

BRANCH-AND-BOUND PARA UM PROBLEMA DE SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS COM DESGASTE

Vívian Ludimila Aguiar Santos
Thales Francisco Mota Carvalho
André Gustavo Santos
José Elias Claudio Arroyo

Universidade Federal de Viçosa
Campus Universitário, 36570-900, Viçosa-MG, Brasil
vivian.santos@ufv.br, thales.mota@ufv.br, andreufv@gmail.com, jarroyo@dpi.ufv.br

RESUMO

Pode-se perceber o quanto a oferta de materiais tem crescido nos últimos anos devido à ampliação de vendas pela internet. Para auxiliar as empresas a continuarem firmes em seus mercados, surgiram eficientes sistemas de planejamento e controle da produção. É neste contexto que se encontram os problemas de sequenciamento de tarefas em máquinas paralelas, os quais visam reduzir o tempo de processamento final em cada máquina para assegurar a alocação eficiente dos recursos. Uma variante destes problemas considera os desgastes que as tarefas causam nas máquinas ao serem processadas. Este desgaste diminui o desempenho das máquinas levando ao aumento do tempo do processamento das tarefas ao longo do tempo. O objetivo deste problema é encontrar as sequências de processamento de tarefas em cada máquina de tal maneira que os desgastes das máquinas sejam reduzidos e consequentemente minimizar o tempo máximo de conclusão de todas as tarefas, conhecido como *makespan*. Encontram-se na literatura um modelo matemático e heurísticas para resolver este problema. Tal problema é NP-Difícil para mais de uma máquina e seu modelo matemático é não-linear, o que impossibilita encontrar soluções ótimas utilizando *solvers* de otimização como o IBM CPLEX. Sendo assim, este trabalho propõe o desenvolvimento de um algoritmo exato para determinar soluções ótimas do problema.

O algoritmo é baseado em uma busca em profundidade em árvore, onde cada nó interno representa uma solução parcial e cada nó folha representa uma solução completa. Ao utilizar este tipo de busca para explorar todas as combinações possíveis de uma instância de 20 tarefas e 4 máquinas, seria necessário visitar $\sim 1.46 \times 10^{12}$ nós, por exemplo. Para diminuir a quantidade de nós visitados, propomos um algoritmo de busca baseado em *branch-and-bound*. Esta técnica é derivada do algoritmo de busca em profundidade e representa um refinamento da busca por força bruta (enumeração completa), no qual boa parte das soluções pode ser eliminada sem serem explicitamente examinadas. Neste caso, aplicamos uma poda: após encontrar uma solução completa, os próximos nós a serem gerados, em qualquer máquina, que obtiverem o tempo de processamento final maior do que o *makespan* da solução completa poderão ser podados. Além disso, foi feito um melhoramento que primeiramente atribui a tarefa à máquina que possuirá o menor tempo de processamento final após a adição desta tarefa e posteriormente atribui esta tarefa às outras máquinas. Foram realizados testes computacionais com programação paralela com 600 instâncias disponibilizadas na literatura. Entre estas, foi possível encontrar a solução ótima de 307 instâncias em um período de três semanas. Em todas as instâncias percebeu-se que a quantidade de nós explorados reduziu aproximadamente 99% com o uso das podas. Pode-se concluir que os cortes possibilitaram o algoritmo executar em tempo hábil na maioria das instâncias, diminuindo o número de nós visitados e consequentemente o tempo gasto para encontrar a solução ótima.

PALAVRAS CHAVE. *Scheduling*, Busca em árvore, *branch-and-bound*.

OC - Otimização Combinatória.