Universidade Federal de Viçosa – *Campus* Florestal

CCF 480 - Meta-heurísticas

Trabalho Prático I

Professor: Marcos Henrique Soares Mendes

Gabriel Morais Reis - 3497

Florestal
2021

1) Introdução

A presente documentação tem por objetivo discutir as implementações e resultados obtidos para o trabalho prático 1 da disciplina de meta-heuristicas. Para a realização desse trabalho, utilizou-se a linguagem python para implementação dos algoritmos e geração dos gráficos pedidos. Os arquivos com a implementação de cada algoritmo estão separados, sendo hc.py o correspondendo ao hill climbing e ils.py correpondente ao iterated local search. O arquivo para geração das métricas e gráficos solicitados também foi feito separado e é o results.py

2) Hill Climbing

A implementação do algoritmo hill climbing foi feita de acordo com o pseudo-código encontrado nos slides da disciplina e no livro "Essentials of Metaheuristics" de Sean Luke. Assim como nos pseudo-códigos referenciados, o algoritmo é dividido em 3 partes, sendo elas inicialização, modificação e avaliação/seleção.

Para modificar o intervalo de cada uma das váriaveis de decisão basta modificar o valor da variável `limits`, na qual a coluna 0 representa o limite inferior e a coluna 1 representa o limite superior, enquanto a linha 0 se refere à variável x e a linha 1 à variável y.

Ao final da execução do algoritmo, o valor final obtido para a função objetivo, assim como os valores de x e y que geram esse valor, são salvos em um arquivo separado para que depois seja feita as análises solicitadas.

2.1) Função `evaluate`

A função evaluate no código desenvolvido é a responsável por calcular o resultado da função objetivo ao receber um par de valores para as variáveis de decisão, portanto, para que o código possa ser utilizado para diferentes funções objetivos, basta modificar o cálculo retornado pela função `evaluate`.

2.2) Inicialização

A inicialização do algoritmo ocorre entre as linhas 19 e 22. O valor de r (número aleatório utilizada para a inicialização conforme pseudo-código) foi definido como sendo um número aleatório entre 0 e 1 para todas as variáveis de decisão. Assim, para cada váriavel de decisão, primeiro geramos um número aleatório entre 0 e 1 e depois disso o utilizamos em conjunto aos limites superiores e inferiores para essa variável de decisão para gerar a solução inicial, que será armazenada no vetor `v`. Depois de gerada essa solução, chamamos a função `evaluate` para saber qual o valor da função objetivo dada essa combinação.

2.3) Modificação

A etapada de modificação no algoritmo ocorre dentro da função `tweak`. Como p (probabilidade de adicionar ruído) foi considerado como sendo 1, temos que todos os elementos do vetor `v` serão modificados. Assim, ao entrar no loop definido, teremos para cada elemento dentro do vetor `v` a adição de um número aleatório escolhido uniformemente

`n`, que é um número entre `-r[i]` e `+r[i]`, onde `r` é o valor definido para o ruído de cada elemento do vetor `v`.

A definição de `r` é dada da seguinte forma: é feita a subtração do limite superior com o inferior para cada elemento de `v` e, então, o resultado dessa subtração é divido por um valor arbitrário definido na variável `rate`. A variável `rate`, para a questão número 1, teve seu valor definido em 5, já para a questão número 2, o valor foi definido em em 20

2.4) Avaliação/seleção

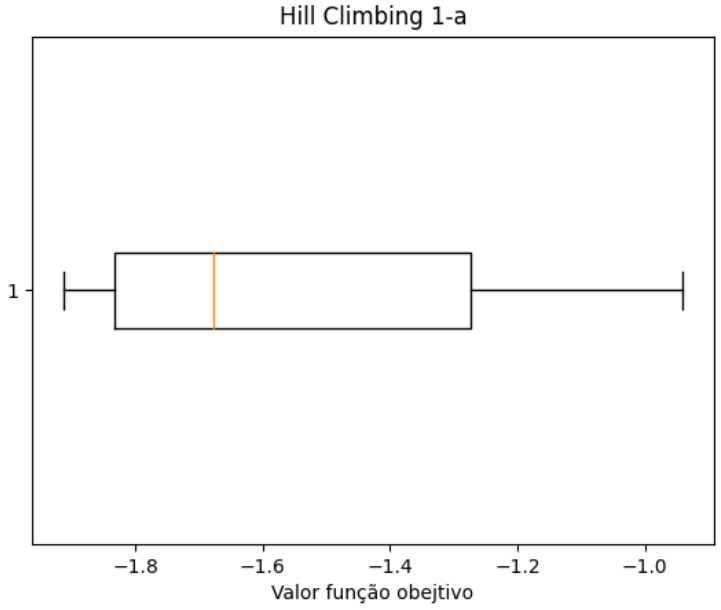
Nessa parte, temos um loop que irá executar 1000 vezes (valor arbitrário utillizado). Para cada passada desse loop teremos a modificação dos valores de `v` salvas em um vetor temporário auxiliar e depois a avaliação do novo valor objetivo dada essas modificações em `v`. Caso as modificações reduzam o valor da função objetivo, visto que estamos tratando de um caso de minimização, os valores do vetor `v` e da solução ótima são atualizados.

3) Resultados

Os resultados apresentados serão arredondas para 10 casas decimais.

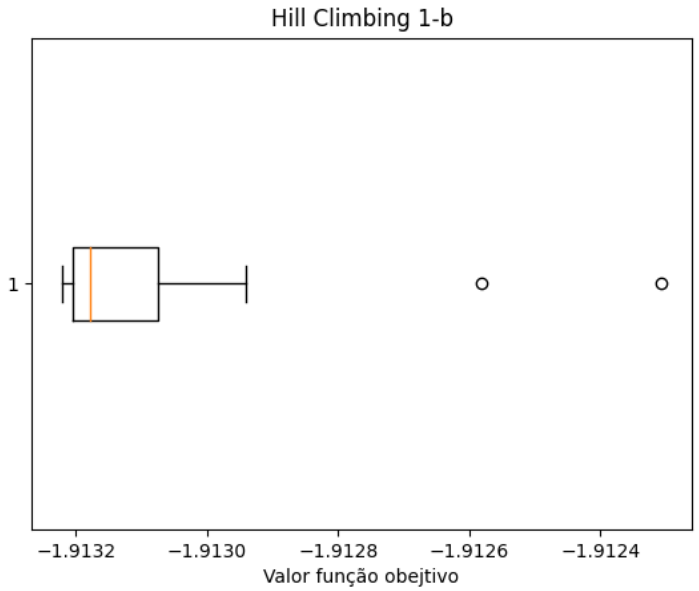
3.1) Questão 1-a

Algoritmo	Mínimo	Máximo	Média	Desvio-padrão
HC	-1.9131970579	-0.9398231426	-1.5648358813	0.3000937607



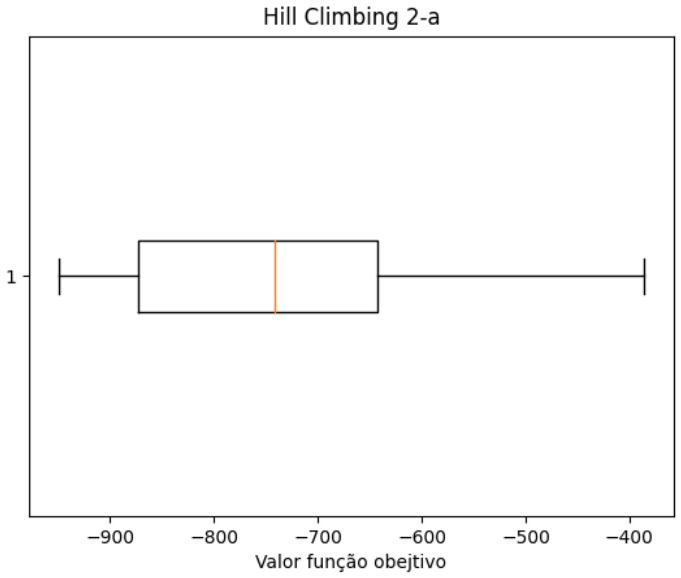
3.2) Questão 1-b

Algoritmo	Mínimo	Máximo	Média	Desvio-padrão
HC	-1.9132221701	-1.9123058876	-1.9130979372	0.0001984715



3.3) Questão 2-a

Algoritmo	Mínimo	Máximo	Média	Desvio-padrão
HC	-949.7486091442	-385.6179340881	-743.3713858846	148.5507412805



3.4) Questão 2-b

Algoritmo	Mínimo	Máximo	Média	Desvio-padrão
HC	-959.6375033164	-958.4513587668	-959.0885368479	0.2528285915