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.

1. **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.

1. **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(χ) = 28 - h(χ), 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.

1. **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 χ1 e χ2, da função calculada aplicando esses pontos (χ1, χ2) 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 mínimo da função

ʄ(χ1, χ2) = χ12 + χ22,

χ1, χ2 ϵ [-100, 100]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.9630 σ = 0.0000

χ = (0.0005, 0.0001) χ = (-0.0003, 0.0000) χ = (-0.2356, -0.5280)

ʄ(χ) = 0.0000 ʄ(χ) = 0.0000 ʄ(χ) = 0.3343

* Função 2

Encontre o valor máximo da função

ʄ(χ1, χ2) =

χ1 ϵ [-2, 4]

χ2 ϵ [-2, 5]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.2729 σ = 0.0000

χ = (0.0108, 0.0094) χ = (1.6974, 1.6973) χ = (1.6993, 1.7005)

ʄ(χ) = 1.0064 ʄ(χ) = 2.0031 ʄ(χ) = 2.0031

* Função 3

Encontre o valor mínimo da função

ʄ(χ1, χ2) =

χ1, χ2 ϵ [-8, 8]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.2847 σ = 0.0000

χ = (-7.9955, -7.9966) χ = (0.0002, -0.0001) χ = (0.0279, -0.0283)

ʄ(χ) = 15.9597 ʄ(χ) = 0.0005 ʄ(χ) = 0.1541

* Função 4

Encontre o valor mínimo da função

ʄ(χ1, χ2) =

χ1, χ2 ϵ [-5.12, 5.12]

Gráfico

Descrição gerada automaticamenteGráfico

Descrição gerada automaticamenteGráfico

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.2912 σ = 0.0000

χ = (-4.9732, -4.9780) χ = (0.0000, -0.0000) χ = (0.0139, -0.0229)

ʄ(χ) = 49.7500 ʄ(χ) = 0.0000 ʄ(χ) = 0.1425

* Função 5

Encontre o valor mínimo da função

ʄ(χ1, χ2) =

χ1, χ2 ϵ [-10, 10]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.9896 σ = 0.0000

χ = (-9.9010, -10.0000) χ = (6.7309, 10.0000) χ = (-0.0029, 0.9924)

ʄ(χ) = 1.4301 ʄ(χ) = 0.9765 ʄ(χ) = 1.9997

* Função 6

Encontre o valor máximo da função

ʄ(χ1, χ2) = χ1 ‧ sen(4πχ1) – χ2 ‧ sen(4πχ2 + π) +1

χ1, χ2 ϵ [-1, 3]

Gráfico

Descrição gerada automaticamenteGráfico

Descrição gerada automaticamenteGráfico, Diagrama

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.2076 σ = 0.0000

χ = (-1.0000, -1.0000) χ = (2.6273, 2.6275) χ = (2.6246, 2.6234)

ʄ(χ) = 1.0000 ʄ(χ) = 6.2524 ʄ(χ) = 6.2474

* Função 7

Encontre o valor mínimo da função

ʄ(χ1, χ2) = –sen(χ1) ‧ sen(χ12/π)2‧10 – sen(χ2) ‧ sen(2 χ22/π)2‧10

χ1, χ2 ϵ [0, π]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.7655 σ = 0.0000

χ = (1.6703, 1.5689) χ = (2.2025, 1.5674) χ = (2.2081, 1.5668)

ʄ(χ) = -1.0061 ʄ(χ) = -1.8008 ʄ(χ) = -1.8002

* Função 8

Encontre o valor mínimo da função

ʄ(χ1, χ2) =

χ1, χ2 ϵ [-200, 20]

Gráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamenteGráfico, Gráfico de superfície

Descrição gerada automaticamente

**Hill Climbing Local Random Search Global Random Search**

ε = 0.1000 σ = 0.2545 σ = 0.0000

χ = (-199.8888, -199.9940) χ = (-101.5779, -126.0393) χ = (-171.8284, -96.7674)

ʄ(χ) = 77.1813 ʄ(χ) = -174.3570 ʄ(χ) = -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: ʄ(χ) = 28 – h(χ), onde h(χ) 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

Gráfico, Gráfico de mapa de árvore

Descrição gerada automaticamente Gráfico, Gráfico de mapa de árvore

Descrição gerada automaticamente

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

1. **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.