

UFMG - Departamento de Engenharia Elétrica  
Fundamentos de IA  
Trabalho 2: Algoritmos Genéticos

Cristiano Leite de Castro  
João Pedro Campos

## Instruções Gerais

### Objetivo

Familiarizar-se com a ideia e a implementação de algoritmos genéticos. Obter uma noção de como o tamanho da população, decisões de implementação e hiperparâmetros afetam o desempenho dos algoritmos. Os principais hiperparâmetros a serem estudados são a **taxa de cruzamento** e a **taxa de mutação**.

### Grupos

Este trabalho deve ser feito em grupos de três ou duas pessoas. Podem ser mantidos os grupos do trabalho anterior.

### Sua tarefa

Implemente um programa em Python para resolver a Tarefa 1 e a Tarefa 2, especificadas mais abaixo. Com escolhas de design cuidadosas, a mesma base do programa pode ser usada para ambas as tarefas, mas também podem fazer dois programas, se preferirem.

Após implementar o programa, teste as tarefas usando diferentes combinações de tamanhos de população e probabilidades de mutação e cruzamento. **Cabe a você** decidir quais valores usar! Recomenda-se buscar na literatura valores típicos para validar sua escolha. Cabe a você também decidir como fazer os experimentos para comparar as versões dos algoritmos. Resuma estes resultados em uma tabela e discuta as implicações. Lembre-se de usar uma métrica apropriada para comparar os resultados, por exemplo, o número total de chamadas da função objetivo, ou simplesmente o tempo de execução.

### O que deve ser entregue

1. Um relatório completo, incluindo: uma descrição da sua implementação, incluindo comentários sobre suas escolhas de design e como sua seleção,

mutação e cruzamento funcionam. Os resultados, resumindo as combinações de probabilidades de mutação e cruzamento e tamanhos de população, e suas reflexões sobre os resultados. Use o que julgar pertinente: tabelas e gráficos são bem-vindos.

2. O código do programa, em arquivo separado (não inserir código no relatório.)

## Avaliação

A avaliação baseia-se principalmente na qualidade do relatório entregue, na qualidade dos testes que você realiza e na análise inteligente dos resultados. Justifique corretamente as decisões de implementação e o design dos experimentos. Cuidado também com a correção gramatical, apresentação das tabelas e figuras, bem como o uso e a formatação das referências consultadas.

## Tarefa 1 - Um problema de minimização com 10 variáveis

Minimize a função:

$$f(x_1, \dots, x_{10}) = \sum_i x_i^2$$

onde  $-5 \leq x_i \leq 5 \forall i$ . Todas as variáveis são **reais, contínuas**.

## Seleção

Você pode usar o método de seleção padrão, onde a probabilidade de seleção é proporcional à aptidão. Normalmente, um indivíduo selecionado não é removido do conjunto de seleção (ou seja, da geração anterior), sendo permitido participar de seleções subsequentes, ou seja, um indivíduo com alta aptidão tem alta probabilidade de ser selecionado várias vezes. Outra estratégia de seleção comum é a chamada “seleção por torneio”. É parte da tarefa pesquisar sobre formas de seleção e implementar aquela que julgarem mais adequada, justificando a escolha.

## Cruzamento

Dois indivíduos são selecionados como pais para produzir um ou dois novos descendentes por meio de cruzamento. Um ou vários pontos de cruzamento são escolhidos aleatoriamente.

## Mutação

Para mutar um indivíduo, um (ou qualquer número de) valores é substituído por um novo valor dentro do intervalo permitido. Reparem que, neste problema, as variáveis são contínuas, e portanto operadores clássicos de mutação (como o *bit-flip*) não funcionam. Pesquisem e implementem um operador de mutação adequado a variáveis contínuas, como por exemplo a mutação gaussiana:

$$x_i \leftarrow x_i + \mathcal{N}(0, \sigma)$$

## Tarefa 2 - Um problema da mochila

Imagine-se como um ladrão cercado por preciosos tesouros. Você gostaria de levar todos eles. Infelizmente, há um limite de quanto você pode carregar. Então você é forçado a deixar alguns tesouros para trás. Agora, você deve decidir quais tesouros levar e quais deixar. Você quer, é claro, maximizar o valor dos tesouros que traz para casa, mas sua mochila é limitada e só aguenta determinada carga  $C$ .

Vamos denotar  $w_i$  o peso do item  $i$ ,  $p_i$  o valor do item  $i$  e  $x_i$  ser 1 se você levar o item e 0 se deixá-lo. Além disso,  $C$  denota a carga máxima que você pode carregar.

## Modelagem

Você deve modelar esse problema na forma de um problema de otimização, especificando a função objetivo e a restrição. Apresente seu modelo no relatório.

## Instância do problema

O arquivo **mochila.txt** contém os dados a serem utilizados. O formato do arquivo é como segue:

```
n
c

p_1 w_1
p_2 w_2
...
p_n w_n
```

## Cuidados especiais

Nesta tarefa, você deve ter cuidado especial ao criar ou modificar indivíduos por mutação ou cruzamento para garantir que a restrição seja sempre cumprida (ou penalizada na função objetivo). Descobrir uma maneira de realizar isso faz parte da tarefa. Dica: como lidávamos com restrições em otimização não linear?

## **Cruzamento**

Você pode usar pontos de cruzamento, mas sinta-se à vontade para projetar outros meios de *crossover*.

## **Mutação**

Este é um problema combinatório, e a maneira mais natural de modelá-lo é por meio de um vetor binário. Um operador clássico de mutação nesse caso é o *bit-flip*, mas você pode propor outros também.

Bom trabalho!