

Algoritmos Genéticos

Carolina Ribeiro Xavier

Março de 2024

1 Algoritmos genéticos com representação dos parâmetros por números reais

Os AGs para estimativa parâmetros reais se diferenciam dos AGs vistos anteriormente, principalmente, em dois aspectos: a representação do indivíduo e os operadores genéticos. No entanto, o fluxograma que vimos anteriormente ainda é válido.

2 Questões de projeto

A mudança na representação reflete na mudança de implementação de detalhes de todos os operadores genéticos. Os principais focos de mudança são listadas a seguir:

- Função objetivo; (parâmetros não precisam de transformação)
- Cruzamento;
- Mutação;

2.1 Função objetivo

Para que possamos comparar os resultados das implementações, vamos manter a mesma função objetivo, e faremos a minimização da mesma.

$$f(x) = -20\epsilon^{-0.2}\sqrt{\frac{1}{n}\sum x_i^2} - \epsilon^{\frac{1}{n}}\sum \cos(2\pi x_i) + 20 + \epsilon \quad (1)$$

2.2 Tipo de representação Real

Para problemas de ajuste de parâmetros podemos usar um vetor números reais, que reflete exatamente o conjunto de variáveis que queremos encontrar.

Para $n = 2$ na 1, por exemplo, teríamos duas variáveis a serem ajustadas, ao escolhermos a representação real, o indivíduo do AG seria um vetor de duas posições do tipo real, daí cada vetor é exatamente a solução candidata e suas posições podem ser aplicadas diretamente como parâmetros da função objetivo.

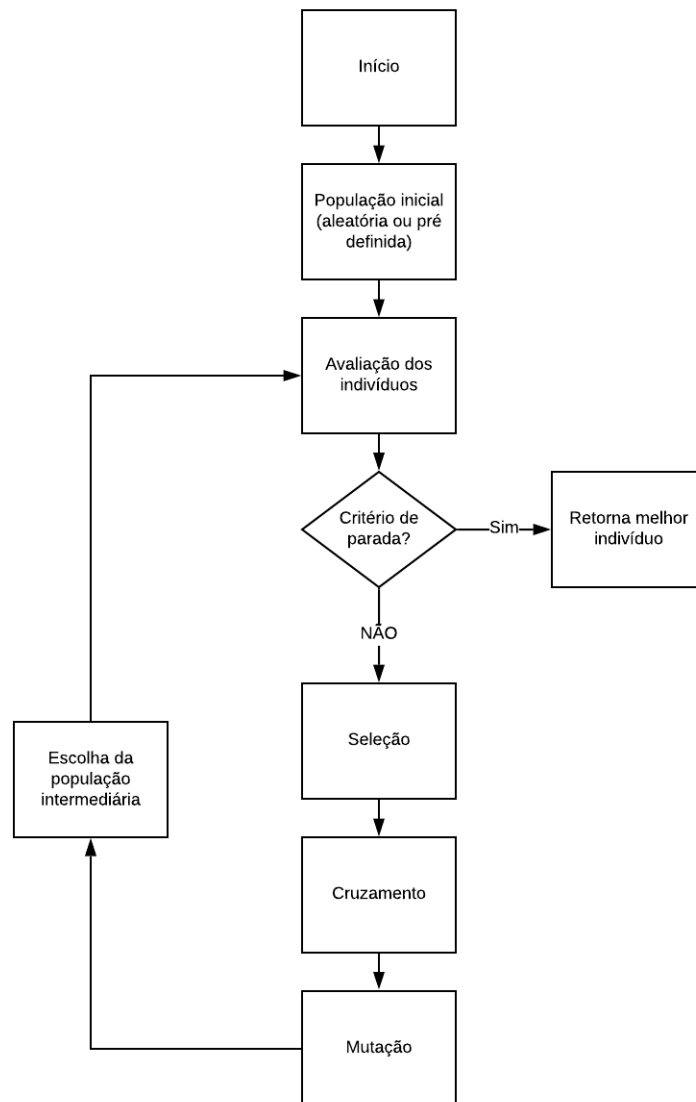


Figure 1: Fluxograma de um algoritmo genético básico

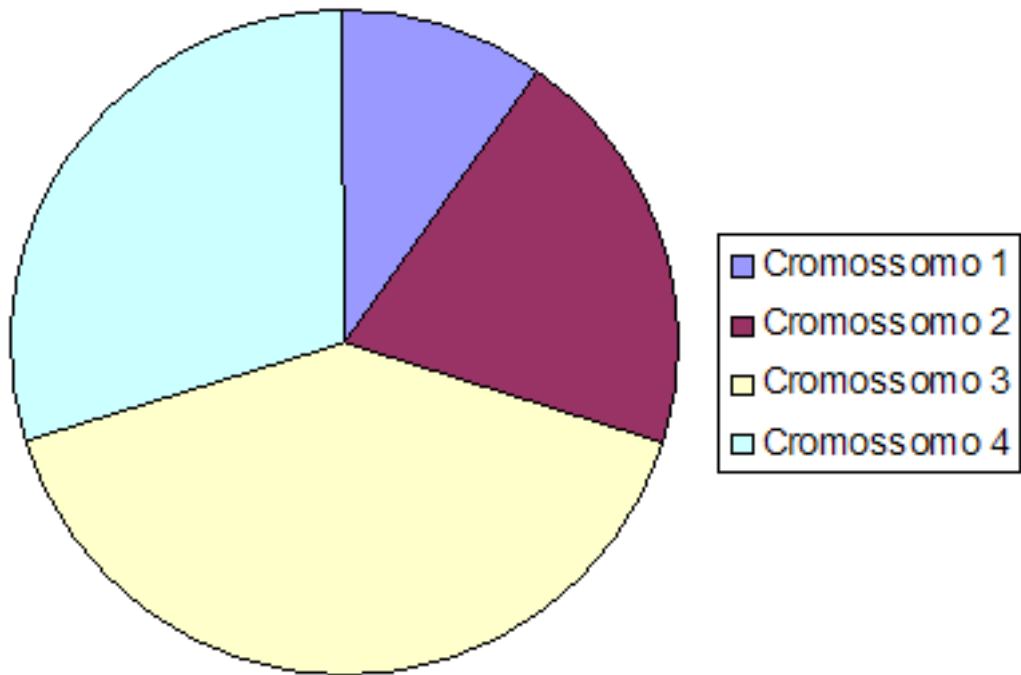


Figure 2: Critério de seleção por roleta

2.3 Uma nova estratégia de seleção de pais

Uma operação importante que antecede todas as outras é a seleção de pais. Na primeira implementação utilizamos o torneio. Agora utilizaremos um método mais elaborado, a roleta.

Para simulação de uma roleta devemos seguir os seguintes passos, considerando n_{pop} o número de indivíduos na população:

- 0 - Caso o objetivo seja **minimização**, deve-se inverter os valores de fitness, caso o zero não seja um valor possível de fitness, $1/\text{fitness}$ é adequado, se zero for um valor possível, será necessário ajustar o caso do $\text{fitness}=0$. Nesse ponto, pode-se fazer uma suavização dos valores também.
- 1 - calcular a soma dos valores de fitness de todos os indivíduos: $total = \sum_{i=1}^{n_{pop}} fit[i]$
- 2 - normalizar o valor de fitness, dividindo cada um o valor de fitness de cada um pela soma calculada anteriormente e guardando esses valores em um novo vetor, assim, a soma dos elementos desse vetor será um $\forall_i, roleta[i] = \frac{fit[i]}{total} = 1$.
- 3 - para seleção de cada pai realizar os seguintes passos até que se tenha o número desejado de pais:
 - 3.1 sortear um número $r = [0..1]$

- 3.2 percorrer o vetor *roleta* acumulando em uma variável a soma dos valores desse vetor até que, no passo j , o valor da soma acumulada alcance ou ultrapasse o valor r , assim o j -ésimo indivíduo do vetor será selecionado como pai.
- 3.3 Se o número de pais não for suficiente, voltar para o 3.1.

2.4 Operadores genéticos

Os cruzamentos mais comuns para representações por números reais são os baseados em média ou os que envolvem o sorteio de valores em torno dos valores de cada característica dos pais, como o BLX- α e o BLX- $\alpha\beta$, que serão detalhados a seguir.

Algoritmo BLX- α

- selecione dois pais da geração anterior X e Y.
- crie dois filhos X' e Y' como segue:
- for $i = 1$ to dimensao faça
 - $d[i] = |X[i] - Y[i]|$
 - sorteie um número real aleatório u no intervalo
 $(\min(X[i], Y[i]) - \alpha * d[i], \max(X[i], Y[i]) + \alpha * d[i])$
 - $u = \text{rand}(\min(X[i], Y[i]) - \alpha * d[i], \max(X[i], Y[i]) + \alpha * d[i])$
 - $X'[i] = u$
 - sorteie um número real aleatório u no intervalo
 $(\min(X[i], Y[i]) - \alpha * di, \max(X[i], Y[i]) + \alpha * di)$
 - $u = \text{rand}(\min(X[i], Y[i]) - \alpha * di, \max(X[i], Y[i]) + \alpha * di)$
 - $Y'[i] = u$

α - parâmetro positivo real, valor sugerido $\alpha = 0.5$

Algoritmo BLX- $\alpha\beta$

- Selecione dois pais da geração anterior X e Y.
- Considerando que X é melhor do que Y (se não for trocar X com Y)
- Crie dois filhos X' e Y' como segue:
- for $i = 1$ to n do
 - $d[i] = |X[i] - Y[i]|$
 - if $(X[i] \leq Y[i])$ then
 - sorteie um número real aleatório u no intervalo
 $(X[i] - \alpha * d[i], Y[i] + \beta * d[i])$
 - $u = \text{rand}(X[i] - \alpha * d[i], Y[i] + \beta * d[i])$
 - $X'[i] = u$
 - sorteie um número real aleatório u no intervalo

```

(X[i] - α * d[i], Y[i] + β * d[i])
u = rand(X[i] - α * d[i], Y[i] + β * d[i])
Y'[i] = u
else
    sorteie um número real aleatório u no intervalo
    (Y[i] - β * d[i], X[i] + α * d[i])
    u = rand(Y[i] - β * d[i], X[i] + α * d[i])
    X'[i] = u
    sorteie um número real aleatório u no intervalo
    (Y[i] - β * d[i], X[i] + α * d[i])
    u = rand(Y[i] - β * d[i], X[i] + α * d[i])
    Y'[i] = u

```

α e β são parâmetros reais positivos, valores sugeridos $\alpha = 0.75$ e $\beta = 0.25$.

A principal diferença entre os dois algoritmos é que no BLX- α , os valores a serem sorteados estão em torno dos valores dos dois pais indistintamente, enquanto no BLX- $\alpha\beta$ os valores estão mais próximos do pai com o melhor fitness, por isso os possíveis valores de α serão superiores aos valores de β .

Lembrem-se: Após a etapa de cruzamento, os indivíduos da população intermediária podem ser somente indivíduos novos, ou caso a taxa de cruzamento seja diferente de 100%, pode ter ainda indivíduos antigos, de outras gerações. Esses indivíduos estarão sujeitos a sofrer mutação de seus genes.

2.5 Mutação

A mutação mais comum para representação por número reais envolve a substituição da posição sorteada para ser mutada por um valor no intervalo definido para a variável, para essa implementação iremos fazer dessa forma, a posição sorteada para ser mutada terá seu valor substituído por um valor aleatório no intervalo definido inicialmente como limite do parâmetro que está sendo mutado.

2.6 Elitismo

O elitismo é uma estratégia de manutenção da(s) melhor(es) solução(ões) através das gerações. Essa estratégia deixa a solução do algoritmo mais elegante, pois a curva de convergência se torna, no mínimo, monótona, não oscilando quanto a aproximação da solução desejada.

Para implementação do elitismo iremos transferir o melhor indivíduo da geração atual para a população intermediária, em um posição aleatória.

3 Implementação

Agora vamos implementar um AG simples com representação real, sendo cada cromossomo exatamente os parâmetros da função da Equação 1. A implementação será de acordo com o fluxograma da Figura 1.

Defina a estrutura de dados que você irá armazenar os indivíduos e seus respectivos valores de fitness e inicie a sua implementação.

Outros valores a serem definidos são:

- Tamanho da população;
- Número máximo de gerações a serem executadas;
- Critério de seleção de pais (Roleta)
- Taxa de cruzamento (Calibrar)
- Cruzamento por $\text{blend-}\alpha\beta$ crossover
- Taxa de mutação (Calibrar)
- Mutação (substituição do valor a ser mutado por um valor qualquer do intervalo)
- Sugere-se usar o elitismo, que compreende a sobrevivência do melhor indivíduo para geração seguinte.

Junto com essa implementação você fará a análise da eficiência da mesmo e um estudo dos parâmetros, descritos nas seções que seguem.

4 Avaliando a eficiência da execução

Vamos começar a avaliar a eficiência de nossos algoritmos de forma criteriosa. Para isso, iremos "salvar" alguns dados intermediários de nossas execuções para plotagem de gráficos e análises estatísticas básicas dos nossos experimentos.

Devemos ter em mente que o AG é um algoritmo probabilístico, e portanto, deve ser executado várias vezes com o mesmo conjunto de parâmetros para obtenção de resultados confiáveis. Cada execução pode ser avaliada pelo melhor indivíduo da última geração, a média dos indivíduos da última geração, ou ainda, a evolução da população através das gerações. Para isso, a cada execução, vamos salvar em arquivos texto os dados necessários para plotar gráficos e construir tabelas para análises posteriores.

Podemos salvar:

- Um arquivo com a lista de indivíduos de cada geração e os respectivos fitness;
- Um arquivo com o melhor indivíduo de cada geração e seu fitness.

Executaremos o AG por diversas vezes (nesse caso, como a função não é muito complicada pode ser 10 vezes), vamos arquivar os valores que devemos analisar a cada execução e por fim calcular, média aritmética, desvio padrão, identificar a melhor e a pior execução, em termos de fitness, para um mesmo conjunto de parâmetros.

Para cada geração de cada execução, serão salvos o fitness do melhor indivíduo e a média dos valores de fitness de todos os indivíduos. Daí, poderemos plotar gráficos e desenhar tabelas que comparem a evolução de diferentes execuções para um determinado conjunto de parâmetros com o passar das gerações.

5 Calibragem dos parâmetros

Fazer um experimento fatorial completo variando taxas de mutação, os valores de tamanho de população e número de gerações e, por vezes, operadores genéticos e fazer um estudo estatístico que qual conjunto dentre todos os possíveis, melhor resolve o nosso problema.

Exemplo:

Taxa de mutação: 1% 5% 10%

Taxa de cruzamento: 60% 80% 100%

Tamanho da população: 25 50 100

Número de gerações: 25 50 100

Cruzamento: de ponto, média aritmética blend- $\alpha\beta$

Mutação: por perturbação ou por substituição de valor

Com ou sem elitismo

Vejam exemplos a seguir:

6 Comparação com implementação 1

Escreva sobre a diferença da eficiência da implementação 1 com essa implementação.

Taxa de cruzamento	Elitismo	Probabilidade de mutação	Cruzamento	#Gerações	#População	Melhor aptidão		Aptidão média	
						μ	σ	μ	σ
0.8	False	0.10	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.439	0.137
0.8	False	0.10	$BLX-\alpha$	100	100	4.441e-16	0.000e+0	0.424	0.123
0.8	True	0.05	$BLX-\alpha$	100	100	4.441e-16	0.000e+0	0.221	0.127
0.8	True	0.10	$BLX-\alpha\beta$	100	50	4.441e-16	0.000e+0	0.450	0.223
0.8	True	0.10	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.467	0.112
1.0	False	0.01	$BLX-\alpha\beta$	50	100	4.441e-16	0.000e+0	0.065	0.036
1.0	False	0.01	$BLX-\alpha\beta$	100	50	4.441e-16	0.000e+0	0.033	0.073
1.0	False	0.01	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.030	0.044
1.0	False	0.05	$BLX-\alpha$	100	100	4.441e-16	0.000e+0	0.181	0.114
1.0	False	0.05	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.139	0.057
1.0	False	0.10	$BLX-\alpha$	100	100	4.441e-16	0.000e+0	0.390	0.106
1.0	False	0.10	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.364	0.097
1.0	True	0.01	$BLX-\alpha$	100	100	4.441e-16	0.000e+0	0.041	0.034
1.0	True	0.01	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.033	0.037
1.0	True	0.05	$BLX-\alpha\beta$	50	100	4.441e-16	0.000e+0	0.158	0.074
1.0	True	0.05	$BLX-\alpha\beta$	100	25	4.441e-16	0.000e+0	0.189	0.179
1.0	True	0.05	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.180	0.116
1.0	True	0.10	$BLX-\alpha\beta$	50	100	4.441e-16	0.000e+0	0.392	0.091
0.8	False	0.05	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.223	0.114
0.6	False	0.10	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.632	0.176
1.0	True	0.10	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.271	0.087
0.6	False	0.01	$BLX-\alpha\beta$	100	100	4.441e-16	0.000e+0	0.068	0.052
1.0	True	0.05	$BLX-\alpha$	100	100	7.993e-16	1.123e-15	0.202	0.107
0.6	True	0.01	$BLX-\alpha\beta$	100	100	7.993e-16	1.123e-15	0.035	0.034
1.0	True	0.01	$BLX-\alpha\beta$	50	100	7.993e-16	1.123e-15	0.024	0.049

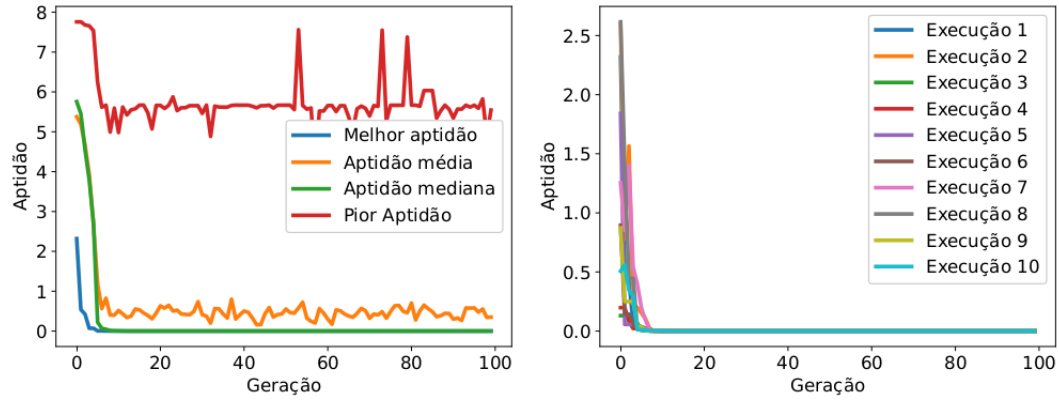


Figura 1: Execução do melhor conjunto de parâmetros (0.8,False,0.10, $BLX-\alpha\beta$,100,100)

Figure 3: Exemplo