Algoritmos evolutivos para solução de equações não lineares

João Felipe Gobeti Calenzani
Universidade Federal do Espírito Santo
Vitória, Brazil
joao.calenzani@edu.ufes.br

Victor Nascimento Neves
Universidade Federal do Espírito Santo
Vitória, Brazil
victor.n.neves@edu.ufes.br

***Resumo*—Este estudo aborda a aplicação de algoritmos evolutivos para a solução de sistemas de equações não lineares, um problema comum em diversas áreas da engenharia, física, economia e ciências aplicadas. Foram utilizados dois algoritmos evolutivos principais: o Algoritmo Genético (GA) e a Estratégia Evolutiva (ES). Esses algoritmos, inspirados na evolução biológica, empregam processos como seleção, recombinação e mutação para explorar o espaço de soluções de maneira eficiente e robusta. A eficácia dos algoritmos foi avaliada usando dois sistemas de equações não lineares, denominados P1 e P2. A performance dos algoritmos foi analisada em termos de melhor resultado, pior resultado, média, desvio padrão e mediana dos valores de fitness ao longo de 50 execuções independentes. Os resultados mostram que ambos os algoritmos são eficazes na resolução dos sistemas de equações não lineares, com o GA apresentando uma ligeira vantagem em alguns casos.**

***Index Terms*—GA, ES, Equações não lineares, Sistemas de equação**

I. Introdução

(Trabalho prático para a disciplina de Computação Natural do programa de Mestrado do Departamento de Informática - UFES. Semestre 2024/01. Prof. Renato Krohling)

Equações não lineares são aquelas onde a relação entre as variáveis não pode ser expressa como uma combinação linear simples. Em vez disso, elas podem incluir termos como produtos de variáveis, potências, funções exponenciais, logaritmos e outras operações que não preservam a linearidade. Resolver equações não lineares pode ser substancialmente mais complexo do que resolver suas contrapartes lineares devido à presença de múltiplas raízes, comportamentos caóticos e dificuldades de convergência.

Um sistema de equações não lineares consiste em múltiplas equações não lineares que devem ser resolvidas simultaneamente. Esses sistemas são comuns em muitos campos de engenharia, física, economia, e ciências aplicadas, onde os modelos frequentemente envolvem interações complexas entre várias variáveis. Encontrar soluções para esses sistemas pode ser desafiador devido à natureza intrincada das equações envolvidas.

Para abordar a solução de sistemas de equações não lineares, os algoritmos de otimização estocástica, como Algoritmos Genéticos (GA) e Estratégias Evolutivas (ES), são indicados. Esses algoritmos inspirados na evolução biológica utilizam processos como seleção, recombinação e mutação para explorar o espaço de soluções de maneira robusta.

II. Metodologia

A. Otimização de P1 e P2 pelo método da penalização

Para avaliar a eficácia dos algoritmos de otimização, realizamos testes com dois sistemas de equações não lineares. A Figura 1 representa as equações do primeiro sistema, denominado P1, e a Figura 2 representa as equações do segundo sistema, denominado P2.

Resolvemos os sistemas de equações não lineares utilizando algoritmos evolutivos, tratando o problema como uma otimização com restrições, usando o método da penalização [3]. Esse método transforma o problema com restrições em um problema sem restrições, adicionando uma função penalidade ao valor da função objetivo. A função penalidade impõe um custo adicional quando as restrições não são satisfeitas, guiando assim a busca pela solução dentro da região viável. Matematicamente, se temos uma função objetivo $f(\mathbf{x})$ e um conjunto de restrições $g_i(\mathbf{x}) \leq 0$, o problema de otimização pode ser escrito como:

$$\min_{\mathbf{x}} \left(f(\mathbf{x}) + \sum_i \rho \cdot \max(0, g_i(\mathbf{x}))^2 \right)$$

onde ρ é um parâmetro de penalização que determina o peso das violações das restrições. Ao resolver esse problema com algoritmos evolutivos, podemos encontrar soluções aproximadas para os sistemas de equações não lineares de forma eficiente.

Problema P1:

$$\begin{cases} f_1(\mathbf{x}) = 0.8(x_1^2 + x_1 - 1)x_3 + 0.12x_1^2 + 2.16x_1 - 0.12 = 0 \\ f_2(\mathbf{x}) = (1 + x_1^2)x_4 + 0.4x_1^2 - 1.6x_1 - 0.4 = 0 \\ f_3(\mathbf{x}) = (1 + x_1^2)x_5 + x_1^2 - 1 = 0 \\ f_4(\mathbf{x}) = (1 + x_1^2)x_6 + 0.8(x_1^2 + x_1 - 1) = 0 \\ f_5(\mathbf{x}) = x_3x_7 - 0.02x_6 - x_5 - x_3x_4 - 0.16x_4 = 0 \\ f_6(\mathbf{x}) = x_7^2 - 2x_4x_7 + x_6^2 + x_4^2 - x_2^2 = 0 \\ f_7(\mathbf{x}) = x_8 - x_2x_3 = 0 \\ f_8(\mathbf{x}) = 0.0476x_3x_8^{12} + x_3 - 2.104 = 0 \end{cases}$$

Fig. 1: Primeiro Sistema de equações (P1)

Problema P2:

$$\begin{cases} f_1(\mathbf{x}) = x_5 + x_4 - 1.803 = 0 \\ f_2(\mathbf{x}) = (x_2 + x_3)x_5 + 6.19116x_4 - 1.803(1.497 + 0.035) = 0 \\ f_3(\mathbf{x}) = x_6 + x_4 - 0.328 = 0 \\ f_4(\mathbf{x}) = 0.28801x_6 - x_2x_3x_5 = 0 \\ f_5(\mathbf{x}) = (-6.19116x_1 + x_1x_3 + x_2x_5 - x_3x_5)x_6 + x_1x_3x_5 = 0 \\ f_6(\mathbf{x}) = 1.571x_7 + x_4 - 1.803 = 0 \\ f_7(\mathbf{x}) = x_8 - 0.000856x_7^2 = 0 \\ f_8(\mathbf{x}) = (x_5 - x_1)x_9 - x_1x_5 = 0 \\ f_9(\mathbf{x}) = x_9 - 377x_2x_8 = 0 \end{cases}$$

Fig. 2: Segundo Sistema de equações (P2)

B. Algoritmo Genético (GA)

Algoritmos Genéticos são métodos de otimização inspirados nos princípios da seleção natural e genética [1]. Eles representam soluções potenciais como indivíduos em uma população e iterativamente aplicam operadores genéticos para evoluir essas soluções. Os principais componentes de um GA incluem:

- Seleção: Escolha dos indivíduos mais aptos para reprodução.
- Recombinação (Crossover): Combinação de pares de indivíduos para criar novos descendentes.
- Mutação: Pequenas alterações aleatórias nos indivíduos para introduzir diversidade.

GAs são particularmente úteis para problemas de otimização complexos e não lineares devido à sua capacidade de escapar de mínimos locais e explorar vastos espaços de solução.

C. Estratégias Evolutivas (ES)

Estratégias Evolutivas são um tipo de algoritmo evolutivo que enfatiza mutações e seleções mais do que recombinações [4]. Em uma ES, uma população de soluções é iterativamente modificada por mutações gaussianas e a seleção é baseada na aptidão dos indivíduos. A ES é eficiente para otimização de parâmetros contínuos e é conhecida por sua robustez e simplicidade.

A aplicação de algoritmos como GA e ES para resolver sistemas de equações não lineares é de grande importância devido à natureza complicada e multifacetada desses problemas. Esses algoritmos não fazem suposições sobre a linearidade ou continuidade das funções envolvidas, permitindo-lhes lidar com uma ampla gama de problemas não lineares complexos de forma eficaz.

D. Escolha de Hiperparâmetros

Nesta seção, detalhamos os hiperparâmetros utilizados nos Algoritmos Genéticos (GA) e nas Estratégias Evolutivas (ES) para a solução dos sistemas de equações não lineares, explicando o significado de cada um e os valores escolhidos para este estudo.

1) Parâmetros do Algoritmo Genético: Os Algoritmos Genéticos utilizam uma série de hiperparâmetros que influenciam diretamente a performance do algoritmo [2]. Os principais hiperparâmetros utilizados neste estudo são:

- **Tamanho da População (pop_size)**: Número de indivíduos na população. Um tamanho maior pode aumentar a diversidade genética, mas também aumenta o custo computacional.
  Valor Utilizado: 50.
- **Número de Gerações (max_gen)**: Quantidade de iterações que o algoritmo executa. Mais gerações permitem uma exploração mais extensa do espaço de soluções.
  Valor Utilizado: 1000.
- **Taxa de Crossover (pc)**: Probabilidade de dois indivíduos cruzarem e trocarem genes. Valores altos incentivam a recombinação genética, enquanto valores baixos favorecem a estabilidade das soluções.
  Valor Utilizado: 0.8.
- **Taxa de Mutação (pm)**: Probabilidade de um gene em um indivíduo sofrer mutação. Mutação ajuda a manter a diversidade genética e evita a convergência prematura .
  Valor Utilizado: 0.1.
- **Sigma de Mutação (sigma)**: Desvio padrão da distribuição normal utilizada para aplicar mutações. Controla a magnitude das mudanças nos genes durante a mutação.
  Valor Utilizado: 0.1.
- **operador de crossover aritmético (alpha)**: Frequentemente utilizado no operador de crossover aritmético. Este operador é um método de recombinação onde os descendentes são gerados a partir de uma média ponderada dos pais. O valor de α controla a proporção em que os genes dos pais contribuem para os descendentes. Considerando P1 e P2 como os pais, temos F1 e F2 como sendo os filhos:

  $$F1 = \alpha \cdot P1 + (1 - \alpha) \cdot P2$$
  $$F2 = (1 - \alpha) \cdot P1 + \alpha \cdot P2$$

  Valor Utilizado: 0.5.
- **Elitismo (elitism)**: Preservação dos melhores indivíduos de uma geração para a próxima, garantindo que a qualidade da solução não diminua.
  Valor Utilizado: True (ativado).

2) Parâmetros da Estratégia Evolutiva: As Estratégias Evolutivas também possuem uma série de hiperparâmetros que são essenciais para o seu funcionamento [4]. Os principais hiperparâmetros utilizados são:

- **Número de Gerações (n_iter)**: Número de iterações que o algoritmo executa. Mais gerações permitem uma busca mais detalhada por soluções ótimas.
  Valor Utilizado: 5000.
- **Tamanho do Passo (step_size)**: Controla a amplitude das mutações aplicadas às soluções. Um tamanho de passo

maior pode facilitar a exploração do espaço de busca, enquanto um passo menor pode melhorar a precisão das soluções.
Valor Utilizado: 0.15

- **Número de Pais (mu)**: Número de melhores indivíduos selecionados para gerar a próxima geração. Um maior número de pais pode aumentar a diversidade genética, mas também aumenta a complexidade computacional.
Valor Utilizado: 20.
- **Número de Filhos (lambda)**: Número de descendentes gerados a partir dos pais selecionados. Um maior número de filhos pode permitir uma melhor exploração do espaço de busca.
Valor Utilizado: 100.

Os hiperparâmetros selecionados para ambos os algoritmos foram escolhidos com base em experimentações prévias e na literatura existente, visando alcançar um equilíbrio entre diversidade genética, precisão na solução e custo computacional.

III. Resultados

Neste estudo, aplicamos um Algoritmo Genético (GA) e uma Estratégia Evolutiva (ES) para resolver os sistema de equações não lineares P1 e P2. Para cada sistema de equações, realizamos 50 execuções independentes do GA e do ES, denominadas épocas, registrando o histórico de fitness ao longo das gerações para cada execução. Um resumo dos resultados das estratégias está apresentado na Tabela I. Para os sistemas P1 e P2, e para as execuções dos algoritmos GA (Algoritmo Genético) e ES (Estratégia Evolutiva), são fornecidos os seguintes dados: melhor e pior resultado, média, desvio padrão e mediana de todas as épocas.

TABELA I: Resultados

Sistema P1			
GA		**ES**	
Estatística	**Valor**	**Estatística**	**Valor**
Melhor	0.074284	Melhor	0.176835
Pior	1.259605	Pior	0.253844
Média	0.617314	Média	0.211438
Desvio Padrão	0.376377	Desvio Padrão	0.016379
Mediana	0.641347	Mediana	0.211622
Sistema P2			
GA		**ES**	
Estatística	**Valor**	**Estatística**	**Valor**
Melhor	0.002112	Melhor	0.017299
Pior	0.537605	Pior	0.068579
Média	0.04693	Média	0.041201
Desvio Padrão	0.110061	Desvio Padrão	0.011815
Mediana	0.012548	Mediana	0.039265

As Figuras 3, 4, 5 e 6 ilustram a evolução das sucessivas épocas em cada rodada dos algoritmos GA e ES. Especificamente, elas mostram a evolução do GA para o P1, do ES para o P1, do GA para o P2 e do ES para o P2, respectivamente. A iteração com o melhor resultado está destacada em vermelho, enquanto as demais épocas, que apresentaram resultados inferiores, estão em azul. Os gráficos estão apresentados em escala logarítmica para melhor visualização dos dados.

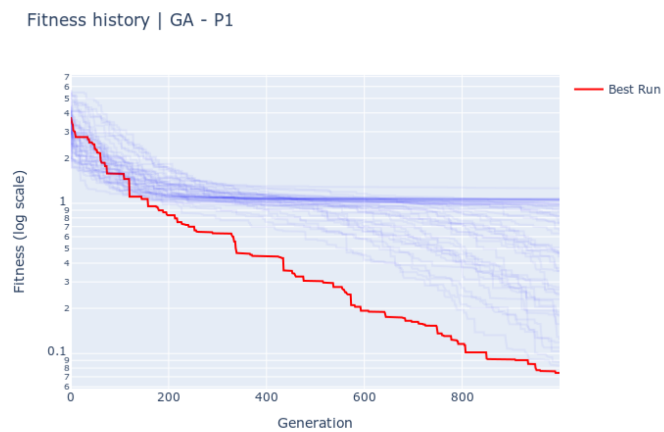

Fig. 3: Gráfico de fitness para P1 utilizando GA

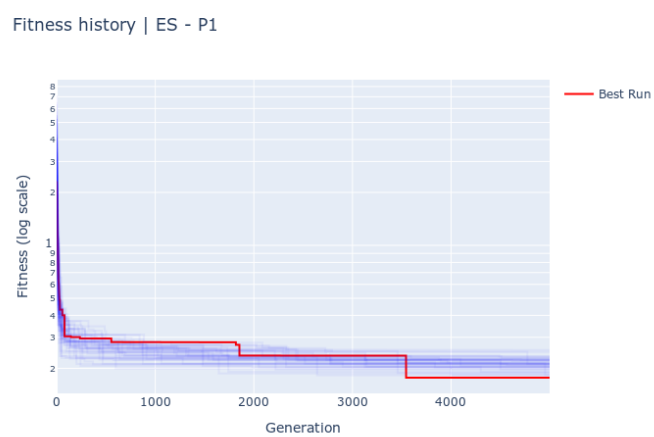

Fig. 4: Gráfico de fitness para P1 utilizando ES

IV. Conclusão

Neste trabalho, investigamos a aplicação de Algoritmos Genéticos (GA) e Estratégias Evolutivas (ES) para a solução de sistemas de equações não lineares. Os resultados demonstram que ambos os algoritmos são capazes de encontrar soluções próximas a zero, indicando uma boa aproximação para resolver os sistemas propostos. O uso de uma escala logarítmica nos gráficos de fitness permitiu uma melhor visualização das variações nos valores baixos de fitness, proporcionando uma compreensão mais clara do desempenho dos algoritmos. A análise dos hiperparâmetros revelou a importância de uma escolha adequada para equilibrar a diversidade genética, a precisão das soluções e o custo computacional. Em termos de performance, o GA mostrou-se

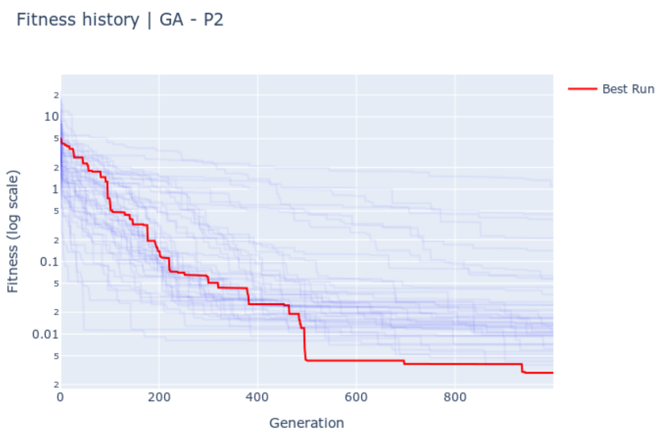

Fig. 5: Gráfico de fitness para P2 utilizando GA

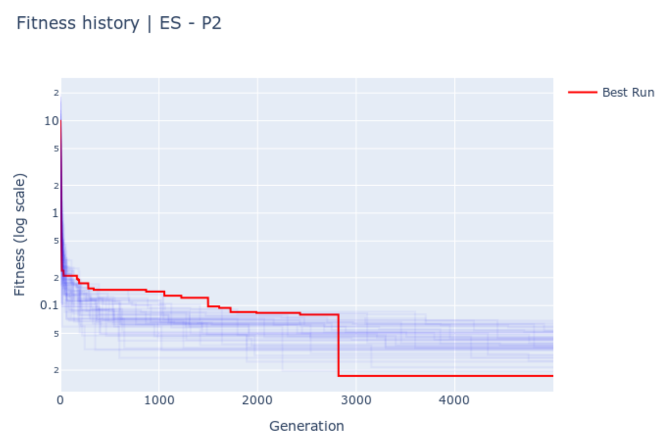

Fig. 6: Gráfico de fitness para P2 utilizando ES

ligeiramente superior em alguns casos, embora o ES também tenha apresentado resultados competitivos. Futuramente, estudos adicionais podem explorar variações nos hiperparâmetros e a combinação de diferentes técnicas evolutivas para melhorar ainda mais a eficácia desses algoritmos na resolução de sistemas de equações não lineares complexos.

Referências

[1] Zbigniew Michalewicz. *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media, 2013.

[2] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, 4(1):1–32, 1996.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

[4] Hans-Paul Schwefel. *Evolution and Optimum Seeking*. Wiley, 1995.