Constructing Differential Evolution via LLM Prompt Chaining

A Competition Entry on Competition on LLM-designed Evolutionary Algorithms at The Genetic and Evolutionary Computation Conference (GECCO) 2025

Dominik Papaj
dmpapaj@student.agh.edu.pl
AGH University of Krakow
Poland

Tomasz Karpiński
tkarpinski@student.agh.edu.pl
AGH University of Krakow
Poland

Rohit Salgotra
rohits@agh.edu.pl
AGH University of Krakow
Poland

Abstract

This paper presents an extended abstract, designed as a competition entry on LLM-designed Evolutionary Algorithms. We present DEuLLM, a differential evolution (DE) constructed entirely through prompt engineering with the help of a large language model (LLM). DEuLLM integrates key advances in DE, such as success-history-based parameter adaptations, distance-based feedback, linear population size reduction, and careful boundary control, without any code design outside the LLM-driven dialogue. The development process demonstrates how iterative prompting can guide the design of a competitive optimizer from scratch through natural language interaction. The results show that our proposed DEuLLM is able to locate the global optimum for 15 out of 24 test problems.

CCS Concepts

• **Theory of computation → Bio-inspired optimization**.

Keywords

Evolutionary Computation, Large Language Model, Benchmarking

ACM Reference Format:
Dominik Papaj, Tomasz Karpiński, and Rohit Salgotra. 2025. Constructing Differential Evolution via LLM Prompt Chaining: A Competition Entry on Competition on LLM-designed Evolutionary Algorithms at The Genetic and Evolutionary Computation Conference (GECCO) 2025. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3712255.3735095

1 Introduction

Evolutionary algorithms (EAs) are powerful tools for solving complex optimization problems, particularly high-dimensional, nonlinear, and multimodal instances. Among these algorithms, differential evolution (DE) has gained significant attention due to its simplicity, efficiency, and robustness [3]. Several variants of DE have consistently achieved the highest ranking in CEC competitions, showcasing their effectiveness in solving challenging optimization problems [1]. However, DE algorithms often require careful tuning of control parameters (such as mutation, crossover, population size) for better performance. To address this issue, numerous adaptive and self-adaptive DE variants have been proposed, such as jDE, SHADE, and LSHADE [4], which dynamically adjust parameters during the search process. Although these approaches have significantly improved the robustness and generalization of DE, they still rely on predefined heuristics or extensive computational overhead.

In recent years, the advent of large language models (LLMs) has opened new avenues for automating the design of algorithms [5]. These models, trained in vast datasets, are capable of generating complex code, optimizing solutions, and enhancing algorithmic structures based on user inputs and feedback. In this work, we use LLMs (Chat GPT 4.0 and Deep Seek AI) to design a DE variant named DEuLLM (differential evolution using LLM). Through a process of iterative prompt chaining, we employ an LLM to guide the algorithm's evolution, adapting the parameters and strategies required for robust performance in high-dimensional, multimodal optimization tasks.

We apply DEuLLM to the GNBG generated benchmark suite [2] to validate the results obtained. The GNBG suite is used by competition organizers as a standard to evaluate the performance of new LLM-based algorithms. With its diverse set of high-dimensional, multi-modal test functions, the suite provides an objective platform for assessing the strengths and weaknesses of optimization algorithms. By using GNBG, the competition organizers ensure that LLM-driven optimization algorithms are rigorously tested on a wide range of box-constained problem scenarios. The DEuLLM algorithm represents our entry for the Competition on LLM-designed Evolutionary Algorithms at GECCO 2025.

2 Prompt Chain: Building DEuLLM with an LLM

This section details the proposed DEuLLM algorithm. The proposed algorithm uses an iterative Prompt Chain process carried out using LLM (ChatGPT 4.0 and Deep Seek AI). This process is exemplified using a prompt chain, where each step progressively refines the algorithm. Below is a detailed overview of the prompt chaing that led to the creation of DEuLLM:

(1) **Prompt I (Initial Request):** *"I need a powerful differential evolution algorithm for high-dimensional, multi-modal problems. What's a robust baseline we can build on?"*
LLM Response: *"I recommend SHADE (Success-History Adaptive DE). It uses historical memory for parameters F and CR to adapt dynamically. Want me to draft the MATLAB implementation?"*
(2) **Prompt II (Code Generation):** *"Yes, please provide the baseline implementation of SHADE with core operators in Python."*

GECCO '25 Companion, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3735095

LLM Response: *"Here is a basic implementation of SHADE. It includes the core operators: mutation, crossover, selection, and historical memory updates ..."*

(3) **Prompt III (Enhancing Adaptation):** *"Improve parameter adaptation to incorporate feedback from successful mutations."*
LLM Response: *"To include distance-based feedback, we'll track the Euclidean distance between parent and trial vectors and use it to weight updates of F and CR."*

(4) **Pormpt IV (Mutation Improvement):** *"Boost mutation with archive-assisted strategy."*
LLM Response: *"Implemented an enhanced mutation strategy where each individual is perturbed towards a randomly chosen p-best and a differential vector from the population and archive."*

(5) **Prompt V (Boundary Handling):** *"Add reflective bounds to keep solutions feasible."*
LLM Response: *"Done. Each mutated vector is adjusted to reflect back into the bounds if it violates the constraints."*

(6) **Prompt VI (Benchmark Integration):** *"Integrate with GNBG, log evaluations, and track when $error \leq 10^{-8}$."*
LLM Response: *"Integrated EDEuLLM with GNBG. It logs function evaluation history, absolute error, and when it first reaches the target threshold (10^{-8})."*

(7) **Prompt VII (Logging for Analysis):** *"Log F, CR, distances, and evaluations per generation."*
LLM Response: *"Added a PromptLog structure with entries for F, CR, distance improvement, and function evaluations. This is saved and analyzed post hoc."*

(8) **Prompt VIII (Population Size Reduction):** *"Add population size reduction to reduce the computational burden"*
LLM Response: *"Implemented LPSR: population size reduces linearly from initial NP to a minimum value as function evaluations increase."*

(9) **Prompt IX (Final Step):** *"Prepare a final version."*
LLM Response: *"DEuLLM is ready! Includes:"*
- Adaptive F and CR with distance weights.
- Archive-assisted mutation.
- Reflective bounds.
- GNBG benchmarking.
- LPSR and full log.

The DEuLLM proposed above combines adaptive parameter control, distance-aware learning, and archive-assisted exploration to tackle high-dimensional, multimodal GNBG problem instances. Integration of historical SHADE memory with novel enhancements - including reflective boundary handling, linear population reduction (LPSR), and rigorous benchmarking via GNBG - DEuLLM achieves robust performance under strict evaluation budgets.

3 Benchmarks & Experimental Results

According to the competition rules, we have used the GNBG benchmark suite, consisting of 24 benchmark problems [2]. We kept all control parameters consistent for all the problem instances and set them as:

- Initial population 100

The benchmark settings are as follows.

- 24 problem instances - f_1 to f_{24}; $D = 30$; and 31 runs.
- Stopping Criteria: $500,000$ (f_1 to f_{15}); $1,000,000$ (f_{16} to f_{24}) function evaluations
- Acceptance threshold: Absolute error = 10^{-8}.

All experiments were carried out on Python using Visual Studio Code on a Windows 11 operating system, with an Intel Core i7 processor and 32 GB of RAM. The results are summarized in Table 1, which presents 24 problem instances along with the absolute error, reported as mean (standard deviation), and success rate.

Table 1: Results of DEuLLM on GNBG Test Suite

Instance	Absolute Error	FEs to Threshold	Success Rate [%]
f_1	6.06E-14(1.02E-13)	30506.2 (684.17)	100
f_2	0.00E+00(0.00E+00)	85578.2 (1475.10)	100
f_3	3.03E-14(2.88E-14)	47259.7 (1736.79)	100
f_4	2.27E-14(2.83E-14)	34583.2 (933.175)	100
f_5	3.36E-02(3.66E-02)	141586(5500.95)	53.33
f_6	9.33E-02(1.50E-02)		0
f_7	0.00E+00(0.00E+00)	40296.6 (1233.74)	100
f_8	0.00E+00(0.00E+00)	40815.4(1423.96)	100
f_9	0.00E+00(0.00E+00)	234206(20275.2)	100
f_{10}	2.40E-07(1.31E-06)	48208.3(2813.73)	96.66
f_{11}	8.49E-01(4.65E+00)	51245.7(3130.27)	90
f_{12}	4.51E-04(2.47E-03)	51598.9(2642.44)	96.67
f_{13}	4.67E+00(1.47E+01)		0
f_{14}	8.47E+01(1.17E+02)		0
f_{15}	5.30E+00(1.82E-01)		0
f_{16}	6.14E+02(1.30E+02)	32401(0)	3.33
f_{17}	6.07E+02(5.31E+01)		0
f_{18}	5.52E+02(2.69E+02)	52636.2(2211.49)	20
f_{19}	2.82E+02(1.51E+02)		0
f_{20}	5.00E+01(3.31E-01)		0
f_{21}	1.08E+01(1.32E+01)	136932(0)	3.33
f_{22}	2.48E+01(2.88E+01)		0
f_{23}	9.58E+00(6.19E+00)	131894(10630.1)	13.33
f_{24}	1.12E+00(1.54E+00)	188984(15686.6)	23.33

4 Conclusion

In this paper, we present the results of DEuLLM algorithm as an entry to the competition on Evolutionary Algorithms using LLM. From Table 1, we can see that DEuLLM was able to optimize 15 out of 24 problems with a success rate 100% for 7 cases (f_1, f_2, f_3, f_4, f_7, f_8 and f_9 in all 31 runs. Although, GNBG benchmark suite is from the previous year, but since we are submitting for the competition, we are providing only results taken by us and not any comparison.

References

[1] Noor H Awad, Mostafa Z Ali, and Ponnuthurai N Suganthan. 2017. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 372–379.

[2] Yazdani Danial, Mohammad Nabi Omidvar, Yazdani Delaram, Deb Kalyanmoy, and H Gandomi Amir. 2023. GNBG: A generalized and configurable benchmark generator for continuous numerical optimization. (2023). arXiv:2312.07083

[3] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11 (1997), 341–359.

[4] Ryoji Tanabe and Alex Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation*. IEEE, 71–78.

[5] Niki van Stein and Thomas Bäck. 2024. Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation* (2024).