

Problemas de sequenciamento e balanceamento de linhas de produção são encontrados em várias aplicações práticas, como, por exemplo, no planejamento de linhas de produção, na atribuição de tarefas a servidores e no atendimento de clientes. Tanto no problema de sequenciamento como no de balanceamento, temos de alocar um conjunto de tarefas a máquinas de forma que um objetivo do planejamento seja otimizado. Dentro dos objetivos mais comuns na prática, temos a minimização do tempo de fluxo ponderado, o total de atraso, o tempo de ciclo e o número de máquinas usadas.

Os problemas supracitados diferem no modelo das máquinas: ao passo de que em problemas de sequenciamento tenhamos um conjunto de máquinas paralelas, nos problemas de balanceamento temos uma sequência linear de máquinas. Além das diferenças no modelo das máquinas, os dois problemas possuem um grande número de variações com diferentes restrições, sendo as mais importantes aquelas relacionadas à homogeneidade das máquinas e à presença de uma ordem parcial para execução das tarefas.

Até o presente momento, o objetivo do trabalho consistiu em estudar, implementar, estender e comparar soluções existentes para as variantes SALBP-1 e BPP-P, focando, em particular, no uso de programação dinâmica. A primeira das variantes – *Simple Assembly Line Balancing Problem* – trata da alocação de um conjunto de tarefas com restrições de precedência não-estrita (i.e., se a tarefa  $a$  precede  $b$ ,  $a$  aparece junto ou antes de  $b$ ) no menor número possível de máquinas em sequência, dado um tempo de ciclo que deve ser obedecido. Na segunda – *Bin Packing Problem with Precedence Constraints* – temos de alocar tarefas com relação de precedência estrita (se tarefa  $a$  precede  $b$ ,  $a$  aparece sempre antes de  $b$ ) no menor número de *bins* possível sem exceder a capacidade destes. Para esta variante em particular, nosso algoritmo é o primeiro a abordá-la utilizando programação dinâmica.

Utilizando a implementação atual de nosso algoritmo conseguimos resolver 54.28% (SALBP-1) e 7.81% (BPP-P) das 269 instâncias mais utilizadas na literatura, limitando o tempo de execução de nosso algoritmo em 1 hora. Em algumas dessas instâncias, nosso algoritmo possui tempo de execução semelhante ao estado-da-arte da área (que, normalmente, utiliza a técnica *branch-and-bound* ao invés de programação dinâmica). Até o presente momento, o principal *milestone* a ser vencido diz respeito à otimização no uso da memória (tendo em vista que, nos testes realizados, tivemos ocorrência maior de estouro de memória do que do tempo de execução). Já identificamos otimizações que podem diminuir o uso da memória em pelo menos 25 vezes e que certamente terão um impacto considerável nos próximos testes que serão executados.