Atividade Prática 1 - Algoritmos de Busca

- Avaliar o algoritmo Hill Climbing para as bases P01 a P07;
- Desenvolver a função de aptidão knapsack no Mlrose;
- Apresentar a melhor solução encontrada e comparar com a melhor solução global disponível para a base de dados

Responsável: Marcos Angelo Cemim

```
from urllib.request import urlopen
import numpy as np
import six
import sys
sys.modules['sklearn.externals.six'] = six
import mlrose
import time
import warnings
warnings.filterwarnings("ignore")
```

```
#Iterates over 7 bases
for _ in range(1,8):

    base = f'p0{_}'
    # Assign values of current base to variables
    c = int(urlopen(f'https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/{base}_c.txt').read().decode('utf-8').split()
    w = [int(x) for x in urlopen(f'https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/{base}_w.txt').read().decode('ut
    p = [int(x) for x in urlopen(f'https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/{base}_p.txt').read().decode('ut
    s = [int(x) for x in urlopen(f'https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/{base}_s.txt').read().decode('ut

    # Print to check mistakes
    # print(f'{"*"*15} Base: {base} {"*"*15}')
    # print(f'Capacity: {c}')
    # print(f'Weight: {w}')
    # print(f'Profit: {p}')
    # print(f'Optimal Selection: {s}')

    # Define fitness function (total profit = solution_array * profit_array) . If total weight > capacity, penalizes returnin
    def fn_fitness(solution):
        if sum(np.multiply(solution, w).tolist()) <= c:
            return sum(np.multiply(solution, p).tolist())
        else:
            return 1

    # Assign fitness function to mlrose format
    fitness = mlrose.CustomFitness(fn_fitness)

    # Define problem
    problema = mlrose.DiscreteOpt(length = len(s), fitness_fn = fitness,
                                  maximize = True, max_val = 2)

    # Run "Hill Climb" algorithm
    start_time_hc = time.time()
    best_fit_hc = 0
    len_curve_hc = 0
    while best_fit_hc < sum(np.multiply(s, p).tolist()):
        solution_hc, best_fit_hc, curve_hc = mlrose.hill_climb(problema, restarts=10, curve=True)
        len_curve_hc += len(curve_hc)
    end_time_hc = time.time()

    # Results
    print(f' Base P0{_} '.center(98, '*'))
    print(f"Algorithm:       Hill Climb")
    print(f'Solutions Tried: {len_curve_hc}')
    print(f'Fitness Value:   {best_fit_hc:.0f}')
    print(f'Solution found:  {solution_hc.tolist()}')
    print(f'--------------------------------------')
    print(f'Best Fitness:    {sum(np.multiply(s, p).tolist()):.0f}')
    print(f'Best Solution:   {s}')
    print(f'--------------------------------------')
    print(f'Time (ms):       {1000 * (end_time_hc - start_time_hc):.4f}')
    print(f'Array Size:      {len(s)}')
    print()


```

```
****************************************** Base P01 ******************************************
Algorithm:       Hill Climb
Solutions Tried: 6
Fitness Value:   309
Solution found:  [1, 1, 1, 1, 0, 1, 0, 0, 0, 0]
--------------------------------------
Best Fitness:    309
Best Solution:   [1, 1, 1, 1, 0, 1, 0, 0, 0, 0]
--------------------------------------
Time (ms):       2.0020
Array Size:      10

****************************************** Base P02 ******************************************
Algorithm:       Hill Climb
Solutions Tried: 7
Fitness Value:   51
Solution found:  [0, 1, 1, 1, 0]
--------------------------------------
Best Fitness:    51
Best Solution:   [0, 1, 1, 1, 0]
--------------------------------------
Time (ms):       1.0908
Array Size:      5

****************************************** Base P03 ******************************************
Algorithm:       Hill Climb
Solutions Tried: 9
Fitness Value:   150
Solution found:  [1, 1, 0, 0, 1, 0]
--------------------------------------
Best Fitness:    150
Best Solution:   [1, 1, 0, 0, 1, 0]
--------------------------------------
Time (ms):       1.5182
Array Size:      6

****************************************** Base P04 ******************************************
Algorithm:       Hill Climb
Solutions Tried: 13
Fitness Value:   107
Solution found:  [1, 0, 0, 1, 0, 0, 0]
--------------------------------------
Best Fitness:    107
Best Solution:   [1, 0, 0, 1, 0, 0, 0]
--------------------------------------
Time (ms):       1.9996
Array Size:      7

****************************************** Base P05 ******************************************
Algorithm:       Hill Climb
Solutions Tried: 23
Fitness Value:   900
Solution found:  [1, 0, 1, 1, 1, 0, 1, 1]
--------------------------------------
Best Fitness:    900
Best Solution:   [1, 0, 1, 1, 1, 0, 1, 1]
--------------------------------------
Time (ms):       1.9903
```