

Algoritmos Genéticos

Carolina Ribeiro Xavier

Março de 2025

O que são algoritmos genéticos?

Algoritmos genéticos são algoritmos de busca baseados na mecânica da seleção natural e na genética natural, inspirados na teoria da evolução apresentada por Charles Darwin.

Conceitos Básicos

- ▶ **Genes:** características que formam o indivíduo.
- ▶ **Cromossomo:** conjunto de genes, representando uma solução candidata.
- ▶ **População:** conjunto de indivíduos candidatos à solução do problema.
- ▶ **Geração:** cada iteração do AG produz uma nova população.

Questões de Projeto

Aspectos que variam conforme o problema:

- ▶ Função objetivo
- ▶ Tipo de representação
- ▶ Estratégia de seleção
- ▶ Cruzamento (taxa)
- ▶ Mutação (taxa)
- ▶ Elitismo

Função Objetivo

- ▶ Mede a aptidão de um indivíduo para resolver o problema.
- ▶ Pode ser de minimização ou maximização.
- ▶ Exemplo:

$$f(x) = -20\epsilon^{-0.2\sqrt{\frac{1}{n}\sum x_i^2}} - \epsilon\frac{1}{n}\sum \cos(2\pi x_i) + 20 + \epsilon \quad (1)$$

Representação dos Cromossomos

- ▶ Binária: necessita de mapeamento para valores reais.
- ▶ Real: representação direta dos parâmetros.

$$x = x_{min} + \frac{(x_{max} - x_{min})}{2^n - 1} * INT(bin) \quad (2)$$

Estratégia de Seleção

Torneio:

- ▶ Selecciona dois indivíduos aleatoriamente.
- ▶ O mais apto é escolhido como pai.

Estratégia de Seleção

```
float * torneio(int npop, float *fit){ //considerando minimizacao

    int vpais[npop];
    pv = 0.9;
    int i = 0, vencedor;
    while ( i < npop ){
        p1 = random(0,npop);  p2 = random(0,npop);
        while( p1 == p2 ){
            p2 = random(0,n);
        }
        r = random (0,1);
        if( fit[p2] > fit[p1]) //p1 e melhor
        {
            vencedor = p1;
            if(r>pv)  vencedor = p2;
        }else // p2 e melhor
        {
            vencedor = p2;
            if(r>pv)  vencedor = p1;
        }
        vpais[i] = vencedor;
        i++;
    }
    return vpais;
}
```

Cruzamento

Operação fundamental para intensificação da busca.

- Cruzamento de um ou mais pontos.

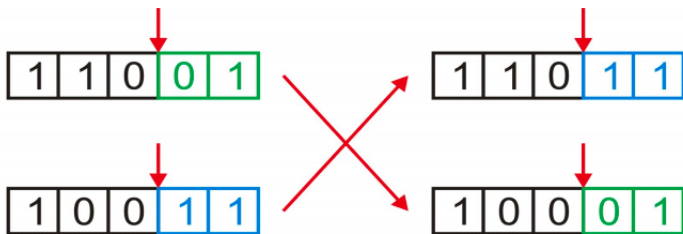


Figure: Cruzamento de um ponto

Mutação

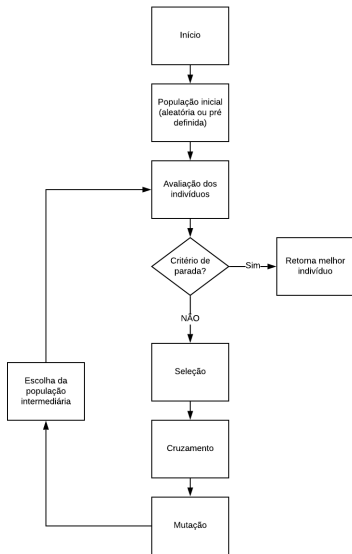
- ▶ Introduz diversidade na população.
- ▶ Permite escapar de ótimos locais.

Taxa típica: 1% a 10%.

Elitismo

- ▶ Mantém as melhores soluções entre gerações.
- ▶ Garante convergência mais estável.

Fluxograma de um AG



Implementação

Parâmetros a definir:

- ▶ Tamanho da população
- ▶ Número máximo de gerações
- ▶ Critério de seleção
- ▶ Taxas de cruzamento e mutação
- ▶ Uso do elitismo

Pseudocódigo

```
void genericAG(){
    int npop = 100; //tamanho da populacao
    int nger = 100; // numero de geracoes
    int nelite = 2;
    int pop[100][6], pop_intermediaria[100][6];

    int pais[100]; //se cada cruzamento gerar 2 filhos
    float fit[100];

    float Pc = 1; // probabilidade de cruzamento
    float Pm = 0.05; // probabilidade de mutacao

    cria_populacao_inicial (npop, pop);

    g = 0;
    while( g < nger){
        fit = avalia_populacao(npop, pop);
        pais = seleciona_pais(npop, pop, fit);
        pop_intermediaria = cruzamento(npop, pais, Pc);
        mutacao(npop, pop_intermediaria, Pm);
        elitismo(nelite);
        pop = pop_intermediaria.copy();
        g++;
    }
    imprime_populacao_final(pop);
}
```