



ANÁLISE DA FLUTUAÇÃO DO GARGALO EM *FLOW SHOP* PERMUTACIONAL COM TEMPOS DE *SETUP* ASSIMÉTRICOS E DEPENDENTES DA SEQUÊNCIA

Alexandre Damas de Barros

João Vitor Moccellin

Departamento de Engenharia de Produção,
Escola de Engenharia de São Carlos, USP,
Avenida Trabalhador Sancarlense, 400, CEP 13566-590, São Carlos, SP,
e-mails: adbarros@prod.eesc.usp.br e jvmoccel@sc.usp.br

Recebido em 10/9/2002

Aceito em 10/8/2003

Resumo

Este artigo apresenta um método heurístico Simulated Annealing (SA) para a programação flow shop permutacional com tempos de setup assimétricos e dependentes da seqüência (ASDST), tendo por foco uma restrição relevante ao output de um sistema de produção, que é o estágio gargalo. O algoritmo proposto, denominado Metaheurística Simulated Annealing de Busca do Gargalo Flutuante (BGaFSA), analisa possível mudança no gargalo do sistema decorrente da ordenação das tarefas. O procedimento SA procura identificar a máquina gargalo real do sistema de produção. A medida de desempenho é a duração total da programação (makespan).

Palavras-chave: programação da produção, flow shop permutacional, tempos de setup, gargalos.

1. Introdução

Grande parte das pesquisas em *scheduling* considera os tempos de *setup* como não relevantes ou de pequena variabilidade (Allahverdi *et al.*, 1999) e, geralmente, os incluem nos tempos de processamento. Esse procedimento simplifica muito a análise em determinadas aplicações, principalmente quando os tempos de *setup* são consideravelmente menores que os tempos de processamento, ou em casos em que o *setup* destina-se à produção de lotes e é executado somente uma vez para um grande lote de produção. No entanto, para os casos em que os tempos de *setup* apresentam razão significativa diante dos tempos de processamento, há a necessidade de tratá-los diferenciadamente, uma vez que eles têm relação direta com a disponibilidade de equipamentos e acarretam custos específicos, como a necessidade de pessoal especializado para sua execução. O tratamento em separado dos tempos de *setup* pode levar, com a otimização do critério de desempenho adotado, a melhorias no atendimento à demanda e à facilidade no gerenciamento do sistema de produção.

O objetivo deste trabalho é apresentar um algoritmo para seqüenciamento de tarefas em *flow shop* permutacional com tempos de *setup* assimétricos e dependentes da seqüência (*Asymmetric Sequence Dependent Setup Times* – ASDST), chamado Metaheurística *Simulated Annealing* de Busca do Gargalo Flutuante (BGaFSA), que tem por foco o estágio gargalo do sistema (ou, mais especificamente, a máquina gargalo). O critério de desempenho é a duração total da programação (*makespan*), que corresponde à utilização eficiente dos recursos produtivos (Baker, 1974). Os problemas em ASDST são fortemente *NP-hard* (Das *et al.*, 1995) e a sua notação é definida por $Fm | S_{ijk}, prmu | C_{max}$ (Pinedo, 1995).

O algoritmo analisa, após a otimização da matriz assimétrica de tempos de *setup* do estágio gargalo inicialmente identificado, se a condição de gargalo se mantém para esse estágio ou se muda para outro estágio, e considera o impacto dessa flutuação no desempenho geral do sistema para propor a melhor seqüência das tarefas. Essa mudança no gargalo é possível e bastante provável em função de os tempos de *setup* serem assimétricos e dependentes da seqüência,

situação em que pode haver variação significativa na somatória dos tempos de *setup* em decorrência da alteração na ordenação das tarefas nos diversos estágios de produção (máquinas).

A técnica de otimização utilizada pelo BGaFSA é a metaheurística *Simulated Annealing* (SA), que tem por base uma analogia entre resfriamento de um sólido e otimização de sistema com muitas variáveis independentes (Parthasarathy & Rajendran, 1997). O uso da técnica SA é justificado pela capacidade de efetuar movimentos “colina acima” no espaço de soluções viáveis do problema, explorando os “vales” na tentativa de obter uma solução ótima global.

O *Simulated Annealing* pode ser considerado uma generalização do método “descendente”, no qual a busca não estendida para um mínimo global é terminada após um mínimo local ser obtido, podendo ser classificado como método heurístico melhorativo de busca aleatória na vizinhança (Moccellini, 1994).

Neste artigo procurou-se enfatizar que a aplicação do conceito de gargalo (Goldratt & Fox, 1997) no ambiente de produção considerado (*flow shop* permutacional com ASDST) deve ser adequadamente efetuada, pois o mesmo pode mudar de posição relativamente ao *set* de tarefas seqüenciadas, sendo que essa situação pode levar a perdas no sistema em função da dificuldade de identificar a posição real do gargalo e atribuir-lhe os cuidados necessários.

A seção 2 trata da influência das restrições na eficiência de um sistema produtivo e na seção 3 são relatadas as principais pesquisas com ASDST em *flow shop* permutacional. A descrição do algoritmo BGaFSA é feita na seção 4, seguindo-se a experiência computacional para avaliar o algoritmo. Na última seção, são apresentadas as conclusões e considerações finais do trabalho.

2. Influência das restrições na eficiência de um sistema produtivo

2.1 Definição de restrições

Em determinado sistema produtivo, restrições são todos os processos, máquinas, meios ou até comportamentos que impedem o sistema de atingir o máximo de seu desempenho. Segundo Stein (1997), as restrições podem ser definidas como:

- Comportamentais: quando o comportamento está em conflito com a realidade.
- Gerenciais: políticas gerenciais deficientes que restringem a utilização máxima dos recursos disponíveis.
- De capacidade: quando uma demanda local em um recurso excede a capacidade disponível.
- De mercado: quando a demanda de mercado é menor que a capacidade de recursos disponíveis.

- Logísticas: quaisquer problemas decorrentes dos sistemas de planejamento e controle da empresa.

O presente trabalho considera somente a restrição de capacidade, mas todas estão inter-relacionadas, afetando-se mutuamente. Na restrição de capacidade, o conceito mais difundido no chão de fábrica é o de estágio gargalo de produção ou, simplesmente, gargalo.

2.2 Administração de gargalo

Para Goldratt & Fox (1997), os gargalos representam restrições à saída (ou *output*) do sistema de produção. Pelo fato de ser a principal restrição do sistema, a qualidade de sua administração é essencial para atender à demanda, o que significa manter o estágio gargalo do sistema no processo de transformação (manufatura) de insumos disponível pelo maior tempo possível, reduzindo ao máximo o tempo de espera entre tarefas sucessivas e outras variáveis que afetam essa condição, como as relacionadas às filas de espera nos estágios anteriores de produção (estoque em processamento) e, eventualmente, interrupções de processamento. Para aumentar a eficiência do sistema produtivo, o estágio gargalo deve ter sua administração destacada em relação ao restante do sistema, uma vez que qualquer perda de desempenho nesse recurso significa perda direta estendida a todo o sistema, enquanto uma eventual perda em outro recurso pode ser mais facilmente recuperada e diluída no tempo da produção.

2.3 Impacto dos tempos de *setup* na eficiência do sistema

Conceitualmente, o tempo de *setup* de uma máquina abrange desde o final do processamento de uma tarefa até o início da tarefa seguinte. Para Flynn (1987), o tempo necessário para o *setup* tem relação direta com o grau de similaridade entre duas tarefas processadas sucessivamente em uma mesma máquina. Portanto, se duas tarefas a serem processadas em seqüência são similares, o tempo requerido para o *setup* será relativamente pequeno. Entretanto, se forem completamente diferentes, o tempo será proporcionalmente maior. Kim & Bobrowski (1994) definem assim a importância do tempo de *setup*: (1) o tempo de *setup* constitui parte do tempo de fluxo que afeta diretamente a taxa de saída do sistema de produção; (2) o custo da unidade do tempo de *setup* usualmente é mais alto que o custo unitário do tempo de processamento, uma vez que envolve o custo da máquina parada e o custo da mão-de-obra de um técnico especializado; e (3) o *setup* pode, com freqüência, precisar ser executado por um técnico com alto nível de especialização, o qual é um recurso limitado e nem sempre disponível.

O *setup* pode ser, então, visto como custo relevante no processo produtivo. Sua redução diminui a necessidade de pessoal para sua execução e, simultaneamente, aumenta o

tempo disponível da máquina. Além disso, proporciona reduções de estoque em processo e de *lead-time* de processamento (Barros, 2002).

3. Pesquisas com ASDST em *flow shop* permutacional

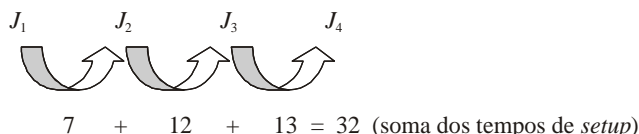
3.1 Caracterização dos problemas com ASDST

Os problemas com *Asymmetric Sequence Dependent Setup Times* são comumente encontrados em sistemas de produção com diversos tipos de itens, ou quando uma máquina multipropósito processa diversas tarefas (Baker, 1974). Os tempos de *setup* dependentes da seqüência são caracterizados pelo fato de o tempo de *setup* da máquina, após o processamento da tarefa j , para executar a tarefa i , expresso por s_{ji} , ser diferente do tempo de *setup* entre a tarefa j e uma outra tarefa k , expresso por s_{jk} , dentro da matriz de tempos de *setup* da mesma máquina. A assimetria ocorre quando o tempo de *setup* da tarefa j para a tarefa i é diferente do tempo de *setup* da máquina na seqüência inversa, ou seja, da tarefa i para a tarefa j .

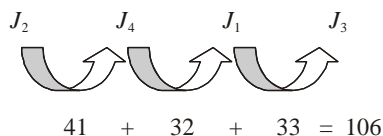
O impacto da assimetria e da dependência da seqüência pode ser percebido pelo exemplo a seguir, que trata do seqüenciamento de quatro tarefas em determinada máquina. Considere a seguinte matriz de tempos de *setup*:

	J_1	J_2	J_3	J_4
J_1	---	7	33	27
J_2	18	---	12	41
J_3	25	17	---	13
J_4	32	45	9	---

Para a seqüência $S_1 = \{J_1, J_2, J_3, J_4\}$, tem-se



Para outra seqüência, $S_2 = \{J_2, J_4, J_1, J_3\}$, a soma dos tempos de *setup* é significativamente maior:



O problema tradicional para esse caso é o Problema do Caixeiro Viajante (*Traveling Salesman Problem* – TSP). Foi proposto em 1934 por Hassler Whitney, em um

seminário na Universidade de Princeton, e tem esse nome em razão de o caixeiro viajante desejar viajar pelo caminho mais curto entre sua casa e um conjunto de determinadas cidades e retornar para sua casa, podendo, para isso, utilizar um método de otimização, caso ele tenha as distâncias entre cada par de cidades (Flood, 1956). Exemplo típico de ocorrência desse problema em ambiente de produção é no caso de um único equipamento produzir diferentes componentes químicos, requerendo procedimentos diferentes de limpeza entre o processamento de diferentes componentes, por exemplo, para a produção de diferentes cores de tintas, lotes de combustível, serviços gráficos e outros.

3.2 Métodos heurísticos em problemas com ASDST

Allahverdi *et al.* (1999) efetuaram extensa revisão na literatura, envolvendo pesquisas em programação da produção com tempos de *setup* em separado e divididos por critério de desempenho e modelo adotado. A pesquisa em ASDST começa com modelos desenvolvidos para máquina única, destacando o trabalho de Gavett (1965), que apresentou o desenvolvimento de três heurísticos construtivos (NB, NB' e NB'') a partir do TSP, com o objetivo de minimizar o *makespan*. Para problema de ASDST em *flow shop*, também com objetivo de minimizar o *makespan*, os trabalhos mais relevantes são de Simons Jr. (1992), Ríos-Mercado & Bard (1998, 1999). Simons Jr. (1992) apresenta quatro métodos heurísticos construtivos, em que os denominados *Minit* e *Micot* são aplicados à última máquina do sistema e a seqüência obtida é considerada para as demais. Os outros dois, *Total* e *Setup*, consistem na aplicação do *Heurístico de Stinson* (Stinson & Smith, 1982) para as matrizes dos tempos de *setup* das tarefas do *flow shop*. Ríos-Mercado & Bard (1998) apresentaram dois métodos heurísticos denominados *NEHT-RB* e *GRASP*, sendo o NEHT-RB uma derivação do NEH, e Ríos-Mercado & Bard (1999) desenvolveram um algoritmo chamado *HIBRID* e compararam com o já desenvolvido GRASP em problemas em que a razão entre tempos de *setup* e processamento é da ordem de 0,1 a 0,5, sem resultados expressivos. Parthasarathy & Rajendran (1997) desenvolveram um heurístico baseado no *Simulated Annealing*, apresentando um novo procedimento de geração de vizinhança (RIPS), com o objetivo de reduzir o atraso médio total. Rajendran & Ziegler (1997) desenvolveram um heurístico para minimizar o tempo de fluxo ponderado de tarefas, e Das *et al.* (1995) apresentaram algoritmo heurístico construtivo de *saving index* associado a uma seqüência particular de tarefas.

4. Descrição do algoritmo BGaFSA

Considere o problema de Programação *Flow Shop* Permutacional ASDST com os seguintes dados:

- m = número de máquinas (estágios de produção);
- n = número de tarefas;
- p_{ik} = tempo de processamento da tarefa i na máquina k , $i = 1, 2, \dots, n$ e $k = 1, 2, \dots, m$;
- s_{ij}^k = tempo de *setup* da máquina k quando a tarefa j é processada imediatamente após a tarefa i .

É importante salientar que, neste trabalho, adota-se o *anticipatory setup*, ou seja, a preparação de uma máquina k para uma tarefa j é iniciada logo após a liberação da máquina pela tarefa anterior, não necessitando aguardar o término da tarefa j na máquina $(k - 1)$. Esse caso de *setup* também é considerado nos trabalhos de Simons Jr. (1992), Ríos-Mercado & Bard (1998, 1999) e Das *et al.* (1995).

Sejam as seguintes variáveis:

- $\sigma = j_{[1]} j_{[2]} \dots j_{[u]} \dots j_{[n]}$ – seqüência genérica das n tarefas, em que $j_{[u]}$ representa a tarefa que ocupa a u -ésima posição.
- $I_{[u]}^k$ – tempo de espera da máquina k após sua preparação para processar a tarefa na posição u .
- $T^k(\sigma)$ – intervalo de tempo entre início do processamento da primeira tarefa e término da última tarefa na máquina k , para uma seqüência σ das tarefas, que será denominada carga da máquina k .
- $M(\sigma)$ – duração total da programação (*makespan*) para uma seqüência σ das tarefas.

Considerando uma programação das m máquinas segundo uma seqüência σ , o *makespan* e a carga de uma máquina qualquer k são dados por:

$$M(\sigma) = \sum_{k=1}^{m-1} p_{[1]k} + \sum_{i=1}^n p_{im} + \sum_{u=2}^n s_{[u-1][u]}^m + \sum_{u=2}^n I_{[u]}^m$$

$$T^k(\sigma) = \sum_{i=1}^n p_{ik} + \sum_{u=2}^n s_{[u-1][u]}^k + \sum_{u=2}^n I_{[u]}^k$$

em que $p_{[1]k}$ = tempo de processamento da primeira tarefa da seqüência σ na máquina k .

O objetivo do algoritmo BGaFSA é minimizar o *makespan* por meio da otimização de sua máquina gargalo,

em que para o conjunto de n tarefas e uma seqüência genérica σ , a máquina gargalo será aquela que apresentar a maior carga, ou seja, o maior valor de $T^k(\sigma)$, para $k = 1, 2, \dots, m$.

Os parâmetros e funções considerados para o procedimento SA são:

- temperatura inicial $T_1 = 80$ e temperatura final $T_K = 1$ (Buzzo & Moccellini, 1999);
- função de resfriamento e probabilidade de aceitação de movimentos, conforme Osman & Potts (1989);
- vizinhança de inserção, ou seja, para obter uma seqüência vizinha escolhe-se aleatoriamente uma tarefa i e uma posição u , inserindo-a nessa posição;
- número total de iterações (k) definido por meio de análise computacional prévia, com o objetivo de equilibrar a qualidade da solução e o esforço computacional.

O ponto de partida do algoritmo BGaFSA é a identificação inicial do estágio gargalo g , considerando-se somente os tempos de processamento das tarefas, de forma que:

$$\sum_{i=1}^n p_{ig} = \max_{1 \leq k \leq m} \sum_{i=1}^n p_{ik}$$

Uma seqüência inicial das n tarefas é obtida por meio da aplicação do método heurístico construtivo NB" de Gavett (1965) sobre a matriz dos tempos de *setup* do estágio gargalo g . A seguir, sobre essa seqüência inicial aplica-se o procedimento *Simulated Annealing*, para obter uma seqüência das tarefas que minimize a somatória dos tempos de *setup* s_{ij}^g no gargalo.

Seja S^* a melhor seqüência das tarefas fornecida pelo procedimento SA, com *makespan* $M(S^*)$ e estágio gargalo g^* , ou seja

$$T^{g^*}(S^*) = \max_k T^k(S^*)$$

A seqüência S^* , seu *makespan* $M(S^*)$ e o estágio gargalo g^* são armazenados. É necessário verificar se o estágio gargalo identificado inicialmente se mantém ou se houve alteração, em função da ordenação das tarefas segundo a seqüência S^* . Se o gargalo permanecer o mesmo, o algoritmo é encerrado. Caso contrário, aplicam-se novamente o NB" e o procedimento SA sobre a matriz dos tempos de *setup* do atual gargalo. Esse procedimento é repetidamente aplicado até que o último estágio gargalo obtido seja o mesmo de algum ciclo (NB"+SA) anterior. Nota-se que o algoritmo BGaFSA executa R vezes o procedimento (NB"+SA), em que $2 \leq R \leq (m + 1)$. A solução do problema será aquela seqüência S^* com o menor *makespan* entre as R seqüências obtidas.

5. Experiência computacional

Com o objetivo de avaliar seu desempenho, o BGaFSA foi comparado com o algoritmo *Total* desenvolvido por Simons Jr. (1992), por este ser um método bem referenciado e de eficácia comprovada na literatura (Ríos-Mercado & Bard, 1998, 1999; Allahverdi *et al.*, 1999).

Para o ambiente *flow shop* permutacional com ASDST, Allahverdi *et al.* (1999) destacam o método heurístico desenvolvido por Ríos-Mercado & Bard (1998), denominado GRASP, e os métodos *Total* e *Setup* propostos por Simons Jr. (1992). Os autores mencionam que, para tempos de *setup* e processamento identicamente distribuídos, o algoritmo *Setup* é superior ao GRASP. Tal afirmação tem origem no próprio trabalho de Ríos-Mercado & Bard (1998), que utiliza o algoritmo *Setup* como base de comparação. Por outro lado, no trabalho de Simons Jr. (1992) há a indicação de que o algoritmo *Total* pode ser considerado superior ao *Setup*.

Para a comparação do BGaFSA com o *Total*, foram gerados 20 problemas para cada classe $m \times n \in \{5, 10, 15\} \times \{5, 10, 15\}$ com tempos de *setup* das máquinas e processamento das tarefas identicamente distribuídos no intervalo [10, 200]. A relação e a distribuição dos tempos de *setup* e processamento foram adotadas em função de serem consideradas ideais ao algoritmo *Total* (Simons Jr., 1992). Para a comparação foi desenvolvido um *software* em linguagem *Delphi* 6.

As estatísticas para comparar algoritmos foram definidas de acordo com Barros (2002), conforme segue:

- NV (nº de vitórias ou *outperforms*): quantidade de problemas em que um algoritmo foi superior ao outro na medida de desempenho adotada (*makespan*).

- MRPM (Melhoria Relativa Percentual Média): diferença percentual entre os valores do melhor *makespan* obtido pelos algoritmos.
- TMC (Tempo Médio de Computação): tempo computacional médio necessário para a solução do problema, na classe considerada.

Essas estatísticas de avaliação de desempenho comumente são utilizadas em pesquisas reportadas na literatura, sendo também utilizadas nos trabalhos de Simons Jr. (1992) e Das *et al.* (1995).

A Tabela 1 apresenta os resultados da experiência computacional, agrupados para cada classe de problema. Nota-se que o BGaFSA apresentou desempenho superior em todos os problemas de 5 e 10 tarefas, enquanto o algoritmo *Total* foi melhor nos problemas com 15 tarefas. O desempenho do BGaFSA para os problemas de maior porte pode ser melhorado por meio de uma melhor calibração dos parâmetros do procedimento *Simulated Annealing*. Estudos nessa direção estão sendo realizados. Entretanto, a constatação de que o gargalo pode não ser estático no ambiente de produção tratado justifica a aplicação do BGaFSA, em termos de gerenciamento de chão de fábrica.

Em razão das diferenças nas estruturas dos algoritmos, a diferença no tempo computacional já seria esperada, sendo que o tempo de computação do BGaFSA tende a crescer significativamente com o porte do problema, enquanto para o *Total* esse tempo é reduzido e com pequena variabilidade. Por outro lado, os tempos de computação, dentro de certos limites, já deixaram de ser uma restrição de qualidade e, no caso do BGaFSA, não são excessivos.

Tabela 1 – Comparação entre BGaFSA e *Total*.

Classe	BGaFSA			Total		
	NV	MRPM	TMC [s]	NV	MRPM	TMC [s]
5 n x 5 m	17	8,22%	15,83	3	3,03%	0,11
5 n x 10 m	13	7,23%	15,38	7	4,99%	0,17
5 n x 15 m	15	7,45%	17,10	5	7,09%	0,22
10 n x 5 m	13	5,29%	71,16	7	3,93%	0,30
10 n x 10 m	11	3,64%	98,76	9	4,88%	0,40
10 n x 15 m	12	6,19%	112,24	8	3,32%	0,50
15 n x 5 m	6	1,62%	206,76	14	8,31%	0,66
15 n x 10 m	5	2,80%	292,52	15	6,41%	0,75
15 n x 15 m	5	4,54%	405,58	15	2,84%	0,90

5.1 Flutuação do gargalo *versus* otimização local

Um dos aspectos observados na aplicação do BGaFSA é a melhoria no *makespan* entre a solução obtida com a otimização do primeiro recurso identificado como gargalo, denominada otimização local, e a solução obtida pelo BGaFSA, considerando a flutuação do gargalo. A Figura 1 apresenta, comparativamente, as vitórias obtidas com a otimização local e com a aplicação do BGaFSA, em relação ao *Total*.

Observa-se que o conceito difundido na literatura de otimização local no gargalo "tradicional" (quando se considera somente o tempo de processamento das tarefas) pode não ser adequado para obter a melhor solução para

o problema no ambiente ASDST, conforme mostram os resultados da Figura 1.

A Figura 2 apresenta a melhoria porcentual média obtida com o BGaFSA, em relação ao *makespan* da otimização local.

Uma análise específica do comportamento do BGaFSA foi efetuada quanto à ocorrência do gargalo na última máquina (m) do sistema de produção e também com relação à direção de flutuação do gargalo. Na literatura pesquisada, há métodos que consideram a otimização da última máquina do sistema como a ideal para o sistema como um todo (Simons Jr., 1992), ou seja, o gargalo tende a ser a última máquina em função da somatória do tempo de espera entre tarefas na última máquina. A análise experimental indica que a aceitação de tal tendência não pode ser generalizada, conforme os resultados apresentados na Tabela 2.

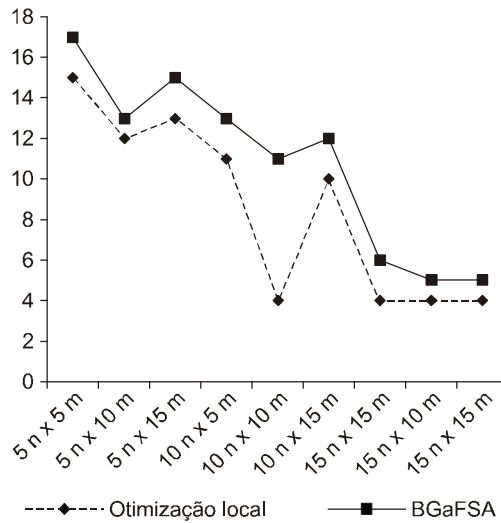


Figura 1 – Número de vitórias sobre o *Total*, considerando a otimização local e o BGaFSA.

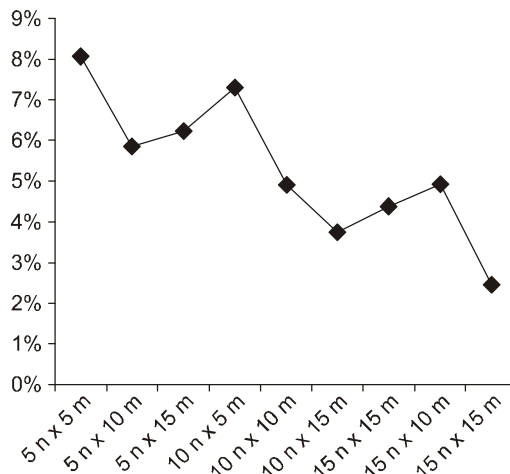


Figura 2 – Melhoria do BGaFSA sobre o *makespan* da otimização local.

Os resultados da Tabela 2 permitem concluir que, em 48% dos problemas, o gargalo flutuou para uma posição anterior ao último estágio de produção. Pode-se observar, também, que o percentual de ocorrências do gargalo final na última máquina do sistema de produção tende a diminuir com o aumento do número de máquinas.

A Tabela 3 apresenta, em termos percentuais, a direção da flutuação do gargalo e a quantidade de problemas em que não houve mudança do gargalo inicial (gargalo estável).

Como pode ser observado, o gargalo pode flutuar para estágios anteriores àquele correspondente ao gargalo inicial, embora com menor intensidade do que a flutuação para estágios posteriores.

6. Conclusões e considerações finais

Este trabalho indica a relevância da análise da flutuação do gargalo para programar a produção em um sistema *flow shop* permutacional com tempos de *setup* assimétricos e dependentes da seqüência. A comparação do algoritmo proposto (BGaFSA) com um excelente método reportado na literatura (*Total*) mostra bom desempenho somente para

problemas de pequeno porte. Tais resultados motivam a continuação dos estudos realizados, com o objetivo de aprimorar o método apresentado. Outros resultados da experiência computacional evidenciam que o gargalo efetivo de sistemas de produção *flow shop* permutacional com ASDST não necessariamente ocorre em seus estágios finais. Além disso, a flutuação do gargalo em função da ordenação das tarefas pode dirigir-se para estágios de produção anteriores àquele com o maior tempo de processamento das tarefas (gargalo inicial). O método BGaFSA mostra que, nesse sistema de produção, o estágio gargalo não deve ser previamente estabelecido, sendo, portanto, resultado da programação da produção.

A programação da produção em ambientes em que os tempos de preparação das máquinas (*setups*) não devem ser incluídos nos tempos de processamento das tarefas, uma vez que são significativamente dependentes da ordenação das tarefas nas máquinas, tende a ser um processo complexo. Em tais ambientes de produção é desejável uma administração diferenciada do estágio gargalo, tendo em vista a possibilidade de sua flutuação e seu impacto no desempenho geral do sistema produtivo.

Tabela 2 – Posição do gargalo final nos problemas em que houve flutuação do gargalo.

	Máquina m	Máquina 1 a (m – 1)
5 n x 5 m	75%	25%
5 n x 10 m	50%	50%
5 n x 15 m	14,3%	85,7%
10 n x 5 m	85,7%	14,3%
10 n x 10 m	64,3%	35,7%
10 n x 15 m	50%	50%
15 n x 5 m	50%	50%
15 n x 10 m	57,1%	42,9%
15 n x 15 m	22,2%	77,8%

Tabela 3 – Direção de flutuação do gargalo.

	Para estágios anteriores	Para estágios posteriores	Estável
5 n x 5 m	0%	40%	60%
5 n x 10 m	5%	35%	60%
5 n x 15 m	5%	30%	65%
10 n x 5 m	5%	30%	65%
10 n x 10 m	5%	65%	30%
10 n x 15 m	5%	55%	40%
15 n x 5 m	0%	20%	80%
15 n x 10 m	0%	35%	65%
15 n x 15 m	15%	30%	55%

Referências Bibliográficas

- ALLAHVERDI, A.; GUPTA, J. N. D.; ALDOWAISAN, T. A review of scheduling involving setup considerations. *Omega*, v. 27, p. 219-239, 1999.
- BAKER, K. R. *Introduction to sequencing and scheduling*. New York: John Wiley and Sons, 1974. Cap. 6, p. 136 -176.
- BARROS, A. D. *Algoritmo metaheurístico para busca do gargalo flutuante em flow shop permutacional com tempos de setup assimétricos e dependentes da seqüência*. 2002. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, USP, São Carlos.
- BUZZO, W. R.; MOCCELLIN, J. V. A influência da temperatura inicial no desempenho de um método híbrido algoritmo genético – simulated annealing para a programação flow shop permutacional. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, INTERNATIONAL CONGRESS OF INDUSTRIAL ENGINEERING, 5., Rio de Janeiro, 1999.
- DAS, S. R.; GUPTA, J. N. D.; KHUMAWALA, B. M. A saving index heuristic algorithm for flow shop scheduling with sequence dependent set-up times. *Journal of the Operational Research Society*, v. 46, p. 1365-1373, 1995.
- FLOOD, M. M. The Traveling-Salesman Problem. *Operations Research* 4, n. 1, p. 61-75, 1956.
- FLYNN, B. B. The effects of setup time on output capacity in cellular manufacturing. *International Journal of Production Research*, v. 25, n. 12, p. 1761-1772, 1987.
- GAVETT, J. W. Three heuristic rules for sequencing jobs to a single production facility. *Management Science*, v. 11, n. 8, p. B166-B176, 1965.
- GOLDRATT, E.; FOX, J. *A meta: um processo de aprimoramento contínuo*. São Paulo: Educador, 1997.
- KIM, S. C.; BOBROWSKI, P. M. Impact of sequence-dependent setup time on job shop scheduling performance. *International Journal of Production Research*, v. 32, n. 7, p. 1503-1520, 1994.
- MOCCELLIN, J. V. Comparison of neighbourhood search heuristics for the flow shop sequencing problem. In: INTERNATIONAL WORKSHOP ON PROJECT MANAGEMENT AND SCHEDULING, 4., 1994, Leuven, Belgium. *Proceedings...* Leuven, Belgium, 1994. p. 228-231.
- OSMAN, I. H. & POTTS, C. N. Simulated annealing for permutation flow-shop scheduling. *Omega*, v. 17, n. 6, p. 551-557, 1989.
- PARTHASARATHY, S.; RAJENDRAN, C. A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs – a case study. *Production Planning & Control*, v. 8, n. 5, p. 475-483, 1997.
- PINEDO, M. *Scheduling: theory, algorithms and systems*. 1. ed. Prentice-Hall, 1995.
- RAJENDRAN, C.; ZIEGLER, H. A heuristic for scheduling to minimize the sum of weighted flow time of jobs in a flow shop with sequence dependent setup times of jobs. *Computers and Industrial Engineering*, v. 33, n. 1-2, p. 281-284, 1997.
- RÍOS-MERCADO, R. Z.; BARD, J. F. Heuristics for flow line problem with setup costs. *European Journal of Operational Research*, v. 110, p. 76-98, 1998.
- RÍOS-MERCADO, R. Z.; BARD, J. F. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times. *Journal of Heuristics*, v. 5, p. 53-70, 1999.
- SIMONS JR., J. V. Heuristics in flow shop scheduling with sequence dependent setup times. *Omega*, v. 20, n. 2, p. 215-225, 1992.
- STEIN, R. E. *The theory of constraints: applications in quality and manufacturing*. Creating the Process of Continuous Profit Improvement. New York: Marcel Dekker. 2. ed. 1997. c. 2, p. 5-16.
- STINSON, J. P.; SMITH, A. W. A heuristic programming procedure for sequencing the static flowshop. *International Journal of Production Research*, v. 20, n. 6, p. 753-764, 1982.

ANALYSIS OF THE BOTTLENECK FLOTATION IN PERMUTATION FLOW SHOP WITH ASYMMETRIC SEQUENCE DEPENDENT SETUP TIMES

Abstract

This paper introduces a Simulated Annealing (SA) heuristic for permutation flow shop with asymmetric sequence dependent setup times (ASDST), focusing a main constraint to the output of a production system, which is the bottleneck production stage. The proposed algorithm, named "Simulated Annealing metaheuristic for floating bottleneck search" (BGaFSA), analyzes a possible change in the bottleneck stage due to the job sequencing. The SA procedure searches for the effective bottleneck stage of the production system. The performance measure is the makespan.

Key-words: *production scheduling, flow shop sequencing, setup times, bottleneck environment.*

A pesquisa relatada neste artigo teve o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).