

Author(s)

Insert your Booktitle, Subtitle, Edition

SPIN Springer's internal project number, if known

– Monograph –

April 20, 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

1	Algoritmos Genéticos	3
1.1	Inspiração e Motivação	3
1.2	Histórico	4
1.3	Visão Geral	4
1.4	Representação Individual	6
1.4.1	Binária	7
1.4.2	Inteira	7
1.4.3	Real	7
1.4.4	Permutação	7
1.5	Dinâmica Populacional	7
1.6	Avaliação de sucesso	7
1.7	Operadores	7
1.7.1	Seleção	7
1.7.2	Recombinação	8
1.7.3	Mutação	9
1.8	Funcionamento	9
1.8.1	Inicialização	9
1.8.2	Parâmetros	9
1.8.3	Critérios de parada	10
1.8.4	Operação	10
1.9	Aplicações	11
1.10	Estudo de caso	11
1.11	Onde usar	11
1.12	Perspectivas futuras	11
	Problems	11
	Solutions	13
	References	15

Part I

Part Title

Algoritmos Genéticos

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 ??.

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 ??, 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.

Table 1.1. Cromossomos de indivíduos e função de avaliação

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

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 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 ??, Problema do Caixeiro Viajante (TSP - *Traveling Salesman Problem*) ??, Otimização de Síntese de Circuitos Eletrônicos ??, 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 1.3, 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.

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

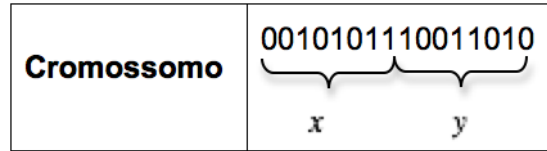


Fig. 1.1. Cromossomo binário com duas entradas

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 detalhada na Seção 1.7.1.

1.3.1 Binária

1.3.2 Inteira

1.3.3 Real

1.3.4 Permutação

1.4 Dinâmica Populacional

1.5 Avaliação de sucesso

1.6 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*).

1.6.1 Seleção

Seleção de pais

Seleção de sobrevivência

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 ???. Esse operador garante que os n melhores indivíduos sejam repassados para a geração seguinte.

1.6.2 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 1.2.

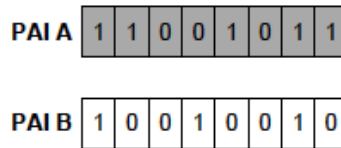


Fig. 1.2. Indivíduos que serão recombinados

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 1.3 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 1.4 ilustra esta operação.
- Cruzamento uniforme ou multiponto: é gerado um indivíduo a partir de *bits* copiados aleatoriamente do primeiro ou do segundo pai. A Figura 1.5 ilustra esta operação.

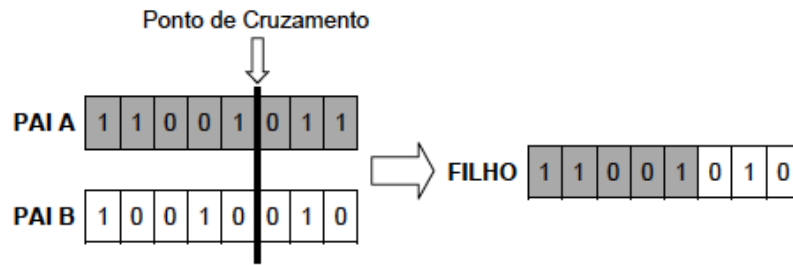


Fig. 1.3. Cruzamento de ponto único

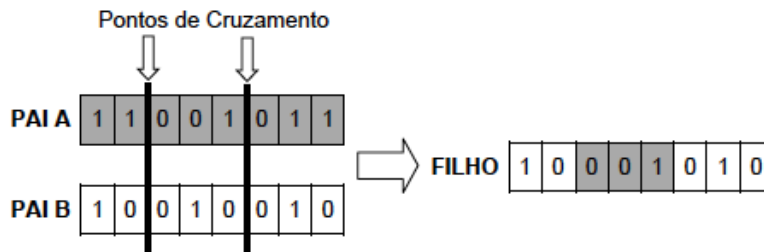


Fig. 1.4. Cruzamento de dois pontos

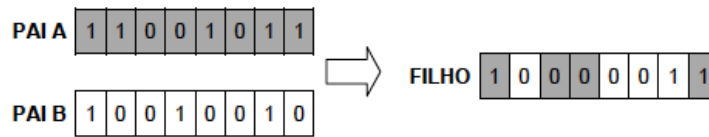


Fig. 1.5. Cruzamento uniforme

1.6.3 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 1.6 mostra a operação de mutação; os bits que têm preenchimento branco são os alterados.

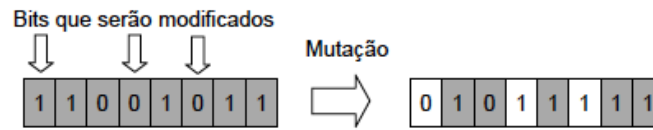


Fig. 1.6. Operação de Mutaçao

1.7 Funcionamento

1.7.1 Inicializaçao

1.7.2 Parâmetros

Os parâmetros existentes nos Algoritmos Genéticos são:

- **Tamanho da Populaçao:** 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 Mutaçao:** 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 soluçao seja destruída. A Taxa de Mutaçao é dependente da aplicaçao, 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.

1.7.3 Critérios de parada

1.7.4 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 1.7 mostra o fluxograma de execução do algoritmo.

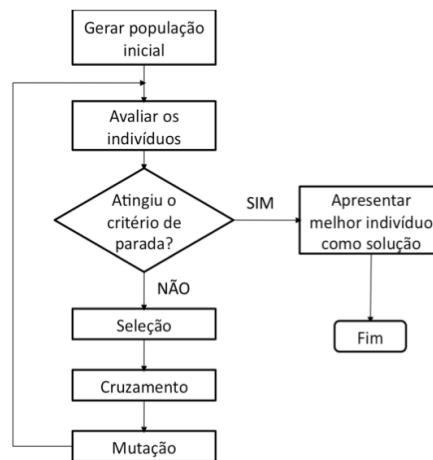


Fig. 1.7. Fluxograma básico de execução do Algoritmo Genético

1.8 Aplicações

1.9 Estudo de caso

1.10 Onde usar

1.11 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

- [KLR73] Kagan, A.M., Linnik, Y.V., Rao, C.R.: Characterization Problems in Mathematical Statistics. Wiley, New York (1973)
- [Mey89] Meyer, P.A.: A short presentation of stochastic calculus. In: Emery, M. (ed) Stochastic Calculus in Manifolds. Springer, Berlin Heidelberg New York (1989)
- [MR97] Miller, B.M., Runggaldier, W.J.: Kalman filtering for linear systems with coefficients driven by a hidden Markov jump process. Syst. Control Lett., **31**, 93–102 (1997)
- [Ros77] Ross, D.W.: Lysosomes and storage diseases. MA Thesis, Columbia University, New York (1977)

kjhkjhkj