IEEE

IEEE Computational Intelligence Society

The 7th International Conference on New Trends in Computational Intelligence

October 17-19, 2025 Jinan, China

BEST PAPER AWARD

Presented to

Yanyan Tan, Wei Zhang, Lizhuang Tan, Wei Zhang, Peiying Zhang, Jiande Sun

For the paper entitled

Collaborative Operator Selection for Multi-Objective Optimization via Large Language Models and Deep Reinforcement Learning

Collaborative Operator Selection for Multi-Objective Optimization via Large Language Models and Deep Reinforcement Learning

Yanyan Tan[1], Wei Zhang[1], Lizhuang Tan[2], Wei Zhang[2], Peiying Zhang[3], Jiande Sun[1,*]
[1]School of Information Science and Engineering, Shandong Normal University, Jinan, China
[2]Key Laboratory of Computing Power Network and Information Security, Ministry of Education,
Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[3]Qingdao Institute of Software, College of Computer Science and Technology,
China University of Petroleum (East China), Qingdao, China
Emails: yytan928@163.com, 2023028041@stu.sdnu.edu.cn, tanlzh@sdas.org,
wzhang@sdas.org, zhangpeiying@upc.edu.cn, jiandesun@hotmail.com
Corresponding author: Jiande Sun Email: jiandesun@hotmail.com

***Abstract*—Evolutionary algorithms (EAs) are among the most effective approaches for solving multi-objective optimization problems (MOPs), yet their performance heavily depends on the choice of variation operators, which often lack adaptability across different problem scenarios and evolutionary stages. To overcome this limitation, we propose a novel cooperative operator selection framework that integrates a deep Q-network (DQN) and a large language model (LLM) as dual decision-makers. The DQN leverages online learning to adaptively select operators based on state-feedback, while the LLM contributes high-level reasoning and prior knowledge. Through a customized collaboration strategy, the two models dynamically predict the most suitable crossover or mutation operator at each generation. To evaluate the proposed approach, we construct a constrained multi-objective optimization model for edge-cloud collaborative task offloading, which aims to jointly minimize task delay and energy consumption under resource and feasibility constraints. Experimental results on both this real-world application and standard benchmark problems demonstrate that the proposed algorithm achieves superior convergence and diversity performance compared to existing methods, highlighting its robustness and generalization capability.**

Index Terms*—Multi-objective optimization, evolutionary algorithms, deep reinforcement learning, large language models, operator selection, cooperative decision-making, constrained multi-objective optimization

I. Introduction

Many real-world problems involve multiple conflicting objectives, making it challenging to find a single solution that satisfies all requirements. These problems, known as multi-objective optimization problems (MOPs) [1], are effectively tackled by multi-objective evolutionary algorithms (MOEAs). MOEAs are population-based methods that do not rely on gradient information and are known for their adaptability and convergence capabilities [2].

To improve search efficiency and solution quality, numerous crossover and mutation operators have been developed [3]. For instance, GAs maintain diversity [4], DE exploits vector differences [5], and PSO uses local and global guidance [6]. However, the performance of individual operators often degrades when applied to different problem types or dynamic environments [7].

To address this, adaptive operator selection strategies based on machine learning (ML) have been proposed [8]. These approaches can be broadly classified into offline recommendation and online selection. While promising, they face challenges such as limited prediction accuracy and difficulty in strategy reasoning, particularly in high-dimensional or dynamic contexts [9].

Recently, deep reinforcement learning (DRL) and large language models (LLMs) have shown strong capabilities in decision-making tasks. DRL, especially a deep Q-network (DQN), learns operator utility through feedback [7], while LLMs excel in reasoning and have been applied in planning, code generation, and heuristic design [10]–[12].

Motivated by this, we propose a collaborative framework where DQN and LLM jointly guide operator selection. Unlike prior methods, the LLM is not used as a direct variation operator but provides high-level reasoning based on prompt-engineered population state inputs. A coordination mechanism enables the two models to dynamically select suitable traditional operators, enhancing the adaptability and intelligence of the search process.

The main contributions of this paper are as follows:

First, we propose a dual-model framework that integrates DQN with LLM for collaborative decision-making in operator selection, significantly enhancing the adaptability and intelligence of multi-objective evolutionary algorithms. This framework employs a prompt-engineering mechanism that enables the LLM to comprehensively understand population states and provide high-quality, reasoning-based decision suggestions for algorithm optimization.

Second, comprehensive experimental validation is conducted on both standard benchmark functions and real-world engineering problems including DelayEnergy optimization. The results demonstrate that our proposed method achieves superior performance in terms of convergence behavior, solution quality, and generalization capability, validating the effectiveness and practicality of the framework in solving complex multi-objective optimization problems.

II. Proposed methods

A. Collaborative Operator Selection via LLM and DQN

Algorithm 1 DQN-Based Operator Selection with Learnable GPT Invocation

Input: $\mathbf{p}$ (current solution), $\mathbf{w}$ (corresponding weight vector),
  OP (candidate operator set), Q (DQN with GPT-action), LLM (large language model)
Output: *op* (selected operator)
$state \leftarrow \{\mathbf{p}, \mathbf{w}\}$
$action \leftarrow$ DQN.ChooseAction($state$) // GPT is included as an extra learnable action in DQN
if $action == |OP| + 1$ **then**
  $features \leftarrow$ ExtractPopulationFeatures()
  $prompt \leftarrow$ EncodePrompt($features$)
  $LLM\_score \leftarrow$ zero vector of size $1 \times |OP|$
  for each $op_i \in OP$ **do**
    $LLM\_score[op_i] \leftarrow LLM(prompt, op_i)$
  end for
  $op \leftarrow$ SelectOperatorViaRoulette(LLM_score)
else
  $op \leftarrow OP[action]$ //DQN directly selects an operator
end if
return *op*

To address the dynamic nature of multi-objective optimization, we propose a dual-model framework that combines DQN and LLMs for operator selection. Unlike prior work using LLMs as variation operators, our method leverages LLMs for high-level reasoning to guide traditional operator selection.

As shown in Algorithm 1, the DQN chooses either a traditional operator or a special "invoke LLM" action based on the current solution $\mathbf{p}$ and its weight $\mathbf{w}$. When the LLM is invoked, population features are encoded into a prompt, and the LLM outputs preference scores for all candidate operators. Roulette selection based on these scores determines the final operator. This enables the DQN to learn when to rely on LLM guidance, replacing static rule-based strategies.

The DQN uses a Q-network with target network and replay buffer, where each transition tuple (s, a, r, s') includes the decision variables, weight vector, and LLM-inferred scores:

$$s = (p_1, \ldots, p_D, w_1, \ldots, w_M, l_1, \ldots, l_K) \tag{1}$$

The Tchebycheff function measures objective aggregation:

$$g^{tch}(x, w, z^*) = \max_i \{w_i(f_i(x) - z_i^*)\} \tag{2}$$

and normalized fitness improvement (NFI) is:

$$\mathrm{NFI}_x = \max\left\{\frac{1 - g^{tch}(x, w, z^*)}{g^{tch}(y, w, z^*)}, 0\right\} \tag{3}$$

The reward for each operator is the maximum NFI observed:

$$\mathrm{reward}_{op} = \max_{(op, NFI_x) \in R} \{NFI_x\} \tag{4}$$

A fully connected network approximates $Q(s, a)$ with layer sizes 128–256–128–64–32. The loss is:

$$L = \frac{1}{|T|} \sum_{t \in T} \left(Q(s_t, a_t) - \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right]\right)^2 \tag{5}$$

To maintain diversity, unselected operators are periodically forced to be explored. When LLM is invoked, selection is based on LLM-derived scores, supporting a balance between DQN exploitation and LLM-guided semantic reasoning.

This mechanism is integrated into MOEA/D, where DQN adaptively decides when to use LLM assistance. It enables context-aware, dynamic operator selection, leading to enhanced performance and generalization across various MOP scenarios.

B. Large Language Models for Operator Selection

In the proposed framework, LLMs act as decision-making assistants that recommend operators based on the current optimization state. Unlike conventional numeric strategies, LLMs interpret structured natural language prompts containing semantic and statistical context to guide selection.

At each generation, a Top-K set of candidate operators is shortlisted by the DQN. The LLM then selects one operator from this set based on a dynamically constructed prompt, which includes:

- Problem name and current optimization progress;
- Generation number and summary statistics;
- For each candidate: name, usage count, and recent performance score.

TABLE I
A COMPARISON OF RESULTS ON ZDT AND DTLZ INSTANCES IN TERMS OF HV.

Problem	m	d	MOEA/D	MOEA/D-DE	MOEA/D-DQN	MOEA/D-LO	MOEA/D-LPOS
ZDT1	2	30	7.1830e-1 (6.34e-4) -	7.1691e-1 (1.48e-3) -	7.2020e-1 (7.58e-5) -	7.1938e-1 (1.02e-3) -	**7.2026e-1 (2.92e-5)**
ZDT2	2	30	4.4009e-1 (2.51e-3) -	4.4269e-1 (1.10e-3) -	4.4484e-1 (7.31e-5) -	4.4388e-1 (2.24e-4) -	**4.4495e-1 (5.59e-5)**
ZDT3	2	30	**6.0833e-1 (2.81e-2)** +	5.9809e-1 (6.25e-4) =	5.9797e-1 (2.52e-5) =	5.9772e-1 (9.16e-4) -	5.9796e-1 (1.58e-5)
ZDT4	2	30	1.9334e-1 (1.44e-1) -	0.0000e+0 (0.00e+0) -	3.0155e-1 (1.59e-1) -	4.0585e-1 (3.08e-1) =	**5.4891e-1 (7.93e-2)**
ZDT6	2	30	3.4046e-1 (4.87e-3) -	3.7024e-1 (3.03e-2) =	3.8890e-1 (2.11e-5) =	3.8778e-1 (4.06e-4) -	**3.8891e-1 (3.98e-7)**
DTLZ1	3	7	8.4092e-1 (1.85e-3) =	7.6802e-1 (1.06e-1) -	8.3882e-1 (3.57e-3) -	6.8954e-1 (1.01e-1) -	**8.4192e-1 (2.45e-3)**
DTLZ2	3	12	**5.6276e-1 (6.37e-5)** +	5.3254e-1 (1.89e-3) -	5.5915e-1 (7.49e-4) -	5.3547e-1 (1.37e-3) -	5.6230e-1 (1.77e-4)
DTLZ3	3	12	2.6311e-1 (2.41e-1) -	1.8798e-1 (2.31e-1) -	3.6599e-1 (2.59e-1) =	0.0000e+0 (0.00e+0) -	**4.0871e-1 (2.33e-1)**
DTLZ4	3	12	4.3804e-1 (1.57e-1) -	5.0660e-1 (3.10e-2) -	5.6047e-1 (1.11e-3) -	5.3547e-1 (2.21e-3) -	**5.6239e-1 (3.28e-4)**
DTLZ5	3	12	1.8337e-1 (2.35e-5) -	**1.9555e-1 (8.71e-5)** +	1.9264e-1 (4.27e-5) -	**1.9473e-1 (1.64e-4)** +	1.9268e-1 (1.65e-5)
DTLZ6	3	12	1.8332e-1 (3.19e-5) -	**1.9593e-1 (1.90e-5)** +	1.9268e-1 (1.64e-5) -	**1.9587e-1 (4.38e-5)** +	1.9269e-1 (1.11e-5)
DTLZ7	3	22	2.5749e-1 (1.21e-3) -	2.1205e-1 (1.59e-2) -	2.6532e-1 (6.78e-4) -	2.3509e-1 (7.02e-4) -	**2.6576e-1 (3.11e-4)**
DelayEnergyEC	2	3	9.3099e-2 (3.13e-3) -	9.9112e-2 (1.55e-5) -	9.9124e-2 (1.14e-5) -	9.8581e-2 (5.95e-4) -	**9.9450e-2 (3.36e-4)**
+/-/=			2/10/1	2/9/2	0/10/3	2/10/1	

TABLE II
A COMPARISON OF RESULTS ON ZDT AND DTLZ INSTANCES IN TERMS OF IGD.

Problem	m	d	MOEA/D	MOEA/D-DE	MOEA/D-DQN	MOEA/D-LO	MOEA/D-LPOS
ZDT1	2	30	4.6834e-3 (4.50e-4) -	5.1703e-3 (7.41e-4) -	3.8954e-3 (1.00e-5) -	4.3424e-3 (8.09e-4) -	**3.8891e-3 (2.96e-6)**
ZDT2	2	30	5.9820e-3 (1.37e-3) -	4.2948e-3 (3.31e-4) -	3.8240e-3 (7.45e-6) -	4.1085e-3 (7.62e-5) -	**3.8137e-3 (5.33e-6)**
ZDT3	2	30	1.6180e-2 (9.26e-3) -	**1.0818e-2 (9.06e-5)** +	1.0936e-2 (1.56e-5) =	1.1551e-2 (3.29e-4) -	1.0944e-2 (2.12e-5)
ZDT4	2	30	5.1144e-1 (1.79e-1) -	7.2974e+0 (3.13e+0) -	3.8773e-1 (1.70e-1) -	3.6383e-1 (4.53e-1) =	**1.3340e-1 (6.36e-2)**
ZDT6	2	30	3.7288e-2 (4.21e-3) -	1.7454e-2 (2.39e-2) =	3.1053e-3 (1.75e-6) =	3.7582e-3 (2.44e-4) -	**3.1043e-3 (4.58e-7)**
DTLZ1	3	7	**1.9376e-2 (3.23e-4)** =	3.9110e-2 (3.26e-2) -	2.0134e-2 (1.08e-3) =	7.3215e-2 (4.03e-2) -	1.9509e-2 (6.18e-4)
DTLZ2	3	12	**5.0308e-2 (2.57e-6)** +	6.9635e-2 (6.90e-4) -	5.1880e-2 (3.43e-4) =	7.0677e-2 (1.38e-3) -	5.2028e-2 (4.71e-4)
DTLZ3	3	12	1.2379e+0 (1.64e+0) -	8.2970e+0 (1.25e+1) -	5.0265e-1 (6.90e-1) =	1.1472e+1 (4.61e+0) -	**3.3712e-1 (5.03e-1)**
DTLZ4	3	12	3.0054e-1 (3.19e-1) =	1.7150e-1 (9.29e-2) -	5.8194e-2 (2.55e-3) -	1.0054e-1 (9.16e-3) -	**5.5750e-2 (8.02e-4)**
DTLZ5	3	12	3.1165e-2 (6.34e-5) -	**1.2154e-2 (7.91e-5)** +	1.8467e-2 (8.73e-5) +	1.2605e-2 (3.32e-4) +	1.8540e-2 (4.28e-5)
DTLZ6	3	12	3.1220e-2 (1.08e-4) -	1.2314e-2 (3.16e-5) +	1.8574e-2 (4.87e-5) =	**1.2314e-2 (1.35e-4)** +	1.8552e-2 (3.42e-5)
DTLZ7	3	22	1.3772e-1 (2.77e-3) -	2.4145e-1 (1.06e-1) -	1.3108e-1 (1.24e-3) -	2.0660e-1 (2.42e-3) -	**1.3061e-1 (1.23e-3)**
DelayEnergyEC	2	3	1.5699e+0 (7.09e-1) -	4.0218e-1 (3.62e-4) =	4.0202e-1 (5.22e-4) =	4.3721e-1 (1.24e-1) -	**4.0190e-1 (3.78e-4)**
+/-/=			1/10/2	3/8/2	1/4/8	2/10/1	

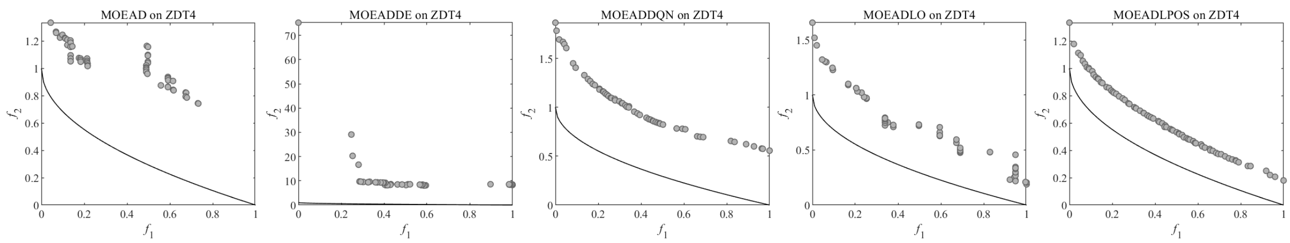

Fig. 1. Approximated PFs on ZDT4 instance

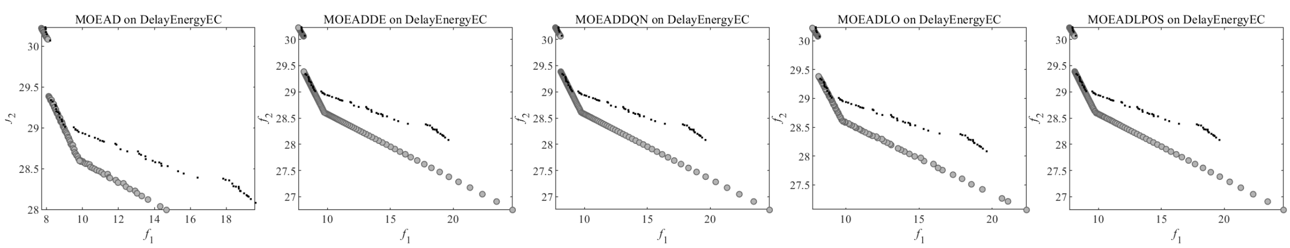

Fig. 2. Approximated PFs on DelayEnergyEC instance

A sample prompt format is shown below:

> **Example Prompt:**
> Now, you are assisting in selecting the most effective variation operator in a multi-objective optimization process.
> The current optimization problem is **DTLZ2**. The algorithm is at **generation 37**, with an overall progress of **45.3%**.
> You are given three candidate operators with usage and performance:
> **1 = SBX** (used: 18, performance: 0.47)
> **2 = M2M** (used: 12, performance: 0.51)
> **3 = DE1** (used: 9, performance: 0.44)
> Based on this information, select the most promising operator.
> Reply ONLY with the number (1, 2, or 3).

This prompt is sent to GPT-3.5-turbo via API, and the returned index is accepted if valid. Otherwise, the top-ranked candidate is chosen as fallback. The LLM's choice reflects learned knowledge of statistical patterns and language semantics, enabling it to consider performance, usage balance, and symbolic relevance.

By incorporating LLM-based reasoning, the framework introduces a flexible and interpretable semantic layer into operator selection, improving adaptability in dynamic search environments.

C. Adaptive Operator Selection via LLM-DQN Collaboration

To enable intelligent operator selection in multi-objective optimization, we propose MOEA/D-LPOS, a framework that integrates DQN and LLMs for context-aware decision making .

At each generation, a decision state $\mathbf{s}_t$ is encoded using individual and population-level descriptors: the current solution $\mathbf{x}_i$, weight vector $\boldsymbol{\lambda}_i$, objective vector $\mathbf{f}_i$, ideal point $\mathbf{z}^*$, diversity metrics, and operator usage history. The DQN outputs a binary gate $d_t \in \{0, 1\}$ indicating whether to consult the LLM.

If $d_t = 1$, a structured prompt is generated to summarize the current context, and the LLM returns a preference distribution $\mathbf{p}_{LLM}$ over candidate operators. The DQN also provides a softmax-normalized distribution $\mathbf{p}_{DQN}$. These are fused using a dynamic weighting coefficient:

$$p_{DQN}(a) = \frac{\exp(Q(s,a)/\tau)}{\sum_{a'} \exp(Q(s,a')/\tau)} \tag{6}$$

$$\mathbf{p}_{\text{final}} = \beta \cdot \mathbf{p}_{LLM} + (1-\beta) \cdot \mathbf{p}_{DQN} \tag{7}$$

$$\beta = \sigma(\text{conf}_{LLM} \cdot \text{cert}_{DQN}) \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, conf_{LLM} is the LLM's top confidence score, and cert_{DQN} reflects Q-value certainty.

If $d_t = 0$, operator selection defaults to ϵ-greedy based on Q-values. The selected operator generates an offspring, and a scalar reward is computed as:

$$r_t = \alpha \cdot \Delta HV_t + (1-\alpha) \cdot \frac{1}{1 + \text{comp\_cost}_t} \tag{9}$$

Both the experience tuple $\langle s_t, a_t, r_t, s_{t+1}, d_t \rangle$ and LLM interactions are stored to update the DQN and refine future prompts. The full framework is summarized in Algorithm 2.

Algorithm 2 MOEA/D-LPOS Framework

Input: Candidate operators OP, LLM model GPT
Initialize population P, weights W, DQN Q, buffer $\mathcal{T}$, pool $\mathcal{E}$
while termination not met **do**
  for each subproblem **do**
    Encode state $\mathbf{s}_t$
    $d_t \leftarrow$ DQN decision
    if $d_t = 1$ **then**
      Query LLM $\rightarrow \mathbf{p}_{LLM}$, compute $\mathbf{p}_{DQN}$, fuse to select operator
    else
      Select operator via ϵ-greedy
    end if
    Generate offspring, compute reward r_t
    Store $\langle s_t, a_t, r_t, s_{t+1}, d_t \rangle$ and LLM data
  end for
  Periodically update DQN and refine prompts
end while

III. EXPERIMENTS

A. DelayEnergyEC: Constrained Multi-objective Optimization Problem

We formulate the edge-cloud collaborative task offloading as a constrained multi-objective optimization problem termed *DelayEnergyEC*, which jointly minimizes task delay and energy consumption. Consider a user task of size D (MB) that can be executed locally, offloaded to MEC server i, or offloaded to remote cloud. The decision variables are task portions x_i, x_c (MB) and connection time T (s), subject to constraint $x_i + x_c \leq D$.

Objective Functions: The delay objective is defined as:

$$f_1(x) = \max\left(\frac{D - x_i - x_c}{V_l}, \frac{x_i}{V_{u\to i}} + \frac{x_i}{V_i}, \frac{x_c}{V_{u\to c}} + \frac{x_c}{V_c}\right) \tag{10}$$

where D is the total task size in MB, x_i and x_c represent the task portions offloaded to MEC server and cloud respectively, V_l denotes the local processing speed, $V_{u\to i}$ and $V_{u\to c}$ are the upload speeds to MEC server and cloud, and V_i and V_c are the processing speeds of MEC server and cloud respectively.

The energy consumption objective is:

$$f_2(x) = (D - x_i - x_c) \cdot P_l + (x_i + x_c) \cdot P_t + T_{\text{mec}} \cdot P_i + T_{\text{cloud}} \cdot P_c \tag{11}$$

where P_l is the local processing power consumption, P_t represents the transmission power for uploading tasks, T_{mec} and T_{cloud} are the connection times to MEC server and cloud respectively, and P_i and P_c denote the power consumption during MEC and cloud connections.

Optimization Model:

$$\begin{aligned} \min_{x_i, x_c, T} \; & F(x) = (f_1(x),\; f_2(x)) \\ \text{s.t.} \quad & x_i + x_c \le D \\ & f_1(x) \le T_d, \quad f_2(x) \le E_{\text{th}} \\ & 0 \le x_i, x_c \le D \\ & 0.01 \le T \le T_{\max} \end{aligned} \tag{12}$$

where T_d represents the maximum tolerable delay threshold, E_{th} is the energy consumption threshold, and $T_{\max}$ denotes the maximum allowable connection time. The constraints ensure that the total offloaded task portions do not exceed the original task size, the delay and energy consumption remain within acceptable limits, and all decision variables stay within their feasible ranges.

To verify the performance of the proposed MOEA/D-LPOS, comparisons were made with the most classical and similar algorithms in the experiments by PlatEMO [13], including MOEA/D [1], which is the most classical MOEA, MOEA/D-DE [5], a variant of MOEA/D, MOEA/D-LO [14], an algorithm that mimics the behavior of LLM, and MOEA/D-DQN [7], an algorithm that uses reinforcement learning to select operators alone. Experimental tests were conducted on ZDTs [15], DTLZs [16], and DelayEnergyEC. The framework of the proposed method integrates the advantages of LLM and optimization strategies, performing better than previously proposed methods that use LLM or reinforcement learning alone.

B. Parameter Settings

The settings for MOEA/D-based algorithms are as follows: The number of subproblems N: 100 for ZDT problems ($M = 2$) and 105 for DTLZ problems ($M = 3$). The number of weight vectors in the neighborhood T: $N/10$. The maximum number of evaluations differs across test suites: 50,000 for ZDT problems, 30,000 for DTLZ problems, and 10,000 for DelayEnergyEC problem, The data size of the DelayEnergyEC problem is 100 MB.

C. Comparison

The experimental results confirm the effectiveness of the proposed MOEA/D-LPOS across both benchmark problems (ZDT, DTLZ) and the real-world DelayEnergyEC task. Table I presents the HV comparison results, where MOEA/D-LPOS achieves the best performance on 9 out of 13 test cases, demonstrating superior convergence accuracy and diversity maintenance with higher HV values indicating better solution quality. Table II shows the IGD results, where MOEA/D-LPOS obtains the lowest IGD values on 8 out of 13 instances, confirming excellent convergence to the true Pareto front. Compared with variants that utilize LLM (MOEA/D-LO) or reinforcement learning (MOEA/D-DQN) alone, MOEA/D-LPOS exhibits more stable and robust performance, significantly outperforming competitors on the majority of test cases as indicated by the statistical significance tests. Figure 1 illustrates the approximated Pareto fronts on the challenging ZDT4 instance, where MOEA/D-LPOS produces a more complete and well-distributed front compared to other algorithms. Figure 2 demonstrates the practical applicability on the real-world DelayEnergyEC problem, showing that MOEA/D-LPOS successfully identifies superior trade-off solutions between delay and energy consumption objectives. These results highlight the practical potential of MOEA/D-LPOS in solving complex multi-objective optimization problems effectively.

IV. Conclusions

In this work, we proposed MOEAD-LPOS, a dual-model evolutionary framework that integrates DQN and LLMs to enhance operator selection in constrained multi-objective optimization. Unlike existing approaches that treat LLMs as direct variation operators, our method utilizes LLMs as high-level decision consultants, providing semantic guidance that complements the feedback-driven learning capability of DQN. This collaborative design enables dynamic and context-aware operator selection, effectively improving the balance between global exploration and local exploitation.Extensive experiments conducted on classical benchmark problems and a real-world edge-cloud task offloading scenario demonstrate that MOEAD-LPOS achieves superior convergence, diversity, and constraint-handling performance compared to state-of-the-art methods. These results confirm the robustness and generalization ability of our approach across diverse problem domains.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2023YFE0208800), the Natural Science Foundation of Shandong Province (No. ZR2022MF346), and the Key Laboratory of Computing Power Network and Information Security, Ministry of Education under Grant No. 2024PY009.

References

[1] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[2] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2008.

[3] B. Zhang, X. Gong, J. Wang, F. Tang, K. Zhang, and W. Wu, "Nonstationary fuzzy neural network based on FCMnet clustering and a modified CG method with Armijo-type rule," *Information Sciences*, vol. 608, pp. 313–338, 2022.
[4] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.
[5] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2010.
[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
[7] Y. Tian, X. Li, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1051–1064, 2022.
[8] J. Wang, Q. Chang, T. Gao, K. Zhang, and N. R. Pal, "Sensitivity analysis of takagi–sugeno fuzzy neural network," *Information Sciences*, vol. 582, pp. 725–749, 2022.
[9] M. E. Aydın and R. Durgut, "Efficient and adaptive operator selection in swarm intelligence using machine learning approaches," *Artificial Intelligence Theory and Applications*, vol. 4, no. 1, pp. 22–32, 2024.
[10] S. Ugare, T. Suresh, H. Kang, S. Misailovic, and G. Singh, "Improving llm code generation with grammar augmentation," *CoRR*, 2024.
[11] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of llm agents: A survey," *arXiv preprint arXiv:2402.02716*, 2024.
[12] Z. Zheng, Z. Xie, Z. Wang, and B. Hooi, "Monte carlo tree search for comprehensive exploration in llm-based automatic heuristic design," *arXiv preprint arXiv:2501.08603*, 2025.
[13] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
[14] F. Liu, X. Lin, S. Yao, Z. Wang, X. Tong, M. Yuan, and Q. Zhang, "Large language model for multiobjective evolutionary optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2025, pp. 178–191.
[15] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 105–145.