



# Computação Natural

## UNIDADE 03

### Compreendendo algoritmos genéticos, algoritmos baseados na evolução e seleção natural

Na última semana, exploramos os conceitos básicos de seleção, mutação e cruzamento em algoritmos genéticos. Contudo, é importante notar que existem várias abordagens distintas para implementar esses elementos, cada uma afetando de maneira única o desempenho do algoritmo. Nesta semana, abordaremos técnicas adicionais, como o elitismo, que são cruciais para preservar as melhores soluções ao longo das diversas gerações. Portanto, forneceremos uma visão geral dos métodos mais utilizados, ampliando nossa compreensão dessas estratégias essenciais.

## Seleção

---

Como discutido anteriormente, o processo de seleção em algoritmos genéticos é fundamentalmente baseado em probabilidades, com indivíduos de maior aptidão tendo uma probabilidade aumentada de ser escolhidos para a próxima geração. No entanto, surge a questão: como exatamente é determinada a probabilidade de seleção para cada indivíduo? Para abordar isso, vamos considerar duas técnicas de seleção predominantes: a **seleção por roleta** e a **seleção por torneio**.

A **seleção por roleta** adota um método proporcional ao *fitness* de cada indivíduo. Aqui, a probabilidade de um indivíduo ser selecionado é diretamente proporcional ao seu valor de *fitness*. Essa técnica pode ser visualizada metaforicamente como uma roleta em um cassino ou parque de diversões, em que o espaço ocupado por cada indivíduo na roleta é equivalente à sua aptidão. Indivíduos com maior *fitness* ocuparão uma porção maior da roleta, aumentando assim suas chances de seleção (Wirsansky, 2020).

Esse método reflete o princípio de que indivíduos mais aptos devem ter maiores chances de passar seus genes para a próxima geração, um conceito central na evolução natural. Ao mesmo tempo, a **seleção por roleta** mantém a diversidade genética ao permitir que até mesmo indivíduos com menor *fitness* tenham uma chance, ainda que pequena, de serem escolhidos.

Indivíduo	<i>Fitness</i>	Proporção relativa
A	8	7%
B	12	11%
C	27	24%
D	4	3%
E	45	40%
F	17	15%

Tabela 1: Exemplo de indivíduos com seus valores de *fitness* e proporções relativas. Fonte: Adaptado de Wirsansky (2020).

O capítulo do livro de Wirsansky apresenta um exemplo ilustrativo para explicar o conceito da **seleção por roleta** em algoritmos genéticos. Nesse exemplo, a tabela 1 exibe seis indivíduos identificados pelas letras A, B, C, D, E e F, com seus respectivos valores de *fitness*. Uma coluna importante na tabela é a da **proporção relativa**, que calcula a participação de cada indivíduo no *fitness* total, ou seja, a proporção do valor de *fitness* de cada um em relação à soma do *fitness* de todos os indivíduos.

Acompanhando a tabela, temos a figura 1, que mostra uma representação gráfica de uma roleta. Essa roleta é dividida em segmentos que representam cada indivíduo, com o tamanho do segmento proporcional ao *fitness* relativo do indivíduo. Um ponto de seleção na roleta é usado para determinar qual indivíduo será escolhido para a próxima geração. Ao girar a roleta, os indivíduos com maior *fitness*, ocupando segmentos maiores, têm uma probabilidade aumentada de serem selecionados, refletindo suas proporções relativas na tabela.

Esse processo de girar a roleta é repetido várias vezes em cada geração, até que se acumule um número suficiente de indivíduos para formar a próxima geração. O exemplo de Wirsansky é fundamental para entender como a seleção por roleta funciona na prática, demonstrando visualmente o impacto do *fitness* na probabilidade de seleção de um indivíduo.

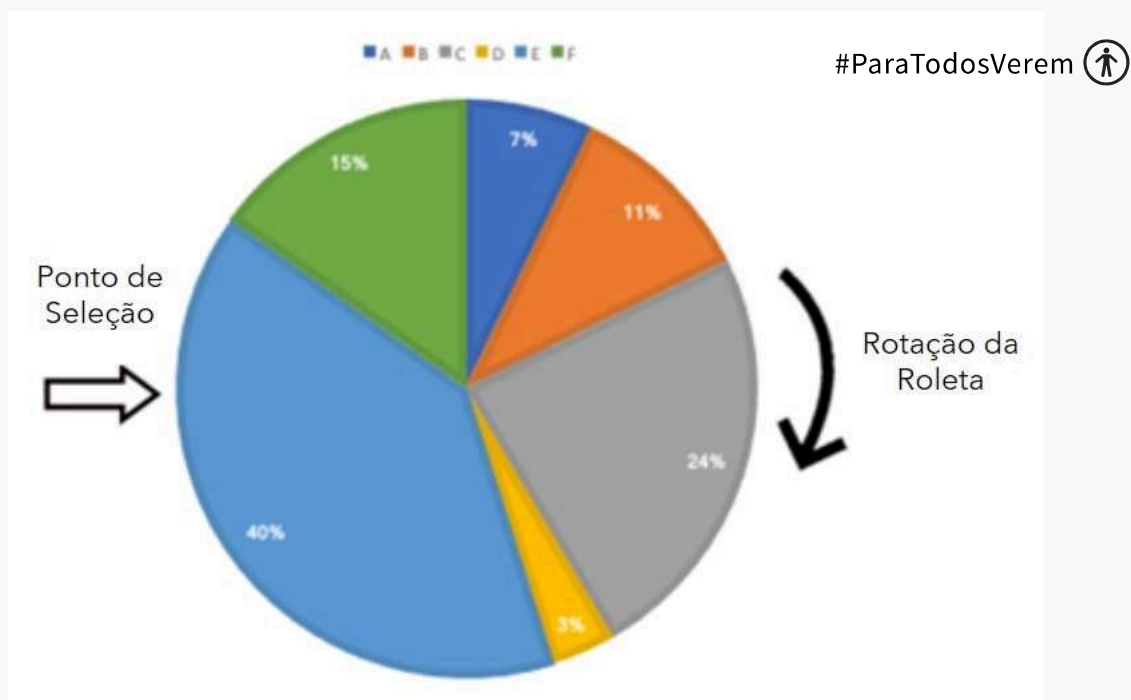


Figura 1: Exemplo da seleção por roleta. Fonte: Adaptado de Wirsansky (2020).

A **seleção por roleta**, método utilizado nos algoritmos genéticos, possui uma característica marcante: a possibilidade de um mesmo indivíduo ser selecionado múltiplas vezes para a próxima geração devido à proporção de seu segmento na roleta. Isso pode levar a uma concentração de soluções semelhantes e reduzir a diversidade genética da população. Para mitigar esse efeito, variantes como a **amostragem universal estocástica** são empregadas. Nesse método alternativo, em vez de girar a roleta repetidamente, múltiplos pontos de seleção são estabelecidos. Assim, quando a roleta é girada uma única vez, vários indivíduos são escolhidos simultaneamente, um para cada ponto de seleção. Isso permite uma distribuição mais equilibrada das chances de seleção entre os indivíduos, como descrito por Wirsansky.

Além da **seleção por roleta**, um método amplamente discutido e utilizado na prática de algoritmos genéticos, existe outra técnica de seleção que merece destaque: a **seleção por torneio**. Esse método oferece uma abordagem alternativa e complementar à seleção por roleta, trazendo uma dinâmica competitiva que simula um torneio entre os indivíduos da população. Enquanto a seleção por roleta se baseia em probabilidades proporcionais ao *fitness*, a seleção por torneio introduz um elemento de disputa direta, no qual os indivíduos competem entre si em *rounds* para determinar quem avança para a próxima geração com base na superioridade de seu *fitness*.



Figura 2: Exemplo da seleção por torneio. Fonte: Adaptado de Wirsansky (2020).

A **seleção por torneio** funciona como uma série de competições, ou *rounds*, em que a cada *round* um subconjunto de indivíduos é escolhido aleatoriamente da população total. Dentre estes, o indivíduo com o maior valor de *fitness* é selecionado para prosseguir. Esse processo é bem exemplificado na Figura 2 mencionada por Wirsansky, na qual os indivíduos A, B e F são escolhidos para um *round* específico, e o indivíduo F é selecionado por ter o maior *fitness* entre os participantes.

Um aspecto crucial da **seleção por torneio** é o tamanho do torneio, isto é, o número de indivíduos que competem em cada *round*. Um torneio maior aumenta as chances de que os indivíduos mais aptos sejam escolhidos, uma vez que há mais competidores e, portanto, uma maior probabilidade de que os mais aptos estejam presentes. Por outro lado, isso também reduz a probabilidade de indivíduos com menor *fitness* serem selecionados. Esse método é particularmente útil para manter a diversidade genética dentro da população, ao mesmo tempo em que favorece os indivíduos mais fortes, conforme explicado por Wirsansky.

## Cruzamento

---

Dentro do amplo espectro de técnicas de **cruzamento** em algoritmos genéticos, que são fundamentais para a recombinação genética e a geração de novas soluções, destacam-se métodos como o **cruzamento por ponto único**, o **cruzamento por dois pontos** e o **cruzamento uniforme**.

O **cruzamento por ponto único** é uma técnica que envolve a seleção de um único ponto ao longo do cromossomo dos indivíduos pais. Esse ponto de cruzamento é escolhido de maneira aleatória e será o mesmo para ambos os pais. A partir desse ponto, os genes situados à direita são trocados entre os dois cromossomos pais, resultando na criação de dois descendentes que possuem uma mistura dos genes de ambos os pais. Este método é eficaz na criação de variabilidade genética mantendo segmentos de genes dos pais, e é visualizado na Figura 3 (Wirsandy, 2020).

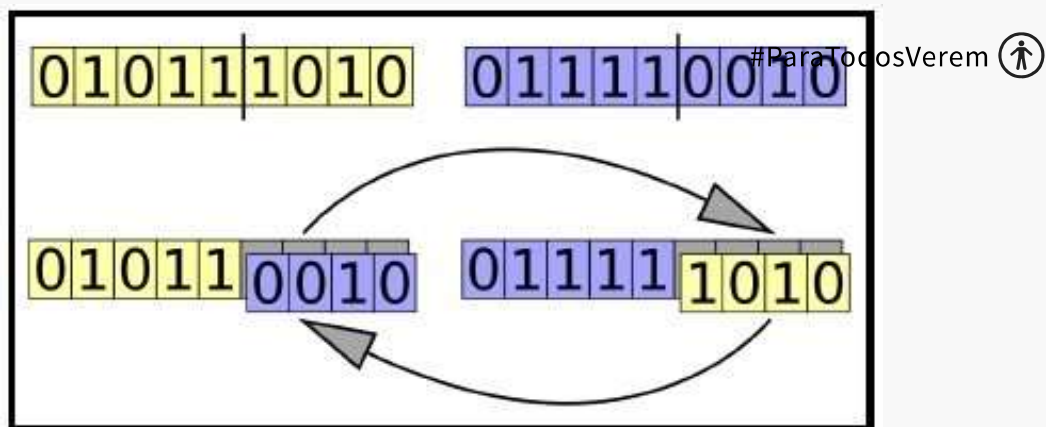


Figura 3: Exemplo de mutação por ponto único. Fonte: Adaptado de <https://commons.wikimedia.org/wiki/File:Computational.science.Genetic.algorithm.Crossover.One.Point.svg>.

O **cruzamento por dois pontos** é uma extensão do método de **cruzamento por ponto único** em algoritmos genéticos, trazendo uma complexidade adicional ao processo de recombinação genética. Nesta técnica, dois pontos distintos são selecionados aleatoriamente ao longo dos cromossomos dos pais. Os genes localizados entre esses dois pontos são então trocados entre os cromossomos pais. Essa troca resulta na formação de dois novos indivíduos, ou filhos, que herdam combinações de genes de ambos os pais. A figura 4 mencionada ilustra esse processo, demonstrando como os segmentos genéticos são reorganizados para criar diversidade genética na descendência (Wirsandy, 2020).

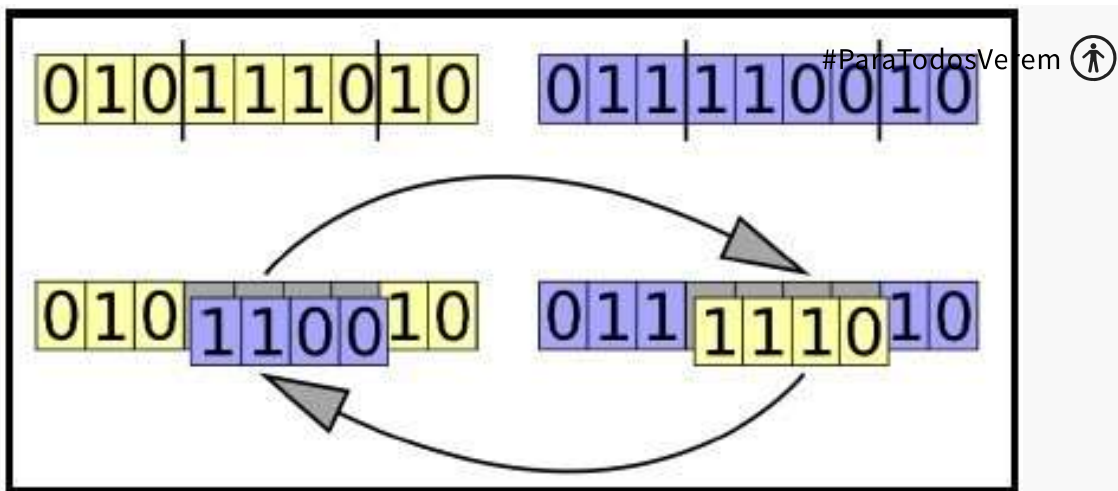


Figura 4: Exemplo de mutação por dois pontos. Fonte: Adaptado de <https://commons.wikimedia.org/wiki/File:Computational.science.Genetic.algorithm.Crossover.Two.Point.svg>.

O **cruzamento uniforme** introduz uma abordagem ainda mais granular ao processo de recombinação genética em algoritmos genéticos. Nesse método, cada gene no cromossomo dos filhos é considerado de forma independente, com uma chance igual de ser herdado do pai ou da mãe. Especificamente, quando a probabilidade é definida como 50%, cada pai tem exatamente a mesma probabilidade de passar seu gene específico para o filho. Conforme descrito por Wirsandy e ilustrado na figura 5, isso pode resultar em filhos cujos genes são uma mistura aleatória dos pais. Em um exemplo prático, um segundo filho pode receber exatamente o oposto da combinação de genes do primeiro filho, refletindo uma espécie de complemento das escolhas genéticas, garantindo que todas as variações parentais possíveis sejam consideradas.

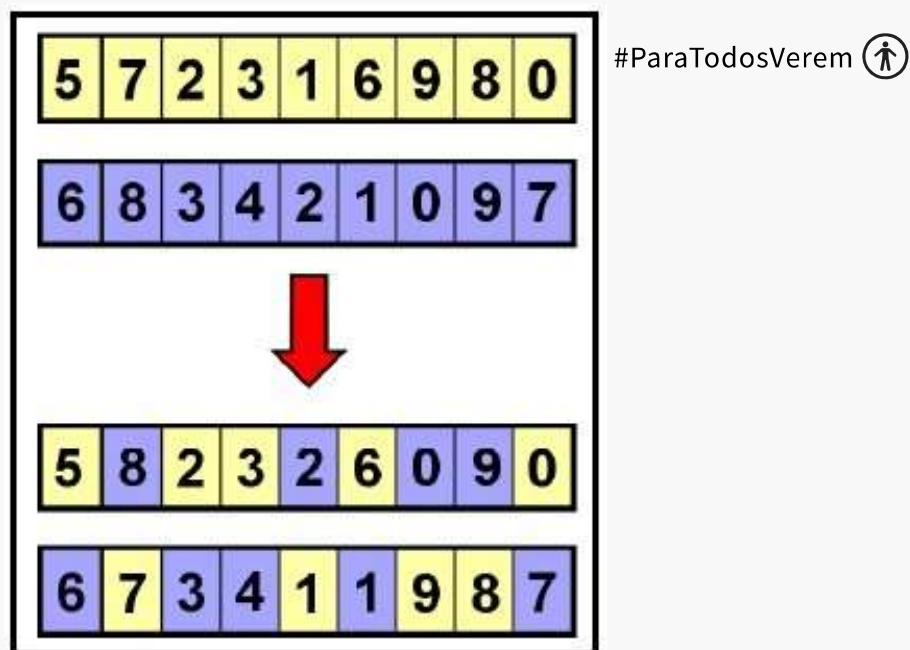


Figura 5: Exemplo de mutação uniforme. Fonte: Adaptado de Wirsansky (2020).



A **mutação** em algoritmos genéticos é um evento aleatório e geralmente ocorre com uma probabilidade baixa, garantindo uma diversidade genética sem comprometer a estabilidade da evolução das soluções. Uma taxa de mutação excessivamente alta pode, no entanto, prejudicar a *performance* do algoritmo, levando a uma exploração excessiva que pode destruir boas soluções em vez de refinar as existentes, conforme apontado por Wirsansky (2020).

Na prática de algoritmos genéticos, encontramos uma variedade de métodos de mutação que contribuem para a diversidade genética da população. Entre eles, destacam-se a **mutação da troca de *bit***, a **mutação de troca** e a **mutação de inversão**. Cada um oferece uma abordagem diferente para alterar os cromossomos dos indivíduos, garantindo que novas características possam emergir e serem exploradas no processo de busca de soluções ótimas.

Entre as técnicas de mutação, a **mutação da troca de *bit*** é bastante direta: em um cromossomo binário, um *bit* é selecionado ao acaso e o seu valor é invertido — se era 1, torna-se 0, e se era 0, torna-se 1. Essa técnica é exemplificada na figura 6 e é uma forma eficaz de introduzir variações pontuais sem alterar a estrutura geral do cromossomo.

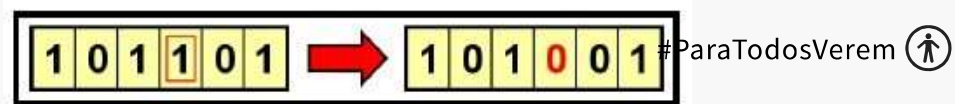


Figura 6: Exemplo de mutação da troca de *bit*. Fonte: Adaptado de Wirsansky (2020).

Na **mutação de troca**, uma técnica utilizada em algoritmos genéticos, dois genes são escolhidos aleatoriamente no cromossomo, e suas posições são trocadas. Esse método introduz variação ao rearranjar o material genético existente sem alterar a frequência dos genes. Por exemplo, conforme descrito por Wirsansky, se os genes nas posições 3 e 6 forem selecionados na figura 7, o gene que estava na posição 3 será trocado com o que estava na posição 6. Este tipo de mutação é útil para explorar novas regiões do espaço de busca, mantendo a diversidade genética da população.

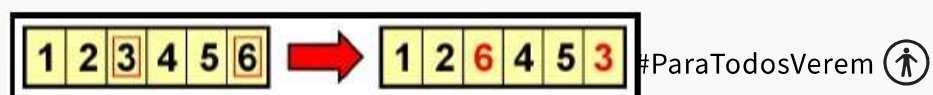


Figura 7: Exemplo de mutação de troca. Fonte: Adaptado de Wirsansky (2020).

Na **mutação de inversão**, um mecanismo usado em algoritmos genéticos, uma sequência específica de genes dentro do cromossomo é escolhida e a ordem dos genes nessa sequência é invertida. Por exemplo, como ilustrado na figura 8, se os quatro últimos genes de um cromossomo são selecionados para a inversão, suas posições são rearranjadas, de modo que o último gene passa a ser o primeiro da sequência selecionada, o penúltimo se torna o segundo, e assim por diante. Esse processo resulta em uma nova configuração genética que pode introduzir variações significativas na população, ajudando na exploração do espaço de soluções do algoritmo.



Figura 8: Exemplo de mutação de inversão. Fonte: Adaptado de Wirsansky (2020).

## Elitismo

---

Os processos de seleção, cruzamento e mutação são cruciais na evolução de indivíduos em algoritmos genéticos, mas esses métodos podem acidentalmente eliminar indivíduos de alto desempenho, ou seja, aqueles com maiores valores de *fitness*. Para mitigar esse risco, uma técnica conhecida como **elitismo** é empregada. Conforme Wirsansky descreve, o elitismo envolve a duplicação dos melhores *n* indivíduos para a próxima geração, com *n* sendo um parâmetro predefinido. Esses indivíduos de elite não só são preservados, mas também podem participar do processo de seleção, atuando como progenitores de novos indivíduos. Essa abordagem garante que soluções valiosas, já descobertas em gerações anteriores, não sejam perdidas, evitando assim o desperdício de recursos na redescoberta dessas soluções.

Nesta semana, temos uma ótima oportunidade de aprender na prática com uma [videoaula](#) focada em algoritmos genéticos, utilizando *python*. Essa será uma chance valiosa para aplicar os conceitos teóricos que estudamos, experimentando com bibliotecas específicas do *python*. Essa abordagem prática e direta vai ajudar a consolidar seu entendimento sobre como esses algoritmos funcionam e como podem ser efetivamente implementados.



**IMPORTANTE**  
**IMPORTANTE**



Olá, estudante, todos os códigos que vamos ver nas videoaulas estão disponíveis no github (<https://github.com/yohangumiel/Aulas-PUC-PR/tree/main/Computacao-Natural>). Para esta aula, utilize este código.

## | Conclusão da Unidade e Referências

À medida que concluímos esta unidade, é evidente o nosso progresso na compreensão dos algoritmos genéticos, abordando conceitos avançados de seleção, cruzamento, mutação e a importância do elitismo. Esse conhecimento nos equipa com uma poderosa ferramenta para resolver uma ampla gama de problemas de otimização e adaptação. Conforme seguimos em frente, podemos confiar em nossa capacidade de aplicar esses princípios em desafios do mundo real, capacitando-nos a encontrar soluções criativas e eficazes. Esta unidade é apenas o começo de uma jornada contínua de aprendizado e aplicação de algoritmos genéticos em nosso campo de estudo.

WIRSANSKY, E. **Hands-on genetic algorithms with python**: applying genetic algorithms to solve real-world deep learning and artificial intelligence problems. Birmingham: Packt Publishing Ltd, 2020.



© PUCPR - Todos os direitos reservados.