

# Tópicos Especiais em MI (CC0347) - Trabalho 1

Considere o conjunto  $N = \{1, 2, \dots, n\}$  e o problema da mochila 0-1 definido a seguir,

$$\max \sum_{i \in N} p_i x_i \quad (1)$$

s.a

$$\sum_{i \in N} w_i x_i \leq c \quad (2)$$

$$x_i \in \{0, 1\}, \text{ para } i \in N. \quad (3)$$

onde  $p_i \in \mathbb{Z}_+$ ,  $w_i \in \mathbb{Z}_+$  para  $i \in N$ ,  $c \in \mathbb{Z}_+^*$ .

Considere agora os dois algoritmos a seguir:

**Algoritmo 1 (Dantzig relaxado)** Ordene em ordem não-crescente os itens de acordo com a relação  $\frac{p_i}{w_i}$ ,  $i \in N$ , inserimos na mochila os itens  $i \in K$ , tal que  $K \subseteq N$  e  $\sum_{i \in K} w_i \leq c$ . Seja  $h \in N$  o item tal que  $\sum_{i \in K} w_i + w_h > c$ , definimos  $x_h$  como item de parada. A solução do KP é dada por  $x_i = 1$  para  $i \in K$ ,  $x_h = \frac{(c - \sum_{i \in K} w_i)}{w_h}$ , onde  $h = |K| + 1$  e  $x_i = 0$  para  $i \in N \setminus (K \cup \{h\})$ .

**Algoritmo 2 (Dantzig viável)** Ordene em ordem não-crescente os itens de acordo com a relação  $\frac{p_i}{w_i}$ ,  $i \in N$ , inserimos na mochila os itens  $i \in K$ , tal que  $K \subseteq N$  e  $\sum_{i \in K} w_i \leq c$ . A solução do KP é dada por  $x_i = 1$  para  $i \in K$  e  $x_i = 0$  para  $i \in N \setminus K$ .

O Algoritmo 1 pode ser utilizado para resolver o problema da mochila 0-1 para o caso relaxado, ou seja,  $x_i \in [0, 1]$  para  $i \in N$ .

O Algoritmo 2 pode ser utilizado para encontrar uma solução viável, limite inferior, para o problema da mochila 0-1.

Considere as instâncias armazenadas na pasta **instances\_knapsack** do repositório github da disciplina e execute as seguintes atividades:

1. Implemente em Python o Algoritmo 1 para resolver as instâncias em questão.
2. Implemente em Python o Algoritmo 2 para resolver as instâncias em questão.
3. Modele e resolva as instâncias em questão como um MIP usando o Gurobi API Python.

Formato das instancias:

```
n # número de itens

p_{1} p_{2} ... p_{n} # beneficios dos itens

c # capacidade da mochila

w_{1} w_{2} ... w_{n} # pesos dos itens
```

**Observação:** Os códigos devem ser disponibilizados no github(do aluno).