Intelligent Coordinated Scheduling of Mobile Energy Storage Systems for Power Grid Operation Driven by Large Language Models

Jiawei Wang, *Graduate Student Member, IEEE,* Haotian Zhao, *Member, IEEE,* Qinglai Guo, *Fellow, IEEE,* and Hongbin Sun, *Fellow, IEEE*

***Abstract*—As an emerging source of grid flexibility, mobile energy storage systems (MESSs) can perform diverse tasks required by the power grid under various scenarios. However, the scheduling of MESSs exhibits strong spatial-temporal coupling and energy-mobility coupling, leading to high decision complexity. Fully leveraging the potential of MESSs in such dynamic environments requires enhanced adaptability in scheduling strategies. Traditional mathematical programming and data-driven methods often struggle to generalize efficiently across diverse scenarios and tasks, forcing MESS operators to rely heavily on human expertise and manual intervention. To address these challenges, this paper proposes an intelligent collaborative decision framework powered by the Large Language Model (LLM). The framework orchestrates a multi-stage process including instruction translation, problem configuration, and code execution while embedding a spatiotemporal decision model of MESS as domain-specific knowledge. This integration combines the respective strengths of LLMs and mathematical optimization. Furthermore, a benchmark dataset covering diverse scenarios and multi-objective tasks, along with corresponding evaluation metrics, is constructed. Numerical experiments compare multiple MESS decision frameworks to show the effectiveness of the proposed LLM-driven intelligent decision approach and validate the potential of LLMs for industrial decision optimization.**

***Index Terms*—large language model, mobile energy storage system, coordinated scheduling, intelligent decision-making.**

I. Introduction

Recently, the large-scale integration of distributed renewable energy, the differences of user-side load characteristics, and the increasing complexity of grid operations have posed serious challenges to the secure and economic operation of power systems. Against this backdrop, mobile energy storage systems (MESSs) have emerged as a new source of flexibility. With their unique spatiotemporal regulation and responsive capabilities, MESSs are becoming a key tool for enhancing grid optimization.

Under normal conditions, MESS can support economic and low-carbon operation. For example, [1] formulated a multi-objective optimization model in which MESS participates in energy arbitrage and ancillary services to maximize profit, while [2] demonstrated how MESS improves overall system economics and renewable energy utilization. Under risk scenarios, where the grid faces overload, congestion, or extreme weather events, MESS can enhance resilience and mitigate system risks [3]. For instance, [4] optimized MESS charging and discharging strategies to support voltage regulation in distribution networks, and [5] deployed MESS fleets for outage restoration, improving recovery efficiency of critical loads after extreme weather events.

Coordinated scheduling between MESSs and the power grid involves substantial decision complexity. On one hand, MESSs have higher degrees of freedom, requiring joint optimization of charging/discharging over time and movement across locations, leading to strong spatial-temporal and energy-mobility coupling [6]. On the other hand, their operating environments, task objectives, and system conditions are often dynamic and diverse. To unlock their full potential across multi-scenario and multi-task contexts, MESS scheduling strategies must be capable of interpreting complex instructions and adapting autonomously.

Moreover, MESS operators, such as distribution aggregators, virtual power plant managers, or storage service providers, represent new decision-making entities in grid operation. The inherent complexity demands strong expertise in optimization, modeling, and programming, often necessitating costly human labor and ongoing expert support. Consequently, MESS owners require an intelligent decision framework acting as a copilot that is capable of handling diverse and complex operational demands, autonomously generating scheduling strategies, and significantly reducing decision-making complexity.

Currently, conventional MESS scheduling methods primarily rely on precise mathematical programming, formulated and solved using optimization modeling languages [7], [8]. Although these methods are mature and reliable, their fixed formulations can only handle predefined scenarios and tasks, making it hard for them to handle dynamic and adaptive decision environments. For MESSs, characterized by diverse functions and decision spaces, such rigidity limits their full potential. On the other hand, data-driven approaches represented by deep reinforcement learning (DRL) have shown promise in solving MESS decision problems [9], [10]. However, due to architectural and scalability constraints, DRL struggles to generalize across multiple scenarios, tasks, and heterogeneous objectives. Its training relies on carefully designed scalar reward functions, which restricts the adaptability of trained agents when faced with new or complex situations.

Recently, the rise of large language models (LLMs) has opened new avenues for flexible and intelligent decision-making. With strong capabilities in natural language understanding, logical reasoning, and contextual learning, LLMs provide a promising foundation for building intelligent decision frameworks [11]. Moreover, LLMs offer MESS operators a natural and user-friendly interface, lowering the barrier to

J. Wang, H. Zhao, Q. Guo, and H. Sun are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China.

This work was supported by the National Key R&D Program of China (No. 2022YFB2404000) and the National Natural Science Foundation of China (NSFC) under Grant (No. 52321004).

979-8-3315-8138-1

deploying optimization-based decision tools. Although several studies have explored integrating LLMs into power system applications, such as load forecasting, document analysis, and knowledge-based query tasks [12], frameworks capable of handling complex scheduling decisions are still limited. General-purpose LLMs often lack the domain expertise and reliability needed for such tasks. Therefore, effective LLM-driven intelligent decision-making must integrate LLM capabilities with domain-specific knowledge to overcome the limitations of traditional methods and fully realize the potential of LLMs.

Therefore, this paper does not introduce physical improvements to the MESS spatiotemporal scheduling model. Its main purpose is to explore an intelligent decision-making framework in which an LLM serves as an optimization orchestrator. In this framework, the MESS model is invoked as expert-level domain knowledge, demonstrating the advantages of combining the perceptive understanding capabilities of LLMs with the optimization power of mathematical programming.

Based on the above analysis, the main contributions of this study are as follows:

1. An LLM-driven intelligent decision framework is proposed for coordinated scheduling of MESSs, enabling automated generation of scheduling plans from high-level commands and intents under dynamic decision environments.

2. A multi-stage orchestration process, including instruction translation, problem configuration, and code execution, is designed through prompt engineering, embedding MESS spatiotemporal decision models as domain-specific knowledge. The LLM acts as an orchestrator rather than a direct policy generator, combining the interpretive reasoning ability of LLMs with the precision of mathematical optimization.

3. A benchmark dataset for coordinated MESS scheduling is constructed, encompassing diverse scenario conditions and task types, to evaluate and compare the performance of different decision frameworks under flexible and variable environments.

II. Problem Description and Modeling

This paper focuses on a mixed integer linear programming (MILP) formulation for the coordinated optimization of the MESS fleet and the power grid. The model can also serve as useful knowledge and experience of experts in the field of MESS coordinated optimization [7].

A. Spatiotemporal Decision Model of Mobile Energy Storage Systems

The characteristics of MESSs include mobility and energy aspects, which correspond to discrete and continuous decisions, respectively.

The movement of MESS is described by a set of binary variables. For a MESS unit $v \in \mathcal{V}$ at time $t \in \mathcal{T}$: $\gamma_{v,n,m,t}$ indicates whether it is traveling from node n to node m. $\omega_{v,n,t}$ indicates whether it stays at node n. $\alpha_{v,n,t}$ indicates whether it arrives at node n. $\beta_{v,n,t}$ indicates whether it leaves node n. $\theta_{v,n,t}$ is an auxiliary variable used to ensure consistency between arrival and travel states. Here, $\mathcal{N}$ represents the set of nodes.

$$\sum_{n\in\mathcal{N}} \omega_{v,n,t} \leq 1 - \sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{N}} \gamma_{v,n,m,t},\ \forall v \in \mathcal{V}, \forall t \in \mathcal{T} \tag{1a}$$

$$\begin{aligned}\alpha_{v,n,h} - \beta_{v,n,h} =& \omega_{v,n,h} - \omega_{v,n,t-1},\\ &\forall v \in \mathcal{V}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}\end{aligned} \tag{1b}$$

$$\sum_{n\in\mathcal{N}} (\alpha_{v,n,t} + \beta_{v,n,t}) \leq 1,\ \forall v \in \mathcal{V}, \forall t \in \mathcal{T} \tag{1c}$$

$$\sum_{m\in\mathcal{N}} \gamma_{v,n,m,t} \geq \beta_{v,n,t},\ \forall v \in \mathcal{V}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{1d}$$

$$\begin{aligned}\alpha_{v,m,t} - \theta_{v,m,h} = \sum_{n\in\mathcal{N}} (\gamma_{v,n,m,t-1} - \gamma_{v,n,m,t}),\\ \forall v \in \mathcal{V}, \forall m \in \mathcal{N}, \forall t \in \mathcal{T}\end{aligned} \tag{1e}$$

$$\sum_{n\in\mathcal{N}} (\alpha_{v,n,t} + \theta_{v,n,t}) \leq 1,\ \forall v \in \mathcal{V}, \forall t \in \mathcal{T} \tag{1f}$$

$$\begin{aligned}\gamma_{v,n,m,t} \geq& \gamma_{v,n,m,t-1} - \gamma_{v,n,m,t-T_{n,m,t}},\\ &\forall v \in \mathcal{V}, \forall n, m \in \mathcal{N}, \forall t \in \mathcal{T}\end{aligned} \tag{1g}$$

Constraint (1a) requires that each MESS unit can only be in one state at any given time. Constraint (1b) describes how departure and arrival behaviors cause changes in the staying state. Constraint (1c) ensures that no concurrent departure or arrival occurs within the same time period. Constraint (1d) defines that a departure triggers the start of a traveling state. Constraints (1e) and (1f) describe that the end of a traveling state leads to an arrival. Constraint (1g) specifies that traveling from node n to node m requires $T_{n,m,t}$ time periods.

When staying at a node, a MESS unit can exchange power with the grid, reflecting its energy characteristics, which are represented by a set of continuous variables. For a MESS unit $v \in \mathcal{V}$ at time $t \in \mathcal{T}$: $E_{v,t}$ denotes the energy level. $P^{\mathrm{CHA}}_{v,n,t}$ and $P^{\mathrm{DIS}}_{v,n,t}$ denote the charging and discharging power at node n. E^{MAX}_{v} is the energy storage capacity. $P^{\mathrm{CHA,MAX}}_{v}$ and $P^{\mathrm{DIS,MAX}}_{v}$ are the maximum charging and discharging power. ρ represents the self-discharge rate. η^{CHA} and η^{DIS} represent charging and discharging efficiency. Δt is the length of each time period.

$$0 \leq E_{v,t} \leq E^{\mathrm{MAX}}_{v},\ \forall v \in \mathcal{V}, \forall t \in \mathcal{T} \tag{2a}$$

$$\begin{aligned}E_{v,t} = (1-\rho)\, E_{v,t-1} + \sum_{n\in\mathcal{N}} \left(P^{\mathrm{CHA}}_{v,n,t} \cdot \eta^{\mathrm{CHA}} \cdot \Delta t - \frac{P^{\mathrm{DIS}}_{v,n,t} \cdot \Delta t}{\eta^{\mathrm{DIS}}} \right),\\ \forall v \in \mathcal{V}, \forall t \in \mathcal{T} \setminus \{0\}\end{aligned} \tag{2b}$$

$$0 \leq P^{\mathrm{DIS}}_{v,n,t} \leq \omega_{v,n,t} \cdot P^{\mathrm{DIS,MAX}}_{v},\ \forall v \in \mathcal{V}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2c}$$

$$0 \leq P^{\mathrm{CHA}}_{v,n,t} \leq \omega_{v,n,t} \cdot P^{\mathrm{CHA,MAX}}_{v},\ \forall v \in \mathcal{V}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2d}$$

$$P^{\mathrm{DIS}}_{v,n,t} \cdot P^{\mathrm{CHA}}_{v,n,t} = 0,\ \forall v \in \mathcal{V}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{2e}$$

Constraint (2a) sets the upper and lower limits of energy level. Constraint (2b) represents the change in energy level caused by charging and discharging. Constraints (2c) and (2d) require that MESS units can charge or discharge only when located at a specific node. Constraint (2e) ensures that charging and discharging cannot occur simultaneously.

Equations (1) and (2) jointly characterize the overall behavior of the MESS fleet and serve as the core mechanism for optimization decisions.

B. Coordinated Scheduling for Grid Operation Supported by Mobile Energy Storage Systems

The power grid operation is represented by the linearized power flow model. Actions of MESS units cause power

979-8-3315-8138-1

variations at network nodes, supporting the grid in achieving economic, secure, and low-carbon objectives. These objectives are guided by the optimization problem's objective function. Different scenarios and tasks lead to different formulations of the objective function, as shown in equation (3).

$$\min \begin{cases} f_1 = \sum_{t\in\mathcal{T}} \sum_{g} C_g(p_{g,t}) \\ f_2 = \sum_{t\in\mathcal{T}} \sum_{v} C_v(\gamma_{v,n,m,t}; P^{\text{CHA}}_{v,n,t}; P^{\text{DIS}}_{v,n,t}) \\ f_3 = \sum_{t\in\mathcal{T}} \sum_{b} C_o(p^o_{b,t}),\ f_4 = \sum_{t\in\mathcal{T}} \sum_{n} C_s(p^s_{n,t})\ \cdots \end{cases} \tag{3}$$

Under normal operating conditions, optimization primarily focuses on economic efficiency. For example, f_1 represents the total generation cost associated with the output $p_{g,t}$ of each generator g. f_2 represents the dispatch cost of the MESS fleet, related to both movement and charging/discharging actions. When the system faces certain security risks, such as excessive load growth or line overload, mobile energy storage can help mitigate these risks. For instance, f_3 denotes the total line overload penalty cost, depending on the overload amount $p^o_{b,t}$ of each line b. In extreme situations, where many lines are interrupted due to extreme weather, mobile energy storage can support emergency power supply and load recovery. For example, f_4 represents the total load-shedding penalty cost, related to the shed load $p^s_{n,t}$ at each node n.

The operation mode of the MESS fleet must adapt to different scenarios and tasks. During normal conditions, safety should not be compromised for cost reduction, while in the presence of severe security risks, economic concerns should be temporarily set aside. The importance and priority of objectives are always adjusted according to the decision context.

III. LLM-Driven Coordinated Scheduling of Mobile Energy Storage Systems

To address dynamic and flexible decision environments, this paper proposes an LLM-driven optimization orchestration framework with a multi-stage process. Based on a fundamental spatiotemporal decision model for MESS, the framework consists of three stages: instruction translation, problem configuration, and code execution.

A. Instruction Translation

The operator of a MESS fleet may impose diverse and customized operational requirements. These instructions are often flexible and highly varied. At this stage, the LLM performs instruction translation, extracting and organizing key information from high-level commands provided by the decision-maker, such as the intended objectives, available resources, and the configuration or relaxation of constraints.

To ensure reliability, a prompt template is designed with role descriptions, task definitions, format specifications, few-shot examples, and chain-of-thought (CoT) guidance [13]. The LLM is required to output results in a unified and structured JSON format [14].

Professional intent understanding and format alignment require substantial domain knowledge in power grids, mobile energy storage, and optimization, where general-purpose language models may exhibit knowledge gaps. This can lead to inaccurate outputs or format mismatches when dealing with complex operating rules or special market information. Beyond the prompt engineering adopted in this paper, retrieval-augmented generation (RAG) can be further integrated by designing an experience memory pool to retrieve similar historical dispatch cases, which enables the LLM to more accurately interpret concepts such as security and economy and to correctly orchestrate optimization problems based on this understanding. In addition, fine-tuning techniques can improve LLM performance in specific vertical domains. For example, [15] applies lightweight fine-tuning to help LLMs accurately understand dispatch tasks and format alignment in power distribution networks.

B. Problem Configuration

Based on the structured information extracted from the instructions, this stage updates the base model to generate a customized model suited to the current decision environment. The following examples illustrate how such updates are implemented.

For the task objective, since the instruction translation stage identifies the current decision focus, it becomes the basis for configuring the objective function and constraints. For example, if the LLM detects that the system is in a normal operating state with the goal of minimizing total operating cost, the corresponding objective and additional constraints are configured as shown in equation (4). In this case, the model minimizes overall cost while satisfying load supply and security constraints.

$$\begin{aligned} &\min\ f_1 + f_2 \\ &\quad = \sum_{t\in\mathcal{T}} \sum_{g} C_g(p_{g,t}) + \sum_{t\in\mathcal{T}} \sum_{v} C_v(\gamma_{v,n,m,t}; P^{\text{CHA}}_{v,n,t}; P^{\text{DIS}}_{v,n,t}) \\ &\text{s.t. } \forall p^o_{b,t} = 0, \forall p^s_{n,t} = 0 \\ &\qquad \text{Other Constraints in Basic Model} \end{aligned} \tag{4}$$

In another example, if the LLM identifies a security-risk scenario, where the goal is to mitigate line overloads, the configuration changes accordingly as shown in equation (5). In such cases, high load levels and line congestion risks require dispatching MESS units to enhance system security.

$$\begin{aligned} &\min\ f_3 = \sum_{t\in\mathcal{T}} \sum_{b} C_o(p^o_{b,t}) \\ &\text{s.t. } \forall p^s_{n,t} = 0 \\ &\qquad \text{Other Constraints in Basic Model} \end{aligned} \tag{5}$$

For additional constraints, external events can be incorporated into the decision model. For instance, if the LLM detects from the instruction that MESS unit v_1 is unavailable due to maintenance, the model adds a constraint as shown in equation (6). Under this condition, v_1 performs no charging or discharging actions, and equation (1) ensures that it remains stationary at its initial node n^{ini} without any further modification.

$$\begin{aligned} &\forall P^{\text{CHA}}_{v_1,n,t} = 0,\ \forall P^{\text{DIS}}_{v_1,n,t} = 0 \\ &\omega_{v_1,n,t} = 0,\ \forall n \neq n^{\text{ini}} \end{aligned} \tag{6}$$

With the capabilities of the LLM, any diverse requirements or instructions can be implemented through customized updates built upon the base model.

979-8-3315-8138-1

C. Code Execution

The code execution environment is standardized to Gurobipy. As a mature and widely used optimization modeling tool, Gurobipy and its Python syntax are well embedded in modern LLM pretraining. With clearly designed prompts, the generated code aligns with the model's strengths. Moreover, to improve reliability, many intermediate expressions (e.g., f_1, f_2, f_3, f_4) are precomputed as "gurobipy.LinExpr" objects and bound to the model, allowing direct use during updates.

Therefore, the model updates described in Section III-B can be conveniently implemented using the statements "model.setObjective()" and "model.addConstr()". After invoking "model.update()" on the modified model, a customized optimization problem aligned with the current decision environment is obtained. The solver then automatically produces the optimized MESS scheduling result.

Since the configuration of optimization problems by the LLM is an incremental process conducted under predefined rules and guidance, the impact of output instability on decision-making is reduced. If the configuration generated by the LLM contains syntactic errors or cannot be successfully solved by the solver, the current code and error messages can be fed back to the LLM for correction, or the process can revert to a standard fixed optimization routine, thereby ensuring that the decision process remains uninterrupted.

IV. Numerical Experiments

The codes and workflows used in the numerical experiments are available in [13].

A. Evaluation Datasets and Metrics

The experiment is based on a modified IEEE 14-bus power system that includes multiple generators, load nodes, and a fleet composed of several MESS units.

In each test case, load levels are randomly perturbed, line outages are randomly introduced, the initial positions and availability of MESS units are randomly determined, and task objectives are randomly selected. A natural-language description corresponding to each scenario-task combination is then generated as a high-level scheduling instruction. For each case, a manually written optimization program corresponding to that scenario-task combination is executed with sufficient solving time and treated as the reference "perfect optimizer," providing the label for the optimal strategy.

Two metrics are defined to evaluate the proposed method, as shown in (7).

$$F = \frac{N^{\text{Feasible}}}{N^{\text{Total}}} \times 100\%,\ S = \left| \frac{\text{Obj}^{\text{after}} - \text{Obj}^{\text{before}}}{\text{Obj}^{\text{perfect}} - \text{Obj}^{\text{before}}} \right| \times 100\% \tag{7}$$

The first metric is the feasibility rate F, which not only measures whether the method can generate a valid result but also checks whether the obtained strategy satisfies the specific requirements stated in the task instruction. For example, in a line-overload mitigation task, if the instruction specifies that no load shedding is allowed, a strategy that eliminates overloads at the cost of load loss is still considered infeasible.

The second metric is the task score S. Using the grid's original operation without any MESS participation as the baseline, the improvement achieved by incorporating the MESS fleet can be measured as a relative enhancement in the objective function. The performance achieved by the perfect optimizer after sufficient computation time is defined as 100%, allowing direct comparison of the quality of scheduling strategies generated by other methods.

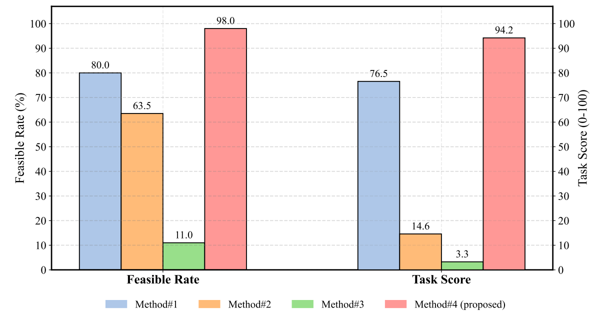

Fig. 1. Comparison of Metrics Across Different Methods.

B. Results and Comparisons

This paper evaluates and compares multiple methods for the coordinated scheduling of the MESS fleet.

Method#1: Fixed optimization code. A manually written, verified optimization program is used across all decision environments. All possible task objectives are incorporated into the objective function through weighted aggregation.

Method#2: Deep reinforcement learning. The MESS scheduling problem is formulated as a Markov decision process, where the reward function includes weighted terms for all task objectives. A reinforcement learning agent is trained using proximal policy optimization (PPO).

Method#3: LLM-Direct policy generation. MESS actions are encoded in text form, and the LLM directly generates textual scheduling actions after receiving task instructions and system information.

Method#4: LLM-driven optimization orchestrator. The LLM interprets the instruction, updates the base optimization model accordingly, and generates the final solution with an optimization solver.

All methods involving optimization use Gurobipy (12.0.2) as the tool, with the same solver settings. The PPO algorithm in Method#2 is implemented with stable-baselines3 (2.7.0). Method#3 and Method#4 both adopt "qwen3-next-80b-a3b-instruct" as the LLM, an open-source model with outstanding performance, featuring enhanced text understanding and logical reasoning capabilities [14]. The open-source nature of the model facilitates academic reproducibility while maintaining relatively low inference and deployment costs.

Figure 1 shows the overall performance of different methods on the benchmark dataset.

Figure 2 illustrates the distribution of task scores across individual test cases. Each dot represents the task score of a single test case (with slight jitter applied to avoid overlap). When a method produces an infeasible scheduling strategy, the task score is set to zero, significantly reducing its average performance.

Overall, Method#4 achieves the best results across both metrics, demonstrating that combining LLMs with optimization enables adaptive understanding of goals and constraints in diverse decision environments while leveraging the advantage of mathematical optimization tools.

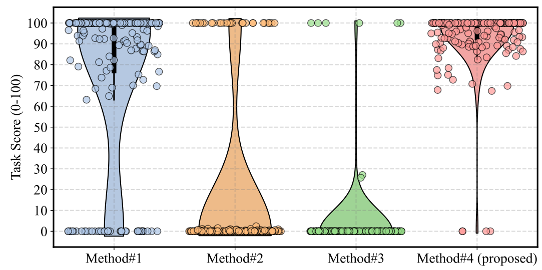

Fig. 2. Distribution of Task Scores Across Test Cases for Different Methods.

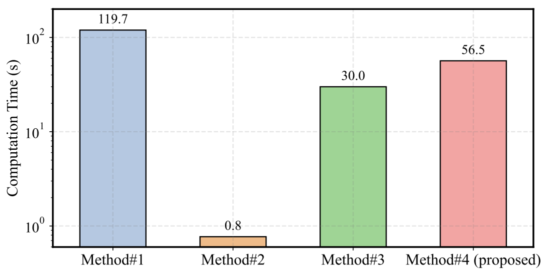

Fig. 3. Comparison of Computation Times Across Different Methods.

Method#1 ranks second. Although a fixed optimization program can sense physical system variations through structured data, it lacks adaptive adjustment capability. The use of relaxation terms, penalties, and weighted multi-objectives may lead to incomplete task fulfillment or infeasibility when additional constraints are not properly satisfied. This shows that fixed optimization programs are limited in dynamic decision environments.

Method#2 performs poorly. Although random load fluctuations and line outages are introduced in the training environment, the DRL agent trained under a fixed scalar reward struggles to generalize across varying scenarios and tasks.

Method#3 performs the worst, producing almost no feasible scheduling results. Even with prompt-engineering techniques and infeasible action masking logic, the directly generated strategies frequently violate basic constraints such as SOC limits. This indicates that general-purpose LLMs lack the domain knowledge and professional ability required to produce complex sequential decisions directly.

Figure 3 compares the computation time required for each method to produce scheduling results for a single test case. Method#2 generates results the fastest, as the trained DRL agent only needs a forward propagation process of small neural networks. Method#1 and Method#4 require longer computation time due to the inherent complexity of solving MILP problems. This also indicates that solver performance is the dominant factor affecting decision time, while the additional LLM inference and communication introduced in Method#4 is not dominant. At the day-ahead stage or on an hourly time scale, given the benefits in task flexibility and scalability, this overhead is acceptable. Despite not involving optimization, Method#3 also requires considerable time because LLMs must process long input prompts, generate large volumes of text, and parse outputs into structured schedules, consuming substantial tokens and processing time.

These findings suggest that integrating the strengths of LLMs and mathematical optimization enables more flexible and efficient intelligent decision-making.

V. Conclusion

As an emerging flexible resource, mobile energy storage systems exhibit complex spatiotemporal and energy-mobility coupling characteristics. The increasing diversity of scenarios and tasks places higher demands on the adaptive adjustment capability of scheduling strategies. Traditional mathematical optimization and data-driven approaches face challenges in rapidly adapting and generalizing to diverse decision environments, which may result in operators having to devote substantial effort and cost to rely on the knowledge and labor of human experts. This paper proposes an intelligent decision-making framework for coordinated scheduling of mobile energy storage systems to support power system operation, using the cognitive understanding capability of LLMs and the precision of mathematical programming to achieve adaptive and intelligent decision-making. Furthermore, a benchmark dataset encompassing diverse scenarios and task objectives, along with corresponding evaluation metrics, is constructed. Numerical experiments demonstrate the effectiveness of the proposed LLM-driven intelligent decision framework.

References

[1] X. Yan, C. Gao, and B. Francois, “Multi-objective optimization of a virtual power plant with mobile energy storage for a multi-stakeholders energy community,” *Appl. Energy*, vol. 386, p. 125553, May 2025.

[2] Z. Lu, X. Xu, Z. Yan, M. Shahidehpour, W. Sun, and D. Han, “Distributionally robust chance constrained optimization method for risk-based routing and scheduling of shared mobile energy storage system with variable renewable energy,” *IEEE Trans. Sustain. Energy*, vol. 15, no. 4, pp. 2594–2608, Oct. 2024.

[3] J. Wang, Q. Guo, B. Wang, and H. Sun, “Risk assessment of energy-transportation coupled system with unified integration of multi-type emergency decisions under extreme weather,” in *2024 IEEE 7th Student Conference on Electric Machines and Systems (SCEMS)*, Nov. 2024, pp. 1–6.

[4] Z. Zhang, Z. Y. Dong, and C. Yip, “When mobile energy meets active distribution networks: A security–economic coordination perspective,” *IEEE Trans. Smart Grid*, vol. 15, no. 3, pp. 3126–3140, May 2024.

[5] S. Yao, P. Wang, X. Liu, H. Zhang, and T. Zhao, “Rolling optimization of mobile energy storage fleets for resilient service restoration,” *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1030–1043, Mar. 2020.

[6] J. Wang, Q. Guo, H. Sun, and M. Chen, “Collaborative optimization of logistics and electricity for the mobile charging service system,” *Appl. Energy*, vol. 336, p. 120845, Apr. 2023.

[7] G. He, J. Michalek, S. Kar, Q. Chen, D. Zhang, and J. F. Whitacre, “Utility-scale portable energy storage systems,” *Joule*, vol. 5, no. 2, pp. 379–392, Feb. 2021.

[8] J. Wang, Y. Sheng, H. Ge *et al.*, “Charging pricing for autonomous mobility-on-demand fleets based on game theory,” *J. Mod. Power Syst. Clean Energy*, vol. 12, no. 6, pp. 2006–2018, Nov. 2024.

[9] Z. Ding, Z. Xu, J. Wang *et al.*, “Resilient load restoration with distributed mobile resources: A model-free approach,” *IEEE Trans. Ind. Appl.*, vol. 61, no. 4, pp. 5525–5536, Jul.-Aug. 2025.

[10] J. Wang, Q. Guo, and H. Sun, “Planning approach for integrating charging stations and renewable energy sources in low-carbon logistics delivery,” *Appl. Energy*, vol. 372, p. 123792, Oct. 2024.

[11] C. Huang, Z. Tang, S. Hu *et al.*, “ORLM: A customizable framework in training large models for automated optimization modeling,” *Oper. Res.*, pp. 1–24, May 2025.

[12] S. Majumder, L. Dong, F. Doudi *et al.*, “Exploring the capabilities and limitations of large language models in the electric energy sector,” *Joule*, vol. 8, no. 6, pp. 1544–1549, Jun. 2024.

[13] J. Wang, “Supplementary materials for intelligent coordinated scheduling of mobile energy storage systems for power grid operation driven by large language models,” Oct. 2025. [Online]. Available: https://github.com/DEPOT-JW/LLM-for-Mobile-Energy-Storage-System

[14] A. Yang, A. Li, B. Yang *et al.*, “Qwen3 technical report,” May 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[15] X. Yang, C. Lin, H. Liu, Q. Wang, Y. Yang, and W. Wu, “One request, multiple experts: Llm orchestrates domain specific models via adaptive task routing,” Nov. 2025. [Online]. Available: https://arxiv.org/abs/2511.12484

979-8-3315-8138-1