Smart Computation Offloading for Mobile Edge Computing: A Large Language Model Approach

Xuesong Han[1], Wentao Zhou[2], Inkyu Lee[1]
[1]Korea University, [2]Huawei Singapore Research Center
xuesonghan@korea.ac.kr, wtzhou@ieee.org, inkyu@korea.ac.kr

***Abstract*—Mobile Edge Computing (MEC) is a key technology for low-latency applications, but computation offloading remains a challenging non-convex optimization problem. Traditional methods like meta-heuristics often fall into local optima, while Deep Reinforcement Learning (DRL) suffers from high training costs and poor adaptability. This paper proposes PSLO, a novel Large Language Model (LLM) optimizer based on Particle Swarm Optimization (PSO). By integrating the swarm intelligence of PSO into the reasoning capabilities of LLMs, PSLO effectively solves the offloading decision problem. Simulation results demonstrate that PSLO outperforms standard PSO, genetic algorithms, and ant colony baselines in terms of convergence speed and cost minimization.**

I. INTRODUCTION

The growth of computation-intensive applications, such as augmented reality (AR) and autonomous driving, places immense pressure on the limited battery and processing capacities of mobile devices. Mobile Edge Computing (MEC) addresses this by enabling users to offload tasks to nearby servers. The core challenge in MEC is the joint optimization of offloading decisions and resource allocation to minimize system latency and energy consumption [1].

This problem is formulated as a Mixed-Integer Non-Linear Program (MINLP). Conventional convex optimization approaches require relaxation of binary variables, leading to suboptimal solutions. Meta-heuristic algorithms like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) avoid gradients but are prone to premature convergence in high-dimensional spaces due to their stochastic nature. Recently, Deep Reinforcement Learning (DRL) [2] has shown promise but faces significant hurdles, including high training overheads, the cold-start problem, and difficulty in generalizing across varying network sizes.

Large Language Models (LLMs) have demonstrated emergent capabilities in reasoning and optimization [3]. However, most existing LLM-based optimizers often lack collaborative mechanisms [4]. This paper introduces PSLO, which integrates the cooperative evolution of PSO with the semantic reasoning of LLMs. Unlike DRL, PSLO requires no training and adapts to different settings via prompt modification.

II. SYSTEM MODEL

We consider an MEC system with N users and one MEC server. Each user has an indivisible computation task, which can be executed either locally or offloaded to the MEC server. The offloading decision is denoted by $\alpha_n \in \{0, 1\}$, where 0 denotes local execution and 1 denotes MEC computing. The objective is to minimize a weighted sum of execution delay and energy consumption [2]:

$$\min_{\mathcal{A},f} \sum_{n=1}^{N} \left[\alpha_n C_n^o + (1-\alpha_n) C_n^l\right] \tag{1}$$

where C_n^o and C_n^l are the costs for offloading and local execution, respectively. The problem is subject to resource constraints: $\sum_{n=1}^{N} \alpha_n f_n \leq F$, where F is the MEC server's capacity and f_n is the allocated frequency.

III. PROPOSED METHOD: PSLO

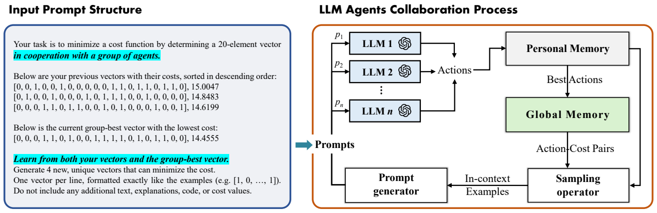
We propose the Particle Swarm LLM Optimizer (PSLO), an LLM-powered optimizer designed to solve for the discrete offloading variable $\mathcal{A} = [\alpha_1, \ldots, \alpha_N]$. With $\mathcal{A}$ determined by PSLO, the remaining problem of continuous resource allocation f is transformed into a convex form and solved.

A. Semantic Mapping of PSO Components

Traditional PSO updates a particle's position using inertia, cognition, and social influence. PSLO maps these components into a language context, enabling LLMs to reason about the search direction rather than merely computing it. The mapping strategy is summarized in Table I.

TABLE I
SEMANTIC MAPPING OF PSO COMPONENTS TO LLM PROMPTS

PSO Component	LLM Mechanism
Inertia	The LLM observes historical patterns from previous solutions to maintain momentum.
Cognition	The LLM identifies personal strategies from its memory of past solutions.
Society	The LLM analyzes the global best solution to incorporate swarm knowledge.
Stochasticity	The LLM generates multiple solutions, where multiple outputs promote diversity.



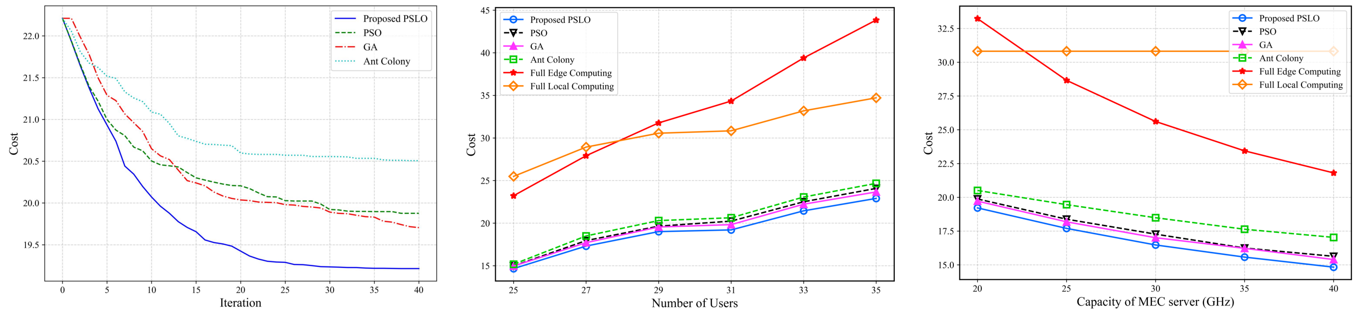
Your task is to minimize a cost function by determining a 20-element vector ***in cooperation with a group of agents.***

Below are your previous vectors with their costs, sorted in descending order:
[0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0], 15.0047
[0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0], 14.8483
[0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1], 14.6199

Below is the current group-best vector with the lowest cost:
[0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0], 14.4555

Learn from both your vectors and the group-best vector.
Generate 4 new, unique vectors that can minimize the cost.
One vector per line, formatted exactly like the examples (e.g. [1, 0, …, 1]).
Do not include any additional text, explanations, code, or cost values.

Fig. 1. Overall framework of the proposed PSLO method.



(a)

(b)

(c)

Fig. 2. (a) Convergence performance with 30 users, (b) scalability with varying numbers of users, and (c) scalability with varying MEC capacity.

B. Framework of the PSLO Method

The PSLO operates with a swarm of LLM agents, as illustrated in Fig. 1. The iterative process is as follows:

1) *Initialization:* A population of random actions is generated. The convex solver evaluates the cost of each.

2) *Memory Update:* The resulting action–cost pairs are stored in the personal memory of each agent. Subsequently, the action with the minimum cost is identified and used to update the global memory, ensuring that all agents share the current swarm-best solution.

3) *Sampling Operation:* A sampling operator retrieves and ranks the action–cost pairs from both personal and global memories to construct the input context for each agent.

4) *Prompt Construction:* A structured prompt is dynamically generated. It integrates the task description for cost minimization, the agent's personal experience, and the collective social experience. The prompt explicitly instructs the LLM to learn from the divergence between its history and the swarm-best to generate new and diverse actions.

5) *LLM-Based Inference:* Given the constructed prompt, the LLM performs semantic reasoning rather than linear combination. For instance, when the swarm-best suggests offloading for specific users while the agent's historical experience indicates otherwise, the LLM infers an improved strategy by reasoning over the cost difference.

6) *Evaluation:* The newly generated actions are evaluated by the solver, and Steps 2)–6) are iteratively repeated until the maximum number of iterations is reached.

IV. SIMULATION RESULTS

We evaluate PSLO in a multi-user single-server MEC environment where the server and user CPU frequencies are set to 20 GHz and 1 GHz, respectively. Using gpt-3.5-turbo with 4 agents, we compare PSLO against PSO, GA, Ant Colony, and fixed offloading schemes.

A. Convergence Performance

Fig. 2(a) illustrates the convergence behavior of different optimization algorithms with 30 users. It can be observed that PSLO achieves a rapid reduction during the early iterations and converges significantly faster than the baselines. In terms of solution quality, PSLO attains the lowest cost among all compared methods. This improvement can be attributed to the collaborative reasoning capability of the LLM-based agents, which enhances global exploration and mitigates the risk of being trapped in local optima.

B. Scalability Performance

To evaluate scalability, we investigate system performance under varying numbers of users and different MEC server capacities. In Fig. 2(b), the number of users increases from 25 to 35, resulting in a rapidly expanding search space. The system cost of all methods increases with the number of users; however, PSLO consistently maintains the lowest cost growth rate compared to the baselines.

Fig. 2(c) shows the impact of MEC server capacity. As server capacity increases from 20 GHz to 40 GHz, all methods benefit from improved computational resources, leading to reduced costs. Notably, PSLO consistently outperforms the baselines across all capacity settings. These results indicate that PSLO effectively adapts to changes in system scale and resource availability, demonstrating strong scalability and robustness in large-scale MEC environments.

V. CONCLUSION

This paper proposes PSLO, an LLM-based approach for MEC computation offloading. By integrating PSO into a semantic framework, we enable LLMs to act as intelligent, cooperative agents. PSLO achieves faster convergence and lower costs than traditional methods without the need for extensive training. Future work will explore deploying distilled Edge-LLMs to reduce inference latency.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (No. RS-2022-NR070834).

REFERENCES

[1] S. Dong, J. Tang, K. Abbas, R. Hou, J. Kamruzzaman, L. Rutkowski, and R. Buyya, "Task offloading strategies for mobile edge computing: A survey," Comput. Netw., vol. 254, p. 110791, 2024.

[2] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), 2018, pp. 1–6.

[3] H. Lee, W. Zhou, M. Debbah, and I. Lee, "On the convergence of large language model optimizer for black-box network management," IEEE Trans. Commun., vol. 73, no. 11, pp. 11385–11402, Nov. 2025.

[4] H. Lee, M. Kim, S. Baek, W. Zhou, N. Lee, M. Debbah, and I. Lee, "Large language models for knowledge-free network management: Feasibility study and opportunities," IEEE Access, vol. 13, pp. 187092–187106, 2025