



**Universidade do Estado do Rio de Janeiro**  
Centro de Tecnologia e Ciências  
Instituto de Matemática e Estatística

Filipe Pessoa Sousa

**Modelos de Regressão Simbólica Através de *Biased*  
*Random-Key Genetic Programming* em Aplicações na Física**

Rio de Janeiro

2023

Filipe Pessôa Sousa

**Modelos de Regressão Simbólica Através de *Biased Random-Key Genetic Programming* em Aplicações na Física**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Orientador: Prof. Dr. Igor Machado Coelho  
Orientadora: Prof.<sup>a</sup> Dra. Cristiane Oliveira de Faria

Rio de Janeiro

2023

CATALOGAÇÃO NA FONTE  
UERJ / REDE SIRIUS / BIBLIOTECA CTC/A

---

S725      Sousa, Filipe Pessoa  
Modelos de regressão simbólica através de *Biased Random-Key Genetic Programming* em aplicações na Física / Filipe Pessoa Sousa. - 2023.  
120 f.

Orientadores: Igor Machado Coelho, Cristiane Oliveira de Faria  
Dissertação (Mestrado em Ciências Computacionais) – Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística.

1. Programação heurística - Teses. 2. Otimização matemática - Teses. 3. Algoritmos de computador - Teses. I. Coelho, Igor Machado. II. Faria, Cristiane Oliveira de. III. Universidade do Estado do Rio de Janeiro. Instituto de Matemática e Estatística. IV. Título

CDU 004.023

---

Patricia Bello Meijinhos CRB7/5217 - Bibliotecária responsável pela elaboração da ficha catalográfica

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

---

Assinatura

---

Data

Filipe Pessôa Sousa

**Modelos de Regressão Simbólica Através de *Biased Random-Key Genetic Programming* em Aplicações na Física**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Aprovada em 21 de Dezembro de 2023.

Banca Examinadora:

---

Prof. Dr. Igor Machado Coelho (Orientador)  
Universidade Federal Fluminense

---

Prof.<sup>a</sup> Dra. Cristiane Oliveira de Faria (Orientadora)  
Instituto de Matemática e Estatística - UERJ

---

Prof. Dr. Gustavo Silva Semaan  
Universidade Federal Fluminense

---

Prof. Dr. Fabiano de Souza Oliveira  
Instituto de Matemática e Estatística - UERJ

Rio de Janeiro

2023

## AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que contribuíram de alguma forma para a realização deste trabalho. Cada pessoa que cruzou meu caminho durante essa jornada desempenhou um papel importante em minha trajetória, e sou grato por cada uma delas. Agradeço a Deus por me guiar até aqui e destaco aqueles que foram fundamentais para alcançar este objetivo.

Em primeiro lugar, dedico meus sinceros agradecimentos aos meus orientadores, Igor Machado Coelho, com quem compartilhei dias de trabalho na UFF, agradeço pelo constante suporte, e a Cristiane Oliveira de Faria, que me incentivou a ingressar no programa. Nunca imaginei chegar tão longe, e sou imensamente grato pela orientação e incentivo de ambos. Ambos são exemplos de professores para mim.

Expresso minha gratidão à UERJ por ser uma das melhores universidades do Estado do Rio de Janeiro, proporcionando um ambiente acolhedor, apesar das dificuldades enfrentadas pela educação pública. A UERJ permanece resistente.

Agradeço aos meus pais, Maria Inês de Sousa e Jose Manoel de Sousa, por me inspirarem a alcançar este patamar. Ambos foram fundamentais para que eu pudesse concluir essa jornada, e reconheço que sou uma pessoa privilegiada por contar com o apoio deles.

Um agradecimento especial ao CEASM (Centro de Ações Solidárias da Maré) e ao CPV (Curso Pré-Vestibular), que fez parte da minha trajetória, auxiliando-me a ingressar na universidade e possibilitando que eu contribuísse para que mais pessoas alcançassem essa conquista ao longo de uma década.

À minha companheira, Vanessa Américo da Costa, expresso minha gratidão por sua ajuda e crença em mim. Mesmo quando duvidava de que chegaria até o final dessa jornada, ela sempre me apoiou, permitindo a realização desse feito.

Agradeço ao CCOMP e a todos os professores do programa. Em todas as disciplinas, busquei dar o meu melhor, enfrentando adversidades e superando o medo para me tornar a minha melhor versão.

Aos professores Fabiano de Souza e Gustavo Silva, que fizeram parte da minha banca, agradeço por suas contribuições e conhecimentos valiosos para este trabalho.

Expresso minha gratidão ao Governo Federal pela oportunidade de usufruir de um ensino público de qualidade em uma universidade pública. Em particular, agradeço a CAPES pelo suporte financeiro concedido para a realização deste trabalho.

Por fim, agradeço a todos os amigos e familiares pela paciência nas minhas ausências devido aos estudos, pelo apoio incondicional e pelo carinho demonstrado ao longo dessa jornada.

A natureza é um enorme jogo de xadrez disputado por deuses, e que temos o privilégio de observar. As regras do jogo são o que chamamos de física fundamental, e compreender essas regras é a nossa meta.

*Richard Feynman*

## RESUMO

SOUSA, Filipe Pessoa. *Modelos de Regressão Simbólica Através de Biased Random-Key Genetic Programming em Aplicações na Física*. 2023. 119 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

Este trabalho se concentra na aplicação da meta-heurística *Biased Random-Key Genetic Programming* (BRKGP) no contexto do Problema de Regressão Simbólica. O objetivo é aproximar a forma algébrica de uma função desconhecida a partir de entradas e saídas numéricas. O algoritmo BRKGP, baseado em programação genética, desempenha um papel crucial nesse processo, otimizando a geração de expressões simbólicas. O estudo apresenta resultados preliminares e compara a precisão com diferentes tamanhos de amostras, derivadas de problemas de física. A pesquisa procura explorar a aplicação da BRKGP na regressão simbólica, buscando contribuir para a validação de funções físicas e promover avanços. Destaca-se a importância de aprimoramentos nas técnicas computacionais, evidenciando desafios como indeterminações e números extremos. Resultados notáveis indicam aplicações abrangentes em diversos campos. O estudo conclui ressaltando a eficiência da abordagem inovadora e apontando desafios a serem enfrentados em estudos futuros, incluindo o refinamento de estratégias de tratamento de dados e a eliminação de indeterminações.

Palavras-chave: Meta-heurísticas. Otimização. BRKGP. Regressão Simbólica.

## ABSTRACT

SOUSA, Filipe Pessoa. *Symbolic Regression Models Through the Biased Random-Key Genetic Programming for Physics Applications*. 2023. 119 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

This work focuses on the application of the meta-heuristic Biased Random-Key Genetic Programming (BRKGP) in the context of the Symbolic Regression Problem. The goal is to approximate the algebraic form of an unknown equation based on numerical inputs and outputs. The BRKGP algorithm, based on genetic programming, plays a crucial role in this process by optimizing the generation of symbolic expressions. The study presents preliminary results and compares accuracy with different sample sizes derived from physics problems. The research aims to explore the application of BRKGP in symbolic regression, seeking to contribute to the validation of physical equations and promote advancements. The importance of improvements in computational techniques is emphasized, highlighting challenges such as indeterminacies and extreme numbers. Notable results indicate comprehensive applications in various fields. The study concludes by emphasizing the efficiency of the innovative approach and pointing out challenges to be addressed in future studies, including the refinement of data treatment strategies and the elimination of indeterminacies.

Keywords: Meta-heuristics. Optimization. BRKGP. Symbolic Regression.

## LISTA DE FIGURAS

Figura 1	- Abordagem para encontrar a melhor solução. Adaptado de [1] . . . . .	18
Figura 2	- Abordagens Meta-heurísticas. Adaptado de [2] . . . . .	21
Figura 3	- Funcionamento de um Algoritmos Genético padrão. Retirado de [3] . .	22
Figura 4	- Processo de transição entre gerações consecutivas. Adaptado de [4] . .	25
Figura 5	- Exemplo de Programação Genética. Adaptado de [4] . . . . .	26
Figura 6	- Exemplo de <i>subtree crossover</i> . Adaptado de [4] . . . . .	27
Figura 7	- Exemplo de <i>subtree mutation</i> . Adaptado de [4] . . . . .	27
Figura 8	- Fluxograma da PG. Retirado de [5] . . . . .	28
Figura 9	- Estrutura cromossômica usada para codificar uma <i>Program-Expression</i> com um vetor de chaves aleatórias $h + t + k$ . Adaptado de [4] . . . . .	29
Figura 10	- Arquitetura do BRKGP. Adaptado de [4] . . . . .	30
Figura 11	- Exemplo de Pilha. Retirada [6] . . . . .	35
Figura 12	- Exemplo de Mapeamento dos numerais e das operações. Retirada [6] .	36
Figura 13	- Ilustração esquemática do <i>AI Feynman Algorithm</i> . Retirada [7] . . . .	39
Figura 14	- Gráfico comparativo para a Função do Período Orbital, entre a Regres- são Linear vs O Problema-alvo vs A Solução Encontrada . . . . .	105
Figura 15	- Gráfico comparativo para a Função da Distribuição de Probabilidade, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada	106
Figura 16	- Gráfico comparativo para a Função do Período Orbital, entre a Regres- são Linear vs O Problema-alvo vs A Solução Encontrada . . . . .	107
Figura 17	- Gráfico comparativo para a Função da Distribuição de Probabilidade, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada	108

## LISTA DE TABELAS

Tabela 1	- Influência nos Parâmetros e Descrições. . . . .	50
Tabela 2	- Parâmetros globais do BRKGP. . . . .	50
Tabela 3	- Descrição dos problemas-teste . . . . .	54
Tabela 4	- Constantes Utilizadas no Estudo . . . . .	55
Tabela 5	- Tempo entre 5 execuções para a Distância Euclidiana . . . . .	56
Tabela 6	- RMSE entre 5 execuções para a Distância Euclidiana . . . . .	57
Tabela 7	- Tempo médio (s) entre 5 execuções para a massa relativística . . . . .	59
Tabela 8	- RMSE entre 5 execuções para a massa relativística . . . . .	60
Tabela 9	- Tempo médio (s) entre 5 execuções para as Lentes Delgadas . . . . .	62
Tabela 10	- RMSE entre 5 execuções para as Lentes Delgadas . . . . .	63
Tabela 11	- Tempo médio (s) entre 5 execuções para o Produto Interno . . . . .	64
Tabela 12	- RMSE entre 5 execuções para o Produto Interno . . . . .	65
Tabela 13	- Tempo médio (s) entre 5 execuções para a Força Gravitacional . . . . .	67
Tabela 14	- RMSE entre 5 execuções para a Força Gravitacional . . . . .	68
Tabela 15	- Tempo médio (s) entre 5 execuções para o Período Orbital de dois Corpos	70
Tabela 16	- RMSE entre 5 execuções para o Período Orbital de dois Corpos . . . . .	70
Tabela 17	- Tempo médio (s) entre 5 execuções para o Campo Elétrico Puntiforme	73
Tabela 18	- RMSE entre 5 execuções para o Campo Elétrico Puntiforme . . . . .	73
Tabela 19	- Tempo médio (s) entre 5 execuções para a Função de Densidade de Probabilidade . . . . .	75
Tabela 20	- RMSE entre 5 execuções para a Função de Densidade de Probabilidade	76
Tabela 21	- Tempo médio (s) entre 5 execuções para a Função das Ondas Eletro- magnéticas . . . . .	78
Tabela 22	- RMSE entre 5 execuções para a função das Ondas Eletromagnéticas . .	78
Tabela 23	- Tempo médio (s) entre 5 execuções para a Função de Distribuição de Bose-Einstein . . . . .	80
Tabela 24	- RMSE entre 5 execuções para a função de Distribuição de Bose-Einstein	81
Tabela 25	- Observações sobre as funções . . . . .	82
Tabela 26	- Observações sobre os resultados . . . . .	82
Tabela 27	- RMSE após a implementação da análise da unidade entre 5 execuções para a Distância Euclidiana . . . . .	85
Tabela 28	- Tempo(s) após a implementação da análise da unidade entre 5 execu- ções para a Distância Euclidiana . . . . .	86
Tabela 29	- RMSE após a implementação da análise da unidade entre 5 execuções para a Massa Relativística . . . . .	87

Tabela 30 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Massa Relativística . . . . .	87
Tabela 31 - RMSE após a implementação da análise da unidade entre 5 execuções para a Lentes Delgadas . . . . .	88
Tabela 32 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Lentes Delgadas . . . . .	89
Tabela 33 - RMSE após a implementação da análise da unidade entre 5 execuções para a Produto Interno . . . . .	90
Tabela 34 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Produto Interno . . . . .	90
Tabela 35 - RMSE após a implementação da análise da unidade entre 5 execuções para a Força Gravitacional . . . . .	91
Tabela 36 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Força Gravitacional . . . . .	92
Tabela 37 - RMSE após a implementação da análise da unidade entre 5 execuções para a Período Orbital . . . . .	93
Tabela 38 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Período Orbital . . . . .	94
Tabela 39 - RMSE após a implementação da análise da unidade entre 5 execuções para a Campo Elétrico Puntiforme . . . . .	96
Tabela 40 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Campo Elétrico Puntiforme . . . . .	96
Tabela 41 - RMSE após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas . . . . .	97
Tabela 42 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas . . . . .	98
Tabela 43 - RMSE após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas . . . . .	100
Tabela 44 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas . . . . .	100
Tabela 45 - RMSE após a implementação da análise da unidade entre 5 execuções para a Distribuição de Bose-Einstein . . . . .	102
Tabela 46 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Distribuição de Bose-Einstein . . . . .	102
Tabela 47 - Resultados da Regressão Linear Simples para Diferentes Funções e Amostras . . . . .	104
Tabela 48 - Resultados da Regressão Linear Simples para Diferentes Funções e Amostras . . . . .	104

## LISTA DE ALGORITMOS

Algoritmo 1 - BRKGA. Adaptado de [8]	24
Algoritmo 2 - DecodeHead. Adaptado de [4]	30
Algoritmo 3 - DecodeTail. Adaptado de [4]	31
Algoritmo 4 - BRKGP	49
Algoritmo 5 - Função para identificar qual tipo de operação	51
Algoritmo 6 - Empilhar	51
Algoritmo 7 - Calculo_Indivíduo	52
Algoritmo 8 - Decoder	53

## LISTA DE ABREVIATURAS E SIGLAS

BRKGA	<i>Biased Random-Key Genetic Algorithm</i>
BRKGP	<i>Biased Random-Key Genetic Programming</i>
AG	Algoritmo Genético
PG	Programação Genética
RKGA	<i>Random Key Genetic Algorithm</i>
AGBR	Algoritmos Genéticos Baseados em Regras
RBK	<i>Random Biased Key</i>
PL	Programação Linear
PI	Programação Inteira
PNL	Programação Não-Linear
PE	<i>Program-Expression</i>
RMSE	<i>Root Mean Square Error</i>
IRACE	<i>Iterated Racing for Automatic Configuration of Algorithms</i>
GiNaC	<i>GiNaC is Not a CAS</i>
GMP	<i>GNU Multiple Precision Arithmetic Library</i>
CLN	<i>Class Library for Numbers</i>
EMA	<i>EMpirical Analysis of algorithms</i>

## SUMÁRIO

	INTRODUÇÃO . . . . .	14
1	CONCEITOS TEÓRICOS . . . . .	18
1.1	Otimização . . . . .	18
1.2	Heurísticas . . . . .	19
1.3	Meta-heurística . . . . .	20
1.4	Algoritmo Genético . . . . .	21
1.5	Random-Key Genetic Algorithms . . . . .	23
1.6	Biased Random-Key Genetic Algorithms . . . . .	23
1.7	Programação Genética . . . . .	25
1.8	Biased Random-Key Genetic Programming . . . . .	29
1.9	Paradigma de <i>Less Is More</i> . . . . .	31
2	REGRESSÃO SIMBÓLICA E APLICAÇÕES NA FÍSICA . . . . .	33
2.1	Aplicações na Física . . . . .	33
3	REVISÃO LITERATURA . . . . .	35
3.1	BRKGP com Pilhas . . . . .	35
3.2	EMA - <i>EMpirical Analysis of algorithms</i> . . . . .	36
3.3	Feynman . . . . .	38
4	USO DE BIBLIOTECAS EXTERNAS EM C/C++ . . . . .	41
4.1	CAS – Computer Algebraic System . . . . .	41
4.2	Octave . . . . .	41
4.3	Mathomatic . . . . .	42
4.4	ViennaMath . . . . .	43
4.5	Biblioteca GiNaC em C/C++ . . . . .	44
4.6	OptFrame . . . . .	45
5	METODOLOGIA PROPOSTA . . . . .	46
5.1	Lidando com erros numéricos . . . . .	46
5.2	Kahan Summation . . . . .	47
5.3	Pseudo-código Proposto . . . . .	48
5.4	Decoder . . . . .	50
6	RESULTADOS . . . . .	54
6.1	Distância Euclidiana . . . . .	55
6.2	Massa Relativística . . . . .	58
6.3	Lentes Delgadas . . . . .	61
6.4	Produto Interno . . . . .	63
6.5	Força Gravitacional . . . . .	66
6.6	Período Orbital de dois Corpos . . . . .	69

6.7	<b>Campo Elétrico Puntiforme</b>	71
6.8	<b>Função de Densidade de Probabilidade</b>	74
6.9	<b>Ondas Eletromagnéticas</b>	77
6.10	<b>Distribuição de Bose-Einstein</b>	79
6.11	<b>Discussões sobre o resultado</b>	81
6.12	<b>Teste com as grandezas físicas</b>	83
6.12.1	<u>Distância Euclidiana</u>	85
6.12.2	<u>Massa Relativística</u>	86
6.12.3	<u>Lentes Delgadas</u>	88
6.12.4	<u>Produto Interno</u>	89
6.12.5	<u>Força Gravitacional</u>	91
6.12.6	<u>Período Orbital</u>	93
6.12.7	<u>Campo Elétrico Puntiforme</u>	95
6.12.8	<u>Ondas Eletromagnéticas</u>	97
6.12.9	<u>Análise dos resultados para as grandezas</u>	99
6.13	<b>Implementação de novas funções unárias</b>	99
6.13.1	<u>Função de Densidade de Probabilidade</u>	100
6.13.2	<u>Distribuição de Bose-Einstein</u>	101
6.13.3	<u>Sobre a implementação de novas funções</u>	103
6.14	<b>Regressão Linear Simples</b>	104
6.14.1	<u>Comparação Gráfica</u>	105
6.15	<b>Teste Durbin</b>	108
6.15.1	<u>Amostra de tamanho 10</u>	109
6.15.2	<u>Amostra de tamanho 20</u>	109
6.15.3	<u>Amostra de tamanho 40</u>	110
6.15.4	<u>Amostra de tamanho 80</u>	110
	<b>CONCLUSÕES</b>	111
	<b>REFERÊNCIAS</b>	115

## INTRODUÇÃO

O campo da otimização desempenha um papel crucial em inúmeras disciplinas científicas e aplicações práticas, proporcionando uma base sólida para a busca das melhores soluções em uma ampla variedade de problemas. Desde a otimização de funções matemáticas [9] até a definição de estratégias eficazes em cenários complexos do mundo real [10, 11], a otimização é essencial para melhorar processos, economizar recursos e tomar decisões mais informadas.

Devido à natureza NP-Difícil [12] de diversos desses problemas de otimização clássicos, como o Problema do Caixeiro Viajante [13, 14] que consiste em visitar um conjunto de cidades uma única vez e retornar ao ponto de origem com a menor distância possível, a abordagem por heurísticas tem sido historicamente efetiva na resolução desses problemas em baixo tempo computacional, embora não garantam otimalidade [15]. Extensões modernas desse e outros problemas clássicos continuam sendo estudadas ainda hoje [16], com foco especial no campo das meta-heurísticas [17, 18].

Este trabalho concentra-se na aplicação da meta-heurística *Biased Random-Key Genetic Programming* (BRKGP) [4] em um contexto específico: o Problema de Regressão Simbólica. O objetivo do problema é determinar com precisão a forma algébrica de uma função desconhecida, apenas com entradas e saídas numéricas correspondentes àquela função. O Problema de Regressão Simbólica é relevante em diversos contextos, especialmente na física e em áreas afins, onde a análise de dados e a descoberta de funções representativas são essenciais. Nesse contexto, o algoritmo proposto para resolução da regressão simbólica opera utilizando um conjunto inicial de dados para gerar expressões simbólicas, avaliando e selecionando as melhores representações simbólicas do fenômeno estudado. A meta-heurística BRKGP, baseada em programação genética, desempenha um papel crucial na otimização desse processo, dividindo as soluções em grupos elite e não elite, com base em chaves aleatórias decodificadas.

Como experimentos, buscamos comparar a precisão dos resultados obtidos com diferentes tamanhos de amostras (10, 20, 40 e 80). Vale destacar que as amostras utilizadas são oriundas de problemas de física, com o propósito de validar funções previamente documentadas nos livros de Richard Feynman [19, 20, 21, 22]. A contribuição inovadora deste estudo reside no fato de ser o primeiro trabalho conhecido na literatura a explorar a aplicação da meta-heurística baseada em BRKGP para abordar o problema de regressão simbólica dentro do meio físico. Essa abordagem visa aprimorar nossa compreensão e revelar novos horizontes para a otimização de problemas que incorporam a regressão simbólica.

Resultados preliminares desse estudo indicam possibilidades de aplicações práticas abrangentes, abarcando domínios como a física, saúde e economia, entre outros, incluindo

a própria área de computação. A capacidade de otimizar modelos simbólicos em diversas áreas pode levar a avanços significativos na compreensão e solução de problemas complexos, tornando a pesquisa em BRKGP e Regressão Simbólica um campo promissor e multidisciplinar. Resultados parciais também demonstram a importância de um maior aprofundamento nas técnicas computacionais do campo, dada a natureza altamente desafiadora do problema.

O texto está estruturado em sete capítulos. O primeiro destina-se à apresentação da teoria que abrange o campo da otimização até o BRKGP, incluindo a introdução de conceitos cruciais, como o Paradigma do “*Less Is More*”, que serão explorados nos resultados. O segundo capítulo explora a regressão simbólica e suas aplicações no âmbito da física. O terceiro capítulo oferece uma revisão da literatura pertinente, enquanto o quarto destaca as principais bibliotecas de software investigadas para este estudo. No quinto capítulo, é detalhada a metodologia proposta para a presente pesquisa, apresentando diversas abordagens utilizadas, inclusive por meio de pseudo-códigos. Os resultados são expostos no sexto capítulo.

## MOTIVAÇÃO

A história da descoberta revolucionária da órbita elíptica de Marte por Johannes Kepler em 1601 é marcada por 40 tentativas frustradas realizadas em quatro anos, isso utilizando os melhores tabelas de dados do mundo. Isso destaca a importância de encontrar expressões matemáticas que correspondam com precisão a conjuntos de dados observacionais [7]. Este processo, conhecido como regressão simbólica na forma de  $y = f(x_1, x_2, \dots, x_n)$ , tem desafios particulares quando aplicado a fenômenos físicos complexos, nos quais a relação entre variáveis pode envolver uma função desconhecida ou incluir ruído.

A utilização de ferramentas de regressão simbólica na análise de fenômenos físicos é motivada por sua capacidade única de lidar com conjuntos de dados complexos de maneira interpretável. Em um ambiente onde a compreensão das relações fundamentais é crucial, o uso da regressão simbólica, essa destaca-se como uma abordagem que não apenas ajusta modelos aos dados, mas busca explicitar as expressões matemáticas subjacentes [7, 23, 24].

Os fenômenos físicos passam por um aumento contínuo no tamanho dos conjuntos de dados, principalmente na área da astronomia e física das partículas, onde a coleta de dados é constante. Isso motivou esforços para automatizar tarefas de regressão, principalmente em modelagens mais simples, como a regressão linear. No entanto, o problema geral de regressão simbólica permanece desafiador devido à sua natureza combinatorial, especialmente quando o espaço de busca de expressões simbólicas é exponencialmente grande no número de variáveis. Além disso, em um ambiente onde grandes conjuntos de

dados são comuns, a eficiência computacional da regressão simbólica é uma vantagem significativa. A geração de modelos simbólicos pode simplificar a representação matemática, tornando-a mais gerenciável e facilitando a extração de informações úteis.

Existe uma dificuldade em lidar com um espaço de busca vasto, o que é comum em diversos problemas complexos. Para superar essa limitação, algoritmos genéticos [18] são aplicados com sucesso em buscas direcionadas em espaços exponencialmente grandes, substituindo abordagens de força bruta por estratégias inspiradas na biologia, como mutação, seleção, herança e recombinação. Uma alternativa promissora para abordar problemas de espaço exponencialmente grande é a utilização da programação genética [25], uma técnica que se destaca por sua eficácia na busca de soluções em espaços de busca complexos. Ao contrário dos algoritmos genéticos convencionais, a programação genética não se limita a manipular parâmetros, mas opera diretamente sobre estruturas simbólicas.

A aplicação da programação genética na regressão simbólica no contexto da física apresenta-se como uma estratégia robusta e flexível. Ao enfrentar desafios relacionados à modelagem de fenômenos físicos complexos, a programação genética destaca-se por sua capacidade única de explorar soluções em forma de expressões simbólicas, aproximando-se de maneira mais precisa da realidade física.

## Objetivo

### Objetivo Geral

O objetivo geral deste estudo visa validar um algoritmo de otimização que combina regressão simbólica com a meta-heurística BRKGP em aplicações dentro do campo da Física. A intenção principal é aprimorar o algoritmo, tornando-o capaz de identificar funções físicas previamente validadas de forma eficiente e simplificada, contribuindo assim para a otimização de processos na área. Nesse contexto, busca-se desenvolver uma ferramenta robusta e precisa que possa ser utilizada, facilitando a identificação e validação de funções fundamentais em diversos cenários científicos e aplicados.

### Objetivos específicos

- Realizar uma revisão abrangente da literatura, explorando técnicas de otimização, regressão simbólica e meta-heurísticas como BRKGA, com foco na sua aplicação específica ao contexto da Física, a fim de embasar o desenvolvimento da pesquisa;
- Desenvolver e aprimorar um algoritmo de regressão simbólica direcionado à identi-

ficação eficiente e simplificada de funções físicas previamente validadas;

- Implementar em C++ o algoritmo de programação genética BRKGP, explorando seu potencial para encontrar soluções ótimas para funções físicas complexas, simplificando ao máximo o processo com base na estratégia de “Less Is More”;
- Conduzir experimentos computacionais rigorosos, avaliando o desempenho do método proposto em termos de precisão na identificação de funções físicas, eficiência computacional e eficácia;
- Realizar uma comparação detalhada dos resultados obtidos com o desempenho de outras abordagens existentes no campo da otimização de funções físicas;
- Analisar minuciosamente os resultados, destacando as conquistas e limitações do método proposto em um contexto acadêmico, fornecendo uma avaliação crítica de sua utilidade na otimização de problemas físicos;
- Discutir as contribuições significativas deste estudo para o avanço da pesquisa em otimização de funções físicas e identificar possíveis áreas de pesquisa futura que possam aprimorar ainda mais o método desenvolvido.

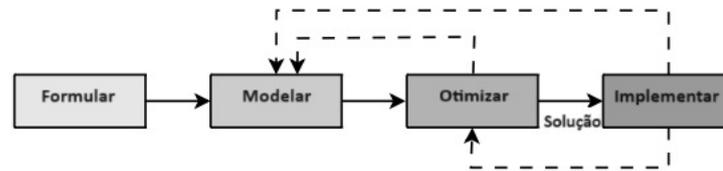
Com a consecução desses objetivos específicos, espera-se atingir o objetivo geral de desenvolver um método de otimização altamente eficaz e eficiente, que integre a regressão simbólica e a meta-heurística BRKGP em aplicações no campo da Física. O propósito final é aprimorar substancialmente o algoritmo, permitindo que ele identifique funções físicas validadas de maneira mais eficiente e simplificada, contribuindo, assim, para a otimização de processos no domínio da Física. Nesse contexto, almeja-se criar uma ferramenta robusta e precisa, simplificando a identificação e a validação de funções fundamentais em diversas situações científicas e aplicadas. Paralelamente, pretende-se avaliar meticulosamente o desempenho do método proposto por meio de experimentos computacionais rigorosos, solidificando seu potencial como uma abordagem valiosa para a otimização de problemas físicos, contribuindo, assim, para o avanço das pesquisas nesse campo.

# 1 CONCEITOS TEÓRICOS

## 1.1 Otimização

A otimização tem sido uma disciplina essencial na área da ciência da computação, sendo voltada para a busca da melhor solução em determinados contextos. Seu principal objetivo é maximizar ou minimizar uma função objetivo, considerando um conjunto de restrições. A otimização possui ampla aplicação em diversas áreas, como engenharia [26], logística [27], economia [28], aprendizado de máquina [29] e inteligência artificial [30]. Uma aplicação amplamente reconhecida na área de otimização consiste em determinar o caminho mais curto que requer o menor consumo de recursos. Um exemplo clássico desse tipo de problema é o “problema do caixeiro-viajante” [14]. No contexto desse desafio, cada rota entre os pontos tem um custo associado, e o objetivo é percorrer cada ponto procurando minimizar o consumo total de recursos. Uma abordagem que pode ser utilizada para verificar o melhor caminho pode ser ilustrado na Figura 1

Figura 1 - Abordagem para encontrar a melhor solução.  
Adaptado de [1]



São encontradas várias abordagens e técnicas para resolver problemas de otimização. Dentre as mais conhecidas estão os métodos de programação linear (PL), programação inteira (PI), programação não-linear (PNL), busca em vizinhança, algoritmos genéticos (AG), entre outras meta-heurísticas [31, 32]. Cada método tem suas próprias particularidades e é adequado para diferentes tipos de problemas. Em especial, meta-heurísticas se diferem das heurísticas clássicas e das técnicas de busca local, notadamente por sua capacidade de escapar de ótimos locais de baixa qualidade, em busca de um ótimo global, e também capacidade adaptação entre diversos problemas [32].

Uma das meta-heurísticas amplamente utilizadas é o algoritmo genético. Inspirado no processo de evolução por seleção natural, esse método busca soluções ótimas por meio de uma abordagem evolutiva. Ele começa com uma população de soluções selecionadas, avaliadas com base na função objetivo. Em seguida, ocorrem processos de seleção, cruzamento e mutação para gerar novas soluções. Esse ciclo se repete até que uma solução satisfatória seja encontrada ou um critério de parada, seja atingido.

A literatura acadêmica oferece uma vasta quantidade de referências sobre otimiza-

ção [33, 34, 31, 35]. A otimização é uma área com constantes evoluções, e é impulsionada pela demanda por soluções mais eficientes e precisas. Atualmente, com o desenvolvimento das tecnologias computacionais, novos métodos e abordagens são elaboradas, como otimização baseada em aprendizado de máquina e algoritmos de otimização estocástica [29]. Essas pesquisas procuram aprimorar a capacidade de obter resultados diante de problemas complexos e encontrar soluções cada vez melhores.

Em suma, o processo de otimização é essencial na minimização de fatores como tempo, custo e risco, ou na maximização de variáveis como lucro, qualidade e eficiência [1]. A otimização também desempenha um papel crucial na ciência da computação, fornecendo ferramentas e técnicas para resolver problemas desafiadores. Seja no campo acadêmico ou na indústria, a aplicação da otimização possibilita aprimorar a eficiência, a qualidade e a tomada de decisões em diversos domínios. É uma área em constante progresso, que continua a impulsionar avanços significativos na computação e em outras áreas de conhecimento.

## 1.2 Heurísticas

Os algoritmos heurísticos, também conhecidos como heurísticas de otimização, são métodos que buscam encontrar soluções aproximadas para problemas de otimização. Essas abordagens surgem como uma resposta à limitação de resolver de forma eficiente diversos problemas de otimização que se enquadram na classe NP-difícil. Existem diferentes classes de métodos heurísticos, essas classes se dividem em três categorias principais, as heurísticas construtivas, as de busca local e as meta-heurísticas. [36]

O cerne da heurística reside na sua capacidade de explorar espaços de busca complexos, como aqueles frequentemente encontrados em problemas de otimização combinatória, de forma eficiente e eficaz. As heurísticas operam sob a premissa de que é possível encontrar soluções aceitáveis em tempo razoável, embora não seja garantido que essas soluções sejam as ótimas. Assim, essas abordagens buscam compromissos entre a qualidade da solução e a utilização de recursos computacionais, tornando-as particularmente valiosas em cenários práticos.

A eficácia das heurísticas é frequentemente avaliada em termos de qualidade da solução encontrada, tempo de execução e consumo de recursos computacionais. Experimentos computacionais rigorosos são conduzidos para validar o desempenho dessas abordagens.

Seu estudo e aplicação constituem um campo de pesquisa fundamental que continua a evoluir, contribuindo para uma variedade de aplicações práticas em diversas disciplinas acadêmicas e setores industriais. A evolução das heurísticas para as meta-heurísticas marcou um avanço significativo na resolução de problemas de otimização complexos. En-

quanto as heurísticas eram aplicáveis apenas a problemas específicos, as meta-heurísticas introduziram uma abordagem mais flexível e eficaz, permitindo o equilíbrio dinâmico entre diversificação e intensificação na busca por soluções de alta qualidade. Esse refinamento na estratégia de busca permitiu lidar com mais facilidade a problemas complexos e NP-difíceis de forma mais eficiente [37].

### 1.3 Meta-heurística

Os estudos acerca das Meta-heurísticas ganhou intensidade na década de 1980, no sentido de proporcionar uma estrutura teórica e com caráter mais geral, gerando mais flexibilidade no modelo. Esses algoritmos de otimização, que buscam encontrar soluções aproximadas para problemas de alta complexidade computacional, são frequentemente utilizados em problemas de grande escala e que apresentam múltiplos ótimos locais, sendo aplicáveis em diversas áreas do conhecimento, como engenharia, finanças, ciências da computação, entre outras [32].

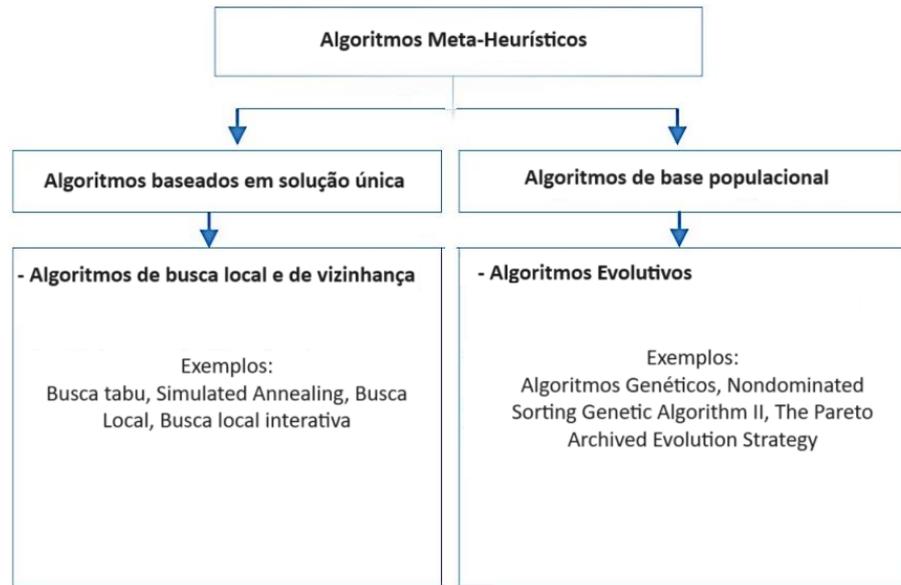
As meta-heurísticas são baseadas em heurísticas, são estratégias para solucionar problemas de maneira eficiente e que não garantem a obtenção da solução ótima, embora consigam escapar de ótimos locais de baixa qualidade. Dessa forma, as meta-heurísticas utilizam essas heurísticas como base para realizar buscas sistemáticas em espaços de solução complexos, visando encontrar soluções que atendam a determinados critérios de otimização. Podemos, classificar as meta-heurísticas em dois tipos, as de trajetória e a busca populacional [32].

As meta-heurísticas de trajetória tipicamente fazem um processo de busca local, visando a exploração do espaço de soluções e realizado mediante movimentos (operações sobre vizinhanças), os quais são aplicados a cada passo sobre a solução corrente, resultando em outra solução promissora em sua vizinhança. Já as principais meta-heurísticas utilizadas atualmente estão o algoritmo genético [38, 39], o simulated annealing [40, 41], o particle swarm optimization [42, 43], o ant colony optimization [44], o tabu search [45], entre outros [46]. Na figura 2 estão ilustradas alguns métodos classificados como Meta-heurísticas. Cada uma dessas meta-heurísticas apresenta vantagens e limitações específicas, sendo escolhida conforme as características do problema a ser resolvido.

O funcionamento das meta-heurísticas pode ser descrito como um processo iterativo, em que a cada iteração uma nova solução é gerada e avaliada de acordo com uma função objetivo previamente definida. Com base na avaliação da solução, a meta-heurística decide se mantém ou descarta a solução, ou ainda se realiza alguma operação de mutação, ou recombinação com outras soluções já avaliadas [47].

O objetivo final das meta-heurísticas é encontrar uma solução que otimize a função objetivo de forma satisfatória, mesmo que essa solução não seja necessariamente a solução

Figura 2 - Abordagens Meta-heurísticas. Adaptado de [2]



ótima do problema. Por isso, as meta-heurísticas são consideradas soluções aproximadas e podem ser utilizadas em situações em que a obtenção da solução ótima é inviável ou demanda um tempo computacional muito elevado.

Em resumo, as meta-heurísticas são algoritmos de otimização que utilizam estratégias heurísticas para buscar soluções aproximadas em espaços de solução complexos. Esses algoritmos são aplicáveis em diversas áreas do conhecimento e apresentam um funcionamento iterativo, em que a cada iteração uma nova solução é gerada e avaliada de acordo com uma função objetivo previamente definida. O objetivo final é encontrar uma solução que otimize a função objetivo de forma satisfatória, mesmo que não seja a solução ótima do problema [32].

#### 1.4 Algoritmo Genético

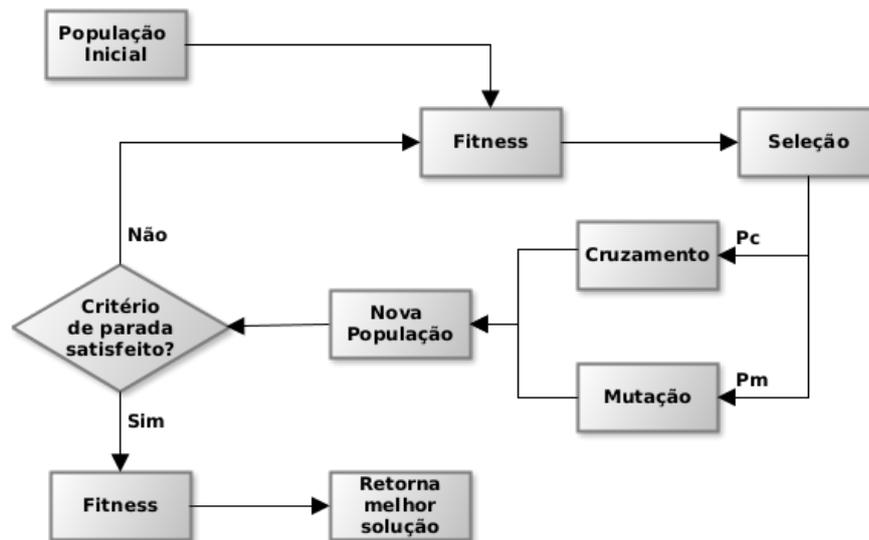
Os algoritmos genéticos (AGs) são uma técnica de otimização inspirada no processo de evolução biológica. Eles têm sido amplamente utilizados em problemas complexos, nos quais é difícil encontrar soluções ótimas utilizando métodos tradicionais. O princípio fundamental dos AGs é simular a evolução natural, aplicando operadores genéticos (seleção, cruzamento/*crossover* e mutação) em uma população de soluções candidatas. Essas soluções são representadas por indivíduos, que possuem um conjunto de genes que codificam uma possível solução para o problema [32].

Os Algoritmos Genéticos (AGs) oferecem diversas vantagens na resolução de problemas de otimização complexos. Eles são capazes de explorar amplamente o espaço de

busca, lidando com alta dimensionalidade e não linearidade. Os AGs permitem a avaliação independente de conjuntos de parâmetros, aproveitando o paralelismo implícito nas cadeias de bits de tamanho variável. Além disso, são numericamente robustos, convergindo em situações em que outras estratégias falham. Sua aplicação abrange áreas como planejamento de rotas, design de sistemas e otimização de parâmetros [48].

Na Figura 3, é apresentado o funcionamento do algoritmo genético, delineando o processo passo a passo até atingir o critério de parada. Em cada iteração, conhecida como geração, os melhores indivíduos são escolhidos com base em uma função de aptidão (fitness). Operadores genéticos, como cruzamento e mutação, são então aplicados aos indivíduos selecionados, com o objetivo de gerar novos descendentes a partir do material genético de seus progenitores. A probabilidade de realizar cada operação ( $P_m$  – probabilidade de mutação e  $P_c$  – probabilidade de cruzamento) influencia a aplicação desses operadores durante o processo evolutivo.

Figura 3 - Funcionamento de um Algoritmos Genético padrão. Retirado de [3]



Diversos estudos têm sido realizados para aprimorar os AGs e adaptá-los a diferentes cenários. Por exemplo, a utilização de estratégias de seleção sofisticadas, como a seleção por torneio ou a seleção por roleta, pode aumentar a eficiência do algoritmo [49]. Além disso, o uso de técnicas de elitismo, que preservam as melhores soluções encontradas ao longo das gerações, pode acelerar a convergência para a solução ótima [50].

Existem também variantes dos AGs, como os Algoritmos Genéticos Baseados em Regras (AGBR), que utilizam regras de produção para representar as soluções [51]. Essa abordagem é particularmente útil em problemas nos quais as soluções podem ser descritas por um conjunto de regras condicionais.

O aperfeiçoamento dos algoritmos genéticos para o RKGA (Random Key Genetic Algorithm) representa um avanço importante na otimização [52]. O RKGA utiliza uma

representação de chave aleatória, permitindo maior flexibilidade e eficiência computacional através de um processo de decodificação, enquanto abstrai os demais componentes complexos de um algoritmo genético clássico.

### 1.5 Random-Key Genetic Algorithms

Os algoritmos genéticos baseados em chaves aleatórias (RKGA, do inglês “*Random-Key Genetic Algorithms*”) são uma abordagem eficaz para resolver problemas de otimização combinatória. Esses algoritmos utilizam uma representação cromossômica baseada em chaves aleatórias, em que cada chave é mapeada para uma solução do problema em questão [52]. Uma das principais características do RKGA é a utilização de operadores genéticos previamente definidos, como crossover e mutação, para explorar e combinar as soluções (representadas como chaves aleatórias). Esses operadores permitem a criação de novas soluções a partir das soluções existentes, promovendo a diversidade e a busca por soluções ótimas. O RKGA também emprega uma estratégia de seleção por elitismo, que determina quais soluções serão mantidas e reproduzidas para a próxima geração. Para garantir diversidade, é gerada uma população de mutantes aleatórios a cada geração.

### 1.6 Biased Random-Key Genetic Algorithms

Os algoritmos genéticos baseados em chaves aleatórias enviesadas (BRKGA, do inglês “*Biased Random-Key Genetic Algorithms*”) são uma variante dos algoritmos genéticos que se destacam pela sua capacidade de convergência mais rápida, comparado ao RKGA. Assim como o RKGA, o BRKGA utiliza uma representação cromossômica baseada em chaves aleatórias, mapeadas para uma solução viável do problema em questão por meio de um processo de decodificação. A escolha de genes no cruzamento do BRKGA tende a favorecer os pais de elite, caracterizando assim o enviesamento de chaves que dá o seu nome. O diagrama geral do algoritmo do BRKGA é apresentado na Figura 4.

A construção de soluções iniciais no BRKGA é feita por meio de geração aleatória de valores reais entre 0 e 1, restando ao usuário decidir a quantidade de chaves aleatórias, bem como o processo de decodificação que mapeia um vetor de chaves em uma solução para seu problema. Além disso, o BRKGA utiliza uma estratégia de elitismo, preservando as melhores soluções encontradas ao longo das gerações. Essas soluções de elite são combinadas com outras soluções por meio dos operadores genéticos, como crossover e mutação, a fim de explorar o espaço de busca de maneira mais eficiente. O Algoritmo 1 detalha o BRKGA.

Os BRKGAs têm sido aplicados em uma variedade de problemas complexos, como

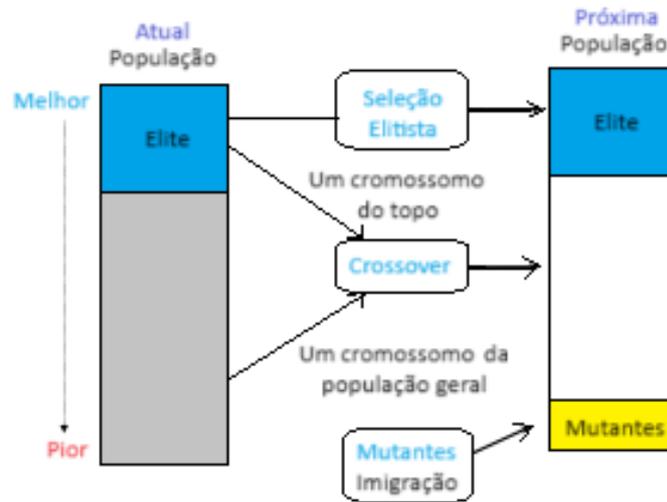
Algoritmo 1 - BRKGA. Adaptado de [8]

```

procedure BRKGA( $|P|$ ,  $|Pe|$ ,  $|Pm|$ ,  $n$ ,  $\rho$ )
  Initialize the value of the best found solution:  $f^* \leftarrow \infty$ 
  Generate the population  $P$  with vectors of  $n$  random keys
  while stopping criterion is not satisfied do
    Decode and evaluate the cost of each new solution in  $P$ 
    Partition  $P$  into two sets:  $Pe$  and  $Pe^c$ 
    Initialize the population of the next generation:  $P^+ \leftarrow Pe$ 
    Generate the set of mutants  $Pm$ , each mutant with  $n$  random keys
    Add  $Pm$  to the population of the next generation:  $P^+ \leftarrow P^+ \cup Pm$ 
    for  $i$  from 1 to  $|P| - |Pe| - |Pm|$  do
      Choose parent  $a$  randomly from  $Pe$ 
      Choose parent  $b$  randomly from  $Pe^c$ 
      for each  $j$  from 1 to  $n$  do
        Toss a biased coin with probability  $\rho > 0.5$  of landing heads
        if coin toss results in heads then
           $c[j] \leftarrow a[j]$ 
        else
           $c[j] \leftarrow b[j]$ 
        end if
      end for
      Add the child  $c$  to the population of the next generation:  $P^+ \leftarrow P^+ \cup \{c\}$ 
    end for
    Update the population:  $P \leftarrow P^+$ 
    Find the best solution  $\chi^+$  in  $P$ :  $\chi^+ \leftarrow \arg \min\{f(\chi) \mid \chi \in P\}$ 
    if  $f(\chi^+) < f^*$  then
       $\chi^* \leftarrow \chi^+$ 
       $f^* \leftarrow f(\chi^+)$ 
    end if
  end while
  return  $\chi^*$ 
end procedure

```

Figura 4 - Processo de transição entre gerações consecutivas.  
Adaptado de [4]



o problema do caixeiro viajante [53], o problema de programação de horários[52] e o problema de roteamento de veículos[54]. Eles têm se mostrado eficazes na busca de soluções de alta qualidade e têm sido amplamente utilizados em diferentes áreas de pesquisa e aplicação.

## 1.7 Programação Genética

A Programação Genética (PG) emerge como uma abordagem inovadora dentro do vasto campo dos Algoritmos Evolucionários, inspirada no princípio darwiniano da evolução e na genética. Assim como a evolução natural molda espécies para sobreviver em ambientes dinâmicos, os Algoritmos Evolucionários, incluindo a Programação Genética, visam otimizar o desempenho de soluções em potencial para resolver problemas específicos [55].

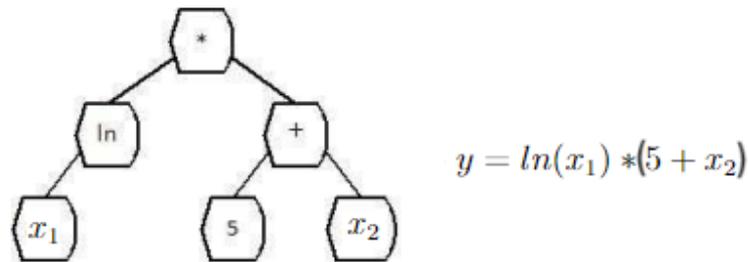
Os principais Algoritmos Evolucionários, tais como Algoritmos Genéticos, Programação Genética, Estratégias Evolutivas e Programação Evolutiva, operam seguindo o princípio da evolução. Dentre esses, a Programação Genética se destaca para problemas que envolvem expressões algébricas, permitindo que computadores resolvam problemas sem necessidade de programação explícita [25]. Esta abordagem parte de declarações de alto nível sobre o que precisa ser feito, gerando automaticamente programas de computador para solucionar problemas, caracterizando-se como “programação automática” [56].

A Programação Genética tem sido amplamente adotada em diversas áreas de pesquisa, demonstrando utilidade em cenários específicos, como, por exemplo, na área de reparação de bugs [57], para resolver problemas de classificação [58] ou na área de regres-

são simbólica fazendo uso de programação genética [59].

Diferentemente da programação convencional, que utiliza linhas de código, a Programação Genética adota representações baseadas em árvores sintáticas. Essas árvores sintáticas, compostas por terminais (variáveis e constantes) e funções (nós internos), permitem a aplicação eficiente do operador de cruzamento em genótipos de tamanhos variáveis [25]. Uma árvore sintática típica é exemplificada na Figura 5, representando a função  $y = \ln(x_1) * (5 + x_2)$

Figura 5 - Exemplo de Programação Genética. Adaptado de [4]



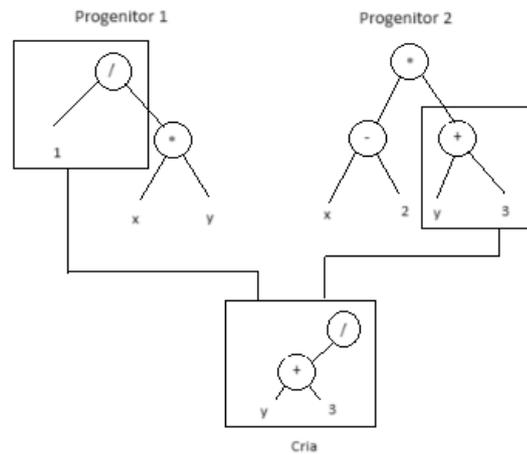
A variabilidade nas soluções é crucial para a eficácia da Programação Genética, sendo a inicialização aleatória das árvores um ponto-chave. Métodos como *full* (onde todas as folhas da árvore são expandidas até uma profundidade máxima pré-definida) e *grow* (onde as folhas da árvore são expandidas aleatoriamente até uma profundidade máxima pré-definida) são comuns, cada um com suas próprias características. O método *ramped half and half* introduz uma combinação dessas abordagens, utilizando metade da população com o método *full* e a outra metade com o método *grow*. Essa estratégia visa garantir uma gama mais ampla de tamanhos e formas para as árvores na população inicial, combinando a uniformidade do método *full* com a diversidade de tamanhos do método *grow*. [60].

A seleção de indivíduos na Programação Genética é realizada probabilisticamente com base na aptidão, usando frequentemente o método de torneio. Esse método não apenas beneficia os mais aptos, mas também mantém a diversidade do conjunto de soluções, preservando a busca por soluções inovadoras [60].

A Programação Genética dá ênfase ao operador de cruzamento, sendo o *subtree crossover*, Figura 6, o mais utilizado. Este método escolhe aleatoriamente um ponto de cruzamento em dois pais, combinando suas *subtrees* para criar descendentes. O *subtree mutation*, Figura 7, é comum para introduzir variação, substituindo uma *subtrees* por outra gerada aleatoriamente[60].

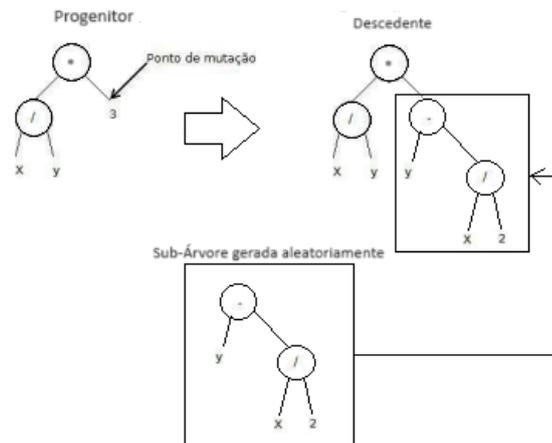
A Programação Genética destaca-se como uma ferramenta versátil em ambientes onde outras abordagens podem falhar. Sua habilidade em descobrir soluções simbólicas para problemas complexos, aliada à sua capacidade de manipular estruturas de código de forma eficaz, a torna uma escolha valiosa para uma variedade de aplicações acadêmicas

Figura 6 - Exemplo de *subtree crossover*. Adaptado de [4]



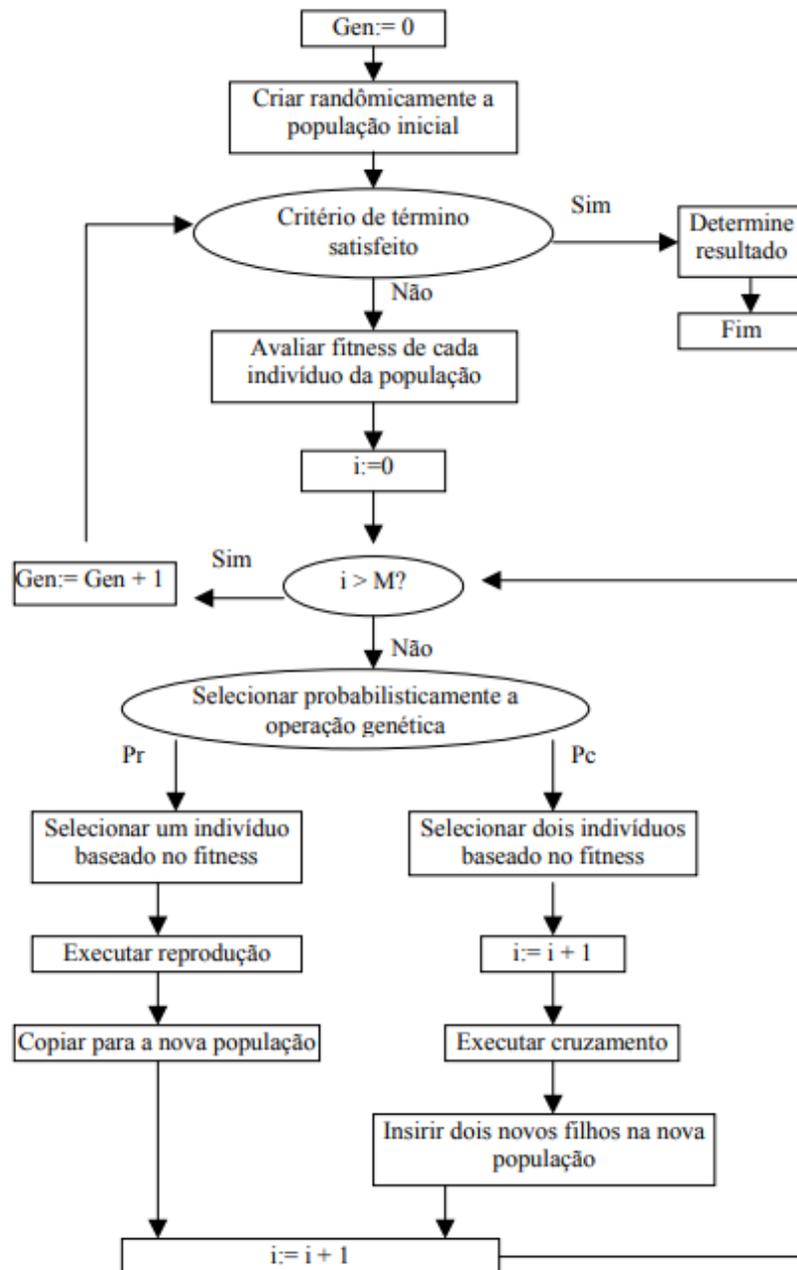
e práticas. Ao proporcionar flexibilidade, variedade e uma abordagem evolutiva para a resolução de problemas, a Programação Genética permanece como uma área empolgante e em constante evolução na computação evolutiva.

Figura 7 - Exemplo de *subtree mutation*. Adaptado de [4]



A Figura 8 representa os passos adotados na técnica. Na notação utilizada,  $M$  representa o tamanho da população,  $i$  é o número de indivíduos na população, e  $Pr$  e  $Pc$  denotam as probabilidades das operações de reprodução e cruzamento, respectivamente. Essas probabilidades serão explicadas em detalhes mais adiante.

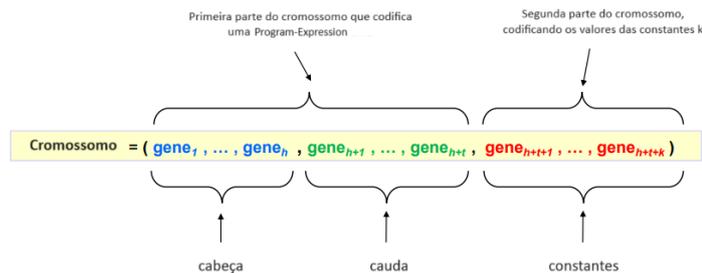
Figura 8 - Fluxograma da PG. Retirado de [5]



## 1.8 Biased Random-Key Genetic Programming

A representação dos programas em BRKGP são baseadas em chaves aleatórias restritas (BRKGP, do inglês “Biased Random-Key Genetic Programming”) é feita por meio de cromossomos lineares, onde a cabeça (*head*), veja o algoritmo 2, codifica a estrutura do programa; a cauda (*tail*), veja o algoritmo 3, os valores das constantes; e as funções de decodificação transformam esses cromossomos em expressões de programa. A decodificação envolve a identificação de funções, variáveis e constantes, culminando na formação da árvore de sintaxe do programa [4]. A representação de programas no BRKGP utiliza uma abordagem de expressão linear, conhecida como *Program-Expression* (PE), que codifica uma árvore de sintaxe em uma sequência de elementos, Figura 9. Essa representação oferece flexibilidade e eficiência na manipulação e evolução de programas. Ainda assim, mais à frente iremos explicar as diferenças da nossa BRKGP proposta (baseada em pilhas), em contraste à decodificação de BRKGP proposta na literatura.

Figura 9 - Estrutura cromossômica usada para codificar uma *Program-Expression* com um vetor de chaves aleatórias  $h + t + k$ . Adaptado de [4]

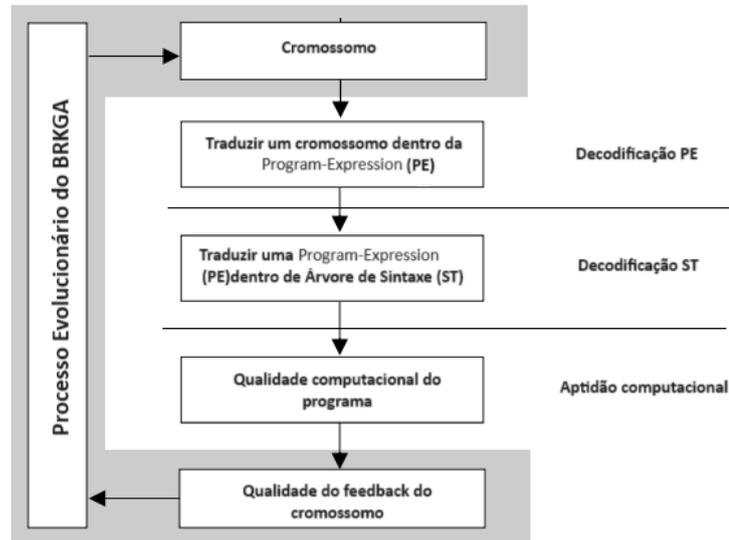


O BRKGP é uma abordagem inovadora baseada em quatro componentes principais: algoritmo genético com chaves aleatórias viés (BRKGA), cromossomos, procedimento de decodificação e função de aptidão. A escolha enviesada de pais para acasalamento diferencia o BRKGP de algoritmos genéticos tradicionais. A evolução ocorre no espaço contínuo de hiperespaço, explorando soluções por meio do decodificador[4], Figura 10.

O BRKGP emerge como uma técnica promissora na evolução de programas computacionais, oferecendo abordagens inovadoras na escolha de pais para acasalamento e na representação eficiente de programas. Sua aplicabilidade em problemas complexos, como regressão simbólica, destaca-se, proporcionando descobertas automáticas de relações em conjuntos de dados [4].

A estrutura do BRKGP, com uma seção dedicada à representação de programas e outra às constantes, proporciona uma maior modularidade e adaptabilidade. Isso permite que o algoritmo lide eficientemente com diferentes tipos de problemas, desde regressão

Figura 10 - Arquitetura do BRKGP. Adaptado de [4]



Algoritmo 2 - DecodeHead. Adaptado de [4]

```

1: for  $r = 1$  to  $h$  do
2:    $idx \leftarrow dc[r] \times (nF + nV + nK)$   $\triangleright$  onde  $[x]$  é o menor inteiro maior ou igual a  $x$ .  $\triangleright$ 
   Decodificando  $c[r] \rightarrow PE[r]$ 
3:   if ( $idx \leq nF$ ) then  $\triangleright$  vai ser uma função
4:      $PE[r] \leftarrow vF[idx]$ 
5:   else if ( $nF < idx \leq nF + nV$ ) then  $\triangleright$  vai ser uma variável
6:      $PE[r] \leftarrow vV[idx - nF]$ 
7:   else if ( $nF + nV < idx \leq nF + nV + nK$ ) then  $\triangleright$  vai ser uma constante
8:      $PE[r] \leftarrow vK[idx - nF - nV]$ 
9:   end if
10: end for
  
```

simbólica até a descoberta de regras em conjuntos de dados complexos. O BRKGP representa um avanço significativo na pesquisa em algoritmos evolutivos, e tem demonstrado versatilidade e eficácia em uma variedade de contextos. Suas contribuições para a resolução de problemas complexos e a descoberta automática de soluções consolidam seu papel.

Algoritmo 3 - DecodeTail. Adaptado de [4]

```

1: for  $r = h + 1$  to  $h + t$  do
2:    $idx \leftarrow dc[r] \times (nV + nK)$     $\triangleright$  onde  $[x]$  é o menor inteiro maior ou igual a  $x$ .    $\triangleright$ 
   Decodificando  $c[r] \rightarrow PE[r]$ 
3:   if  $(idx \leq nV)$  then                                      $\triangleright$  vai ser uma variável
4:      $PE[r] \leftarrow vV[idx]$ 
5:   else if  $(nV < idx \leq nV + nK)$  then                        $\triangleright$  vai ser uma constante
6:      $PE[r] \leftarrow vK[idx - nV]$ 
7:   end if
8: end for

```

## 1.9 Paradigma de *Less Is More*

O paradigma do “Less Is More” tem se destacado como uma abordagem inovadora e eficaz na resolução de problemas complexos, especialmente na área de otimização. Essa filosofia, embasada na ideia de encontrar a quantidade mínima de elementos necessários para alcançar a eficiência desejada, tem sido aplicada com sucesso em diversas disciplinas, incluindo a pesquisa operacional, onde a otimização de processos é crucial.

No contexto da otimização, a abordagem “Less Is More” visa simplificar os métodos e algoritmos, reduzindo a complexidade sem comprometer a eficácia. Isso está em contraposição à tendência histórica de buscar soluções mais elaboradas e intrincadas. A teoria por trás desse paradigma sustenta que, ao minimizar o número de elementos envolvidos, é possível obter resultados mais eficientes e, muitas vezes, mais eficazes.

Um exemplo notável dessa abordagem é evidenciado no artigo de Mladenovic et al. (2016) [61], onde o paradigma “Less Is More” é aplicado com sucesso para resolver o problema obnoxious p-median (OpMP). O estudo propõe uma heurística básica de busca em vizinhanças variáveis, utilizando apenas uma estrutura de vizinhança de intercâmbio. Além disso, é introduzida uma estratégia de busca local simples, denominada “facility best improvement”, que se revela mais eficiente do que estratégias convencionais.

Outro exemplo do paradigma “Less Is More” emerge como uma abordagem crucial na tomada de decisões sob incerteza. A distinção entre risco e incerteza, destacada por Frank Knight [62], ressalta a complexidade inerente a esse cenário, onde a simplificação deliberada e o emprego de heurísticas rápidas revelam-se estratégias robustas. Ao lidar

com a incerteza, a eficácia na tomada de decisões muitas vezes reside na capacidade de ignorar informações irrelevantes, adotando uma abordagem mais focada nos elementos essenciais. Nesse contexto, o “Less Is More” não apenas oferece efetividade decisória, mas também otimiza a alocação de recursos, conferindo uma perspectiva valiosa para gestores enfrentando a complexidade empresarial.

A relevância desse paradigma vai além da mera simplificação algorítmica. Propõe que a eficácia está intrinsecamente ligada à simplicidade, à identificação do essencial e à eliminação do supérfluo.

Além disso, a aplicação desse paradigma tem o por objetivo a busca por explicações mais claras sobre por que uma heurística funciona bem. Ao minimizar o número de ingredientes na busca, facilita-se a compreensão das razões subjacentes ao sucesso do método.

Neste trabalho, exploramos o conceito de “Less Is More” na expectativa de simplificar estratégias demasiadamente complexas, propostas na literatura, para o problema de regressão simbólica. Também aplicamos esse conceito ao simplificar o BRKGP da literatura mediante uma estrutura de decodificação por pilhas, o que elimina a necessidade de representar o programa (ou função) gerado como uma árvore sintática, avaliando diretamente cada operação enquanto aparece no vetor de chaves aleatórias. Finalmente, também efetuamos uma Decodificação Ativa, no intuito de evitar geração de operações inviáveis (como tentar aplicar uma soma quando somente um operando está disponível na pilha, ao invés de dois), aumentando drasticamente o espaço de soluções viáveis visitadas durante a execução do algoritmo BRKGP proposto (inspirado na “decodificação real de DNA” [63], mesmo quando falhas existem no processo).

## 2 REGRESSÃO SIMBÓLICA E APLICAÇÕES NA FÍSICA

A regressão simbólica é uma técnica de aprendizado de máquina que tem sido usada para descobrir modelos matemáticos simbólicos que apresentam a relação entre um conjunto de variáveis de entrada e uma variável de saída. A regressão simbólica é um método que parte de um grupo de pontos, onde cada ponto está relacionado a um valor, sendo  $p = (x_0, x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , enquanto o valor que está relacionando  $y \in \mathbb{R}$ . Essa é uma técnica que resulta em uma função implícita  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , onde  $f(p) = y$ , o algoritmo procura definir uma função onde o erro seja o menor possível [6].

A metodologia de regressão simbólica inicia com a elaboração de um conjunto de dados de treinamento contendo valores conhecidos das variáveis de entrada e de saída. Logo após, um algoritmo de regressão simbólica é aplicado aos dados de treinamento para encontrar o melhor modelo simbólico que se ajuste aos dados. O algoritmo busca uma função com erro mínimo entre a previsão do modelo e os valores de saída reais.

Esta técnica pode ser utilizada para fins de experimentos científicos, previsões e estatísticas, seu horizonte de soluções é vasto, o que acaba causando uma inviabilidade de resolução do problema de forma analítica. Devido a isto, problemas de regressão simbólica tendem a usar com mais frequência a Programação Genética (PG) [64]. A PG gera populações de expressões simbólicas aleatórias e as avalia em relação ao conjunto de treinamento. As expressões mais bem ajustadas são selecionadas para cruzamento e mutação, gerando uma nova população. Esse processo é continuado até que uma solução adequada seja encontrada.

Uma vez que um modelo simbólico tenha sido encontrado, ele pode ser usado para prever a saída correspondente a novos valores de entrada. A regressão simbólica tem sido amplamente utilizada em diversas áreas, como finanças, engenharia e medicina, para modelagem de sistemas complexos e tomada de decisões.

### 2.1 Aplicações na Física

A regressão simbólica, uma técnica de aprendizado de máquina, tem se destacado como uma ferramenta poderosa na descoberta de relações analíticas em dados físicos. Na elaboração do trabalho foi explorado diversas abordagens recentes que aplicam a regressão simbólica em contextos físicos, destacando a relevância e eficácia desses métodos.

No estudo intitulado “*AI Feynman: A physics-inspired method for symbolic regression*”, Udrescu e Tegmark[7] propõem um algoritmo recursivo multidimensional de regressão simbólica que combina técnicas de redes neurais com inspirações da física. Ao aplicar este método a 100 funções das “Feynman Lectures on Physics” [19, 20, 21, 22], os

autores conseguem superar softwares previamente disponíveis como o Eureka, alcançando uma taxa de sucesso de 90% em um conjunto de teste mais desafiador.

Outro trabalho, “*Multi-objective Symbolic Regression for Physics-aware Dynamic Modeling*”, de Kubalík et al. [23], destaca a importância de incorporar propriedades físicas desejadas no processo de aprendizado de modelos dinâmicos. O método proposto otimiza modelos levando em consideração tanto o erro de treinamento quanto a conformidade com propriedades físicas específicas. Experimentos com sistemas reais, como robôs móveis e sistemas de manipulação magnética, mostram que o algoritmo gera modelos precisos e justificados fisicamente.

Em “*Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data*”, Vaddireddy et al. [24] apresentam uma abordagem modular para extrair física oculta de observações discretas e esparsas. Utilizando programação genética e regressão sequencial, os autores conseguem descobrir modelos para fenômenos fluidodinâmicos, ilustrando o potencial dessas técnicas em problemas complexos.

O trabalho de Tenachi et al. [65], “*Deep symbolic regression for physics guided by units constraints*”, introduz o framework  $\Phi$ -SO, que utiliza aprendizado profundo para descobrir expressões simbólicas consistentes com unidades físicas. Essa abordagem mostra-se eficaz na eliminação de soluções fisicamente impossíveis, proporcionando uma melhoria significativa no desempenho.

Por fim, “*A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge*”, de Keren et al. [66], apresenta o SciMED, um framework que combina sabedoria disciplinar com métodos de regressão simbólica. Experimentos com dados de um sistema de esfera em queda demonstram a robustez do SciMED na descoberta de expressões simbólicas significativas, destacando seu desempenho superior em relação a pacotes de software concorrentes.

Esses estudos evidenciam que a regressão simbólica não apenas se destaca na descoberta de relações físicas complexas, mas também oferece percepções interpretáveis, fundamentais para a compreensão aprofundada de fenômenos físicos variados. Essas abordagens mostram-se promissoras para aplicações mais amplas na modelagem e compreensão de sistemas físicos complexos.

### 3 REVISÃO LITERATURA

#### 3.1 BRKGP com Pilhas

Uma versão preliminar do BRKGP com pilhas implementado em C foi publicado em “Programação Genética com Chaves Aleatórias Viciadas em Notação Pós-Fixada para o Problema de Regressão Simbólica” [6] apresentando uma metodologia eficiente para lidar com o desafiador problema da regressão simbólica. A técnica empregada neste estudo integra conceitos da Programação Genética e do BRKGA, a combinação desses resulta em uma aplicação que é uma extensão o BRKGP, adaptando-os para uma representação em notação pós-fixada, proporcionando uma abordagem singular na resolução deste tipo de problema. Apresentamos melhorias recentes em cima da proposta anterior, para abordar mais funções, e também preparar o programa para a linguagem C++ com dependências externas complexas para lidar com simplificação de expressões simbólicas.

A representação da solução é estruturada mediante três vetores essenciais: a pilha, a ordem dos numerais e a ordem das operações. A utilização da pilha como estrutura central demonstra a inovação da metodologia, pois permite a manipulação eficiente das expressões matemáticas, seguindo a lógica de operações binárias e unárias. A pilha, com valores no domínio de -1 a 2, destaca-se por indicar a natureza e o tipo das operações a serem realizadas, garantindo a coerência lógica da expressão, Figura 11, onde -1 representa uma operação binária entre duas variáveis ou constantes, 0 representa uma operação unária que envolve apenas uma constante ou variável, 1 representa uma variável ou constante, e 2 representa um espaço vazio que não tem interferência na expressão.

Figura 11 - Exemplo de Pilha.  
Retirada [6]

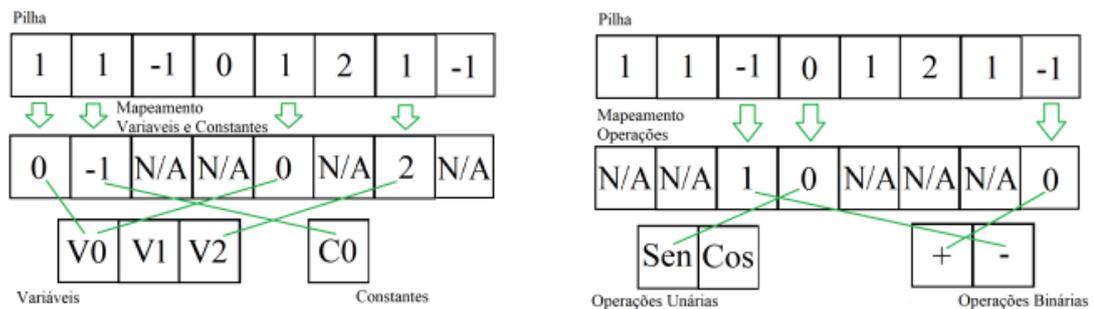
1	1	-1	0	1	2	1	-1
---	---	----	---	---	---	---	----

A ordem dos numerais e a ordem das operações surgem como complementos cruciais na construção da solução. A primeira mapeia os índices das variáveis e constantes, enquanto a segunda determina a ordem e a natureza das operações realizadas pela pilha. Esses vetores proporcionam uma representação mais abstrata e independente da expressão, o que facilita a manipulação genética.

A eficácia da abordagem é evidenciada pelo uso de técnicas de Programação Genética, uma extensão dos algoritmos genéticos que está fundamentada na teoria da evolução por seleção natural de Darwin. O processo evolutivo envolve a criação e avaliação de indivíduos, cruzamento de características e mutações. A preferência por indivíduos mais

aptos, a utilização de cruzamentos com elites e a introdução de novos indivíduos aleatórios contribuem para uma exploração mais abrangente do espaço de soluções, Figura 12, na primeira pilha, temos um exemplo da ordem dos numerais, onde a pilha realiza um mapeamento das variáveis e das constantes. Os valores da primeira pilha, que representam operações binárias ou a representação de variáveis, ou constantes, são transferidos para a próxima pilha, que realizará o mapeamento subsequente. Nela, observamos que o domínio dos índices é positivo, com 0 representando a primeira variável e a quantidade de variáveis subtraída por 1 representando a última. Por outro lado, as constantes são representadas pelo índice -1. Em resumo, se tivermos 3 variáveis, teremos três valores de índice na pilha que mapeia as variáveis e constantes, variando de 0 a 2, enquanto as constantes são representadas pelo índice -1. Na segunda pilha onde é mapeado as operações os valores de -1 e 0 são colocados em outra pilha que ordena as operações, temos aqui -1 representando uma operação binária e 0 uma operação unária.

Figura 12 - Exemplo de Mapeamento dos numerais e das operações. Retirada [6]



Em suma, a metodologia se destaca pela sua capacidade de lidar com expressões simbólicas complexas e pela ênfase na simplicidade da representação, evitando a complexidade associada a estruturas baseadas em árvores. A abordagem também oferece flexibilidade por meio de parâmetros ajustáveis, permitindo adaptações a diferentes cenários práticos.

### 3.2 EMA - *EMpirical Analysis of algorithms*

A análise empírica de algoritmos desempenha um papel crucial na avaliação prática do desempenho desses algoritmos, que complementam as limitações dos métodos analíticos. No âmbito dessa pesquisa, a ferramenta *EMpirical Analysis of algorithms* (EMA) [67] se destaca como uma abordagem inovadora, oferecendo uma estimativa robusta da complexidade assintótica de algoritmos em diversos ambientes e linguagens de programação.

A metodologia do EMA se baseia em uma análise empírica profunda, permitindo a geração de funções de complexidade que descrevem de maneira precisa e automática o comportamento de algoritmos em diferentes cenários. Destacam-se os seguintes aspectos da metodologia

1. O EMA realiza múltiplas execuções de um mesmo algoritmo para diferentes tamanhos de entrada, cobrindo uma ampla gama de casos. Esse enfoque abrangente permite a captura de padrões de desempenho em diferentes escalas;
2. Durante as execuções, o EMA monitora o consumo de recursos, incluindo tempo de CPU e uso de memória. Essa abordagem detalhada proporciona uma compreensão abrangente do comportamento do algoritmo em termos de eficiência computacional;
3. O EMA utiliza técnicas avançadas de regressão não-linear para ajustar modelos matemáticos às observações empíricas. Essa abordagem estatística contribui para a identificação precisa dos parâmetros das funções de complexidade;
4. Uma característica distintiva do EMA é sua capacidade de gerar automaticamente funções candidatas para descrever a complexidade do algoritmo. A seleção da função final é orientada por critérios que visam a simplicidade e precisão.

O EMA se destaca pelo uso diferenciado de ferramentas, o EMA é capaz de analisar algoritmos implementados em diversas linguagens de programação, proporcionando flexibilidade aos pesquisadores e desenvolvedores. A ferramenta aborda diversas classes de complexidade, incluindo polilogarítmica, polinomial e exponencial. Essa amplitude de cobertura torna o EMA aplicável a uma variedade de algoritmos em diferentes domínios.

O EMA emerge como uma ferramenta robusta e inovadora para a análise empírica de algoritmos. Sua metodologia abrangente, capacidade de geração automática de funções de complexidade e versatilidade linguística o tornam uma escolha valiosa para pesquisadores, desenvolvedores e acadêmicos envolvidos na avaliação prática de algoritmos em diversos contextos. A eficácia do EMA foi evidenciada por meio de comparações com outras ferramentas e sua aplicação em estudos de caso, consolidando seu papel como uma ferramenta confiável e avançada na análise empírica de algoritmos.

Além das contribuições fundamentais apresentadas pelo EMA, a integração do algoritmo de regressão simbólica representa um avanço significativo no aprimoramento da capacidade preditiva da ferramenta. A regressão simbólica proporciona uma camada adicional de interpretabilidade aos resultados, ao expressar a complexidade do algoritmo em termos de símbolos matemáticos familiares. Essa abordagem não apenas facilita a compreensão dos modelos gerados, mas também oferece percepções valiosas sobre as relações subjacentes aos dados empíricos.

Ao incorporar o algoritmo de regressão simbólica no EMA, torna-se possível não apenas identificar as características gerais das funções de complexidade, mas também

entender a natureza específica das interações entre as variáveis. Isso é particularmente crucial em algoritmos complexos, nos quais o desempenho pode ser influenciado por múltiplos fatores de maneira não linear. A regressão simbólica, ao representar essas relações de forma explícita, fornece uma perspectiva mais detalhada e interpretável sobre o comportamento do algoritmo.

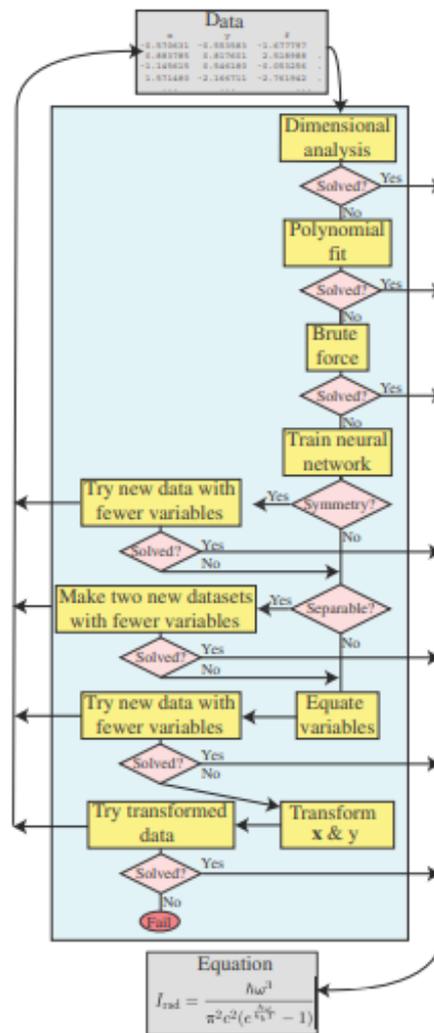
Essa combinação entre abordagem empírica do EMA e a interpretabilidade proporcionada pela regressão simbólica não apenas fortalece a confiabilidade das análises, mas também amplia o escopo das conclusões extraídas. O entendimento detalhado das características matemáticas e das funções geradas contribui para uma análise mais profunda e informada das complexidades algorítmicas, estabelecendo assim um novo patamar para análise empírica de algoritmos.

### 3.3 Feynman

O método proposto no trabalho “*AI Feynman: A physics-inspired method for symbolic regression*” [7] emprega uma abordagem para enfrentar a complexidade associada à descoberta simbólica de funções genéricas altamente complicadas, as quais, em muitos casos, são consideradas praticamente impossíveis de serem identificadas por técnicas convencionais de regressão simbólica. Na física a complexidade é inerente a funções genéricas  $f(x_1, \dots, x_n)$ , e suas descobertas levam tempo. O algoritmo utilizado na literatura segue uma sequência de etapas cuidadosamente projetadas para lidar com as nuances específicas de funções físicas e otimizar a descoberta simbólica. Na Figura 13:

1. O algoritmo inicia com uma análise dimensional rigorosa, onde são consideradas as unidades físicas associadas à função  $f$  e a cada variável independente. Esta análise serve como um guia restritivo, garantindo que as unidades sejam consistentes e fornecendo um alicerce sólido para a descoberta simbólica;
2. Reconhecendo a prevalência de comportamentos polinomiais em fenômenos físicos, o algoritmo realiza um ajuste polinomial estratégico de baixa ordem, explorando a natureza polinomial subjacente das funções;
3. A abordagem de força bruta é aplicada com inteligência, explorando uma diversidade de formas funcionais potenciais de maneira direcionada e estratégica;
4. Para capturar complexidades não lineares, uma rede neural é treinada adaptativamente, ajustando-se aos padrões complexos presentes nos dados.
5. O algoritmo itera, experimentando continuamente com novos dados e, inicialmente, um conjunto reduzido de variáveis para otimizar a busca simbólica.

Figura 13 - Ilustração esquemática do *AI Feynman Algorithm*. Retirada [7]



6. A manipulação de variáveis é realizada de maneira avançada, incorporando transformações específicas para explorar diversas formas funcionais, levando a uma expressão simbólica mais abrangente.
7. É um processo altamente iterativo, onde é refinado constantemente a expressão simbólica à medida que novas informações são assimiladas e novas tentativas são empreendidas.

O AI Feynman combina diversas técnicas, desde análise dimensional até experimentação iterativa, visando contornar as dificuldades inerentes à descoberta simbólica. Ao incorporar propriedades específicas de funções físicas, como simetria, separabilidade e ordem polinomial, o AI Feynman demonstra uma eficácia notável na identificação de expressões simbólicas que modelam com precisão fenômenos físicos complexos, destaca-se notavelmente em comparação com o software de regressão simbólica Eureka. Enquanto o Eureka alcança uma taxa de descoberta de 68% para as funções dos mistérios de Feynman e apenas 15% para funções adicionais, o AI Feynman supera esses resultados significativamente, atingindo taxas impressionantes de 100% e 90%, respectivamente. Esta superioridade é especialmente evidente em mistérios mais desafiadores, nos quais a estratégia baseada em redes neurais do AI Feynman é empregada de forma iterativa, revelando-se mais eficaz em reduzir sistematicamente o problema a formas mais simples com menos variáveis, em comparação com a abordagem de busca extensiva do Eureka.

Essa abordagem complexa não apenas amplia a compreensão das relações subjacentes nos dados, mas também respeita as limitações físicas e unidades associadas, consolidando-se como uma ferramenta excepcionalmente poderosa para a pesquisa científica em diversas disciplinas.

## 4 USO DE BIBLIOTECAS EXTERNAS EM C/C++

### 4.1 CAS – Computer Algebraic System

Um Sistema de Álgebra Computacional (CAS), é uma ferramenta poderosa que lida com a resolução, manipulação e representação de expressões matemáticas de forma analítica. Sua função vai além das calculadoras e softwares convencionais, pois trabalha diretamente com expressões simbólicas, permitindo a manipulação de variáveis, fórmulas e funções de maneira simbólica.

As principais características de um CAS incluem a capacidade de resolver uma ampla gama de problemas matemáticos, abrangendo áreas como álgebra linear, cálculo, funções diferenciais e funções algébricas. Ao representar parâmetros e variáveis como símbolos em funções, um CAS possibilita a resolução, manipulação e avaliação analítica dessas expressões, evitando a necessidade de gerar dados numéricos.

Uma das vantagens notáveis dos CAS é a capacidade de realizar cálculos simbólicos exatos, em contraste com métodos numéricos que podem sofrer com erros de ponto flutuante. Algoritmos sofisticados incorporados nos CAS permitem a solução e simplificação de funções algébricas, sistemas de funções e funções diferenciais, contribuindo para a precisão e desempenho em diversas aplicações, incluindo otimização.

### 4.2 Octave

A regressão simbólica, uma abordagem poderosa na modelagem matemática, emprega algoritmos para encontrar expressões analíticas que descrevem relacionamentos entre variáveis. Ao adotar o Octave como ferramenta para manipulação simbólica, é possível explorar recursos como a função **'simplify'**, que desempenha um papel crucial na simplificação de expressões resultantes da regressão simbólica.

A função **'simplify'** no Octave é projetada para reduzir expressões a formas mais simples, removendo redundâncias algébricas e apresentando resultados de maneira mais elegante.

A capacidade de simplificar expressões é particularmente valiosa na otimização de modelos simbólicos gerados pela regressão. No entanto, ao integrar essa abordagem com a implementação prática, surgem desafios significativos.

Um desses desafios reside na transição entre a implementação prática em linguagens como C++ e o ambiente simbólico do Octave. Enquanto a regressão simbólica pode gerar expressões complexas e detalhadas, a implementação eficiente dessas expressões em linguagens de programação de baixo nível requer uma forma simplificada e otimizada.

A simplificação simbólica, embora útil, pode não garantir a forma mais eficiente para a implementação prática.

A utilização de duas etapas distintas no processo, uma para a geração simbólica e outra para a simplificação no ambiente Octave, introduz a necessidade crítica de sincronização e validação contínua. Quando ocorrem alterações na expressão em C++, é necessário realizar ajustes correspondentes na etapa de simplificação simbólica. Essa interdependência entre as etapas adiciona uma camada adicional de complexidade ao processo, pois qualquer modificação na geração simbólica deve ser validada e ajustada na fase de simplificação, garantindo assim a consistência e a precisão do resultado.

Outra consideração crítica é o desempenho computacional. A simplificação de expressões pode ser computacionalmente intensiva, especialmente para expressões complexas. Em cenários em que o tempo de execução é uma consideração crucial, a simplificação simbólica pode ser uma operação custosa.

Embora a manipulação simbólica no Octave, especialmente por meio da função *simplify*, ofereça uma abordagem elegante para simplificar expressões resultantes da regressão simbólica, a integração prática dessas expressões em linguagens de programação e os desafios associados à sincronização e desempenho computacional foram considerados. Dito isto, optou-se por não usar a função de simplificação de funções do Octave `'simplify'`.

### 4.3 Mathomatic

A presente pesquisa propõe uma investigação detalhada sobre a biblioteca Mathomatic, uma biblioteca de C notável, que desempenhou um papel significativo no domínio da regressão simbólica. Desenvolvida pelo pesquisador George Gesslein II, a Mathomatic foi concebida com o propósito de simplificar operações algébricas avançadas e explorar soluções simbólicas para funções matemáticas complexas. Gesslein, infelizmente, faleceu recentemente, deixando para trás um legado marcante no campo da matemática simbólica.

A Mathomatic, ao longo dos anos, destacou-se por sua capacidade de realizar cálculos simbólicos automatizados, tornando-se uma ferramenta valiosa para pesquisadores e acadêmicos envolvidos em projetos de regressão simbólica. Seu algoritmo inovador permitia a manipulação simbólica de expressões matemáticas, facilitando a análise de modelos e a dedução de relações entre variáveis. A biblioteca tornou-se uma escolha frequente em cenários onde a compreensão profunda das relações matemáticas subjacentes era essencial.

No entanto, é importante notar que, lamentavelmente, a Mathomatic encerrou seu desenvolvimento após o falecimento de George Gesslein II. Este fato representa não apenas uma perda para a comunidade acadêmica, mas também suscita questões sobre o futuro das ferramentas associadas. A ausência de suporte contínuo e atualizações pode impactar

o uso generalizado da biblioteca em projetos futuros.

Do ponto de vista metodológico, a Mathematic destacava-se por sua abordagem algorítmica eficiente na resolução de funções, permitindo que os pesquisadores explorassem modelos matemáticos de maneira simplificada e precisa. No entanto, a interrupção do desenvolvimento e manutenção pode levantar questionamentos sobre a viabilidade contínua do uso da Mathematic em pesquisas futuras. É notável, que os usuários que vão optar por empregar esta biblioteca devem considerar cuidadosamente os desafios associados à ausência de suporte técnico.

#### 4.4 ViennaMath

A manipulação eficiente de expressões matemáticas é crucial em diversos algoritmos, e o ViennaMath se apresenta como uma opção de biblioteca para essa finalidade. Abordaremos aspectos essenciais da manipulação de expressões com o ViennaMath, uma biblioteca de C++, destacando suas funcionalidades e limitações, especialmente no contexto de regressões simbólicas.

O ViennaMath oferece uma infraestrutura flexível para a definição e avaliação de expressões matemáticas em C++. A manipulação dessas expressões é o foco, destacando a importância de adaptar e transformar as expressões conforme as necessidades dos algoritmos.

A avaliação de expressões com o ViennaMath é efetuada por meio do operador de parênteses `'()`'. No entanto, a biblioteca introduz a função `'viennamath::eval()'`, que proporciona uma interface genérica para avaliações tanto em tempo de compilação quanto em tempo de execução. É notável que, em casos nos quais a expressão contém partes não tratáveis em tempo de compilação, a avaliação é automaticamente transferida para tempo de execução.

Entretanto, ao considerar a aplicação do ViennaMath em regressões simbólicas, surge uma limitação relevante. A biblioteca, até a versão 1.0.0, demonstra restrições significativas na expansão de expressões em tempo de execução, suportando exclusivamente a expansão em tempo de compilação. Isso representa uma barreira substancial quando lidamos com algoritmos que requerem manipulações simbólicas dinâmicas, como em regressões simbólicas.

A simplificação de expressões é uma etapa crucial em algoritmos matemáticos. O ViennaMath disponibiliza a função `'simplify()'`, que visa reduzir expressões desnecessárias. Isso é particularmente útil para otimizar expressões antes da avaliação. No entanto, é vital observar que, ao lidar com regressões simbólicas, a capacidade de simplificação pode ser limitada quando comparada a outras bibliotecas especializadas nessa área.

Ao explorar as capacidades do ViennaMath, deparamo-nos com algumas limitações

específicas para o contexto de regressões simbólicas. A biblioteca apresenta restrições notáveis na manipulação simbólica dinâmica, impedindo a expansão de expressões em tempo de execução, o que é essencial para algoritmos avançados de regressão simbólica.

O ViennaMath destaca-se como uma biblioteca eficiente para manipulação simbólica em tempo de compilação, oferecendo um conjunto sólido de ferramentas para expressões matemáticas. No entanto, ao considerar o cenário de regressões simbólicas, é importante ponderar suas limitações, especialmente quanto à capacidade de expansão dinâmica de expressões em tempo de execução.

#### 4.5 Biblioteca GiNaC em C/C++

O GiNaC (GiNaC is Not a CAS) surge da percepção de que a maioria dos sistemas de álgebra computacional (CAS) atuais são linguisticamente e semanticamente empobrecidos. Embora sejam ferramentas poderosas para aprender matemática e resolver problemas específicos, muitos desses sistemas carecem de estruturas linguísticas modernas que permitam a criação de projetos em larga escala. O GiNaC é uma iniciativa para superar essa lacuna, estendendo uma linguagem de programação bem estabelecida e padronizada (C++) com capacidades simbólicas fundamentais.

No âmbito de cálculos simbólicos e manipulação de álgebra computacional, o GiNaC oferece uma gama diversificada de classes que representam objetos matemáticos fundamentais, cada uma desempenhando um papel crucial em análises simbólicas avançadas, alguns exemplos são:

**Símbolos e Expressões:** A representação simbólica é fundamental para a análise matemática avançada, e no GiNaC, símbolos são declarados como variáveis simbólicas que podem ser incorporadas em expressões matemáticas. Essa capacidade de representação simbólica é vital para a formulação e manipulação de funções algébricas complexas.

**Números e Precisão:** A biblioteca utiliza a Class Library for Numbers (CLN) para manipular números em diferentes formatos, incluindo inteiros, racionais e complexos. Essa integração fornece um nível de precisão necessário para cálculos rigorosos, evitando erro de ponto flutuante.

**Operações Algébricas:** O GiNaC suporta operações algébricas essenciais, como somas, produtos e potências. A capacidade de simplificação automática dessas expressões para formas canônicas é crucial para garantir a consistência e a precisão dos resultados simbólicos.

**Integrais e Derivadas:** A manipulação simbólica de integrais e derivadas é uma parte intrínseca da análise matemática avançada. O GiNaC oferece suporte a expressões simbólicas de integrais definidas, permitindo a representação simbólica dessas operações em um contexto científico.

**Matrizes e Álgebra Linear:** A classe `matrix` possibilita a representação e manipulação de matrizes de expressões simbólicas. Essa funcionalidade é essencial para aplicações científicas que envolvem álgebra linear, como análises de sistemas lineares e transformações matriciais.

**Objetos Indexados e Tensores:** A manipulação simbólica de objetos indexados e tensores é crucial em diversas aplicações. As classes `idx` e `indexed` fornecem uma estrutura robusta para expressar quantidades tensoriais, essenciais para análises em campos como física teórica e engenharia.

**Álgebra Não Comutativa:** O GiNaC também lida com álgebras não comutativas, como a álgebra de Clifford e a álgebra de Lie  $\mathfrak{su}$ . Essas classes são particularmente relevantes em contextos científicos relacionados à física de partículas e teoria de grupos.

O GiNaC fornece uma infraestrutura computacional robusta para análises simbólicas avançadas. Suas classes especializadas oferecem um conjunto abrangente de ferramentas para representação simbólica, simplificação de expressões e manipulação algébrica, atendendo às demandas rigorosas de cálculos simbólicos em diversos contextos.

## 4.6 OptFrame

OptFrame é um framework desenvolvido para modelar e resolver problemas desafiadores de otimização por meio de técnicas metaheurísticas. Desenvolvido em C++ moderno por pesquisadores da Universidade Federal de Ouro Preto e Universidade Federal Fluminense, pretende proporcionar tanto eficiência computacional elevada quanto facilidade de uso. Sua última versão, v5, foi lançada em 2023, introduzindo diversas características de programação funcional e capacidades C++11/14/17/20.

OptFrame oferece suporte a várias metaheurísticas famosas, como: Algoritmo Genético e Estratégias de Evolução, Greedy Randomized Adaptive Search Procedures (GRASP), Iterated Local Search (ILS), Simulated Annealing, Busca Tabu, BRKGA e Variable Neighborhood Search (VNS).

O OptFrame se destaca por oferecer uma variedade de metaheurísticas avançadas, e se torna uma escolha poderosa para a resolução de problemas de otimização complexos. Essas metaheurísticas vão além das abordagens convencionais, permitindo aos usuários explorar soluções por várias abordagens.

## 5 METODOLOGIA PROPOSTA

Dentre as estratégias empregadas para aprimorar o desempenho do algoritmo proposto, uma das abordagens-chave é a integração da biblioteca GiNaC. Esta biblioteca desempenha um papel crucial ao simplificar funções e realizar análise dimensional dos dados. Em um contexto físico, a análise das unidades associadas a cada variável desempenha um papel vital, não apenas na compreensão do fenômeno em estudo, mas também na identificação da função que melhor representa o conjunto de dados. A capacidade da GiNaC de lidar com a simplificação de expressões e incorporar a dimensionalidade dos dados fortalece significativamente o processo de modelagem e otimização do algoritmo. No âmbito operacional, destaca-se que a implementação do GiNaC já está em funcionamento no ambiente Windows e Linux/Ubuntu.

Outra faceta crucial da metodologia adotada neste trabalho, também em andamento, é a implementação do módulo de ajuste de parâmetros por meio do IRACE (Iterated Racing for Automatic Configuration of Algorithms). Considerando que estamos utilizando parâmetros previamente explorados em estudos anteriores[6], ainda não foram delineadas medidas específicas para identificar os parâmetros ótimos. Contudo, é essencial salientar que esta lacuna já está claramente definida como um objetivo subsequente. Os testes conduzidos até o momento forneceram percepções valiosas, e o desenvolvimento contínuo do módulo de ajuste de parâmetros está em andamento. A busca pelos parâmetros ideais representa uma etapa de melhoria do algoritmo.

### 5.1 Lidando com erros numéricos

No contexto da física, é amplamente reconhecido que existem constantes com valores extremamente pequenos ou grandes. Esse fato apresenta um desafio significativo que pode afetar adversamente diversos algoritmos, aumentando o tempo de processamento em muitas execuções. No âmbito deste estudo, enfrentamos essa questão e desenvolvemos uma estratégia para lidar com as constantes físicas, de modo a ajustá-las a intervalos aceitáveis que permitissem ao algoritmo operar sem truncamento.

A abordagem adotada envolveu a introdução de um método iterativo de ajuste de escala cujo valor era igual a  $10^{10}$ . Essa escolha foi motivada pelo fato de que  $10^{10}$  é um valor que pode eficazmente redimensionar a maioria das constantes físicas, tornando-as mais adequadas para as operações do algoritmo. Essa estratégia provou ser eficaz e demonstrou melhorar significativamente a precisão das funções geradas. Um exemplo notável foi a Distribuição de Bose-Einstein, que contém duas constantes muito pequenas, para esse caso, ao invés de multiplicar a base  $10^{10}$ , é realizado a divisão, sendo assim

seu valor é reduzido a escala desejada. O algoritmo frequentemente as reduzia a zero, mas com a introdução desse método iterativo, as funções puderam ser manipuladas com sucesso.

Considerando os resultados favoráveis dessa abordagem de pré-processamento, surge a perspectiva de incorporá-la ao algoritmo de forma automática. A ideia é permitir que o algoritmo identifique automaticamente constantes que são muito grandes e aplique esse redimensionamento dinâmico durante a execução. Isso proporcionaria uma adaptação contínua às características das funções, otimizando a precisão e eficiência do algoritmo em uma variedade de contextos. Essa implementação automática do pré-processamento seria um avanço significativo para tornar o algoritmo mais robusto e versátil em diferentes situações.

## 5.2 Kahan Summation

O Algoritmo de Soma de Kahan, também conhecido como soma compensada e algoritmo de soma com transporte, é uma técnica projetada para minimizar a perda de significância no resultado total obtido pela adição de uma sequência de números de ponto flutuante de precisão finita. Essa abordagem visa mitigar os erros de arredondamento inerentes às representações finitas de números de ponto flutuante.

A perda de significância ocorre devido à limitação na representação precisa de certos números na forma de ponto flutuante, como é comum nas linguagens de programação, como Java, que utilizam tipos de ponto flutuante de precisão simples (float) e dupla precisão (double). Esses números são representados em um formato de fração, mantissa e expoente.

O erro de ponto flutuante surge quando números que não podem ser exatamente representados na forma de ponto flutuante são submetidos a operações aritméticas. Por exemplo, o número decimal 0,1 não pode ser representado de maneira exata em binário, resultando em um erro cumulativo ao realizar operações com ele.

O Algoritmo de Soma de Kahan aborda esse problema mantendo uma compensação (uma variável para acumular pequenos erros) durante a execução da soma. Essa compensação é atualizada a cada iteração para corrigir os erros acumulados. O Algoritmo de Soma de Kahan é uma ferramenta importante para lidar com problemas de precisão em operações com números de ponto flutuante, contribuindo para resultados mais precisos em cálculos numéricos.

### 5.3 Pseudo-código Proposto

O BRKGP é uma meta-heurística baseada em algoritmos genéticos que combina elementos da programação genética com as ideias do algoritmo BRKGA. O BRKGP é projetado para resolver problemas de regressão simbólica, onde o objetivo é encontrar uma função implícita que relaciona um conjunto de entradas a um valor de saída.

Este método é uma meta-heurística que aborda problemas de regressão simbólica, nos quais se procura encontrar uma função implícita que relaciona conjuntos de dados empíricos. Esse tipo de problema é comumente encontrado em experimentos científicos, previsões e estatísticas.

A principal característica do BRKGP reside na representação e manipulação de soluções. Cada solução, chamada de indivíduo, é codificada em um vetor que combina quatro elementos: a pilha, a ordem dos numerais, a ordem das operações e as sementes. Esses elementos são essenciais para a construção da expressão matemática que representa a solução. A pilha determina a ordem de avaliação da expressão, enquanto a ordem dos numerais e das operações mapeia as variáveis e operadores matemáticos. As sementes são usadas para garantir que a informação não se perca, sendo necessário para armazenar nessa estrutura a semente para resgatar a mesma solução em um passo posterior e com isso não gerar indeterminismos.

O processo de otimização é conduzido por uma população de indivíduos, cada um representando uma solução candidata. A evolução da população ocorre por meio de operadores genéticos, como mutação e cruzamento. A seleção dos indivíduos é guiada por um critério de adaptação, avaliado pela qualidade das soluções em relação aos dados de treinamento.

Um aspecto notável do BRKGP é a ênfase na decodificação ativa das soluções. A decodificação converte a representação codificada dos indivíduos em expressões matemáticas compreensíveis. Esse processo permite avaliar o desempenho das soluções e calcular seu custo em comparação com os dados de treinamento. A abordagem de decodificação ativa é crucial para evitar soluções inviáveis e garantir a qualidade das expressões geradas.

Além disso, o BRKGP incorpora estratégias específicas para lidar com a natureza simbólica e abstrata das funções a serem otimizadas. A introdução de elementos da teoria da evolução, como mutação, cruzamento e seleção natural, inspira-se na Programação Genética e contribui para a busca eficiente de soluções no vasto espaço de possíveis expressões matemáticas.

O Algoritmo 4 é parametrizado para fornecer flexibilidade e adaptabilidade a diferentes cenários práticos. Os parâmetros incluem o tamanho da população, a proporção de elites e mutantes, bem como critérios de parada, como o número máximo de reinícios e o limite de gerações sem melhoria.

Na Tabela 1, apresentam-se os parâmetros de entrada e sua disposição estratégica

Algoritmo 4 - BRKGP

**Require:** Parâmetros do BRKGP  $P$ ; Conjunto de Entradas e Saídas para avaliação  $Q = (Inp, Out)$ ; Critérios de parada ( $restartMax, noImprovementMax$ )

**Ensure:** Melhor indivíduo decodificado  $I$  com custo  $f$

```

1:  $restart \leftarrow 0$ 
2:  $custoMelhorIndividuo \leftarrow \infty$ 
3: while  $restart < restartMax$  do
4:    $noImprovement \leftarrow 0$ 
5:    $Desempilhar \leftarrow GeraPopulacaoAleatoria()$ 
6:    $Desempilhar \leftarrow DecodificaPopulacao(Q, Desempilhar)$ 
7:    $Desempilhar \leftarrow OrdenaIndividuosPeloCusto(Desempilhar)$ 
8:    $custoMelhorIndividuo \leftarrow custoMelhorIndG(Desempilhar)$ 
9:   while  $noImprovement < noImprovementMax$  do
10:     $Desempilhar_{mutante} \leftarrow GeraMutanteGeracaoKPlus1(P, Desempilhar)$ 
11:     $Desempilhar_{mutante} \leftarrow Crossover(P, Desempilhar_{mutante})$ 
12:     $Desempilhar_{mutante} \leftarrow DecodificaDesempilharulacao(Q, Desempilhar_{mutante})$ 
13:     $Desempilhar_{mutante} \leftarrow OrdenaIndividuosPeloCusto(Desempilhar_{mutante})$ 
14:    if  $custoMelhorIndGkPlus1(Desempilhar_{mutante}) <$ 
 $custoMelhorIndGk(Desempilhar)$  then
15:       $noImprovement \leftarrow 0$ 
16:    else
17:       $noImprovement \leftarrow noImprovement + 1$ 
18:    end if
19:    if  $custoMelhorIndGkPlus1(Desempilhar_{mutante}) < custoMelhorIndividuo$ 
then
20:       $custoMelhorIndividuo \leftarrow custoMelhorIndGkPlus1(Desempilhar_{mutante})$ 
21:       $restart \leftarrow 0$ 
22:    end if
23:     $Desempilhar \leftarrow Desempilhar_{mutante}$ 
24:  end while
25:   $restart \leftarrow restart + 1$ 
26: end while

```

para otimizar a calibração dos elementos. Podemos ver como cada aspecto influencia no algoritmo.

Tabela 1 - Influência nos Parâmetros e Descrições.

Influência em	Nome	Descrição
Regressão simbólica	<i>StackLen</i>	Tamanho máximo da expressão matemática
Regressão simbólica	<i>MaxConst</i>	Quantidade máxima de constantes permitidas
BRKGP	<i>PopulationLen</i>	Quantidade de indivíduos na população
BRKGP	<i>EliteSize</i>	Porcentagem de elites na população
BRKGP	<i>MutantSize</i>	Porcentagem de mutantes na população
BRKGP	<i>EliteBias</i>	Chance de propagação de gene elite
Chamadas ao BRKGP	<i>RestartMax</i>	Reinício do BRKGP com população aleatória
BRKGP	<i>NoImprovementMax</i>	Quantidade máxima de gerações sem melhora
Validação	<i>Validation</i>	Porcentagem da entrada reservada para treino
Aleatório	<i>Seed</i>	Semente do gerador de números aleatórios

Legenda: Descrição da influência de cada parâmetro e sua respectiva descrição

Fonte: Adaptado de [6], 2023.

Na Tabela 2, estão delineados os parâmetros selecionados para esta investigação, os quais foram previamente examinados no trabalho de Guimarães et al [6]. Embora tenham demonstrado uma boa acurácia no referido estudo, é relevante observar que esses parâmetros constituem apenas uma configuração inicial para a presente pesquisa.

Tabela 2 - Parâmetros globais do BRKGP.

Nome	Descrição	Valor adotado
<i>populationLen</i>	Tamanho da população	100
<i>eliteSize</i>	Porcentagem de elites na população	25%
<i>mutantSize</i>	Porcentagem de mutantes na população	10%
<i>eliteBias</i>	Chance de propagação de gene elite	85%

Legenda: Descrição dos parâmetros globais utilizados no algoritmo do BRKGP

Fonte: Adaptado de [6], 2023.

## 5.4 Decoder

A função Decoder é central para o funcionamento do algoritmo, ela quem faz a distinção entre as variáveis e constantes ou até sobre as operações binárias e unárias. Essa função desempenha seu papel na formação e manipulação da pilha, onde são armazenadas

as expressões aritméticas, sua execução é pareada com outras funções que tem importância para a execução do código.

O processo passa pela iteração de uma sequência de cromossomos que representa a expressão a ser avaliada. Uma variável é empregada para rastrear a posição atual na sequência. A função `IsOperation`, descrita pelo Algoritmo 5 é então utilizada para discernir o tipo de operação representada pelo cromossomo em questão, podendo ser binária, unária, de empilhar ou especial.

Algoritmo 5 - Função para identificar qual tipo de operação

```

1: function ISOPERATION(rk, op)
2:   if (op = OpType::Binary) and ( $0 \leq rk < 2500$ ) then return true
3:   end if
4:   if (op = OpType::Unary) and ( $2500 \leq rk < 5000$ ) then return true
5:   end if
6:   if (op = OpType::Push) and ( $5000 \leq rk < 7500$ ) then return true
7:   end if
8:   if (op = OpType::Special) and ( $7500 \leq rk < 10000$ ) then return true
9:   end if
   return false
10: end function

```

A manipulação da pilha ocorre conforme o tipo de operação identificada como mostrado no Algoritmo 6. Operações binárias removem dois elementos da pilha e adicionam o resultado, enquanto operações unárias removem um elemento e adicionam o resultado. Operações do tipo empilhar incrementam a pilha com um novo elemento, podendo ser uma variável ou constante. Operações especiais, por sua vez, não afetam diretamente a pilha.

Algoritmo 6 - Empilhar

```

1: function MAKEPUSH(rk)
2:   if ISOPERATION(rk, OpType::Binary) then ▷ If Binary (<2500), becomes PUSH
3:     return rk + 5000
4:   else if ISOPERATION(rk, OpType::Special) then           ▷ If Special (>=7500),
   becomes PUSH
5:     return rk - 2500
6:   else                                                       ▷ If Unary, perform PUSH
7:     return rk + 2500
8:   end if
9: end function

```

Além disso, a função realiza ajustes específicos para lidar com variáveis e constantes. A parte decimal do cromossomo é utilizada para calcular um identificador dentro do intervalo de variáveis e constantes. A contagem de constantes é atualizada para operações do tipo empilhar seguidas de constantes.

O Algoritmo 7 atua na preparação da pilha, gerenciando operações aritméticas, identificando variáveis e constantes, e, quando aplicável, considerando unidades. Sua implementação reflete a avaliação eficiente de expressões simbólicas. Inicialmente, a função define algumas variáveis, como contador para contar elementos na pilha, contador para contar constantes, contador para operações de menor custo, e outro contador para armazenar unidades associadas, um parâmetro para indicar correções futuras, um contador para o número de variáveis usadas, e parâmetro para rastrear variáveis usadas.

Algoritmo 7 - Calculo\_Indivíduo

```

1: function CALCULO_INDIVIDUO(individuo)
2:   // Definir estruturas de dados necessárias
3:   Pilha ← []
4:   // Iterar sobre os elementos do cromossomo
5:   for cada gene em individuo do
6:     if gene é uma operação binária then
7:       // Realizar operação binária
8:       operando2 ← Desempilhar()
9:       operando1 ← Desempilhar()
10:      resultado ← executar_operação_binária(gene, operando1, operando2)
11:      // Empilhar resultado
12:      Empilhar(resultado)
13:     else if gene é uma operação unária then
14:       // Realizar operação unária
15:       operando ← Desempilhar()
16:       resultado ← executar_operação_unária(gene, operando)
17:       // Empilhar resultado
18:       Empilhar(resultado)
19:     else if gene é uma operação de Empilhar then
20:       // Decidir se é uma variável, constante, ou especial (NOP)
21:       valor ← determinar_valor(gene)
22:       // Empilhar valor
23:       Empilhar(valor)
24:     end if
25:   end for
26:   // Resultado
27:   resultado_final ← Desempilhar()
28:   return resultado_final
29: end function

```

O decoder em um algoritmo genético, pode ser visualizado no Algoritmo8, como o BRKGA, é uma parte fundamental do processo que transforma representações internas abstratas em soluções concretas para um problema específico. No contexto do BRKGA, as soluções candidatas são representadas por sequências de chaves aleatórias. Essas chaves aleatórias são valores contidos no intervalo  $[0, 1]$ .

O processo de decoding é responsável por converter essas chaves aleatórias em so-

luções factíveis e avaliáveis para o problema em questão. Essa transformação é necessária porque as operações de crossover e mutação são realizadas no espaço das chaves aleatórias, uma representação mais conveniente para o algoritmo genético.

Algoritmo 8 - Decoder

```
1: procedure DECODE(população, problema, cenário, semente)
2:   for  $i \leftarrow 0$  to tamanho da população do
3:     if custo do indivíduo  $i = 0$  then
4:       Armazene a semente usada no ajuste da pilha
5:       Realize o ajuste da pilha para modificar as chaves aleatórias
6:       Obtenha informações sobre a pilha após o ajuste ( $si$ )
7:       Atribua o tamanho verdadeiro da pilha (trueStackSize) ao indivíduo  $i$ 
8:       Avalie a solução decodificada usando a função SolutionEvaluator
9:       if problema inclui unidades de saída then
10:        Verifique se a unidade de saída é consistente
11:        if não todas as variáveis  $k$  foram usadas then
12:          Aplique penalidade exponencial
13:        end if
14:      end if
15:    end if
16:  end for
17: end procedure
```

## 6 RESULTADOS

Para a condução da primeira fase de experimentos computacionais deste trabalho, empregou-se um computador equipado com um processador, Intel(R) Core(TM) i7-11800H e 32GB de memória RAM. O sistema operacional utilizado foi o Windows 11, juntamente com o ambiente de desenvolvimento Microsoft Visual Studio Enterprise 2022.

Em Silviu-Marian (2020) [7] foi abordado a aplicação de Regressão Simbólica no âmbito da descoberta de funções físicas, e comparou estes resultados com os que foram obtidos com o software Eureka, baseado em Redes Neurais. Este algoritmo (IA Feynman), utilizou algumas abordagens que aumentaram o seu desempenho, como análise dimensional, ajuste polinomial e força bruta, culminando em resultados altamente promissores.

Tabela 3 - Descrição dos problemas-teste

Problema	Função	Variáveis	Constantes
Distância Euclidiana	$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$	4	0
Massa Relativística	$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}}$	2	2
Lentes Delgadas	$f = \frac{1}{\frac{1}{d_1} + \frac{1}{d_2}}$	3	1
Produto Interno	$\mathbf{A} = A_x \cdot B_x + A_y \cdot B_y + A_z \cdot B_z$	6	0
Força Gravitacional	$\mathbf{F} = \frac{G \cdot m_1 \cdot m_2}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$	6	3
Período Orbital de dois corpos	$t = \frac{2\pi a^{3/2}}{\sqrt{G(m_1 + m_2)}}$	1	5
Campo Elétrico	$E = \frac{q}{4\pi\epsilon_0 r^2}$	2	3
Densidade de Probabilidade	$f = \frac{e^{-\frac{\theta}{2}}}{\sqrt{(2\pi)}}$	1	3
Onda Eletromagnética	$k = \sqrt{\frac{\omega^2}{c^2} - \frac{\pi^2}{n^2}}$	2	2
Distribuição de Bose-Einstein	$n = \frac{1}{e^{\frac{h\omega}{k_B T}} - 1}$	2	4

Legenda: Descrição dos problemas-teste com suas funções correspondentes e o número de variáveis e constantes associadas a cada problema. Problemas retirados de [7]

Fonte: O autor, 2023.

Nesse contexto, este trabalho se dedica a avaliar a eficácia da metodologia de Regressão Simbólica combinada com BRKGP, aplicando em algumas destas funções de interesse retirados do trabalho de Silviu-Marian (2020) [7], os problemas que serão analisados, são funções já consolidadas em livros didáticos, cada problema tem quantidade de variáveis e constantes diferenciadas, as funções estão ilustradas na Tabela 3. Em consonância com tais premissas, optou-se por empregar uma calibração de desempenho que já havia sido previamente avaliada com sucesso em Guimarães et al. [6].

O cálculo do erro considerado foi o *Root Mean Square Error*, sendo utilizado o  $y_i$

sendo os valores observados, e o  $\hat{y}_i$  os valores previstos:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

A seleção dos problemas que serão testados foi baseada na complexidade intrínseca de cada função. Fatores como o número de variáveis e o valor das constantes foram minuciosamente ponderados no processo de escolha das funções que compuseram o conjunto de problemas, assegurando que houvesse diversidade representativa tanto nas operações como no domínio pertencente da função. Na Tabela 4 temos descritas as constantes que vão ser utilizadas em alguns problemas.

Tabela 4 - Constantes Utilizadas no Estudo

Constante	Valores
<i>Velocidade da Luz (c)</i>	$3 \times 10^8$ m/s
<i>Constante Gravitacional (G)</i>	$6.67408 \times 10^{-11}$ m <sup>3</sup> /kg/s <sup>2</sup>
<i>Massa da Terra (m<sub>1</sub>)</i>	$5.972 \times 10^{24}$ kg
<i>Massa da Lua (m<sub>2</sub>)</i>	$7.34 \times 10^{22}$ kg
<i>Número de Euler (e)</i>	2.71828
<i>Número Pi (π)</i>	3.14159
<i>Permissividade do Vácuo (ε<sub>0</sub>)</i>	$8.854 \times 10^{-12}$ F/m
<i>Constante de Boltzmann (k<sub>b</sub>)</i>	$1.380649 \times 10^{-23}$ J/K
<i>Constante de Planck (h)</i>	$6.62607015 \times 10^{-34}$ J · s

Legenda: Tabela de Constantes utilizadas durante os experimentos computacionais

Fonte: O autor, 2023.

No código desenvolvido, foram implementadas operações binárias que englobam as quatro operações matemáticas fundamentais: soma, subtração, multiplicação e divisão. Adicionalmente, foram incorporadas operações unárias, tais como raiz quadrada e potenciação. A escolha criteriosa dessas operações baseou-se em uma análise abrangente de uma amostra composta por dez funções distintas. Essa seleção foi orientada pelo fato de que todas essas funções, representativas do conjunto de dados considerado, fazem uso de alguma dessas operações.

## 6.1 Distância Euclidiana

A função da distância euclidiana é uma ferramenta fundamental em diversas áreas da física, proporcionando uma medida direta da separação entre dois pontos em um espaço euclidiano. Esta função, que encontra sua origem na geometria euclidiana, é uma

representação matemática da distância “mais curta” entre dois pontos em um plano ou espaço tridimensional. Em seu famoso livro *"The Feynman Lectures on Physics,"* [19] o físico Richard Feynman explora a aplicação dessa função em contextos físicos, destacando sua importância na formulação e resolução de problemas.

A expressão matemática para a distância euclidiana entre dois pontos  $P(x_1, y_1)$  e  $Q(x_2, y_2)$  no plano é dada por:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

Na física, a distância euclidiana é frequentemente empregada em problemas que envolvem movimento de partículas, localização de objetos e análise espacial.

A função da Distância Euclidiana tem quatro variáveis e nenhuma constante. Foi escolhido o intervalo dos números inteiros de  $-10$  a  $10$  para ser o domínio das variáveis, essa escolha se deve a facilidade do cálculo, além de ser útil no contexto computacional.

No contexto da função da Distância Euclidiana, a Tabela 5 ilustra uma consistência no comportamento temporal. À medida que o tamanho da amostra aumenta, o tempo de processamento também cresce. É importante observar, durante esses testes, que há um aumento no Desvio Padrão à medida que a amostra de dados se expande. Este fenômeno reflete a maior variabilidade do conjunto de dados, uma vez que o uso de mais dados resulta em uma distribuição mais ampla.

Esse aumento na variabilidade também se reflete no intervalo entre os valores máximos e mínimos coletados durante o experimento. Notavelmente, esse intervalo, que separa os extremos da distribuição, tende a aumentar com o aumento da amostra, impactando diretamente a média.

Tabela 5 - Tempo entre 5 execuções para a Distância Euclidiana

Função	Amostras (n)			
	10	20	40	80
Média	942,88	1534,38	2349,04	4347,44
Máximo	1165,35	1809,66	2902,55	5934,29
Mínimo	801,99	1193,53	1791,24	3490,88
Desvio Padrão	155,12	220,82	420,55	971,68

Legenda: Resultados de Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a Distância Euclidiana com base em 5 execuções

Fonte: O autor, 2023.

No que diz respeito ao erro na Tabela 6, embora o erro se mostre controlado, os resultados finais obtidos, geraram funções que excluiu uma das variáveis, ou seja, uma função com apenas três das variáveis originais. Uma possível causa desse comportamento

se deve a amostra não ser homogênea no domínio de valores das variáveis. Notamos uma tendência interessante no Desvio Padrão, que parece diminuir à medida que o tamanho da amostra aumenta.

Podemos concluir que, à medida que o tamanho da amostra de teste aumenta, é esperado que o tempo exiba maior variação, enquanto a variação no erro tende a diminuir. Ao analisar a média dos erros, observamos um padrão mais semelhante ao comportamento temporal, mostrando um aumento à medida que a amostra de dados se expande. É notável que, à medida que mais dados são inseridos no experimento, o intervalo entre os valores mínimo e máximo diminui, o que contribui para os valores observados de Média e Desvio Padrão.

Tabela 6 - RMSE entre 5 execuções para a Distância Euclidiana

Função	Amostras (n)			
	10	20	40	80
Média	3,26	3,52	3,59	3,76
Máximo	4,52	4,19	4,25	4,14
Mínimo	2,12	2,62	3,38	2,93
Desvio Padrão	1,07	0,69	0,37	0,49

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a Distância Euclidiana com base em 5 execuções

Fonte: O autor, 2023.

Ao analisar os resultados provenientes das funções geradas pelo algoritmo, constatamos que nenhuma delas conseguiu replicar de precisamente a forma original da função da Distância Euclidiana. É digno de nota que a maioria das funções geradas incorpora apenas três das variáveis presentes no problema, revelando um padrão de comportamento atípico. Entre os resultados, a função que obteve o menor erro foi:

$$\frac{4x_2 - y_1}{\sqrt{x_2}} - x_2 \quad (3)$$

Contudo, sob a perspectiva do Paradigma *Less Is More*, podemos interpretar esses resultados como aceitáveis, pois refletem a simplificação das expressões, mesmo que não atinjam a representação exata da função original. Este paradigma prioriza soluções mais elegantes mesmo que não reproduzam precisamente a complexidade da função original. Nesse contexto, a capacidade do algoritmo em gerar soluções mais simples pode ser vista como um aspecto positivo.

Analisando este resultado em comparação com os demais, é evidente que a presença de operações unárias exerce um impacto considerável no erro. Os erros acumulativos associados a operações como potenciação e raiz quadrada resultam em um aumento signi-

ficativo na distância em relação ao resultado original. Além disso, vale ressaltar que este resultado não fez uso de todas as variáveis disponíveis.

A escolha do intervalo do domínio também pode ter desempenhado um papel importante. Notavelmente, se o algoritmo fosse compelido a incorporar todas as variáveis, poderia ter demandado um esforço computacional maior para alcançar o resultado mais próximo. Entretanto, é crucial analisar se esse *trade off* [68] justifica-se.

Comparando-o com os demais resultados, podemos afirmar que este desempenho foi satisfatório, considerando que nos outros testes chegamos a erros extremamente altos.

## 6.2 Massa Relativística

A teoria da relatividade, proposta por Albert Einstein no início do século XX, revolucionou fundamentalmente nossa compreensão do espaço, tempo e, por conseguinte, da energia e da massa. Um aspecto central dessa teoria é a famosa função  $E = mc^2$  que estabelece uma relação profunda entre a energia ( $E$ ) e a massa ( $m$ ) de um objeto, indicando que a massa pode ser convertida em energia e vice-versa.

No âmbito da física relativística, a massa de um objeto em movimento não é constante, como postulado pela física clássica, mas depende de sua velocidade. Para explorar essa ideia, adentramos na função da massa relativística, um dos pilares da teoria de Einstein. Essa função é expressa por:

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (4)$$

onde  $m_0$  é a massa de repouso do objeto, ( $v$ ) é sua velocidade e ( $c$ ) é a velocidade da luz no vácuo.

Para compreender melhor essa formulação, vale-nos a analogia apresentada por Richard Feynman em suas obras [19]. Feynman, destaca que à medida que um objeto se aproxima da velocidade da luz, sua massa efetiva aumenta significativamente. Essa mudança na massa implica que a energia necessária para continuar acelerando o objeto também aumenta, aproximando-se da velocidade da luz. Nesse cenário, seria requerida uma quantidade infinita de energia para atingir a velocidade da luz, tornando essa última um limite intransponível.

A função da massa relativística tem implicações profundas na compreensão de fenômenos físicos a altas velocidades, como aqueles observados em partículas subatômicas aceleradas em aceleradores de partículas. A contribuição de Feynman ao explicar esses conceitos de maneira acessível e envolvente reflete-se na compreensão ampla dessa função e, por conseguinte, da teoria da relatividade.

A função da massa relativística tem duas variáveis, sendo o  $v$  ( velocidade do objeto) e o  $m_0$  ( Massa de repouso do objeto), e duas constantes, sendo o  $c$  ( Velocidade da luz no vácuo) e o 1 da função original. Foi escolhido o intervalo dos números naturais de 1 a 100 para ser o domínio do  $m_0$ , enquanto para  $v$  foi escolhido o intervalo dos números naturais de 1 a 300.000.000. O intervalo de 1 a 100 para  $m_0$  foi escolhido para abranger uma gama de valores razoáveis para massas de objetos em repouso, sem ultrapassar valores excessivamente grandes que pudessem comprometer a eficiência computacional. Ao escolher um intervalo de 1 a 300.000.000 cobrimos um espectro amplo de velocidades possíveis para objetos em movimento para  $v$ . Essa função possui uma constante que não sofre variação, chamada de  $c$ , seu valor é 300.000.000. Foi escolhido 300.000.000 ao invés de 299.792.458 de modo a evitar erros que pudessem comprometer o teste computacional.

Embora a escolha do valor de 300.000.000 seja uma aproximação, ela é suficientemente precisa para os propósitos do teste, garantindo, ao mesmo tempo, uma execução mais suave e menos propensa a erros numéricos que poderiam afetar a validade dos resultados obtidos no contexto do experimento em questão.

No contexto da função da Massa Relativística, a Tabela 7 ilustra um comportamento temporal coerente, evidenciando que o tempo de processamento aumenta conforme o tamanho da amostra. É importante destacar que, durante os testes, observa-se um aumento no Desvio Padrão à medida que a amostra de dados cresce. É notável que um maior volume de dados pode resultar em uma maior variabilidade no conjunto de dados. Os resultados, nesse aspecto, não diferem dos obtidos anteriormente na função da Distância Euclidiana em termos do crescimento da amostra. No entanto, nota-se um tempo de processamento maior em comparação com o teste anterior. O mesmo padrão observado na Distância Euclidiana se reflete na variação dos mínimos e máximos. O aumento na variabilidade dos dados é um resultado direto do maior número de pontos, contribuindo para um tempo de processamento mais extenso.

Tabela 7 - Tempo médio (s) entre 5 execuções para a massa relativística

Função	Amostras (n)			
	10	20	40	80
Média	1640,66	3836,85	6025,90	10316,39
Máximo	1872,14	4770,70	8963,29	12898,30
Mínimo	1318,00	2450,93	4202,51	7758,15
Desvio Padrão	216,67	889,65	1966,46	1852,39

Legenda: Resultados de Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a massa relativística com base em 5 execuções

Fonte: O autor, 2023.

No que tange ao erro, a Tabela 8 revela uma dinâmica peculiar. Os resultados de erro mostram-se mais favoráveis para uma amostra de tamanho 20, sendo este o conjunto

de dados que mais se aproximou do erro nulo, indicando uma variação mínima entre as diversas amostras testadas e o menor erro encontrado. Esse comportamento irregular sugere que a função da Massa Relativística apresenta uma sensibilidade particular à variação do tamanho amostral, possivelmente relacionada à complexidade das operações envolvidas e o intervalo utilizado para as variáveis.

É interessante observar também o comportamento quando lidamos com uma amostra de tamanho 40, onde a média dos erros se mostrou mais elevada. No entanto, na amostra de tamanho 80, não se observou esse crescimento irregular na média dos erros, o que pode indicar uma estabilização ou mesmo uma diminuição dessa média. Esse padrão de comportamento nos sugere que a relação entre o tamanho da amostra e o erro na função da Massa Relativística não segue uma tendência linear, destacando uma complexidade na influência do tamanho amostral sobre o desempenho da função.

Tabela 8 - RMSE entre 5 execuções para a massa relativística

Função	Amostras (n)			
	10	20	40	80
Média	2,83	1,39	8,55	4,46
Máximo	3,00	1,90	11,31	5,83
Mínimo	2,69	0,97	6,82	3,63
Desvio Padrão	0,13	0,38	1,73	0,86

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a massa relativística com base em 5 execuções

Fonte: O autor, 2023.

O algoritmo revelou funções de extrema complexidade devido aos elevados valores tanto da velocidade quanto da constante da velocidade da luz no vácuo. Essa complexidade crescente nas funções, embora não tenha resultado em valores excessivamente altos para o *Root Mean Square Error*, destaca-se como uma consequência notável. Esse fenômeno sugere que, com valores mais elevados para as variáveis em questão, as funções geradas pelo algoritmo tendem a assumir formas mais intrincadas, refletindo possivelmente a natureza intrincada das interações entre os parâmetros da função.

Após a realização da bateria de testes para a função da Massa Relativística, observamos que os erros obtidos foram muito pequenos, além disso, nenhum dos resultados chegou a forma original da função. No entanto, é notável que a maioria das funções geradas apresentou uma complexidade considerável. Essa complexidade pode ser atribuída, em parte, às constantes com um número significativo de casas decimais, o que contribui para a intrincada natureza das funções, além também das já ditas operações numéricas realizadas na função.

Dentre as funções avaliadas, aquela que obteve o menor erro foi a seguinte:

$$\frac{\left(\frac{v^2\left(\frac{m_0 v(2v-4m_0^2)}{3 \times 10^8} + v + 3 \times 10^8\right)}{9 \times 10^{16}} + 3 \times 10^8\right)}{m_0^2 - v + 6 \times 10^8} + m_0 \quad (5)$$

A função acima foi a função que mais se aproximou da função original. O resultado obtido, se mostrou uma função complexa, embora com o erro extremamente reduzido. Podemos levantar questões interessantes sobre a relação entre a precisão dos resultados e a complexidade das funções geradas pelo algoritmo.

Ao analisar os resultados obtidos em comparação com os demais, observamos que, embora o resultado não siga o paradigma do *Less Is More*, alcançou um erro significativamente pequeno. Na função original, tínhamos pelo menos duas operações unárias de potenciação e uma raiz, e essas operações exercem considerável influência nos resultados, especialmente em determinados domínios.

Apesar de o algoritmo ter utilizado todas as variáveis do problema, não conseguiu se aproximar substancialmente da função original.

### 6.3 Lentes Delgadas

A função das lentes delgadas é uma expressão fundamental na óptica, uma área da física que explora o comportamento da luz. Segundo o físico Richard Feynman, em seu livro *The Feynman Lectures on Physics* [19], as lentes delgadas desempenham um papel crucial na compreensão e manipulação da luz em diversos contextos científicos e tecnológicos.

A função em questão é derivada da lei da refração, que descreve como a luz muda de direção ao passar de um meio para outro. Em lentes delgadas, que têm uma espessura muito menor do que seu raio de curvatura, a função relaciona a distância focal da lente ( $f$ ), a distância do objeto até a lente ( $d_1$ ), distância da imagem até a lente ( $d_2$ ) e o índice de refração do meio onde a lente está inserida ( $n$ ). Essa relação é expressa pela fórmula:

$$f = \frac{1}{\frac{1}{d_1} + \frac{n}{d_2}} \quad (6)$$

Essa função é essencial para entender como lentes delgadas convergem ou divergem a luz, permitindo a criação de sistemas ópticos avançados, como câmeras, microscópios e telescópios. A função das Lentes Delgadas tem três variáveis e apenas uma constante, sendo esse o valor 1 da função original. Foi escolhido o intervalo dos números racionais de 1 a 3 como o domínio das variáveis. Essa escolha visa simplificar os cálculos, dependendo

menos recursos computacionais. Além da praticidade computacional, levamos em conta a aplicação prática da função, considerando fenômenos físicos específicos. Considerando os fenômenos físicos que representam o domínio para a situação específica, onde  $n$  considera materiais comuns como o vidro, e  $d_1$  e  $d_2$  consideram imagens posicionadas relativamente perto ou distante da lente. Essa abordagem visa equilibrar eficiência computacional com relevância prática na resolução da função para a situação específica considerada.

Na análise mais aprofundada da função das Lentes Delgadas, conforme apresentado na Tabela 9, observamos um desempenho notável em relação ao tempo de execução. À medida que ampliamos o tamanho da amostra, percebemos uma tendência de aumento nas variações temporais, indicando uma correlação positiva com o crescimento dos dados de entrada. Esse comportamento sugere consistência e previsibilidade na execução do algoritmo, aspectos cruciais para a confiabilidade dos resultados.

Além disso, vale ressaltar o crescimento expressivo do Desvio Padrão à medida que o tamanho da amostra aumenta. Esse aumento é um reflexo direto do incremento no número de pontos, assim como do alargamento do intervalo entre os valores mínimos e máximos obtidos nos testes. Essa observação destaca não apenas a variabilidade dos resultados, mas também a sensibilidade da função das Lentes Delgadas às mudanças na amostragem.

Tabela 9 - Tempo médio (s) entre 5 execuções para as Lentes Delgadas

Função	Amostras (n)			
	10	20	40	80
Média	780,56	1283,58	2303,82	4409,40
Máximo	864,70	1447,87	2478,12	5645,27
Mínimo	722,56	1167,18	2135,71	3889,72
Desvio Padrão	60,19	120,63	139,30	727,21

Legenda: Resultados de Tempo médio (s), Máximo, Mínimo e Desvio Padrão para as Lentes Delgadas com base em 5 execuções

Fonte: O autor, 2023.

No âmbito do erro, a Tabela 10 revela um desempenho excepcional. Ao atingir um erro nulo em todas as amostras testadas, o algoritmo demonstra uma incrível capacidade de discernir a solução ótima, mesmo em cenários de amostragem reduzida. Esse resultado ressalta a robustez e a precisão do algoritmo, evidenciando sua alta competência na identificação de soluções ótimas mesmo quando confrontado com um conjunto limitado de dados. Essa notável capacidade de generalização destaca a eficácia do algoritmo nas situações em que a disponibilidade de dados pode ser um desafio.

As funções geradas pelo algoritmo demonstraram um desempenho notável, uma vez que todas as variáveis foram incorporadas, resultando em erros nulos em todos os testes. Apesar desse resultado incomum, o algoritmo obteve respostas satisfatórias para

Tabela 10 - RMSE entre 5 execuções para as Lentes Delgadas

Função	Amostras (n)			
	10	20	40	80
Média	0,00	0,00	0,00	0,00
Máximo	0,00	0,00	0,00	0,00
Mínimo	0,00	0,00	0,00	0,00
Desvio Padrão	0,00	0,00	0,00	0,00

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para as Lentes Delgadas com base em 5 execuções

Fonte: O autor, 2023.

resolver o problema das Lentes Delgadas, apresentando resultados favoráveis na maioria dos casos, embora alguns tenham apresentado desafios específicos, algumas funções encontradas embora tenham obtido erro nulo, foram funções que se adequaram bem apenas ao domínio fornecido.

Após uma minuciosa bateria de testes realizados para avaliar a eficácia do algoritmo na resolução da função das Lentes Delgadas, destacou-se sua notável precisão ao acertar a função original em nove instâncias distintas das vinte execuções. Intrigantemente, essa assertividade apresentou-se de maneira mais evidente em amostras de maior magnitude, indicando uma tendência positiva à medida que o número de pontos cresce. Esse padrão consistente de desempenho sugere uma confiabilidade robusta do algoritmo, evidenciando sua capacidade de lidar de maneira coerente com desafios mais exigentes e intrincados.

Ao analisar este resultado em comparação com os demais, é notável um desempenho surpreendentemente bom. No entanto, vale ressaltar que esta função se utiliza apenas de operações binárias. A simplicidade inerente a essas operações pode ter facilitado o processo para o algoritmo, contribuindo para que alcançasse resultados mais próximos da função original. É válido considerar que a escolha do domínio também pode ter desempenhado um papel crucial nesse resultado positivo. Embora a eficácia do algoritmo possa variar em intervalos maiores, o uso exclusivo de operações binárias sugere uma maior probabilidade de obtenção de resultados com erro nulo.

## 6.4 Produto Interno

A função do produto interno desempenha um papel crucial na física, sendo uma ferramenta fundamental para compreender e descrever fenômenos que envolvem vetores e espaços vetoriais. Richard Feynman, em seu livro *The Feynman Lectures on Physics*[19], destaca a importância desta função e sua aplicação em diversos campos da física teórica.

No contexto da álgebra linear tridimensional, o produto interno é uma operação que combina dois vetores tridimensionais, fornecendo uma medida da similaridade entre eles. A função do produto interno entre dois vetores tridimensionais  $\mathbf{A}$  e  $\mathbf{B}$  é dada por:

$$\mathbf{A} \cdot \mathbf{B} = A_x \cdot B_x + A_y \cdot B_y + A_z \cdot B_z \quad (7)$$

Esta função reflete não apenas a multiplicação elemento a elemento dos componentes dos vetores, mas também a soma destes produtos, fornecendo um único escalar como resultado. No contexto físico, esta operação é frequentemente utilizada para calcular o trabalho realizado por uma força ao longo de uma trajetória, entre outras aplicações.

A importância do produto interno se estende para áreas como a teoria eletromagnética, onde é utilizado para calcular o fluxo de um campo elétrico mediante uma superfície. Além disso, em mecânica quântica, o produto interno é fundamental para definir o conceito de estados normalizados, essenciais para interpretações probabilísticas e previsões precisas. A função do Produto Interno tem seis variáveis e nenhuma constante. Foi escolhido o intervalo dos números inteiros de  $-10$  a  $10$  para ser o domínio das variáveis, com o intuito de que essa escolha facilite os cálculos, e torne as soluções mais gerenciáveis, para o contexto computacional.

Na análise detalhada da função do Produto Interno, os resultados obtidos, conforme evidenciados na Tabela 11, revelam uma eficiência notável no que diz respeito ao tempo de execução. Este desempenho otimizado é caracterizado por tempos de processamento consistentes, indicando a robustez do algoritmo em diferentes configurações amostrais. Surpreendentemente, o tempo de execução não apresentou sensibilidade significativa ao aumento do tamanho da amostra.

Tabela 11 - Tempo médio (s) entre 5 execuções para o Produto Interno

Função	Amostras (n)			
	10	20	40	80
Média	1272,59	1243,92	2088,62	3950,93
Máximo	1698,42	1384,56	2459,86	5149,38
Mínimo	844,29	1144,50	1850,11	3302,33
Desvio Padrão	308,41	103,63	255,44	737,20

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para o Produto Interno com base em 5 execuções

Fonte: O autor, 2023.

Ao examinarmos a transição de uma amostra de tamanho 10 para 20, observamos uma diminuição no intervalo entre os valores mínimos e máximos, resultando em uma redução do Desvio Padrão. Embora os tempos médios para amostras de tamanho 10 e 20

tenham sido comparativamente semelhantes, a redução na variação reflete uma resposta favorável à expansão da amostra. Este fenômeno destaca a adaptabilidade do algoritmo a diferentes volumes de dados, garantindo um desempenho estável e consistente mesmo com amostras maiores.

Adicionalmente, é relevante notar que o erro associado aos resultados permanece em níveis satisfatórios, como podemos ver na Tabela 12, com exceção da primeira amostra de tamanho 10, que apresentou um erro consideravelmente elevado em uma das execuções. Contudo, ao analisar as amostras subsequentes, a eficiência e a precisão do algoritmo na resolução dessa função tornam-se evidentes, reforçando a robustez da abordagem utilizada. A estabilidade dos resultados, mesmo diante da variação no tamanho amostral, destaca a confiabilidade do método empregado para validar a função do Produto Interno.

Tabela 12 - RMSE entre 5 execuções para o Produto Interno

Função	Amostras (n)			
	10	20	40	80
Média	3,38	0,00	0,00	0,00
Máximo	8,84	0,00	0,00	0,00
Mínimo	0,00	0,00	0,00	0,00
Desvio Padrão	4,12	0,00	0,00	0,00

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para o Produto Interno com base em 5 execuções

Fonte: O autor, 2023.

No decorrer da análise dos resultados produzidos pelo algoritmo, observou-se que o mesmo obteve êxito em sete ocasiões ao determinar corretamente a função do Produto Interno. Essa assertividade mostrou-se mais proeminente em amostras de maior magnitude. Esse padrão de desempenho sugere uma maior confiabilidade e eficácia do algoritmo à medida que o volume de dados aumenta, evidenciando sua capacidade de lidar de maneira consistente com problemas mais complexos.

Ao analisar os resultados do Produto Interno em comparação com os demais, observamos um padrão de comportamento semelhante ao encontrado na função das Lentes Delgadas. A predominância de operações binárias parece simplificar o processo para o algoritmo, tornando mais fácil a busca por respostas mais próximas da função original. A escolha do domínio também pode ter desempenhado um papel significativo na capacidade do algoritmo de encontrar soluções.

Além disso, as operações utilizadas na função do Produto Interno não geraram indeterminações, ao contrário da função das Lentes Delgadas, que envolvia operações com restrições, como a divisão. Essa diferença pode ser um dos motivos pelos quais o resultado de tempo para a função do Produto Interno foi tão mais baixo em comparação com a função das Lentes Delgadas para tamanhos amostrais maiores. Vale ressaltar

que, em funções propensas a indeterminações, os resultados tendem a exigir um custo computacional maior, especialmente quando o tamanho da amostra é ampliado.

## 6.5 Força Gravitacional

A função da Força Gravitacional é uma expressão fundamental na física que descreve a atração gravitacional entre duas massas. Essa lei foi formalizada por Sir Isaac Newton e continua sendo uma pedra angular da física clássica.

$$F = \frac{G \cdot m_1 \cdot m_2}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (8)$$

A relação entre a força gravitacional ( $F$ ), as massas ( $m_1$ ) e ( $m_2$ ) e as posições relativas ( $x_2, y_2, z_2$ ) e ( $x_1, y_1, z_1$ ) das duas massas no espaço tridimensional.

Ao aplicar esta função, é possível entender como a força gravitacional age entre corpos massivos, influenciando movimentos celestiais, órbitas planetárias e outros fenômenos cósmicos. A constante ( $G$ ) na função é a constante gravitacional universal, um valor que conecta a intensidade da força gravitacional ao produto das massas envolvidas e à distância entre elas.

É interessante destacar que a função da Força Gravitacional foi uma das primeiras descrições matemáticas bem-sucedidas do universo. Ela foi um componente crucial da revolução científica liderada por Newton no século XVII.

A compreensão dessa função é vital em campos como a astronomia, onde é utilizada para prever órbitas de planetas e satélites. Além disso, ela é um pilar na compreensão de eventos astrofísicos, como a formação de estrelas e galáxias.

É relevante contextualizar a função da Força Gravitacional dentro do arcabouço mais amplo da física newtoniana. O livro “The Feynman Lectures on Physics’ [19]’ de Richard Feynman pode ser um recurso valioso para explorar não apenas a função, mas também as implicações filosóficas e conceituais da lei da gravitação.

Feynman, em sua abordagem única, oferece percepções que transcendem a matemática, incorporando uma compreensão mais profunda das interações fundamentais que moldam o universo. Dessa forma, o estudo da Força Gravitacional se torna não apenas uma exploração matemática, mas uma jornada para compreender as leis que governam o cosmos. A função da Força Gravitacional tem seis variáveis e três constantes, sendo as constantes da função,  $G$  que representa a constante de gravitação universal, com o valor de  $6,67 \times 10^{-11}$ , para as massas foram escolhidas considerando o estudo da Terra e Lua, logo foram utilizadas as massas dos respectivos corpos celestes, sendo a massa do primeiro igual a  $5,972 \times 10^{24}$  e a massa do segundo igual a  $7,342 \times 10^{22}$ . Foi escolhido o intervalo

dos números inteiros de  $-192200$  a  $192200$  para ser o domínio das variáveis  $x_1, x_2, y_2$  e  $y_1$ , visto que esses valores representam a distância entre a Terra e a Lua, já para o  $z_1$  e  $z_2$  o intervalo de  $-12750$  a  $12750$ , com o intuito de que essa escolha represente o dado evento de forma real, e torne as soluções mais próximas de um experimento mesmo que esses valores computacionalmente sejam mais custosos.

No contexto da função da Força Gravitacional, a Tabela 13, observou-se um impacto notável nos resultados temporais, à medida que o número de variáveis aumenta, exercendo uma influência significativa sobre o desempenho do algoritmo. Embora o padrão de tempo de execução nas amostras demonstre um aumento, uma análise mais detalhada revela uma variação notável, especialmente na amostra de 80, que apresentou um resultado abaixo do esperado, quebrando a proporcionalidade na variação observada nas amostras de 10 a 40. No entanto, a média dos tempos ainda reflete uma proporcionalidade geral nos resultados.

Tabela 13 - Tempo médio (s) entre 5 execuções para a Força Gravitacional

Função	Amostras (n)			
	10	20	40	80
Média	1923,19	2597,09	5482,43	6977,52
Máximo	2718,98	4470,54	7907,41	7890,18
Mínimo	1145,61	1579,87	3987,36	5725,09
Desvio Padrão	660,79	1141,76	1541,39	807,25

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a Força Gravitacional com base em 5 execuções

Fonte: O autor, 2023.

No contexto da análise de erro, a Tabela 14 revela um padrão consistente de elevada imprecisão em todos os casos, destacando uma notável sensibilidade do algoritmo ao tamanho da amostra, ao intervalo dos dados e aos erros de truncamento. Esta tendência fica evidente quando observamos os resultados, onde o algoritmo apresenta desafios em explorar soluções mais otimizadas, muitas vezes ficando estagnado em ótimos locais.

Essa limitação é agravada pelos erros numéricos decorrentes das constantes elevadas, restringindo a capacidade de busca do algoritmo. Vale ressaltar que, em termos da média dos erros, a amostra de tamanho 20 destaca-se ao apresentar o menor erro em comparação com os outros conjuntos, embora o resultado mais satisfatório tenha sido obtido com a amostra de tamanho 10.

Uma observação intrigante é a redução do Desvio Padrão à medida que o tamanho da amostra aumenta, sugerindo uma tendência dos dados a diminuir a variação entre os valores mínimos e máximos. Este fenômeno pode ser interpretado como uma estabilização nas respostas do algoritmo à medida que a quantidade de dados aumenta, indicando uma maior consistência nos resultados.

Tabela 14 - RMSE entre 5 execuções para a Força Gravitacional

Função	Amostras (n)			
	10	20	40	80
Média	$1,71 \times 10^{27}$	$8,22 \times 10^{26}$	$1,01 \times 10^{27}$	$2,19 \times 10^{27}$
Máximo	$5,58 \times 10^{27}$	$1,09 \times 10^{27}$	$1,06 \times 10^{27}$	$2,28 \times 10^{27}$
Mínimo	$4,14 \times 10^{26}$	$6,37 \times 10^{26}$	$9,41 \times 10^{26}$	$2,08 \times 10^{27}$
Desvio Padrão	$2,18 \times 10^{27}$	$1,87 \times 10^{26}$	$4,67 \times 10^{25}$	$8,85 \times 10^{25}$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a Força Gravitacional com base em 5 execuções

Fonte: O autor, 2023.

Durante o período de extensivos testes, observou-se que a função da Força Gravitacional registrou o maior índice de erro em comparação com as demais funções do experimento. Essa ocorrência pode ser atribuída à sua suscetibilidade a imprecisões numéricas, especialmente quando envolvidas constantes de magnitude considerável. Essa limitação pode estar relacionada à dificuldade do algoritmo em lidar eficientemente com valores numéricos extensos, impactando diretamente na precisão dos resultados. Dentre os resultados gerados pelo algoritmo, a função que obteve o menor erro para representar a função foi expressa abaixo:

$$\frac{\left( \frac{((2x_2)z_2 \times 10^{10} + (59720z_2)^2 + (10^{10})^2)z_1 - (x_2^2 + 10^{10})^2}{z_2^2} - 10^{10} \right)^2}{x_1 - x_2} \quad (9)$$

Essa constatação ressalta a importância de considerar a sensibilidade do algoritmo a variáveis específicas e a necessidade de ajustes ou otimizações para aprimorar sua robustez em face de constantes numéricas expressivas.

Ao examinarmos o desempenho da função da Força Gravitacional em comparação com os outros resultados, é evidente o impacto das operações unárias, juntamente com a influência dos domínios escolhidos. A presença dessas operações pode ter contribuído para desafios adicionais na busca por uma solução próxima à função original.

É importante destacar que, ao selecionar um fenômeno específico dentro do contexto da Força Gravitacional, com um intervalo mais equilibrado entre os números, o algoritmo poderia ter obtido resultados mais favoráveis. A sensibilidade ao intervalo escolhido é uma consideração crucial, pois pode influenciar significativamente a capacidade do algoritmo de se ajustar adequadamente aos dados e, assim, alcançar uma maior precisão na modelagem.

## 6.6 Período Orbital de dois Corpos

A dinâmica orbital de dois corpos é um tema fundamental na física celeste, destacando-se pela aplicação direta da Lei da Gravitação Universal de Newton. Essa função desempenha um papel crucial na compreensão dos movimentos cíclicos que governam sistemas binários no cosmos. O período orbital, representado pela função:

$$t = \frac{2\pi d^{3/2}}{\sqrt{G(m_1 + m_2)}} \quad (10)$$

Em seu livro *Classical Mechanics* [69] (Mecânica Classica), Goldstein explora detalhadamente os princípios da mecânica clássica, proporcionando uma base teórica robusta para a análise de sistemas físicos complexos. Dentro desse contexto, a função do período orbital surge como uma expressão elegante que encapsula as forças gravitacionais interativas entre dois corpos celestes.

O período orbital,  $(t)$ , denota o tempo necessário para que dois corpos completem uma órbita ao redor de um centro de massa comum. A distância média entre esses corpos é representada por  $d$ , enquanto a constante gravitacional universal é representado por  $G$ . As massas dos corpos são indicadas por  $m_1$  e  $m_2$ , respectivamente. A função do período orbital de dois corpos, embasada nas contribuições teóricas de Goldstein e fundamentada nos princípios de Newton, proporciona uma compreensão profunda dos movimentos orbitais no universo.

A função em questão envolve uma única variável, representada por  $d$ , e cinco constantes fundamentais. A variável  $d$  abrange o intervalo dos números naturais entre 363104 e 405696, delimitando a média da distância entre a Lua e a Terra. As constantes são denotadas por  $\pi$ ,  $G$ ,  $m_1$ ,  $m_2$  e o valor 2 da função original, assumindo os valores 3.1415926535,  $6.67408 \times 10^{-11}$ ,  $5.972 \times 10^{24}$  (massa da Terra) e  $7.34 \times 10^{22}$  (massa da Lua), respectivamente. A escolha cuidadosa desse intervalo decorre da necessidade de considerar a distância real entre a Lua e a Terra, tornando a análise mais precisa e contextualizada. Dentre as funções extraídas do artigo de [7], esta é a única proveniente do livro de Goldstein, não sendo parte das 100 funções principais, mas sim das 30 funções adicionais mencionadas no mesmo artigo. Essas funções adicionais foram retiradas de diversas obras de física, sendo esta especificamente originária do livro de Goldstein.

Ao analisar os resultados na Tabela 15, é evidente um padrão de crescimento tanto no tempo médio quanto no desvio padrão. Notavelmente, as amostras de 10 e 20 apresentaram resultados bastante próximos, indicando um comportamento atípico. No entanto, ao examinarmos os valores mínimos e máximos, percebemos um aumento no intervalo da amostra de 20, sugerindo uma consistência nos resultados entre essas duas amostras. Isso nos leva à conclusão de que, entre essas amostras específicas, a diferença

Tabela 15 - Tempo médio (s) entre 5 execuções para o Período Orbital de dois Corpos

Função	Amostras (n)			
	10	20	40	80
Média	2163,15	2905,35	8107,59	10465,69
Máximo	3247,47	3643,45	12253,70	18501,20
Mínimo	1140,37	1909,90	4539,79	6589,38
Desvio Padrão	831,46	822,41	2759,18	4722,36

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para o Período Orbital de dois Corpos com base em 5 execuções

Fonte: O autor, 2023.

não é significativa, reforçando a tendência de crescimento constante nos dados de tempo.

Ao analisar o erro na Tabela 16, notamos que os valores de erro são extremamente pequenos, indicando que o algoritmo gerou funções muito próximas do esperado, quase sem erro aparente. Contudo, ao examinarmos a média dos erros, observamos um comportamento distinto, onde as amostras 10 e 40, assim como as amostras 20 e 80, apresentaram erros semelhantes. Esse padrão não é refletido no Desvio Padrão, que demonstra uma variabilidade inconsistente. Ao considerarmos os intervalos de mínimos e máximos, notamos uma tendência de crescimento, sugerindo que, com mais testes, essa tendência pode se tornar mais evidente. Esse fenômeno pode ser interpretado como uma indicação da sensibilidade do sistema a variações no tamanho da amostra.

Tabela 16 - RMSE entre 5 execuções para o Período Orbital de dois Corpos

Função	Amostras (n)			
	10	20	40	80
Média	$1,27 \times 10^{-01}$	$2,00 \times 10^{-01}$	$1,39 \times 10^{-01}$	$2,09 \times 10^{-01}$
Máximo	$1,70 \times 10^{-01}$	$2,29 \times 10^{-01}$	$1,76 \times 10^{-01}$	$2,77 \times 10^{-01}$
Mínimo	$7,92 \times 10^{-02}$	$1,81 \times 10^{-01}$	$1,06 \times 10^{-01}$	$1,32 \times 10^{-01}$
Desvio Padrão	$3,38 \times 10^{-02}$	$2,39 \times 10^{-02}$	$3,33 \times 10^{-02}$	$5,20 \times 10^{-02}$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para o Período Orbital de dois Corpos com base em 5 execuções

Fonte: O autor, 2023.

Dentre as funções obtidas, a com menor erro ficou longe da função original, embora com erro bem próximo de nulo, demonstrou alta complexidade no resultado gerado, e nas operações também. Mesmo sendo uma função de apenas uma variável, o intervalo tanto de constantes como do próprio intervalo utilizado pode ter tido um impactado negativo no desempenho do mesmo.

Dentre as funções obtidas, aquela que exibiu o menor erro revelou-se bastante

distante da forma original da função, embora tenha alcançado uma precisão quase nula. Esse resultado denota uma alta complexidade tanto na totalidade quanto nas operações envolvidas. Apesar de ser uma função com apenas uma variável, a escolha do intervalo para  $d$ , como as constantes, pode ter impactado negativamente no desempenho do modelo.

$$\frac{d(4d - \sqrt{d}) - 4d}{10^{10}} + 9.89997 \quad (11)$$

Analisando os demais resultados, nenhum deles conseguiu se aproximar esteticamente da função original. Os resultados para essa função em particular revelaram-se extremamente complexos, destacando-se pela sua intrincada estrutura.

Ao examinarmos os resultados sob a ótica das demais funções, fica evidente que a função em questão apresentou uma proximidade notável com o erro nulo, indicando um progresso encorajador. Possíveis melhorias podem ser exploradas por meio da recalibração dos parâmetros do algoritmo ou, ainda, considerando o tratamento mais refinado do intervalo de estudo. Essas abordagens têm o potencial de aumentar a precisão do modelo, permitindo que o algoritmo se aproxime ainda mais do resultado desejado.

Destaca-se que, em relação às operações unárias, a função em análise faz uso desse tipo de operação duas vezes, uma abordagem diferenciada em comparação com a maioria dos resultados, que empregam três ou mais operações unárias e, conseqüentemente, apresentam erros mais elevados. O efeito marcante das operações unárias, combinado com intervalos numéricos demasiadamente extensos, torna-se notório no desenvolvimento dos resultados. Essa observação sugere que uma análise mais refinada desses fatores pode ser crucial para otimizar o desempenho do algoritmo e alcançar resultados ainda mais precisos.

## 6.7 Campo Elétrico Puntiforme

A fundamentação teórica para esta função remonta às contribuições de grandes físicos, e é tratada no livro de Richard Feynman, *The Feynman Lectures on Physics* [19] o livro serve como um guia venerável para estudantes e entusiastas da física. Em sua obra, Feynman aborda com maestria os conceitos e princípios fundamentais da física, fornecendo percepções valiosas sobre a natureza do universo. A função em questão pode ser observada abaixo:

$$E = \frac{q}{4\pi\epsilon_0 r^2} \quad (12)$$

A função do campo elétrico de uma carga puntiforme expressa a intensidade do campo elétrico ( $E$ ) gerado por uma carga ( $q$ ) em um ponto específico do espaço, a uma distância ( $r$ ) da carga. A dependência inversa do quadrado da distância ( $r^2$ ) destaca a característica de propagação do campo elétrico, demonstrando que sua intensidade diminui rapidamente à medida que nos afastamos da carga puntiforme. Esta relação, derivada das leis de Coulomb, evidencia a natureza radial e esférica do campo elétrico gerado por uma carga puntiforme.

A função do Campo Elétrico gerado por uma carga puntiforme envolve duas variáveis,  $q$  e  $r$ , e três constantes,  $\varepsilon_0$  (permissividade elétrica do vácuo) e  $\pi$ . O intervalo escolhido para a variável  $q$  abrange os números inteiros de  $10^{-19}$  a  $10^{19}$ , representando cargas típicas em experimentos de partículas subatômicas. Já para a variável  $r$ , o intervalo considerado é de 1 a 100, refletindo distâncias relevantes em estudos envolvendo interações elétricas em escalas atômicas.

As constantes  $\varepsilon_0$  e  $\pi$  possuem valores específicos:  $\varepsilon_0$  é a permissividade elétrica do vácuo, com  $8.854 \times 10^{-12}$ , e  $\pi$  é a constante matemática aproximadamente igual a 3.14159. Além disso, temos também a constante 4, sendo o valor original da função.

Essa escolha de intervalo é estrategicamente feita para abranger eventos específicos dentro do estudo de partículas subatômicas, onde variações significativas de carga e distância são observadas. Esses valores representativos permitem uma análise mais aprofundada das interações elétricas em escalas extremamente pequenas, contribuindo para a compreensão de fenômenos ao nível subatômico.

Ao conduzir uma análise minuciosa da função do Campo Elétrico, os resultados apresentados na Tabela 17 indicam que esta execução se destaca como a mais exigente em termos de recursos computacionais. A média de tempo de processamento cresce consideravelmente à medida que o tamanho da amostra aumenta, atingindo um tempo de execução que ultrapassa às 4 horas quando a amostra atinge 80 elementos. Essa característica é diretamente influenciada pelos valores elevados das constantes presentes na função, tornando os testes computacionais mais penosos, isso se deve à sua natureza intrincada e complexa da função.

É relevante destacar que a complexidade inerente à função do Campo Elétrico exerce uma influência significativa sobre o comportamento dos índices de erro. Embora os valores de erro apresentados na Tabela 18 sejam menores em comparação com a função da Força Gravitacional, eles ainda permanecem substanciais quando considerados em relação aos demais resultados obtidos. Essa observação sugere que a natureza intrincada da função, aliada aos valores das constantes envolvidas, contribui para um grau considerável de incerteza nos resultados. Um ponto notável é o crescimento da média dos erros, embora a amostra 20 tenha exibido um comportamento diferenciado e promissor. Essa peculiaridade nos leva a conjecturar a possibilidade da existência de um tamanho de amostra ideal para conduzir esses testes, uma vez que, dentre todos os resultados, esta amostra

Tabela 17 - Tempo médio (s) entre 5 execuções para o Campo Elétrico Puntiforme

Função	Amostras (n)			
	10	20	40	80
Média	2859,11	5721,72	8903,17	17055,02
Máximo	3263,73	7894,88	10719,40	19622,50
Mínimo	2398,61	3952,16	7060,80	13589,00
Desvio Padrão	415,71	1796,00	1490,39	2772,40

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para o Campo Elétrico Puntiforme com base em 5 execuções

Fonte: O autor, 2023.

apresentou uma ordem de grandeza menor nos erros.

A análise da função do Campo Elétrico enfatiza a necessidade de considerar cuidadosamente a complexidade das funções e o impacto das constantes envolvidas ao aplicar o algoritmo. Embora essa execução tenha demandado mais recursos e tenha apresentado um nível relativamente mais alto de erro, ela ainda contribui para o entendimento da eficácia e limitações do algoritmo em cenários mais desafiadores.

Tabela 18 - RMSE entre 5 execuções para o Campo Elétrico Puntiforme

Função	Amostras (n)			
	10	20	40	80
Média	$3,36 \times 10^{20}$	$2,11 \times 10^{19}$	$4,10 \times 10^{20}$	$5,53 \times 10^{20}$
Máximo	$1,60 \times 10^{20}$	$4,69 \times 10^{18}$	$1,30 \times 10^{19}$	$3,60 \times 10^{20}$
Mínimo	$5,02 \times 10^{20}$	$2,63 \times 10^{19}$	$4,25 \times 10^{20}$	$1,13 \times 10^{21}$
Desvio Padrão	$1,16 \times 10^{20}$	$1,60 \times 10^{19}$	$3,97 \times 10^{20}$	$1,83 \times 10^{20}$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para o Campo Elétrico Puntiforme com base em 5 execuções

Fonte: O autor, 2023.

Ao analisarmos atentamente os resultados produzidos pelo algoritmo, é possível identificar a função a seguir como aquela que obteve o melhor desempenho. Em contraste com a função original do Campo Elétrico, os resultados gerados demonstram uma divergência substancial. Esse afastamento indica a dificuldade do algoritmo em reproduzir com precisão a complexidade inerente à função original, o que sugere limitações em sua capacidade de modelagem em casos mais intrincados.

$$\frac{(2q + 1 \times 10^{12})}{\left(\frac{r}{8.854^2}\right)^2} + \frac{\left(\frac{q}{(2r)^2}\right) \left(\frac{1 \times 10^{12}}{\pi}\right)}{8.854} - q \quad (13)$$

Comparando os resultados obtidos com os demais, é evidente que o intervalo dos dados desempenhou um papel significativo nas discrepâncias observadas. Nota-se que esta função apresenta apenas uma operação unária, o que, em comparação com a função do Período Orbital de dois corpos (que tinha duas operações unárias), resultou em um impacto menos acentuado dessa operação. Contudo, é importante destacar que o intervalo considerável utilizado em  $q$  nesta função pode ter tido um impacto negativo substancial nos resultados.

Em testes futuros, para alcançar resultados mais promissores, pode ser benéfico explorar fenômenos mais específicos que imponham restrições mais rigorosas ao intervalo dos números utilizados no estudo. Isso pode contribuir para reduzir possíveis efeitos adversos de intervalos extensos e melhorar a precisão geral do algoritmo, além de uma calibração dos parâmetros.

## 6.8 Função de Densidade de Probabilidade

No contexto da física estatística, a distribuição normal surge como uma descrição estatística de eventos aleatórios, é utilizada na teoria quântica e em muitos outros ramos da física moderna. Essa função também se destaca no estudo de flutuações quânticas, onde as incertezas inerentes ao comportamento quântico de partículas subatômicas são modeladas por distribuições normais. Richard Feynman, um físico teórico, aborda conceitos fundamentais relacionados à probabilidade quântica em seu livro *The Feynman Lectures on Physics* [19]. A função da densidade de probabilidade:

$$f = e^{-\frac{\theta^2}{2}} \cdot \frac{1}{\sqrt{2\pi}} \quad (14)$$

A função da densidade de probabilidade desempenha um papel vital na modelagem matemática de fenômenos probabilísticos na física e em várias disciplinas científicas, fornecendo uma estrutura poderosa para entender e prever o comportamento de sistemas físicos sujeitos a flutuações e incertezas.

A função de densidade de probabilidade apresenta uma estrutura simples, composta por uma variável e três constantes fundamentais. A única variável, representada por  $\theta$ , foi delimitada no intervalo dos números racionais de  $-1$  até  $1$ . Esta escolha específica de intervalo é justificada pela correspondência com a distribuição padrão, onde esse intervalo abrange aproximadamente um desvio padrão da média.

As três constantes essenciais na função são  $\pi$ ,  $e$  e o valor  $2$  da função original, representadas pelos valores numéricos  $3.14159$  e  $2.71828$ , respectivamente. Essas constantes desempenham papéis cruciais na modelagem matemática da distribuição de probabilidade.

A seleção do intervalo, além de ser coerente com a distribuição padrão, também é orientada pela simplicidade matemática. Trabalhar com um intervalo restrito simplifica cálculos e facilita interpretações do algoritmo, contribuindo para resultados mais claros e compreensíveis. A padronização do intervalo escolhido confere uniformidade aos resultados, tornando-os mais facilmente comparáveis e interpretáveis em diversos contextos.

Ao examinar minuciosamente a função da densidade de probabilidade da distribuição normal, a Tabela 19 revela características notáveis. Em relação ao tempo de execução, observa-se uma relação direta com o aumento do tamanho da amostra. Esse comportamento alinha-se à expectativa de que o processamento computacional se intensifica à medida que mais dados são considerados, refletindo em um tempo consideravelmente maior para amostras de maior magnitude. Essa tendência é acompanhada pelo aumento do Desvio Padrão, que, por sua vez, é uma consequência direta do incremento no intervalo entre os valores mínimo e máximo. Esse fenômeno sugere uma expansão na distribuição dos pontos, contribuindo para o aumento do intervalo observado. Dessa forma, podemos inferir uma sensibilidade do algoritmo em relação ao tamanho da amostra quando aplicado a essa função específica.

Tabela 19 - Tempo médio (s) entre 5 execuções para a Função de Densidade de Probabilidade

Função	Amostras (n)			
	10	20	40	80
Média	1180,84	2291,01	3277,96	7599,09
Máximo	1423,35	2817,47	4222,25	12332,10
Mínimo	1038,12	1924,55	2190,37	5291,39
Desvio Padrão	170,28	359,92	767,27	2849,91

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a Função de Densidade de Probabilidade com base em 5 execuções

Fonte: O autor, 2023.

No que diz respeito ao erro, destacamos que, embora não tenha alcançado exatamente o valor zero, ele se aproximou consideravelmente desse limiar, como podemos notar na Tabela 20. Esse é um indicativo de que o algoritmo foi capaz de produzir resultados altamente precisos para a função da densidade de probabilidade da distribuição normal. É interessante notar que, apesar dos resultados iniciais terem sido elevados durante a execução do código, o algoritmo demonstrou uma notável capacidade de convergir para valores de erro próximos de zero à medida que o processo avançava. Esse comportamento sugere que o algoritmo é capaz de lidar de forma eficaz com as complexidades matemáticas envolvidas nessa função, mesmo que inicialmente apresente desafios. Notamos que há a ocorrência de um aumento na média dos erros conforme a amostra cresce, comportamento também reflete no Desvio Padrão.

Em relação às funções obtidas pelo algoritmo, é importante ressaltar que, embora

Tabela 20 - RMSE entre 5 execuções para a Função de Densidade de Probabilidade

Função	Amostras (n)			
	10	20	40	80
Média	$9,61 \times 10^{-4}$	$1,43 \times 10^{-3}$	$2,62 \times 10^{-3}$	$2,55 \times 10^{-3}$
Máximo	$1,32 \times 10^{-3}$	$1,99 \times 10^{-3}$	$4,03 \times 10^{-3}$	$4,04 \times 10^{-3}$
Mínimo	$1,32 \times 10^{-3}$	$1,99 \times 10^{-3}$	$4,03 \times 10^{-3}$	$4,04 \times 10^{-3}$
Desvio Padrão	$5,25 \times 10^{-4}$	$7,84 \times 10^{-4}$	$1,04 \times 10^{-3}$	$1,94 \times 10^{-3}$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a Função de Densidade de Probabilidade com base em 5 execuções

Fonte: O autor, 2023.

o erro tenha sido extraordinariamente baixo, era esperado que o algoritmo não atingisse uma resposta com erro nulo. Isso se deve ao fato de que a operação exponencial, um atributo necessário para que o algoritmo alcance a função original, ainda não havia sido incorporada ao algoritmo. No entanto, o propósito desse teste era observar como o algoritmo se comportaria diante desse tipo de omissão. Posteriormente, com a inclusão de novos parâmetros unários, poderemos realizar comparações mais detalhadas.

A análise dos resultados revelou que a função obtida pelo algoritmo, embora tenha apresentado um erro praticamente nulo, é substancialmente mais complexa do que a função original. Esse resultado era esperado, uma vez que a ausência da operação de exponenciação levou o algoritmo a empregar um grande número de operações quadráticas e raízes. A complexidade adicional indica a dificuldade do algoritmo em encontrar uma representação simplificada da função original sem o uso de operações exponenciais.

$$\frac{\theta\left(\frac{2\theta^3}{\pi^2} - \theta\right) + 2}{4 + \sqrt{\pi}} \quad (15)$$

Ao analisarmos os resultados obtidos nesta função, destacam-se uma notável desenvoltura e resultados promissores. Como mencionado anteriormente, a ausência de funções exponenciais teve um impacto positivo no desenvolvimento do algoritmo. Os resultados alcançados são bastante encorajadores, evidenciando a eficácia da escolha cuidadosa de intervalos e parâmetros. Em diversos casos, o erro quase atingiu níveis nulos, indicando uma precisão notável.

Vale ressaltar que, embora as operações unárias estejam presentes, seu impacto foi significativamente reduzido em comparação com outras funções que possuíam uma complexidade maior no número de operações (tanto binárias como unárias). A simplicidade desta função, aliada à escolha adequada de parâmetros e intervalos, parece ter contribuído de maneira positiva para o desempenho geral do algoritmo. Este resultado sugere que, em contextos futuros.

## 6.9 Ondas Eletromagnéticas

A função das Ondas Eletromagnéticas emerge como uma expressão derivada das fundamentais funções de Maxwell, fornecendo uma descrição precisa do comportamento dos campos elétricos e magnéticos no espaço.

De aplicação abrangente, essa função desempenha um papel central na teoria das ondas eletromagnéticas. Além disso, sua utilidade se estende aos domínios da óptica e das comunicações, notadamente na análise de antenas e na compreensão da propagação de ondas eletromagnéticas.

Richard Feynman, um físico teórico, contribuiu significativamente para a compreensão desses fenômenos em sua obra monumental *The Feynman Lectures on Physics* [20]. Suas explorações sobre eletrodinâmica quântica e eletromagnetismo oferecem percepções valiosas para aqueles que buscam desvendar as complexidades das ondas eletromagnéticas. A função de dispersão, representada por

$$k = \sqrt{\frac{\omega^2}{c^2} - \frac{\pi^2}{d^2}} \quad (16)$$

Aqui,  $k$  denota o número de onda,  $\omega$  é a frequência angular,  $c$  é a velocidade da luz no vácuo e  $d$  é a distância entre elementos em um arranjo periódico. A função de dispersão para ondas eletromagnéticas é uma janela para estudar a velocidade da luz e a estrutura espacial do meio de propagação.

A função em questão incorpora duas variáveis e duas constantes, desempenhando um papel significativo em fenômenos acústicos em larga escala. No que diz respeito às variáveis, consideramos  $d$  e  $\omega$ , o primeiro variando no intervalo dos números naturais entre 1000 e 10000, enquanto o segundo abrange os naturais de  $10^{15}$  a  $10^{21}$ . No âmbito das constantes, destacam-se  $\pi$ , cujo valor é 3.14159, e  $c$ , com uma magnitude de  $3 \times 10^8$ .

A escolha específica desses intervalos é orientada pelo cenário em que a função é aplicada, centrando-se em fenômenos acústicos de grande escala. Essa seleção estratégica reflete a amplitude de aplicações da função em contextos onde esses parâmetros assumem valores dentro dessas faixas específicas.

Além disso, a delimitação desses intervalos não é arbitrária; ao contrário, ela visa otimizar os cálculos. Ao restringir as variáveis a esses intervalos particulares, não apenas mantemos a relevância para as aplicações práticas, mas também facilitamos os cálculos, o que se traduz em eficiência computacional e resultados mais ágeis. Assim podemos ter uma abordagem computacionalmente eficaz para a resolução de problemas.

Ao examinarmos a Tabela 21, notamos uma tendência de crescimento na média dos tempos à medida que o tamanho da amostra aumenta, um comportamento que se reflete também no Desvio Padrão. Destaca-se que esse padrão é acompanhado por um

aumento na amplitude do intervalo entre os valores mínimos e máximos, o que explica o crescimento no Desvio Padrão.

Tabela 21 - Tempo médio (s) entre 5 execuções para a Função das Ondas Eletromagnéticas

Função	Amostras (n)			
	10	20	40	80
Média	1918,47	2487,52	4984,64	7793,84
Máximo	2224,96	3419,41	8492,36	12744,00
Mínimo	1376,78	2102,41	2993,52	5791,12
Desvio Padrão	343,32	549,69	2229,17	2861,86

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a função das Ondas Eletromagnéticas com base em 5 execuções

Fonte: O autor, 2023.

Ao analisar a Tabela 22, é evidente um resultado extraordinariamente elevado na amostra de tamanho 10, o que impactou significativamente a média, levando-a a um valor extremamente alto. Contudo, à medida que o tamanho da amostra aumenta, observa-se uma estabilização nos resultados médios, com pouca discrepância entre eles. Em relação ao Desvio Padrão, nota-se novamente a influência do resultado anômalo na amostra de tamanho 10, refletindo uma maior variabilidade. Para as demais amostras, destaca-se uma redução no Desvio Padrão, especialmente na amostra de tamanho 40, explicada pelo menor intervalo entre os valores mínimo e máximo obtidos nos testes.

Tabela 22 - RMSE entre 5 execuções para a função das Ondas Eletromagnéticas

Função	Amostras (n)			
	10	20	40	80
Média	$1,26 \times 10^{27}$	$1,95 \times 10^9$	$1,60 \times 10^9$	$1,40 \times 10^9$
Máximo	$6,31 \times 10^{27}$	$2,06 \times 10^9$	$1,62 \times 10^9$	$1,51 \times 10^9$
Mínimo	$9,99 \times 10^8$	$1,72 \times 10^9$	$1,59 \times 10^9$	$1,26 \times 10^9$
Desvio Padrão	$2,82 \times 10^{27}$	$1,42 \times 10^8$	$1,54 \times 10^7$	$1,22 \times 10^8$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a função das Ondas Eletromagnéticas com base em 5 execuções

Fonte: O autor, 2023.

Dentre os resultados obtidos pelo algoritmo, o que mais chegou próximo de resultado foi nulo foi:

$$\frac{w}{-(10000\sqrt{3} + \pi)^2 d - d - \pi + 300000000} \quad (17)$$

A expressão obtida apresenta uma combinação complexa de termos. Essa complexidade reflete a dificuldade do algoritmo em reproduzir fielmente a função original, podemos observar que a função encontrada consegue utilizar todas as variáveis fornecidas ao problema.

Ao comparar os resultados obtidos com outros, torna-se evidente o impacto reduzido das operações unárias nesta função específica. Com cinco operações unárias, essa expressão se destaca em complexidade em comparação com outras analisadas. No entanto, a escolha de um intervalo de dados mais restrito pode ter beneficiado o desempenho do algoritmo em relação a outras funções, como a do Campo Elétrico Puntiforme. Esta última, apesar de ter menos operações unárias, enfrentou desafios decorrentes do amplo intervalo de dados.

## 6.10 Distribuição de Bose-Einstein

Esta formulação é fundamental para compreender o comportamento estatístico dessas partículas, em que  $n$  representa o número médio de partículas em um estado de energia específico,  $\omega$  é a frequência angular desse estado,  $k_B$  é a constante de Boltzmann,  $T$  é a temperatura absoluta do sistema, e  $k$  é a constante de Planck. A função da Distribuição de Bose-Einstein é expressa como:

$$n = \frac{1}{e^{\left(\frac{k\omega}{k_B T}\right)} - 1} \quad (18)$$

Richard Feynman serve como uma referência valiosa em seu livro *The Feynman Lectures on Physics* [21], aborda a Distribuição de Bose-Einstein e suas aplicações em diversos fenômenos. A abordagem matemática desta função e suas implicações físicas são cruciais para a descrever sistema quântico constituído de spin inteiros e calcular os valores médios de suas grandezas.

Sobre a função em questão, destacam-se duas variáveis e quatro constantes fundamentais. As variáveis em análise são  $\omega$  e  $T$ , representando, respectivamente, a frequência angular e a temperatura. O  $\omega$  varia no intervalo dos números racionais entre  $4 \times 10^{14}$  e  $7,5 \times 10^{14}$ , enquanto  $T$  abrange valores naturais de 1000 a 100000. Essa delimitação de intervalos é estratégica, pois direciona a função para eventos específicos, centrando-se em temperaturas elevadas associadas à luz visível. Além disso, essa escolha otimiza os cálculos computacionais, promovendo eficiência e agilidade na execução do programa.

No que diz respeito às constantes, temos  $k_B$ , a constante de Boltzmann, com valor aproximado de  $1.380649 \times 10^{-23}$ ,  $h$ , a constante de Planck, aproximadamente  $6.62607015 \times 10^{-34}$ , e o  $e$  com valor de 2,718281828. Também temos a constante da função original

com valor igual a 1.

Ao examinarmos os resultados apresentados na Tabela 23, percebemos um padrão comum no comportamento desses dados. Evidencia-se um aumento nas médias dos tempos à medida que o tamanho da amostra aumenta, uma tendência que também se manifesta no Desvio Padrão. Esse fenômeno pode ser explicado pelo crescimento do intervalo entre os valores mínimos e máximos à medida que a amostra se torna mais extensa, o que, por sua vez, contribui para o aumento do Desvio Padrão.

Essa análise reflete a sensibilidade do algoritmo ao tamanho da amostra, indicando que, à medida que mais dados são considerados, a variação nos tempos de execução também aumenta. Esse comportamento pode ser atribuído à complexidade inerente à Distribuição de Bose-Einstein e às interações entre as variáveis e constantes envolvidas.

Tabela 23 - Tempo médio (s) entre 5 execuções para a Função de Distribuição de Bose-Einstein

Função	Amostras (n)			
	10	20	40	80
Média	2275,64	2780,55	4963,58	8583,63
Máximo	3414,18	3255,70	6131,64	10817,90
Mínimo	1556,77	2428,82	3013,28	5903,09
Desvio Padrão	763,31	318,98	1431,00	1790,42

Legenda: Resultados do Tempo médio (s), Máximo, Mínimo e Desvio Padrão para a função de Distribuição de Bose-Einstein com base em 5 execuções

Fonte: O autor, 2023.

Ao analisarmos a Tabela 24, percebemos uma peculiaridade no comportamento da média dos erros em relação ao tamanho da amostra. Diferentemente do esperado, não há um crescimento proporcional à medida que a amostra aumenta. Um ponto notável é o resultado incomum obtido para a amostra de tamanho 40, que apresentou o maior valor para o erro médio, influenciando diretamente no Desvio Padrão, que também se tornou o mais elevado.

Os resultados das amostras de tamanho 10 e 20 mostram uma diferença na média dos erros, embora os Desvios Padrão dessas amostras sejam próximos. Isso pode ser explicado pela divergência nos intervalos entre os valores mínimos e máximos, que, embora distintos, apresentam uma variação semelhante.

Surpreendentemente, a amostra de tamanho 80 revelou-se como a de melhor desempenho neste teste específico. Esse resultado contraintuitivo sugere que, para essa distribuição específica, um tamanho amostral maior pode ser benéfico para o algoritmo, proporcionando resultados mais precisos e consistentes.

Dentre as funções obtidas pelo algoritmo a que mais se aproximou da função original foi:

Tabela 24 - RMSE entre 5 execuções para a função de Distribuição de Bose-Einstein

Função	Amostras (n)			
	10	20	40	80
Média	$6,21 \times 10^{-2}$	$3,80 \times 10^{-2}$	$1,61 \times 10^{-1}$	$2,48 \times 10^{-2}$
Máximo	$8,65 \times 10^{-2}$	$5,95 \times 10^{-2}$	$5,77 \times 10^{-1}$	$2,97 \times 10^{-2}$
Mínimo	$5,02 \times 10^{-2}$	$2,51 \times 10^{-2}$	$4,00 \times 10^{-2}$	$1,96 \times 10^{-2}$
Desvio Padrão	$1,50 \times 10^{-2}$	$1,38 \times 10^{-2}$	$2,32 \times 10^{-1}$	$3,61 \times 10^{-3}$

Legenda: Resultados do RMSE Médio, Máximo, Mínimo e Desvio Padrão para a função de Distribuição de Bose-Einstein com base em 5 execuções

Fonte: O autor, 2023.

$$\frac{(\frac{10}{w} + e)(\frac{w}{T^2+T} + 2T^2 + 2T)}{T} - \frac{\sqrt{T+e}}{e} - 0,00209869 \quad (19)$$

Observa-se que, embora a função tenha apresentado um erro praticamente nulo, seu resultado distanciou-se consideravelmente da função original. Esse resultado era esperado, uma vez que o algoritmo não tinha a capacidade de lidar com funções exponenciais. O propósito desse teste era avaliar como o algoritmo responderia diante dessa limitação.

Ao compararmos os resultados obtidos com outras funções, podemos extrair conclusões semelhantes às observadas na Função de Densidade de Probabilidade, uma vez que ambas têm a mesma natureza exponencial. Embora cada função possua suas peculiaridades individuais, é evidente que esta função não obteve resultados tão expressivos quanto a Função de Densidade de Probabilidade. No entanto, é notório que o intervalo de dados teve um impacto considerável. Isso se deve ao fato de as constantes presentes na função serem valores muito pequenos, aumentando a probabilidade de ocorrência de erros de underfitting. A escolha adequada dos parâmetros, assim como ajustes no intervalo dos dados, além da inserção de novos parâmetros (unários), pode ser crucial para melhorar a capacidade do algoritmo em lidar com essa função específica.

## 6.11 Discussões sobre o resultado

Uma análise abrangente das funções utilizadas é apresentada na Tabela 25, destacando a quantidade de operações de cada função. Observa-se que as operações unárias exercem um impacto significativo no algoritmo, acentuando possíveis erros numéricos.

Essas constatações ressaltam a importância de considerar o intervalo utilizado na execução do algoritmo, pois operações como raiz ou potenciação tendem a acentuar erros. Isso destaca a relevância de uma cuidadosa escolha de parâmetros, especialmente no que

Tabela 25 - Observações sobre as funções

Fenômeno	Binárias	Unárias
Distância Euclidiana	3	3
Massa Relativística	3	3
Lentes Delgadas	4	0
Produto Interno	5	0
Força Gravitacional	8	3
Período Orbital de dois Corpos	6	2
Campo Elétrico Puntiforme	4	1
Função de Densidade de Probabilidade	4	3
Ondas Eletromagnéticas	3	5
Distribuição de Bose-Einstein	5	1

Legenda: Análise das operações (Binárias e Unárias) - Resumo dos Resultados

Fonte: O autor, 2023.

diz respeito ao intervalo de valores das variáveis envolvidas nas funções. Um intervalo inadequado pode impactar significativamente a precisão do algoritmo, tornando crucial uma análise criteriosa e ajustes refinados nos parâmetros utilizados. Essa abordagem é fundamental para minimizar possíveis erros decorrentes do intervalo de dados.

Na Tabela 26, foram realizadas observações sobre os resultados, considerando, Intervalos Grandes como números maiores que  $10^6$ , pois o algoritmo tendia a truncar números que fossem maiores que esses ou pequenos demais  $10^{-6}$ . Para Todas as Variáveis, considerou-se a natureza do resultado em relação ao uso ou não de todas as variáveis no melhor resultado, e, por fim, a quantidade de acertos em 20 testes com amostras diferentes.

Tabela 26 - Observações sobre os resultados

Fenômeno	Intervalos Grandes	Todas as Variáveis	Acertos
Distância Euclidiana	Não	Não	0
Massa Relativística	Sim	Sim	0
Lentes Delgadas	Não	Sim	9
Produto Interno	Não	Sim	7
Força Gravitacional	Sim	Não	0
Período Orbital de dois Corpos	Sim	Sim	0
Campo Elétrico Puntiforme	Sim	Sim	0
Função de Densidade de Probabilidade	Não	Sim	0
Ondas Eletromagnéticas	Sim	Sim	0
Distribuição de Bose-Einstein	Sim	Sim	0

Legenda: Análise dos intervalos e do uso das variáveis - Resumo dos Resultados

Fonte: O autor, 2023.

Observa-se uma prevalência de acertos em funções que não fazem uso de intervalos

los grandes. Embora essas funções não tenham utilizado operações unárias, o que é um fator a ser considerado, é interessante notar que a função da Distância Euclidiana não fez uso de intervalos grandes, mas também não utilizou todas as variáveis, o que pode ter contribuído para não atingir a função original desejada. Outra função que não utilizou intervalos muito grandes foi a Função da Densidade de Probabilidade. No entanto, como destacado anteriormente, essa função possui uma função ainda não implementada (Função exponencial). Em comparação com a Distribuição de Bose-Einstein, que obteve resultados próximos de zero, observamos que, de fato, o intervalo utilizado tende a impactar o algoritmo, dificultando a obtenção de um resultado ótimo.

Ao analisar o uso ou não de todas as variáveis, nota-se que a maioria dos resultados conseguiu utilizar todas as variáveis, com algumas exceções, como a função da Força Gravitacional, que apresentou um erro extremamente alto, possivelmente impactado pelos intervalos grandes utilizados nos testes. Além disso, a função da Distância Euclidiana obteve um erro baixo, embora não tão baixo quanto a Função da Densidade de Probabilidade. Essas nuances ressaltam a complexidade na escolha de intervalos e na utilização de todas as variáveis para atingir resultados mais precisos em algoritmos de Regressão Simbólica.

## 6.12 Teste com as grandezas físicas

As grandezas físicas podem desempenhar um papel fundamental na formulação e compreensão de modelos de regressão simbólica, representando aspectos mensuráveis e quantificáveis do mundo natural. No contexto da modelagem simbólica, essas grandezas se tornam variáveis de entrada junto aos dados coletados, desempenhando um importante papel para a construção de equações simbólicas que modelam relações entre diferentes fenômenos físicos.

A precisão na definição e medição de grandezas físicas é essencial para o desenvolvimento de modelos de regressão simbólica robustos e representativos. Estas grandezas, muitas vezes expressas por variáveis simbólicas, permitem a análise matemática das relações entre diferentes entidades físicas, possibilitando a criação de equações simbólicas que resumem comportamentos de sistemas físicos.

Ao empregar grandezas físicas em modelos de regressão simbólica, é possível capturar nuances e padrões subjacentes aos dados observados, contribuindo para uma compreensão mais profunda dos fenômenos estudados.

Nesta parte do estudo, foi realizado alguns testes com as funções já estudadas nesse trabalho, dito isto, as unidades foram implementadas nos testes. Duas equações ficaram de fora dessa rodada de testes, devido à função exponencial não tratar bem as unidades, sendo assim seria necessária outra implementação para o código realizar esse

tipo de operação.

Nessa etapa, adotou-se uma estratégia na manipulação dos dados. Inicialmente, empregou-se um conjunto de dados composto por 40 amostras. Desse conjunto, selecionou-se uma subamostra de 20 elementos para a constituição de um segundo conjunto. Posteriormente, a partir desse segundo conjunto, optou-se pela seleção de uma subamostra de 10 elementos, resultando na formação de um terceiro conjunto. Essa abordagem foi escolhida para alinhar-se ao desenvolvimento do código, uma vez que propicia uma consistência metodológica nos testes subsequentes.

Ademais, empregaram-se valores distintos para os parâmetros específicos do problema, uma escolha embasada na propriedade das unidades de conduzirem a abordagens mais sustentáveis. Nesse contexto, não se fez necessário ampliar o *exploration* ou o *exploitation* no algoritmo. Com essa consideração, optou-se por fixar os valores de *noimprovementmax* e *restart* em 100, buscando equilibrar as facetas de *exploration* e *exploitation*. Os demais parâmetros, tanto globais quanto específicos, permaneceram inalterados. Adicionalmente, a escolha de configurar a pilha (*stacklen*) com um tamanho de 30 justifica-se pelo fato de uma das funções-alvo possuir esse tamanho específico na pilha.

Introduziu-se um novo parâmetro, *moreDiversity*, com o propósito de enriquecer a diversidade dos cromossomos. Os testes conduzidos revelaram resultados benéficos, especialmente para a função da Distância Euclidiana. O parâmetro *moreDiversity* opera ao proporcionar um incremento na diversidade em momentos específicos, quando fornecido ao algoritmo.

Agora, abordando a análise dimensional, ela é uma abordagem essencial para garantir a consistência e validade das equações físicas, considerando as unidades das grandezas envolvidas. Ao explorar como essa análise é aplicada em diferentes operações matemáticas. Podemos diferenciar algumas operações entre si.

- Nas operações de **soma** e **subtração**, é fundamental que as grandezas tenham as mesmas dimensões. Considere a adição de dois comprimentos:  $L_1$  e  $L_2$ . Aqui, ambas as grandezas devem ter a dimensão de comprimento (metros) para garantir consistência dimensional. Caso as grandezas sejam diferentes não é possível realizar esse tipo de operação
- Ao **multiplicar** ou **dividir** grandezas, suas dimensões são combinadas. Se multiplicarmos a velocidade (dimensão de metros/segundo) pelo tempo (dimensão de segundos), obtemos a distância (dimensão de metros).
- Quando elevamos uma grandeza a uma **potência**, multiplicamos suas dimensões pela própria dimensão na potência correspondente. Se elevamos o comprimento ao quadrado, obtemos a área.
- Ao aplicar uma **raiz quadrada** ou outra raiz, dividimos as dimensões pela raiz

correspondente. Se tivermos uma área e aplicarmos a raiz quadrada, obtemos o comprimento.

### 6.12.1 Distância Euclidiana

A primeira função submetida a teste foi a Distância Euclidiana. É relevante salientar que todas as quatro variáveis do problema, assim como a saída da função, são expressas em metros (m). Os resultados desse teste, apresentados na Tabela 27, revelaram-se positivos. O algoritmo demonstrou acurácia ao ajustar a função em todas as iterações com 10 e 20 amostras. Contudo, ao lidar com 40 amostras, verificou-se um erro médio de 2.0331.

Tabela 27 - RMSE após a implementação da análise da unidade entre 5 execuções para a Distância Euclidiana

Função	Amostras (n)		
	10	20	40
Média	0	0	2.0331
Máximo	0	0	2.0892
Mínimo	0	0	1.9580
Desvio Padrão	0	0	0.0516

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Distância Euclidiana utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 28, destaca-se uma melhoria significativa. O algoritmo não apenas alcançou uma resposta substancialmente mais rápida do que na execução desprovida do emprego de grandezas físicas, mas também convergiu para a função do livro. Observa-se ainda que o experimento com uma amostra de tamanho 20 demonstrou desempenho superior, registrando o menor tempo entre os resultados avaliados.

O primeiro resultado obtido foi a função 20:

$$\sqrt{(x_2 - x_3)^2 + (x_0 + (((x_0 + x_0) - x_0) - ((x_1 + x_0) + (x_1 - x_1)))) \cdot \sqrt{\sqrt{1}}^2} \quad (20)$$

Essa função, quando simplificada, resulta na função alvo,  $\sqrt{(x_0 - x_1)^2 + (x_2 - x_3)^2}$ , o resultado observado foi para uma amostra de 10 amostras. Para 20 amostras a função encontrada foi a função 21:

$$\sqrt{\sqrt{((x_3 - x_2)^2 + (((1 - (1 * 1)) - (1 * 1))/1) * (x_1 - x_0))^2}} \quad (21)$$

Tabela 28 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Distância Euclidiana

Função	Amostras (n)		
	10	20	40
Média	393.5220	103.1739	763.5622
Máximo	402.4930	115.3460	791.9550
Mínimo	387.5790	95.2495	719.6330
Desvio Padrão	5.8002	8.9723	28.8071

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Distância Euclidiana utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Que também resulta na função alvo dos testes. Agora para a 40 amostras, temos o seguinte resultado, visualizado na função 22:

$$\frac{\sqrt{x_1^2}}{\left(\frac{1}{(1+1)}\right)} + \frac{\sqrt{(((x_1 + x_1) - ((x_1 + x_1) - x_1)) - x_0)^2}}{1^2} \quad (22)$$

Aqui podemos observar que o código evitou usar todas as variáveis do problema, quando simplificada a função 23 é o resultado encontrado:

$$\sqrt{(x_1 - x_0)^2} + \frac{\sqrt{x_1^2}}{2} \quad (23)$$

### 6.12.2 Massa Relativística

A segunda função sujeita a testes foi a Massa Relativística. Diferentemente da função da Distância Euclidiana, suas unidades de medida são distintas. Para as variáveis de entrada, temos a massa (kg) e a velocidade (m/s), enquanto para a saída temos apenas a massa (kg). Além destas, a função envolve a constante da velocidade da luz (m/s). Os resultados deste teste, apresentados na Tabela 29, revelaram-se ligeiramente afastados da forma original da função, conforme encontrada na literatura. O algoritmo evidenciou resultados promissores para uma amostra de tamanho 10, no entanto, para amostras de tamanho 20, houve uma discrepância significativa em relação aos testes com menos ou mais amostras. Para uma amostra de tamanho 40, o erro foi identificado como sendo o dobro da execução com 10 amostras.

Nos resultados temporais apresentados na Tabela 30, destaca-se resultados satisfatórios visto que na primeira execução antes da implementação das grandezas, o algoritmo

Tabela 29 - RMSE após a implementação da análise da unidade entre 5 execuções para a Massa Relativística

Função	Amostras (n)		
	10	20	40
Média	3.525022	13.4297	6.579192
Máximo	3.89	13.94	6.7391
Mínimo	3.21	13.2245	6.46266
Desvio Padrão	0.2816825384	0.2942786944	0.1422967509

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Massa Relativística utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

necessitava de mais tempo de processamento. Além disso, a vantagem agora dessa implementação é uma função mais próxima da literatura.

Tabela 30 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Massa Relativística

Função	Amostras (n)		
	10	20	40
Média	1458.818	765.741	2392.238
Máximo	1516.56	861.952	2573.26
Mínimo	1404.17	718.945	2283.57
Desvio Padrão	43.79907956	58.32273499	122.832588

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Massa Relativística utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

O primeiro resultado obtido com 10 amostras foi a função 24:

$$\frac{\sqrt{m_0} \sqrt{(m_0 + (m_0 + m_0)) + \frac{m_0 + m_0}{\frac{3 \times 10^8}{v} - 1}} \times 1 \times 1 \times v \times 1}{v + v} \quad (24)$$

Embora a função respeite as grandezas físicas, ela ainda se distancia pouco do resultado desejado, com mais algumas calibrações dos parâmetros o algoritmo pode chegar ao resultado desejado. Outra função obtida com a 20 amostras foi a 25:

$$\frac{\frac{(v \times v)((v-v)+v)(v+v)}{((3 \times 10^8)^2)^2} + 1}{1} \times \frac{m_0 \times m_0}{\frac{m_0}{1}} \quad (25)$$

Para a amostra de tamanho 40 a função 26 foi o resultado de saída do algoritmo:

$$\frac{\left(\frac{v^2+v^2}{1}\right)}{\left(\frac{3 \times 10^8 + 3 \times 10^8}{\frac{v}{3 \times 10^8}} - v\right)^2} \times \frac{m_0}{\left(\frac{v}{3 \times 10^8}\right)^2} + m_0 \quad (26)$$

Os resultados se mostraram promissores em relação ao teste sem as unidades, podemos observar que as funções se tornaram menos complexas e sem tanta redundância do que na primeira execução.

### 6.12.3 Lentes Delgadas

A terceira função submetida a testes foi a função das Lentes Delgadas. Sua grandeza física assemelha-se à da Distância Euclidiana. Para as variáveis de entrada, temos metros (m) e uma das variáveis com valor adimensional (representado pelo '\*'), enquanto a saída também é expressa em metros (m). Os resultados deste teste, apresentados na Tabela 31, revelaram um desempenho notável do código. Diferentemente da primeira execução, onde algumas execuções atingiram o ótimo, nesta fase, todas as execuções alcançaram o ótimo.

Tabela 31 - RMSE após a implementação da análise da unidade entre 5 execuções para a Lentes Delgadas

Função	Amostras (n)		
	10	20	40
Média	0	0	0
Máximo	0	0	0
Mínimo	0	0	0
Desvio Padrão	0	0	0

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Lentes Delgadas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 32, observa-se uma melhoria de quase 50% em relação à primeira execução. A implementação das unidades revelou-se não apenas um meio para otimizar esta função, mas também para aprimorar os resultados já obtidos.

Nessa execução os resultados não se diferenciaram muito, a não ser pela simplificação da função, observável principalmente nos testes com menos amostras. O resultado obtido com 10 amostras foi a função 27:

Tabela 32 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Lentes Delgadas

Função	Amostras (n)		
	10	20	40
Média	48.37974	39.48094	97.3292
Máximo	48.5483	40.0688	117.526
Mínimo	48.1729	39.0875	86.6806
Desvio Padrão	0.1660665921	0.3650266881	11.90819156

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Lentes Delgadas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

$$n \times \frac{1}{\left(\frac{n}{d_2} + \frac{1}{d_1}\right) \sqrt{n^2}} \quad (27)$$

A função 28 foi obtida para uma amostra de tamanho 20:

$$\frac{d_1 \times d_2}{\frac{(1^2)^2}{d_1} \times (d_1 d_2 + d_1 (d_1 n))} \quad (28)$$

E por fim, a função 29 foi obtida para uma amostra de tamanho 40:

$$\frac{d_1}{\sqrt{(1^2) + \left(\sqrt{\frac{nd_1}{(1\sqrt{d_2})^2}}\right)^2}} \quad (29)$$

#### 6.12.4 Produto Interno

A quarta função submetida a testes foi a função do Produto Interno, cujas grandezas físicas também se assemelham à Distância Euclidiana. Para as variáveis de entrada, utilizamos a unidade metro (m), enquanto a saída é expressa em área, com a unidade de medida metro ao quadrado ( $m^2$ ). Os resultados deste teste, apresentados na Tabela 33, destacam um desempenho aquém do esperado para as duas primeiras amostras. Entretanto, a última amostra, contendo mais dados, convergiu para um resultado próximo ao ótimo. Vale notar que algumas funções respondem positivamente ao aumento dos pontos, uma estratégia que não resultou em bons desempenhos para a Distância Euclidiana.

Os resultados temporais apresentados na Tabela 34 mostram uma diminuição do

Tabela 33 - RMSE após a implementação da análise da unidade entre 5 execuções para a Produto Interno

Função	Amostras (n)		
	10	20	40
Média	19.05702	29.828	0.0000
Máximo	19.0798	30.1035	0.0000
Mínimo	19.019	29.281	0.0000
Desvio Padrão	0.02349195607	0.3186171292	0.0000

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Produto Interno utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

tempo à medida que as amostras aumentam. Cabe ressaltar que a amostra maior, para essa função, teve um resultado benéfico.

Tabela 34 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Produto Interno

Função	Amostras (n)		
	10	20	40
Média	791.6212	485.2592	259.5666
Máximo	852.5650	487.9700	261.0460
Mínimo	738.0310	482.3980	257.6860
Desvio Padrão	50.4213	2.4822	1.4409

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Produto Interno utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nessa execução, obteve-se resultados menos complexos do que na primeira. Além disso, as equações geradas respeitam a variável de saída. No entanto, nesta execução, o algoritmo não fez uso de todas as variáveis de entrada, excluindo as variáveis  $x_5$  e  $x_1$  do resultado. O resultado obtido com 10 amostras foi a função 30:

$$\left( \frac{x_3 x_4 \times x_2}{x_4^2 + (x_3 + x_0)^2} \right) \times (x_4 \times (1 + 1 \times 1) + (x_4 + x_2)) \quad (30)$$

A função obtida para uma amostra de tamanho 20, também apresentou o mesmo problema, desta vez, a variável  $x_0$  não apareceu na função 31 de saída:

$$\left( ((x_3 + x_4) \times \left( x_4 \times \sqrt{\frac{1 + 1}{x_1 + (x_1 + (x_1 + (x_5 - x_2)))^2}} \right) \right) \right) \times (1 \times 1) \times 1) \times x_1 \quad (31)$$

E por fim, a função 32 que gerou a saída desejada utilizando uma amostra de tamanho 40:

$$x_2x_3 + 1 \left( x_1x_0 + \left( x_5x_4 \sqrt{\frac{(1 \times 1)^2}{1}} \right) \sqrt{1^2} \right) \quad (32)$$

Embora, o resultado de saída pareça da função 33 distante do desejado, ao ser simplificada, encontramos a função alvo.

$$x_0x_1 + x_2x_3 + x_4x_5 \quad (33)$$

### 6.12.5 Força Gravitacional

A quinta função sujeita a testes foi a função da Força Gravitacional. Sua grandeza física assemelha-se à da Distância Euclidiana. Para as variáveis de entrada, temos a distância em metros (m), enquanto a saída ocorre na unidade Newton (N). Esta função possui três constantes, que têm grandeza, sendo a massa da Terra ( $m_1$ ) e a massa da Lua ( $m_2$ ), dadas em quilogramas (kg), e a Constante Gravitacional (G) dada em ( $N * m^2/kg^2$ ). Os resultados deste teste, apresentados na Tabela 35, destacam um resultado semelhante ao primeiro teste. No entanto, cabe ressaltar que a função encontrada aqui se adequou melhor à análise dimensional dos dados. Assim, embora o erro aqui tenha sido igualmente alto, a função encontrada tem uma correspondência mais aceitável com os dados de entrada e de saída.

Tabela 35 - RMSE após a implementação da análise da unidade entre 5 execuções para a Força Gravitacional

Função	Amostras (n)		
	10	20	40
Média	$7.54 \times 10^{26}$	$1.47 \times 10^{27}$	$1.52 \times 10^{27}$
Máximo	$7.73 \times 10^{26}$	$1.82 \times 10^{27}$	$1.83 \times 10^{27}$
Mínimo	$7.17 \times 10^{26}$	$1.00 \times 10^{27}$	$1.22 \times 10^{27}$
Desvio Padrão	$2.23014 \times 10^{25}$	$3.03397 \times 10^{26}$	$2.27763 \times 10^{26}$

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Força Gravitacional utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 36, observa-se um tempo de execução significativamente menor em comparação com a primeira execução. Na amostra de tamanho 20, o algoritmo ficou preso em um ótimo, o que resultou em uma execução

mais rápida do que nas outras instâncias, onde o algoritmo conseguia alcançar resultados um pouco mais distantes antes de convergir.

Tabela 36 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Força Gravitacional

Função	Amostras (n)		
	10	20	40
Média	1701.5680	485.2592	1312.4620
Máximo	1775.2800	487.9700	1689.3000
Mínimo	1617.0400	482.3980	1179.2800
Desvio Padrão	65.0831	2.4822	213.1343

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Força Gravitacional utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nessa execução, obteve-se uma função que, embora siga a análise dimensional, não atende completamente ao desejado. A abordagem utilizada para amenizar os erros numéricos, com o emprego da base  $10^{10}$ , não contribuiu para que o algoritmo gerasse uma saída satisfatória. Além disso, vale notar que a função 34 produzida não utilizou todas as variáveis de entrada, deixando de considerar as variáveis  $x_1$ ,  $x_2$  e  $x_4$ . O resultado obtido com 10 amostras é apresentado:

$$\left( \frac{\frac{1}{734} \left( \left( (x_5 \left( (x_0 \times x_0) x_5 \right) x_5 \right) \times \frac{x_3}{0.667408 + 0.667408} \right) \times 0.667408^2 \right) 0.667408^2}{10^{10}} \right)^2 \quad (34)$$

A função 35 obtida para uma amostra de tamanho 20, também apresentou o mesmo problema, no entanto, aqui as variáveis que ficaram de fora, são  $x_3$ ,  $x_4$  e  $x_5$  não apareceu na função de saída:

$$\left( \left( \frac{\sqrt{\sqrt{\sqrt{\left( (x_1 + (x_2 - x_2)) + x_0 \right)^2}}}}{734} \times (734 \times 734) \right) \times 734 \right) \times \left( \sqrt{0.667408} \times \left( \sqrt{\left( \sqrt{10^{10}} \right)^2} \right)^2 \right) \quad (35)$$

E por fim, a função 36 que gerou a saída desejada utilizando uma amostra de

tamanho 40:

$$\left( (734^2) \times \left( \sqrt{\left( \left( (0.667408 \times 59720)^2 \right)^2 \right) \times 0.667408^2} \right) \right) \times (0.667408 \times \left( 10^{10} \times \sqrt{\sqrt{\sqrt{\left( (x_0 + x_1)^2 \right)^2}}}} \right) \quad (36)$$

Podemos observar a complexidade das equações obtidas pelo algoritmo, a qual pode ser ocasionada tanto pela complexidade intrínseca da função quanto pelos valores numericamente grandes envolvidos. Para esta execução, seria pertinente considerar alguma abordagem adicional de pré-processamento, visando mitigar esse tipo de erro e buscar equações mais simplificadas e interpretáveis. Este é um aspecto a ser explorado na busca por aprimoramentos no desempenho do algoritmo.

### 6.12.6 Período Orbital

A sexta função submetida a testes foi a função do Período Orbital. Para suas variáveis de entrada, temos a distância dada em metros (m), enquanto a saída ocorre em segundos (s). Esta função possui três constantes, todas com dimensões físicas: a massa da Terra ( $m_1$ ) e a massa da Lua ( $m_2$ ), expressas em quilogramas (kg), e a Constante Gravitacional (G), medida em ( $m^3/(kg * s^2)$ ). Os resultados deste teste, apresentados na Tabela 37, destacam-se por apresentar resultados com pouquíssima diferença em relação à primeira execução. É relevante ressaltar que o resultado obtido nesta fase está mais alinhado com a análise dimensional das grandezas físicas, representando, assim, um resultado mais próximo do esperado. Este alinhamento com a análise dimensional é crucial para garantir a coerência física das equações geradas pelo algoritmo.

Tabela 37 - RMSE após a implementação da análise da unidade entre 5 execuções para a Período Orbital

Função	Amostras (n)		
	10	20	40
Média	1.659032	36.7624	1.5790
Máximo	1.89772	36.962	1.9782
Mínimo	1.23772	36.518	1.2098
Desvio Padrão	0.2589156765	0.188391348	0.2722

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Período Orbital utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 38, destaca-se um tempo significativamente menor em comparação com a primeira execução. Além disso, é relevante observar que, na primeira execução, o tempo para a primeira amostra já atingia cerca de 2000 segundos, enquanto nesta segunda fase, o algoritmo consegue atingir resultados mais aceitáveis em menos tempo.

A redução substancial no tempo de execução é um indicativo positivo do aprimoramento do desempenho do algoritmo na resolução da função do Período Orbital. Essa melhoria pode ser crucial em cenários onde a eficiência computacional é fundamental.

Tabela 38 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Período Orbital

Função	Amostras (n)		
	10	20	40
Média	1483.2860	1204.7100	1659.2480
Máximo	1526.4800	1231.7900	1676.0300
Mínimo	1446.7900	1152.9500	1626.6100
Desvio Padrão	33.1906	30.5385	19.2268

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Período Orbital utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nesta execução, obteve-se uma função que, embora siga a análise dimensional desejada, ainda difere do resultado desejado. A abordagem utilizada para mitigar os erros numéricos, mediante o uso da base  $10^{10}$ , não contribuiu para que o algoritmo produzisse uma saída numericamente satisfatória. No entanto, é notável que a grandeza dimensional foi respeitada. Apresenta-se o resultado obtido com 10 amostras, o resultado foi a função 37:

$$\frac{\left( \frac{\frac{d^2}{d}}{d} \sqrt{\frac{d}{0.667408 \sqrt{734^2 \left( \frac{1 \times 10^{10}}{d} \right)^2}}}}{\sqrt{2}} \sqrt{1 \times 10^{10}} \right)}{\sqrt{2}} \quad (37)$$

A função 38 obtida para uma amostra de tamanho 20, também apresentou o mesmo problema:

$$\frac{\left( \frac{\sqrt{\sqrt{\sqrt{d} \cdot d^2 \cdot d}}}{(d+d)} \right) \times 734}{\frac{0.667408}{59720 \times \frac{734}{\left( \frac{3.14159}{2} \right)}}} \quad (38)$$

E por fim, a função 39 que gerou a saída desejada utilizando uma amostra de tamanho 40:

$$\frac{\frac{d}{d}}{\sqrt{\frac{\frac{d}{d} \times 10^{10}}{\left(\left(\sqrt{d^2}\right)^2\right)^2} \times 734 \times \sqrt{2^2} \times (0.667408 \times d)}} \quad (39)$$

As equações geradas pelo algoritmo demonstram uma complexidade considerável, especialmente quando comparadas à primeira execução, onde as equações estavam mais distantes da análise dimensional desejada. Essa complexidade pode ser atribuída à natureza intrínseca da função ou aos valores numéricos significativamente grandes, como observado na função anterior (Força Gravitacional).

Para essa execução, seria interessante considerar outras abordagens de pré-processamento a fim de mitigar esse tipo de erro. Isso pode envolver a exploração de técnicas específicas para lidar com valores extremos ou ajustes nos parâmetros do algoritmo para melhor acomodar as características da função. A busca por estratégias mais refinadas pode contribuir para resultados mais próximos da análise dimensional desejada.

### 6.12.7 Campo Elétrico Puntiforme

A sétima função submetida a testes foi a função do Campo Elétrico Puntiforme. As variáveis de entrada para esta função são o raio dado em metros (m) e a carga fornecida em Coulombs (C), enquanto a saída é expressa na unidade Newton por Coulomb (N/C). Esta função envolve apenas uma constante, o coeficiente de permissividade no vácuo, expresso em  $(C^2/N \cdot m^2)$ .

Os resultados deste teste, apresentados na Tabela 39, revelam um desempenho melhor, embora ainda um pouco distante do ótimo desejado. Cabe ressaltar que, na execução da amostra de tamanho 40, ocorreu um erro desconhecido no código, impossibilitando sua execução para essa amostra. No entanto, para as amostras de tamanho 10 e 20, o código conseguiu desempenhar bem, alcançando resultados superiores aos da primeira execução. Nessa execução, o erro para ambas as amostras não se distanciou muito um do outro.

Nos resultados temporais apresentados na Tabela 40, observa-se um tempo significativamente menor em comparação com a primeira execução, na qual tangenciava os 3000 segundos. Nesta execução, a média é inferior a 2000 segundos, além de apresentar um erro menor em relação à primeira execução.

Essa redução no tempo de execução é um indicativo positivo do aprimoramento do desempenho do algoritmo na resolução da função do Campo Elétrico Puntiforme. Essa melhoria pode ser crucial, especialmente em cenários onde a eficiência computacional é

Tabela 39 - RMSE após a implementação da análise da unidade entre 5 execuções para a Campo Elétrico Puntiforme

Função	Amostras (n)	
	10	20
Média	9998.82	9998.6
Máximo	10001.823	10001.52
Mínimo	9990.673	9990.28
Desvio Padrão	4.601720439	4.682925581

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Campo Elétrico Puntiforme utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

essencial.

Tabela 40 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Campo Elétrico Puntiforme

Função	Amostras (n)	
	10	20
Média	1993.3940	1352.4280
Máximo	2145.8400	1386.0000
Mínimo	1814.1900	1307.4300
Desvio Padrão	140.4989	29.1575

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Campo Elétrico Puntiforme utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Esta execução segue a análise dimensional desejada, no entanto, escapa de uma função simplificada, assim como nas duas execuções anteriores. O problema persistente ocorre novamente, onde a abordagem de usar a base  $10^{10}$  para mitigar os erros numéricos, não contribuiu para uma saída satisfatória do algoritmo. O resultado obtido com 10 amostras, foi a função 40:

$$\frac{\left(\frac{q}{r}\right)^2}{(q \times 0.08854)} \left( \frac{3.14159}{4\sqrt{\sqrt{(3.14159^2)^2}} \cdot \frac{q}{3.14159}} \right) \times 1 \times 10^{10} \quad (40)$$

A função 41 obtida para uma amostra de tamanho 20, tem o mesmo problema que na função com uma amostra de tamanho 10:

$$\frac{\frac{q}{r} \times 10^{10}}{\left(0.08854 \times \left(4 \times \frac{0.08854 \times \left(\frac{3.14159 \times \sqrt{r}}{0.08854}\right)}{\sqrt{r}}\right)\right)} \quad (41)$$

### 6.12.8 Ondas Eletromagnéticas

A nona função submetida a testes foi a função das ondas eletromagnéticas. As variáveis de entrada para esta função são dadas em radianos por segundo (rad/s) e metros (m), enquanto a saída é expressa em radianos por metro (rad/m). Esta função possui duas constantes significativas: o número  $\pi$  (rad) e a velocidade da luz ( $m/s$ ).

Os resultados deste teste, apresentados na Tabela 41, indicam um desempenho melhor na primeira execução. Entretanto, mesmo que ainda não tenha alcançado o ótimo, a segunda execução resultou em erros da ordem de grandeza de  $10^9$ , o que é considerado plausível.

Vale ressaltar que o tamanho da amostra não demonstrou um impacto direto nos resultados do erro. A diferença entre as execuções foi praticamente inexistente, sugerindo que a variação no número de amostras não teve influência significativa nos resultados obtidos.

Tabela 41 - RMSE após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas

Função	Amostras (n)		
	10	20	40
Média	10000.47519	10000.57647	10000.6750
Máximo	10001.271	10000.8371	10000.7920
Mínimo	9999.92385	10000.3021	10000.5240
Desvio Padrão	0.547211694	0.2054264866	0.1030

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Onda Eletromagnéticas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 42, destaca-se um tempo que, quando comparado com as amostras de tamanho 40 ou até 20 da primeira execução, revela um desempenho superior. Apenas para a amostra de tamanho 10 foi encontrado um resultado semelhante ao da primeira execução.

Essa observação sugere que, para amostras menores, a segunda execução obteve um tempo de execução mais eficiente em comparação com a primeira. Contudo, a eficácia

dessa melhoria parece diminuir à medida que o tamanho da amostra aumenta. Esses resultados temporais podem fornecer insights valiosos sobre o comportamento do algoritmo em diferentes condições e orientar possíveis ajustes para otimizar ainda mais o desempenho.

Tabela 42 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas

Função	Amostras (n)		
	10	20	40
Média	2152.2960	1499.6180	2402.5680
Máximo	4008.5000	1534.8900	4315.5400
Mínimo	1585.8300	1485.6700	1742.6700
Desvio Padrão	1041.0790	20.2798	1079.3082

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Onda Eletromagnéticas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Para essa função, a análise dimensional produziu resultados dentro do esperado; no entanto, a função ainda não atingiu o resultado desejado. As constantes com valores numéricos elevados e as variáveis de entrada contribuem para um aumento nos erros. Diante dessa situação, algumas estratégias podem ser consideradas para mitigar possíveis erros durante a execução do algoritmo. Observou-se também que, em muitas execuções, a variável ( $d$ ) acaba sendo anulada devido ao seu valor muito pequeno em relação a  $\omega$ . Na primeira execução, a variável de distância ( $d$ ) não foi considerada satisfatória para a função pelo código. Apresenta-se o resultado obtido com 10 amostras, evidenciando a necessidade de ajustes ou estratégias específicas para lidar com as características particulares desta função 42:

$$\frac{\omega^2 \cdot (3 \times 10^8 \cdot 3.14159)}{(3 \times 10^8 \cdot 3.14159) \cdot (\sqrt{\omega \cdot 3 \times 10^8})^2} \quad (42)$$

A função 43 obtida para uma amostra de tamanho 20 já considera ambas as variáveis, no entanto, ainda se distancia do resultado desejado. Realizando a simplificação da função, observa-se que a variável da distância ( $d$ ) se anula:

$$\left( \frac{d \cdot \omega \cdot \left( \sqrt{\left( \frac{3 \times 10^8}{3.14159} \right)^2} \right)}{\frac{(\sqrt{d^2 \cdot 3 \times 10^8}) \times 3 \times 10^8}{3.14159}} \right) \quad (43)$$

A função 44 obtida para uma amostra de tamanho 40 mostra, ao analisar as va-

riáveis, que mesmo a variável da distância ( $d$ ) aparecendo, ela se anula, tornando-se dispensável:

$$3.14159 \cdot \left( 3 \times 10^8 \cdot \left( \frac{3 \times 10^8}{3 \times 10^8 \cdot \left(\frac{d}{d}\right)} \cdot \frac{d}{d \cdot (3.14159 \cdot (3 \times 10^8)^2)} \right) \right) \cdot \omega \quad (44)$$

### 6.12.9 Análise dos resultados para as grandezas

A utilização de grandezas físicas revelou-se benéfica para o desempenho do algoritmo BRKGP. Observa-se uma melhoria nas funções de saída, mesmo quando não coincidem exatamente com as funções de referência da literatura. Em tais casos, o algoritmo frequentemente converge para funções que respeitam a análise dimensional física das unidades envolvidas.

É importante ressaltar a necessidade de explorar abordagens adicionais para mitigar os erros que surgem durante o processo. Esses erros podem representar obstáculos para a convergência para o ótimo da função. Aspectos como o uso de uma pilha de tamanho considerável e a estratégia de adotar a base  $10^{10}$  para atenuar erros numéricos associados a valores muito elevados merecem uma avaliação mais detalhada.

Outras considerações incluem a possibilidade de ajustes nos parâmetros do algoritmo, visando otimizar seu desempenho e superar desafios específicos encontrados durante a execução. Em conjunto, essas reflexões sugerem a importância de uma abordagem mais holística e aprofundada na busca por soluções robustas e eficazes no contexto do problema em questão.

### 6.13 **Implementação de novas funções unárias**

Outro ponto relevante deste estudo envolve a implementação de funções exponenciais e logarítmicas. Nessa segunda fase, foram realizados testes apenas com funções exponenciais, visto que duas das funções selecionadas possuíam esse tipo de operação.

Como o uso de unidades não foi implementado para lidar com funções exponenciais, optou-se por realizar os testes explorando apenas o potencial dessa operação no código. Cabe ressaltar que uma das funções do estudo é uma função com entrada e saída adimensionais, a Função de Densidade de Probabilidade.

### 6.13.1 Função de Densidade de Probabilidade

A Função de Densidade de Probabilidade é a oitava função submetida a rodada de testes. A variável de entrada para essa função é o  $\theta$  que não possui dimensão (adimensional), sua saída também é expressa sem dimensão.

Os resultados deste teste, apresentados na Tabela 43, indicam um desempenho melhor nessa segunda execução, obtendo resultados que chegaram ao ótimo. Os erros gerados são pequenos como na primeira execução.

O tamanho da amostra não demonstrou um impacto direto nos resultados do erro. A diferença entre a amostra de tamanho 10 e de tamanho 40 entre a foi praticamente inexistente, a única amostra que se diferenciou das demais foi a de tamanho 20.

Tabela 43 - RMSE após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas

Função	Amostras (n)		
	10	20	40
Média	0.00	$2.66 \times 10^{-3}$	0.00
Máximo	0.00	$3.83 \times 10^{-3}$	0.00
Mínimo	0.00	$1.84 \times 10^{-3}$	0.00
Desvio Padrão	0.00	$7.60 \times 10^{-4}$	0.00

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Onda Eletromagnéticas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Nos resultados temporais apresentados na Tabela 44, destaca-se que se saíram melhores que na primeira execução, não só obtendo resultados menores de tempo, mas com maior acurácia na geração da função.

Tabela 44 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Onda Eletromagnéticas

Função	Amostras (n)		
	10	20	40
Média	299.6710	746.5864	513.6980
Máximo	309.6630	792.0490	529.7170
Mínimo	288.6150	700.6950	479.4080
Desvio Padrão	8.0435	41.4115	20.3297

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Onda Eletromagnéticas utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Abaixo, apresenta-se o resultado obtido com 10 amostras, embora o resultado não

tenha se mostrado igual à literatura encontrada nos livros, o algoritmo conseguiu usar a função exponencial que representa um avanço, já que o algoritmo sabe que precisa desse tipo de operação para chegar a uma função que represente bem aquele evento:

$$\frac{2}{\frac{1}{4} \left( \frac{1}{2} (\theta^2 + 2\sqrt{\theta^2}) + e^{\theta^2+2} \right) + 3.14159} \quad (45)$$

A função obtida para uma amostra de tamanho 20 já não considera a função exponencial como uma opção, o que pode evidenciar o erro encontrado:

$$\frac{(((\theta + 2) - \theta) \cdot \theta) \cdot (\theta - (\theta + \theta)) + 3.14159 + 3.14159}{(2 + 2)^2 + \theta \cdot (2 \cdot \theta)} \quad (46)$$

A função obtida para uma amostra de tamanho 40 mostra outra função, diferente da primeira amostra, mas que evidencia bem o evento desejado, podemos observar que essa função é a função da literatura, antes de simplificar:

$$\frac{\frac{((\theta - \theta)^2 + e^{(2 - (\theta - \theta)^2)})}{\sqrt{2}}}{\sqrt{e^{(3.14159 + (2 + \theta^2))}}} \quad (47)$$

### 6.13.2 Distribuição de Bose-Einstein

A Função de Distribuição de Bose-Einstein é a décima função submetida a rodada de testes. A variável de entrada para essa função possui duas variáveis de entrada com grandezas físicas, além de duas constantes também possuírem unidades, no entanto, nesse teste não foi considerado seu uso, já que o algoritmo ainda não é capaz de lidar com as grandezas dentro da função exponencial.

Os resultados deste teste, apresentados na Tabela 45, indicam um resultado similar a primeira execução. Os resultados do erro não se mostraram muito diferente, mesmo com uma abordagem utilizando a função exponencial.

Nos resultados temporais apresentados na Tabela 46, destaca-se uma melhora no tempo de execução, sinalizando uma possível melhora, no entanto, é preciso ressaltar um dos resultados que consumiu tempo excessivo, que alcançou 40.807,6 segundos, foi a execução que mais consumiu tempo, lembrando que nessa execução com funções exponenciais, optou-se por um *noImprovementMax* igual a 50, visto que as execuções com 100 consumiam tempo computacional muito maiores. Nota-se um efeito negativo quando funções exponenciais são utilizadas como função unária nos testes.

Tabela 45 - RMSE após a implementação da análise da unidade entre 5 execuções para a Distribuição de Bose-Einstein

Função	Amostras (n)		
	10	20	40
Média	$2.22 \times 10^{-2}$	$4.39 \times 10^{-2}$	$2.24 \times 10^{-2}$
Máximo	$3.81 \times 10^{-2}$	$6.21 \times 10^{-2}$	$2.54 \times 10^{-2}$
Mínimo	$1.73 \times 10^{-2}$	$2.11 \times 10^{-2}$	$2.04 \times 10^{-2}$
Desvio Padrão	$8.93 \times 10^{-3}$	$1.48 \times 10^{-2}$	$2.12 \times 10^{-3}$

Legenda: Resultados do RMSE médio, Máximo, Mínimo e Desvio Padrão para a função da Distribuição de Bose-Einstein utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Tabela 46 - Tempo(s) após a implementação da análise da unidade entre 5 execuções para a Distribuição de Bose-Einstein

Função	Amostras (n)		
	10	20	40
Média	1212.9540	1759.0880	16148.1640
Máximo	1234.7400	1810.0000	40807.6000
Mínimo	1189.7400	1705.3500	9448.1900
Desvio Padrão	19.2305	41.2278	13795.3076

Legenda: Resultados do Tempo(s) médio, Máximo, Mínimo e Desvio Padrão para a função da Distribuição de Bose-Einstein utilizando as grandezas físicas, com base em 5 execuções

Fonte: O autor, 2023.

Observa-se que o resultado ainda são equações complexas e que não satisfazem a grandeza física desejada, aqui também foi implementado a estratégia de usar a base  $10^{10}$  com o intuito de mitigar erros de arredondamento, no entanto, não surgiu efeitos positivos, dito isto, a estratégia a seguir com essa função é utilizar outra abordagem de pré-processamento ou uma calibração dos parâmetros que comporte melhor esse tipo de operação. Abaixo temos a função obtida com a amostra de tamanho 10:

$$\left( \frac{T + T}{\omega} \left( 1 \times 10^{10} - \frac{\omega}{(T + \sqrt{(0.000662607 + 0.00138065) \times 1 \times 10^{10}}) \sqrt{\log(T)}} \right) + \frac{1}{T} \right) + \sqrt{\sqrt{0.00138065}} \quad (48)$$

A função obtida para uma amostra de tamanho 20 ainda não está distante da primeira, sua função se encontra complexa e ainda distante do desejado, além de não satisfazer a análise dimensional das grandezas físicas.

$$\frac{T \times 0.00138065 - \log(T) + 0.00138065}{0.000662607 \times \left( \frac{\omega}{1 \times 10^{10} - \sqrt{\left( \left( \frac{\omega + \omega}{T} + \omega \right) \times \sqrt{\omega} \right)}} \right)} - \sqrt{0.000662607} \quad (49)$$

A função obtida para uma amostra de tamanho 40 também não se saiu muito bem, podemos observar que essa função além de complexa não se enquadra com a análise dimensional.

$$\left( \frac{\left( \left( \frac{\omega \times 0.000662607 - T}{T} \right) + T + T \right) \times 1 \times 10^{10}}{\omega} \times 1 - \sqrt{\sqrt{0.00138065}} \right) - e^{(1-e^1)} \quad (50)$$

### 6.13.3 Sobre a implementação de novas funções

Cada nova função adicionada ao algoritmo a partir deste ponto precisa ser analisada em relação ao seu comportamento, uma vez que cada operação numérica afeta de maneira distinta as grandezas físicas. Além disso, como mencionado anteriormente, cada operação binária tem um impacto variado, e é importante observar que a adição não tem o mesmo efeito na grandeza que a multiplicação. Dessa forma, podemos concluir que a expansão desse tipo de abordagem tem o potencial de aumentar a eficácia dos resultados.

A função em destaque aqui é a Função de Densidade de Probabilidade, já que foi a função que conseguiu mitigar todos os problemas que ocorreram na primeira execução,

chegando inclusive ao ótimo desejado.

## 6.14 Regressão Linear Simples

Foram conduzidos testes visando determinar a regressão linear simples com os dados do estudo, com o propósito de compará-la com os resultados preliminares e com aqueles obtidos após a implementação das grandezas. Foram escolhidas as funções que apresentaram menor erro em cada conjunto de amostras.

Na Tabela 47 e Tabela 48, são apresentados os valores do RMSE para cada Função da reta. Observa-se que, na maioria dos casos, o algoritmo demonstrou desempenho superior, com exceção da Período Orbital, onde a Regressão Linear alcançou menor erro. Adicionalmente, na primeira amostra da onda Eletromagnética, a regressão linear simples também apresentou desempenho superior em comparação ao algoritmo. Entretanto, para as demais amostras, o algoritmo revelou resultados mais eficazes em relação à Regressão Linear.

Tabela 47 - Resultados da Regressão Linear Simples para Diferentes Funções e Amostras

RMSE	Dist. Euclidiana	Massa Relativística	Lentes Delgadas	Prod.Interno	Força Gravit.
Amostra 10	3.578563578	27.61895586	0.05052	29.56875	4.26E+27
Amostra 20	4.285103506	13.49871	0.03171	62.58769	1.23E+28
Amostra 40	4.470066116	32.18682	0.03616	41.51577	2.07E+27
Amostra 80	4.710397078	37.09559	4.29E-02	5.97E+01	3.79E+27

Legenda: Resultados da Regressão Linear Simples para Diferentes Equações e Amostras

Fonte: O autor, 2023.

Tabela 48 - Resultados da Regressão Linear Simples para Diferentes Funções e Amostras

RMSE	Per. Orbital	Campo Eletr.	Dens. Prob.	Ondas Eletromag.	Boson-Einstein
10	0.124	$5.75 \times 10^{26}$	0.0296	$2.24 \times 10^9$	0.146
20	0.0686	$8.60 \times 10^{25}$	0.039	$2.47 \times 10^9$	0.196
40	0.0882	$5.86 \times 10^{27}$	0.0533	$2.11 \times 10^9$	0.253
80	0.0939	$9.61 \times 10^{27}$	0.0514	$2.07 \times 10^9$	0.188

Legenda: Resultados da Regressão Linear Simples para Diferentes Equações e Amostras

Fonte: O autor, 2023.

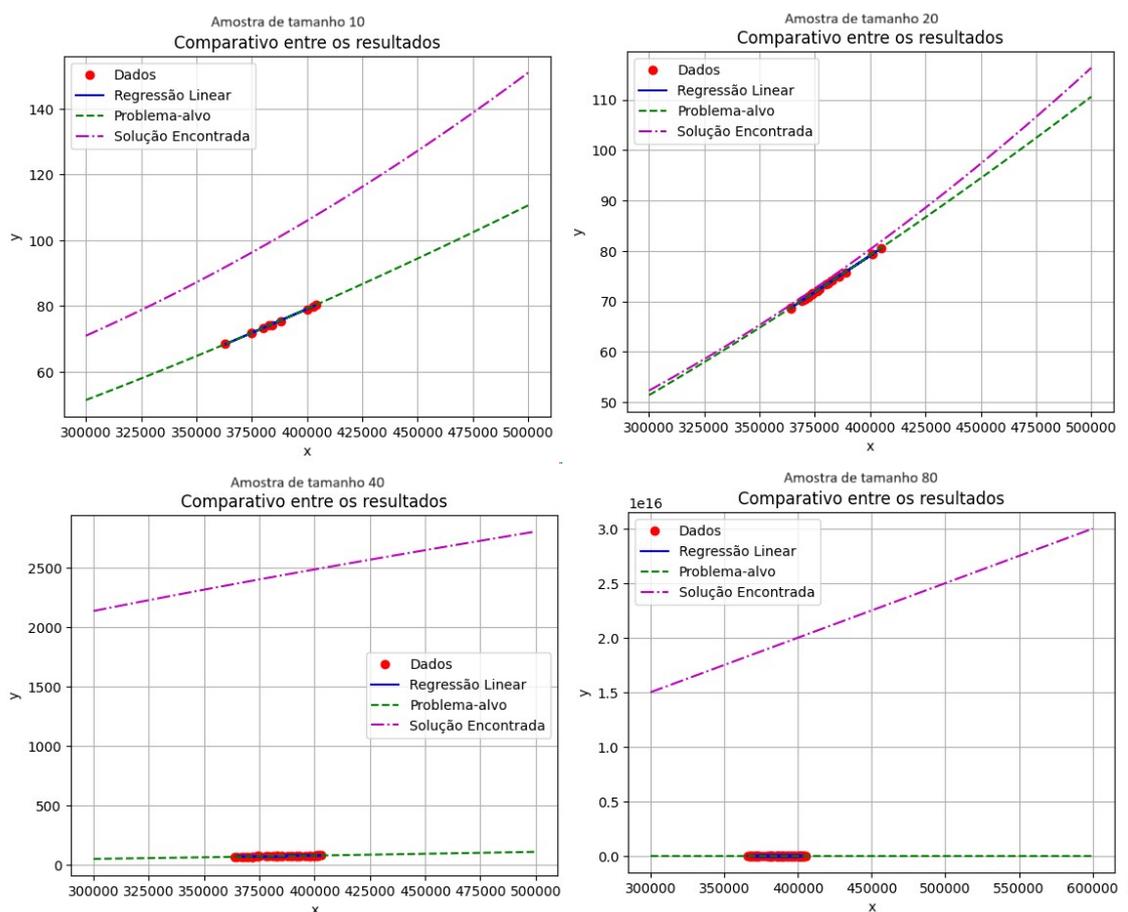
Com base nos resultados apresentados, é possível inferir que o algoritmo em questão demonstra eficácia ao produzir resultados superiores quando comparado à abordagem da Regressão Linear.

### 6.14.1 Comparação Gráfica

Neste tópico, será conduzida uma análise comparativa entre duas funções, a Função do Período Orbital e a Função de Densidade de Probabilidade, ambas pertencentes ao  $\mathbb{R}^2$ . A análise será realizada por meio de gráficos, proporcionando uma visão abrangente do comportamento das funções em questão.

Iniciaremos a exploração pela Função do Período Orbital, seus gráficos podem ser observados na Figura 14. Durante a primeira fase de testes, as quatro funções avaliadas não apresentaram um desempenho satisfatório de forma geral. No entanto, algumas delas conseguiram obter resultados promissores, adaptando-se de maneira eficaz ao problema em questão.

Figura 14 - Gráfico comparativo para a Função do Período Orbital, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada

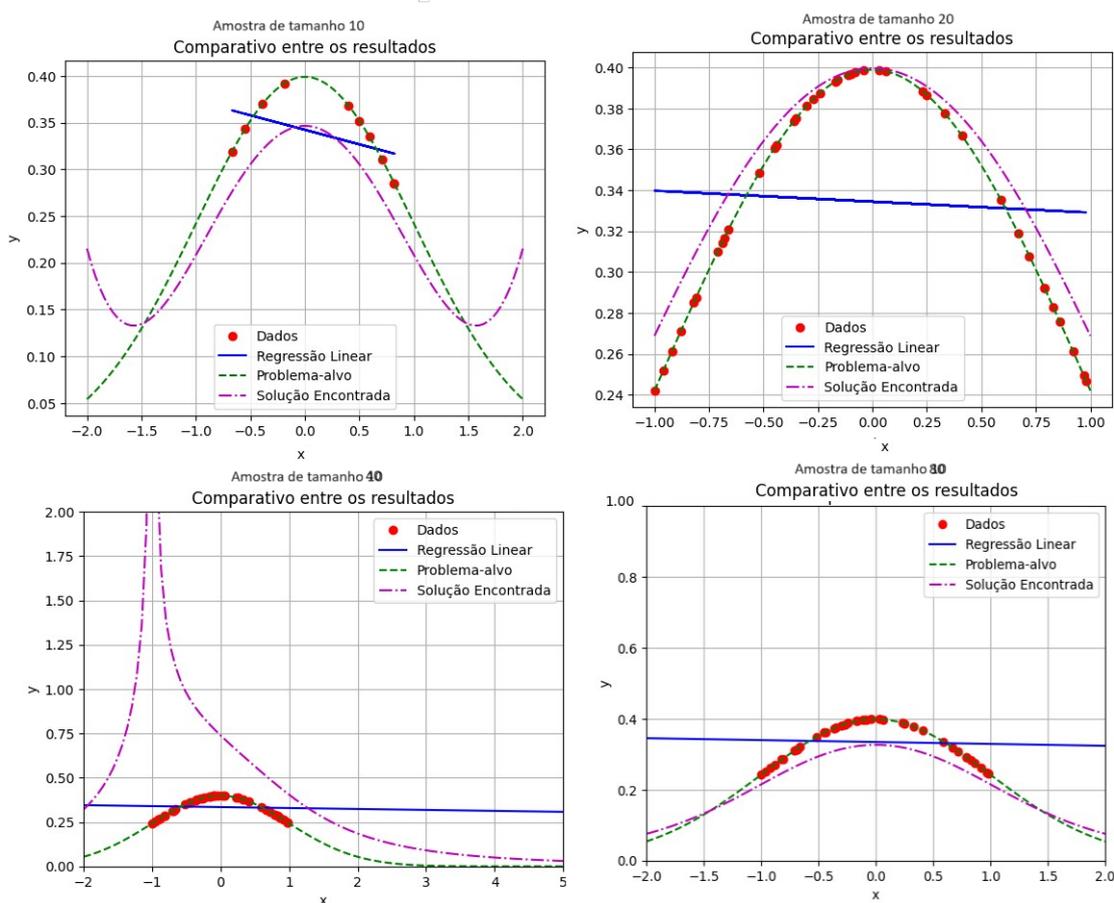


A representação gráfica revela que a função mais ajustada aos dados corresponde àquela com um tamanho amostral de 20, embora não seja a que apresenta o menor erro identificado. A função com o menor erro é associada a um tamanho amostral igual a 10, evidenciando, no entanto, uma distância significativa em relação aos pontos originais. As funções com tamanhos amostrais de 40 e 80, ao contrário, não demonstraram impacto

positivo, sendo superadas pela Regressão Linear.

A Função de Densidade de Probabilidade, seu gráfico pode ser visualizado na Figura 15 se adequou bem aos resultados na primeira fase, mesmo não tendo o suporte das funções exponenciais, ele chegou perto, nesse caso a reta acaba não sendo uma boa opção, mesmo na amostra de tamanho 40, os resultados se encaixaram melhor que na reta.

Figura 15 - Gráfico comparativo para a Função da Distribuição de Probabilidade, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada



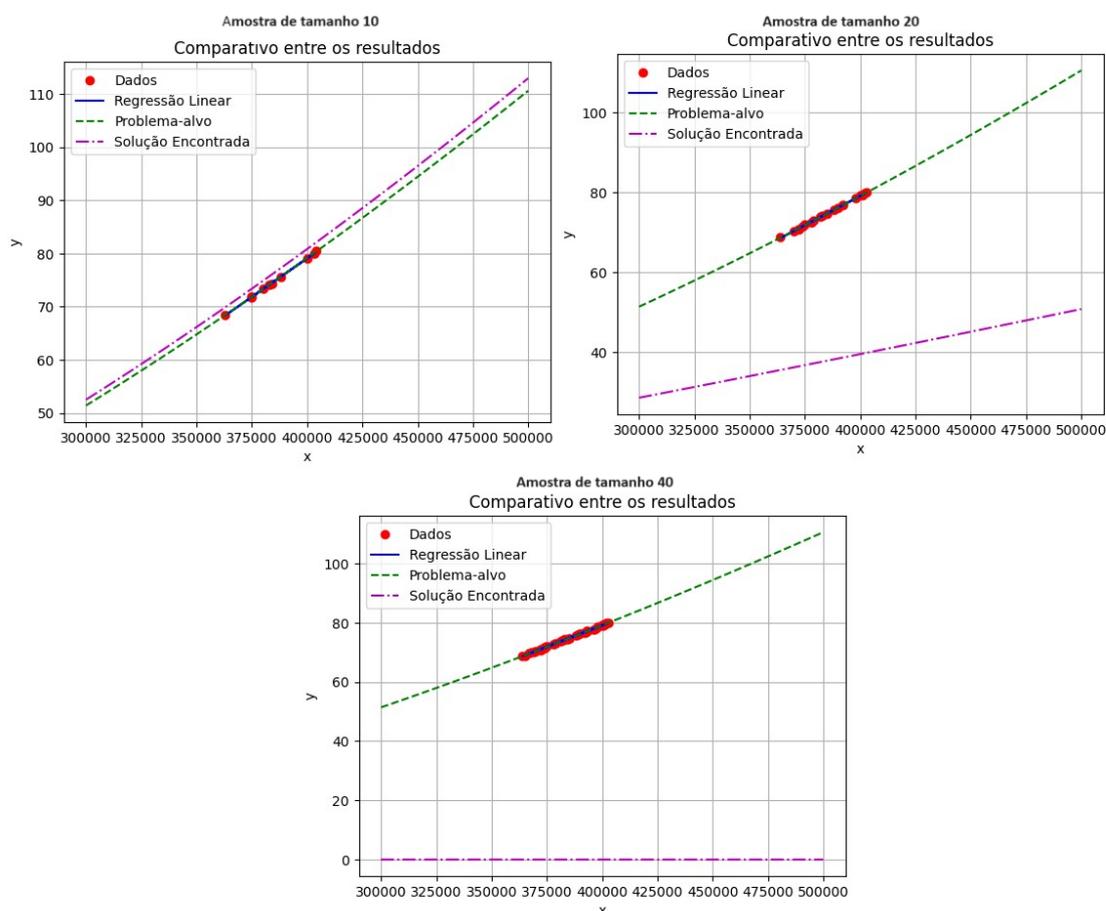
Ao analisarmos o erro, observamos que a função que melhor se ajustou foi a primeira, com um tamanho de amostra igual a 10. Entretanto, visualmente, as funções com tamanhos de amostra 20 e 80 também apresentaram ajustes adequados. A função gerada pela amostra de tamanho 40 foi a única que se distanciou um pouco, porém ainda superou o desempenho da reta de ajuste.

Avançando para a segunda fase do estudo, exploraremos os gráficos correspondentes à implementação de novas funções, incluindo a função exponencial. Vale ressaltar que, mesmo nesta fase, as funções exponenciais não incorporam as unidades correspondentes. Para a função do Período Orbital, não ocorreu nenhuma modificação, exceto pela implementação da análise dimensional das grandezas físicas.

Analisando as funções apresentadas na Figura 16, para a função do Período Orbital, destaca-se que o desempenho do algoritmo foi mais eficaz com um conjunto menor de dados. Apesar da função apresentar um erro inferior na amostra de tamanho 40, visualmente, a solução obtida com a amostra de tamanho 10 se aproxima mais do problema-alvo.

Cumprе ressaltar que a abordagem na seleção da amostra nesta fase adota uma perspectiva distinta, indicando que a amostra de tamanho 10 está contida nas amostras de tamanho 20 e 40.

Figura 16 - Gráfico comparativo para a Função do Período Orbital, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada

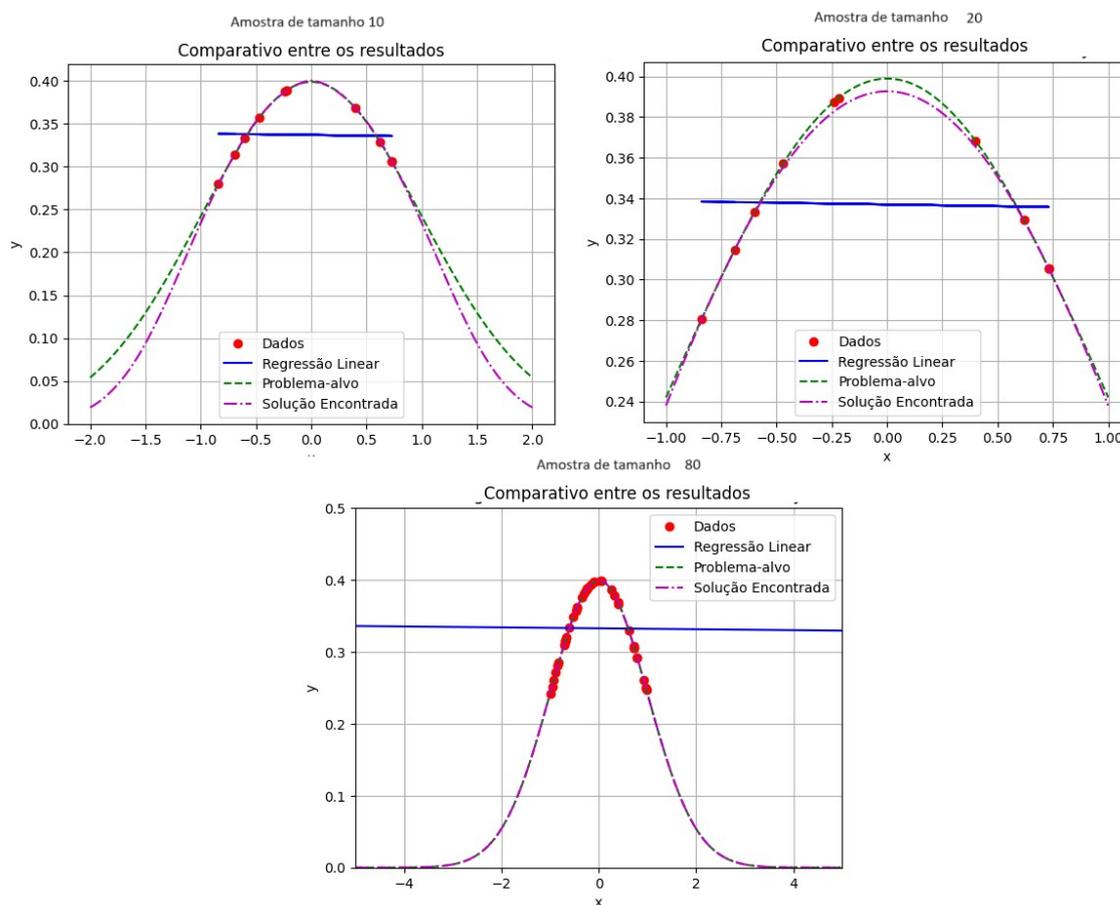


Ao analisar as funções identificadas para representar a Distribuição de Probabilidade, conforme evidenciado na Figura 17, destaca-se um desempenho superior quando há uma quantidade maior de dados. Notavelmente, a função encontrada na amostra de tamanho 40 coincidiu exatamente com aquela documentada na literatura. É relevante ressaltar que o erro na amostra de tamanho 10 foi igual a zero, e a função derivada dessa amostra demonstrou uma adaptação satisfatória aos dados, embora sua eficácia se restrinja ao intervalo específico considerado.

No caso da função associada ao tamanho amostral igual a 20, observou-se uma função razoável em um intervalo limitado, destacando-se que essa função não apresenta

características de uma função exponencial.

Figura 17 - Gráfico comparativo para a Função da Distribuição de Probabilidade, entre a Regressão Linear vs O Problema-alvo vs A Solução Encontrada



### 6.15 Teste Durbin

O teste de Durbin-Watson é uma ferramenta estatística utilizada para avaliar a presença de autocorrelação nos resíduos de um modelo de regressão. O teste é particularmente aplicável em análises de séries temporais e regressões lineares, onde a suposição de independência dos erros é fundamental.

Esse teste tem sua importância residindo na sua capacidade de identificar padrões sistemáticos nos resíduos do modelo, indicando se há correlação serial entre observações adjacentes. A autocorrelação dos resíduos compromete a eficiência dos estimadores, podendo gerar estimativas viesadas e imprecisas. Assim, ao detectar a presença de autocorrelação, o teste de Durbin-Watson orienta a necessidade de ajustes no modelo para garantir a validade das inferências estatísticas.

O valor crítico do teste é comparado ao resultado estatístico obtido, e se o valor

observado estiver significativamente abaixo ou acima da faixa crítica, sugere-se autocorrelação positiva ou negativa.

#### 6.15.1 Amostra de tamanho 10

O estudo revelou uma variedade de resultados nos testes de Durbin aplicados às diferentes equações analisadas. Vale destacar que a presença de valores NaN (Not a Number) em alguns testes, como observado no Produto Interno, indica uma limitação na avaliação dessas formulações específicas. Isso pode ser atribuído a características particulares das equações ou peculiaridades nas soluções encontradas pelo algoritmo.

Ao considerar os resultados obtidos nos demais testes, é possível observar que a maioria das pontuações ficou abaixo de 2, sugerindo ausência de autocorrelação significativa nos resíduos. Essa tendência indica que as equações testadas apresentaram, em sua maioria, uma boa capacidade em lidar com padrões temporais nos dados.

Entretanto, a análise mais aprofundada pode revelar nuances nos resultados, como pequenas variações nas pontuações que ainda se mantêm abaixo do limite aceitável de 2. Essas variações podem ser indicativos de diferentes níveis de eficácia entre as equações, proporcionando uma oportunidade para uma análise mais detalhada da influência de cada formulação na captura de padrões temporais nos resíduos.

#### 6.15.2 Amostra de tamanho 20

Os resultados dos testes de Durbin-Watson para a amostra de 20 apresentam uma variedade de pontuações nas diferentes equações analisadas. Vale notar que, similar à amostra de 10, a presença de valores NaN em alguns testes, como observado na Produto Interno e Força Gravitacional, indica limitações na avaliação dessas formulações específicas.

Ao considerar os resultados nos demais testes, observamos pontuações que, em geral, se mantiveram abaixo de 2, indicando a ausência de autocorrelação significativa nos resíduos. Essa tendência sugere que as equações, na amostra de 20, também apresentaram uma boa capacidade em lidar com padrões temporais nos dados.

Entretanto, como mencionado anteriormente, pequenas variações nas pontuações podem indicar diferentes níveis de eficácia entre as equações. As análises detalhadas dessas variações podem fornecer percepções sobre a influência de cada formulação na captura de padrões temporais nos resíduos, permitindo uma avaliação mais aprofundada da robustez do modelo proposto.

### 6.15.3 Amostra de tamanho 40

Na amostra de 40, os resultados dos testes de Durbin-Watson indicam variações nas pontuações para diferentes equações. Como mencionado anteriormente, a presença de valores NaN em alguns testes pode limitar a interpretação desses resultados específicos, como observado na Massa Relativística, Produto Interno, Força Gravitacional e Distribuição de Probabilidade.

Contudo, analisando os testes disponíveis, as pontuações em geral se mantiveram em níveis que indicam uma ausência geral de autocorrelação significativa nos resíduos. As pontuações inferiores a 2 sugerem que as equações, na amostra de 40, ainda conseguem lidar de maneira eficaz com os padrões temporais nos dados.

A variação nas pontuações entre as equações destaca a influência das formulações específicas na capacidade de capturar os padrões temporais nos resíduos. Uma análise mais detalhada dessas variações pode fornecer percepções valiosas sobre como cada função se comporta em relação à autocorrelação, contribuindo para uma compreensão mais aprofundada da eficácia do modelo proposto.

### 6.15.4 Amostra de tamanho 80

Na amostra de 80, os resultados dos testes de Durbin-Watson para diferentes equações apresentam variações nas pontuações. Observa-se novamente a presença de valores NaN em alguns casos, como na Massa Relativística, Produto Interno e Período Orbital, o que pode limitar a interpretação desses resultados específicos.

Entretanto, considerando os testes disponíveis, as pontuações ainda se encontram em níveis que sugerem uma ausência geral de autocorrelação significativa nos resíduos. Pontuações inferiores a 2 indicam que, mesmo na amostra de 80, as equações mantêm sua eficácia em lidar com padrões temporais nos dados.

A variação nas pontuações entre as equações destaca mais uma vez a influência das formulações específicas na capacidade de capturar padrões temporais nos resíduos. Uma análise mais aprofundada dessas variações pode fornecer percepções adicionais sobre o desempenho relativo de cada função em relação à autocorrelação.

## CONCLUSÕES

Em resumo, os resultados obtidos por meio da aplicação da meta-heurística BRKGP em um algoritmo de Regressão Simbólica representam uma significativa contribuição para a validação de funções físicas tanto na primeira como na segunda execução. Ao explorar essa abordagem, nossa pesquisa visa não apenas melhorar a precisão na modelagem de fenômenos físicos, mas também estender esse aprimoramento para outras áreas do conhecimento.

A eficácia dessa combinação inovadora de técnicas representa um passo importante na direção de avanços mais amplos na modelagem simbólica. Essa abordagem não se limita apenas à física, mas tem o potencial de influenciar positivamente a resolução de problemas complexos em diversos campos. Além de validar funções fundamentais na física, a intenção é proporcionar uma metodologia robusta que possa ser aplicada em contextos multidisciplinares.

Ao concluir esta etapa do estudo, destacamos a observação de que a eficiência dos resultados está intrinsecamente ligada à natureza das operações presentes nas funções e aos dados de entrada. É evidente que na primeira fase dos testes as operações binárias obtiveram melhores resultados, como observado na função das Lentes Delgadas e na do Produto Interno, enquanto as funções que possuíam algum traço de operação unária se distanciaram do ótimo, especialmente quando combinadas com intervalos numéricos significativamente extensos. Já na segunda fase de testes, os resultados se mostraram bem mais promissores. A inserção das grandezas físicas como um objeto de verificação da função encontrada elimina funções que não conseguem representar bem aquele fenômeno, limitando o número de funções que podem ser geradas. Isso otimizou o desempenho do algoritmo não só em tempo, mas no número de operações para encontrar uma função que atenda aos requisitos. A maior dificuldade até o momento é otimizar os dados de entrada para evitar os erros numéricos, já que isso é um fator que pode ocasionar em erros indesejáveis.

Os resultados notáveis obtidos na primeira execução para as funções do Produto Interno e das Lentes Delgadas representam um avanço significativo, pois o algoritmo conseguiu identificar com precisão as funções originais. Para a segunda execução, o algoritmo acertou não só a função do Produto Interno e das Lentes Delgadas, mas também a função da Distância Euclidiana usando as grandezas físicas, bem como a função de Distribuição de Probabilidade. Cabe destacar que esta última é uma função adimensional, e a inserção da função exponencial foi crucial para obter o resultado desejado. O estudo proporcionou percepções valiosas, abrindo novas perspectivas e oportunidades para investigações futuras.

Nesse contexto, a adaptação contínua do método, considerando as especificidades

das operações e intervalos, emerge como um aspecto para otimizar os resultados e explorar todo o potencial do algoritmo proposto. Cabe ressaltar que alguns dos nossos resultados de tempo se saíram melhores que no trabalho de Udrescu e Tegmark [7]. Além de utilizarmos menos dados para testes, para a Distância Euclidiana, ele utiliza 100 dados de entrada e necessita de 544 segundos, em nosso trabalho foi necessário apenas 10 dados e levou 393 segundos. Se aumentarmos para uma amostra com 20 dados, o tempo melhora para 103 segundos. Além disso, esse mesmo trabalho faz a comparação dos seus resultados com um solver Eureka, que não consegue encontrar a função da Distribuição de Probabilidade e a da Distância Euclidiana, funções essas encontradas pelo algoritmo proposto.

## Problemas e Limitações

Alguns resultados na primeira execução revelaram funções com indeterminações do tipo divisão por zero, onde após executar as operações encontramos um 0 no denominador. Esse é um aspecto que denota a não otimização completa e a presença de redundâncias nos resultados gerados pelo algoritmo. Após a simplificação das funções, observou-se a ocorrência de indeterminações do tipo  $\frac{x}{0}$ , introduzindo um desafio inesperado neste estudo. Na segunda fase dos testes com a implementação das grandezas físicas, percebemos que esse tipo de indeterminação não se faz presente. Logo, podemos concluir que o uso de unidades pode contribuir para evitar esse tipo de problema. Contudo, a implementação GiNaC, uma biblioteca de álgebra simbólica, só foi utilizada para realizar a análise das grandezas; ainda não foi implementada a simplificação das funções de saída. Isso pode ser observado ao ver os resultados de saída da segunda fase de testes. No entanto, não foi preciso utilizar o módulo de simplificação para mitigar o problema de indeterminação, uma vez que o uso das grandezas resolveu esse problema encontrado após a primeira fase de testes.

Uma questão desafiadora que surgiu durante a primeira e a segunda fase do estudo foi a presença de números extremamente pequenos ou grandes, levando o algoritmo a truncar frequentemente esses valores. Esse comportamento resultava em erros numéricos de truncamento, nos quais o algoritmo, por vezes, convergia para um valor nulo ou, em situações com números muito grandes, enfrentava erros de execução. Para lidar com essa problemática, adotou-se uma estratégia de pré-processamento de dados. Especificamente, essa abordagem envolveu a introdução de fatores multiplicativos ou divisivos para ajustar as constantes envolvidas nas funções. No entanto, essa estratégia apenas mitigou parcialmente o problema. Comparando a primeira execução com a segunda, nota-se que o erro sofreu uma diminuição, indicando a necessidade de outras abordagens para reduzir o erro gerado pelos dados de entrada.

Inicialmente, a maior complexidade estava relacionada às constantes, pois os in-

tervalos escolhidos para o domínio das variáveis podem ser ajustados mais flexivelmente, e outras fenomenologias podem ser escolhidas para estudo. Por exemplo, ao lidar com a constante de Planck, que é da ordem de  $6,62607015 \times 10^{-34}$ , optou-se por utilizar um método iterativo de ajuste de escala multiplicando-a por  $10^{10}$  três vezes, resultando em um valor mais manejável para o algoritmo. Essa abordagem foi aplicada em casos nos quais as constantes tinham valores muito pequenos. Entretanto, essa prática foi restrita às constantes, e é necessário considerar métodos para resolver o problema com os dados de entrada que possuem valores grandes ou pequenos. Uma abordagem a ser considerada é a normalização dos dados de entrada para realizar os testes, o que pode ajudar a mitigar esse problema.

Quando se tratava de valores muito grandes, como na massa da Terra, a estratégia era reduzir a constante para um valor mais manejável, permitindo que o algoritmo operasse de maneira eficaz. Esse processo de adaptação prévia dos dados desempenhou um papel crucial na mitigação dos desafios associados a números extremos e contribuiu para a estabilidade e eficiência do algoritmo.

## Perspectivas de pesquisa

Dado o progresso notável alcançado nesta pesquisa, é imperativo estabelecer uma orientação clara para futuras investigações. Uma das primeiras implementações futuras e a unidade de simplificação do GiNaC que pode oferecer a simplificação de funções. Muitos dos resultados gerados pelo algoritmo tanto na primeira como na segunda fase exibem funções com várias redundâncias, incluindo variáveis repetidas, indeterminações e casos em que a função resulta em zero, encontrados na primeira fase. A capacidade do GiNaC de simplificar funções pode ser um recurso valioso para aprimorar a eficiência do algoritmo na análise de respostas, eliminando redundâncias e variáveis desnecessárias. No contexto atual da pesquisa, a simplificação das funções obtidas pelo algoritmo foi realizada por meio da ferramenta Wolfram. Entretanto, ao integrar a funcionalidade de simplificação do GiNaC, o algoritmo poderá realizar esse processo internamente, resultando em uma abordagem mais autônoma e eficiente. Essa implementação oferece não apenas conveniência operacional, mas também representa uma expansão substancial na autonomia do algoritmo em futuras aplicações.

Outra abordagem que pode ser útil ao trabalho é implementar o crescimento do tamanho da pilha conforme a execução do algoritmo se inicia, um desafio são equações que são grandes e pequenas, isso modifica o tamanho da pilha, nesse teste o tamanho da pilha ficou fixo em 41 na primeira fase e 30 na segunda fase, no entanto, com uma execução que analisa isso de maneira automatizada é muito mais eficaz. A proposta é iniciar o algoritmo com uma pilha de tamanho 10, onde ele vai verificar encontrar um erro, após essa primeira

execução ele vai executar para uma pilha de tamanho 20, e se seu resultado for pior que o primeiro ele fica com o primeiro resultado, se for melhor ele continua, até encontrar o ótimo. Essa abordagem pode ser útil principalmente porque durante os testes, o tamanho da pilha influenciou o tempo de execução de algumas amostras.

No trabalho desenvolvido pela Macario (2018) acerca do EMA [67], destaca-se a eficácia do Algoritmo de Levenberg-Marquardt, que demonstrou um desempenho notável. Contudo, à medida que o algoritmo em questão continua a evoluir, é vislumbrado a realização de testes a fim de comparar e avaliar seu desempenho. A inclusão da regressão simbólica no código é uma perspectiva que pode ser promissora, pois pode gerar funções que enriqueçam ainda mais o escopo do trabalho.

Se tem ainda como propósito obter os dados provenientes do EMA e integrá-los ao nosso algoritmo. A intenção é realizar ajustes e comparações minuciosas dos resultados obtidos, visando enriquecer ainda mais o escopo deste trabalho. Ao incorporar dados do EMA, busca-se não apenas validar a eficácia do nosso algoritmo em diferentes contextos, mas também explorar oportunidades para melhorias.

Uma abordagem adicional que vem sendo testada é a aplicação do IRACE para a sintonização de parâmetros. Essa estratégia visa aprimorar ainda mais o desempenho do algoritmo, especialmente em vista dos testes futuros planejados. O IRACE é uma ferramenta valiosa para explorar e otimizar os parâmetros, permitindo a busca por configurações ideais que maximizem a eficiência do algoritmo.

Ao empregar o IRACE, planejamos conduzir experimentos sistemáticos para avaliar diferentes conjuntos de parâmetros. O objetivo é identificar combinações que conduzam a um desempenho otimizado do algoritmo.

Também está previsto realizar uma comparação entre os resultados obtidos por esta implementação e aqueles alcançados pelo OptFrame. Essa análise comparativa busca identificar pontos fortes e áreas de melhoria em ambas as abordagens, objetivando aprimorar o desempenho e a eficácia dos algoritmos de otimização.

## REFERÊNCIAS

- 1 TALBI, E.-G. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009. Citado 3 vezes nas páginas 7, 18 e 19.
- 2 ALMUFTI, S. M. Historical survey on metaheuristics algorithms. *International Journal of Scientific World*, Science Publishing Corporation, v. 7, n. 1, p. 1, 2019. Citado 2 vezes nas páginas 7 e 21.
- 3 ESCOBAR, C. M. d. S.; IZIDORO, S. C. Otimização de percurso para plotter vertical utilizando algoritmos genéticos. Citado 2 vezes nas páginas 7 e 22.
- 4 GONÇALVES, J. F.; RESENDE, M. G. C. *Biased Random-Key Genetic Programming*. 2018. Citado 9 vezes nas páginas 7, 10, 14, 25, 26, 27, 29, 30 e 31.
- 5 FRANZEN, E. Estudo e implementação da programação genética para síntese de fala. 2002. Citado 2 vezes nas páginas 7 e 28.
- 6 GUIMARAES, P. V.; JUNIOR, A. S. da S.; COELHO, I. M. Programação genética com chaves aleatórias viciadas em notação pos-fixada para o problema de regressão simbólica. *LII Simpósio Brasileiro de Pesquisa Operacional*, 2020. Citado 7 vezes nas páginas 7, 33, 35, 36, 46, 50 e 54.
- 7 UDRESCU, S.-M.; TEGMARK, M. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, American Association for the Advancement of Science, v. 6, n. 16, p. eaay2631, 2020. Citado 8 vezes nas páginas 7, 15, 33, 38, 39, 54, 69 e 112.
- 8 RESENDE, M. G. Introdução aos algoritmos genéticos de chaves aleatórias viciadas. *Simpósio Brasileiro de Pesquisa Operacional*, p. 3680–3691, 2011. Citado 2 vezes nas páginas 10 e 24.
- 9 MORGON, N. H.; CUSTÓDIO, R. Funções de base: o ajuste variacional. *Revista Chemkeys*, n. 2, p. 1–11, 2001. Citado na página 14.
- 10 GASSEN, G. et al. Proposta de um modelo de programação linear para otimização do planejamento agregado de produção de brocas para empresa multinacional. *Revista Produção Online*, v. 19, n. 1, p. 21–43, 2019. Citado na página 14.
- 11 MEDEIROS, J. Enxame de partículas como ferramenta de otimização em problemas complexos de engenharia nuclear. *Universidade Federal do Rio de Janeiro*, 2005. Citado na página 14.
- 12 GAREY, M. R.; JOHNSON, D. S. *Computers and intractability*. [S.l.]: W H Freeman & Worth Publishers, 1979. Citado na página 14.
- 13 FLOOD, M. M. The traveling-salesman problem. *Operations research*, INFORMS, v. 4, n. 1, p. 61–75, 1956. Citado na página 14.
- 14 COOK, W. J. et al. *The traveling salesman problem: a computational study*. [S.l.]: Princeton university press, 2011. Citado 2 vezes nas páginas 14 e 18.

- 15 LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, Informs, v. 21, n. 2, p. 498–516, 1973. Citado na página 14.
- 16 POP, P. C. et al. A comprehensive survey on the generalized traveling salesman problem. *European Journal of Operational Research*, 2023. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221723005581>>. Citado na página 14.
- 17 GENDREAU, M.; POTVIN, J.-Y. et al. *Handbook of metaheuristics*. [S.l.]: Springer, 2010. v. 2. Citado na página 14.
- 18 GENDREAU, M.; POTVIN, J.-Y. et al. *Handbook of metaheuristics*. [S.l.]: Springer, 2019. v. 3. Citado 2 vezes nas páginas 14 e 16.
- 19 FEYNMAN, R.; LEIGHTON, R.; SANDS, M. *Lições de Física de Feynman, vol. 2, 2008*. [S.l.]: Bookman. Citado 9 vezes nas páginas 14, 33, 56, 58, 61, 63, 66, 71 e 74.
- 20 FEYNMAN, R. P.; LEIGHTON, R. B.; SANDS, M. *Lições de física–vol. 1. Tradução de Adriana VR da Silva e Kaline R. Coutinho. Porto Alegre: Bookman, 2008*. Citado 3 vezes nas páginas 14, 33 e 77.
- 21 FEYNMAN, R. P. *Lições de Física de Feynman: A Edição Definitiva. 4 Volumes*. [S.l.]: Editora Bookman, 2008. Citado 3 vezes nas páginas 14, 33 e 79.
- 22 FEYNMAN, R. P.; LEIGHTON, R.; SANDES, M. *Lições de Física de Feynman–Eletromagnetismo e Matéria*. [S.l.]: Porto Alegre, Bookman. 624p, 2008. Citado 2 vezes nas páginas 14 e 33.
- 23 KUBALÍK, J.; DERNER, E.; BABUŠKA, R. Multi-objective symbolic regression for physics-aware dynamic modeling. *Expert Systems with Applications*, Elsevier, v. 182, p. 115210, 2021. Citado 2 vezes nas páginas 15 e 34.
- 24 VADDIREDDY, H. et al. Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Physics of Fluids*, AIP Publishing, v. 32, n. 1, 2020. Citado 2 vezes nas páginas 15 e 34.
- 25 KOZA, J. R. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, Springer, v. 4, p. 87–112, 1994. Citado 3 vezes nas páginas 16, 25 e 26.
- 26 SILVA, C. M. d. et al. *Otimização do processo de encapsulação de resíduo do abate de Aves visando aplicação cosmética*. Dissertação (Mestrado), 2023. Citado na página 18.
- 27 ROCHA, G. R. et al. Aplicação de algoritmo genético para roteirização e carregamento de veículo. Universidade Estadual Paulista (Unesp), 2023. Citado na página 18.
- 28 ASSUNÇÃO, T. F. *Otimização de rotas para transportes de resíduos da construção civil no município de Uberlândia-mg*. Universidade Federal de Uberlândia, 2023. Citado na página 18.

- 29 ABREU, R. D. S. Aprendizado de máquina supervisionado: classificação de músicas por gênero musical. 2023. Citado 2 vezes nas páginas 18 e 19.
- 30 AGUIAR, R.; FERREIRA, A. Aplicação das principais técnicas de aprendizado de máquina na segurança viária—estado da arte. Citado na página 18.
- 31 NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. [S.l.]: Springer, 2006. Citado 2 vezes nas páginas 18 e 19.
- 32 SOUZA, M. J. F. *Inteligência Computacional para Otimização*. Ouro Preto, 2022. Citado 3 vezes nas páginas 18, 20 e 21.
- 33 BOYD, S.; VANDENBERGHE, L. *Convex Optimization*. [S.l.]: Cambridge University Press, 2004. Citado na página 19.
- 34 BERTSEKAS, D. P. *Nonlinear Programming*. [S.l.]: Athena Scientific, 1999. Citado na página 19.
- 35 BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear Programming and Network Flows*. [S.l.]: John Wiley & Sons, 2010. Citado na página 19.
- 36 ARROYO, J. E. C. et al. Heurísticas e metaheurísticas para otimização combinatória multiobjetivo. *Doutorado em Engenharia Elétrica (Tese de doutorado)-Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas*, 2002. Citado na página 19.
- 37 BECCENERI, J. C. Meta-heurísticas e otimização combinatória: Aplicações em problemas ambientais. *INPE, Sao José dos Campos*, 2008. Citado na página 20.
- 38 RODRIGUES, F. L. et al. Metaheurística algoritmo genético para solução de problemas de planejamento florestal com restrições de integridade. *Revista Árvore, SciELO Brasil*, v. 28, p. 233–245, 2004. Citado na página 20.
- 39 JÚNIOR, F. C. d. L. Algoritmo q-learning como estratégia de exploração e/ou exploração para metaheurísticas grasp e algoritmo genético. Universidade Federal do Rio Grande do Norte, 2009. Citado na página 20.
- 40 RODRIGUES, F. L. et al. Metaheurística simulated annealing para solução de problemas de planejamento florestal com restrições de integridade. *Revista Árvore, SciELO Brasil*, v. 28, p. 247–256, 2004. Citado na página 20.
- 41 JR, A. R. M.; SHANG, J.; KUPPUSAMY, S. Simulated annealing heuristics for the dynamic facility layout problem. *Computers & operations research*, Elsevier, v. 33, n. 8, p. 2431–2444, 2006. Citado na página 20.
- 42 SANTOS, F. M. d. P. et al. Coordenação de relés direcionais de sobrecorrente por meio de evolutionary particle swarm optimization. *ENCONTRO REGIONAL IBERO-AMERICANO DO CIGRÉ*, v. 15, p. 1–8, 2013. Citado na página 20.
- 43 MÜLLER, V. Otimização de layouts industriais através do método enxame de partículas. 2007. Citado na página 20.

- 44 ARIAS-ROJAS, J. S.; JIMÉNEZ, J. F.; MONTOYA-TORRES, J. R. Solving of school bus routing problem by ant colony optimization. *Revista EIA*, Escuela de ingeniería de Antioquia, n. 17, p. 193–208, 2012. Citado na página 20.
- 45 SUBRAMANIAN, A. et al. Aplicação da metaheurística busca tabu ao problema de alocação de aulas a salas em uma instituição universitária. *Revista Produção Online*, v. 11, n. 1, p. 54–75, 2011. Citado na página 20.
- 46 PIRIM, H.; EKSIÖGLU, B.; BAYRAKTAR, E. Tabu search: a comparative study. INTECH Open Access Publisher London, UK, 2008. Citado na página 20.
- 47 SANTOS, H. C. M.; FRANÇA, P. M. Meta-heurística para programação da produção com tempos de preparação dependentes da sequência. *Gestão & Produção*, 1995. Citado na página 20.
- 48 MIRANDA, M. N. de. *Algoritmos Genéticos: Fundamentos e Aplicações*. 2007. Citado na página 22.
- 49 MORAIS, J. et al. Identificação de sistemas não lineares—utilização de algoritmos genéticos para seleção de estrutura polinomial. Citado na página 22.
- 50 PACHECO, M. A. C. et al. Algoritmos genéticos: princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, v. 28, 1999. Citado na página 22.
- 51 VÁZQUEZ-RODRÍGUEZ, J. A.; PETROVIC, S. A new dispatching rule based genetic algorithm for the multi-objective job shop problem. *Journal of Heuristics*, Springer, v. 16, p. 771–793, 2010. Citado na página 22.
- 52 GONÇALVES, J. F.; RESENDE, M. G. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, Springer, v. 17, n. 5, p. 487–525, 2011. Citado 3 vezes nas páginas 22, 23 e 25.
- 53 SNYDER, L. V.; DASKIN, M. S. A random-key genetic algorithm for the generalized traveling salesman problem. *European journal of operational research*, Elsevier, v. 174, n. 1, p. 38–53, 2006. Citado na página 25.
- 54 RUIZ, E. et al. Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. *Computers & Industrial Engineering*, Elsevier, v. 133, p. 207–219, 2019. Citado na página 25.
- 55 SANTOS, A. R. d. et al. Síntese de árvores de padrões fuzzy através de programação genética cartesiana. Universidade do Estado do Rio de Janeiro, 2014. Citado na página 25.
- 56 DIAS, D. M.; PACHECO, M. A. C.; JANEIRO-RJ-BRASIL, R. de. Programação genética linear com inspiração quântica. *Pontifícia Universidade Católica do Rio de Janeiro-PUC-Rio*, 2010. Citado na página 25.
- 57 FORREST, S. et al. A genetic programming approach to automated software repair. In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. [S.l.: s.n.], 2009. p. 947–954. Citado na página 25.

- 58 ESPEJO, P. G.; VENTURA, S.; HERRERA, F. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 40, n. 2, p. 121–144, 2009. Citado na página 25.
- 59 AFZAL, W.; TORKAR, R. On the application of genetic programming for software engineering predictive modeling: A systematic review. *Expert Systems with Applications*, Elsevier, v. 38, n. 9, p. 11984–11997, 2011. Citado na página 26.
- 60 O'NEILL, M. *Riccardo Poli, William B. Langdon, Nicholas F. McPhee: A Field Guide to Genetic Programming: Lulu. com, 2008, 250 pp, ISBN 978-1-4092-0073-4*. [S.l.]: Springer, 2009. Citado na página 26.
- 61 MLADENOVIĆ, N. et al. Less is more approach: basic variable neighborhood search for the obnoxious p-median problem. *International Transactions in Operational Research*, Wiley Online Library, v. 27, n. 1, p. 480–493, 2020. Citado na página 31.
- 62 MOUSAVI, S.; GIGERENZER, G. Risk, uncertainty, and heuristics. *Journal of Business Research*, Elsevier, v. 67, n. 8, p. 1671–1678, 2014. Citado na página 31.
- 63 LIEBOVITCH, L. S. et al. Is there an error correcting code in the base sequence in dna? *Biophysical Journal*, Elsevier, v. 71, n. 3, p. 1539–1544, 1996. Citado na página 32.
- 64 GRINGS, A. *Regressão Simbólica via Programação Genética: um Estudo de Caso com Modelagem Geofísica*. 2006. 32 f. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Uberlândia, Uberlândia, 2006. Citado na página 33.
- 65 TENACHI, W.; IBATA, R.; DIAKOGIANNIS, F. I. Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *arXiv preprint arXiv:2303.03192*, 2023. Citado na página 34.
- 66 KEREN, L. S.; LIBERZON, A.; LAZEBNIK, T. A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge. *Scientific Reports*, Nature Publishing Group UK London, v. 13, n. 1, p. 1249, 2023. Citado na página 34.
- 67 SOUZA, J. M. d. et al. Sobre ferramentas para análise automatizada de algoritmos. Universidade do Estado do Rio de Janeiro, 2018. Citado 2 vezes nas páginas 36 e 114.
- 68 CAMPBELL, D. E.; KELLY, J. S. Trade-off theory. *The American Economic Review*, JSTOR, v. 84, n. 2, p. 422–426, 1994. Citado na página 58.
- 69 GOLDSTEIN, H.; POOLE, C.; SAFKO, J. *Classical mechanics*. [S.l.]: American Association of Physics Teachers, 2002. Citado na página 69.