

Fernando Buarque
Carmelo Bastos

Computação Evolucionária

SPIN Springer's internal project number, if known

– Monograph –

April 24, 2010

Springer

Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Your dedication goes here

Preface

Here come the golden words

place(s),
month year

First name Surname
First name Surname

Contents

Part I	Part	Title	
<hr/>			
1		Algoritmos Genéticos	3
1.1		Inspiração, Histórico e Motivação	3
1.2		Visão Geral	4
1.3		Representação Individual	5
1.3.1		Binária	6
1.3.2		Inteira	7
1.3.3		Real	7
1.3.4		Permutação	7
1.4		Avaliação de sucesso	8
1.5		Operadores	8
1.5.1		Seleção	8
1.5.2		Recombinação	11
1.5.3		Mutação	12
1.6		Funcionamento	13
1.6.1		Inicialização	13
1.6.2		Parâmetros	13
1.6.3		Critérios de parada	14
1.6.4		Operação	14
1.7		Aplicações	15
1.8		Estudo de caso	15
1.9		Onde usar	15
1.10		Perspectivas futuras	15
		Problems	15
		Solutions	17
		References	19

Part I

Part Title

Algoritmos Genéticos

Equipe:

- George Moraes
- Carlos Timoteo
- Edinaldo
- Elton Oliveira

1.1 Inspiração, Histórico e Motivação

O problemas de otimização estão presentes nas mais diversas áreas da ciência, desde a matemática e física até a economia e sociologia. Grande parte da matemática desenvolvida no século XVIII visava o contexto de otimização de funções através de derivadas para determinar os pontos críticos de uma função. Os métodos de derivação são exatos e eficazes nesse contexto, mas apenas quando se trata de funções contínuas e deriváveis no ponto tmo. No entanto, para a maioria dos problemas do mundo real (por exemplo, problemas de otimização combinatória, otimização de síntese de circuitos eletrônicos, roteamento de veículos, otimização multiobjetivo) a aplicação dos métodos matemáticos de otimização tende a ser difícil devido a complexidade das funções que descrevem o espaço de soluções.

Historicamente, as primeiras iniciativas na área da Computação Evolucionária foram de biólogos e geneticistas interessados em simular os processos vitais de um ser humano em um computador, na época (final da década de 50) foram denominados de "processos genéticos". Na década de 60, o cientista John Holland, da Universidade de Michigan, iniciou um estudo, no qual era implementado uma população de indivíduos os quais possuíam cromossomos e eram sujeitos a operações de seleção, mutação e cruzamento. Holland tinha dois objetivos: melhorar o entendimento dos processos naturais de adaptação e projetar um sistema artificial com propriedades similares aos sistemas naturais. Os estudos de Holland passaram a ser conhecido como Algoritmos Genéticos. Esses estudos foram publicados em 1975 [Hol1975].

1.2 Visão Geral

Os Algoritmos Genéticos são métodos generalizados de busca e otimização que simulam os processos naturais de evolução, baseados nas idéias de seleção natural propostas por Charles Darwin [Dar1859], visando aplicá-las na resolução de problemas reais, como maximização e minimização de funções. Nessas simulações há uma população de indivíduos, os quais possuem uma constituição genética (cromossomo) que são codificações de soluções de um problema. Os indivíduos serão selecionados com o passar das gerações (iterações do algoritmo) em busca de melhores soluções. Geralmente, o conjunto de soluções iniciais é criado aleatoriamente e a cada geração esse conjunto é evoluído. Para simular o efeito de evolução, utilizam-se critérios de avaliação e seleção dos indivíduos. O primeiro determina a aptidão (ou *fitness*), enquanto que o segundo seleciona, de forma determinística ou probabilística, os indivíduos com maior aptidão que irão compor as gerações futuras. Assim, a tendência é que ao longo das gerações os indivíduos mais aptos sobrevivam e consequentemente ocorra a convergência para uma solução ótima (ou aproximadamente ótima) do problema, já que cada indivíduo representa uma solução.

Para cada tipo de problema pode haver uma maneira específica de se avaliar os indivíduos. Tomando um simples problema como exemplo: minimizar a função quadrática $f(x) = x^2$ no domínio $D(f) = [0, 15]$. Para esse caso, um número real pertencente ao domínio da função pode ser um possível cromossomo de um indivíduo. Supondo uma população inicial de dois indivíduos, na qual cada um possui um cromossomo que é um número binário. A avaliação dos indivíduos é realizada de acordo com o Algoritmo 1.

Algoritmo 1.1: Avaliação dos indivíduos

Converter valor do cromossomo para número decimal;
 Usar o número encontrado como a variável independente x da função;
 Usar o valor da avaliação da função $f(x)$ como aptidão do indivíduo;

A Tabela 1.1 mostra um exemplo com cromossomos dos dois indivíduos da população, gerados aleatoriamente, e os valores correspondentes de x e $f(x)$. Como este exemplo trata de um problema de minimização, o indivíduo I1 é considerado mais apto que o I2, pois seu cromossomo corresponde a um valor mais próximo ao mínimo da função.

A função que se deseja otimizar no problema é denominada função objetivo (ou *fitness function*). No exemplo descrito esta função era simples e conhecida ($f(x) = x^2$), no entanto os Algoritmos Genéticos são, geralmente, aplicados em problemas complexos de otimização. Uma das vantagens dos Algoritmos Genéticos a simplificação que eles permitem na formulação e solução de problemas de otimização. Por exemplo, não é necessário saber como funciona a função objetivo, é preciso apenas tê-la disponível para aplicá-la aos

Table 1.1. Cromossomos de indivíduos e função de avaliação

	Cromossomo x $f(x)$		
I1	0001	1	1
I2	0101	5	25

indivíduos, comparar os resultados e conhecer o formato das entradas. Sendo assim, as aplicações são para problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução; problemas com muitas restrições ou condições que não podem ser representadas matematicamente; e problemas com grandes espaços de busca. Alguns exemplos são: Otimização Combinatória [PS1998], Problema do Caixeiro Viajante (TSP - *Traveling Salesman Problem*) [SE2009], Otimização de Síntese de Circuitos Eletrônicos [LC1996], dentre outras aplicações.

Essa gama de aplicações para os Algoritmos Genéticos não seria possível apenas com os critérios que definem a aptidão dos indivíduos e o operador de seleção. Se utilizado somente esse operador a busca por soluções ficaria limitada apenas ao conjunto inicial de soluções, uma vez que não haveria alteração das soluções geradas. Por isso, outros dois operadores são empregados nos Algoritmos Genéticos, o operador de mutação e o de cruzamento (ou recombinação). O primeiro realiza alterações no cromossomo dos indivíduos e o segundo faz combinações entre dois indivíduos (os pais) gerando um terceiro (um filho) com o cromossomo constituído pela combinação dos cromossomos dos pais. Com isso, os novos cromossomos gerados, sejam por mutação ou por cruzamento, irão constituir novas soluções e, por conseguinte, irão diversificar o conjunto de soluções aumentando a capacidade de exploração do algoritmo e contribuindo para a busca por melhores soluções.

1.3 Representação Individual

Como citado na Seção 1.2, um indivíduo é uma representação de uma solução. Eles possuem dois atributos básicos:

Cromossomo: Codificação de uma solução. O cromossomo deve ser uma representação capaz de cobrir todo o conjunto dos valores no espaço de busca, e deve ter tamanho finito. Geralmente o cromossomo é uma sequência de *bits*. Além da representação binária, também se utilizam sequências de números inteiros, números reais, letras, ou outros tipos de dados específicos de um problema abordado. Essas sequências são, comumente, colocadas em *arrays*, pois facilitam as operações de cruzamento e mutação. Para problemas com múltiplas entradas é possível combinar as entradas (genes do indivíduo) em uma única sequência de *bits* como mostra a Figura 1.1.

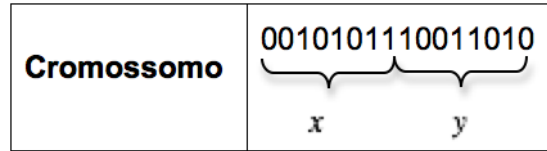


Fig. 1.1. Cromossomo binário com duas entradas

Aptidão (ou *fitness*): Qualidade da solução representada pelo indivíduo. É um atributo importante no momento de aplicação dos operadores sobre a população. Na seleção, os indivíduos mais aptos têm maior tendência de serem escolhidos do que os indivíduos menos aptos. No cruzamento, os indivíduos mais aptos terão maior possibilidade de se combinarem para gerar novas soluções. Por fim, na mutação, é possível estabelecer que certa quantidade de indivíduos mais aptos não sofrem a ação desse operador, para que as melhores soluções perdurem ao longo das gerações. Essa técnica é chamada de elitismo e será explicada na Seção 1.5.1.

Há diversas maneiras de fazer a representação individual, mas desde já é importante ressaltar que a representação de um indivíduo em Algoritmos Genéticos depende de sua conveniência em representar soluções para um dado problema. Por exemplo, em problemas de otimização de funções, uma das aplicações mais clássicas dos Algoritmos Genéticos, é interessante que os indivíduos sejam representados como entidades numéricas, onde cada solução candidata (indivíduo) será dada possivelmente por um conjunto de valores das variáveis da função a ser otimizada.

Os três tipos de representação numérica mais utilizados em Algoritmos Genéticos são: representação binária, representação inteira e representação real (ou representação de ponto flutuante).

Em muitos problemas, todavia, a representação de indivíduos torna-se mais conveniente pela utilização de um modo denominado permutação, no qual normalmente é levado em conta a ordem em que os atributos considerados aparecem na representação do indivíduo.

1.3.1 Binária

Este tipo de representação é amplamente utilizado em Algoritmos Genéticos. Sua implementação é normalmente muito simples e pode ser aplicada em diversas situações.

Consiste na representação de um cromossomo (indivíduo) como uma cadeia de bits, na qual cada bit ou conjunto de bits na cadeia representa um gene (característica) particular. Na representação binária de indivíduos para um problema particular, é importante que todas as representações possíveis do padrão de bits utilizado esteja dentro do domínio de possíveis soluções e que, equivalentemente, tal padrão de bits escolhido possa representar todos os elementos desse domínio (espaço de busca).

A representação binária é ideal para a codificar atributos binários de indivíduos, por exemplo, se o indivíduo é alto ou baixo, se é gordo ou magro, se é feminino ou masculino, se é terrestre ou marinho etc. Por outro lado, também é possível representar atributos que estejam dentro de uma faixa de valores (inteiros ou reais), como idade, altura, peso, ou até mesmo dentro de um conjunto de classes possíveis, como por exemplo, se o indivíduo em questão é peixe, ave, réptil, anfíbio ou mamífero.

Para casos de representações binárias para atributos não essencialmente binários (valores inteiros, reais, ou conjunto de classes), algum tipo de codificação extra deve ser utilizada. Por exemplo, para a representação de inteiros é possível utilizar a notação de complemento de dois, para reais, a notação de ponto flutuante e para as diferentes classes de um indivíduo seria possível utilizar um caractere ASCII para cada uma das classes ou formar nosso próprio padrão de codificação. Entretanto, nos casos em que atributos inteiros ou reais precisem ser representados, as representações inteiras e de ponto flutuante são mais adequadas, respectivamente.

1.3.2 Inteira

A representação inteira de indivíduos é mais adequada quando precisamos representar soluções que estão dentro de um intervalo inteiro de valores ou mesmo quando nosso espaço de busca é representado por valores enumerados, por exemplo, os valores $\{1,2,3,4,5,6,7\}$ representando Segunda, Terça, Quarta, Quinta, Sexta, Sábado. Genericamente, utilizamos a representação individual inteira quando os atributos de um indivíduo pertencem a qualquer conjunto inteiro de valores.

1.3.3 Real

Esse tipo de representação é também chamado de representação de ponto flutuante, devido ao tipo de codificação utilizada para a representação de números reais nos computadores. Semelhantemente ao que ocorre na representação inteira, a representação real é utilizada quando os indivíduos a serem representados estão dentro de um conjunto real de valores, ou quando tais valores estão dispostos em uma distribuição contínua. Obviamente essa representação contínua é hipotética para fins de representação computacional, dado que o conjunto de valores reais capazes de serem representados pela máquina é limitado.

1.3.4 Permutação

A representação por permutação é utilizada quando os atributos do problema são símbolos. Uma outra particularidade da representação por permutação é que, para certos problemas, não é permitido a um mesmo valor aparecer

mais de uma vez na sequência. Um problema de otimização que poderia ser resolvido utilizando-se esse tipo de representação é o conhecido problema do caixeiro viajante, no qual desejasse encontrar o menor percurso de maneira que partindo de uma cidade, dentre várias, seja possível passa por todas as cidades exatamente uma vez.

1.4 Avaliação de sucesso

A avaliação de sucesso é dada pela função de *fitness*. Cada indivíduo é avaliado a partir dessa função, que indicará sua aptidão, que por sua vez indica o quão bom é a solução representada por um indivíduo. A aptidão também é importante na aplicação dos operadores, explicados na Seção 1.5. Por exemplo, os indivíduos mais aptos têm maiores chances de sobreviver e passarem para a próxima geração.

1.5 Operadores

Nos Algoritmos Genéticos os operadores são fundamentais para o sucesso desse método de busca. Sem eles não seria possível realizar busca em amplitude e em profundidade no espaço de soluções:

- Busca em amplitude: consiste em visitar regiões que ainda não foram visitadas no espaço de busca, para tal é preciso criar diversidade no conjunto de soluções. Isso irá aumentar a capacidade de exploração (*exploration*) do algoritmo.
- Busca em profundidade: consiste em concentrar a busca em regiões mais promissoras do espaço de busca, ou seja, regiões que onde se encontram boas soluções para o problema. Esse tipo de busca tende a refinar as soluções encontradas de maneira que elas converjam para um ponto, podendo ser um máximo (ou mínimo) global ou local. A busca em profundidade está ligada à capacidade de prospecção (*exploitation*) do algoritmo. Uma boa técnica de otimização deve combinar esses aspectos, *exploration* e *exploitation*, para encontrar a solução ótima (ou aproximadamente ótima) de um problema (i.e. máximo ou mínimo global). Os Algoritmos Genéticos combinam ambos os aspectos através dos operadores de seleção (*exploitation*), mutação e cruzamento (*exploration*).

1.5.1 Seleção

Assim como na natureza segundo o princípio da seleção natural, os mais aptos ao meio sobrevivem, no algoritmo genético os indivíduos (ou potenciais soluções) de uma determinada geração com melhor avaliação na função de aptidão (fitness) serão selecionados e poderão gerar novos indivíduos passando

suas características, após a recombinação entre eles, para a geração seguinte. Espera-se, dessa forma, que a geração seguinte apresente melhores soluções, em média, para o problema. A solução é dada pelo melhor indivíduo, o mais apto, da última geração produzida. Esse operador é responsável pela convergência do algoritmo.

Em seguida serão citados os mecanismos de seleção. É importante destacar que eles diferenciam-se entre si pela diferente forma de induzir variação na aptidão média da população, a chamada pressão seletiva.

Seleção de pais

Seleção por roleta

Um método de seleção bastante utilizado é a seleção proporcional à aptidão, também conhecida como Seleção por Roleta. Os indivíduos são selecionados de forma probabilística. Cada um tem a probabilidade de ser escolhido proporcionalmente à sua aptidão (i.e. quanto mais apto mais terá chance de ser selecionado). A Figura 1.2 mostra um exemplo de roleta com quatro indivíduos.

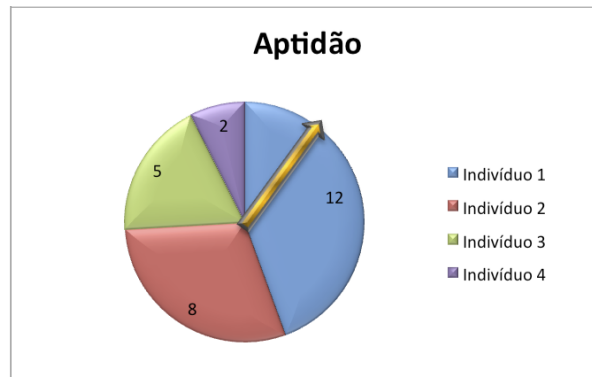


Fig. 1.2. Roleta com seções proporcionais à aptidão dos indivíduos

Ao "girar" esta roleta o indivíduo 1 terá a maior probabilidade de ser selecionado. Esse processo pode ser descrito segundo o Algoritmo 2.

Esse tipo de seleção pode apresentar problemas em situações com ocorrência de super-indivíduos, que atraem a convergência do processo evolucionário prematuramente; ou, no outro extremo, de competição próxima, quando a pressão seletiva é baixa demais devido a grande semelhança de aptidão dos indivíduos. Para reduzir esses problemas pode ser feita uma normalização do *fitness* dos indivíduos.

Algoritmo 1.2: Seleção dos indivíduos por roleta

Calcule a soma T dos valores de aptidão de todos os indivíduos da população;
 Ordene a população de forma decrescente de aptidão;
 Gere um número aleatório r no intervalo $[0, T]$;
 Percorra toda a população somando as aptidões dos indivíduos (soma s).
 Quando a soma s for maior que r , pare e retorne o indivíduo atual;

Seleção por torneio

Para a seleção por torneios, um grupo de quantidade fixa de indivíduos é aleatoriamente selecionado da população, e os melhores no grupo são selecionados. Nota-se que quanto maior o tamanho desse grupo, maior pressão seletiva.

Seleção de sobrevivência

Esse mecanismo consiste em selecionar certa quantidade de indivíduos para compor a geração seguinte. É uma forma de manter a memória ao longo do processo de busca. Existe basicamente duas abordagens para selecionar (ou descartar) indivíduos durante as gerações:

- Substituição baseado em idade: É uma estratégia utilizada nas primeiras versões dos Algoritmos Genéticos. Nessa abordagem cada indivíduo não sobrevive mais de uma geração, é sempre substituído por um filho na geração seguinte. Esse método pode reduzir a velocidade de convergência do algoritmo.
- Substituição baseado no *fitness*: O operador de recombinação (explicado na Seção 1.5.2) gera filhos, assim, aumentando a quantidade de indivíduos na população. Com isso, é necessário reduzir a quantidade de indivíduos que vão compor a próxima geração, voltando a quantidade inicial de indivíduos. Os indivíduos podem ser escolhidos usando o método da roleta tantas vezes quantas forem a quantidade de indivíduos que se quer selecionar. É possível, também, utilizar uma seleção por torneio. Outras abordagens também podem ser utilizadas como:

Descarte dos piores

Nesse método a população é ordenada do melhor para o pior *fitness* e n os piores indivíduos são descartados. n é a quantidade que excede a quantidade inicial de indivíduos na população. Essa estratégia pode levar conduzir rapidamente a melhoria da média de *fitness* da população, o que também pode levar a situações de convergência prematura, pois a população será guiada sempre pelos melhores indivíduos reduzindo a capacidade de gerar diversidade no processo de busca. Por essa razão, normalmente essa abordagem é usada em conjunto com grandes populações e prevenção de não gerar indivíduos duplicados.

Elitismo

A recombinação e a mutação podem excluir o melhor indivíduo da população. Para evitar que a melhor (ou as melhores) solução seja descartada da população emprega-se essa técnica chamada elitismo [SP1994]. Essa estratégia garante que os n melhores indivíduos sejam repassados para a geração seguinte. Isso pode ajudar a velocidade de convergência do algoritmo (bem mais suave que o método de Descarte dos piores) e no maior refinamento das melhores soluções, no entanto pode reduzir a capacidade do algoritmo escapar de ótimos locais durante a busca. Apesar dessa desvantagem, normalmente o elitismo melhora o desempenho dos Algoritmos Genéticos.

1.5.2 Recombinação

O operador de recombinação é conhecido também como operador de cruzamento. Ele combina os cromossomos de dois indivíduos pais gerando um ou dois filho(s) na população. Para descrever o funcionamento desse operador serão considerados dois indivíduos cujos cromossomo de tipo binário são mostrados na Figura 1.3.

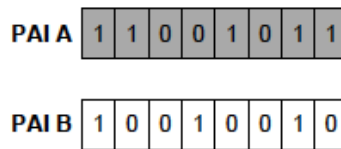


Fig. 1.3. Indivíduos que serão recombinados

Existem várias formas de realizar a operação de recombinação. Três dessas formas serão explicadas:

- Cruzamento de ponto único: é escolhido um ponto de cruzamento aleatoriamente, em seguida é copiado parte do cromossomo de cada pai, depois as partes são concatenadas gerando o cromossomo de um filho. Uma parte desse filho é formada pelos *bits* que vão do primeiro até o ponto de cruzamento de um pai e pelos *bits* que vão do ponto de cruzamento até o último *bit* do outro pai. A Figura 1.4 ilustra esta operação.
- Cruzamento de dois pontos: são escolhidos dois pontos de cruzamento aleatoriamente, em seguida é copiado parte do cromossomo de cada pai, depois as partes são concatenadas gerando o cromossomo de um filho. Uma parte desse filho é formada pelos *bits* que estão dentro da faixa gerada pelos pontos de cruzamento em um pai e pelos *bits* que estão fora da faixa no outro pai. A Figura 1.5 ilustra esta operação.

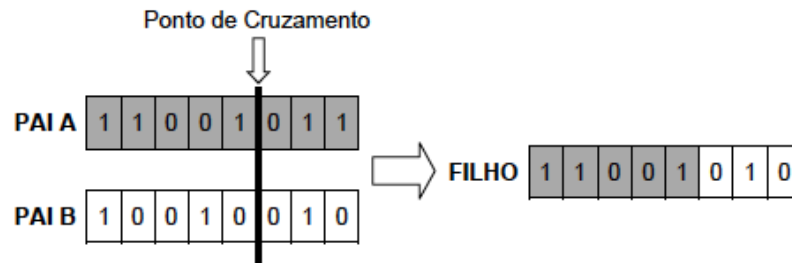


Fig. 1.4. Cruzamento de ponto único

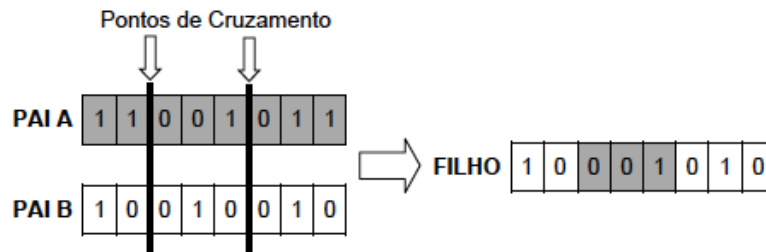


Fig. 1.5. Cruzamento de dois pontos

- Cruzamento uniforme ou multiponto: é gerado um indivíduo a partir de *bits* copiados aleatoriamente do primeiro ou do segundo pai. A Figura 1.6 ilustra esta operação.

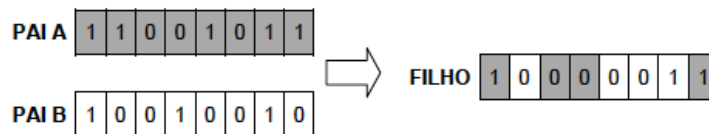


Fig. 1.6. Cruzamento uniforme

1.5.3 Mutação

O papel principal do operador de mutação é mudar aleatoriamente parte das informações codificadas de um cromossomo para criar novas soluções. Por outro lado, a operação de mutação pode ser empregada para reordenar codificações inválidas (não pertencentes ao espaço de busca), obtendo novas soluções viáveis e válidas.

No caso de indivíduos de cromossomo de tipo binário, a operação de mutação irá escolher alguns bits aleatoriamente e invertê-los. No caso de dos

cromossomos de tipo inteiro ou real, quando um número é escolhido pra sofrer mutação este será substituído por outro número gerado aleatoriamente. A Figura 1.7 mostra a operação de mutação; os bits que têm preenchimento branco são os alterados.

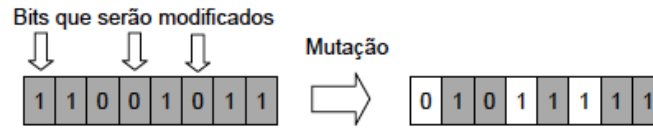


Fig. 1.7. Operação de Mutação

1.6 Funcionamento

1.6.1 Inicialização

De um modo geral, a inicialização dos Algoritmos Genéticos consiste em colocar o valor dos parâmetros e a população de indivíduos com dados aleatórios. No entanto, se há algum conhecimento prévio do espaço de busca do problema pode-se restringir os intervalos dos dados de entrada para que os indivíduos fiquem mais próximos das regiões mais promissoras do espaço de busca, ou seja, aquelas regiões que podem conter a solução ótima para o problema. Essa prática pode melhorar os resultados obtidos e reduzir o tempo de execução do algoritmo.

1.6.2 Parâmetros

Os parâmetros existentes nos Algoritmos Genéticos são:

- **Tamanho da População:** Determina o número de indivíduos na população, afetando diretamente o desempenho global e a eficiência dos Algoritmos Genéticos. Com uma população pequena, o desempenho pode ser insatisfatório, pois assim a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece maior cobertura do domínio do problema, além de prevenir convergências prematuras para máximos (ou mínimos) locais ao invés de globais. No entanto, o aumento da população, gera um aumento no custo computacional do algoritmo.
- **Taxa de Cruzamento:** Quanto maior for esta taxa, mais rapidamente novos indivíduos serão introduzidos na população. Mas se esta for muito alta, a maior parte da população pode ser substituída, aumentando a possibilidade de ocorrer perda de indivíduos de alta aptidão. Com um valor baixo,

o algoritmo pode tornar-se muito lento, afetando a velocidade de convergência, já que serão gerados poucos indivíduos novos a cada geração. Logo, reduz a capacidade de busca em amplitude.

- **Taxa de Mutação:** Determina a probabilidade com que uma mutação ocorrerá. Uma baixa taxa de mutação previne que um dado indivíduo fique estagnado no espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória além de aumentar muito a possibilidade de que uma boa solução seja destruída. A Taxa de Mutação é dependente da aplicação, mas para a maioria dos casos é entre 1 e 10%.
- **Taxa de Elitismo:** Determina a taxa de indivíduos mais aptos que serão repassados para a próxima geração. Se essa taxa for muito alta pode acarretar em estagnação na busca, pois muitos indivíduos de uma geração serão automaticamente repassados para a geração seguinte. Isto reduzirá a diversificação da população ao longo das gerações. Então as taxas mais baixas são as mais empregadas. Em muitos casos apenas o melhor indivíduo da população é automaticamente repassado para a geração seguinte.

1.6.3 Critérios de parada

Os critérios de parada podem variar de acordo com o problema que está sendo tratado. Alguns problemas exigem mais precisão nos resultados em detrimento de maior tempo de execução, outros exigem respostas rápidas em detrimento de menor precisão nos resultados. Mesmo com esses aspectos existem alguns critérios de parada comumente utilizados, são eles:

- **Número de iterações:** Quando atinge o número de gerações estabelecido.
- **Solução ótima encontrada:** Quando se conhece a solução ótima.
- **Convergência:** Quando nas últimas k gerações não houve melhora no *fitness* médio da população ou no melhor *fitness*.

1.6.4 Operação

Inicialmente cria-se a população. Os cromossomos dos indivíduos presentes nessa primeira geração podem ser gerados aleatoriamente ou gerados nas regiões mais promissoras do espaço de busca. Essa segunda forma é possível quando se tem um conhecimento prévio sobre o espaço de busca do problema. Em seguida cada indivíduo é avaliado de acordo com a função de avaliação do problema para definir a aptidão. Após a fase de avaliação, testa-se um critério de parada, que é a condição na qual se considera que o algoritmo encontrou uma solução aceitável ou simplesmente quando se atingir a quantidade máxima de gerações estipuladas inicialmente. Caso o critério de parada tenha sido satisfeito então o melhor indivíduo é apresentado como a solução do problema e a execução é encerrada. Caso contrário, aplicam-se os operadores na população, em seguida partindo para a próxima geração (nova iteração do algoritmo). O algoritmo retorna à fase de avaliação e segue os passos já descritos. A Figura 1.8 mostra o fluxograma de execução do algoritmo.

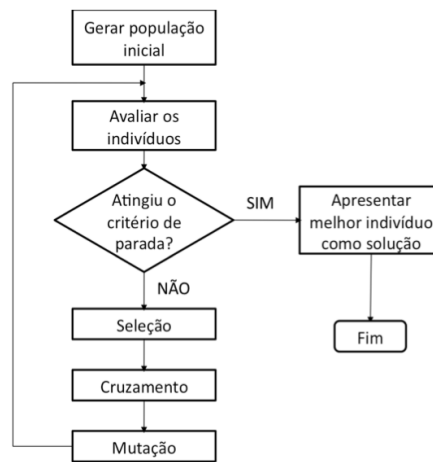


Fig. 1.8. Fluxograma básico de execução do Algoritmo Genético

1.7 Aplicações

1.8 Estudo de caso

1.9 Onde usar

1.10 Perspectivas futuras

Problems

1.1. The problem¹ is described here. The problem is described here. The problem is described here.

1.2. Problem Heading

(a) The first part of the problem is described here.

(b) The second part of the problem is described here.

¹Footnote

Solutions

Problems of Chapter ??

1.1 The solution is revealed here.

1.2 Problem Heading

- (a) The solution of first part is revealed here.
- (b) The solution of second part is revealed here.

References

- [Dar1859] DARWIN, C. On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. Editora: John Murray, Londres, (1859).
- [Hol1975] HOLLAND, J. H. Adaptation in Natural and Artificial Systems, Ann Arbor, MI: The University of Michigan Press, (1975).
- [LC1996] LIENIG, J.; COHOON, J. P. Genetic algorithms applied to the physical design of VLSI circuits: A survey. International Conference on Evolutionary Computation, Berlin, vol. 1141, p. 839-848, (1996).
- [PS1998] PAPADIMITRIOU, C. H.; STEIGLITZ, K. Combinatorial Optimization: Algorithms and Complexity. Estados Unidos, Edição: 1, Editora: DOVER SCIENCE, (1998).
- [SE2009] SALLABI, O. M.; EL-HADDAD, Y. An Improved Genetic Algorithm to Solve Traveling Salesman Problem. Proceedings of World academy of Science, Engineering and Technology, Roma, Itália, v. 40, (2009).
- [SP1994] SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A survey. IEEE Computer, vol. 27, p. 17-26, (1994).

kjhkjhkj