

# Aprendizagem Supervisionada na Predição de Curvas de Valores Fitness em Algoritmos Genéticos

Renua Meireles Almeida<sup>1</sup>, Otávio Noura Teixeira<sup>1</sup> (Orientador)

<sup>1</sup>Faculdade de Engenharia de Computação – Universidade Federal do Pará (UFPA)  
Tucuruí – PA – Brasil

{renua.meireles, onoura}@gmail.com

**Abstract.** *This paper continues the efforts of a recent approach that proposed the prediction of fitness value curves in the evolutionary process of Genetic Algorithms (GAs). The current work proposes to deepen and suggest new methods and environments in order to understand, among other aspects, the influences of the size and quality of the dataset, as well as the characteristics of the problems used. It was found that the new methods contributed to a more accurate understanding, allowing to highlight some difficulties that the model is susceptible to, mainly caused by little executions data available for training when the problem solved by the GA does have many local minimums.*

**Resumo.** *O presente artigo dá continuidade nos esforços de uma abordagem recente, a qual propôs a predição de curvas de valores fitness no processo evolutivo dos Algoritmos Genéticos (AGs). O trabalho atual propõe se aprofundar, sugerir novos métodos e ambientes com objetivo de entender, dentre outros aspectos, as influências do tamanho e da qualidade do conjunto de dados bem como as características dos problemas utilizados. Constatou-se que os novos métodos contribuíram para um entendimento mais preciso desta abordagem, permitindo evidenciar algumas dificuldades que o modelo está suscetível, principalmente causadas em cenários com poucas execuções disponíveis para treinamento durante a resolução de problemas com vários mínimos locais.*

## 1. Introdução

Os AGs, propostos inicialmente em [Holland 1975], são métodos estocásticos de busca e otimização inspirados pela Teoria da Evolução de Darwin [Goldberg 1989], onde indivíduos, representando soluções para um dado problema, interagem entre si através de algumas operações definidas, gerando novos indivíduos que carregam as características genéticas dos pais para as próximas gerações.

No seu formato mais simplificado [Goldberg 1989], os AGs possuem três operações: Reprodução, onde pares de indivíduos são selecionados de acordo com sua aptidão (fitness); Cruzamento entre indivíduos, que geram novos indivíduos para a próxima geração com as características genéticas dos pais; Por fim, os indivíduos gerados, possuem uma pequena chance de passar por mutação, que é uma pequena perturbação em seu material genético [Mitchell 1998].

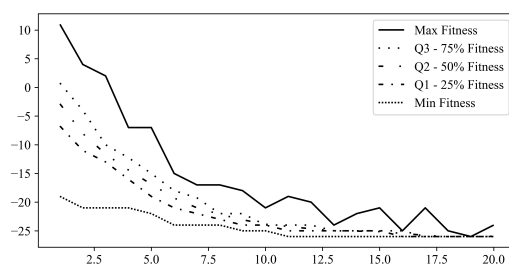
Para a execução de um AG, é necessário definir alguns parâmetros básicos, mais precisamente, o tamanho da população, os percentuais de cruzamento e mutação,

e um critério de parada, onde para este, apesar de existirem algumas alternativas [Safe et al. 2004], comumente pode ser definido por um número fixo de gerações. E a performance do algoritmo possui fortes relações com a definição desses elementos [Mitchell 1998, Mosayebi and Sodhi 2020].

Da mesma forma, o elitismo, apresentado em [Jong 1975], se tornou um outro parâmetro de inicialização comum que pode ser definido, este, consiste em garantir que um determinado número de melhores indivíduos sejam mantidos para a próxima geração.

Uma forma simplificada de inspecionar o processo evolutivo de um AG, é registrar, tipicamente os valores máximo, médio e mínimo de fitness dos indivíduos em cada geração, e com isso, é possível gerar um gráfico de linhas apresentando os valores fitness por geração [Almeida et al. 2021b, Almeida et al. 2021a].

Semelhantemente, é possível expandir essa visão ao adicionar outros aspectos dos valores fitness, como a variância ou até mesmo os quartis, que conseguem apresentar uma visão mais detalhada da distribuição dos valores fitness pelas gerações. A exemplo, a Figura 1 apresenta uma situação para um problema minimização, onde os valores fitness convergem para um valor mínimo local ou global.



**Figura 1. Exemplo de gráfico Fitness x Geração.**

Devido à sua estocasticidade, AGs geralmente são executados diversas vezes sob os mesmos parâmetros definidos, isto para conseguir inferir com mais confiabilidade acerca do comportamento e dos resultados obtidos [Mitchell 1998]. Contudo, usualmente não ocorre a reutilização de dados mais internos que são gerados no decorrer das execuções. Consequentemente, trabalhos que utilizam a mesma versão de um AG em problemas já utilizados em outros experimentos, acabam percorrendo um mesmo processo exaustivo de execuções repetidas.

Tratando-se de reutilização de dados, seja para detecção de padrões, aprimoramento de experiência, dentre outros. É possível de adotar o contexto de modelos de aprendizagem de máquina, e, em casos em que um conjunto de dados já está vinculado diretamente aos resultados que foram obtidos, se torna viável a utilização de modelos supervisionados de aprendizagem de máquina. Nos quais o modelo consegue inferir resultados de novas experiências com base no conhecimento adquirido em experiências passadas [Norvig and Russell 2013].

Formalmente, de acordo com [Norvig and Russell 2013], a tarefa de um modelo supervisionado é, dado um conjunto de dados de treinamento de  $N$  pares conhecidos  $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ , onde  $Y_i$  foi gerado por uma função desconhecida  $y = f(x)$ , encontrar a função  $h$ , que se aproxima da função original  $f$ . E, ao final do

treinamento, um conjunto de teste, distinto do conjunto de treinamento, é utilizado para avaliar sua performance. O par  $(X, Y)$  pode ser referenciado também como features e targets do modelo.

Este trabalho, propõe um método para construção de um conjunto de dados para tornar possível a reutilização das experiências nas execuções dos AGs, e com isso, treinar um modelo supervisionado para prever diferentes aspectos no comportamento dos valores fitness durante o processo evolutivo de um AG, e isto, com a utilização de parâmetros de inicialização, bem como alguns parâmetros característicos dos problemas.

Esta é uma versão continuada de um trabalho anterior [Almeida et al. 2021a], o qual introduziu esta abordagem com alguns resultados promissores, porém, voltado à um cenário específico. Agora, foi proposta uma análise expandida dos resultados, sob contexto de um conjunto de problemas definidos em [Jong 1975], bem como uma apresentação de novos métodos e uma análise de performance sobre efeitos da quantidade de dados e do problema utilizado. Por fim, também houve a consideração de novos aspectos dos valores fitness para o conjunto de variáveis nos targets.

O conteúdo do artigo foi organizado da seguinte forma: A Seção 2 aborda alguns aspectos importantes de trabalhos anteriores nessa linha, enquanto que a Seção 3 apresenta um detalhamento em subseções percorrendo cada aspecto do ambiente criado e utilizado para os experimentos. Na Seção 4 são realizadas análises de performance do modelo em diferentes aspectos sob as situações definidas. Por fim, a Seção 5 enfatiza algumas percepções alcançadas com as análises durante a Seção 4, contribuições ao trabalho original e recomendações para trabalhos futuros também são realizadas.

## 2. TRABALHOS CORRELATOS

Inicialmente, esta abordagem foi sugerida em [Almeida et al. 2021b], porém sem experimentações suficientes para gerar análises com base nos resultados, logo depois, [Almeida et al. 2021a] apresentou uma continuidade nos esforços, desta vez em experimentações em um problema específico, uma versão da função *Schwefel* com 30 dimensões.

Estes experimentos, se focaram na ideia inicial em predição dos valores máximo, médio e mínimo de fitness, e isto foi realizado utilizando o modelo Floresta Aleatória, que foi treinado com um número fixo de execuções de AG, e então, sendo capaz de prever o comportamento das curvas de valores fitness das execuções futuras ao receber apenas os parâmetros de inicialização destas.

Um ponto de atenção no trabalho anterior, se dá no método utilizado para o treinamento do modelo, o qual consistiu em coletar uma porcentagem fixa de um conjunto de dados grande, onde a performance do modelo foi mensurada pelo Erro Médio Absoluto (MAE) sobre os registros restantes do conjunto de dados, sendo este conjunto, constituído por registros de 1000 execuções da AG na resolução do problema dado.

Contudo, observa-se que esta abordagem não contempla cenários prováveis. Cenários estes, caracterizados por inicialmente o modelo não possuir acesso a um conjunto de dados favorável, ou até mesmo, uma situação em que o modelo é aprimorado conforme as execuções ocorrem partindo de um ponto em que não exista nem mesmo uma execução inicial. Estas e outras questões se tornam foco no trabalho atual.

### 3. METODOLOGIA

Da mesma forma que em [Almeida et al. 2021a], foi necessário cobrir uma série de processos, de modo a alcançar o objetivo definido, onde alguns destes são novos e outros sofreram adaptações. Tendo início pela definição da estrutura do AG a ser utilizado, a qual foi levemente adaptada, seguindo para a escolha, desta vez, de um conjunto de problemas ao invés de um único como anteriormente. Em continuidade, o modelo de aprendizagem foi definido, juntamente com as features e targets, onde neste ponto, foram propostas mudanças especificamente nos targets.

Devido à inexistência de uma base de dados pública com esses dados relacionados ao processo interno dos AGs, foi adotado a criação de um específico para a utilização nas experimentações deste trabalho, seguindo o mesmo modelo sugerido no trabalho original, porém com a coleta de mais uma informação, para possibilitar a organização do conjunto de dados de maneira que possibilite outras formas de avaliação do modelo. Por último, foram definidas novas estratégias de avaliação do modelo que contemplem as necessidades do trabalho.

Um detalhamento de cada um dos processos é realizado nas subseções a seguir.

#### 3.1. Algoritmo Genético

A estrutura da versão de um AG simples [Goldberg 1989] foi utilizada com algumas mudanças. A codificação do indivíduo é binária, os indivíduos são selecionados pelo método de torneio, com definição de três participantes e uma rodada. Foi utilizado o cruzamento de dois pontos como método, bem como o *swap* como método para mutação. Por fim, elitismo é uma opção, e a reposição dos novos indivíduos na população é aleatória.

Segue abaixo uma relação dos parâmetros necessários para a inicialização do AG descrito:

- $P$ : valor inteiro que define o tamanho fixo da população;
- $G$ : valor inteiro que define a geração de parada;
- $C_p$ : valor decimal que define a probabilidade de cruzamento;
- $M_p$ : valor decimal que define a probabilidade de mutação;
- $E$ : valor inteiro que define o número de elitistas.

#### 3.2. Definição dos Problemas

As formas do conjunto de teste de De Jong foram escolhidas para compor os problemas a serem resolvidos pelo AG nos experimentos. Este conjunto de problemas é composto por funções de benchmark de valor real com diferentes características, sendo todas elas, problemas de minimização.

Todas as funções  $f_i$  do conjunto de problemas utilizam representação binária ( $x_{base_2}$ ) dando suporte para as operações do AG, com isso, alguns parâmetros precisam ser definidos com objetivo de decodificar a representação binária em uma representação decimal ( $x_{base_{10}}$ ) e escalada entre os limites da função ( $x_{decodificado}$ ) para suportar os cálculos das funções sobre os indivíduos, onde o valor fitness dos indivíduos é o próprio valor da função na versão decodificada e escalada destes  $f_i(x_{decodificado})$ . Para isso, cada função temos:

- $a$ : O limite inferior do domínio;

- $b$ : O limite superior do domínio;
- $D$ : Quantidade de dimensões da função;
- $n$ : Quantidade de bits por dimensão;
- $S = 2^{n \cdot D}$ : Soluções possíveis;
- $\Delta x = \frac{b-a}{2^n}$ : Fator de resolução (Espaçamento entre as soluções possíveis para uma dimensão);
- $x_{decodificado} = (\Delta x \cdot x_{base_{10}}) + a$ :  $x_{base_{2}}$  decodificada e escalada.

As funções são definidas da seguinte forma:

### 3.2.1. Parábola ( $f_1$ )

$$f_1(x) = \sum_{i=1}^{10} x_i^2 \quad (1)$$

Sujeito a  $a = -5.12$ ,  $b = 5.12$ ,  $D = 10$ ,  $n = 10$ ,  $\Delta x = 0.01$ ,  $S = 10^{30}$ ,  $\min(f_1) = f_1(x_{opt}) = 0$ ,  $x_{opt} = [0, 0, \dots, 0]$ , e  $\max(f_1) = f_1(x_{fmax}) = 262.1$ ,  $x_{fmax} = [\pm 5.12, \pm 5.12, \dots, \pm 5.12]$ .

### 3.2.2. Rosenbrock's Saddle Function ( $f_2$ )

$$f_2(x) = 100 \cdot (x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (2)$$

Sujeito a  $a = -2.048$ ,  $b = 2.048$ ,  $D = 2$ ,  $n = 12$ ,  $\Delta x = 0.001$ ,  $S = 1.7 \times 10^6$ ,  $\min(f_2) = f_2(x_{opt}) = 0$ ,  $x_{opt} = [1, 1]$ , e  $\max(f_2) = f_2(x_{fmax}) = 3905.9$ ,  $x_{fmax} = [-2.048, -2.048]$ .

### 3.2.3. Step Function ( $f_3$ )

$$f_3(x) = \sum_{i=1}^5 [x_i] \quad (3)$$

Onde  $[x_i]$  representa o maior inteiro menor ou igual a  $x_i$ .

Sujeito a  $a = -5.12$ ,  $b = 5.12$ ,  $D = 5$ ,  $n = 10$ ,  $\Delta x = 0.01$ ,  $S = 10^{15}$ ,  $\min(f_3) = f_3(x_{opt}) = -30$ ,  $x_{opt} = [-5.12, -5.12]$ , e  $\max(f_3) = f_3(x_{fmax}) = 25$ ,  $x_{fmax} = [5.12, 5.12]$ .

### 3.2.4. Quadratic Function with Gaussian Noise ( $f_4$ )

$$f_4(x) = \sum_{i=1}^{30} ix^4 + GAUSS(\mu = 0, \sigma = 1) \quad (4)$$

Sujeito a  $a = -1.28$ ,  $b = 1.28$ ,  $D = 30$ ,  $n = 8$ ,  $\Delta x = 0.01$ ,  $S = 10^{72}$ ,  $\min(f_4) = f_4(x_{opt}) = 0$ ,  $x_{opt} = [0, 0, \dots, 0]$ , e  $\max(f_4) = f_4(x_{fmax}) = 1248.2$ ,  $x_{fmax} = [\pm 1.28, \pm 1.28, \dots, \pm 1.28]$ .

### 3.2.5. Shekel's Foxhole Function ( $f_5$ )

$$\frac{1}{f_5(x)} = \frac{1}{k} + \sum_{j=1}^{25} \frac{1}{f_j} \quad (5)$$

Onde,

$$f_j = c_j + \sum_{i=1}^2 (x_i - A_{ij})^6;$$

$$[A_{ij}] = \begin{bmatrix} -32, & -16, & 0, & 16, & 32, & -32, & \dots, & 16, & 32 \\ -32, & -32, & -32, & -32, & -32, & -16, & \dots, & 32, & 32 \end{bmatrix};$$

$$[c_j] = [1, 2, 3, \dots, 25];$$

e  $k = 500$ .

Sujeito a  $a = -65.536$ ,  $b = 65.536$ ,  $D = 2$ ,  $n = 17$ ,  $\Delta x = 0.001$ ,  $S = 16^9$ ,  $\min(f_5) = f_5(x_{opt}) = 1$ ,  $x_{opt} = [-32, -32]$ , e  $\max(f_5) = k = 500$ .

Esta é uma função multimodal interessante, ela possui 25 mínimos locais aproximadamente nos pontos  $\{A_{1j}, A_{2j}\}_{j=1}^{25}$ , e com os valores destes mínimos dados por  $x = (A_{1j}, A_{2j}) \cong c_j$ .

### 3.3. Features e Targets do Modelo

Devido aos resultados positivos nos trabalhos anteriores, o modelo de Floresta Aleatória [Breiman 2001] foi escolhido. Seguindo para a definição das features e targets, como a abordagem do trabalho sugere, os parâmetros de inicialização devem se fazer presentes nas features, em complemento, a geração atual foi definida também sendo capaz de oferecer ao modelo uma referência às gerações específicas, para que a predição possa ser realizada em qualquer ponto do processo evolutivo. Por fim, foram definidos alguns parâmetros do próprio problema que contribuem com uma referência direta às características do problema. A Tabela 1 relaciona todas as features e seus respectivos tipos de dados.

**Tabela 1. Features do Modelo.**

$x_i$	Feature	Tipo
$x_1$	Geração Atual	Inteiro
$x_2$	Tamanho da População	Inteiro
$x_3$	Geração de Parada	Inteiro
$x_4$	Probabilidade de Cruzamento	Decimal
$x_5$	Probabilidade de Mutação	Decimal
$x_6$	Elitismo	Inteiro
$x_7$	Dimensão	Inteiro
$x_8$	Limite inferior do Domínio	Decimal
$x_9$	Limite superior do Domínio	Decimal

Indo além do que foi adotado nos trabalhos anteriores, o modelo possui a tarefa de inferir cinco métricas de valores fitness diferentes. A Tabela 2 apresenta a relação dos targets e seus respectivos tipos de dados.

**Tabela 2. Targets do Modelo.**

$y_i$	Target	Tipo
$y_1$	Valor Máximo de Fitness	Decimal
$y_2$	Quartil Q3 (75%) dos valores Fitness	Decimal
$y_3$	Quartil Q2 (50%) dos valores Fitness	Decimal
$y_4$	Quartil Q1 (25%) dos valores Fitness	Decimal
$y_5$	Valor mínimo de Fitness	Decimal

### 3.4. Conjunto de Dados

Para este processo, em essência, seguiu-se de acordo com o sugerido no trabalho original, realizando a coleta de dados gerados da execução de diversas instâncias de AG. Para cada função, 1000 execuções foram definidas no total, onde em cada, os parâmetros de inicialização foram uniformemente randomizados sob uma restrição de intervalos predefinidos, que também foram mantido de acordo com a Tabela 3.

**Tabela 3. Restrição de intervalos para os parâmetros de inicialização.**

Parâmetro	Intervalo
Tamanho da População	$50 \leq x_2 < 200$
Geração de Parada	$50 \leq x_3 < 500$
Probabilidade de Cruzamento	$0.70 \leq x_4 < 1.00$
Probabilidade de Mutação	$0.00 \leq x_5 < 0.05$
Elitismo	$0 \leq x_6 < 10$

Após cada execução de instância de AG,  $G$  registros são adicionados ao conjunto de dados, onde  $G = x_3$ . E, no decorrer do processo evolutivo, todos os parâmetros de inicialização são mantidos constantes, ou seja, em uma mesma execução apenas a geração atual ( $x_1$ ) e os targets se alteram. Além disso, as features  $x_7$ ,  $x_8$  e  $x_9$  se mantêm constantes e são específicas de acordo com o problema.

Ressalta-se que o modelo não considera o contexto de execução, suas predições possuem referência apenas de uma geração específica. Desta forma, um elemento importante a ser coletado nos registros do conjunto de dados, se dá por um identificador das execuções. dessa forma, é possível coletar amostras do conjunto de dados que correspondem à mesma execução, isto se torna um facilitador ao analisar o número de execuções necessárias que levaram o modelo à determinadas performances.

### 3.5. Treinamento e Avaliação do Modelo

O método original definido nos trabalhos anteriores consideram que uma porcentagem  $p$  fixa de **registros aleatórios** seja selecionada para treinamento do modelo, enquanto que o restante seja utilizado para avaliação da performance do modelo, este formato considera a posse de um conjunto grande de dados.

Agora, é proposto que seja garantido que **todos os registros** de  $N_{treino}$  execuções sejam selecionados para o treinamento do modelo, igualmente, todos os registros de  $N_{teste}$  execuções para o conjunto de teste. Este método é apresentado como uma alternativa para o cenário onde o modelo precise ser treinado a medida que novas execuções ocorram.

Diante disso, o modelo será avaliado em duas perspectivas, a primeira corresponde à performance geral do modelo (PGM), e a segunda à performance isolada do modelo (PIM).

**PGM:** considera o quão bem o modelo realiza as predições em um conjunto de dados de teste qualquer, em essência, representa o valor MAE sobre os registros testados independente da quantidade ou das execuções da precedência dos mesmos.

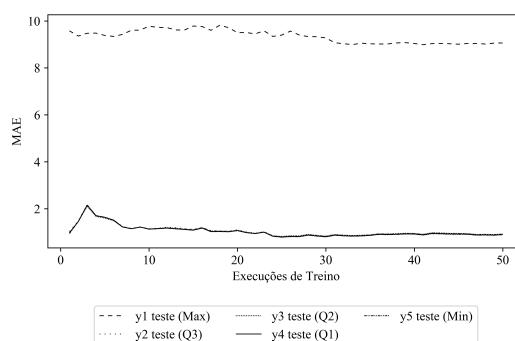
**PIM:** considera a utilização da métrica de erro MAE para avaliar a performance do modelo em uma única execução específica. Com isso, é possível obter uma visão analítica, a qual leva em consideração o quão próximo o modelo chegou em prever o comportamento das curvas no decorrer do processo evolutivo em uma execução. Para isso, foi definido um gráfico de linha comparativo, no mesmo formato apresentado na Figura 1, mas neste momento com a presença dos valores fitness previstos ao lado dos valores reais.

A análise da performance do modelo também considera o efeito na mudança do tamanho do conjunto de dados. Em resumo, para cada problema, seguiu-se os seguintes passos:

- 01: Isolar uma amostra para teste com 100 execuções e seus respectivos registros da base ( $N_{teste} = 100$ );
- 02: Treinar o modelo, inicialmente com dados de uma execução ( $N_{treino} = 1$ ), e então,  $N_{treino}$  é incrementado em um até que o modelo não consiga obter melhorias significativas;
- 03: Análise PGM;
- 04: Análise PIM em execuções de interesse;

#### 4. RESULTADOS E DISCUSSÕES

A Figura 2 apresenta a PGM sobre conjunto de teste para a função  $f_1$ . O primeiro ponto que se evidencia está no comportamento consideravelmente semelhante no processo de aprendizagem do modelo para os targets  $y_2, y_3, y_4$  e  $y_5$ , ao mesmo passo em que se diferem consideravelmente do target  $y_1$ . Por conta disso, o target  $y_1$  é desconsiderado nas análises gráficas comparativas aos outros targets.



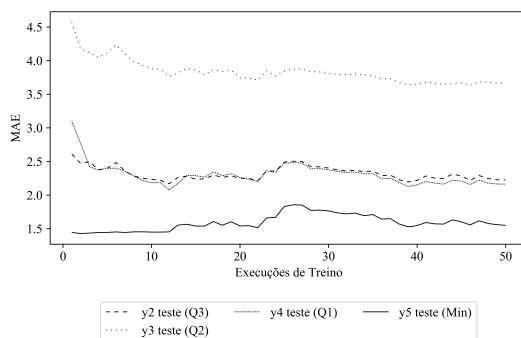
**Figura 2. PGM x  $N_{treino}$  para  $f_1$ .**

Como no trabalho original, o modelo de fato teve dificuldades no processo de predição dos valores fitness dos piores indivíduos ( $y_1$ ), isto ocorre principalmente por conta da alta instabilidade no padrão seguido pelos valores de geração em geração.

Um outro fator que se faz evidente pode ser verificado na alta PGM mesmo com o modelo sendo treinado com uma única execução  $N_{treino} = 1$ , não apenas isto, mesmo com o aumento no número de execuções para treinamento, não há ganhos consideráveis na PGM.

Neste caso em específico, é de se esperar esse tipo de comportamento por conta das características da função  $f_1$ , sendo simétrica, e não possuindo mínimos locais, as curvas de valores fitness tendem a ser similares mesmo com alterações nos parâmetros de inicialização. Logo, este comportamento também se repetiu para as outras funções que possuem essa mesma característica.

Contudo, nota-se uma pequena diferença quando se trata da função  $f_5$ , com exceção do target ( $y_5$ ) modelo tende a obter melhorias com o aumento das amostras de treino, sendo esta, uma função que possui vários ótimos locais, mais comumente o AG terá comportamentos diferentes a cada execução, dessa forma ao receber novos exemplos, o modelo tende a obter melhorias de performance, como pode ser observado na Figura 3.



**Figura 3. PGM x  $N_{treino}$  para  $f_5$ .**

De forma geral, fica nítido a superioridade do modelo ao prever os valores fitness do target  $y_5$ , isso pode ser evidenciado em todas as funções. As Tabelas 4 e 5 apresentam a PGM para todos os targets para as funções  $f_1$  e  $f_5$  respectivamente depois de uma e quarenta execuções de treinamento.

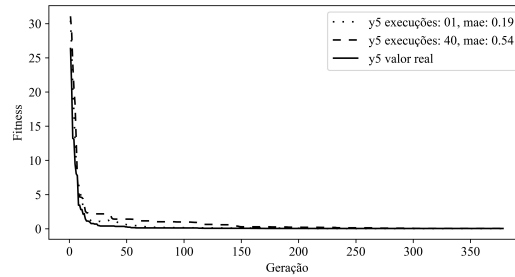
**Tabela 4. PGM depois de uma e quarenta execuções de treinamento para  $f_1$ .**

Target	$N_{treino} = 1$	$N_{treino} = 40$
$y_1$	9.58	9.04
$y_2$	1.03	0.94
$y_3$	0.97	0.95
$y_4$	0.97	0.92
$y_5$	0.94	0.91

Na figura 4 pode ser visto a PIM para o target  $y_5$  como referência, visto que é alvo das previsões mais assertivas do modelo, depois de uma e quarenta execuções de treinamento, em comparação lado a lado dos valores fitness reais gerados na execução do AG para a função  $f_1$ , onde a execução em questão possui identificador 390 ( $x_2 = 134$ ,  $x_3 = 379$ ,  $x_4 = 0.8214$ ,  $x_5 = 0.0283$ ,  $x_6 = 4$ ).

**Tabela 5. PGM depois de uma e quarenta execuções de treinamento para  $f_5$ .**

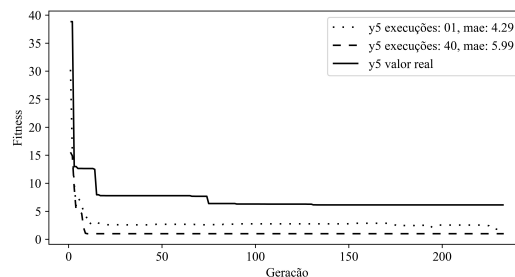
Target	$N_{treino} = 1$	$N_{treino} = 40$
$y_1$	193.28	180.49
$y_2$	2.61	2.21
$y_3$	4.57	3.65
$y_4$	3.09	2.15
$y_5$	1.44	1.54



**Figura 4. PIM depois de uma e quarenta execuções de treinamento para  $f_1$  e target  $y_5$ .**

Devido à sua simplicidade, a função  $f_1$  torna o processo de aprendizagem do modelo bastante favorável, o que não é muito realista, porém se torna eficaz ao utilizar como referência em quesitos comparativos à outras funções.

Em suma, o modelo consegue performar bem nas funções unimodais, todavia, quando os resultados da função  $f_5$  são levados em consideração, percebe-se que a PIM começa a decair a medida que o AG gera diferentes comportamentos nas curvas de valores fitness, comportamentos dados tipicamente por convergência para mínimos locais ou diferentes velocidades de convergência. A Figura 5 mostra a PIM em uma situação onde o AG convergiu para um mínimo local que se encontra aproximadamente no valor 6, isto, para a execução com identificador 190 ( $x_2 = 62$ ,  $x_3 = 233$ ,  $x_4 = 0.7345$ ,  $x_5 = 0.0072$ ,  $x_6 = 5$ ).

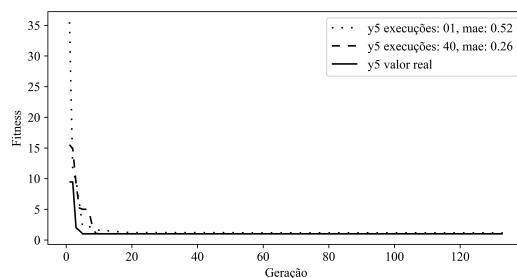


**Figura 5. PIM depois de uma e quarenta execuções de treinamento para  $f_5$  e target  $y_5$ .**

Neste ponto, detecta-se uma grande limitação nas previsões do modelo, quando as execuções de treinamento são limitadas e não apresentam possíveis comportamentos que as curvas podem adotar, o modelo possui a forte tendência em memorizar os comportamentos apresentados, se tornando não flexível ao prever outros possíveis comportamentos

das curvas de valores fitness que podem surgir no processo evolutivo.

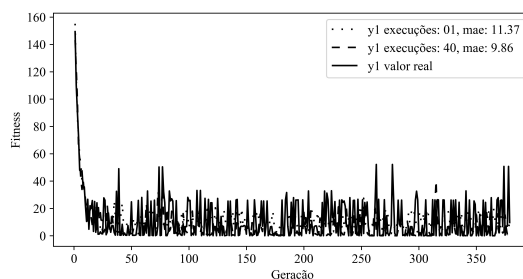
Para confirmar isto, é possível analisar a Figura 6, a qual apresenta a PIM para a função  $f_5$ , sob contexto de uma execução que convergiu para o mínimo global da função, a execução em questão possui identificador 777 ( $x_2 = 190$ ,  $x_3 = 133$ ,  $x_4 = 0.9324$ ,  $x_5 = 0.0446$ ,  $x_6 = 7$ ). Tendo em vista isso, conseqüentemente conforme a complexidade da função aumenta, o modelo necessita de mais execuções de treinamento para conseguir melhorar sua performance.



**Figura 6. PIM depois de uma e quarenta execuções de treinamento para  $f_5$  e target  $y_5$ .**

Como mencionado anteriormente, o modelo possui dificuldades em prever o target  $y_5$ . A própria busca por se ajustar aos valores de treino, trazem um alto risco, tendo em vista estocasticidade deste target, podendo apresentar mudanças bruscas de geração em geração.

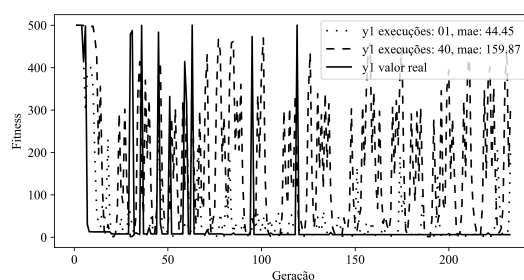
Observa-se com mais precisão esse evento na Figura 7, que apresenta a PIM para  $f_1$  na execução 390, onde mesmo sendo uma função simplificada, se torna um grande desafio para o modelo. Dependendo da função, o resultado pode ser desastroso, e um exemplo desse caso pode ser visto na Figura 8, que contém os resultados da PIM para a função  $f_5$  na execução 190.



**Figura 7. PIM depois de uma e quarenta execuções de treinamento para  $f_1$  e target  $y_1$ .**

## 5. CONCLUSÃO

O presente trabalho alcançou seus objetivos. Contribuições foram realizadas através da melhoria e sugestão de novos métodos. Novos problemas foram investigados, conseqüentemente, novas percepções surgiram, o que permite uma visão mais clara acerca desta abordagem.



**Figura 8. PIM depois de uma e quarenta execuções de treinamento para  $f_5$  e target  $y_1$ .**

O novo método de treinamento sugerido, de fato apresenta o possível comportamento do modelo em uma situação de aplicação real, os problemas e dificuldades que o mesmo poderia se deparar, principalmente causados durante as primeiras execuções que fossem disponibilizadas de alguma forma.

Diante do abordado um questionamento surge, com o comportamento do modelo sendo tão similar para algumas funções com características semelhantes, seria possível a implementação de meta-modelos capazes de serem utilizados para problemas similares? Se sim, em quais aspectos esses problemas deveriam ser similares?

Dessa forma, com a melhoria e aprofundamento dos métodos, se torna mais acessível o caminho para futuras expansões acerca da abordagem proposta, o que implica na solidificação cada vez maior das bases dessa abordagem, proporcionando o entendimento de como um modelo pode se comportar diante de diferentes cenários, aplicado em outras classes de problemas, e/ou outras versões de AG ou classes de algoritmos baseados em população.

## AGRADECIMENTOS

Este trabalho e a pesquisa por trás, não seria possível sem o direcionamento do meu orientador e amigo Otávio Noura Teixeira, bem como o laboratório MEC<sup>2</sup>A e seus integrantes, que foram responsáveis por instigar um ambiente de desenvolvimento acadêmico-científico e evolução pessoal de seus membros, este trabalho é um dos primeiros resultados desses esforços.

Esta pesquisa foi, parcialmente mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPQ, através do Programa Institucional de Bolsas de Iniciação Científica - PIBIC, por meio da Universidade Federal do Pará - UFPA, campus Tucuruí. Eterna gratidão pelo apoio e incentivo.

## Referências

Almeida, R., Rodrigues, R., Ribeiro, D., and Teixeira, O. (2021a). Fitness value curves prediction in the evolutionary process of genetic algorithms applied to benchmark function. In *Anais da I Escola Regional de Alto Desempenho Norte 2 e I Escola Regional de Aprendizado de Máquina e Inteligência Artificial Norte 2*, pages 17–20, Porto Alegre, RS, Brasil. SBC.

- Almeida, R. M., Ribeiro, D. M. F., Rodrigues, R. M., and Teixeira, O. N. (2021b). *Fitness Value Curves Prediction in the Evolutionary Process of Genetic Algorithms*, page 221–222. Association for Computing Machinery, New York, NY, USA.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Goldberg, D. E. (1989). Genetic algorithms in search. *Optimization, and Machine Learning*.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The U. of Michigan Press.
- Jong, K. D. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, USA.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.
- Mosayebi, M. and Sodhi, M. (2020). Tuning genetic algorithm parameters using design of experiments. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20*, page 1937–1944, New York, NY, USA. Association for Computing Machinery.
- Norvig, P. and Russell, S. (2013). *Inteligencia Artificial*. ELSEVIER EDITORA.
- Safe, M., Carballido, J., Ponzoni, I., and Brignole, N. (2004). On stopping criteria for genetic algorithms. In Bazzan, A. L. C. and Labidi, S., editors, *Advances in Artificial Intelligence – SBIA 2004*, pages 405–413, Berlin, Heidelberg. Springer Berlin Heidelberg.