

### 3. ALGORITMOS GENÉTICOS

#### PARTE 1

- 3.1 [Analogia Física: a evolução das espécies](#)
- 3.2 [A Tradução Matemática: o algoritmo básico](#)
- 3.3 [Codificação dos Indivíduos](#)
- 3.4 [Avaliação da Aptidão de um Indivíduo](#)
- 3.5 [Operador Cruzamento](#)
- 3.6 [Operador Mutação](#)

### 3. ALGORITMOS GENÉTICOS

#### 3.1. Analogia Física: a evolução das espécies

Até meados do século 19, os naturalistas acreditavam que cada espécie havia sido criada separadamente por um ser supremo ou através de geração espontânea. O trabalho do naturalista Carolus Linnaeus sobre a classificação biológica de organismos despertou o interesse pela similaridade entre certas espécies, levando a acreditar na existência de uma certa relação entre elas. Outros trabalhos influenciaram os naturalistas em direção à teoria da seleção natural, tais como os de Jean Baptiste Lamarck, que sugeriu uma teoria evolucionária no "uso e desuso" de órgãos; e de Thomas Robert Malthus, que propôs que fatores ambientais tais como doenças e carência de alimentos, limitavam o crescimento de uma população.

Depois de mais de 20 anos de observações e experimentos, Charles Darwin apresentou em 1858 sua teoria de evolução através de seleção natural, simultaneamente com outro naturalista inglês Alfred Russel Wallace. No ano seguinte, Darwin publica o seu "On the Origin of Species by Means of Natural Selection" com a sua teoria completa, sustentada por muitas evidências colhidas durante suas viagens a bordo do Beagle.

Este trabalho influenciou muito o futuro não apenas da Biologia, Botânica e Zoologia, mas também teve grande influência sobre o pensamento religioso, filosófico, político e econômico da época. A teoria da evolução e a computação nasceram praticamente na mesma época: Charles Babbage, um dos fundadores da computação moderna e amigo pessoal de Darwin desenvolveu sua máquina analítica em 1833. Ambos provavelmente estariam surpresos e orgulhosos com a ligação entre estas duas áreas.

Por volta de 1900, o trabalho de Gregor Mendel, desenvolvido em 1865, sobre os princípios básicos de herança genética, foi redescoberto pelos cientistas e teve grande influência sobre os futuros trabalhos relacionados à evolução. A moderna teoria da evolução combina a genética e as idéias de Darwin e Wallace sobre a seleção natural, criando o princípio básico de Genética Populacional: a variabilidade entre indivíduos em uma população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética. Este princípio foi desenvolvido durante os anos 30 e 40, por biólogos e matemáticos de importantes centros de pesquisa.

Nos anos 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi John Holland quem começou, seriamente, a desenvolver as primeiras pesquisas no tema. A idéia inicial de Holland foi tentar imitar algumas etapas do processo de evolução natural das espécies incorporando-as a um algoritmo computacional. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro "Adaptation in Natural and Artificial Systems", hoje considerado a Bíblia de Algoritmos Genéticos. Desde então, estes algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização e aprendizado de máquina. (<http://www.iagenetico.hpg.ig.com.br/introd.html> )

[voltar para ALGORITMOS GENÉTICOS](#)

### 3.2. A Tradução Matemática: o algoritmo básico

Um “modelo simplificado” do processo de seleção natural poderia ser expresso por:

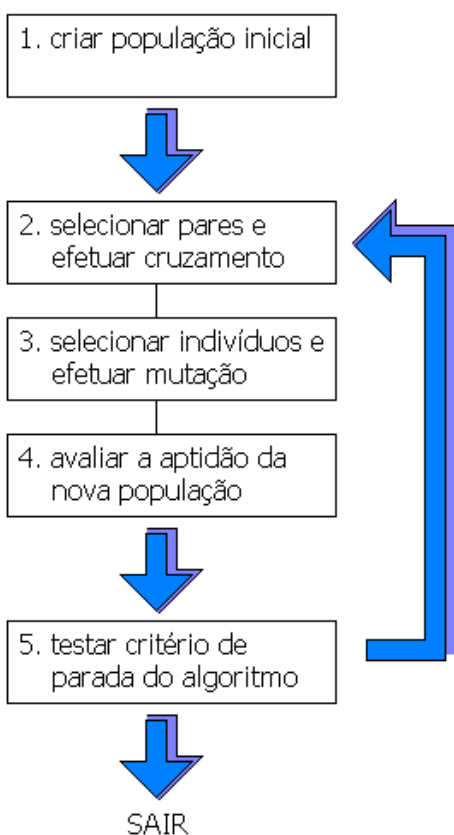
- (i) as características (boas ou ruins) dos indivíduos estão registradas nos seus genes, de forma codificada;
- (ii) cada geração de indivíduos em uma população é formada pela combinação do material genético dos indivíduos da geração anterior;
- (iii) ocasionalmente, ocorrem mutações aleatórias no material genético de um indivíduo, que podem torná-lo mais ou menos apto a sobreviver em seu meio;
- (iv) os indivíduos mais aptos têm maior chance de sobreviverem e conseguirem passar seus genes para a próxima geração, ou seja, têm maior chance de se reproduzirem;
- (v) qualquer indivíduo - mais ou menos apto - tem a mesma probabilidade de sofrer mutação.

Com inspiração nestas premissas simples, é possível formular algoritmos para efetuar a otimização de uma função objetivo arbitrária, os chamados Algoritmos Genéticos (AGs).

Na analogia matemática, a aptidão está relacionada ao valor da função objetivo. Cada indivíduo corresponde a uma possível solução do problema de otimização e cada geração é uma iteração do algoritmo. Os AGs operam sobre um conjunto de possíveis soluções do problema, e não

apenas uma (como métodos determinísticos ou o método de Recozimento Simulado, vistos nas aulas anteriores). Em cada iteração, operadores são aplicados aos indivíduos para simular os fenômenos de seleção natural, de geração de novos indivíduos pela reprodução sexuada e de mutação aleatória dos indivíduos.

A figura ao lado mostra as principais etapas do algoritmo genético básico. A cada iteração do algoritmo as soluções mais promissoras do problema (os indivíduos mais aptos) têm maior probabilidade de serem selecionadas para gerarem novas soluções tentativa (descendentes). Em seguida, alguns indivíduos são selecionados aleatoriamente, independente de sua aptidão, para sofrerem uma mudança arbitrária (mutação). O valor da função objetivo é então calculado para cada elemento deste novo conjunto de soluções tentativa. Os métodos de seleção são projetados para escolher preferencialmente indivíduos com maiores notas de aptidão, embora não exclusivamente, a fim de manter a diversidade da população.



Os passos 2 a 5 são continuamente repetidos até que um critério de parada seja satisfeito. Estes critérios são dependentes do problema, mas geralmente correspondem a ter sido encontrada

uma resposta suficientemente boa para o problema ou ter sido executado um certo número de iterações do algoritmo (gerações).

Assim como a natureza caminha no sentido de maximizar a aptidão dos indivíduos, os algoritmos genéticos caminham no sentido de otimizar o valor da função objetivo. Estes algoritmos, apesar de serem computacionalmente muito simples, são bastante poderosos. Não há provas matemáticas rigorosas da convergência do algoritmo, mas as aplicações do algoritmo vêm crescendo nos últimos anos, principalmente devido às vantagens que este apresenta quando comparado a outros métodos de otimização. Podem-se destacar as seguintes:

- i) O algoritmo não requer o cálculo de derivadas da função objetivo;
- ii) Descontinuidades na função objetivo não afetam o desempenho do algoritmo;
- iii) A presença de mínimos locais não reduz a eficiência do algoritmo;
- iv) O algoritmo se aplica a problemas em que a função objetivo não pode ser representada por uma função matemática.

Nas próximas seções, cada etapa dos Algoritmos Genéticos será detalhada.

[voltar para ALGORITMOS GENÉTICOS](#)

### 3.3. Codificação dos Indivíduos

Na natureza, todas as características do indivíduo, responsáveis por seu grau de aptidão, estão codificadas nos genes que constituem os cromossomos. Analogamente, os Algoritmos Genéticos otimizam o valor da função objetivo (o “fenótipo” da solução, a sua característica observável) atuando sobre a representação codificada dos genes dos indivíduos (o “genótipo” da solução). Assim, o primeiro passo ao se empregar um Algoritmo Genético é definir que codificação será empregada.

Há várias maneiras de representar um indivíduo de forma codificada. Tradicionalmente, os indivíduos são representados por vetores binários, isto é, cada elemento do vetor pode ser preenchido de 2 maneiras apenas: 1 ou 0. Se há 2 maneiras de preencher um dígito, há 4 maneiras de preencher uma seqüência de 2 dígitos (00, 01, 10 e 11), 8 maneiras de preencher uma seqüência de 3 dígitos (000, 001, 010, 100, 011, 101, 110 e 111), etc... Generalizando, há  $2^n$  maneiras de preencher uma seqüência de  $n$  dígitos.

A grande vantagem desta representação é o fato de ela ser independente do problema. Uma vez encontrada a representação em vetores binários de um conjunto de indivíduos, os operadores genéticos padrão podem ser utilizados, o que facilita a utilização dos AGs em diferentes classes de problemas.

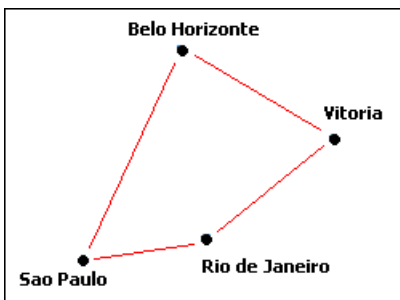
Como um exemplo, analisemos o problema do caixeiro viajante que precisa passar pelas 4 capitais da Região Sudeste do Brasil. Será necessário empregar 2 dígitos na representação de cada cidade, pois este é o número mínimo de dígitos capaz de representar 4 diferentes números. Podemos representar um determinado roteiro através da concatenação dos códigos associados às cidades vi-

sitadas. Por exemplo, usando a associação mostrada na tabela abaixo, podemos codificar as rotas mostradas nos esquemas em seguida como seqüências de 8 dígitos.

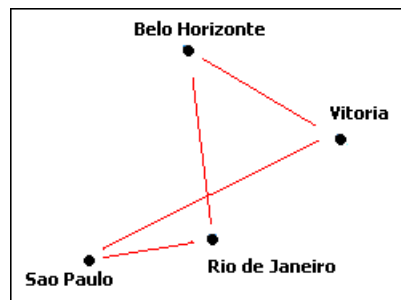
CIDADE	CÓDIGO
Rio de Janeiro	00
São Paulo	01
Belo Horizonte	10
Vitória	11



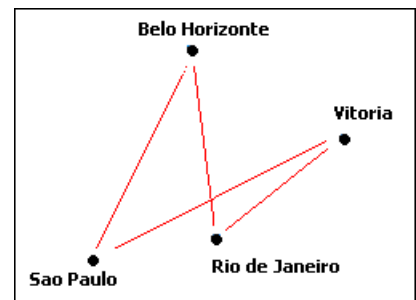
Obviamente não existe sentido em codificar um problema do caixeiro viajante com 4 cidades, pois só há três possíveis rotas, mostradas abaixo. Mas, para o problema com 15 cidades, já há a necessidade de um método de otimização para auxiliar na tarefa de encontrar a melhor rota. Neste problema, são necessários 4 dígitos para codificar as cidades, e cada rota será identificada por uma seqüência de  $15 \cdot 4 = 60$  dígitos.



codificação 01101100



codificação 01100011



codificação 01101100

Se uma variável real estiver sendo codificada, é necessário primeiro discretizar o domínio. Apenas como um exemplo imagine que uma das variáveis a otimizar em um problema seja uma fração molar. Esta variável encontra-se entre 0 e 1, e a depender da precisão requerida para representá-la, um número diferente de dígitos é necessário. Para representá-la com uma precisão de 0,1 já haverá 11 possíveis valores para esta variável, e serão necessários 4 dígitos para representá-la. Observe que serão “ocupadas” (associadas a um valor numérico da variável) apenas 11 das 16 possíveis seqüências binárias formadas com 4 dígitos. Isto significa que 5 seqüências de dígitos não estarão associadas a qualquer valor da variável  $x$ . Indivíduos que possuem uma destas codificações para representar a variável  $x$  são chamados de espúrios.

A codificação binária, apesar de ser a mais tradicional e conferir generalidade aos operadores, não é a mais adequada para qualquer tipo de problema. Apenas para ilustrar, veja que há apenas 3 rotas possíveis no PCV com 4 cidades, mas é possível codificar  $2^8 = 256$  rotas com 8 dígitos!

Outras formas de codificação – adaptadas a cada tipo de problema – serão apresentadas na próxima aula e durante os exemplos.

[voltar para ALGORITMOS GENÉTICOS](#)

### 3.4. Avaliação da Aptidão de um indivíduo

Avaliar um indivíduo num AG significa determinar o seu nível de aptidão de sobrevivência. Ou seja, num AG sobrevivem prioritariamente os indivíduos mais aptos. Em problemas de otimização, o critério de sobrevivência é determinado pelo valor da função objetivo.

A depender do problema, avaliar o valor da função objetivo pode envolver a avaliação de uma expressão algébrica, a resolução de um sistema algébrico não linear ou mesmo a integração de um conjunto de equações diferenciais. Qualquer que seja o procedimento numérico, nesta etapa é preciso decodificar a variável binária e avaliar a qualidade da solução representada.

No problema do caixeiro viajante, o objetivo é minimizar o trajeto percorrido para visitar todas as cidades. Cada indivíduo corresponde a uma possível rota, e o valor da aptidão de um indivíduo é a distância percorrida nesta rota.

Um outro detalhe importante para a implementação do método: é possível (e até bastante provável) que indivíduos espúrios sejam gerados ao longo da execução do algoritmo genético. A maneira mais tradicional de lidar com este problema é, no momento de avaliar a aptidão do indivíduo, detectar que ele não é válido e atribuir um valor muito ruim de aptidão. Assim, ele não conseguirá se reproduzir e estes genes “defeituosos” vão eventualmente ser eliminados da população pelo processo de seleção natural.

Uma das muitas alterações feitas no algoritmo básico para torná-lo competitivo frente a outras técnicas de otimização foi adaptar os operadores para que estes indivíduos espúrios não sejam gerados. Desta forma reduz-se o custo computacional do algoritmo, já que apenas soluções válidas são consideradas. Estas particularidades serão discutidas com mais detalhe na próxima aula, onde os Algoritmos Genéticos modificados serão discutidos.

[voltar para ALGORITMOS GENÉTICOS](#)

### 3.5. Operador Cruzamento

O operador cruzamento consiste em gerar um ou dois cromossomos filhos a partir das informações dos dois cromossomos pais. Suponha que, por algum mecanismo, tenham sido escolhidos os dois cromossomos pais P1 e P2, representados abaixo como cadeias de 32 bits:

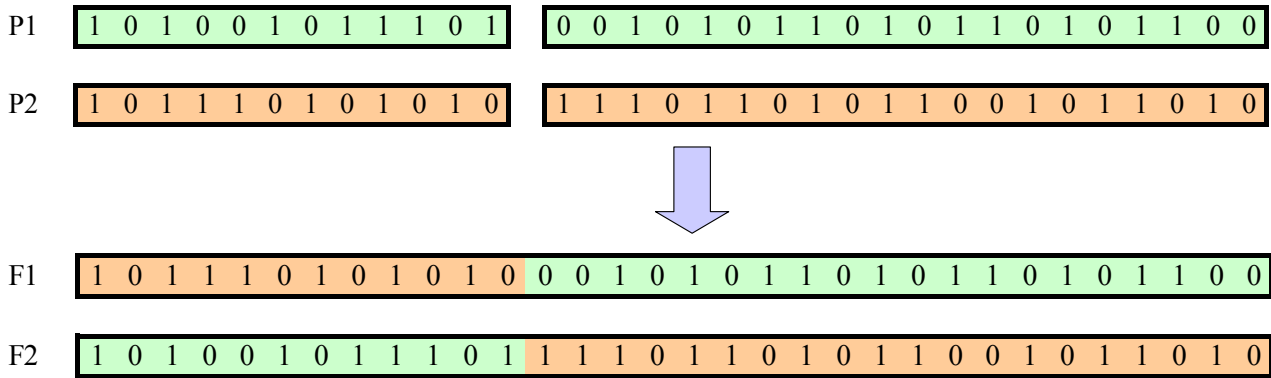
P1 

1	0	1	0	0	1	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

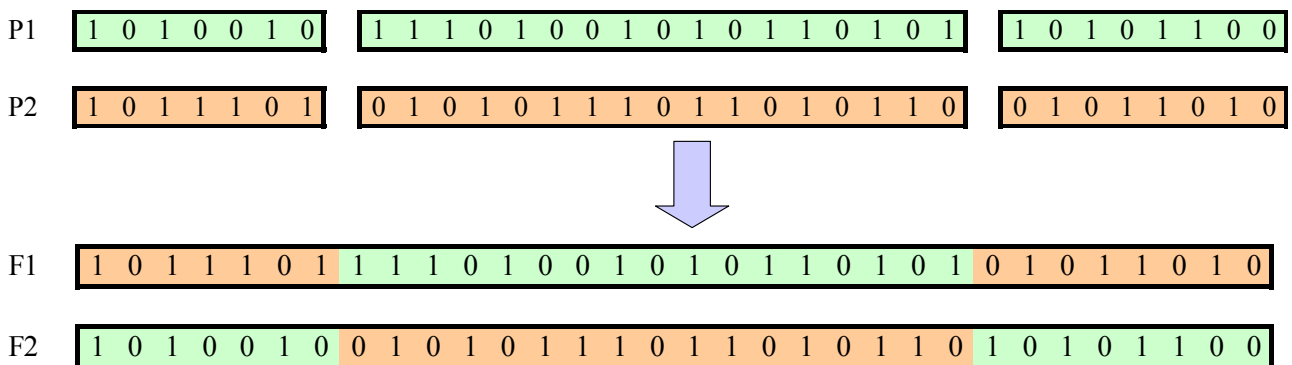
P2 

1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

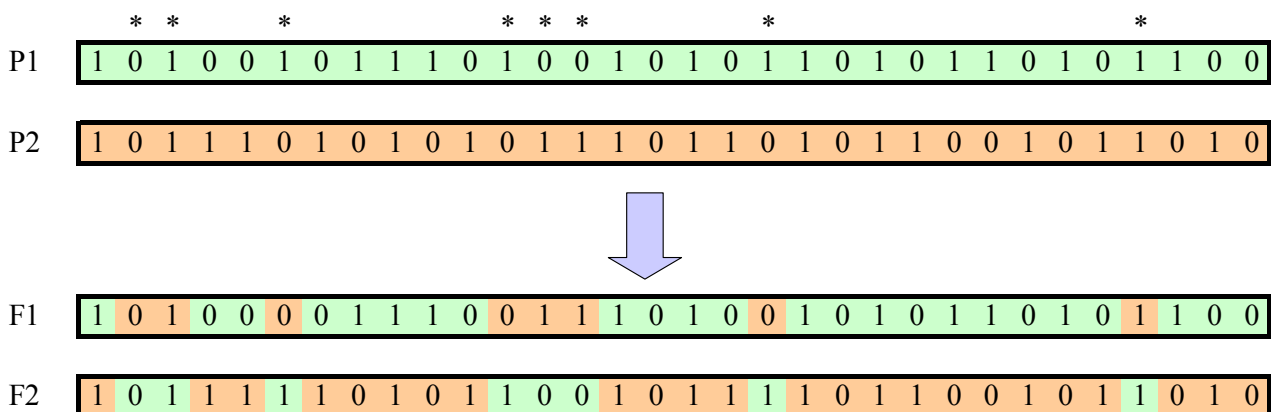
No operador mais clássico, chamado de **operador cruzamento simples**, inicialmente seleciona-se de forma aleatória uma posição, onde se efetua um corte entre dois genes adjacentes em cada cromossomo pai. No nosso exemplo, o corte foi feito entre o 12º e o 13º bit. É possível gerar até dois filhos (F1 e F2) pela concatenação dos cromossomos dos pais, mas não é incomum que apenas um destes filhos seja introduzido na nova população. O número de filhos por cruzamento (1 ou 2) é uma parâmetro a ser definido pelo usuário.



Há um número infinito de variações deste algoritmo básico. Talvez a mais simples seja o **cruzamento simples em múltiplos pontos**, onde não um mais vários pontos de corte são determinados aleatoriamente. Abaixo está representado o cruzamento com 2 pontos de corte (7º / 8º bit e o 24º / 25º bit).



Outra variação bastante freqüente é o chamado cruzamento uniforme, onde determina-se a percentagem de genes que vão ser trocados, procede-se o sorteio destes genes, e efetua-se a troca do material genético dos pais nestas posições.



No exemplo acima, um parâmetro do algoritmo seria 25% de substituição de genes no cruzamento. Para os dois pais selecionados, foram aleatoriamente sorteadas as posições de troca de material genético como 2, 3, 6, 12, 13, 14, 19 e 29. Observe que há 50% de chance de a troca de um gene não modificar o indivíduo (como por exemplo para os genes 2, 3 e 29).

Em uma população real, nem todos os “casais” têm filhos ao mesmo tempo, e há a coexistência de indivíduos mais novos e mais velhos, possivelmente trocando material genético. Da mesma forma, nos Algoritmos Genéticos, nem todos os indivíduos passam pelo operador cruzamento em todas as iterações, e muitas vezes os indivíduos (especialmente os bons) são mantidos na população por várias gerações.

O procedimento implementado em um Algoritmo Genético padrão é detalhado a seguir. Se há  $N_{ind}$  indivíduos na população, e deseja-se obter 1 filho por cruzamento, são selecionados por um “sorteio especial”  $N_{ind}$  pares de indivíduos. Este sorteio favorece os indivíduos mais aptos, mas não exclui a possibilidade de os menos aptos serem sorteados. Existe uma certa probabilidade  $p$  de o operador cruzamento ser aplicado, tipicamente da ordem de 70%. Para cada um dos pares formado, é sorteado um número aleatório  $r$  uniformemente distribuído no intervalo  $[0,1]$ . Se  $r \leq p$ , então o operador cruzamento é aplicado a este casal, gerando um indivíduo na nova população. Se  $r > p$  então não haverá um filho, e um dos pais deverá ser introduzido na nova população. Após a avaliação dos  $N_{ind}$  casais,  $N_{ind}$  indivíduos (novos ou antigos) terão sido introduzidos na nova geração de indivíduos.

[voltar para ALGORITMOS GENÉTICOS](#)

### 3.6. Operador Mutação

O operador mutação altera aleatoriamente um ou mais bits de um cromossomo. Este operador é utilizado para permitir uma diversificação no processo de busca e introduzir diversidade. Há pesquisadores que defendem a tese de que este é o verdadeiro operador genético, o responsável por toda a evolução.

Assim como o operador cruzamento, o operador mutação não é aplicado a todos os indivíduos. Existe uma probabilidade  $p$  (tipicamente da ordem de 5%) de o operador ser aplicado a um indivíduo. Para cada indivíduo da população, sorteia-se um número randômico  $r$  uniformemente distribuído no intervalo  $[0,1]$ . Se  $r \leq p$ , então o operador mutação é aplicado a este indivíduo, gerando um indivíduo diferente na nova população. Se  $r > p$  então não haverá mutação, e o indivíduo é deixado inalterado. Observe que todos os indivíduos estão igualmente sujeitos a sofrer mutação.

Para aplicar o operador mutação, normalmente sorteia-se aleatoriamente um bit, e este é alterado. No exemplo abaixo, o 11º bit do indivíduo foi sorteado, e conseqüentemente foi alterado de 1 para 0.

1 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 0

1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 0