Algoritmo genético

Algoritmos Bioinspirados

Heitor Lourenço Werneck
github.com/heitor57
heitorwerneck@hotmail.com

5 de outubro de 2020

1 Introdução

Algoritmos genéticos se referem a uma família de modelos computacionais inspirados na evolução(seleção natural e genética). Esses algoritmos representam a solução problema atráves de estruturas semelhantes a cromossomos e aplicam operadores de recombinação para preservar os dados/informações vitais [7, 1]. Normalmente um algoritmo genético será baseado em população e usará operadores de seleção e recombinação para buscar uma solução no espaço de busca [7].

Algoritmos genéticos são utilizados em diversos cenários pela natureza generalista de seu modelo, como por exemplo: sistemas de recomendação [4, 5]; visão computacional [2]; robôs de limpeza [8]; recuperação de informação [6] e agendamento de voos [3].

Os algoritmos genéticos também possuem a capacidade de fazer *exploitation* e *exploration* que são características de modelos estado da arte em diversas áreas, como por exemplo: *Multi-Armed Bandits*; Nuvem de partículas e outros. Uma característica extremamente importante para diversos cenários.

Esse trabalho consiste na implementação e análise de um algoritmo genético para solução de um problema de otimização de uma função.

2 Problema

O problema é simples e consiste na minimização da função da equação 1.

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

É importante notar que como o problema será representado em um sistema computacional como um vetor de números binários então o espaço de busca terá 2^{bits} possibilidades de solução. Isso mostra que é um problema difícil, pois possui uma complexidade de ordem exponencial no espaço de soluções, por isso é importante usar alguma heurística ou meta-heurística (que no caso é algoritmo genético) para solucionar esse problema caso não tenha solução analítica viável.

3 Solução

A estruturas basicas para o algoritmo foram determinadas da seguinte forma: os genes são números binários de tamanho $\#Bits$ (definido posteriormente) tal que é transformado em decimal dentro de um espaço linear $[x_{min}, x_{max}]$ para ser avaliado pela função objetivo. A estrutura foi feita dessa forma pois permite mais flexibilidade nas operações genéticas e também delimita um espaço de busca discreto que pode ajudar na solução de alguns problemas.

O intervalo de busca de valores foi dentro do domínio $x_{min} = -2$ e $x_{max} = 2$, isso por que foi analisado que dentro dessa espaço há bons valores para minimização da função. Foi utilizado uma população com tamanho 50 e o número de iterações utilizados foi de 20. O espaço linear delimitado dentro deste domínio

contém $2^{\#Bits}$ valores sendo $\#Bits = 6$ e 2 genes ($\#Genes = 2$; genes de $\#Bits$ bits) utilizados. Esses valores serão variados posteriormente para análise do comportamento do algoritmo.

A estrategia de seleção utilizada foi a roleta, tal que é selecionado dois indivíduos aleatórios na população e o indivíduo com maior aptidão (menor valor na função objetivo) será selecionado como o primeiro pai a compor o par de pais com probabilidade de 90%. Após isso o processo é repetido excluindo o indivíduo já selecionado e dois pais são obtidos.

Para a intensificação do espaço de busca foi utilizado o cruzamento, tal que o algoritmo irá selecionar um ponto dentro do intervalo $[1, \#Bits \cdot \#Genes - 2]$ e esse ponto irá ser utilizado para o particionamento dos pedaços do genoma para criação dos filhos. A taxa de cruzamento utilizada por padrão foi de 100%.

Após isso é utilizado uma mutação na população, para diversificação da população, essa mutação é feita com base em uma probabilidade de cada celula ser alterada, no caso o valor padrão utilizado foi de 1%.

Com a mutação nos filhos o processo de elitismo é feito na população anterior, uma porcentagem de indivíduos com maior aptidao é dado como entrada e esses indivíduos vão substituir os piores indivíduos da geração atual. Foi feito a abordagem por porcentagem ao invés de somente um indíviduo para ser possível a análise da variação do parâmetro. O valor padrão foi de 5%.

Foi escolhido não mutar o indivíduos da elite para gerar uma curva de convergência sem oscilação.

Tabela 1: Valores padrões para os parâmetros

Parâmetro	Valor	Descrição
x_{min}	-2	Limite inferior do espaço discreto de busca
x_{max}	2	Limite superior do espaço discreto de busca
#População	50	Quantidade de indivíduos
#Gerações	50	Quantidade de gerações
#Bits	6	Tamanho do gene em bits
#Genes	2	Quantidade de genes
Taxa de cruzamento	1.0	Taxa do número de indivíduos que vão cruzar
Taxa de elitismo	0.05	Taxa do número de indivíduos a serem preservados
Taxa de mutados	1.0	Taxa do número de indivíduos aptos a serem mutados da população total
Taxa de mutação	0.01	Probabilidade da mutação de um bit em um gene
Probabilidade do vencedor	0.9	Probabilidade do vencedor de uma disputa ser pai

Uma visão geral de como o algoritmo funciona é dado a seguir:

Algoritmo 1 Algoritmo genético

1: Inicia a população com cromossomos aleatórios
2: **for** $i = 1$ to #Gerações **do**
3: Cruza os indivíduos utilizando torneio para selecionar os pais e de acordo com a taxa de cruzamento
4: Completa o número de índivíduos com os melhores indivíduos da população anterior caso a taxa de cruzamento não seja de 100%
5: Faz a mutação dos indivíduos da população atual de acordo com a taxa de mutados e de mutação
6: Seleciona os melhores indivíduos da população anterior de acordo com a taxa de elitismo e cada um deles substitui um indíviduo aleatório da geração corrente
7: **end for**

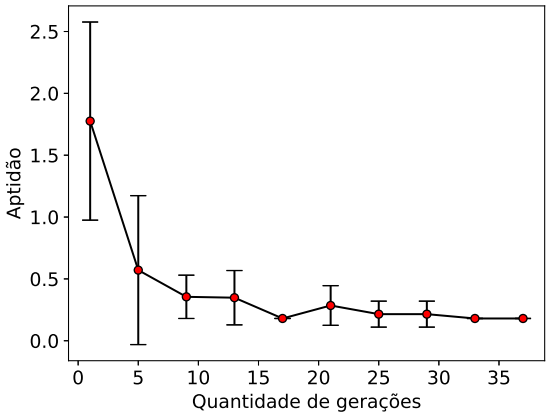
4 Análise de resultados

4.1 Análise do comportamento de cada parâmetro isoladamente

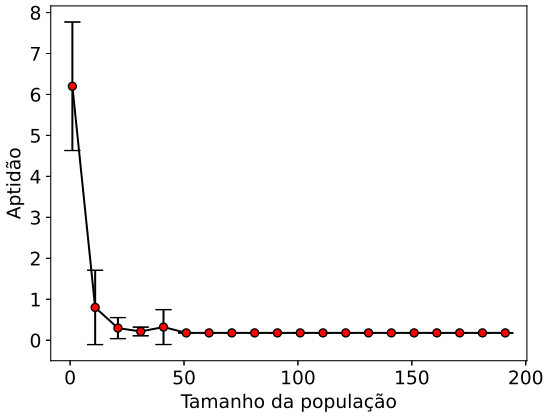
Para análisar os resultados foram utilizados os valores padrões mencionados anteriormente e um parâmetro é selecionado e analisado por vez, sendo esses os parâmetros mais significativos. Primeiro será analisado os parâmetros mais triviais e comprovado seu comportamento que já é esperado.

Primeiro é importante mencionar que cada ponto no gráfico é a média do melhor indivíduo na última geração de 10 execuções do mesmo parâmetro.

É possivel ver pela figura 1(a) que com o aumento do número de gerações a tendência é que chegue mais proximo do mínimo, uma constatação esperada que foi comprovada assim como na figura 1(b).



(a)

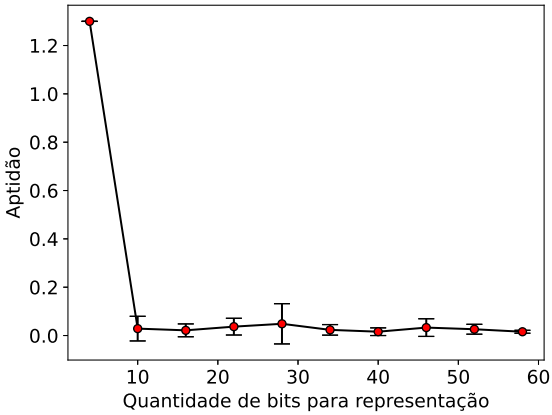


(b)

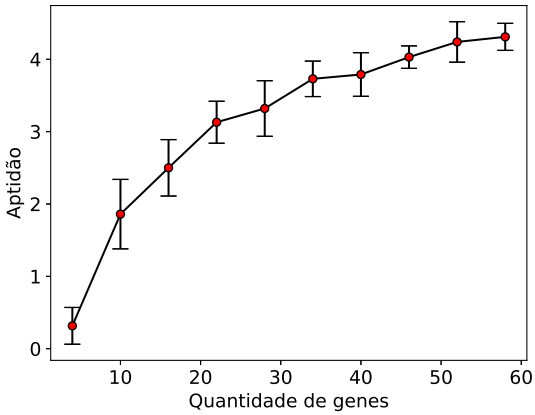
Figura 1: Variação dos parâmetros.

Já na figura 2(a) o número de bits é variado e quanto maior o número de bits mais números existem no espaço de busca com isso, assim como o valor ótimo pode ser obtido com mais precisão também é mais custoso achar o ponto ótimo.

Pela figura 2(b) é possível observar que quanto mais genes mais o algoritmo se afastou do ótimo, isso deve-se a o aumento de dificuldade na obtenção do ótimo pela quantidade de novos genes que precisam se aproximar dos números ótimos.



(a)



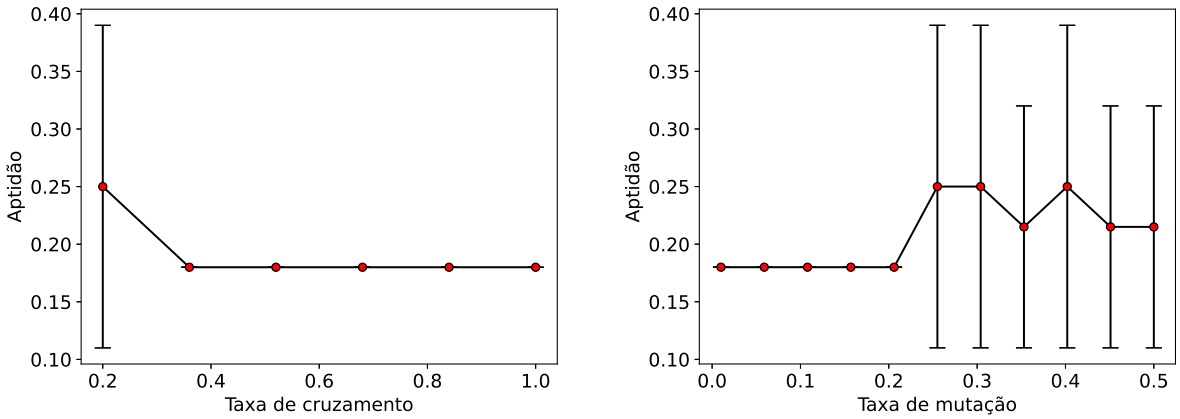
(b)

Figura 2: Variação dos parâmetros

A figura 3(a) mostra que é importante ser feito cruzamento para uma busca pela solução ótima efetiva. Como o espaço de busca considerado é bem pequeno esse parâmetro e outros não apresentam um comportamento com bastante variabilidade, porém em futuros trabalhos essa análise podera ser feita de maneira mais visível.

ção precisa ter um limite se não somente ira gerar desordem na busca pelo ótimo, isso é comprovado no gráfico

No geral a figura 3(b) mostra que a taxa de mutação so atrapalha quando a probabilidade de mutação é muito alta, pois desse modo hávera muita variação/diversificação nos valores o que causa muita desordem para ser possível fazer *exploitation*.



(a) (b)

Figura 3: Variação dos parâmetros

4.2 Otimizando hiperparâmetros

Agora para solucionar o problema será feito a busca pelo melhor parâmetro que soluciona o problema. Como o valor padrão para o número de bits é muito baixo, se torna muito fácil achar a solução ótima e fica sem sentido a busca pelo melhor parâmetro visto que a maioria das soluções são ótimas (ótima em relação ao domínio), devido a isso será utilizado 25 bits. Para isso será considerado os parâmetros que possuem mais impacto: gerações; população; taxa de cruzamento e taxa de mutação. A tabela 2 a seguir descreve os domínios de busca para cada parâmetro.

Tabela 2: Domínio de busca de cada parâmetro.

#Gerações	{50,100}
#População	{50,100}
Taxa de cruzamento	{0.6, 0.8, 1.0}
Taxa de mutação	{0.01, 0.05, 0.1}

A partir da execução o *top*-10 de parâmetros são descritos na tabela a seguir:

#População	#Gerações	Taxa de cruzamento	Taxa de mutação	Média F_o	Desvio padrão F_o
100	100	0.8	0.01	3.982e-05	2.949474e-05
100	100	1.0	0.01	6.5716e-05	9.0926290e-05
100	100	0.6	0.01	0.0001981	0.00019136
50	100	0.6	0.01	0.0002714	0.000179354
50	100	0.8	0.01	0.000417	0.000392633
100	50	0.6	0.01	0.00049	0.000307831
100	50	0.8	0.01	0.0007104	0.000726924
50	100	1.0	0.01	0.00125232	0.00116128
50	50	0.8	0.01	0.0013694	0.001276186
100	100	1.0	0.05	0.0017521	0.002844937

É interessante se notar que os três melhores conjunto de parâmetros so se diferenciam pela taxa de cruzamento. E mesmo somente pela taxa de cruzamento é possível observar uma diferença significativa entre o *top*-1 e o *top*-3, assim mostrando a importancia desse parâmetro. É possível ver que as melhores soluções para esse problema não utilizam diversificação de maneira ingenua, sendo mais conservador nas operações de mutação e cruzamento. Também é importante notar que o alto valor de população e gerações é importante para essa configuração mais conservadora pois ele converge para a melhor solução lentamente.

Na figura 4.2 há as execuções do algoritmo com o melhor parâmetro, é possível ver que ele chega em um valor baixo rapidamente porém continua convergindo até a solução durante as gerações. Também houve pouca variação nas execuções, como também foi mostrado na tabela, porém no começo poucas execuções demoram a descer para próximo do valor ótimo.

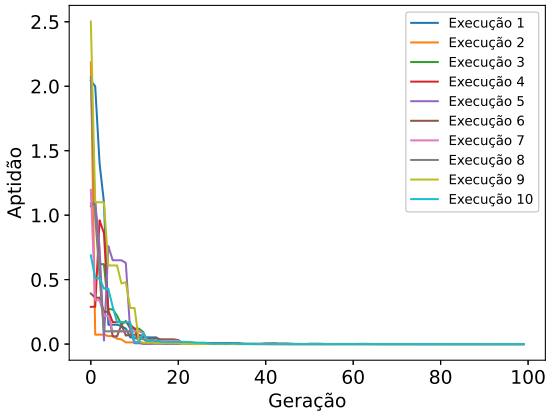


Figura 4: Execuções do melhor parâmetro

5 Conclusão

Com esse trabalho foi possível fazer a análise detalhada do comportamento de um modelo de algoritmo genético. Atráves de multiplos gráficos os parâmetros do modelo foram análisados. Como o problema a ser tratado é bem simples o algoritmo majoritariamente consegue achar o ponto ótimo do domínio.

Referências

[1] David E Goldenberg. Genetic algorithms in search, optimization and machine learning, 1989.

[2] Minglun Gong and Yee-Hong Yang. Quadtree-based genetic algorithm and its applications to computer vision. *Pattern Recognition*, 37(8):1723–1733, 2004.

[3] Loo Hay Lee, Chul Ung Lee, and Yen Ping Tan. A multi-objective genetic algorithm for robust flight scheduling using simulation. *European Journal of Operational Research*, 177(3):1948–1968, 2007.

[4] Jeff Naruchitparames, Mehmet Hadi Güneş, and Sushil J Louis. Friend recommendations in social networks using genetic algorithms and network topology. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2207–2214. IEEE, 2011.

[5] Nitai B Silva, Ren Tsang, George DC Cavalcanti, and Jyh Tsang. A graph-based friend recommendation system using genetic algorithm. In *IEEE congress on evolutionary computation*, pages 1–7. IEEE, 2010.

[6] Dana Vrajitoru. Crossover improvement for the genetic algorithm in information retrieval. *Information processing & management*, 34(4):405–415, 1998.

[7] Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):nil, 1994.

[8] Mohamed Amine Yakoubi and Mohamed Tayeb Laskri. The path planning of cleaner robot for coverage region using genetic algorithms. *Journal of innovation in digital ecosystems*, 3(1):37–43, 2016.