

Trabalho Prático II

Projeto e Análise de Algoritmos

PPGCC - DECOM - UFOP - 2020/PLE

Introdução

O trabalho consiste na implementação de algoritmos de enumeração implícita (*Branch-and-Bound*) e heurísticas para um problema de otimização combinatória que será descrito a seguir.

O Problema da Mochila com Conflitos

O Problema da Mochila com Conflitos (PMC), considera os seguintes dados de entrada:

c capacidade da mochila

I conjunto de itens

l_i lucro do i -ésimo item

p_i peso do i -ésimo item

A conjunto de pares ordenados (i, j) , indicando que os itens i e j não podem ser alocados em conjunto

O problema consiste em definir o subconjunto de lucro máximo que respeite capacidade da mochila e que não contenha nenhum par de itens conflitantes. Formalmente:

maximize:

$$\sum_{i \in I} l_i \cdot x_i \quad (1)$$

sujeito a:

$$\sum_{i \in I} p_i \cdot x_i \leq c \quad (2)$$

$$x_i + x_j \leq 1 \quad \forall (i, j) \in A \quad (3)$$

$$x_i \in \{0, 1\} \quad \forall i \in I \quad (4)$$

Material a Entregar

Resolvedor Eficaz usando *Branch-and-Bound* e heurísticas para o PMC

Como visto em aula, a implementação de um resolvedor eficaz com a técnica de Branch-and-bound depende da sua customização adequada para o problema abordado. Alguns pontos são de fundamental importância para que o mesmo apresente um bom desempenho:

branching a seleção da próxima variável a ser fixada e em que ordem os valores serão fixados;

limites a poda efetiva da árvore acontece quando **limites apertados** são obtidos; para isso, deve-se determinar uma **relaxação adequada**, para a produção de limites superiores bons; a produção de limites inferiores bons pode ser obtida com a execução de **heurísticas**, que podem ser disparadas tanto no nó raiz quanto em soluções parciais também.

A implementação de uma customização efetiva do algoritmo depende da execução de **experimentos sistemáticos** para medição do impacto de diferentes escolhas para cada um desses pontos. O aluno deve entregar um relatório (ver seção seguinte), demonstrando que a definição dos componentes e estratégias incluídas em seu algoritmo foram baseados nessa experimentação.

Avaliação da implementação

corretude produção de soluções que respeitam as restrições do problema em todas as instâncias de teste; conformidade com formatos de entrada e saída e regras técnicas;

desempenho capacidade de executar rapidamente e produzir soluções comprovadamente ótimas na maior parte dos casos; para instâncias maiores, onde o algoritmo tenha terminado por limite de tempo, será avaliada a qualidade da melhor solução encontrada durante a busca.

Aspectos Técnicos

O código pode ser implementado em qualquer linguagem de programação, recomenda-se a utilização da linguagem em que o aluno sinte-se mais **produtivo** para permitir experimentação avançada.

O código deve receber como parâmetro de entrada o nome do arquivo com os dados do problema deve executar por um limite de tempo dependente da máquina. Para verificar qual o limite de tempo na sua máquina baixe o pacote de *Benchmark* disponível em https://www.utwente.nl/en/eemcs/dmmp/hstt/itc2011/benchmarking/Benchmark_ITC3.zip e execute o programa correspondente ao seu sistema operacional.

O programa deve receber dois parâmetros, indicando o arquivo com os dados do problema e o arquivo onde será salva a solução gerada.

Formato de entrada O programa deve ler arquivos, cujo nome deve ser especificado como argumento do programa, no seguinte formato, contendo os dados de entrada no seguinte formato:

```
c
n
l1
l2
...
ln
p1
p2
...
pn
m
i1 j1
i2 j2
...
im jm
```

Onde c é a capacidade da mochila, n é o número de itens, l_1, \dots, l_n são os lucros, p_1, \dots, p_n são os pesos, m indica o número de pares conflitantes, listados em $i_1, j_1 \dots i_m, j_m$.

Formato de saída

A solução gerada deve ser gerada em um arquivo no seguinte formato:

```
lucro totalItems item1 item2 ... itemTotalItems
```

Exemplo

Um pequeno exemplo de instância (toy1) do problema pode ser baixado aqui. Este problema apresenta os seguintes dados:

```
c = 15
n = 8
l = [10, 8, 4, 6, 8, 6, 9, 6]
p = [3, 4, 4, 5, 7, 9, 9, 10]
A = {(1, 4), (2, 4), (2, 5), (2, 7), (4, 7), (4, 8), (5, 6), (5, 7), (5, 8), (6, 7), (6, 8),
      (7, 8)}
```

A Figura 1 ilustra os conflitos entre nós.

A solução ótima para esse problema tem lucro 22 e inclui os itens 1, 2 e 3.

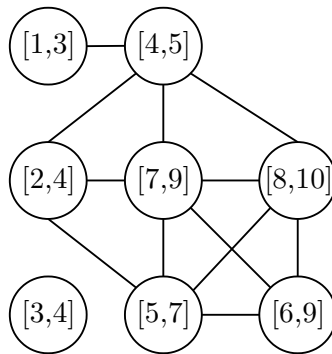


Figure 1: Conflitos da instância toy1, cada nó exibe seu o índice e peso de cada nó.

Relatório

O relatório deve descrever o algoritmo desenvolvido, bem como incluir os resultados experimentos e análises sobre os mesmos. As recomendações do curso sobre projeto e relatórios de experimentos computacionais devem ser seguidas.