

UNIVERSIDADE FEDERAL DE VIÇOSA - CAMPUS FLORESTAL / UFV
CCF 480 - META-HEURÍSTICAS
PROF. MARCUS HENRIQUE SOARES MENDES

VICTOR HUGO REZENDE DOS SANTOS - 3510

TRABALHO PRÁTICO 1 - HILL-CLIMBING E ITERATED LOCAL SEARCH

FLORESTAL
2021

SUMÁRIO

1	INTRODUÇÃO	2
1.1	LINGUAGEM UTILIZADA	2
1.2	SOFTWARE UTILIZADO	2
1.3	REQUISITOS PARA EXECUÇÃO	2
2	DESENVOLVIMENTO	3
2.1	HILL-CLIMBING	3
2.1.1	Inicialização	3
2.1.2	Modificação	3
2.1.3	Avaliação e Seleção	3
2.1.4	Condição de Parada	3
2.2	ITERATED LOCAL SEARCH	4
2.2.1	Inicialização	4
2.2.2	Busca Local	4
2.2.3	Perturbação	4
2.2.4	Critério de Aceitação	4
2.2.5	Estrutura de Memória	5
2.2.6	Condição de Parada	5
2.3	TESTES DE EXECUÇÃO E COMPARAÇÃO DOS ALGORITMOS	5
3	CONCLUSÃO	7
4	REFERÊNCIAS BIBLIOGRÁFICAS	8

1 INTRODUÇÃO

Após, em ambiente virtual de aula, conhecer as meta-heurísticas **Hill-Climbing e Iterated Local Search (ILS)**, o professor Marcus Henrique Soares Mendes propôs a implementação do algoritmo de ambas para minimizar as seguintes funções:

$$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$$

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$

e praticar os conceitos vistos.

Os intervalos para as variáveis de decisão x e y , foram definidos nas alternativas a, b, c e d, sendo:

Função 1:

a) Com $-1.5 \leq x \leq 4$ e $-3 \leq y \leq 4$

b) Com $-1 \leq x \leq 0$ e $-2 \leq y \leq -1$

Função 2:

c) Com $-512 \leq x, y \leq 512$

d) Com $511 \leq x \leq 512$ e $404 \leq y \leq 405$

1.1 LINGUAGEM UTILIZADA

A implementação foi feita utilizando a linguagem de programação **Python**.

1.2 SOFTWARE UTILIZADO

Para a codificação do trabalho prático, foi utilizado o editor de código-fonte **Visual Studio Code**.

1.3 REQUISITOS PARA EXECUÇÃO

A implementação proposta foi implementada e executada no Sistema Operacional Windows, para tal serão necessários: Python Interpreter (3.9.7) e extensão Code Runner do Visual Studio Code para executar os arquivos .py.

2 DESENVOLVIMENTO

2.1 HILL-CLIMBING

2.1.1 Inicialização

Para a meta-heurística Hill-Climbing, a inicialização da solução se dá de forma **aleatória**, assim como descrito nas notas de aula da disciplina. Ou seja, escolhe-se um valor aleatório entre os limites definidos para cada variável.

2.1.2 Modificação

Para a etapa de modificação, que é o primeiro passo do processo iterativo, cada alternativa (a, b, c ou d) recebe valores fixos para a perturbação **r** ser dada na solução corrente, são eles: **1.5; 0.1; 500; 0.7**; respectivamente. Para as alternativas a e b, os valores fixados para r, são considerados, relativamente, pequenos, considerando os intervalos das variáveis de decisão. Esta decisão se deve ao fato de estar buscando favorecer a **exploração**, já que valores pequenos para r tendem a fazer o algoritmo "mover-se lentamente nas colinas", ficando preso em ótimos locais e tornando a busca mais **intensa**. Para as alternativas c e d, os valores são, relativamente, grandes, pois, como se sabe, a região da questão 2 é uma região menos uniforme, e um valor grande para r tende a favorecer a **exploração**, ou seja, a **diversidade** de soluções.

Após definidos os valores para r, a **solução atual** é perturbada conforme as notas de aula, ou seja, define-se um valor aleatório entre **-r e r** e assim tal valor é acrescentado à variável x ou y, vale lembrar que a probabilidade **p** de se adicionar uma perturbação à variável x ou y foi definida como **1** nesta implementação. Além disso, tal perturbação só é adicionada a uma determinada variável se os limites para a variável forem respeitados.

2.1.3 Avaliação e Seleção

Para a avaliação e seleção, verifica-se, ainda no processo iterativo, qual, dentre a **solução atual** ou a **perturbada**, apresenta melhor valor para a função objetivo, ou seja, menor valor, por se tratar de um problema de minimização. Se a perturbada apresentar menor valor, ela é atribuída como a solução atual, que será perturbada na próxima iteração.

Após o término do processo iterativo, a solução atual é tida como a melhor.

2.1.4 Condição de Parada

A condição de parada definida para o Hill-Climbing é 100 iterações consecutivas sem melhora da função objetivo, ou seja, caso a solução perturbada não seja melhor que a atual por 100 vezes consecutivas, o processo iterativo termina e o algoritmo retorna a melhor solução dentre as encontradas. Vale lembrar que, caso a perturbada não seja melhor que a atual por n

vezes, sendo $n < 100$ e na próxima iteração ela seja, a variável responsável por essa contagem é zerada. A ideia é que seja 100 iterações consecutivas sem melhora da função objetivo.

2.2 ITERATED LOCAL SEARCH

2.2.1 Inicialização

Para a meta-heurística Iterated Local Search, a inicialização se dá da mesma maneira que na meta-heurística Hill-Climbing. Tal processo foi descrito na subseção 2.1.1 e será omitido aqui, por ser exatamente o mesmo.

2.2.2 Busca Local

A procedimento usado para efetuar a busca local do ILS foi a meta-heurística Hill-Climbing, descrita na sessão 2.1. O que difere, neste caso, é que a perturbação do Hill-Climbing foi fixada em $r = 0.1$, pois é uma perturbação que, teoricamente, funciona para os limites de todas alternativas e que **favorece a exploração** na busca local, pois é um valor, relativamente pequeno.

2.2.3 Perturbação

A perturbação do ILS foi definida conforme cada problema, já que cada um possui um limite diferente para cada uma das variáveis. Sabe-se que a perturbação deve ser suficientemente grande para evitar um reinício aleatório e suficientemente pequena para encontrar soluções diversas.

A perturbação aqui é baseada nos intervalos de cada alternativa, representando o somatório de $1/50$ avos do intervalo a cada iteração. Para facilitar o entendimento, será explicado como tal perturbação é dada para a alternativa c , que possui os limites: $-512 \leq x, y \leq 512$. Neste intervalo temos 1024 valores inteiros possíveis, então $1024/50 = 20.48$. Na **primeira iteração** do ILS, a perturbação será de **20.48**, na segunda, será **$20.48 + 20.48 = 40.96$** , e assim sucessivamente até o fim do processo iterativo.

Esta forma de perturbação é dinâmica e alterada a cada iteração, permitindo o algoritmo visitar diferentes regiões do espaço de possíveis soluções. Seguindo essa linha de raciocínio, as perturbações iniciais para cada alternativa, respectivamente, foi de: **0.11; 0.04; 20.48; 0.02**.

Definido o valor da perturbação r , a maneira como se dá é exatamente igual à perturbação do Hill-Climbing, descrita na subseção 2.1.2, ou seja, pega-se um valor entre $-r$ e r e acrescenta-se a variável x e y , respeitando os limites definidos.

2.2.4 Critério de Aceitação

O critério de aceitação é **aceitar sempre o ótimo local mais recente**, favorecendo assim, a diversificação (exploração), mas ainda assim todas as soluções obtidas são armazenadas em uma estrutura de memória, para ao final, avaliar a estrutura e escolher aquela com melhor valor.

2.2.5 Estrutura de Memória

A estrutura de memória utilizada é uma lista, que recebe cada solução (valor para x e y) a cada iteração da meta-heurística, ou seja, o s' e o s^* da literatura.

Após o procedimento iterativo, o algoritmo chama um método *define_melhor_solucao* que analisa a estrutura de memória, de modo a calcular o valor da função objetivo para cada posição da lista. Depois, com todos os valores calculados, para todas as soluções obtidas durante o processo iterativo, o método verifica qual é a melhor solução (no caso, a menor, por ser um problema de minimização) e assim retorna para o algoritmo principal, que retorna tal solução como sendo a melhor solução obtida.

2.2.6 Condição de Parada

A condição de parada definida para o Iterated Local Search é a de 50 iterações. Conforme alguns testes, verificou-se que o valor 50 era um bom valor para os resultados.

2.3 TESTES DE EXECUÇÃO E COMPARAÇÃO DOS ALGORITMOS

Para as configurações descritas anteriormente, foram feitas 30 execuções de cada algoritmo para cada alternativa e os resultados foram dados conforme as seguintes tabelas:

Algoritmo	Mínimo	Máximo	Média	Desvio-Padrão
HC	-2,128200	-2,00979	-2,08531	0,037275
ILS	-2,136676	-2,13632	-2,13656	0,000088

Figura 1 – Problema com função objetivo 1 com intervalo a) para as variáveis de decisão

Algoritmo	Mínimo	Máximo	Média	Desvio-Padrão
HC	-1,551802	-1,53703	-1,54615	0,003515
ILS	-1,552153	-1,55112	-1,55164	0,000327

Figura 2 – Problema com função objetivo 1 com intervalo b) para as variáveis de decisão

Algoritmo	Mínimo	Máximo	Média	Desvio-Padrão
HC	-345,645	-312,355	-334,7308	8,849361
ILS	-346,0228	-345,155	-345,889	0,225148

Figura 3 – Problema com função objetivo 2 com intervalo c) para as variáveis de decisão

Algoritmo	Mínimo	Máximo	Média	Desvio-Padrão
HC	-271,4527	-271,437	-271,446	0,004598
ILS	-271,4534	-271,453	-271,453	0,000195

Figura 4 – Problema com função objetivo 2 com intervalo d) para as variáveis de decisão

Para uma melhor observação dos dados obtidos, os valores das 30 execuções foram, também, para cada alternativa, plotados nos seguintes bloxplots:

Obs: 0 representa a execução da meta-heurística Hill-Climbing e 1 representa a execução da meta-heurística Iterated Local Search.

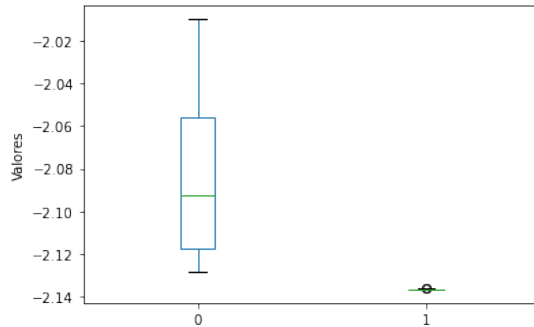


Figura 5 – Boxplot para execução das meta-heurísticas para a letra A

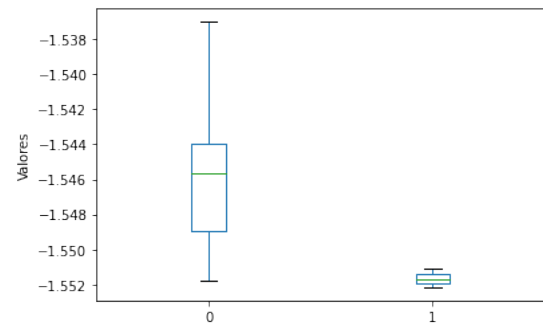


Figura 6 – Boxplot para execução das meta-heurísticas para a letra B

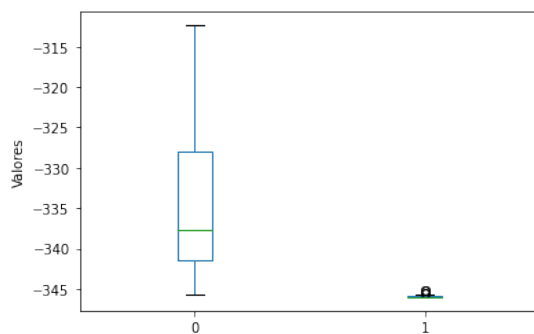


Figura 7 – Boxplot para execução das meta-heurísticas para a letra C

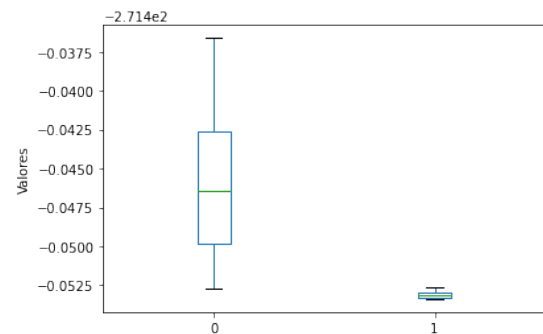


Figura 8 – Boxplot para execução das meta-heurísticas para a letra D

A discussão dos resultados obtidos será apresentada na conclusão deste relatório.

Vale lembrar que, para a execução, seguir as orientações dos comentários colocados no código-fonte, como comentar linha de chamada do ils, caso queira executar o hill-climbing, por exemplo. Como o foco aqui não é na forma da execução e sim nos resultados, o código precisa destes e outros ajustes. Todos eles estão bem descritos em comentários ao longo código-fonte.

3 CONCLUSÃO

A partir dos resultados obtidos e também com os conhecimentos adquiridos ao longo da disciplina, foi possível ver, na prática, a diferença entre as meta-heurísticas Hill-Climbing e Iterated Local Search. Como se sabe, a primeira é útil para encontrarmos o **ótimo local** de uma determinada região, enquanto a segunda consegue encontrar o **ótimo global** com maior eficiência, é claro, dependendo de suas configurações pré-estabelecidas.

As visualizações apresentadas pelos boxplots e também pelos números das tabelas nos mostram como o Hill-Climbing apresentou uma **variação grande** em seus valores ótimos, pois o algoritmo fica **preso em ótimos locais**, enquanto o ILS é mais **preciso** e tende a permanecer **suficientemente próximo ao ótimo global**. Apesar dos valores inteiros serem bem próximos, **os valores decimais é que apresentam a grande diferença** entre ambas meta-heurísticas, o que, na prática, pode ser extremamente determinante, dependendo do problema. Sabemos que há aplicações, nas quais um erro de cálculo pode resultar em uma tragédia. Então, sim, os resultados são consideravelmente distintos entre si, principalmente se considerarmos os **desvios-padrões**.

Apesar do Hill-Climbing apresentar resultados menos satisfatórios, como foi dito anteriormente, ele é indicado para a busca local e se incorporado à um algoritmo que busque o ótimo global por meio de inúmeras buscas locais (como o ILS, por exemplo), ele se mostra bem eficiente, assim como se mostrou durante a elaboração deste trabalho prático. Um ponto interessante do Hill-Climbing é a sua facilidade de ser utilizado em combinação com outras meta-heurísticas.

Contudo, foi uma experiência bastante produtiva e positiva para o desenvolvimento na disciplina, uma vez que nos mostra, por meio da prática, o uso das meta-heurísticas Hill-Climbing e Iterated Local Search e como a busca pelo ótimo global é mais interessante, pois apresenta resultados mais precisos.

4 REFERÊNCIAS BIBLIOGRÁFICAS

S. Luke. Essentials of Metaheuristics. Lulu, [2013].

Michel Gendreau, Jean-Yves Potvin. Handbook of Metaheuristics. Springer, [2010].

MENDES, Marcus Henrique Soares. Meta-heurísticas. Notas de Aula, [2021].