

Dicas para Implementação do Algoritmo Genético

Charles Henrique Porto Ferreira

June 2015

1 Introdução

Neste tutorial vou mostrar como visualizar e implementar o problema da resolução do labirinto usando algoritmos genéticos. As dicas mostradas servirão como guias, porém existem outras formas de se fazer, não sendo necessário seguir todos os passos rigorosamente.

2 Visualização do problema

Definição do problema: Achar a solução de um labirinto de tamanho $n * m$. Exemplo:

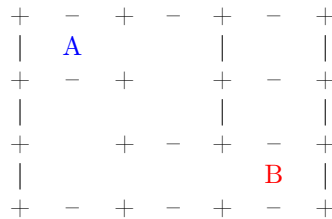
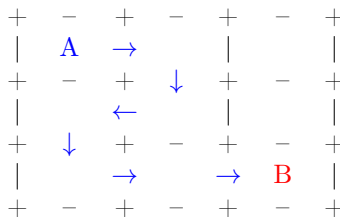


Figura 1: Exemplo de labirinto

Aqui temos um labirinto de tamanho $3 * 3$. Suponha que desejamos sair de um ponto A, definido no canto superior esquerdo, e chegar até um ponto B, definido no canto inferior direito. Uma possível solução para o problema seria:



Podemos representar as setas por direções como mostra a Tabela 1

Tabela 1: Direções

seta	direção
↑	N
↓	S
→	L
←	O

Logo a solução do labirinto mostrado poderia ser resumida na seguinte sequência de direções:

L	S	O	S	L	L
---	---	---	---	---	---

3 Codificando a solução

Agora como poderíamos codificar esse problema em um algoritmo genético?

Primeiramente precisamos decidir como será a representação do cromossomo do algoritmo genético. A solução mostrada anteriormente, em forma de uma lista de direções, já mostra uma sugestão de como implementar isso em um algoritmo genético. Porém seria mais interessante se essa representação fosse em forma de bits ao invés de direções, pois isso facilitaria na hora de fazer as operações de cruzamento e mutação, como será explicado mais para frente. Assim podemos codificar as direções de forma binária.

Tabela 2: Direções em código binário

seta	direção	binário
↑	N	00
↓	S	01
→	L	10
←	O	11

A representação da solução apresentada anteriormente na forma de um cromossomo do algoritmo genético ficaria:

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

Um detalhe importante que precisa ser levado em consideração na hora de representar a solução do algoritmo genético está na quantidade de bits necessários para formar o cromossomo. No exemplo mostrado, o cromossomo tem a quantidade de bits necessário para aquela solução porém isso não quer dizer que essa quantidade seja capaz de resolver qualquer configuração do labirinto. Nesses casos devemos considerar o pior caso para determinar a quantidade de bits, ou seja, considerar o caminho mais longo do labirinto, ver quantos passos são necessários, para finalmente determinar a quantidade de bits.

+	-	+	-	+	-	+
	A	→		→		
+	-	+	-	+	↓	+
		←		←		
+	↓	+	-	+	-	+
		→		→	B	
+	-	+	-	+	-	+

O labirinto mostrado acima possui o pior caso possível, o caso em que é necessário dar o número máximo de passos para achar a solução, que neste caso seriam 8 passos. Como cada passo é representado por dois bits, de acordo com a Tabela 2, nosso cromossomo deve ter 16 bits.

3.1 Algoritmo Genético

Depois de definir como será o cromossomo do AG, precisamos desenvolver um AG que seja capaz de chegar nessa solução. Em outras palavras o AG precisa, sozinho, produzir um cromossomo como aquele mostrado anteriormente - um cromossomo que seja a solução do problema.

3.2 Inicialização

O primeiro passo de um AG é criar a população inicial, que nada mais é do que criar um conjunto de cromossomos que sejam possíveis soluções do problema. Como inicialmente o AG não sabe qual é a solução do problema, a primeira coisa a ser feita é criar um conjunto de cromossomos com valores aleatórios para os bits.

Vamos supor que o algoritmo genético vá gerar 5 cromossomos na população inicial¹.

C1	1	0	1	0	1	1	1	1	0	1	0	0	0	1	0	0
C2	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1
C3	0	1	1	0	1	0	1	1	0	1	0	0	0	1	0	0
C4	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	1
C5	0	0	0	0	1	0	0	1	1	1	0	1	0	1	0	1

¹5 cromossomos é um número pequeno para a população inicial, está sendo usado aqui apenas para efeito de exemplo

Depois de gerar a população inicial, a pergunta que precisamos fazer é: Algum desses cromossomos representa a solução do problema? Caso nenhum deles represente, como eu poderia saber o quanto cada um deles se aproximou da solução final? Essa medida é o que chamamos de *fitness* do indivíduo.

3.3 Cálculo do *Fitness*

O *fitness* serve para definir a qualidade da solução de um cromossomo. Através dele vamos saber se o cromossomo é bom ou ruim. O cálculo do *fitness* é totalmente dependente do tipo de problema que se deseja resolver. Como estamos lidando com um labirinto onde damos passos para se deslocar através dele, podemos usar como *fitness* uma medida que avalie a diferença entre a distância final do cromossomo após percorrer o labirinto e o seu destino final.

Vamos usar o labirinto da Figura 1 para calcular o *fitness* de cada um dos 5 cromossomos. O primeiro passo é decodificar os bits para sabermos quais as direções que devem ser seguidas. O desenho dos cromossomos já foi feito usando cores diferentes a cada dois bits para representar a direção que deve ser tomada, porém vamos “decodificar” os bits mesmo assim apenas para ajudar o entendimento.

C1	L	L	O	O	S	N	S	N
C2	L	O	L	N	N	L	N	S
C3	S	L	L	O	S	N	S	N
C4	L	O	O	O	S	L	N	O
C5	N	N	L	S	O	S	S	S

Executando os passos de cada cromossomo temos a posição final alcançada por cada um deles. Se considerarmos cada bloco do labirinto como uma coordenada (x, y) de uma matriz, poderemos escrever a posição alcançada em termos dessas coordenadas.

Tabela 3: Posição final gerada por cada um dos cromossomos

cromossomo	coordenada final
C1	(0,0)
C2	(1,1)
C3	(0,0)
C4	(0,0)
C5	(2,0)

Dado que a coordenada inicial era o ponto A, a posição (0,0) e desejamos chegar no ponto B, posição (2,2). Podemos usar como *fitness* a distância de Manhattan, que leva em consideração quantos passos são necessários para se chegar a solução final a partir do ponto em que estamos².

$$d = 4 - (|xb - xa| + |yb - ya|) \quad (1)$$

Na Equação 1 a letra d representa a distância de Manhattan, (xa, ya) representa sua posição atual e (xb, yb) representa o destino que se deseja alcançar. O valor 4 é adicionado para transformar o problema de minimização em um problema de maximização. Perceba que se calcularmos puramente a distância de Manhattan o nosso objetivo seria conseguir cada vez um valor menor, minimizando a distância. No labirinto da Figura 1, o número mínimo de passos da posição inicial até a posição final é 4. Logo usando esse número na equação podemos transformar o problema em um problema de maximização³.

Usando a equação 1 podemos calcular o *fitness* de cada um dos cromossomos da Tabela 3

Tabela 4: Posição final e *fitness* de cada um dos cromossomos

cromossomo	coordenada final	<i>fitness</i>
C1	(0,0)	0
C2	(1,1)	2
C3	(0,0)	0
C4	(0,0)	0
C5	(2,0)	2

²O cálculo da distância não leva em consideração as paredes do labirinto

³Não é obrigatório transformar um problema de minimização em um problema de maximização, porém fica mais fácil pensar na solução de um problema quando estamos tentando melhorar a solução de maneira crescente

Como podemos observar pela Tabela 4 nenhum cromossomo conseguiu gerar uma sequência de caminhos que atingisse o objetivo. Isso já era esperado, uma vez que os cromossomos da população inicial são gerados com valores aleatórios. Agora o próximo passo é fazer essa população se multiplicar para conseguir gerar novos indivíduos, ou seja, novas soluções.

3.4 Cruzamento

O Cruzamento é um passo do algoritmo genético usado para gerar novos cromossomos a partir dos cromossomos existentes. Existem diversas maneiras de se implementar um cruzamento. Será mostrado aqui o cruzamento de um ponto, que é a forma mais simples de cruzamento. Neste tipo de cruzamento, primeiramente definimos um ponto - P - do cromossomo que para fazer o corte. Esse ponto é definido aleatoriamente. Depois selecionamos, também aleatoriamente, dois cromossomos da população para serem cruzados, que chamaremos de cromossomos X e Y. Esses dois cromossomos irão gerar dois cromossomos filhos, que chamaremos de F1 e F2. Na Figura 2 temos uma representação visual do processo de cruzamento.

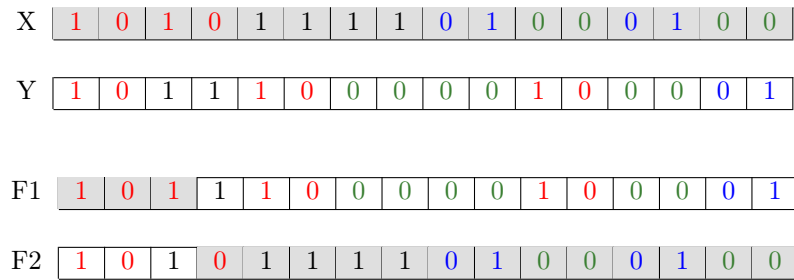


Figura 2: Cruzamento de um ponto

Neste exemplo da Figura 2 foi determinado que o ponto P seria a posição 3 e os cromossomos selecionados foram os cromossomos C1 e C2 respectivamente. Os dois filhos gerados representam duas novas soluções para o problema.

F1	L	O	L	N	N	L	N	S
F2	L	L	O	O	S	N	S	N

Cada uma com seu respectivo fitness:

Tabela 5: Posição final e fitness dos cromossomos filhos

cromossomo	coordenada final	fitness
F1	(1,1)	2
F2	(0,0)	0

A etapa do cruzamento pode ser executada várias vezes durante a mesma geração, para aumentar o tamanho da população e gerar novas soluções. Não existe um número fixo de vezes que deve ser executado, nem um número máximo de filhos a serem gerados. Esses valores são determinados empiricamente.

3.5 Mutação

Até agora nenhum dos cromossomos foi capaz de achar a solução do labirinto. Muitas vezes pode acontecer casos em que uma população evolui até um ponto em que ela converge e não consegue sair deste estado. A mutação tem a função de tentar fazer o algoritmo genético explorar novas áreas permitindo um alcance maior.

A implementação da mutação é bem simples, embora ela possa ser feita de várias maneiras diferentes. A forma básica de executar a mutação consiste em selecionar um cromossomo, escolher um dos seus bits e mudar o valor. Tudo de forma aleatória. Em termos de codificação do problema, a mutação iria mudar uma das direções do cromossomo, o que poderia ser bom ou ruim, dependendo da configuração do cromossomo.

3.6 Seleção

Após as etapas de cruzamento e mutação a população de um algoritmo genético tende a ficar muito grande. Agora é preciso aplicar uma seleção natural aos cromossomos, de forma a selecionar os indivíduos mais aptos para comporem a próxima geração. Existem diversas maneiras de se implementar a seleção usando algoritmos genéticos. Uma das mais conhecidas é a roleta. Esta técnica consiste em dar uma probabilidade de seleção para

cada indivíduo de acordo com o seu *fitness*, portanto os cromossomos que tiverem os maiores *fitness* terão maior probabilidade de serem selecionados. O cálculo da probabilidade de seleção é mostrado na Equação 2.

$$p(i) = \frac{f(i)}{\sum_{j=1}^n f(j)} \quad (2)$$

onde $p(i)$ é a probabilidade do *fitness* $f(i)$ ser selecionado, e n é a quantidade de cromossomos que compõem a população atual.

Cromossomo	Fitness	Probabilidade(%)
C1	0	0
C2	2	33,33
C3	0	0
C4	0	0
C5	2	33,33
F1	2	33,33
F2	0	0

Utilizando os cromossomos gerados até aqui podemos calcular a probabilidade de seleção de cada um deles.

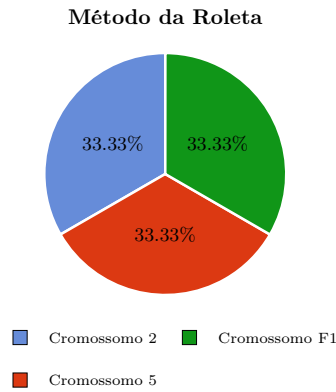


Figura 3: Distribuição de probabilidade de seleção usando o método da roleta

A Figura 3 é uma representação visual da técnica da roleta, na prática podemos implementá-la distribuindo a probabilidade da roleta entre os seus respectivos cromossomos e gerando um número aleatório entre 0 e 100% para decidir quais os cromossomos serão selecionados.

Cromossomo	Distribuição
C2	0 - 33
C5	34 - 66
F1	67 - 100

Agora vamos supor que queremos passar para a próxima geração apenas 4 indivíduos, assim vamos gerar 4 números aleatórios variando de 0 à 100. Considere que os números gerados foram: 29, 57, 85, 8. Neste caso os cromossomos escolhidos seriam: C2, C5, F1, C2.

Perceba que o cromossomo C2 foi selecionado duas vezes. Neste caso podemos colocar uma cópia dele na população. Um fator importante a ser observado foi que alguns cromossomos não entraram na seleção por possuírem um *fitness* igual 0, ou seja, o pior *fitness* possível. Isso acontece nos casos em que eles não conseguiram sair do lugar.

Após a seleção dos indivíduos é necessário que o algoritmo genético reinicie os processos de cruzamento, mutação e seleção. Esses processos são repetidos por várias gerações. Existem dois critérios de paradas que são mais usados para finalizar o algoritmo genético. Um deles consiste em parar a execução do AG caso ele alcance a solução esperada (aplicável a casos em que o problema é bem definido). Em outras situações pode ser definido um número máximo de gerações para o algoritmo genético ser executado⁴. Ou também pode ser usado uma combinação das duas estratégias, caso seja muito difícil alcançar a solução desejada.

A Figura 4 mostra um fluxograma geral de um algoritmo genético

⁴Cada reinício do processo de cruzamento, mutação e seleção representa uma nova geração do AG

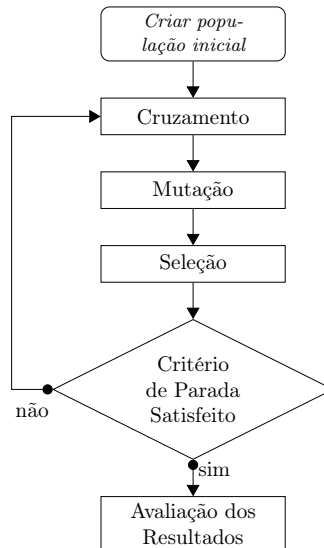


Figura 4: Fluxograma Algoritmo Genético

4 Considerações

Como foi dito anteriormente, existem outras maneiras de se implementar um algoritmo genético, não sendo necessário se prender 100% neste modelo apresentado. É uma prática comum atribuir uma probabilidade para acontecer a mutação e o cruzamento durante as gerações, ou seja, não é obrigatório que os dois sempre aconteçam. Existem casos em que a população inicial não é gerada aleatoriamente, podendo ser feito seguindo alguma heurística. A mesma idéia pode ser aplicada para efetuar a mutação, ou até mesmo para definir os pontos de corte dos cruzamentos.