

# IA e ML Aplicados a Finanças

Prof. Leandro Maciel

## AULA 10: Algoritmos Genéticos

- 1 Seleção de Carteiras
- 2 Algoritmos Genéticos
- 3 Bibliografia

# 1. Seleção de Carteiras

## Etapas em um processo de seleção de carteiras:

- 1. Selecionar  $n$  potenciais ativos (análise fundamentalista ou técnica);
- 2. Qual a proporção do capital ( $w_i$ ) aplicar em cada ativo?
- 3. Objetivo? Maximizar retorno ( $R_p$ )? Minimizar risco ( $\sigma_p$ )?

$$\max_{w_1, \dots, w_n} \{R_p\} \quad \text{ou} \quad \min_{w_1, \dots, w_n} \{\sigma_p\}$$

$$\text{s.a.} \quad \sum_{i=1}^n w_i = 1 \quad \text{carteira totalmente investida}$$

$$0 \leq w_i \leq 1, \forall i \quad \text{posições compradas}$$

- Formulação da função objetivo tem várias alternativas;
- Em finanças: *trade-off* **risco-retorno**;
- **Carteira de Variância Mínima (CVM)** → portfólio de menor risco;
- Objetivo é minimizar risco da carteira (desvio-padrão):

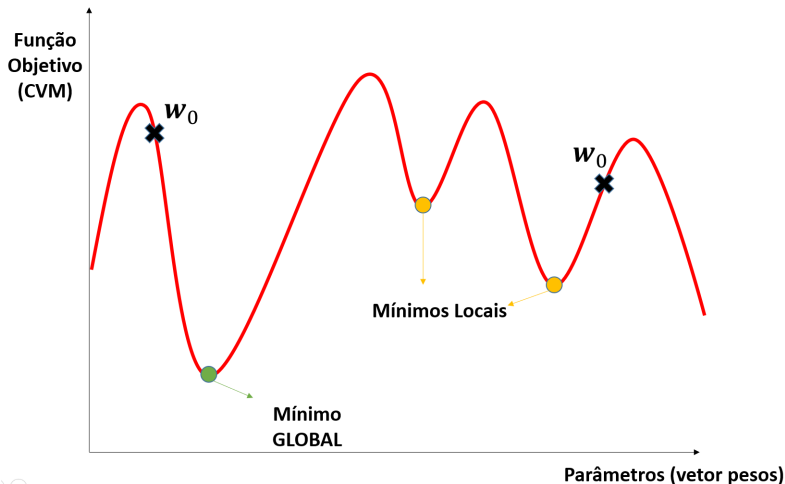
$$\begin{aligned} & \min_{w_1, \dots, w_n} \{ \sigma_p \} \\ \text{s.a. } & \sum_{i=1}^n w_i = 1 \quad \text{e} \quad 0 \leq w_i \leq 1, \forall i \end{aligned}$$

- Variável de decisão  $\rightarrow$  vetor do tipo  $\mathbf{w} = (w_1, w_2, \dots, w_n)$ ;
- Encontrar vetor de pesos que resulta na CVM;
- Há restrições nas variáveis (positividade);
- Função objetivo pode incluir as restrições:

$$\min_{w_1, \dots, w_n} \left\{ \sigma_p + \lambda \left( \sum_{i=1}^n w_i - 1 \right)^2 \right\}$$

$\lambda \rightarrow$  termo de penalização (em geral um valor grande)  $\approx 100$

- Problemas em técnicas baseadas em gradiente:



## 2. Algoritmos Genéticos



**AGs** (J. Holland, 1975 e D. Goldberg, 1989):

- Métodos de otimização inspirados na Teoria da Evolução (IA);
- Meta-heurística (processos cognitivos em decisões não aleatórias);

**Teoria da Evolução de Darwin** (Origem das Espécies, 1859):

- Indivíduos mais aptos sobrevivem no processo;
- Transmitem características para gerações futuras.

### AGs e otimização:

- População de indivíduos → possíveis soluções (cromossomos);
- Indivíduos são “melhorados” (evoluem) de forma iterativa;
- Busca da solução do problema como um **processo evolutivo**;
- **Seleção** dos mais aptos + **transmissão** de características;
- Evolução resulta em indivíduos “mais aptos” (**melhor solução**).

### ■ Etapas da otimização usando AGs:

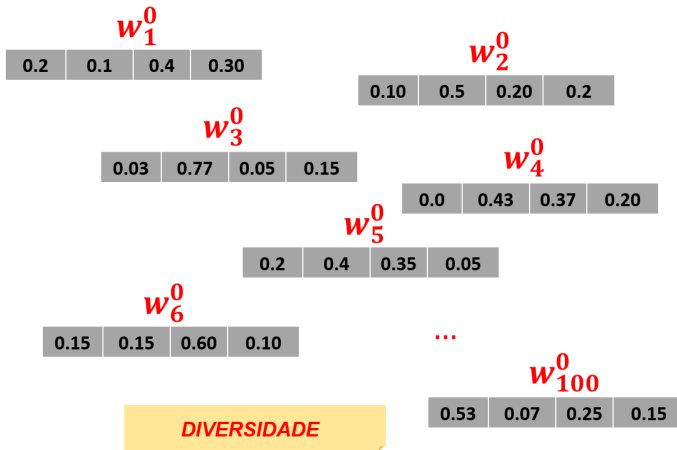
1. Gerar população inicial com indivíduos aleatórios;
2. Avaliar indivíduos de acordo com uma função de **fitness** (objetivo);
3. Selecionar mais aptos (com “melhor valor” da função objetivo);
4. Nova população com base nos mais aptos (**crossover** e **mutação**);
5. Repete o processo evolutivo (iteração - etapa 2).

- Algoritmo de alto nível e poder computacional;
- **Customizável** para uma ampla quantidade de problemas;
- AGs diferem de acordo com:
  1. Forma de representação dos indivíduos;
  2. Estratégia de seleção;
  3. Operadores de busca (crossover e mutação).

### 1. Representação dos indivíduos:

- Cromossomo como uma possível solução da função objetivo;
- Objetivo é minimizar risco da carteira (CVM);
- Vetor de pesos que resulta na carteira de menor risco;
- Ex.:  $\mathbf{w} = \{0.2, 0.1, 0.4, 0.3\}$  é uma possível solução (vetor pesos);
- Representação por meio de números reais (ou bits);
- Gera-se uma população inicial de  $N$  indivíduos.

### População Inicial (Possíveis Soluções)



### 2. Seleção:

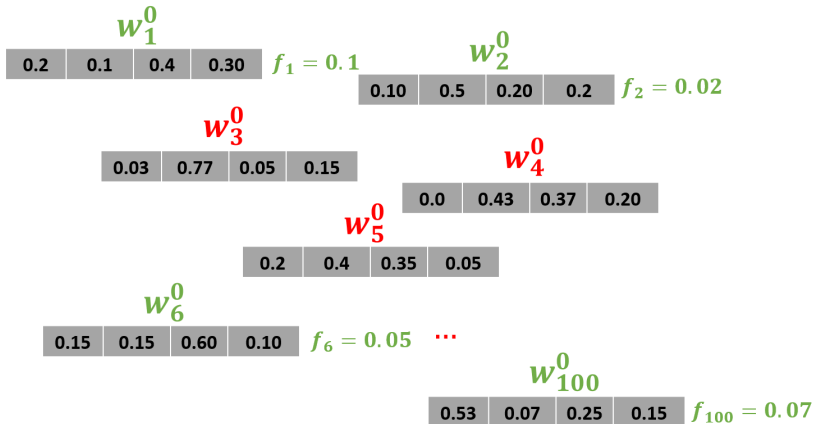
- Escolher “mais aptos” para gerar **cromossomos filhos** (novas soluções);
- Indivíduos com **melhor** valor da função objetivo (melhor ajuste);
- Vetor de pesos que resultam em menor risco da carteira;
- Métodos de seleção:

**Roleta** → ordena por aptidão de acordo com um **corte**;

**Torneio** → amostra aleatória com os  $n$  melhores e repete processo.

## 2. Algoritmos Genéticos

### Seleção Mais Aptos





### 3. Operadores genéticos:

- Seleção gera uma população de potenciais cromossomos pais;
- Cromossomos parentais aleatórios para aplicar operadores;
- Operadores genéticos: **crossover** e **mutação**;
- Produz filhos até completar o tamanho da população;
- Tamanho da população se mantém o mesmo nas gerações.

### Operadores de crossover:

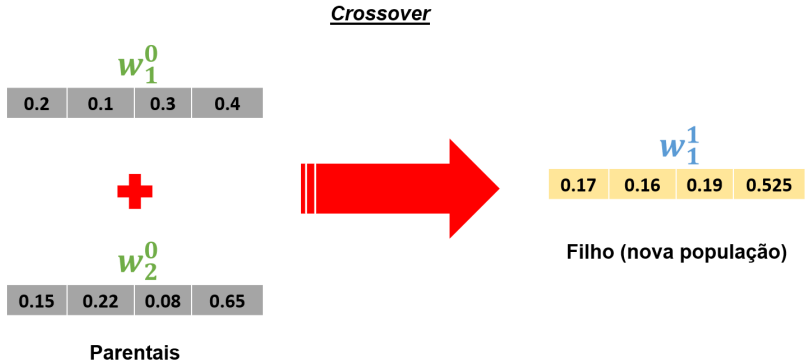
- Aplicado a um par de cromossomos para gerar filhos;
- Objetivo é herdar características dos pais. Exemplo:

$$filho = \frac{(parental_1 + parental_2)}{2} \rightarrow \text{crossover média aritmética}$$

- Para os pais  $\{0.2, 0.1, 0.3, 0.4\}$  e  $\{0.15, 0.22, 0.08, 0.65\}$ :

$$filho = \{0.175, 0.16, 0.19, 0.525\}$$

## 2. Algoritmos Genéticos



- Outros operadores de crossover:

$$filho = \sqrt{parental_1 \cdot parental_2} \rightarrow \text{crossover média geométrica}$$

$$filho = parental_1 + \beta(parental_2 - parental_1) \rightarrow \text{crossover BLX-}\alpha$$

$\beta$  aleatório em  $[-\alpha, 1 + \alpha]$ ,  $\alpha \rightarrow$  parâmetro de controle de diversidade

- Crossover é aplicado de acordo com uma probabilidade;
- **Taxa de Crossover** → geralmente entre 60-90%;
- Decisão de aplicar crossover se baseia em  $r \in [0, 1]$  aleatório:

Se  $r < \text{taxa de crossover}$ , então operador é aplicado;

Caso contrário, filhos se tornam iguais aos pais;

- Mecanismo permite que boas soluções sejam preservadas.

### Operadores de mutação:

- Operador aplicado apenas nos cromossomos filhos;
- Objetivo de aumentar a variabilidade da população;
- **Uniforme**  $\rightarrow$  substitui parâmetro por número aleatório:  $filho = \mathcal{U}[a, b]$ ;
- Operador destrutivo, portanto, aplicado a uma taxa baixa ( $\approx 0 - 20\%$ ).

$\{0.15, 0.1, 0.2, 0.425\} \rightarrow \text{mutação} \rightarrow \{0.15, 0.1, \mathbf{0.30}, 0.425\} \sim \mathcal{U}[0, 1]$

- Processo de evolução se repete sequencialmente;
- **Critérios de parada** (busca): máximo iterações ou convergência;
- AGs são métodos **especialistas**: cada problema uma parametrização;
- Diversos operadores de seleção, crossover e mutação;
- Convergência para máximo ou mínimo **global**.

- Pseudo-código geral para AGs:

---

### Algoritmos Genéticos

---

1.  $Pop(t) \leftarrow$  população de cromossomos na geração  $t$
  2.  $t \leftarrow 1$
  3. Inicializar  $Pop(t)$
  4. Avaliar  $Pop(t)$  (aplicar na função fitness/objetivo)
  5. **while** critério de parada não for satisfeito **do**
  6.  $t \leftarrow t + 1$
  7. Selecionar  $Pop(t)$  a partir de  $Pop(t - 1)$
  8. Aplicar crossover sobre  $Pop(t)$
  9. Aplicar mutação sobre  $Pop(t)$
  10. Avaliar  $Pop(t)$
  11. **end while**
- 

- Método implementado em R no pacote “**AG**”;
- Exemplo computacional na seleção de carteiras...



Outras meta-heurísticas de otimização em IA:

- Evolução Diferencial (*Differential Evolution*);
- Colônia de Formigas (*Ant Colony*);
- Abelhas (*Bee Optimization*);
- Enxame de partículas (*Particle Swarm*);
- Cardumes (*Fish Swarm*).

## ■ Resumo do curso...

- regressão, previsão, classificação e otimização;
- outros métodos de IA e ML;
- aprendizagem não supervisionada e problemas de agrupamento;
- conhecimento para potencial de aplicações.

HOLLAND, John H. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975 (2nd Edition, MIT Press, 1992).

**Prof. Leandro Maciel**

[leandromaciel@usp.br](mailto:leandromaciel@usp.br)