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]

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

ε = 0.1000 σ = 0.7368 σ = 0.0000

χ = (-0.0000, 0.0014) χ = (-0.0002, -0.0004) χ = (-0.1063, 0.0507)

ʄ(χ) = 0.0000 ʄ(χ) = 0.0000 ʄ(χ) = 0.0139

* Função 2

Encontre o valor máximo da função

ʄ(χ1, χ2) = ,

χ1 ϵ [-2, 4]

χ2 ϵ [-2, 5]

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

ε = 0.1000 σ = 0.1428 σ = 0.0000

χ = (0.0112, 0.0105) χ = (1.6969, 1.6969) χ = (1.6982, 1.7092)

ʄ(χ) = 1.0064 ʄ(χ) = 2.0031 ʄ(χ) = 2.0028

* Função 3

Encontre o valor mínimo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [-8, 8]

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

ε = 0.1000 σ = 0.8471 σ = 0.0000

χ = (-7.9941, -7.9911) χ = (-0.0000, 0.0000) χ = (-0.0259, 0.0154)

ʄ(χ) = 15.9591 ʄ(χ) = 0.0001 ʄ(χ) = 0.1091

* Função 4

Encontre o valor mínimo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [-5.12, 5.12]

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

ε = 0.1000 σ = 0.4529 σ = 0.0000

χ = (-4.9744, -4.9745) χ = (0.0001, -0.0000) χ = (-0.0265, -0.0059)

ʄ(χ) = 49.7475 ʄ(χ) = 0.0000 ʄ(χ) = 0.1463

* Função 5

Encontre o valor mínimo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [-10, 10]

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

ε = 0.1000 σ = 0.6724 σ = 0.0000

χ = (-9.9017, -10.0000) χ = (-9.7330, -10.0000) χ = (-0.0110, 1.0323)

ʄ(χ) = 1.4300 ʄ(χ) = 1.4370 ʄ(χ) = 1.9970

* Função 6

Encontre o valor máximo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [-1, 3]

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

ε = 0.1000 σ = 0.3349 σ = 0.0000

χ = (-1.0000, -1.0000) χ = (2.6274, 2.6276) χ = (2.6340, 2.6236)

ʄ(χ) = 1.0000 ʄ(χ) = 6.2524 ʄ(χ) = 6.2403

* Função 7

Encontre o valor mínimo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [0, π]

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

ε = 0.1000 σ = 0.4696 σ = 0.0000

χ = (1.6705, 1.5685) χ = (2.2007, 1.5743) χ = (2.2021, 1.5641)

ʄ(χ) = -1.0060 ʄ(χ) = -1.8007 ʄ(χ) = -1.7995

* Função 8

Encontre o valor mínimo da função

ʄ(χ1, χ2) = ,

χ1, χ2 ϵ [-200, 20]

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

ε = 0.1000 σ = 0.1376 σ = 0.0000

χ = (-199.9003, -199.9998) χ = (-102.0016, -126.0256) χ = (-170.4667, -95.7923)

ʄ(χ) = 77.2035 ʄ(χ) = -174.2273 ʄ(χ) = -211.1383

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[8]) = 28 – h(χ), onde h(x) 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 \_\_\_\_\_ e o decaimento aplicado foi de \_\_\_\_\_, com o propósito de maximizar a função e encontrar as 92 soluções possíveis, que são apresentadas no DataFrame abaixo:

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 execuções. 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 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, a conclusão foi que \_\_\_\_\_\_\_\_\_.

**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

teste

teste