UFOP – Universidade Federal de Ouro Preto
DECOM – Departamento de Computação
Disciplina: Algoritmos e Estruturas de Dados III - CIC 210 – AEDSIII
Página: http://www.decom.ufop.br/prof/haroldo/disciplinas/aedsIII/
Professor: Haroldo Gambini Santos

Trabalho – *Branch* & *Bound*

Tarefa:

Implementar um algoritmo de *Branch* & *Bound* para o problema da mochila 0/1. A implementação deve ser experimentalmente avaliada com instâncias disponibilizadas na página da disciplina.

Considerações sobre a avaliação (em ordem de importância):

- a implementação deve ser correta:
  - soluções incluindo um conjunto de itens que não extrapole a capacidade da mochila;
  - o lucro deve corresponder exatamente aos itens inclusos na mochila;
  - a implementação deve, sempre que houver tempo suficiente, encontrar a solução ótima – ou seja, não deve abortar prematuramente oferecendo como solução ótima uma solução de custo pior.
- a implementação deve ser eficiente:
  - o algoritmo deve oferecer a solução ótima no menor tempo possível;
  - quando a explosão combinatória impedir o algoritmo de provar rapidamente qual é a solução ótima (caso comum), o algoritmo deve, sempre que possível, oferecer uma boa solução rapidamente e continuar a sua busca na árvore até a prova da solução ótima ou a extrapolação de um tempo limite.
  - considerando eficiência do algoritmo, pontos comuns de melhora na implementação de métodos de B&B são:
    - implementação de uma ou mais heurísticas boas: melhora do limite inferior – dica: heurísticas gulosas oferecem rapidamente uma solução factível; podem ser considerados vários critérios gulosos na construção de mais de uma heurística, o que pode aumentar a robustez do método (eventualmente uma heurística funciona bem para uma instância e para outra não, então ter mais de uma executando em um ou mais nós da árvore pode melhorar o desempenho do programa); adicionalmente, pode ser inserida uma busca local na heurística (tentar inserção de algum outro item que ficou de fora ou trocar um item inserido por outro, por ex.), caso o método fique computacionalmente pesado o mesmo pode ser executado somente no início da busca;
    - escolha sábia de em quais variáveis será feito o *branch* primeiro, definindo um caminhamento na árvore que leve rapidamente a boas soluções;
    - obtenção rápida de um bom limite superior – obtido através de alguma relaxação.

Experimentos:

Considerando os itens acima, os experimentos devem avaliar:

- número de instâncias em que o algoritmo concluiu toda a busca em um

tempo pequeno (menor que 10 minutos, por exemplo);

- número de instâncias em que o algoritmo encontrou a solução ótima rapidamente (e não necessariamente concluiu a busca, visto que o Limite Inferior ainda pode ser menor que o Limite Superior) (menos de 2 minutos, por exemplo).

Para facilitar a execução posterior desses experimentos, o programa deve ser facilmente executado com diferentes configurações:

- executar até a conclusão da busca ou a extrapolação de um tempo limite (parâmetro de entrada);
- executar e parar imediatamente quanto encontrar uma solução com lucro igual ao da solução ótima (parâmetro de entrada).

Considerações adicionais:

Disponíveis na página da disciplina, na subseção **Trabalhos**.