Biohacking, Deep WEB & Criptografia

Aula 03

Computação Evolucionária

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Objetivo

Introduzir os conceitos da Computação Evolucionária (CE), explorando algoritmos genéticos (AG) e suas aplicações, e desenvolver um pequeno experimento prático.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Como a natureza encontra soluções para problemas complexos?



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Como a natureza encontra soluções para problemas complexos?

A evolução natural nos mostra que organismos se adaptam ao ambiente ao longo do tempo por meio de **mutações**, **seleção natural** e **reprodução**. Esse conceito pode ser aplicado para resolver problemas computacionais difíceis de maneira inteligente.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Exemplo

- **Pense em como os pássaros desenvolveram asas ao longo de milhões de anos. Eles não tentaram "adivinhar" a melhor forma de voar de uma vez; pequenos ajustes em cada geração resultaram em melhorias progressivas.**
- **O mesmo ocorre na Computação Evolucionária: soluções são geradas, testadas, e as melhores são refinadas ao longo do tempo.**

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

O que é Computação Evolucionária?

A Computação Evolucionária é uma área da Inteligência Artificial que utiliza conceitos inspirados na evolução biológica para resolver problemas de otimização e busca.

"Computação Evolucionária é um conjunto de algoritmos baseados em princípios da seleção natural e genética para encontrar soluções eficientes para problemas complexos."



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

O que é Computação Evolucionária?

A Computação Evolucionária é uma área da Inteligência Artificial que utiliza conceitos inspirados na evolução biológica para resolver problemas de otimização e busca.

"Computação Evolucionária é um conjunto de algoritmos baseados em princípios da seleção natural e genética para encontrar soluções eficientes para problemas complexos."



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Por que usar Computação Evolucionária?

Existem problemas para os quais não há uma fórmula matemática exata que resolva de forma direta. Algoritmos tradicionais muitas vezes são ineficientes quando o espaço de busca é muito grande. A CE é útil porque pode encontrar boas soluções aproximadas em um tempo razoável.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Comparação com outras técnicas de otimização

Método	Vantagens	Desvantagens
Força Bruta	Garante a melhor solução	Extremamente lento para problemas grandes
Gradiente Descendente	Rápido para funções diferenciáveis	Pode ficar preso em mínimos locais
Computação Evolucionária	Funciona em problemas complexos e não lineares	Pode ser mais lenta do que métodos especializados

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Aplicações Reais

A Computação Evolucionária é usada em diversas áreas, como:

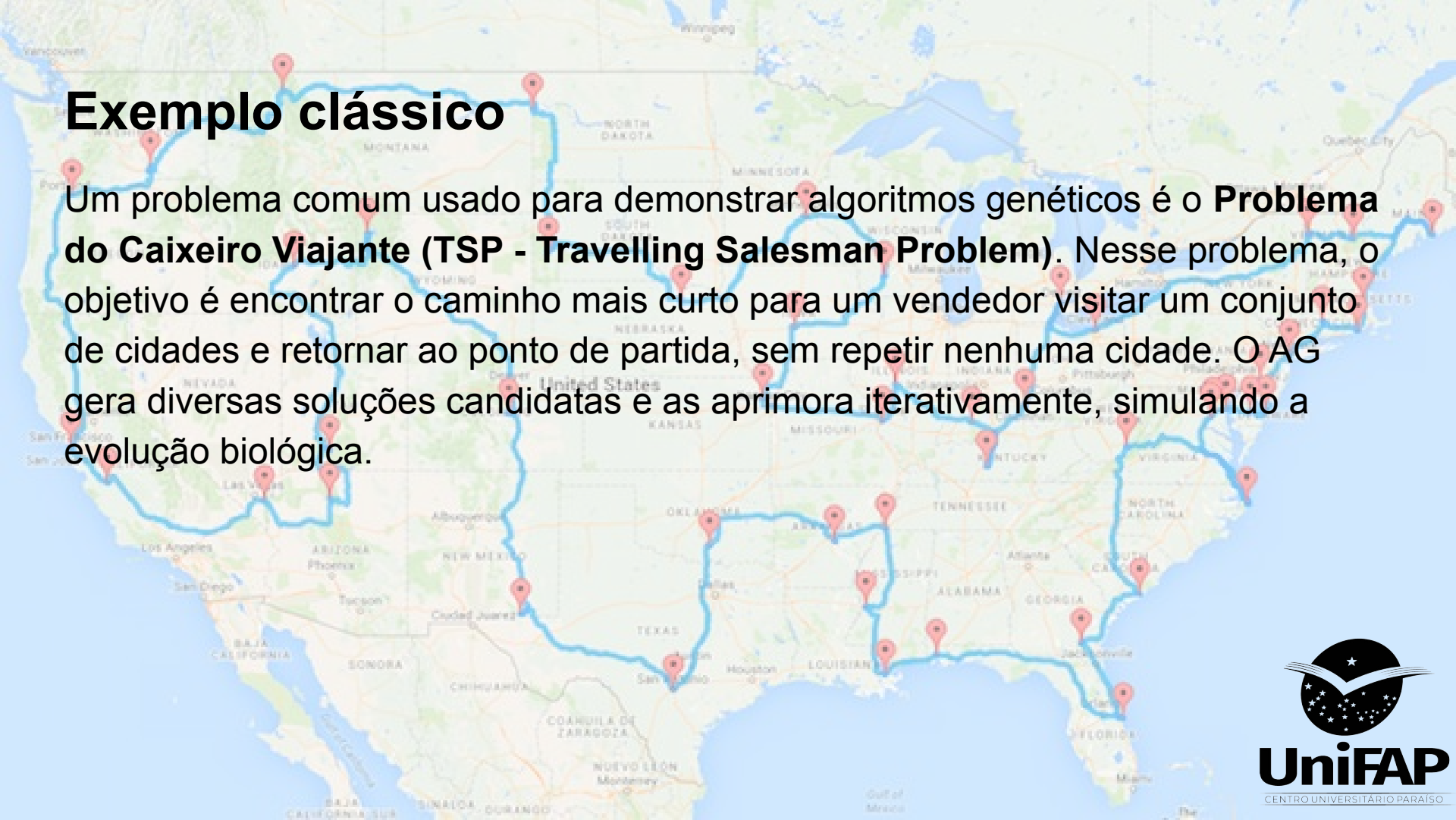
- **Otimização de rotas** (ex: planejamento de logística, roteamento de veículos)
- **Criação de redes neurais** (ex: otimização de hiperparâmetros)
- **Bioinformática** (ex: análise de DNA)
- **Jogos e simulações** (ex: NPCs aprendendo estratégias automaticamente)
- **Engenharia e design** (ex: aerodinâmica de asas de aviões)



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Exemplo clássico

Um problema comum usado para demonstrar algoritmos genéticos é o **Problema do Caixeiro Viajante (TSP - Travelling Salesman Problem)**. Nesse problema, o objetivo é encontrar o caminho mais curto para um vendedor visitar um conjunto de cidades e retornar ao ponto de partida, sem repetir nenhuma cidade. O AG gera diversas soluções candidatas e as aprimora iterativamente, simulando a evolução biológica.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Algoritmos Genéticos (AG) - Conceitos Fundamentais



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Algoritmos Genéticos (AG) - Conceitos Fundamentais

Os Algoritmos Genéticos (AG) são um dos métodos mais populares dentro da Computação Evolucionária. Eles simulam o processo da evolução biológica para encontrar soluções eficientes para problemas difíceis.

Ideia principal:

Em vez de testar todas as possibilidades (como na força bruta), um AG cria uma **população inicial** de soluções e, ao longo de várias gerações, melhora essas soluções por meio de **seleção, cruzamento e mutação**.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Algoritmos Genéticos (AG) - Conceitos Fundamentais

Mecanismos principais:

1. **População inicial:** representação das soluções.
2. **Avaliação (Fitness Function):** como medir a qualidade de uma solução.
3. **Seleção:** Métodos como roleta e torneio.
4. **Cruzamento (Crossover):** combinação de soluções.
5. **Mutação:** introdução de diversidade.
6. **Critério de parada:** como definir o fim da evolução.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Mecanismos principais do Algoritmo Genético

O AG segue um fluxo baseado em etapas fundamentais:

```
Indivíduo 1: 101010
Indivíduo 2: 110011
Indivíduo 3: 000111
...
```

População inicial

- O algoritmo começa gerando várias **soluções aleatórias** para o problema.
- Cada solução é chamada de **indivíduo** e geralmente é representada como um **vetor de números** ou uma **string de bits**.
- Conjunto dessas soluções = **população** inicial.

Exemplo:
Se estivermos resolvendo um problema de maximização de uma função matemática, cada indivíduo pode ser um número representado em binário, como:

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Avaliação (Fitness Function)

Cada indivíduo precisa ser avaliado para saber **quão bom ele é**. Isso é feito por meio da **função de aptidão (fitness function)**, que mede o desempenho da solução.

Exemplo:
Se estamos tentando encontrar **o maior valor possível de uma função f(x)**, a função de aptidão pode ser simplesmente:

```
Fitness = f(x)
```

Quanto maior o valor de f(x), melhor o indivíduo.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Seleção

Agora que temos os indivíduos avaliados, precisamos escolher **quais serão os pais** para gerar a próxima geração.

Métodos comuns de seleção:

1. **Roleta (Seleção por Fitness Proporcional)**
    - Indivíduos com **melhor aptidão** têm maior chance de serem escolhidos.
    - Como uma roleta de cassino, onde setores maiores correspondem a soluções melhores.
2. **Torneio**
    - Escolhe-se um pequeno grupo aleatório da população.
    - O melhor entre eles é selecionado para reprodução.

Exemplo:
Se temos uma população de 5 indivíduos com fitness [10, 20, 30, 40, 50], no método da roleta, o indivíduo com fitness 50 tem maior chance de ser escolhido do que o de fitness 10.

Cruzamento (Crossover)

Depois de escolher os pais, eles precisam gerar **novos indivíduos (filhos)** combinando suas características.

📌 **Métodos comuns:**

1. **Crossover de um ponto** (One-point crossover)
    - Escolhe-se um ponto no vetor dos pais e troca-se a parte restante.

```
Pai 1:  101|010
Pai 2:  110|011
Filho:  101011
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Cruzamento (Crossover)

2. Crossover de dois pontos

- Escolhem-se dois pontos e troca-se a parte entre eles.

3. Crossover uniforme

- Cada bit tem uma chance de ser trocado aleatoriamente.

⚠️ **Se não houver cruzamento, o AG vira uma simples seleção aleatória!**

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Mutação

Para manter **diversidade** na população e evitar que o algoritmo fique preso em soluções ruins, aplicamos mutação.

- A mutação altera **pequenas partes do indivíduo** aleatoriamente.
- Exemplo: inverter um bit (0 → 1 ou 1 → 0)

```
Antes: 101010
Depois: 101110  (mutação no 4º bit)
```

Taxa de mutação geralmente é baixa (ex: **1%** das soluções mutam).

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Critério de parada

O algoritmo precisa parar em algum momento.

Algumas formas comuns de definir isso:

- Número fixo de gerações (ex: **100 gerações**).
- Quando a solução atingir um valor satisfatório (ex: **fitness ≥ 95%**).
- Quando não houver mais melhoria significativa entre gerações.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Experimento

https://colab.research.google.com/drive/1epKJpC8hzH5szIXYszQfPTksRMRfIP1Y?usp=sharing

https://github.com/matheusfd3/IA-flappy-bird



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Link interessantes

https://www.redblobgames.com/

https://www.youtube.com/watch?v=NZlIYr1slAk

https://www.youtube.com/watch?v=P7XHzqZjXQs&t=1550s

https://www.inf.ufpr.br/aurora/disciplinas/datamining/Ceapostila.pdf



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Bibliografia

PEREIRA, Eduardo. Computação evolucionária: aplique os algoritmos genéticos com Python e Numpy. São Paulo, SP: Casa do Código, 2020. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 28 fev. 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/212778

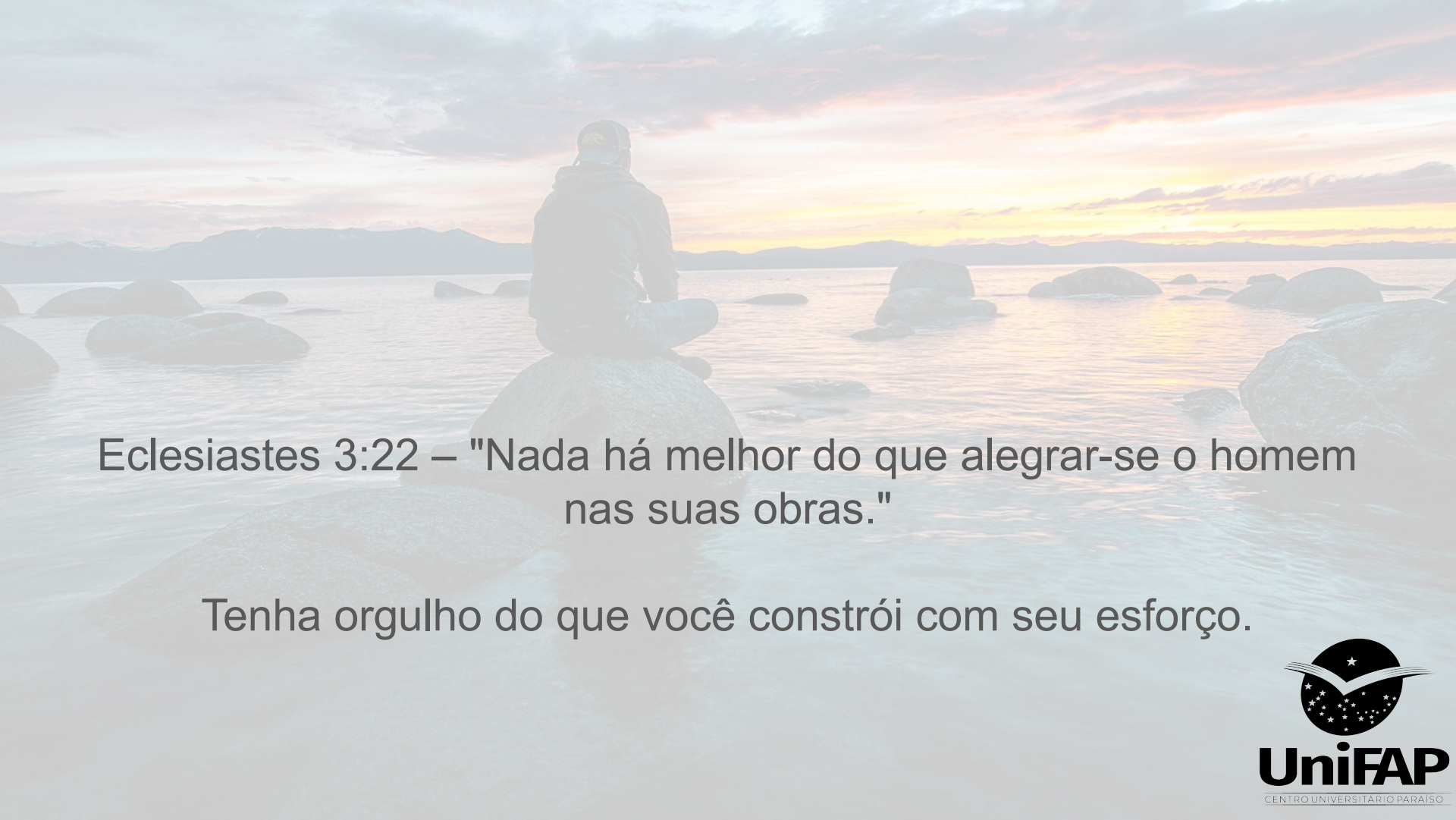


UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Perguntas?!



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Eclesiastes 3:22 – "Nada há melhor do que alegrar-se o homem nas suas obras."

Tenha orgulho do que você constrói com seu esforço.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Obrigado!

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO