

# Treinamento por Algoritmos Genéticos

André Siqueira Ruela



UFOP

Universidade Federal  
de Ouro Preto



**decom**  
departamento  
de computação

# Sumário

- ▶ Revisão sobre AGs.
- ▶ Codificação de uma Rede Neural.
- ▶ AG em treinamento supervisionado.
- ▶ AG em treinamento não supervisionado.



UFOP

Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação

# Revisão: Algoritmos Genéticos

- ▶ Métodos inspirados na teoria Darwiniana da evolução das espécies pela seleção natural.
- ▶ Soluções candidatas para um determinado problema são codificadas em uma cadeia “genética”.
- ▶ Sobrevivência do mais apto: boas soluções têm mais chances de sobrevivência.
- ▶ Hereditariedade: os sobreviventes se reproduzem, passando adiante a sua carga genética.



UFOP

Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

 **decom**  
departamento  
de computação

# Revisão: Algoritmos Genéticos

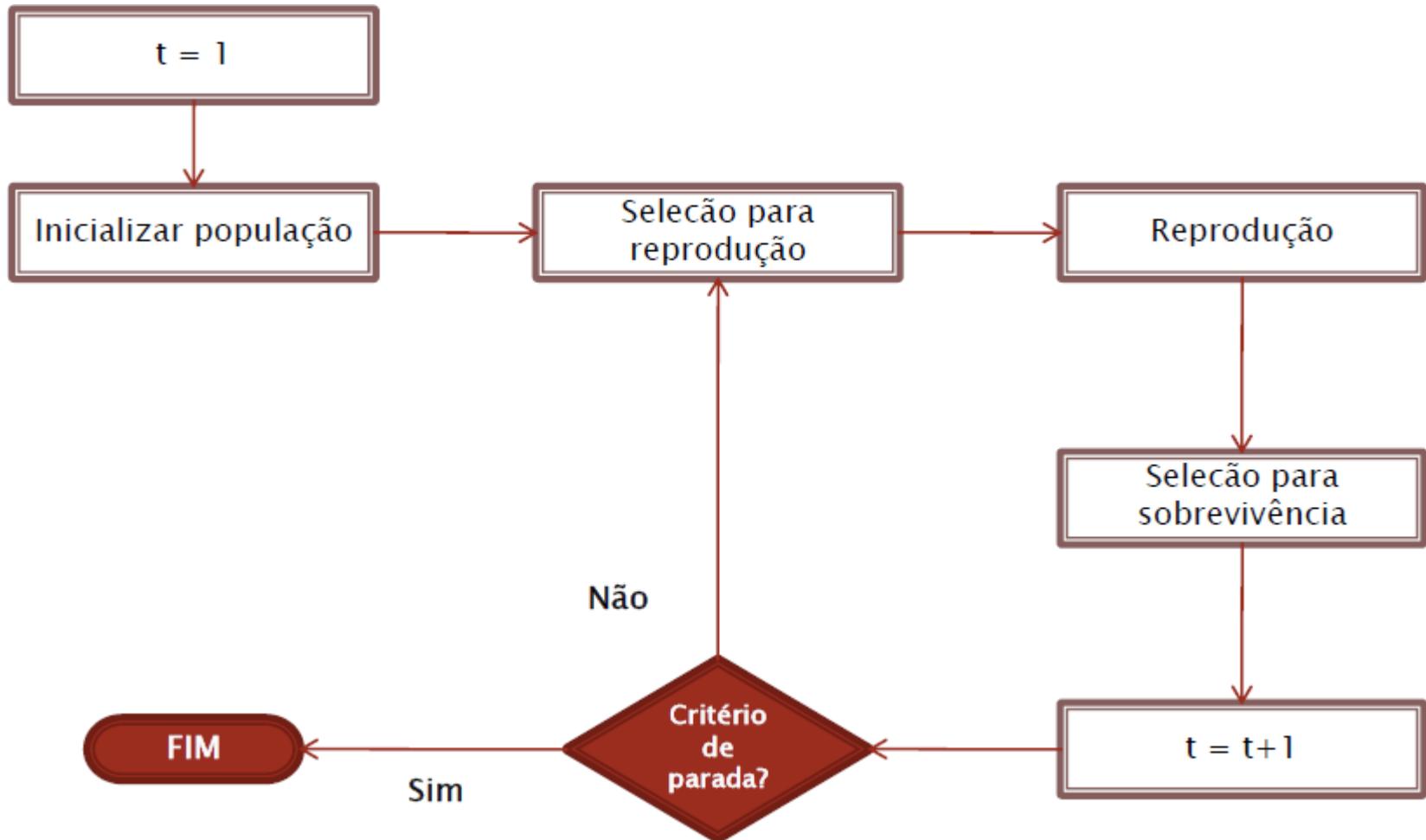
## ▶ Principais etapas:

- Inicialização da população de soluções.
- Avaliação e atribuição do valor de aptidão.
- Seleção dos mais aptos para a reprodução.
- Cruzamento: permutação das cargas genéticas.
- Mutação: alterações aleatórias nos genes.
- Seleção dos mais aptos para a sobrevivência.

## ▶ Algumas características:

- Busca populacional: várias soluções são avaliadas a cada iteração.
- Grande capacidade de evitar ótimos locais.
- Aplicável na solução de problemas não lineares e de alta complexidade.
- Fácil implementação.

# Revisão: Algoritmos Genéticos



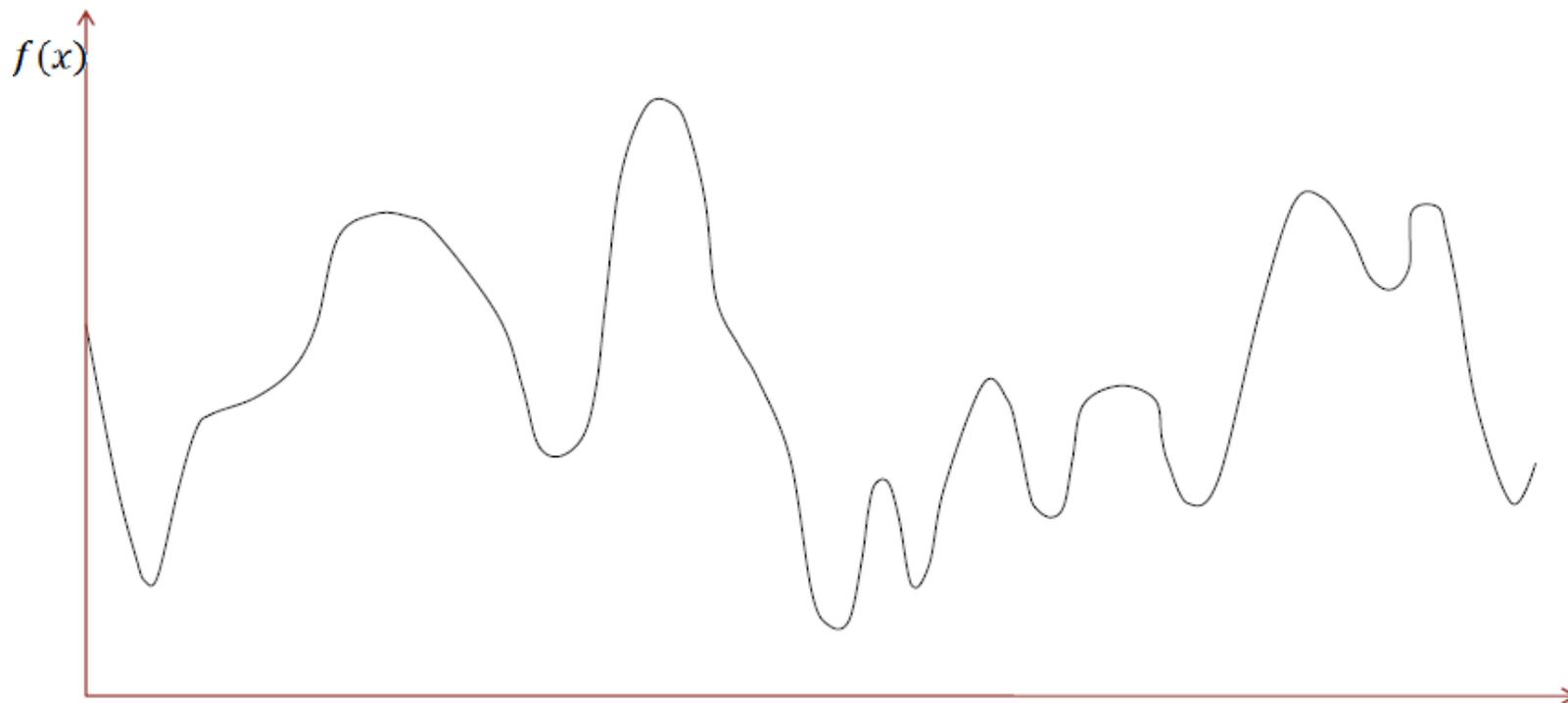
UFOP  
Universidade Federal  
de Ouro Preto

# Redes Neurais: Treinamento

- ▶ O processo de treinamento em uma RNA consiste em modificar os pesos entre os neurônios para que a rede opere de forma mais apropriada.
- ▶ Em um treinamento supervisionado, a minimização do erro global de uma RNA pode ser considerada um problema de otimização.
- ▶ Em um treinamento não supervisionado, a busca por resultados mais apropriados, considerando uma avaliação das respostas, também pode ser traduzido em um problema de otimização.

# Redes Neurais: Treinamento

- ▶ Considerando um treinamento supervisionado, podemos expressar o erro como uma função dos pesos.



UFOP  
Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação

# Redes Neurais: Codificação

- ▶ Para treinar uma RNA por meio de AGs é necessário que os pesos entre os neurônios sejam codificados em soluções candidatas.
- ▶ Ou seja, o cromossomo de cada indivíduo é composto por caracteres que codificam a matriz de pesos de cada camada da RNA.
- ▶ A partir da decodificação do cromossomo, obtém-se a estrutura da RNA.
- ▶ O erro global da RNA ou a avaliação da resposta consiste na aptidão destas soluções. Quanto menor o erro, maior a aptidão.

# Redes Neurais: Codificação

- ▶ Instancia-se agora não apenas uma RNA, mas uma população de RNAs candidatas.
- ▶ As RNAs mais aptas tendem a sobreviver, reproduzir e a propagar bons pesos (genes) pelas gerações futuras.
- ▶ Após um considerável número de iterações (gerações), soluções satisfatoriamente boas emergem, sendo a melhor solução retornada.



UFOP

Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

 **decom**  
departamento  
de computação

# Treinamento supervisionado: XOR

- ▶ A Heaton Research fornece um AG genérico para a adaptação a qualquer problema.
- ▶ O AG implementado ainda conta com o suporte de um *pool* de *threads* para a execução em plataformas multiprocessadas.
- ▶ Exemplos:
  - TrainingSetNeuralGeneticAlgorithm.
  - TrainingSetNeuralChromosome.
  - NeuralGeneticAlgorithm.
  - NeuralChromosome.
  - MatrixCODEC.

# Treinamento não supervisionado: Jogo da Velha

- ▶ O Jogo da Velha é um jogo simples no qual o espaço de busca pode ser facilmente explorado por algoritmos gulosos.
- ▶ Para cada entrada, não é fornecida uma resposta ideal. O ideal consiste na vitória e esta pode ser obtida de várias formas.
- ▶ Nesse caso, o número de vitórias obtido pela RNA pode ser tomado como valor de aptidão.



UFOP  
Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação

# Treinamento não supervisionado: Jogo da Velha

- ▶ A Heaton Research fornece uma implementação baseada no TicTacToe fornecido por Thomas David Baker.
- ▶ Existem diferentes tipos de jogadores prontos:
  - Boring: seleciona sempre a próxima posição.
  - Humano: permite que um humano jogue.
  - Logical: usa a lógica para jogar próximo ao perfeito.
  - MinMax: utiliza o paradigma Min-Max (melhor jogador).
  - Random: seleciona uma posição aleatória.
- ▶ Iremos desenvolver o NeuralPlayer

# Treinamento não supervisionado: Jogo da Velha

- ▶ Argumentos:
  - [opção] [jogador1] [jogador2]
- ▶ Opções de uso:
  - game: um jogo entre um Humano e o jogador2.
  - match: 100 partidas entre os dois jogadores.
  - train: treina uma RNA contra o jogador2.
- ▶ A opção train carrega do disco rígido uma rede neural predeterminada. Caso ela ainda não exista, uma nova rede é criada.



UFOP

Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

 **decom**  
departamento  
de computação

# Treinamento não supervisionado: Jogo da Velha

- ▶ O processo de treinamento de uma rede é longo e pode levar horas.
- ▶ Inicialmente, uma rede neural sempre perde para um jogador MinMax. Após várias iterações (épocas), bons resultados podem ser obtidos.
- ▶ Após o treinamento, a rede é salva novamente no disco rígido para futura utilização.



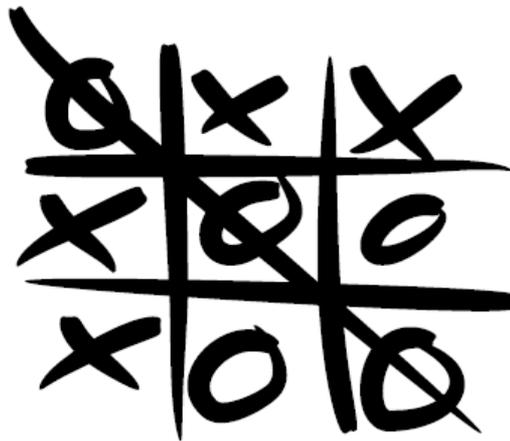
UFOP  
Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação

# Treinamento não supervisionado: Jogo da Velha

- ▶ A estrutura da rede neural mais óbvia talvez seja 9 neurônios na camada de entrada e 9 neurônios na camada de saída.
- ▶ Esta estratégia não é muito boa, pois a rede perde muito tempo treinando para identificar jogadas inválidas.



# Treinamento não supervisionado: Jogo da Velha

- ▶ A estrutura utilizada possui 9 neurônios na camada de entrada, 10 na camada intermediária e 1 neurônio de saída.
- ▶ Uma diferente entrada é fornecida para cada posição válida no tabuleiro.
- ▶ A saída consiste em uma pontuação obtida pela entrada.
- ▶ A entrada que obtiver a maior pontuação consiste na próxima posição a ser jogada.

# Treinamento não supervisionado: Jogo da Velha

- ▶ O NeuralPlayer possui apenas dois métodos: getMove e tryMove.
- ▶ getMove: gera todos os pares x e y de posições disponíveis no tabuleiro e para cada uma chama o método tryMove. Retorna o par x e y de maior pontuação.
- ▶ tryMove: gera um conjunto de entrada, decodifica a rede neural, apresenta a entrada e retorna a pontuação.
- ▶ Cada NeuralPlayer é realiza 100 jogos para avaliação.



UFOP  
Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação

# Treinamento não supervisionado: Jogo da Velha

- ▶ Exemplos:
  - NeuralTicTacToe.
  - TicTacToeGenetic.
  - TicTacToeChromosome.
  - ScorePLayer.
  - SerializeObject.



UFOP

Universidade Federal  
de Ouro Preto

**SEVA**  **Mobilis**

  
**decom**  
departamento  
de computação