

CENTRO UNIVERSITÁRIO MAURÍCIO DE NASSAU
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
MACHINE LEARNING

PROJETO I
TEMA: Evolução Diferencial (Differential Evolution)

LÍDER: Eclécio Carlos - Mat. Nº 01747582
PROGRAMADOR I: Aymar Reydson - Mat. Nº 01703142
PROGRAMADOR II: Sadrak da Silva - Mat. Nº 01753177
REDATOR: Samuel Dion - Mat. Nº 01720760

Natal-RN
2025

LÍDER: - Eclécio Carlos Mat. Nº 01747582
PROGRAMADOR I: - Aymar Reydson Mat. Nº 01703142
PROGRAMADOR II: Sadrak da Silva - Mat. Nº 01753177
REDATOR: - Samuel Dion - Mat. Nº 01720760

PROJETO I

TEMA: EVOLUÇÃO DIFERENCIAL

Relatório apresentado à disciplina de Machine Learning, correspondente à avaliação da 1ª Atividade Prática 2025.2 do curso de Análise e Desenvolvimento de Sistemas da Universidade Federal do Rio Grande do Norte, sob orientação do **Profº Eng. Esp. José Lindenberg de Andrade.**

Professor: Eng. Esp. José Lindenberg de Andrade.

Natal-RN
2025

RESUMO

Este relatório apresenta a aplicação do algoritmo de Evolução Diferencial (DE), um método evolutivo de otimização baseado em população. O trabalho descreve os fundamentos teóricos, os motivos de escolha da aplicação do tema, a metodologia utilizada e os resultados obtidos. A DE foi escolhida pela sua simplicidade e eficiência em problemas complexos de otimização global. O relatório também inclui a implementação em código Python, disponibilizada em apêndice, permitindo a reprodução dos experimentos.

Palavras-chave: Evolução Diferencial; Otimização; Algoritmos Evolutivos.

Sumário

1	Introdução	5
1.1	Objetivos	5
1.1.1	Objetivo Geral	5
1.1.2	Objetivos Específicos	5
2	Desenvolvimento	5
2.1	Referencial Teórico	5
2.2	Metodologia	5
2.3	Execução das Atividades	6
2.4	Resultados	6
2.5	Discussão	6
3	Conclusão	6
A	Apêndice – Código em Python	7

1 Introdução

A Evolução Diferencial (Differential Evolution – DE) é um algoritmo estocástico de otimização global inspirado em conceitos da evolução natural. Criado por Storn e Price em 1995, o método destaca-se por sua simplicidade, robustez e eficiência em encontrar soluções ótimas para problemas não lineares e de alta dimensionalidade.

A escolha deste tema deve-se à relevância crescente dos algoritmos evolutivos no campo de Machine Learning e Inteligência Artificial, sendo aplicados em ajuste de hiperparâmetros, treinamento de redes neurais e problemas complexos de engenharia. Assim, este relatório tem como objetivo estudar o funcionamento do DE, aplicá-lo em um problema de otimização e analisar seus resultados.

1.1 Objetivos

1.1.1 Objetivo Geral

Analisar e aplicar o algoritmo de Evolução Diferencial em um problema de otimização, avaliando sua eficácia.

1.1.2 Objetivos Específicos

- Descrever os conceitos fundamentais da Evolução Diferencial;
- Implementar o algoritmo em Python;
- Avaliar os resultados obtidos em um problema de otimização;
- Discutir vantagens, limitações e possíveis melhorias do método.

2 Desenvolvimento

2.1 Referencial Teórico

A DE é um algoritmo evolutivo baseado em três operadores principais: **mutação**, **crossover** e **seleção**. Diferentemente de outros métodos genéticos, o DE gera novas soluções a partir da diferença entre indivíduos da população, o que garante diversidade e eficiência na exploração do espaço de busca.

2.2 Metodologia

A metodologia adotada consiste em:

1. Gerar uma população inicial de vetores candidatos;
2. Aplicar mutação para criar novos vetores de perturbação;
3. Realizar crossover entre vetores mutados e originais;

4. Selecionar os melhores indivíduos de acordo com a função objetivo;
5. Repetir o processo até o critério de parada.

2.3 Execução das Atividades

A implementação foi realizada em Python utilizando a biblioteca `scipy.optimize.differential_evolution`, aplicada ao problema clássico da função de Rastrigin, conhecida por sua complexidade e presença de múltiplos mínimos locais.

2.4 Resultados

Os resultados indicaram que o algoritmo convergiu para valores próximos do ótimo global, mesmo em um espaço multidimensional. O desempenho foi satisfatório, confirmando a robustez do método.

2.5 Discussão

A análise mostrou que a Evolução Diferencial é eficiente em problemas não lineares, mas a escolha adequada de parâmetros (*fator de escala F , taxa de crossover CR* e tamanho da população) é essencial para o desempenho. Além disso, a simplicidade do método facilita sua implementação em diferentes cenários.

3 Conclusão

O estudo realizado permitiu compreender os fundamentos e a aplicação prática do algoritmo de Evolução Diferencial. Constatou-se que o método apresenta excelente desempenho em problemas complexos de otimização, sendo uma ferramenta relevante em Machine Learning e Inteligência Artificial.

Como trabalhos futuros, sugere-se explorar versões adaptativas do DE, bem como sua aplicação em problemas reais de engenharia e ciência de dados.

Referências

- [1] STORN, R.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 1995.
- [2] SCIPY. Differential Evolution Solver. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html. Acesso em: 21 set. 2025.
- [3] ENGELBRECHT, Andries P. *Computational Intelligence: An Introduction*. Wiley, 2007.

A Apêndice – Código em Python

Listing 1: Implementação em Python da Evolução Diferencial

```
import numpy as np
from scipy . optimize import differential_evolution
# F u n o d e Rastrigin ( problema de teste em o t i m i z a o )
def rastrigin ( x ) :
    A = 10
    return A * len ( x ) + sum ( [( xi **2 - A * np . cos (2* np . pi * xi )
        ) for xi in x ])
# D e f i n i o dos limites de busca
bounds = [( -5.12 , 5.12) ] * 5 # 5 d i m e n s e s
# E x e c u o do algoritmo DE
result = differential_evolution ( rastrigin , bounds , maxiter =1000 ,
    popsize =15 ,
    tol =0.01)

print("Melhor_solucao_encontrada:", result.x)
print("Valor_dafuncao_objetivo:", result.fun)
```