
Trabalho IA (3 pontos)

1. Crie um algoritmo genético para o problema das 8 rainhas visto em sala (2 pontos).
 - (a) Cada indivíduo deve ser representado por um vetor de inteiros de 8 posições. Cada posição do vetor representa a coluna da rainha, enquanto o valor do vetor representa a linha que a rainha está no tabuleiro. Por exemplo, o indivíduo [2, 3, 5, 7, 4, 5, 6, 8] é um tabuleiro 8×8 . O primeiro elemento do vetor (2) indica que existe uma rainha na coluna 1 e linha 2 do tabuleiro. Enquanto, o valor 3, indica que existe uma rainha na coluna 2 e linha 3 do tabuleiro.
 - (b) É importante que cada indivíduo, além da representação vetorial, tenha o número da iteração em que ele foi gerado (campo geração) e outro campo chamado id. Todos os indivíduos devem ter id's diferentes.
 - (c) A função fitness, que medirá a qualidade de um indivíduo, deverá retornar a quantidade de pares de rainhas que não estão se atacando.
 - (d) A probabilidade de um indivíduo ser selecionado para gerar filhos deve ser proporcional ao seu valor da função **fitness** (verificar slides).
 - (e) A operação de **crossover** deve gerar 2 filhos com ponto de corte aleatório. Vale salientar que todos os filhos possuem uma certa probabilidade de mutação.
 - (f) O algoritmo deve ter 3 parâmetros de controle: o tamanho da população (tam_{pop}), o número de iterações máximo que ele irá executar (it_{max}) e a taxa de probabilidade de mutação (tx_{mut}).
 - (g) Nosso algoritmo será uma versão elitista em que a população da próxima iteração é composta pelos n melhores indivíduos da população atual. $n = tam_{pop}$.

Algorithm 1 Genetic Algorithm

```
populacao ← geraPopulacaoInicial ▷ Geração 0
geracao ← 1
while geracao ≤  $it_{max}$  ∧ melhor_fitness_nao_encontrado do
    novaPopulacao = ∅
    for i = 1 to  $tam_{pop}$  do
        pais ← selecionarPai(populacao)
        filhos ← crossover(pais)
        mutar(filhos,  $tx_{mut}$ )
        novaPopulacao.adicionar(filhos)
    end for
    populacao.adicionar(novaPopulacao)
    populacao = melhoresIndividuos(populacao,  $tam_{pop}$ )
    geracao ++
end while
imprima(informacoes_melhor_individuo)
```

- (h) Rode o algoritmo 10 vezes com os parâmetros $it_{max} = 100$, $tam_{pop} = 150$, e $tx_{mut} = 0.2$. Quantas vezes, das 10 execuções, ele encontrou um indivíduo que consegue resolver o problema das 8 rainhas?
 - (i) Faça o mesmo processo, mantendo os parâmetros da questão anterior e alterando o tamanho da população para $tam_{pop} = 100$ e $tam_{pop} = 200$. Aplicando o algoritmo 10 vezes em cada uma dessas configurações, quantas vezes ele encontra a solução?
2. Faça o algoritmo da subida na encosta para resolver o problema das 8 rainhas. (1 ponto)
- (a) A função que irá avaliar a qualidade dos sucessores é $h(x)$ = quantidade de pares de rainhas que se atacam no tabuleiro.
 - (b) O objetivo é minimizar a função de avaliação.
 - (c) Execute o algoritmo diversas vezes utilizando estados iniciais aleatórios. Calcule a quantidade de vezes, em média, em que é preciso executar a subida na encosta para encontrar a solução do problema.