Mens Agitat, 16 (2021) 22-25 *ISSN 1809-4791*

Modelagem Matemática dos Algoritmos Imperialist e Genetic

Mathematical Modeling of Imperialist and Genetic Algorithms

Luciano Nascimento[1*]

[1*]*Grupo de Estudo em Matemática Aplicada/GEMA,Patos-PB, Brasil.*

Resumo

A otimização é altamente demanda nos processos de modelagem matemática. Os algoritmos naturais desempenham um papel fundamental nesta tarefa, pois possibilita a busca por um mínimo global de forma rápida e com baixo custo computacional. Neste contexto, foram analisados dois algoritmos: Competitivo Imperialista e Genético. O objetivo deste trabalho foi realizar uma abordagem comparativa entre os algoritmos, apresentando a modelagem matemática usando o soft MATLAB. Neste, foram implementados os códigos e geradas as saídas das trajetórias de menor custo sobre a curva de contorno da função objetivo e uma evolução do custo sobre as iterações, onde utilizou-se 17 iterações do algoritmo genético para localizar o mínimo global e 24 iterações do imperialista competitivo.

Palavras-chave: Soft MATLAB. Modelagem Matemática. Algoritmos.

Abstract

Optimization is highly demanded in mathematical modeling processes. Natural algorithms play a fundamental role in this task, as it makes it possible to search for a global minimum quickly and with low computational cost. In this context, two algorithms were analyzed: Competitive Imperialist and Genetic. The objective of this work was to carry out a comparative approach between the algorithms, presenting the mathematical modeling using the soft MATLAB. In this, the codes were implemented and the outputs of the lower cost trajectories were generated on the contour curve of the objective function and a cost evolution on the iterations, where 17 iterations of the genetic algorithm were used to locate the global minimum and 24 iterations of the competitive imperialist.

Keywords: Soft MATLAB. Mathematical Modeling. Algorithms.

1. INTRODUÇÃO

O processo de otimização consiste em ajustar as entradas ou características de um dispositivo, processo matemático, baseado na saída de um processo ou função, a qual é denominada custo. Geralmente, a otimização se diferencia da determinação das raízes da função, por utilizar a derivada da função, invés da própria função, no caso das raízes, onde o processo adiciona um nível maior de complexidade para cálculo [1]. Com o desenvolvimento da Teoria dos Jogos, de Von Neumann, foi criada uma área de otimização sem a necessidade do uso de gradientes, para a localização dos mínimos das funções e/ou processo. A partir daí diversos métodos para o cálculo do mínimo foram desenvolvidos, como o método de Nelder-Mead Downhill Simplex, e o método de minimização de linha [2]. Nas últimas décadas, algoritmos baseados em processos naturais foram desenvolvidos, como o algoritmo genético, que replica a evolução de uma população de soluções candidatas, baseada na variação genética e seleção natural [3]. Outro exemplo é o algoritmo de colônia de formigas, que simula o comportamento de forrageamento de formigas. Um outro exemplo se baseia nas relações político-econômicas humanas, como no imperialismo competitivo, onde os países são divididos em dois tipos: estados imperialistas e colônias [6]. Este artigo tem por objetivo realizar um estudo comparativo entre os algoritmos competitivo imperialista (ACI) e o algoritmo genético (AG).

2. MODELAGEM MATEMÁTICA

2.1 Algoritmo Competitivo Imperialista

As colônias são divididas entre os imperialistas, e o poder total do império se dá por intermédio do poder do imperialista e uma porcentagem da média das colônias. Durante a execução do algoritmo, as colônias se movimentam na direção dos imperialistas, e aqueles que não conseguem se manter com o mesmo poder ou decaem, sendo eliminados. No estado final, existe apenas um império e todos os outros países tem o mesmo poder e posição que o imperialista. Inicialmente, um país é definido por um *array*, contendo n variáveis que definem a posição no espaço em que o país se encontra. Assim, um país é representado por:

$$país = [p_1, p_2, p_3, \dots, p_{nvar}]$$

O custo para um país, é determinado pela função custo f, pelas variáveis de entrada $(p_1, p_2, \dots, p_{n_{var}})$, logo

$$custo = f(país) = f(p_1, p_2, p_3, \dots, p_{n_{var}})$$

Para inicializar o algoritmo de otimização, deve-se gerar um população inicial de tamanho N_{pop}. Posteriormente, são selecionado N_{imp}, que serão os países mais poderosos, na forma de impérios. O restante N_{col}, serão as colônias, que são divididas entre os impérios. A divisão é baseada no poder de cada império, o qual pode ser definido por:

$$C_n = c_n - \max\{c_i\} \quad (1)$$

Onde c_n é o custo do enésimo imperialista e C_n é o custo normalizado. De posse do custo normalizado de cada imperialista, é possível definir o poder de cada um desses por:

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (2)$$

Com o poder de cada um deles, é possível determinar o número de colônias com

$$NC_n = round\{p_n * N_{col}\} \quad (3)$$

onde NC_n é número inicial de colônias do enésimo império. Num segundo momento, as colônias começam a se dirigir para o entorno do imperialista, na proporção de $x \sim U(0, \beta x d)$,onde β é um número maior que 1 e d é a distância entre a colônia e o imperialista. Se β>1, ocasiona que as colônias estão próximas do imperialista. Para incluir um fator de desvio, então, faz-se:

$$\theta \sim U(-\gamma, \gamma) \quad (4)$$

onde γ ajusta o parâmetro de desvio da direção original. Valores de β e γ são arbitrário, mas para $\beta = 2$ e $\gamma = \frac{\pi}{4}(rad)$, resultaram é uma boa convergência. Caso uma colônia alcance um valor de mínimo menor que o imperialista, ambos trocam de papéis no contexto do algoritmo, e as demais colônias convergem para a nova posição. O passo seguinte é o cálculo do poder total do império. O poder das colônias é insignificante para o cálculo, mas ainda assim é considerado. Para tanto, o custo total é dado por [4]:

$$T.C_n = custo(imperialista) + \xi media\{custo(colonias_n)\} \quad (5)$$

onde $T.C_n$ é o custo total do enésimo império e ξ é um número positivo com o valor menor que 1. Quanto menor o valor de ξ, mais relevante o poder do imperialista, em detrimento das colônias. Para efeito geral, é utilizado ξ=0,1. No estágio de competição entre os imperialistas, o império que tiver menor poder, tende a perder suas colônias para os imperialistas com maior poder, nem sempre sendo o com maior poder. Para definir o processo de sucessão, é necessário definir a probabilidade de cada império absorver a colônia. O custo total normalizado é definido por:

$$NTC_n = TC_n - \max\{TC_i\} \quad (6)$$

onde TC_N e NTC_N são respectivamente o custo total e o normalizado para cada império. Com isso, o poder de possessão sobre as colônias é definido por

$$p_{p_n} = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \quad (7)$$

Cada P_p se refere a probabilidade de possessão da colônia de por cada império. Para definir as colônias entre os impérios, é criado um vetor nomeado **P**, como:

$$P = \left[p_{p_1}, p_{p_2}, p_{p_3}, \dots, p_{p_{N_{imp}}} \right] \quad (8)$$

Depois, é criado um vetor com o mesmo tamanho de **P** e os elementos são número aleatórios distribuídos uniformemente.

$$R = \left[r_1, r_2, r_3, \dots, r_{N_{imp}} \right] \quad (9)$$

$$r_1, r_2, r_3, \dots, r_{N_{imp}} \sim U(0,1)$$

Depois, é criado um vetor **D**, que é o resultado da subtração de **R** de **P**.

$$D = P - R = \left[D_1, D_2, D_3, \dots, D_{N_{imp}} \right] \quad (10)$$

$$= \left[p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, \dots, p_{p_{N_{imp}}} - r_{N_{imp}} \right]$$

O império que obter o maior dentro os elementos do vetor **D** será o que irá ter a possessão sobre as colônias.

Durante este processo, em um determinado momento, o império que não tiver mais colônias será eliminado. O momento final se dá quando apenas um imperialista restar e todas as colônias tiverem o mesmo valor de custo do imperialista [5].

2.2 Algoritmo Genético

No caso do AG, o indivíduo da população é chamado de cromossomo. Assim como no ACI, o cromossomo é definido como um *array* 1xN_{var} , onde N_{var} é a quantidade de variáveis do domínio do problema [6].

$$cromossomo = [p_1, p_2, p_3, \dots, p_{N_{var}}] \quad (11)$$

Cada cromossomo possui um custo associado, o qual é definido pelo cálculo utilizando a função custo com as variáveis $p_1, p_2, \dots, p_{N_{var}}$

$$custo = f(cromossomo) = f(p_1, p_2, \dots, p_{N_{var}}) \quad (12)$$

Para inicializar o algoritmo, é necessário definir uma população inicial N_{pop}, a qual é dada por

$$populacao = rand(N_{pop}, N_{var}) \quad (13)$$

Todas as variáveis são normalizadas entre 0 e 1, as quais ao serem passadas para a função custo são desnormalizadas, obedecendo a seguinte regra:

$$p = (p_{super} - p_{infer})p_{norm} + p_{infer} \quad (14)$$

onde p_{super} e p_{infer} são os limites definidos para a população inicial e p_{norm} o valores das variáveis normalizadas.

Posteriormente, o processo de seleção natural é aplicado. Para tanto, a população é ordenada de forma decrescente, e os melhores cromossomos, baseados no seu custo são selecionados. A quantidade de sobreviventes é definida por uma variável chamada N_{sobrev} . Os que não são considerados ‘aptos’ são descartados. Isso possibilita que novos cromossomos sejam adicionados e cruzem, possibilitando o aparecimento de novas gerações. A seleção é feita de forma aleatória.

Com isso, o algoritmo entra no estado de cruzamento. Um dos métodos utilizados é o intercâmbio de variáveis entre os genitores.

$$genitor_1 = [p_{m_1}, p_{m_2}, p_{m_3}, p_{m_4}, p_{m_5}, p_{m_6}, \dots, p_{m_{N_{var}}}] \quad (15)$$

$$genitor_2 = [p_{d_1}, p_{d_2}, p_{d_3}, p_{d_3}, p_{d_4}, p_{d_5}, p_{d_6}, \dots, p_{d_{N_{var}}}] \quad (16)$$

Com o cruzamento aleatório, é obtido o seguinte resultado:

$$prole_1 = [p_{m_1}, p_{m_2}, \uparrow p_{d_3}, p_{d_4}, \uparrow p_{m_5}, p_{m_6}, \dots, p_{m_{N_{var}}}] \quad (17)$$

$$prole_2 = [p_{d_1}, p_{d_2}, \uparrow p_{m_3}, p_{m_4}, \uparrow p_{d_5}, p_{d_6}, \dots, p_{d_{N_{var}}}] \quad (18)$$

Um dos problemas encontrados nesta forma de gerar a descendência, reside no fato que não há inserção de novas informações ‘genéticas’, ou seja, sem diversificação. Nesse escopo, o processo responsável por incluir uma diversidade ‘genética’ é a mutação. Como forma de contornar tal problemática, é utilizado o processo de ‘combinação’, ou do inglês, *blending*. Consiste em combinar os valores dos genitores de uma determinada forma, que seja gerado um valor novo.

$$p_{novo} = \beta p_{mn} + (1 - \beta)p_{dn} \quad (19)$$

sendo

β = número aleatório no intervalo [0,1]

p_{mn} = enésima variável do cromossomo materno

p_{dn} = enésima variável do cromossomo paterno

Outro processo importante neste algoritmo é a mutação. Esta adiciona um grau de diversidade nos cromossomos, evitando que a convergência fique restrita a um mínimo local da superfície, por exemplo. Para tanto, é definido uma quantidade de cromossomos que sofrerão mutação na população. Para realizar o estudo comparativo, foi selecionada a seguinte função custo:

$$f(x,y) = 0{,}5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0{,}5}{1 + 0{,}1(x^2 + y^2)} \quad (20)$$

A função em questão apresenta o mínimo em:

$$f(1{,}897; 1{,}006) = -0{,}5231 \quad (21)$$

E tem o domínio compreendido entre $-\infty \leq x, y \leq \infty$.

A Figura 1 ilustra a função custo na modelagem entre os intervalos $-5 \leq x \leq 5$ e $-5 \leq y \leq 5$.

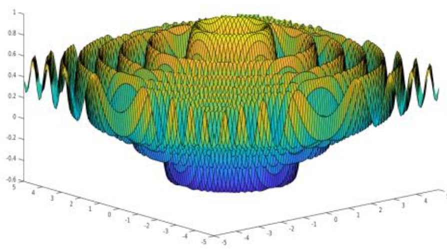

Figura 1. Função custo utilizada na modelgem.
Fonte: Autor

Os algoritmos foram implementados utilizando o soft computacional MATLAB, onde foram obtidos os gráficos de comparativos entre algoritmos. A ilustração da Figura 2, são exibidas as trajetórias de ambos sobre o contorno da função custo. Como ponto inicial de geração da população, foi selecionado o par ordenado x_0 =3,5 e $y_0 = 3{,}5$.

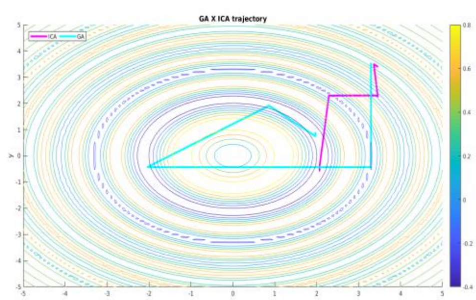

Figura 2. Contorno da função e trajetórias.
Fonte: Autor

Para verificar a evolução da convergência, é apresentado na Figura 3 o gráfico comparativo entre Comparação dos algoritmos AI e AG, o número de iterações e custo mínimo obtido pelos algoritmos. O algoritmo genético obteve o valor mínimo $custo = -0{,}52311$ em 17 iterações, face as 24 do Algoritmo Competitivo Imperialista.

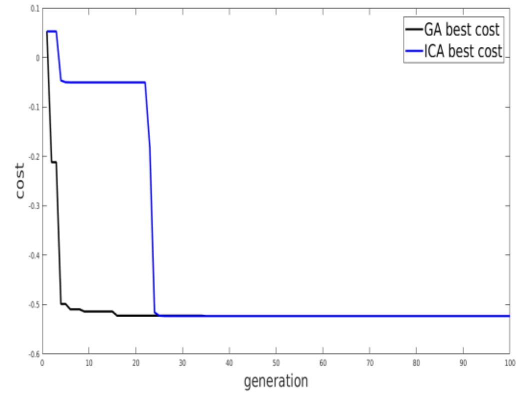

Figura 3. Comparação dos algoritmos AI e AG.
Fonte: Autor

CONCLUSÃO

A modelagem matemática dos dois algoritmos naturais e aplicou em um caso real, pois a otimização de ambos algoritmos envolve operações com matrizes, sem a necessidade de cálculos envolvendo derivadas. A taxa de convergência foi rápida, obtendo a resposta em poucos segundos. Baseado no gráfico comparativo do custo *versus* iterações, no escopo da função custo utilizada, o Algoritmo Competitivo Imperialista se saiu melhor, e com um menor número de iterações obteve o mínimo global da função.

AGRADECIMENTOS

O autor agradece ao GPA pelo suporte computacional.

REFERÊNCIAS

[1] HAUPT, R. L., HAUPT. S. E. *Practical Genetic Algorihtms*, Second Edition, New Jersey: John Wiley & Sons, 2004.

[2] Nelder, J.A, R. Mead. 1965. A Simplex method for function minimization. *ComputJ*. 7.p.308-313,1965.

[3] Schwefel, H. Evolution and Optimum Seeking. New York: Wyley,1995.

[4] Holland, J. H. *Adaptation in Natural and Artificial Systems.*Ann Arbor: University of Michigan Press,1975.

[5] Dorigo, M.; Blum, C. Ant Colony optimization theory. A survey. *Theoretical Computer Science* 344. p. 243 – 278,2005.

[6] Atashpaz-Gargari, E.; Lucas, C. *Imperialist Competitive Algorithm for Optimization Inspired by Imperialistic Competition*. IEEE Congress on Evolutionary Computation.p. 4661 – 4667,2007.