

LABORATÓRIO 5 - 12/JUNHO/2017

st-Caminho com Coleta de Prêmios

Dado grafo orientado $G = (N, A)$, vértices $s, t \in N$, custos nos arcos $c : A \rightarrow \mathbb{R}^+$, valores (prêmios) nos vértices $\pi : N \rightarrow \mathbb{R}^+$, encontrar caminho orientado $P = (N_P, A_P)$ de s a t em G tal que $\sum_{v \in N_P \setminus \{s, t\}} \pi(v) - \sum_{a \in A_P} c(a)$ é máximo.

Sobre a Resolução:

1. Seu programa deve ter duas rotinas com os seguintes cabeçalhos:

```
int <nome_da_rotina>(listdigraph &g, Node s, Node t,  
    vector<ListDigraph::Node> &path, double &LB, double &UB, long maxtime)
```

onde g é um grafo orientado, s e t são dois nós distintos do grafo, $path$ é um vetor que contém a sequência de nós do caminho, começando em s e terminando em t . Os parâmetros LB e UB devem ser atualizados com os melhores limitantes inferior e superior, respectivamente, encontrados na sua rotina. O parâmetro $maxtime$ é o tempo máximo em segundos que sua rotina pode ficar executando. No caso da rotina ser uma heurística ou metaheurística, o valor de LB pode voltar como 0.

A rotina deve retornar um valor 0, se não encontrou solução viável, 1 se encontrou solução ótima (garantidamente ótima) e 2 se encontrou solução heurística. Caso seu algoritmo exato não tenha conseguido obter solução ótima, deve devolver valor 2.

2. Uma rotina deve ser a implementação de uma metaheurística para deve se chamar `prize_collecting_st_path_heuristic`. Pode ser utilizando uma das técnicas para metaheurísticas vistas em aula.
3. Outro módulo deve ser um algoritmo exato, por programação linear inteira (é possível resolver este problema sem utilizar planos de corte) e deve se chamar `prize_collecting_st_path_pli`. Para agilizar a resolução do *solver* de programação linear inteira, você deve obter uma solução heurística e entrar com um valor de *cutoff*.
4. Você receberá um programa principal, que já lê uma entrada para o problema e imprime o valor da solução que você gerar, e você deve implementar suas rotinas em um módulo/arquivo adicional.
5. Você deve entregar seu código bem documentado e modularizado de maneira que fique fácil para ser entendido.
6. Você deve entregar um relatório contendo detalhamento dos programas implementados e também com experimentos computacionais da execução dos seus programas.
7. Prazo de entrega: 26/Junho/2017, 10hs.
8. Este laboratório tem peso 3 (mas a dificuldade não é maior que dois laboratórios comuns).