visitado se ao menos um arco incidente a ele já foi percorrido. Um nó é dito rotulado se todos os arcos incidentes a ele já foram percorridos.

Dada a correspondência nó-cliente, um nó aberto corresponde a um pedido de cliente em aberto; um nó

rotulado corresponde a um pedido de cliente completado.

Qualquer seqüência de percorrimento de arcos do grafo MOSP define uma ordem em que os nós são rotulados. Observemos que seqüências diferentes de percorrimento de arcos podem resultar numa mesma ordem de rotulação de nós, ou seja, dada uma ordem em que os nós são rotulados, podemos ter muitas seqüências diferentes correspondentes de percorrimento de arcos.

A ordem na qual os nós são rotulados determina o número de nós abertos (e quais são eles) ao longo do percorrimento. MOSP, portanto, consiste em determinar um ordenamento de rotulação dos nós que minimiza o número de nós abertos.

2.1 Ordenamento parcial tipo I

Sejam dois nós adjacentes i e j em $G(N,A)$ de grau 2. Existe sempre uma solução ótima para o MOSP em que i e j são rotulados um imediatamente após o outro.

Vamos denotar este ordenamento por $o(i) \equiv o(j)$, em que $o(k)$ é uma função que dá a ordem de rotulamento do nó k na seqüência. Diremos que os nós i e j são equivalentes do tipo I.

2.2 Ordenamento parcial tipo II

Sejam dois nós i e j em $G(N,A)$, A_i o conjunto de nós em N que são adjacentes ao nó i , e A_j o conjunto de nós em N que são adjacentes ao nó j . Se $A_j \subset \{A_i \cup \{i\}\}$ dizemos que existe uma dominância do nó i sobre o nó j . Existe sempre uma solução ótima para o MOSP em que o nó j é rotulado antes da rotulação do nó i . Vamos denotar este ordenamento por $o(j) < o(i)$.

Quando $A_j = A_i$ ou $\{A_i \cup \{i\}\} = \{A_j \cup \{j\}\}$, então existe uma dominância de i sobre j e de j sobre i . Neste caso dizemos que i e j são equivalentes e existe sempre uma solução ótima, em que i e j aparecem um imediatamente após o outro. Vamos denotar este ordenamento por $o(i) = o(j)$ e diremos que os nós i e j são equivalentes do tipo II.

O método enumerativo do tipo *branch and bound* proposto em Becceneri et al. (2004) apenas utiliza a informação de nós equivalentes do tipo II e reduz o grafo MOSP, mantendo apenas um destes nós. A enumeração com ordenamento parcial proposta em Yanasse et al. (2002a) difere da sugerida em Becceneri et al. (2004), pois faz uso de nós equivalentes dos tipos I e II, e nós dominantes.

3 Enumeração com ordenamento parcial

Os métodos descritos em Becceneri et al. (2004) e Yanasse et al. (2002a) seguem o processo de enumeração proposto em Yanasse (1997a), que gera todas as permutações promissoras possíveis de seqüências de finalização

de um pedido de cliente, para selecionar a melhor. A enumeração destas permutações é feita de maneira direta utilizando uma árvore. A partir do nó raiz 0 desta árvore, novos nós são obtidos, cada um deles considerando a possibilidade de finalizar em primeiro lugar um dos pedidos dos clientes. De cada um desses novos nós da árvore, novos nós são obtidos, cada um deles considerando a possibilidade de finalizar em segundo lugar um dos pedidos dos clientes ainda não finalizados, no caminho que vai do nó raiz 0 até o nó sendo ramificado. De maneira similar, de cada um desses novos nós da árvore, novos nós são obtidos, cada um deles considerando a possibilidade de finalizar em terceiro lugar um dos pedidos dos clientes ainda não finalizados, no caminho que vai do nó raiz 0 até o nó sendo ramificado. Sucessivamente, seguindo o mesmo procedimento, as demais ramificações são feitas gerando todas as permutações possíveis de seqüências de finalização.

Para evitar enumerar permutações que não sejam promissoras, ramificações a partir de certos nós desta árvore de enumeração são descartadas comparando-se limitantes inferiores determinados pelas ordens em aberto que resultam com as seqüências parciais já definidas em cada um dos nós, e um limitante superior dado pela melhor solução corrente existente encontrada até o momento da ramificação.

A enumeração é completa, embora seja feita de modo implícito com o uso dos limitantes calculados, portanto, o algoritmo é exato, ou seja, a solução encontrada tem garantia de otimalidade.

Na enumeração proposta em Becceneri et al. (2004), além da equivalência de nós tipo II já mencionada anteriormente, foi também levada em consideração uma nova heurística desenvolvida que determina um bom limitante superior inicial para o problema. No algoritmo de ordenação parcial enumeramos todas as permutações que obedecem aos ordenamentos parciais determinados, e somente elas, para encontrarmos uma solução ótima para o problema. Isto é suficiente, pois existe sempre uma solução ótima que satisfaz a estes ordenamentos parciais do tipo I e II.

Para ilustrar como procedemos a esta enumeração com ordenamento parcial, consideremos, por exemplo, o grafo MOSP da Figura 1. Neste grafo os nós 10 e 11 tem grau 2 e são adjacentes. Portanto, $o(10) \equiv o(11)$. Não existe nenhum outro ordenamento parcial do tipo I além desse já identificado.

Verificamos a seguir se existe dominância entre nós ou ordenamentos do tipo II. Iniciamos em ordem crescente com os nós de menor grau. Identificamos que o nó 9 é dominado pelo nó 11 e, portanto, $o(9) < o(11)$, os nós 6 e 7 são equivalentes, ou seja, $o(6) = o(7)$, os nós 1 e 8 são equivalentes, ou seja, $o(1) = o(8)$, e assim por diante.

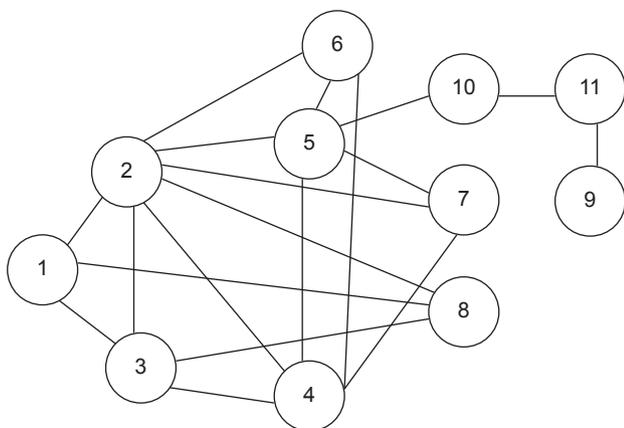


Figura 1. Exemplo de um grafo MOSP.

Com isso, os ordenamentos estabelecidos para o exemplo são os listados na Tabela 3.

Montada a tabela de ordenamentos parciais, passamos à enumeração das permutações que obedecem a estes ordenamentos.

Para se proceder a esta enumeração, sugerimos consolidar as informações de ordenamento da Tabela 3. Na Tabela 4, apresentamos o resultado da consolidação feita em que, em cada linha os nós estão ordenados de acordo com os ordenamentos levantados.

Na implementação computacional, a partir da Tabela 4, construímos por conveniência uma nova Tabela T^0 em que adicionamos novas linhas para cada um dos nós do grafo MOSP que não possui relação nenhuma de precedência com nenhum dos demais nós. No caso do exemplo, todos os 11 nós do grafo têm alguma relação com algum outro nó, portanto, não há linhas a acrescentar e a Tabela T^0 é a própria Tabela 4. Se houvesse, por exemplo, um nó 12 no grafo MOSP que não possuísse nenhuma relação de precedência com os demais nós do grafo, incluiríamos uma linha adicional G5 na Tabela T^0 com a informação simples “o(12)”.

Como já mencionado, o processo enumerativo de rotulação completa de nós segue o apresentado em Yanasse (1997a), exceto que são enumeradas apenas as permutações que obedecem aos ordenamentos parciais indicados na Tabela T^0 . Assim, sugerimos manter uma cópia apropriada desta tabela em cada nó da árvore de enumeração. Com isso, para saber as ramificações a serem consideradas a partir de cada nó na árvore de enumeração, basta consultar a tabela existente no nó.

Iniciamos no nó raiz 0 com a Tabela T^0 que, para o exemplo ilustrativo, é a própria Tabela 4. Da Tabela T^0 no nó raiz 0, observamos que a partir deste nó é necessário considerar a enumeração de três novos nós, um para cada primeiro elemento diferente de cada linha da tabela, ou seja, nó 1 de enumeração para o cliente 9, nó 2 de

Tabela 3. Ordenamentos parciais dos nós do exemplo dado na Figura 1.

Ref.	Ordenamento
P1	$o(10) \cong o(11)$
P2	$o(9) < o(11)$
P3	$o(6) = o(7)$
P4	$o(1) = o(8)$
P5	$o(1) < o(3)$
P6	$o(7) < o(4)$
P7	$o(7) < o(5)$
P8	$o(3) < o(2)$
P9	$o(4) < o(2)$

Tabela 4. Ordenamentos parciais dos nós do exemplo dado na Figura 1.

Ref.	Ordenamento
G1	$o(9) < o(11) \cong o(10)$
G2	$o(1) = o(8) < o(3) < o(2)$
G3	$o(6) = o(7) < o(4) < o(2)$
G4	$o(6) = o(7) < o(5)$

enumeração para o cliente 1, e nó 3 de enumeração para o cliente 6. Observamos que não é necessário enumerar os demais nós, pois estes só podem ser rotulados depois de rotulados os nós que os precedem.

Em cada um dos nós da árvore de enumeração, guardamos uma tabela de ordenamentos parciais, obtida a partir da tabela do nó precedente na árvore de enumeração. Esta tabela é obtida eliminando o nó enumerado da tabela e todos os seus equivalentes do tipo I e do tipo II. Assim, por exemplo, no nó 1 de enumeração, a Tabela T^1 neste nó é:

Tabela T^1

G11. $o(11) \cong o(10)$

G21. $o(1) = o(8) < o(3) < o(2)$

G31. $o(6) = o(7) < o(4) < o(2)$

G41. $o(6) = o(7) < o(5)$

em que o cliente 9 foi eliminado da Tabela T^0 ; no nó 2 de enumeração a Tabela T^2 neste nó é:

Tabela T^2

G12. $o(9) < o(11) \cong o(10)$

G22. $o(3) < o(2)$

G32. $o(6) = o(7) < o(4) < o(2)$

G42. $o(6) = o(7) < o(5)$

em que o cliente 1 e o cliente 8 (por ser equivalente do tipo II ao cliente 1) foram eliminados da Tabela T^0 , e no nó 3 de enumeração, a Tabela T^3 neste nó é:

Tabela T³

G13. $o(9) < o(11) \cong o(10)$

G23. $o(1) = o(8) < o(3) < o(2)$

G33. $o(4) < o(2)$

G43. $o(5)$

em que o cliente 6 e o cliente 7 (por ser equivalente do tipo II ao cliente 6) foram eliminados da Tabela T⁰.

Similarmente ao que ocorreu no nó raiz 0, a partir de cada um dos nós da árvore de enumeração, podemos consultar a tabela armazenada no nó e determinar os novos nós que devem ser enumerados, um para cada primeiro elemento diferente de cada linha da tabela. As tabelas correspondentes aos novos nós enumerados são obtidas de maneira similar ao ilustrado anteriormente.

4 Testes computacionais

O algoritmo de enumeração com ordenamento parcial foi implementado conforme descrito na Seção 3. Para percorrer a árvore de enumeração, adotamos a estratégia de “profundidade - primeiro” aliada a “gulosa”. A heurística apresentada em Becceneri et al. (2004) foi também utilizada para se determinar uma solução inicial para o problema. O algoritmo foi implementado em C++, com Microsoft Visual C++ 6.0, usando a plataforma Windows 98 em um Pentium II, 1,70 GHz, com 512 MB de RAM.

O algoritmo foi testado em diversas instâncias geradas aleatoriamente ou disponíveis na literatura. Para fins de comparação, as mesmas instâncias foram também resolvidas com o algoritmo de Becceneri et al. (2004).

Na Tabela 5 apresentamos os resultados computacionais observados. Nesta tabela, na coluna N1 é apresentado o número de nós enumerados pelo algoritmo de Becceneri et al. (2004). Nesta mesma coluna apresentamos entre parênteses, o número de nós enumerados até o instante quando se obteve a melhor solução. Quando não há indicação de nenhum número entre parênteses, significa que a heurística de Becceneri et al. (2004) gerou a solução ótima, e a enumeração apenas serviu para verificar que a solução era a ótima. Na coluna N2, de maneira similar à coluna N1, apresentamos o número de nós enumerados pelo algoritmo de ordenação parcial.

Nesta Tabela 5, os casos de testes do tipo Txxyy(w)i foram gerados por um gerador desenvolvido em Becceneri (1999). A notação utilizada é a seguinte: “xx” indica o número total de ordens; “yy” indica o número total de tipos de itens; “w” é a quantidade máxima de tipos de itens em cada ordem; e “i” é um número do caso de teste. Os casos de testes do tipo PyyxxNi são os gerados por Faggioli e Bentivoglio (1998). O significado de “xx”, “yy” e “i” são os mesmos indicados no parágrafo anterior.

Os demais casos de testes são de Linhares et al. (1999). Da tabela, observamos que, sistematicamente, o algoritmo de enumeração com ordenamento parcial sugerido em Yanasse et al. (2002a) apresentou tempos de execução menores que o de Becceneri et al. (2004) e o número de nós enumerados nas árvores também são menores.

5 Conclusões e sugestões para trabalhos futuros

Pelo nosso conhecimento, o algoritmo proposto em Becceneri et al. (2004) é o que apresenta o melhor desempenho computacional dentre os algoritmos exatos propostos na literatura para resolução do MOSP. Em Yanasse et al. (2002a), o algoritmo de ordenação parcial, ou seja, o algoritmo modificado de enumeração que faz uso de ordenamentos parciais de nós para reduzir o espaço de busca de uma solução ótima para o problema de minimização do número máximo de ordens abertas, foi sugerido. Desde a data em que foi proposto até o presente momento, este algoritmo não havia sido implementado e, portanto, não se tinha conhecimento de como era o seu desempenho computacional. Com a implementação feita e com os testes computacionais realizados e apresentados neste trabalho, observamos que o algoritmo de ordenação parcial é computacionalmente mais eficiente do que o algoritmo proposto em Becceneri et al. (2004). Com isso, o algoritmo de ordenação parcial de Yanasse et al. (2002a), com a heurística de Becceneri et al. (2004) para determinação de um limitante superior inicial para o problema, passa a ser uma referência relevante de comparação para os novos algoritmos a serem desenvolvidos para o MOSP.

Cabe lembrar que as ordens parciais determinadas não são necessárias em uma solução do problema para que ela seja ótima; o que sabemos é que existe pelo menos uma solução ótima em que esta ordem é obedecida, e é justamente uma solução com este ordenamento que é procurada pelo algoritmo de ordenação parcial. As soluções ótimas determinadas pelo algoritmo de Becceneri et al. (2004) não necessariamente obedecem a estes ordenamentos parciais.

Observarmos que os ordenamentos parciais obtidos podem ser também usados para fornecer limitantes inferiores, possivelmente melhores, para o problema. Eles também podem ser usados em meta-heurística. Por exemplo, em buscas locais (AARS; LENSTRA, 1997) ou quando se geram novas soluções (como em algoritmos genéticos), seqüências que não obedecem a estes ordenamentos podem ser evitadas.

Tabela 5. Resultados dos testes computacionais.

Caso de teste	N1	segundos N1	N2	segundos N2
T2020C(05)01	2730 (20)	0,08	428(20)	0,01
T2020C(05)02	5391	0,1	213	0
T2020C(05)03	2303	0,02	153	0
T2020C(05)04	786	0,01	48	0
T2020C(05)05	1817 (52)	0,02	178(50)	0,01
T4040C(10)01	31666 (71)	1,172	6007(71)	0,28
T4040C(10)02	17428 (75)	0,791	2994(71)	0,22
T4040C(10)03	7863	0,35	1755(329)	0,11
T4040C(10)04	26777 (2478)	0,992	4797(1992)	0,13
T4040C(10)05	39297 (148)	1,392	7731(145)	0,33
T4040C(10)06	27794	1,051	3505	0,08
T4040C(10)07	9290	0,52	1273	0,13
T4040C(10)08	10041	0,51	1899	0,14
T4040C(10)09	14182	0,67	3623	0,17
T4040C(10)10	45334 (285)	1,902	7146(210)	0,33
T4040C(15)01	465	0,05	47	0,01
T4040C(15)02	2796 (114)	0,13	494(74)	0,03
T4040C(15)03	1099 (195)	0,05	180(139)	0,01
T4040C(15)04	1775	0,09	235	0,01
T4040C(15)05	1808	0,08	278	0,02
T4040C(15)06	2074	0,09	244	0,01
T4040C(15)07	1422	0,06	92	0
T4040C(15)08	274	0,011	21	0
T4040C(15)09	1081	0,05	110	0,01
T4040C(15)10	978	0,04	119	0,01
T4040C(20)01	78	0,01	20	0,01
T4040C(20)02	40	0	8	0
T6060C(15)01	9336(61)	0,55	8575(61)	0,49
T6060C(15)02	2304	0,29	2042	0,26
T6060C(15)03	524(191)	0,06	451(180)	0,05
T6060C(20)01	176(119)	0,03	85(65)	0,01
T6060C(20)02	573(120)	0,1	392(99)	0,08
T6060C(20)03	571(414)	0,1	329(251)	0,02
v4090 (23 x 27)	2801137	23,82	19172	0,32
v4470 (37 x 47)	61134983 (197)	900	3713948 (3992)	55,88
W1 (18 x 21)	814(814)	0,03	413(413)	0,02
W3	3940383	240	3929680	240
Wsn (17 x 25)	61541	1,40	10066	0,18
X0 (40 x 48)	3991827	62,11	3712756	59,34
V1 (6 x 8)	30	0	16	0
Data_III (18 x 15)	186(186)	0	149(149)	0
Data_V (37 x 29)	15404456	240	15280198	240
P1020N0	53	0	48	0
P1030N2	11(11)	0	11(11)	0
P2010N0	284(106)	0,02	130(72)	0,01
P2010N1	218	0	157	0
P2010N2	420	0,01	20	0
P2020N0	1015	0	478(141)	0,01
P2020N1	2188	0,02	1269	0,01
P2020N2	736	0,01	550	0
P2030N0	2031(151)	0,02	1056(119)	0,01

Tabela 5. Continuação...

Caso de teste	N1	segundos N1	N2	segundos N2
P2030N1	7090	0,06	5699	0,05
P2030N2	5339	0,12	2189	0,1
P2040N0	6684	0,04	1903	0,01
P2040N1	10919 (20)	0,19	8367 (20)	0,15
P2040N2	10564	0,09	3307	0,03
P2050N0	5681(69)	0,05	4944(65)	0,03
P2050N1	22(22)	0	22(22)	0
P2510N0	663(71)	0,02	68(68)	0
P2510N1	537(201)	0,03	272(23)	0,01
P2510N2	582(19)	0,02	266(119)	0,01
P2520N0	9339(2915)	0,2	3256(720)	0,06
P2520N1	6451(469)	0,08	4483(425)	0,04
P2520N2	5623	0,06	3917	0,04
P2530N0	9767(416)	0,1	8071(367)	0,09
P2530N1	16004	0,16	11663	0,12
P2530N2	30045	0,3	23118	0,23
P2540N0	54542	0,52	27842	0,26
P2540N1	34704(7658)	0,32	14762(4522)	0,14
P2540N2	71938	0,67	46541	0,44
P2550N0	74434(5687)	0,69	72633(5558)	0,68
P2550N1	14380(202)	0,13	6890(135)	0,06
P2550N2	31301(139)	0,28	22091(139)	0,21
P3010N0	1560(262)	0,03	466(156)	0,01
P3010N1	1237	0,03	564(374)	0,01
P3010N2	1195	0,02	281(29)	0
P3020N0	61074	4,31	55297	3,71
P3020N1	52671(185)	0,8	41195(170)	0,56
P3020N2	26563	0,4	13425	0,19
P3030N0	13034(1712)	0,18	12337(1669)	0,16
P3030N1	284552(31)	3,86	253014(31)	3,2
P3030N2	252023(84)	3,45	184829(81)	2,49
P3040N0	5930175	75,92	3154209	38,64
P3040N1	164305(42334)	2,12	144584(42334)	1,91
P3040N2	22319737	268,98	2117469	24,90
P3050N0	1618411(55)	19,33	618535(54)	7,34
P3050N1	29746920	339,46	12176006	136,27
P3050N2	248499	2,89	215611	2,46
P4010N0	8295(2709)	0,24	1656(90)	0,03
P4010N1	29528(4201)	0,82	752(175)	0,02
P4010N2	39416	4,36	508	0,04
P4020N0	14454	0,35	13106	0,3
P4020N1	105274(1268)	2,64	69906(999)	1,6
P4020N2	1652812(162)	39,75	667806(162)	14,42
P4030N0	383966(1034)	8,46	383966(1034)	8,31
P4030N1	876088	19,53	705267	15,38
P4030N2	20005407(258)	426,66	19325000(250)	399,25
P4040N0	962067(20841)	19,22	912413(20194)	18,0
P4040N1	1604516(20977)	33,42	895939(17440)	18,01
P4040N2	3899222(603752)	74,84	3899222(603752)	74,59
P4050N1	1631065(229839)	31,18	1363476(214320)	25,49

An exact algorithm with partial order for the solution of a production scheduling problem: computational experiments

Abstract

In this paper, we present the computational test results of an implementation made of an exact algorithm proposed in the literature to solve a sequencing problem that arises in some productive environments where open orders of clients should be minimized. From the computational tests, it can be observed that the dominance criteria incorporated in the enumeration process of this algorithm reduces the search space, making the algorithm more efficient in terms of execution time.

Keywords: Job sequencing. Enumeration.

Referências

- AARS, E.; LENSTRA, J. K. **Local search in combinatorial optimization**. Chichester: John Wiley & Sons, 1997. 512 p.
- ASHIKAGA, F. M. **Um método frugal para o problema de minimização de pilhas abertas**. 2001. Dissertação (Mestrado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2001.
- BECCENERI, J. C. **O problema de seqüenciamento de padrões para a minimização do número máximo de pilhas abertas em ambientes de cortes industriais**. 1999. Tese (Doutorado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 1999.
- BECCENERI, J. C.; YANASSE, H. H.; SOMA, N. Y. A method for solving the minimization of the maximum number of open stacks problem. In: CLAIO, 2002, Concepción. **Proceedings...** Concepción, Chile, 2002.
- _____. A method for solving the minimisation of the maximum number of open stacks problem within a cutting process. **Computers and Operations Research**, v. 31, n. 14, p. 2315-2332, 2004.
- FAGGIOLI, E.; BENTIVOGLIO, C. A. Heuristic and exact methods for the cutting sequencing problem. **European Journal of Operational Research**, v. 110, n. 3, p. 564-575, nov. 1998.
- FOERSTER, H. E.; WÄSCHER, G. Simulated annealing for order spread minimization in sequencing cutting patterns. **European Journal of Operational Research**, v. 110, n. 2, p. 272-281, 1998.
- LIMEIRA, M. S. **Desenvolvimento de um algoritmo exato para a solução de um problema de seqüenciamento de padrões de corte**. 1998. Dissertação (Mestrado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1998.
- LINHARES, A.; YANASSE, H. H. Connections between cutting-pattern sequencing, VLSI design, and flexible machines. **Computers and Operations Research**, v. 29, n. 12, p. 1759-1772, 2002.
- LINHARES, A.; YANASSE, H. H.; TORREÃO, J. R. A. Linear gate assignment: a fast statistical mechanics approach. **IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems**, v. 18, n. 12, p. 1750-1758, 1999.
- YANASSE, H. H. Minimization of open orders-polynomial algorithms for some special cases. **Pesquisa Operacional**, v. 16, n. 1, p. 1-26, jun. 1996.
- _____. On a pattern sequencing problem to minimize the maximum number of open stacks. **European Journal of Operational Research**, v. 100, n. 3, p. 454-463, 1997a.
- _____. A transformation for solving a pattern sequencing problem in the wood cut industry. **Pesquisa Operacional**, v. 17, n. 1, p. 57-70, 1997b.
- _____. **An exact algorithm for the TREE case of the minimization of open orders problem**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 1997c. (Relatório de Pesquisa do LAC - 001/97).
- _____. **A note on generating solutions of a pattern sequencing problem to minimize the maximum number of open orders**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 1998. (Nota Técnica do LAC, n. 2).
- YANASSE, H. H.; BECCENERI, J. C.; SOMA, N. Y. Um algoritmo exato para um caso especial do problema de seqüenciamento de padrões. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 30., 1998, Curitiba. **Resumos...** Curitiba, PR, 1998. p. 385-386.
- _____. An exact algorithm for a special case of a patterns sequencing problem. In: OFICINA NACIONAL DE PROBLEMAS DE CORTE E EMPACOTAMENTO, 3.; SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 30., 1998, Curitiba. **Anais...** Curitiba, PR, 1998. p. 53-61.
- _____. Bounds for a problem of sequencing patterns. **Pesquisa Operacional**, v. 19, n. 2, p. 249-277, dez. 1999.
- _____. Ordenamento parcial para reduzir o espaço de busca de uma solução ótima para um problema de seqüenciamento de padrões. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 34, 2002, Rio de Janeiro. **Anais...** Rio de Janeiro, RJ, 2002a. 1 Cd-Rom.
- _____. A heuristic to solve a pattern sequencing problem based on partial orderings. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 22; INTERNATIONAL

- CONFERENCE ON INDUSTRIAL ENGINEERING AND OPERATIONS MANAGEMENT, 8., 23-25 oct. 2002, Curitiba, PR. **Proceedings...** Rio de Janeiro: ABEPRO, 2002b. p.105-111.
- YANASSE, H. H.; LIMEIRA, M. S. A branch and bound scheme for solving a pattern sequencing problem. In: EURO, 16., 1998a; Bruxelas, Bélgica. **Proceedings...** Bruxelas, 1998a. p.120.
- _____. Um algoritmo branch-and-bound para resolver um problema de seqüenciamento de padrões. In: OFICINA NACIONAL DE PROBLEMAS DE CORTE E EMPACOTAMENTO, 3.; SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 30., Curitiba. 1998. **Anais...** Curitiba, PR, 1998b. p. 63-68.
- _____. Refinements on an enumeration scheme for solving a pattern sequencing problem. **International Transactions in Operations Research**, v. 11, n. 3, p. 277-292, 2004.
- YUEN, B. J. Heuristics for sequencing cutting patterns. **European Journal of Operational Research**, v. 55, n. 2, p. 183-190, 1991.
- _____. Improved heuristics for sequencing cutting patterns. **European Journal of Operational Research**, v. 87, n. 1, p. 57-64, 1995.
- YUEN, B. J.; RICHARDSON, K. V. Establishing the optimality of sequencing heuristics for cutting stock problems. **European Journal of Operational Research**, v. 84, n. 3, p. 590-598, 1995.

Sobre os autores

Horacio Hideki Yanasse

José Carlos Becceneri

Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais – INPE/LAC, CP 515, CEP 12201-970, São José dos Campos, SP, Brasil, e-mails: horacio@lac.inpe.br; becce@lac.inpe.br

Nei Yoshihiro Soma

Instituto Tecnológico de Aeronáutica – ITA, Centro Técnico Aeroespacial – CTA, CEP 12228-901, São José dos Campos, SP, Brasil, e-mail: nysoma@comp.ita.cta.br

Agradecimentos: Este trabalho foi financiado parcialmente com recursos da FAPESP e CNPq.

Recebido em 8/6/2006

Aceito em 3/7/2007

