Relatório do trabalho de Inteligência artificial

Resolução de diferentes problemas complexos mediante busca/otimização meta-heurística

Carlos Alberto Morais Moura Filho
0223164-6
Universidade de Fortaleza
Ciências da computação
Fortaleza – CE, Brasil

RESUMO – A abordagem de otimização com os algoritmos de busca, dão possibilidade de resolver problemas complexos mais rapidamente, sem um alto custo de processamento computacional.

I. INTRODUÇÃO

O presente relatório tem por objetivo descrever o trabalho desenvolvido para solucionar os diferentes problemas propostos, utilizando as estratégias de busca/otimização meta-heurísticas estudadas na primeira etapa da cadeira de Inteligência artificial, no curso de Ciências da computação da Universidade de Fortaleza, em 2024.1.

II. METODOLOGIA

ALGORITMOS

Para a resolução das oito funções propostas no problema 1, foram utilizados três dos algoritmos estudados em sala:

- HC Hill Climbing – Subida da encosta
- LRS Local Random Search – Busca local aleatória
- GRS Global Random Search – Busca aleatória global

Para a solução do problema 2, primeiro foi necessário modelar a função proposta: $f(\chi) = 28 - h(\chi)$, depois decidir se o problema era de maximização ou minimização desta função, para depois aplicarmos um outro algoritmo estudado previamente; o algoritmo adotado para a solução desse problema foi o Annealing Heat: Têmpera simulada.

III. RESULTADOS

PROBLEMA 1

Foi solicitado a solução de oito funções diferentes, utilizando os algoritmos descritos na metodologia acima, onde, para cada uma das funções, fosse projetada uma sequência de 100 rodadas, de modo que, se armazenasse a solução obtida pelo algoritmo aplicado em cada rodada.

Ao final da execução é apresentado o gráfico com o ponto inicial de partida e o melhor ponto ótimo obtido nas rodadas; juntamente, com os resultados de χ_1 e χ_2, da função calculada aplicando esses pontos (χ_1, χ_2) e o valor do hiperparâmetro, no console do programa; assim como, também, uma tabela (DataFrame) onde podemos observar todos os ótimos obtidos nas rodadas.

Abaixo, um exemplo do DataFrame de resposta que o projeto retorna depois da execução:

	pt_otimo[x]	pt_otimo[y]	f(otimo)	hiperparametro
0	-100.000000	-100.000000	2.000000e+04	0.0
1	-66.399009	-66.444973	8.823763e+03	0.1
2	-34.134414	-34.009034	2.321773e+03	0.1
3	-0.007300	-0.022157	5.442396e-04	0.1
4	-0.007300	-0.022157	5.442396e-04	0.1
...	...	...	...	...
97	0.000372	-0.000203	1.795579e-07	0.1
98	0.000372	-0.000203	1.795579e-07	0.1
99	0.000372	-0.000203	1.795579e-07	0.1
100	0.000372	-0.000203	1.795579e-07	0.1
101	0.000372	-0.000203	1.795579e-07	0.1

[102 rows x 4 columns]

De posse deste DataFrame de resultado, ele pode ser passado para duas funções no programa, uma para exibir os gráficos interativos na tela, utilizando a biblioteca do Matplotlib ou a outra que salva os arquivos de imagens em arquivos '.png' e '.gif' e os resultados do DataFrame em arquivos '.csv'.

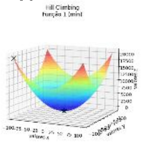
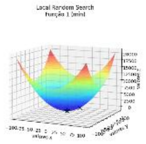
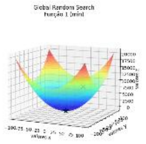
Segue abaixo todos os gráficos e resultados encontrados como solução para as funções proposta no problema:

- Função 1

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = \chi_1^2 + \chi_2^2$,

$\chi_1, \chi_2 \in [-100, 100]$

Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.9630$	$\sigma = 0.0000$
$\chi = (0.0005, 0.0001)$	$\chi = (-0.0003, 0.0000)$	$\chi = (-0.2356, -0.5280)$
$f(\chi) = 0.0000$	$f(\chi) = 0.0000$	$f(\chi) = 0.3343$

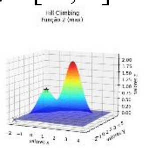
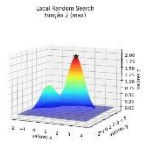
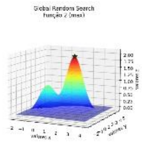
- Função 2

  Encontre o valor <u>máximo</u> da função

$f(\chi_1, \chi_2) = e^{-(x_1^2+x_2^2)} + 2 \cdot e^{-(x1-1.7)**2+(x2-1.7)**2}$

$\chi_1 \in [-2, 4]$

$\chi_2 \in [-2, 5]$

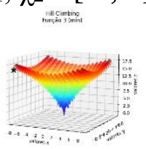
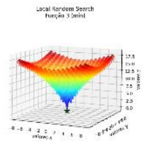
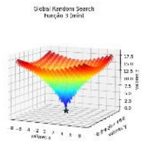
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.2729$	$\sigma = 0.0000$
$\chi = (0.0108, 0.0094)$	$\chi = (1.6974, 1.6973)$	$\chi = (1.6993, 1.7005)$
$f(\chi) = 1.0064$	$f(\chi) = 2.0031$	$f(\chi) = 2.0031$

- Função 3

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = -20 * e^{-0.2\cdot\sqrt{0.5\cdot(x_1^2+x_2^2)}} - e^{0.5\cdot(\cos(2\pi x_1)+\cos(2\pi x_2))} + 20 + e^1$

$\chi_1, \chi_2 \in [-8, 8]$

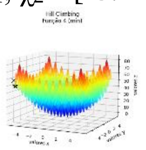
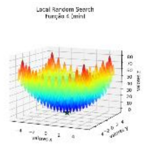
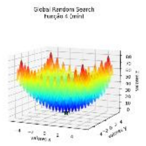
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.2847$	$\sigma = 0.0000$
$\chi = (-7.9955, -7.9966)$	$\chi = (0.0002, -0.0001)$	$\chi = (0.0279, -0.0283)$
$f(\chi) = 15.9597$	$f(\chi) = 0.0005$	$f(\chi) = 0.1541$

- Função 4

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = (x_1^2 - 10\cdot\cos(2\pi x_1) + 10) + (x_2^2 - 10\cdot\cos(2\pi x_2) + 10)$

$\chi_1, \chi_2 \in [-5.12, 5.12]$

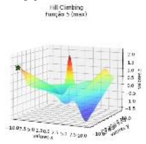
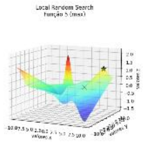
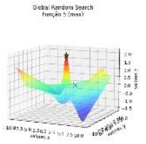
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.2912$	$\sigma = 0.0000$
$\chi = (-4.9732, -4.9780)$	$\chi = (0.0000, -0.0000)$	$\chi = (0.0139, -0.0229)$
$f(\chi) = 49.7500$	$f(\chi) = 0.0000$	$f(\chi) = 0.1425$

- Função 5

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = \frac{x_1 \cdot cos(x_1)}{20} + 2\cdot e^{-(x_1)^2-(x_2-1)^2} + 0.01\cdot x_1 \cdot x_2$

$\chi_1, \chi_2 \in [-10, 10]$

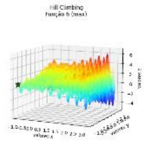
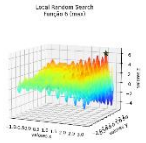
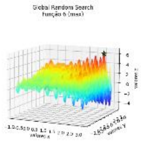
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.9896$	$\sigma = 0.0000$
$\chi = (-9.9010, -10.0000)$	$\chi = (6.7309, 10.0000)$	$\chi = (-0.0029, 0.9924)$
$f(\chi) = 1.4301$	$f(\chi) = 0.9765$	$f(\chi) = 1.9997$

- Função 6

  Encontre o valor <u>máximo</u> da função

$f(\chi_1, \chi_2) = \chi_1 \cdot \text{sen}(4\pi\chi_1) - \chi_2 \cdot \text{sen}(4\pi\chi_2 + \pi) + 1$

$\chi_1, \chi_2 \in [-1, 3]$

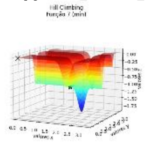
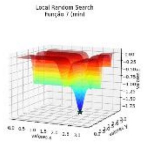
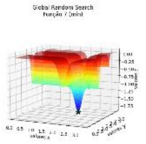
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.2076$	$\sigma = 0.0000$
$\chi = (-1.0000, -1.0000)$	$\chi = (2.6273, 2.6275)$	$\chi = (2.6246, 2.6234)$
$f(\chi) = 1.0000$	$f(\chi) = 6.2524$	$f(\chi) = 6.2474$

- Função 7

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = -\text{sen}(\chi_1) \cdot \text{sen}(\chi_1^2/\pi)^{2\cdot 10} - \text{sen}(\chi_2) \cdot \text{sen}(2\, \chi_2^2/\pi)^{2\cdot 10}$

$\chi_1, \chi_2 \in [0, \pi]$

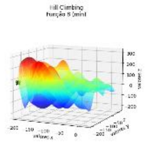
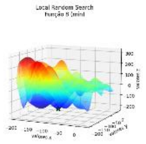
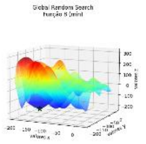
Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.7655$	$\sigma = 0.0000$
$\chi = (1.6703, 1.5689)$	$\chi = (2.2025, 1.5674)$	$\chi = (2.2081, 1.5668)$
$f(\chi) = -1.0061$	$f(\chi) = -1.8008$	$f(\chi) = -1.8002$

- Função 8

  Encontre o valor <u>mínimo</u> da função

$f(\chi_1, \chi_2) = -(x_2+47)\cdot\sin\left(\sqrt{|x_1/2+(x_2+47)|}\right) - x_1\cdot\sin\left(\sqrt{|x_1-(x_2+47)|}\right)$

$\chi_1, \chi_2 \in [-200, 20]$

Hill Climbing	Local Random Search	Global Random Search
$\varepsilon = 0.1000$	$\sigma = 0.2545$	$\sigma = 0.0000$
$\chi = (-199.8888, -199.9940)$	$\chi = (-101.5779, -126.0393)$	$\chi = (-171.8284, -96.7674)$
$f(\chi) = 77.1813$	$f(\chi) = -174.3570$	$f(\chi) = -211.1297$

Todas estas soluções aqui apresentadas foram encontradas aplicando os mesmos algoritmos de buscas nas funções propostas do problema.

PROBLEMA 2

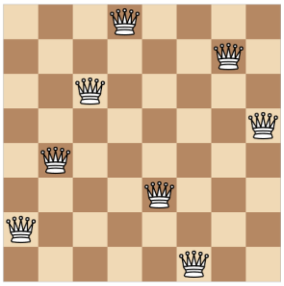
Nesse problema foi modelada a seguinte função: $f(\chi) = 28 - h(\chi)$, onde $h(\chi)$ é o número de pares em conflito, que recebe um vetor de entrada com oito posições retornando um número real.

Foi aplicado o algoritmo de Têmpera simulada, com a temperatura inicial em 100 e o decaimento de 0.93, com o propósito de maximizar a função e encontrar as 92 soluções possíveis, que estão apresentadas no DataFrame abaixo:

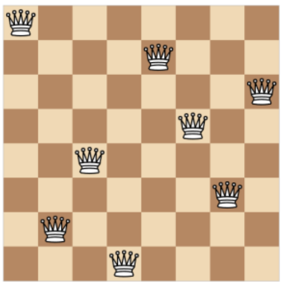
```
                  solucao
0        [0, 0, 0, 0, 0, 0, 0, 0]
1        [0, 0, 0, 0, 0, 0, 0, 0]
2        [0, 0, 0, 0, 0, 0, 0, 0]
3        [0, 0, 0, 0, 0, 0, 0, 0]
4        [0, 0, 0, 0, 0, 0, 0, 0]
..                         ...
87       [0, 0, 0, 0, 0, 0, 0, 0]
88       [0, 0, 0, 0, 0, 0, 0, 0]
89       [0, 0, 0, 0, 0, 0, 0, 0]
90       [0, 0, 0, 0, 0, 0, 0, 0]
91       [0, 0, 0, 0, 0, 0, 0, 0]
[92 rows x 1 column]
```

Duas soluções possíveis encontradas, representadas graficamente num editor de tabuleiros de xadrez online:

- Soluções



[2, 4, 6, 8, 3, 1, 7, 5]



[8, 2, 4, 1, 7, 5, 3, 6]

IV. Conclusões

Para o problema 1 pôde-se concluir que o melhor algoritmo de busca foi o GRS Global Random Search – Busca aleatória global, que encontrou o ótimo de todas as funções em todas as rodadas. O segundo algoritmo com maior acerto foi o LRS Local Random Search – Busca local aleatória, apesar de que, em algumas funções, ele ficou preso em um ótimo local. O algoritmo com menos acertos, o pior de todos por assim dizer, foi o HC Hill Climbing – Subida da encosta, que só conseguiu encontrar o ótimo de uma função (Função 1), por ela ser a única função unimodal do problema.

Já no problema 2, é possível encontrar 92 soluções; sendo 12 distintas e as demais por meio de rotação e reflexão no tabuleiro. O custo computacional para conseguir encontrar todas as 92 soluções foi de ____.

Referências

Russell, Stuart J.; Norvig, Peter; **Inteligência artificial: Uma abordagem moderna.** Editora Campus, 2013

Souza Barbosa, Paulo Cirillo; **Busca/Otimização Meta-Heurística** – UNIFOR, 2024.1

Duplessis, Thibault; **Xadrez online: Editor de tabuleiro.** Disponível em: <http://pt.chesster.ru/editor>. Acessado em: 19 mar 2024.