

Sistemas Flexíveis de Manufatura: escalonamento de Job-Shops sem Deadlock

Berenice Camargo Damasceno

Departamento de Matemática, Faculdade de Engenharia de Ilha Solteira – FE/IS, UNESP
15385-000, Ilha Solteira, SP
E-mail: berenice@mat.feis.unesp.br

Rafael Martins de Oliveira

Departamento de Engenharia Mecânica, Faculdade de Engenharia de Ilha Solteira – FE/IS, UNESP
15385-000, Campus Ilha Solteira, Ilha Solteira, SP
E-mail: rafael.martinsoliveira@gmail.com

RESUMO

A competitividade e a globalização mudaram os rumos da economia. Na produção industrial, a palavra-chave é competitividade. Para ser competitiva, uma empresa deve automatizar, integrar e ser flexível objetivando melhorar sua produtividade, a qualidade dos seus produtos, o tempo e o custo da produção. Atualmente, com novas tecnologias, surgiram sistemas produtivos que apresentam flexibilidade, isto é, que podem ser usados na fabricação de diferentes componentes ou produtos. Esses sistemas são os Sistemas Flexíveis de Manufatura - FMS (do inglês *Flexible Manufacture Systems*).

Este trabalho trata do escalonamento sem *deadlock* de uma classe de FMS com múltiplos recursos. A classe escolhida são os job-shops. Um *job-shop* consiste de um conjunto de tarefas (*jobs*) que deve ser processado e cada *job* necessita de uma seqüência de operações. Cada operação pode necessitar mais de um recurso. O tempo de processamento de cada operação é determinístico. O problema de escalonamento consiste no sequenciamento das operações considerando os recursos disponíveis, evitando os *deadlocks* e minimizando o tempo total de realização das tarefas (*makespan*).

Segundo Tanenbaum [3], existem quatro condições para que se ocorra um *deadlock*, sendo que todas devem ser respeitadas para o surgimento do *deadlock*. Essas quatro condições são:

- 1ª) Não exclusão mútua: em um determinado instante cada recurso, está em uma de duas situações: ou associado a uma única operação ou disponível, isto é, exclusão mútua;
- 2ª) Posse e espera: operações que, em determinado instante, retêm recurso concedidos anteriormente podem requisitar novos recursos;
- 3ª) Preempção: recursos concedidos previamente à uma operação poderiam ser tomados à força dessa operação;
- 4ª) Espera circular: deve existir um encadeamento circular de duas ou mais operações; cada uma delas encontra-se à espera de um recurso que está sendo usado pelo membro seguinte dessa cadeia.

Logo, um método para se evitar o *deadlock* é garantir que uma dessas condições não ocorra.

Este trabalho pretende propor uma solução para o problema de escalonamento de um job-shop considerando a redução do tempo máximo de fabricação (*makespan*) e a não ocorrência das situações de *deadlock*.

Assim, um problema de escalonamento de um *job-shop* deve considerar a produção de mais de um tipo de um *job*, sendo que cada *job* necessita de uma determinada seqüência de operações $\{(M_{j_1, p_{j_1}}, (M_{j_2, p_{j_2}}, \dots, (M_{j_k, p_{j_k}})\}$. Onde $(M_{j_k, p_{j_k}})$ é a operação *jk* do job *j* e ordem *k*, utilizando a

máquina (ou recurso) M_{jk} e com tempo de processamento p_{jk} . Deste modo, o escalonamento em *job shop* segue a partir do sequenciamento das várias operações, dos vários *jobs* dentro de um FMS, obedecendo a ordem de operações dentro de cada *job* e, evitando-se a situação de *deadlock* objetivando a minimização do *makespan*.

Como passo inicial, utilizamos uma heurística mais conservadora, proposta por Damasceno[1], onde uma operação inicia somente após o término da anterior de um mesmo *job* e um *job* só começa após o término do anterior. Deste modo, evita-se o *deadlock*, pois os recursos, que podem ou não ser comuns aos *jobs*, não estarão em situação de impasse. Ou seja, quando uma operação requisitar um determinado recurso, este estará disponível e será liberado. Assim, tal heurística garante a não ocorrência de *deadlock*. Contudo, em relação ao *makespan*, essa heurística se mostrou menos eficiente. Essa constatação pode ser facilmente verificada, pois num sistema com vários recursos, alguns deles podem estar livres durante o processamento de algumas operações. E a simulação dessa heurística mostrou *makespans* que aparentemente poderiam ser minimizados [2].

Deste modo, o próximo passo é propor uma heurística menos conservadora que consiga gerenciar a realização simultânea de diferentes operações visando minimizar o *makespan* sem, contudo, permitir a existência de *deadlocks*. Para garantir a não ocorrência de *deadlocks* a condição escolhida (dentre as mencionadas anteriormente) a ser evitada é a de preempção. Sendo assim, antes do término de uma operação, os recursos necessários para a próxima operação do mesmo *job* não estarão disponíveis e os demais recursos remanescentes dessa operação só estarão disponíveis no início da operação seguinte.

Este trabalho propõe, então uma segunda heurística onde os *jobs* são escalonados um após o outro e para cada *job* o escalonamento consiste na minimização do *makespan* considerando que a condição de preempção seja evitada. As simulações iniciais [2] indicam que essa heurística propõe escalonamentos com *makespan* menores que os da heurística conservadora, e sem *deadlocks*.

Palavras-chave: *Escalonamento, Sistemas Flexíveis de Manufatura, Job-shop, Deadlock.*

Referências

- [1] Damasceno, B.C. “Ordonnancement des systèmes de production multi-ressources avec la prise en compte de blocage”. Tese de Doutorado (*em francês*), Universidade de Metz, França, 1999.
- [2] Oliveira, R.M.; Damasceno, B.C. Estudo de casos de problemas de escalonamento. Pré-print, 2009.
- [3] Tanenbaum, A. S. “Sistemas Operacionais Modernos”, Pearson Prentice Hall, 2ª edição, pp. 117-138, 2008.