3. ALGORITMOS GENÉTICOS

PARTE 2

3. ALGORITMOS GENÉTICOS

3.7. Aspectos de Implementação do Algoritmo

a) Algoritmos de Seleção de Indivíduos

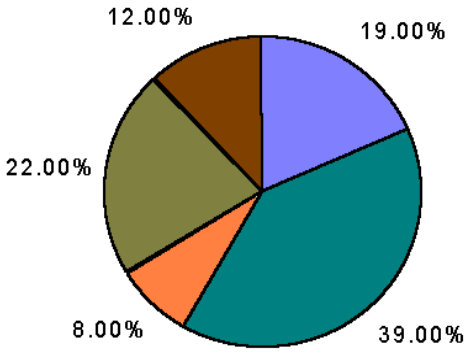
Para que o operador cruzamento seja aplicado, primeiro os "pais" (dois indivíduos da população) precisam ser selecionados. É importante que este procedimento de seleção leve em conta a aptidão dos indivíduos: indivíduos mais aptos devem ter maior probabilidade de serem selecionados, e assim se reproduzirem e passarem seus genes para a próxima geração.

Há vários procedimentos de seleção, sendo os mais usuais a Seleção por Roleta (roulette wheel selection) e a Seleção por Torneio (tournament selection).

No procedimento de **seleção por roleta**, cada indivíduo da população é representado na roleta proporcionalmente à sua aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes, e são escolhidos os indivíduos sorteados na roleta.

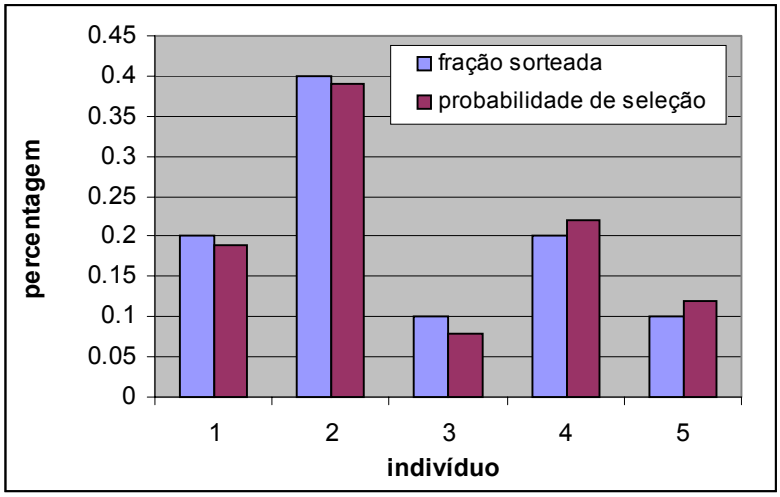
Matematicamente, a probabilidade de seleção de um indivíduo (visualizada como a "fração da roleta") é calculada como a razão entre a aptidão do indivíduo e o somatório de todas as aptidões individuais. Com base neste número, pode ser determinada a faixa de sorteio favorável àquele indivíduo. Na tabela abaixo, o procedimento de roleta é ilustrado para uma população com 5 indivíduos. Observe que a faixa favorável tem extensão determinada pela probabilidade de seleção do indivíduo, e qualquer número aleatório sorteado entre 0 e 1 pode ser associado a um indivíduo.

indivíduo	aptidão	probabilidade de seleção	faixa favorável
1	3.80	19.00%	0 – 0,19
2	7.80	39.00%	0,19 – 0,58
3	1.60	8.00%	0,58 – 0,66
4	4.40	22.00%	0,66 – 0,88
5	2.40	12.00%	0,88 – 1,00
Σ	**20.00**	**100.00%**	



Para exemplificar o procedimento, foram sorteados 10 indivíduos com o auxílio de um gerador de números randômicos uniforme. Os números gerados estão apresentados na tabela abaixo, e observa-se que o procedimento de roleta vai gerar distribuições bastante próximas às de probabilidades de seção dado que o número de indivíduos sorteados seja suficientemente grande e seja empregado um gerador de números randômicos confiável.

número gerado	indivíduo associado
.567	2
.736	4
.300	2
.107	1
.660	3
.231	2
.138	1
1.00	5
.366	2
.701	4



Já no procedimento de **seleção por torneio**, sorteiam-se dois indivíduos ao acaso, comparam-se suas aptidões e o mais apto destes dois é selecionado. Este procedimento é repetido para cada indivíduo a ser selecionado. Não é necessário calcular a probabilidade de seleção ou uma faixa favorável, e como conseqüência este procedimento é muito mais simples e rápido.

b) Seleção dos Parâmetros do Algoritmo

É importante analisar de que maneira alguns parâmetros influem no comportamento dos Algoritmos Genéticos, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis.

Tamanho da População. O tamanho da população afeta o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior. Tipicamente, empregam-se entre 20 e 200 indivíduos.

Número de gerações. O número de gerações está intimamente relacionado ao tamanho da população e ao tempo computacional disponível para a execução do algoritmo. O usuário deve escolher entre usar uma população pequena por muitas gerações ou uma população maior por menos gerações. Se o número de gerações for muito grande, deve ser avaliada a possibilidade de ser efetuada a reinicialização da população periodicamente. Não há um valor comumente aceito na literatura para este parâmetro.

Taxa de Cruzamento. Se esta taxa for muito alta, estruturas com boas aptidões poderão ser retiradas muito rapidamente. Se esta taxa for muito pequena, o algoritmo será muito lento. Tipicamente, a probabilidade de cruzamento é de cerca de 70% quando é gerado um filho por cruzamento.

Taxa de Mutação. Uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória. Tipicamente, a probabilidade de mutação é de cerca de 5%.

Outros parâmetros. Há outros parâmetros a definir no código, mas o seu impacto é menor que os destacados anteriormente. Como um exemplo, podemos citar o número de filhos por cruzamento e a porcentagem de genes trocados no cruzamento uniforme.

voltar para ALGORITMOS GENÉTICOS

3.8. Seleção da Codificação

Os algoritmos genéticos na sua forma original utilizam a representação binária para codificar os indivíduos. A codificação binária certamente confere generalidade ao algoritmo. Uma vez convertidos para o domínio binário, o problema do caixeiro viajante e um problema de otimização paramétrica são virtualmente idênticos. Esta representação, no entanto, não é a mais eficiente para diversos problemas de otimização, por vários motivos.

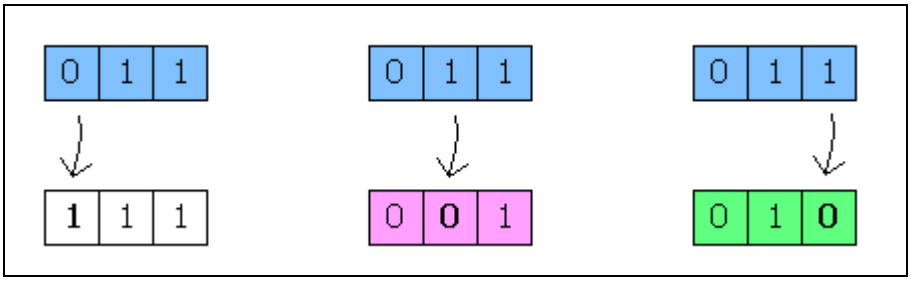
- **Podem ser criados indivíduos não viáveis ou espúrios.**

  Em muitas codificações, nem todos os indivíduos são viáveis. Este conceito fica melhor esclarecido através de um exemplo. Vamos representar através de codificação binária os 5 estados da variável s que representa a cor de um determinado produto. Observe que não há cor associada às codificações 101, 110 e 111.

s	000	001	010	011	100	101	110	111
cor	preto	rosa	verde	azul	laranja	x	x	x

  Há 3/8 de chance que este indivíduos espúrios sejam introduzidos na população inicial, aleatoriamente determinada. Mesmo se forem eliminados os indivíduos espúrios nesta instância, ainda assim estes indivíduos reapareceriam, durante a execução do algoritmo, como produto da aplicação dos operadores genéticos aos indivíduos válidos.

  Imagine que deva ser aplicado à codificação da cor azul (011) o operador mutação, definido como a alteração de um bit aleatório do indivíduo. Se o segundo ou terceiro bits forem alterados, serão obtidas as cores rosa e verde, respectivamente. No entanto, se o primeiro bit for alterado, será obtida um valor da variável ao qual não se associa qualquer cor.



Há basicamente três alternativas para contornar este problema: (i) adaptam-se os operadores genéticos para que sempre seja criado um indivíduo viável a partir de dois indivíduos viáveis; (ii) penaliza-se o indivíduo não viável no momento de calcular sua aptidão; (iii) altera-se o tipo de codificação (por exemplo, representando as 5 cores por números de 1 a 5).

- **A mudança de um bit pode levar a grandes modificações no indivíduo.**

  Quando se usa a codificação binária para representar números, reais ou naturais, não existe a garantia de que números com codificação parecida sejam parecidos. Por exemplo, observe a codificação de alguns números naturais entre 0 e 31. Apenas mudando um dígito (um bit) do indivíduo codificado, podemos obter desde um número parecido até um número significativamente diferente. Esta aparente "falta de coerência" atrapalha a convergência do algoritmo genético.

1111**1**	31	111**1**1	31	11**1**11	31	1**1**111	31	**1**1111	31
1111**0**	30	111**0**1	29	11**0**11	27	1**0**111	23	**0**1111	15

  Para contornar este problema, foi proposta por Gray uma forma de codificação binária onde números vizinhos na escala adotada (real, natural) possuem codificação bastante próxima. O preço a pagar é a maior complexidade no momento de codificação e decodificação do indivíduo. Esta codificação é pouco usada atualmente.

- **A discretização do espaço de busca real aumenta a dimensão do problema.**

  Para representar estas variáveis (reais) como números binários, é primeiro necessário discretizar o espaço de busca real e então criar uma relação entre estes valores e números binários.

  Queremos otimizar a temperatura de operação e a concentração de reagente na alimentação de um processo. A temperatura pode variar entre 50 e 150^{o}C, e a concentração de reagente na corrente de alimentação, entre 15 e 30 g/l. A variável a otimizar $\mathbf{x} = [\ T\ \ C\ ]^T$ é um vetor de 2 posições. O tamanho da variável codificada, no entanto será função do número de divisões adotado. Se forem usadas 3 divisões do intervalo, haverá 4 valores possíveis para cada variável, sendo necessários 2 bits para representar cada uma delas. Cada indivíduo seria representado por um vetor de 4 posições.

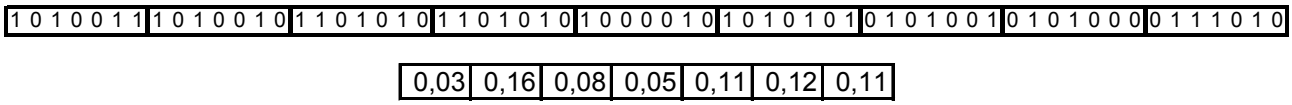
T (^{o}C)	codificação	C	codificação
50	00	15,0	00
66,6	01	20,0	01
83,3	10	25,0	10
100	11	30,0	11

  Provavelmente a precisão requerida seria maior do que esta. O tamanho do vetor codificado aumenta conforma a precisão aumenta, conforme detalhado na tabela abaixo.

número de intervalos:	1	3	7	15	63	255	1023	4095
tamanho do intervalo:								
em T	100	33,3	14,3	6,7	1,6	0,4	0,1	0,02
em C	15	5	2,1	1,0	0,2	0,06	0,02	0,004
bits necessários	1	2	3	4	6	8	10	12
tamanho do indivíduo	2	4	6	8	12	16	20	24

A utilização de representações em níveis de abstração mais altos tem sido investigada. Em particular, a codificação empregando números reais vêm ganhando muito força em problemas de otimização paramétrica, os mais comuns em engenharia.

Como um exemplo, imagine que está sendo resolvido um problema em que as variáveis são as frações molares dos 10 compostos em uma mistura gasosa. Apenas 9 precisam ser parte do nosso problema, e o outro pode ser calculado através da restrição de soma de frações molares. Se fosse necessária uma precisão de 0,01 , seriam necessários 7 dígitos para cada variável, ou seja, um vetor de 63 dígitos. Empregando codificação real, um vetor de 7 posições de variáveis reais seria empregado, e nenhuma consideração *a priori* sobre a precisão do método seria necessária. Compare na figura abaixo as duas representações do mesmo indivíduo. A codificação real representa uma enorme simplificação na representação de vetores reais, e é empregada rotineiramente nos problemas de otimização paramétrica na engenharia.



Um outro exemplo é o Problema do Caixeiro Viajante com *N* cidades. Empregando codificação binária, são necessários *m* dígitos para codificar cada cidade, sendo *m* o menor número inteiro tal que $m > log_2 N$. Logo, o vetor binário que representa cada rota terá tamanho *m.N*. Além disto, vimos que praticamente todos os indivíduos gerados pela aplicação dos operadores clássicos do tamanho serão espúrios, ou não válidos. Os trabalhos reportados na literatura que empregam algoritmos genéticos para resolver o PCV empregam o que alguns autores chamaram de *codificação inteira*, em que as cidades são numeradas de 1 a N. Cada rota é representada por um vetor inteiro de dimensão *N* em que os códigos das cidades são ordenados segundo a ordem de visita às cidades.

Para trabalhar com estas codificações diferenciadas, precisam ser criados operadores específicos. Estes algoritmos particularizados para o problema de interesse costumas ser muito mais eficientes que um AG genérico, já que através da codificação e dos operadores são incorporadas informações sobre o espaço de soluções.

voltar para ALGORITMOS GENÉTICOS

3.9. Operadores Genéticos Modificados

Para trabalhar com estas codificações diferenciadas, precisam ser criados operadores específicos. Estes algoritmos particularizados para o problema de interesse costumam ser muito mais eficientes que um AG genérico, já que através da codificação e dos operadores são incorporadas informações sobre o espaço de soluções. Cada codificação impõe uma forma diferente de adaptação dos operadores.

Operador Cruzamento

O operador cruzamento deve combinar a informação de dois indivíduos e gerar um ou dois outros. Algumas alternativas de cruzamento quando se utiliza codificação real estão exemplificadas na figura abaixo.

- Podemos seguir as mesmas idéias dos operadores clássicos (binários) e copiar parte da informação de cada pai, empregando um ou mais pontos de corte - respectivamente, (a) e (b) na figura abaixo. Estes pontos de corte são sorteados aleatoriamente para cada par de soluções.

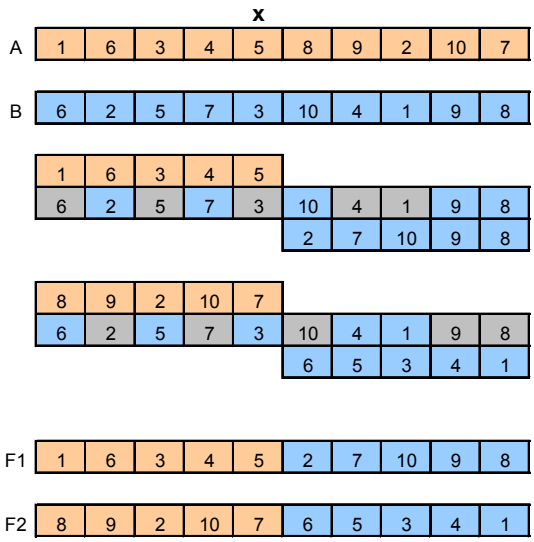
- Outra alternativa bastante comum é fazer a média aritmética de cada posição dos dois vetores, tal que $F1_i = (A_i + B_i)/2$. Este operador atua deterministicamente sobre o par de soluções aleatoriamente escolhido, e apenas um filho por cruzamento é gerado, como se observa no esquema (c).

- Podemos também fazer uma média ponderada dos dois vetores empregando um peso aleatório p, de forma que cada posição $F1_i = (1-p)\ A_i + p\ B_i$. O segundo filho pode ser gerado pela inversão da regra de cruzamento, ou seja, $F2_i = (1-p)\ B_i + p\ A_i$. Observe que o cruzamento por média aritmética corresponde ao sorteio ponderado com p=0,5. Na figura abaixo o esquema (d) mostra a aplicação deste operador com peso p=0,2.

pais		filhos							
		(a)		(b)		(c)	(d) p=0,2	
A	B	F1	F2	F1	F2	F1	F1	F2
0,53	0,84	0,53	0,84	0,53	0,84	0,69	0,78	0,59
0,04	0,31	0,04	0,31	0,31	0,04	0,18	0,26	0,09
0,85	0,19	0,19	0,85	0,19	0,85	0,52	0,32	0,72
0,04	0,03	0,03	0,04	0,03	0,04	0,04	0,03	0,04
0,68	0,26	0,26	0,68	0,68	0,26	0,47	0,34	0,60
0,69	0,78	0,78	0,69	0,69	0,78	0,73	0,76	0,71

Um problema do caixeiro viajante possui a peculiaridade de que não pode haver repetição dentro do vetor de inteiros. Se os operadores levarem isto em consideração, sempre gerarão indivíduos válidos. Um exemplo de operador cruzamento específico poderia ser o mostrado abaixo:

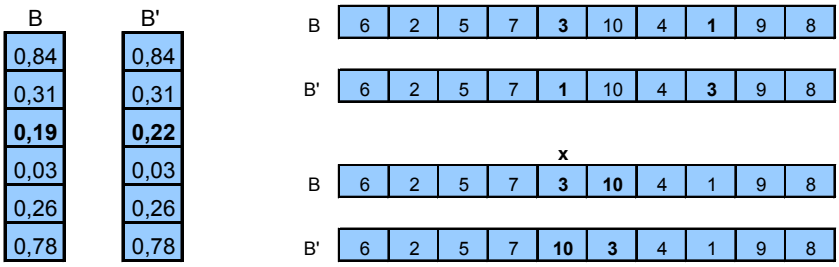
- sortear um ponto de corte aleatoriamente;
- dividir a rota A em dois pedaços;
- eliminar da solução B as cidades que estão no primeiro pedaço de A e formar o filho F1 pela concatenação deste pedaço de A com o que sobrou de B, mantendo a ordem das cidades.
- repetir o procedimento com o segundo pedaço de A para gerar o segundo filho F2, se desejado.



Na figura ao lado, foi ilustrado o procedimento para um PCV com 10 cidades. Foi sorteada a quinta cidade como ponto de corte.

Operador Mutação

O operador deve introduzir mudanças aleatórias em uma parte específica da representação do indivíduo. Podem ser mantidas as idéias dos operadores clássicos, apenas alterando-se sua implementação. Um possível operador mutação para codificação real consistiria em sortear uma posição do vetor e alterá-la (sortear qualquer ponto no domínio desta variável ou alterá-la em um percentual aleatoriamente determinado). Para o PCV, poderíamos simplesmente trocar a ordem de visita a duas cidades, contíguas ou não na rota associada àquekle indivíduo. Estas idéias estão aplicadas nos esquemas abaixo.



Apenas alguns exemplos ilustrativos de operadores genéticos adaptados foram apresentados. Há muitas variantes destes operadores ou mesmo operadores fundamentalmente diferentes sendo citados na literatura. Como leitura complementar sobre a incorporação de estruturas de dados ‘a codificação e aos operadores, sugere-se o fascinante livro de Michalewicz (1996), “Genetic Algorithms + Data Structures = Evolution Programs: Third Edition”.

voltar para ALGORITMOS GENÉTICOS

3.10. Outros operadores: elitismo, reinicialização e *niching*

Vários operadores diferentes foram sendo acrescentados aos Algoritmos Genéticos com o passar dos anos e o acúmulo da experiência na área. Alguns, por serem particularmente interessantes e de caráter geral, merecem destaque.

O operador ***elitismo*** consiste em replicar o melhor indivíduo de uma geração, inalterado, na geração subsequente. Com isto, espera-se que a melhor solução encontrada até o momento não se perca devido ao caráter estocástico do método. Quem diria, os Algoritmos Genéticos já vêm fazendo clonagem há muitos anos!

A ***reinicialização periódica*** da população, para muitos autores, é considerada como um operador que evita a estagnação dos AGs (fenômeno demonstrado chamado de *genetic drift*). Após um certo número de gerações, a população converge para uma região do espaço e o valor da função objetivo é pouco melhorado ao longo de várias gerações. Eventualmente, o operador mutação levará um dos indivíduos para uma região melhor do espaço de soluções, e este indivíduo mais evoluído irá disseminar seus genes na população. No entanto, esta evolução poderia demorar um número muito grande de gerações, tornando o custo computacional do algoritmo proibitivo. Uma medida para acelerar a convergência do código é a ***reinicialização*** da população. Os melhores indivíduos da população são armazenados, a população é aleatoriamente definida (como na partida do algoritmo) e em seguida estes melhores indivíduos são inseridos na população. Com isto, espera-se acelerar a convergência do algoritmo.

Uma outra estratégia interessante é a estratégia de ***niching***. Pode haver uma região indesejada do espaço de busca, por corresponder a um mínimo local, por representar uma região de soluções inviáveis, ou por outro motivo qualquer. Para fazer com que esta região seja evitada pelos indivíduos, costuma-se acrescentar, ao valor da função objetivo, uma penalidade proporcional à proximidade desta região. Assim, os indivíduos nesta região terão menos probabilidade de se reproduzirem e a população se afastará desta região.

Há muitos operadores diferenciados na literatura. A maior parte deles tenta incorporar conhecimento sobre o problema ou sobre o espaço de busca nos operadores genéticos clássicos. lembre-se que se uma codificação diferente da binária for adotada, todos os operadores deverão ser definidos de acordo com esta codificação particular.

voltar para ALGORITMOS GENÉTICOS

3.11. Variantes clássicas: Micro AG e AG Paralelo

Apesar do seu caráter genérico, o desempenho dos AGs na sua forma convencional não se mostra satisfatório para muitos problemas que já possuem algoritmos eficientes para a sua solução aproximada. Há alguns conjuntos de modificações que já são clássicos. Entre eles, o Micro AG e o AG paralelo.

No **Micro Algoritmo Genético** (Micro GA), é empregada uma população muito pequena (até 10 indivíduos) e altas taxas de mutação (até 20%). A reinicialização freqüente e o elitismo são cruciais para a convergência do algoritmo. Para várias aplicações, há uma redução drástica do tempo computacional.

Outra variante é o **Algoritmo Genético Paralelo**, onde algumas populações evoluem em paralelo, como se vivessem em ilhas isoladas. De tempos em tempos, ocorre a migração de indivíduos de uma ilha para a outra, e ocorre a injeção de material genético novo nas populações. O número de indivíduos a migrar e a freqüência com que ocorre esta migração são alguns dos parâmetros do novo operador migração.

voltar para ALGORITMOS GENÉTICOS

3.12. Exemplos Ilustrativos

Na *homepage* do curso há um exemplo de aplicação de algoritmo genético com codificação real. O já conhecido problema de minimização de uma função polinomial é utilizado. Todos os parâmetros adotados e operadores usados estão descritos no programa. Observe que a convergência do AG é muito mais rápida que a do Recozimento Simulado, já que empregamos uma *população* de soluções tentativa, e não apenas uma.

voltar para ALGORITMOS GENÉTICOS