



# Algoritmos Genéticos e Computação Evolutiva

---

## Inteligência Artificial

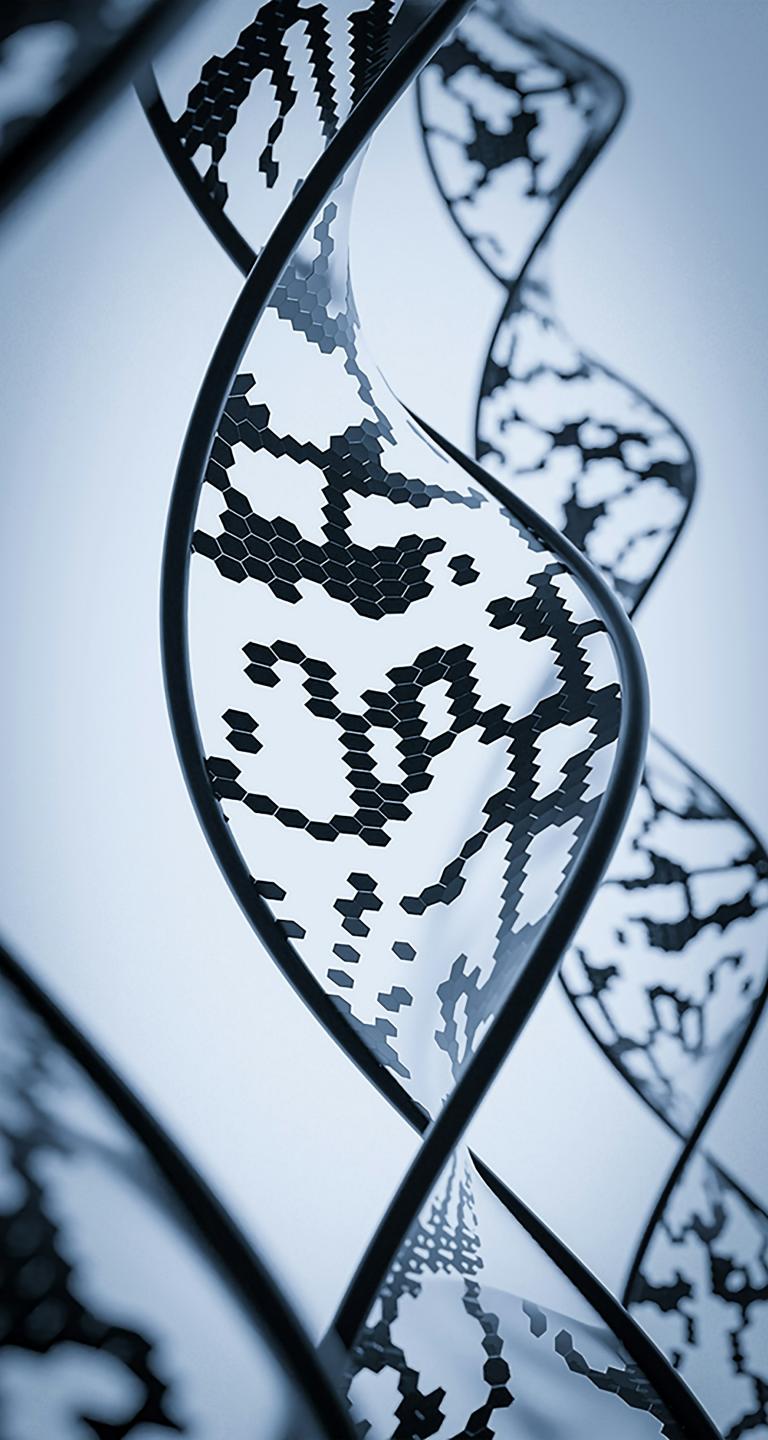
Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

# Objetivos de Aprendizagem

---

- **Definir** a Computação Evolutiva e os Algoritmos Genéticos (AGs) como ferramentas de otimização e busca.
- **Identificar** os componentes fundamentais de um AG: indivíduo, população, função fitness e operadores.
- **Detalhar** os principais operadores genéticos (Seleção, Crossover e Mutação) e suas variações.
- **Analizar** a importância da diversidade populacional, do elitismo e da convergência.



# O Conceito de Computação Evolutiva

---

- Computação Evolutiva (**CE**) é o termo que engloba técnicas de resolução de problemas baseadas em princípios da evolução biológica.
- A **inspiração central** inclui a seleção natural e a herança genética.
- O objetivo da CE é usar o comportamento adaptativo da natureza para resolver problemas computacionalmente difíceis, como otimização e busca.

# O que é um Algoritmo Genético (AG)?

---

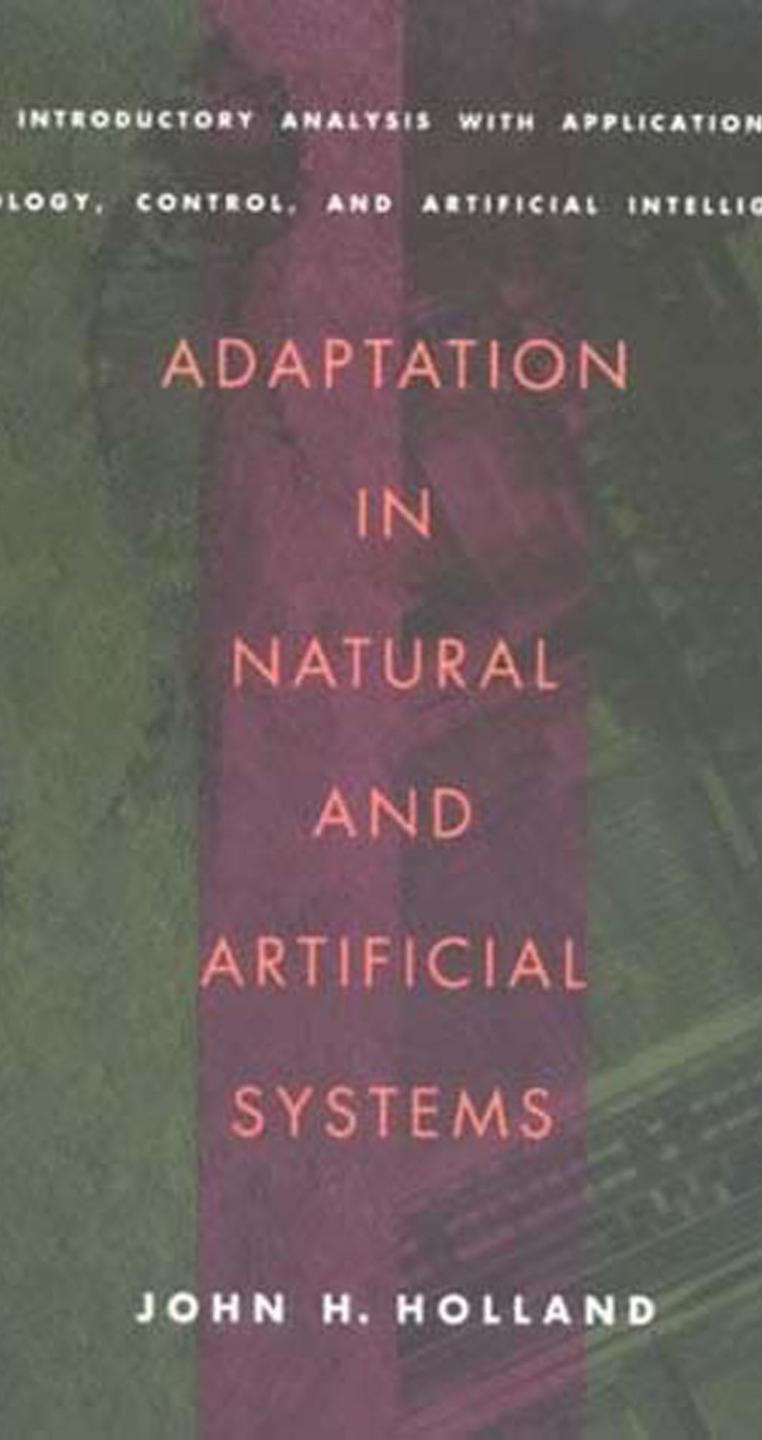
- Um **AG** é uma técnica de busca e otimização utilizada para achar soluções aproximadas.
- Eles são uma forma de **otimização global** e buscam pontos de "alta aptidão".
- AGs operam sobre uma **população** de soluções candidatas que são selecionadas para buscar soluções melhores.
- A evolução se dá por meio de **iterações** (gerações), onde os indivíduos são avaliados, selecionados, recombinados e mutados.
- **Diferenças Chave dos Algoritmos Tradicionais:**
  - Os resultados são apresentados como uma **população de soluções**, não uma solução única.
  - Usam **transições probabilísticas**, e não regras determinísticas.

Survival of the fittest



# Inspiração Biológica

- **Seleção Natural (Darwin-Lawrence):** Organismos mais bem adaptados ao meio têm maiores chances de sobrevivência e de deixar descendentes.
- O princípio básico da evolução é: "Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes".
- AGs imitam mecanismos evolutivos: **Hereditariedade, Mutação, Seleção Natural e Recombinação.**
- AGs aplicam os princípios de seleção e reprodução a uma população de candidatos.



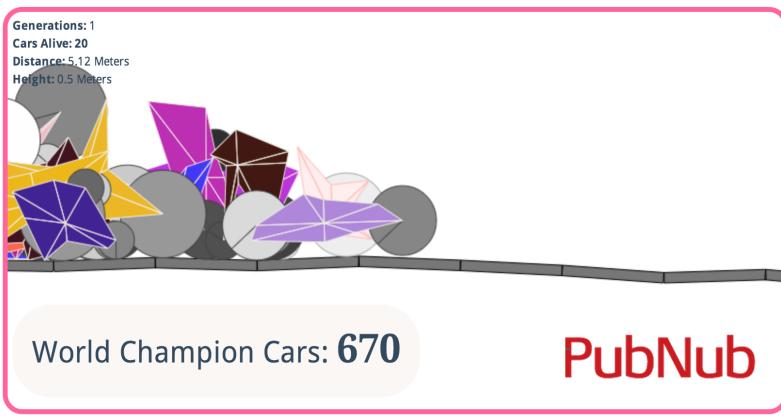
# Contexto Histórico: Os Fundamentos

- **John Henry Holland:** O principal fundador e desenvolvedor dos conceitos de Algoritmos Genéticos.
- Pesquisas iniciais começaram seriamente nos anos 50 e 60, com o desenvolvimento de simulações computacionais de sistemas genéticos.
- **Marco Central (1975):** Holland publica *Adaptation in Natural and Artificial Systems*.
  - Este trabalho estabeleceu a estrutura teórica e formal do campo.
  - Desde então, AGs têm sido aplicados com sucesso em otimização e aprendizado de máquina.

# Grow Your Own Picture

<https://chriscummins.cc/s/genetics/>

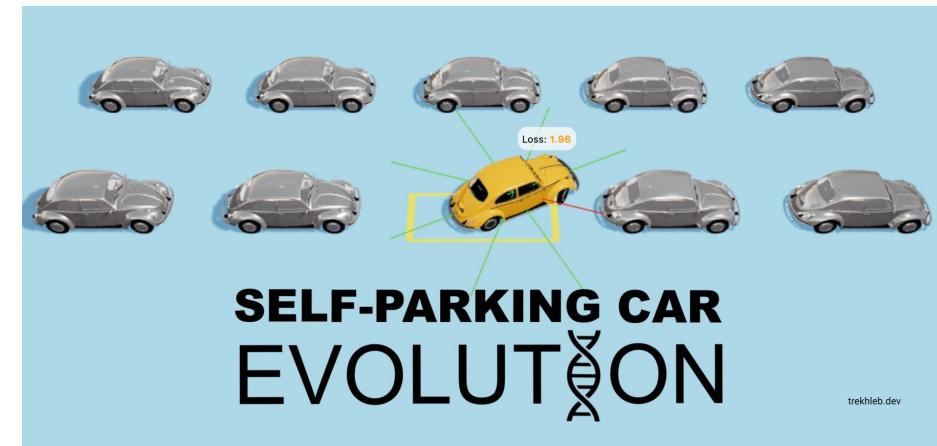
# Mais Exemplos



Genetic Cars

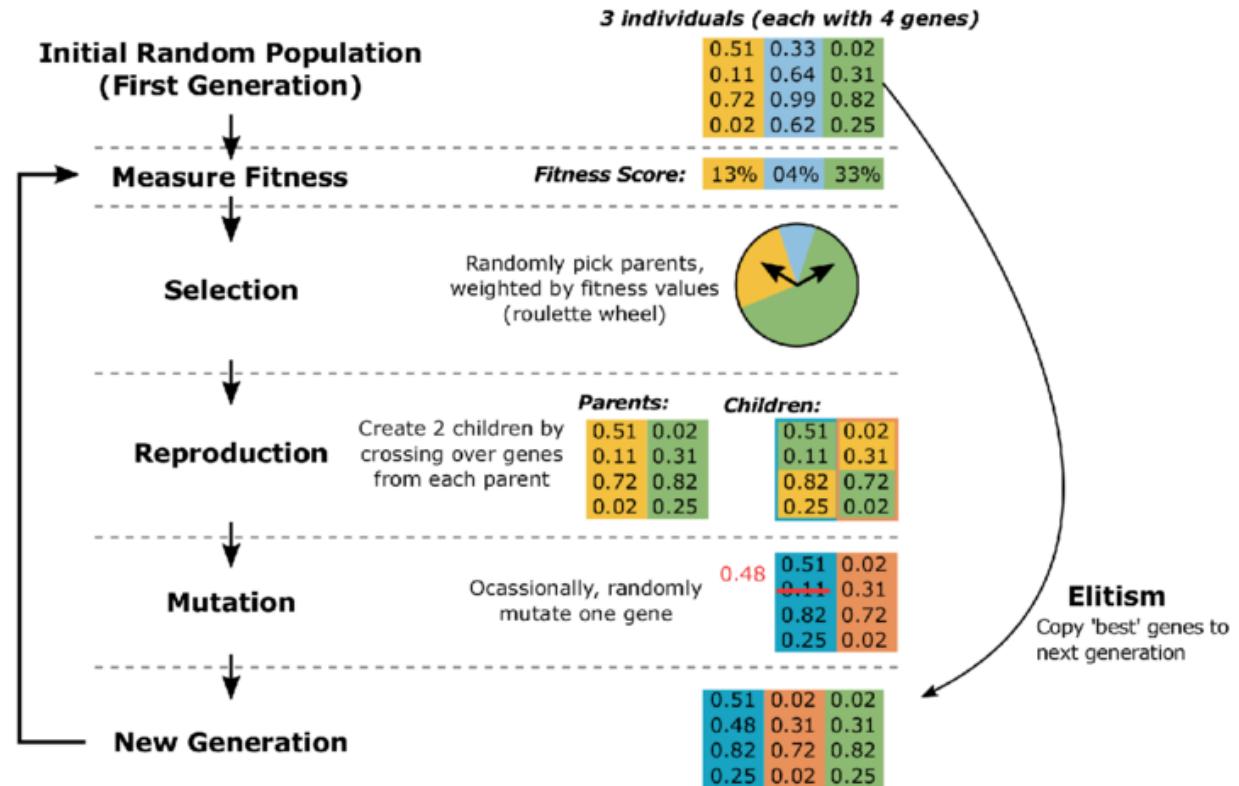
## Genetic Cars 2: Multiplayer

VERSION 2.51  
[Tweet](#) 69  
Distributed Computing Powered  
[Follow @PubNub](#) [Fork](#) [Star](#)  
Powered by [PubNub Real-time Network](#)  
[Speed UP! \( hide cars \)](#)  
[New Population](#)  
Join World:  [Go!](#)



Self-Parking Car Evolution

# Algoritmo Genético: Visão Geral



AGs trabalham com uma população de candidatos **em paralelo**, buscando em diferentes áreas do espaço de solução, diferentemente de métodos tradicionais que usam um único candidato. Fonte da Imagem: [ResearchGate](#).

# Elementos Fundamentais: População e Indivíduo

---

- **População:**
  - É o conjunto de soluções candidatas (indivíduos).
  - Geralmente iniciada de forma **aleatória** para cobrir o espaço de busca e evitar vieses.
  - Um tamanho grande aumenta a diversidade e a chance de explorar bem o espaço, mas eleva o custo computacional.
- **Indivíduo (Cromossomo):**
  - É uma representação abstrata de uma única solução para o problema.

# Codificação e Representação Genética

- O **código genético** é uma representação do estado no espaço de busca do problema.
- O **cromossomo** é composto de **genes**, que são as variáveis ou partes da solução.
- A escolha da **representação** afeta diretamente como o crossover e a mutação serão realizados.

Tipo de Representação	Descrição e Uso	Exemplo
Binária (Clássica)	Sequências de 0s e 1s, muito usada em problemas de otimização clássicos.	<b>Knapsack Problem</b>
Inteira (Permutação)	Usada quando a ordem é importante.	<b>Caixeiro Viajante (TSP)</b>
Real (Float)	Útil para ajuste fino e otimização contínua.	<b>Ajuste de Parâmetros:</b> [0.25, 1.76, -3.14]

# A Função de Avaliação (Fitness Function)

---

- A **função objetivo** é o objeto da otimização. É a ligação do algoritmo com o problema real.
  - Mede o quão "apto" (bom) um indivíduo é para o ambiente (solução do problema).
  - Quanto **maior** o valor do **fitness**, **melhor** a solução.
- O AG não precisa saber como a função funciona ("caixa preta"), apenas ter acesso à avaliação para comparar resultados.
- **Tratamento de Restrições:** Soluções que violam restrições devem ser **penalizadas** na função **fitness**.

# O Ciclo do Algoritmo Genético

---

O AG segue um ciclo iterativo, simulando gerações:

1. **População Inicial:** Geração aleatória de soluções.
2. **Avaliação:** Cálculo do **fitness** para cada indivíduo.
3. **Seleção:** Escolha dos pais (indivíduos mais aptos) para a reprodução.
4. **Reprodução:** Geração de filhos (descendentes) através de:
  - **Crossover (Recombinação):** Troca de material genético.
  - **Mutação:** Introdução de variabilidade genética.
5. **Nova Geração:** Os filhos (e os pais, se houver elitismo) formam a nova população.
6. **Critério de Parada:** Repetir até que uma condição seja atingida (ex: número máximo de gerações).

# Algoritmo Genético Aprende a Jogar Super Mario World

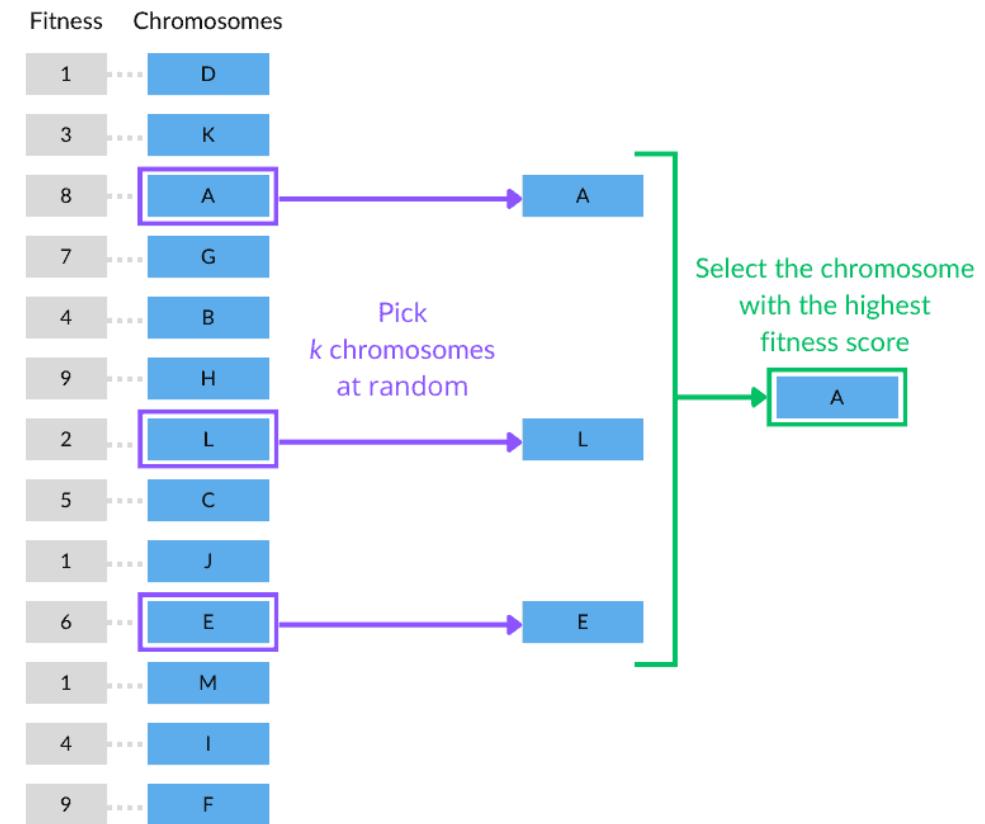


Algoritmo Genético Aprende a Jogar Super Mario World

Veja também o Algoritmo A\* Jogando Mario

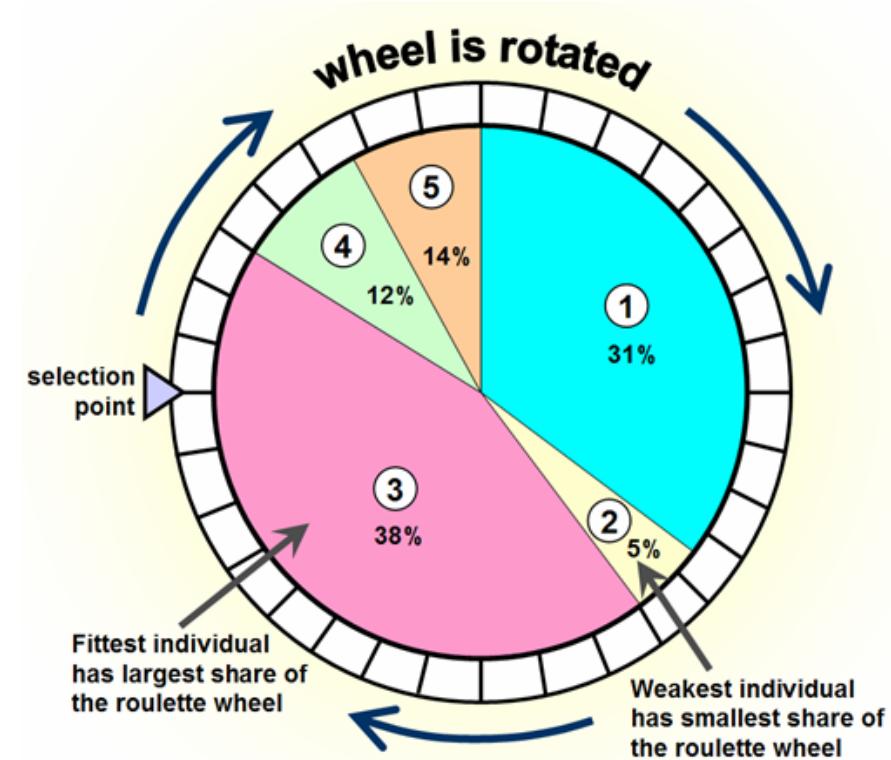
# Operador Básico: Seleção - Escolhendo os Pais Mais Aptos

- Determinar quais indivíduos da população atual terão mais chances de passar seus genes adiante.
- **Objetivo Duplo:**
  - i. Dar preferência a indivíduos com melhor **fitness** (exploração de boas áreas).
  - ii. Manter a diversidade genética (evitando convergência prematura).
- A maioria dos métodos de seleção é projetada para escolher preferencialmente indivíduos com maiores notas de aptidão, mas não exclusivamente.



# Seleção por Roda da Roleta (Roulette Wheel Selection)

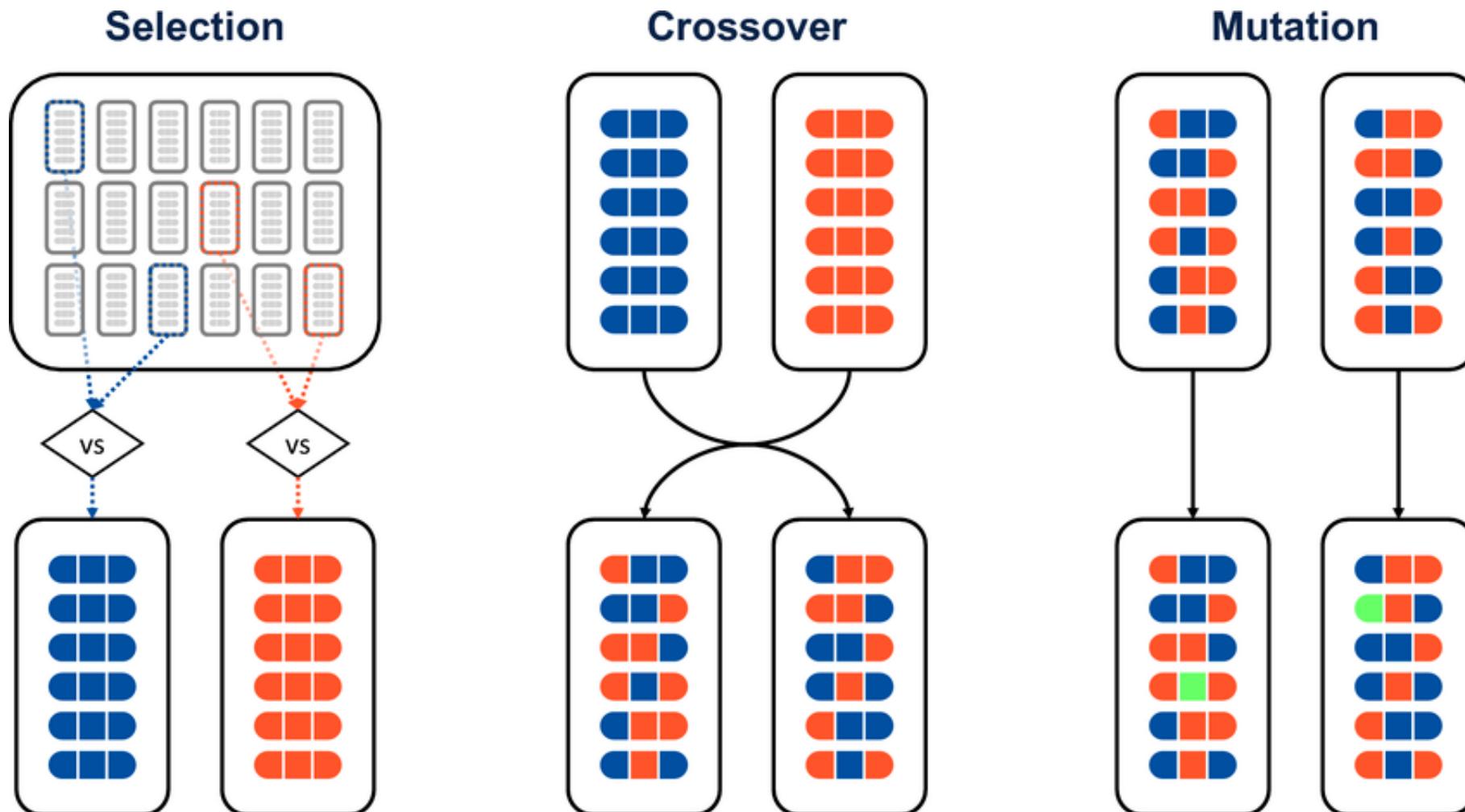
- **Princípio:** Os indivíduos são escolhidos de forma aleatória, mas a probabilidade de escolha é proporcional ao seu **fitness**.
- **Mecanismo:** Cada indivíduo ocupa uma porção da roleta proporcional ao seu índice de aptidão em relação ao total da população.
- Indivíduos com alta aptidão recebem uma porção maior, aumentando sua chance de serem sorteados como pais.
- **Vantagem:** Permite que indivíduos menos aptos ainda tenham uma chance, ajudando a manter a diversidade.



# Operador Básico: Crossover (Recombinação)

O Crossover é o **operador predominante**. É aplicado com uma alta probabilidade  $P_c$ , normalmente entre 70% a 90%. Simula a recombinação genética para garantir que os descendentes herdem características dos pais.

Tipo de Crossover	Mecanismo	Ilustração (Binário)
Um-Ponto	Um ponto de corte é escolhido; as informações genéticas são trocadas a partir dele.	P1 [111]
Dois-Pontos	Escolhe-se dois pontos de corte; a parte intermediária é trocada.	P1 [10]
Uniforme	Cada posição (gene) tem uma probabilidade de vir de um dos pais (ex: 50% de chance para cada).	$P1 + P2 \rightarrow F1$



Fonte: [ResearchGate](#).

# Operador Básico: Mutação - Explorando o Espaço de Busca

---

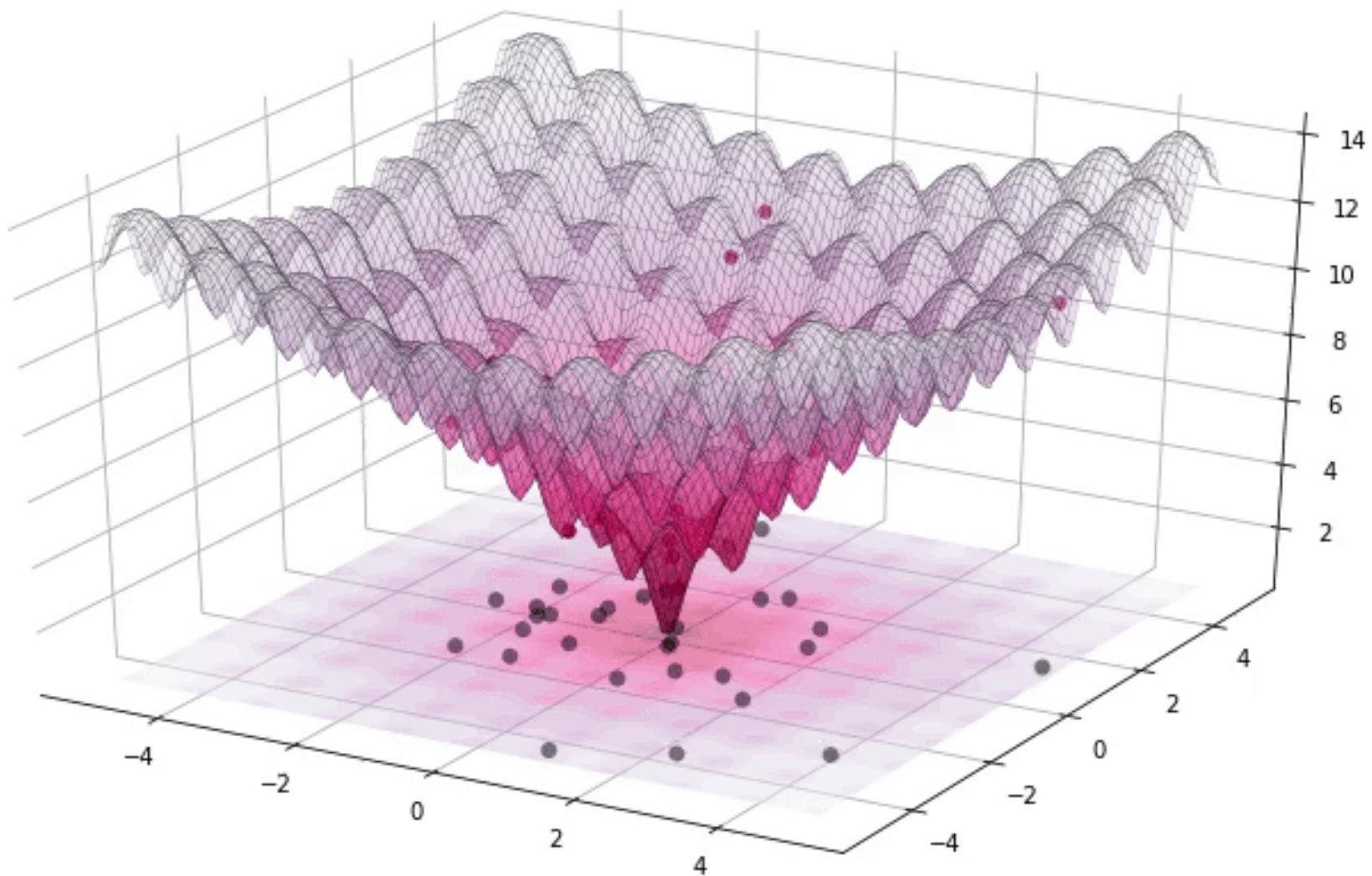
- Introduzir e manter a **diversidade genética** na população.
- Aplicações de mutação são feitas com a **probabilidade mais baixa possível**  $P_m$ , geralmente 0.1%.
- **Benefício:** Impede que a busca fique estagnada em um mínimo local, alterando levemente a direção da busca.

Tipo de Mutação	Representação	Mecanismo
Flip-bit	Binária	Inverte um bit ( $0 \rightarrow 1$ ou $1 \rightarrow 0$ ).
Gaussiana	Real	Adiciona um ruído aleatório (distribuição Gaussiana) ao valor.
Adaptativa	Geral	A taxa de mutação varia: alta no início para exploração, baixa no final para refinamento da solução.

# Diversidade e Convergência Prematura

---

- **Diversidade Genética:** A população deve ter variações suficientes entre os indivíduos para explorar diferentes regiões do espaço de busca.
- **Risco da Baixa Diversidade:** Se a população for muito parecida, o algoritmo perde a capacidade de encontrar soluções melhores.
- **Convergência Prematura:** Ocorre quando o algoritmo converge rapidamente para uma solução que **não é a ótima global** (solução subótima).
  - Causada por: Pressão seletiva muito alta (ex: selecionar **apenas** os melhores) ou **fitness** muito desbalanceado.
- **Estratégias de Evitar:** Permitir a seleção de indivíduos medianos, e usar taxas de mutação mais altas nas primeiras gerações.



Exemplo de Convergência. Fonte: [Pablo Rodriguez-Mier](#).

# Elitismo: Preservando o Conhecimento Adquirido

---

- **O Problema:** Como o processo de reprodução envolve sorte (seleção, crossover e mutação), o melhor indivíduo pode ser descartado.
- **Elitismo:** É uma estratégia comum que consiste em **preservar os melhores indivíduos (elites)** da geração atual, copiando-os diretamente para a próxima geração sem alterações.
- **Benefícios:**
  - Garante que as boas soluções conquistadas não sejam perdidas.
  - Acelera o processo de evolução e a convergência para soluções de alta qualidade.
  - É altamente recomendado, especialmente em AGs elitistas.

# Considerações Práticas: Ajuste de Parâmetros

O ajuste (tuning) de parâmetros influencia o desempenho global e a eficiência dos AGs.

Parâmetro	Descrição e Heurísticas Comuns	Impacto
Tamanho da População	50 a 200 indivíduos (Ex.).	Maior diversidade, maior custo.
Taxa de Crossover ( $P_c$ )	70% a 90%.	Muito alta: pode quebrar estruturas de alta aptidão (boas soluções).
Taxa de Mutação ( $P_m$ )	0.1% a 5% (baixa).	Muito alta: busca se torna essencialmente aleatória.
Elitismo	Número de melhores indivíduos preservados.	Garante que o melhor não seja perdido.

# Considerações Práticas: Critérios de Parada

---

Os AGs rodam em um ciclo de gerações até que uma condição de parada seja atingida.

- **Número Máximo de Gerações:** O AG para após um número pré-definido de iterações (ex: 1000).
- **Tempo Máximo de Execução:** O AG roda por um limite de tempo (ex: 10 minutos).
- **Alcançar Fitness Satisfatório:** A busca termina quando a função objetivo atinge um valor desejado (ideal para problemas com soluções ótimas conhecidas).
- **Convergência da População:** O algoritmo para quando a diversidade é muito baixa (todos os indivíduos estão muito parecidos).
- **Estagnação (Sem Melhoria):** Não há melhoria no melhor indivíduo após X gerações consecutivas.

# Aplicações de Algoritmos Genéticos

---

AGs são eficientes para buscar soluções ótimas ou aproximadamente ótimas em problemas complexos onde métodos tradicionais encontram limitações.

- **Otimização de Sistemas Complexos:** Configuração de sistemas, alocação de tarefas e seleção de rotas.
- **Problemas de Busca e Otimização:** Otimização de hiperparâmetros de modelos, e otimização de aprendizagem de árvore de decisão.
- **Exemplo Clássico: Problema da Mochila** - Maximizando o valor total dos itens sem exceder o limite de peso.
- **Simulação de Modelos Biológicos:** Comportamento e evolução.
- **Outros Exemplos:** Resolução de algoritmos de Sudoku, Engenharia de Sistemas Neurais Artificiais e Composição Musical.

# Resumo e Perspectivas

---

- Um **Algoritmo Genético (AG)** é uma meta-heurística de busca e otimização, baseada na **evolução biológica**. Eles representam uma classe de **algoritmos evolutivos** que buscam soluções aproximadas para problemas complexos.
- **Princípios e Mecanismos:** Os AGs utilizam técnicas inspiradas pela biologia evolutiva, como hereditariedade, mutação, seleção natural e recombinação.
- AGs operam sobre uma **população de candidatos** em paralelo e utilizam **codificações** das soluções, e não os parâmetros em si. Eles usam **transições probabilísticas** e são robustos para problemas com descontinuidades, pois não dependem de derivadas.
- **Aplicações e Extensões:** AGs são aplicados em otimização, aprendizado de máquina (como otimização de hiperparâmetros) e resolução de problemas combinatórios (ex: Sudoku).
- Extensões como a **Programação Genética** utilizam indivíduos que representam programas de computador.

# Perguntas e Discussão

---

1. Cite aspectos que distinguem os Algoritmos Genéticos dos algoritmos tradicionais de busca e otimização? Discuta a implicação prática de o AG usar **transições probabilísticas** em vez de regras determinísticas e operar sobre uma **codificação** das soluções.
2. O processo de **seleção** é uma parte chave do algoritmo, buscando indivíduos mais bem adaptados para se reproduzir. Explique como o método da "**roda da roleta**" atinge o equilíbrio crucial entre dar preferência a indivíduos com melhor fitness (promovendo a exploração de boas áreas) e ainda assim permitir que indivíduos menos aptos tenham uma chance (mantendo a diversidade). Qual é o risco de ter uma **pressão seletiva muito alta**?
3. Explique a função primária do operador de **Crossover** e do operador de **Mutação**. Qual é o principal risco se a taxa de mutação for configurada para ser muito alta?
4. A **função-objetivo** (**fitness**) é o objeto da otimização e a única informação de que o AG necessita. Qual é a grande vantagem dos Algoritmos Genéticos ao lidar com a função-objetivo como uma "caixa preta"?