Automatic Load Allocation in Power Dispatch Master Stations Using Multi-Agent Deep Deterministic Policy Gradient Algorithms

Nan Zhou[1], Yizhen You[1], Yuheng Liu[1], Dongjian Gu[2]
[1]State Grid Jiangsu Electric Co. Ltd., Nanjing 211000, Jiangsu, China
[2]NARI-TECH Nanjing Control Systems Ltd., Nanjing 211106, Jiangsu, China
Corresponding author: Dongjian Gu (e-mail: gudongjian_11@163.com)

ABSTRACT Automatic load allocation in power dispatch master stations has become increasingly challenging due to the uncertainty of renewable generation and the complexity of multi-regional power coordination. This study proposes an automatic load allocation framework based on a Multi-Agent Deep Deterministic Policy Gradient (MA-DDPG) algorithm to improve real-time dispatch efficiency and system adaptability. A distributed multi-agent architecture is established in which each dispatch region operates as an autonomous agent with coordinated communication and decision-making capabilities. By incorporating stochastic models for wind and photovoltaic generation together with a multi-objective optimization strategy that simultaneously minimizes dispatch cost, renewable energy curtailment, and load fluctuation, the proposed approach achieves coordinated global optimization under dynamic operating conditions. Furthermore, adaptive communication weighting and prioritized experience replay are introduced to enhance cooperation efficiency and accelerate policy convergence. Experimental results demonstrate that the improved MA-DDPG framework significantly enhances training efficiency, reduces operational costs, and strengthens load balancing robustness while maintaining stable system performance under renewable and demand uncertainties. The proposed methodology also provides a reliable optimization paradigm for communication-intensive smart grids, where electromagnetic information transmission and distributed sensing infrastructures support real-time coordination and intelligent energy management.

INDEX TERMS multi-agent reinforcement learning, automatic load allocation, power dispatch, smart grid communication, renewable energy integration

I. INTRODUCTION

A. RESEARCH BACKGROUND

WITH the large-scale grid integration of renewable energy (wind and solar power) and the continuous expansion of power grid scale, traditional automatic load distribution in power dispatch master stations faces unprecedented challenges. These challenges are magnified by the necessity of integrating large, flexible loads from high-consumption sectors, particularly the industry, whose dynamic operational demands must be precisely managed for grid stability and successful energy transition. Conventional load distribution methods (such as proportional distribution and equal micro-increment rate) rely on precise mathematical models and deterministic parameters, making them ill-suited to accommodate the high uncertainty in renewable energy output and load demand [1-2]. For instance, when wind power output fluctuates by more than 20% within a short timeframe, the proportional allocation method often leads to excessive load deviations in individual dispatch areas, triggering equipment overload or power shortages. Simultaneously, traditional centralized dispatch models suffer from issues such as a single decision-making center, slow response times, and poor fault tolerance. This makes the entire load allocation system susceptible to paralysis when local dispatch areas experience failures [3-4].

Multi-agent systems (MAS) possess characteristics such as distributed decision-making, flexible coordination, and strong robustness, effectively addressing the shortcomings of traditional centralized dispatch. Each dispatch area is modeled as an independent agent, which formulates local load allocation decisions based on its operational status. Global optimization is achieved through information exchange among agents [5-6]. Deep reinforcement learning (DRL) demonstrates unique advantages in high-dimensional, nonlinear, and dynamic optimization problems. Among these, the deep deterministic policy gradient (DDPG) algorithm directly outputs deterministic control policies in continuous action spaces, aligning well with the “continuous adjustment requirements” of load allocation in power dispatch master stations [7-8]. However, the MA-DDPG algorithm—formed by combining MAS with DDPG—still exhibits limitations in power dispatch master station applications: First, agent-to-agent communication often employs fixed weights, preventing dynamic adjustment based on actual load allocation conditions. Second, the standard MA-DDPG algorithm’s experience replay mechanism uses random sampling, resulting in low utilization of critical samples (e.g., load surge scenarios). Third, the single Critic network tends to overestimate action values, leading to suboptimal load allocation strategies [9-10].

This research holds both theoretical and engineering significance: Theoretically, it establishes a multiagent load allocation framework based on MA-DDPG, filling the gap in distributed reinforcement learning applications for power dispatch master stations and enriching the theoretical system of intelligent dispatch. Engineering-wise, the proposed method significantly enhances the real-time performance and robustness of automatic load allocation, reduces dispatch costs and renewable energy curtailment rates, and provides technical support for the safe and economical operation of power grids. Critically, these improvements offer direct utility to high-consumption industrial users, enabling the industry to leverage its flexible load capacity for optimized coordination, resulting in substantial energy cost savings and a verifiable reduction in its carbon footprint.

B. RESEARCH STATUS AT HOME AND ABROAD

1) Load Allocation Techniques for Power Dispatch Master Stations

Overseas research on load allocation for power dispatch master stations began earlier. In 2018, the U.S. Department of Energy (DOE) proposed an IoT-based distributed dispatch framework enabling real-time sharing of load and supply information across different dispatch regions. However, load allocation still relied on traditional proportional methods, lacking intelligent decision-making capabilities [11]. The EU’s “SmartGrid” project employs Model Predictive Control (MPC) to optimize load allocation, partially addressing renewable energy uncertainties. However, its high computational complexity hinders real-time requirements for largescale grids [12]. In academia, researchers from UC Berkeley proposed a game-theoretic load allocation method treating inter-dispatch zone load distribution as a non-cooperative game. It achieves agent-level interest equilibrium via Nash equilibrium but suffers from slow algorithm convergence, rendering it unsuitable for emergency load regulation [13].

Domestic research focuses on refining traditional methods and applying single-agent reinforcement learning: The State Grid Electric Power Research Institute proposed an improved equal marginal rate method, incorporating renewable energy curtailment penalty factors to optimize load allocation strategies. However, this approach remains reliant on precise cost models and exhibits poor adaptability to uncertainty [14]. Tsinghua University applied standard DDPG algorithms to dispatch center load allocation, improving real-time performance over traditional methods. However, the single-agent framework cannot achieve distributed coordination across multiple dispatch regions, resulting in low decision efficiency in large-scale grids [15]. Southeast University developed a multi-agent load allocation system based on contract network protocols, enabling coordination through task auctions. Yet, high communication overhead caused load allocation delays exceeding 1 second, failing to meet real-time requirements [16].

2) Multi-Agent Deep Reinforcement Learning Techniques

Significant progress has been made in multi-agent deep reinforcement learning by international researchers: Google DeepMind introduced the MA-DDPG algorithm in 2017, extending single-agent DDPG to multi-agent scenarios by adding inter-agent communication modules, demonstrating remarkable effectiveness in gaming and robotic control tasks [17]; MIT improved MA-DDPG by introducing a centralized Critic network to evaluate global action values, enhancing agent coordination. However, the centralized Critic increases computational burden and is unsuitable for large-scale multi-agent systems [18].

Domestic MA-DDPG research primarily focuses on microgrid and energy internet scenarios: Zhejiang University applied MA-DDPG to microgrid load allocation, modeling each distributed generation (DG) unit as an agent to optimize load distribution through agent interactions. However, this approach disregarded renewable energy output uncertainty and was limited to small-scale applications [19]. North China Electric Power University proposed an MA-DDPG algorithm incorporating an attention mechanism to enhance agents’ ability to capture critical information during communication. However, it was tested only in simulation environments and lacks validation with actual power grid data [20].

C. TECHNICAL ROADMAP

Framework Design Phase (1-2 months): Determine scheduling region agent partitioning scheme → Design agent communication coordination mechanism → Define agent states, actions, and reward functions;

Algorithm Improvement Phase (2-3 months): Build standard MA-DDPG algorithm framework → Introduce adaptive communication weight module → Integrate priority experience replay mechanism → Design dualcritic network;

Simulation Platform Construction Phase (2 months): Collect actual operational data from dispatch control center (renewable energy output, load demand, equipment parameters) → Construct dispatch center simulation model using MATLAB/Simulink → Develop MA-DDPG algorithm model based on PyTorch;

Validation and Analysis Phase (1 month): Conduct simulation experiments under various scenarios (normal operation, load surges, renewable energy fluctuations) → Compare algorithm performance → Analyze improved algorithm robustness and adaptability → Finalize research outcomes.

II. MULTI-AGENT SYSTEM FRAMEWORK FOR POWER DISPATCH MASTER STATION

A. DIVISION OF DISPATCH AREA AGENTS

Based on administrative divisions and grid structure, a provincial-level grid dispatch master station is divided into N dispatch area agents (Agent1∼AgentN), with each agent corresponding to an independent regional grid. The division adheres to three principles: First, each dispatch area maintains relatively balanced supply and demand to minimize cross-regional transmission. Second, agents are geographically proximate to reduce communication latency. Third, the number of agents is controlled between 5 and 10 to balance decision efficiency and optimization effectiveness. Taking the Jiangxi provincial grid as an example, it is divided into 6 dispatch area agents. The coverage scope and key equipment parameters for each agent are shown in Table 1.

Each agent possesses independent decision-making capabilities, including load forecasting, renewable energy output prediction, and local load allocation. Simultaneously, agents achieve communication coordination through the dispatch master station communication network to realize global load optimization. The communication network employs 1000Mbps Ethernet, with communication latency between adjacent agents <50ms, meeting the real-time requirements for load allocation.

B. DEFINITION OF AGENT STATE, ACTION, AND REWARD FUNCTION

1) Agent State Space

Each agent's state space comprises local operational parameters and global coordination information, comprehensively reflecting the load allocation environment to inform decision-making. The state vector $s_i(t)$ of Agent i at time t is defined as:

$$s_i(t) = \big[P_{W,i}(t), P_{PV,i}(t), P_{L,i}(t), SOC_{ESS,i}(t), \\ P_{T,i}(t), \Delta P_i(t), \bar{P}(t)\big].$$

where:

$P_{W,i}(t)$: Actual wind power output (MW) of Agent i at time t, measured in real-time by the SCADA system;

$P_{PV,i}(t)$: Actual photovoltaic power output (MW) of Agent i at time t, obtained through real-time measurement;

$P_{L,i}(t)$: Actual load demand (MW) of Agent i at time t, derived from the short-term load forecasting system, with a prediction error of $\pm 5\%$;

$SOC_{ESS,i}(t)$: State of charge of Agent i's energy storage system at time t, ranging from 20% to 80%;

$P_{T,i}(t)$: Power transfer between Agent i and neighboring agents at time t (MW), where positive values indicate inflow and negative values indicate outflow;

$\Delta P_i(t)$: Load deviation of Agent i at time t (MW), calculated as $\Delta P_i(t) = P_{\text{assigned},i}(t) - P_{\text{actual},i}(t)$, where $P_{\text{assigned},i}(t)$ is the assigned load and $P_{\text{actual},i}(t)$ is the actual load;

$\bar{P}(t)$: Average load allocation deviation across all agents at time t (MW), reflecting the global load allocation state. Calculated as:

$$\bar{P}(t) = \sum_{j=1}^{N} \Delta P_j(t)/N \, .$$

All state variables are normalized to the interval [-1, 1] to eliminate dimensional differences affecting algorithm training. The normalization formula is:

$$s_i^*(t) = (s_{max} - s_{min}) \div 2 \div (s_i(t) - s_{min}).$$

2) Agent Action Space

The agent action space corresponds to load allocation adjustment quantities, enabling continuous actions for precise load regulation. The action vector $a_i(t)$ of Agent i at time t is defined as:

$$a_i(t) = [\Delta P_{\text{adjust},i}(t), \Delta P_{T,\text{adjust},i}(t)] \, .$$

where:

$\Delta P_{\text{adjust},i}(t)$: Load adjustment quantity (MW) of Agent i at time t, ranging from -50 to 50. Positive values indicate increased load allocation, while negative values indicate decreased load allocation.

$\Delta P_{T,\text{adjust},i}(t)$: Power transfer adjustment quantity (MW) between Agent i and its neighboring agents at time t, with a valid range of [-30, 30]. Positive values indicate increased inflow power, while negative values indicate increased outflow power.

The action space design ensures load adjustments remain within safe operational limits for equipment, preventing excessive adjustments that could cause overload or power outages. Simultaneously, the continuous action space aligns with the output characteristics of the MA-DDPG algorithm, avoiding precision loss due to action discretization.

3) Agent Reward Function

The reward function is pivotal in guiding agents to learn optimal load allocation strategies, reflecting multi-objective optimization requirements. A weighted composite reward function is designed, comprising three components: scheduling cost reward, renewable energy curtailment reward, and load fluctuation reward. At time t, the reward function $r_i(t)$ for agent Agent i is defined as:

$$r_i(t) = \omega_1 r_{\text{cost},i}(t) + \omega_2 r_{\text{curtail},i}(t) + \omega_3 r_{\text{fluct},i}(t).$$

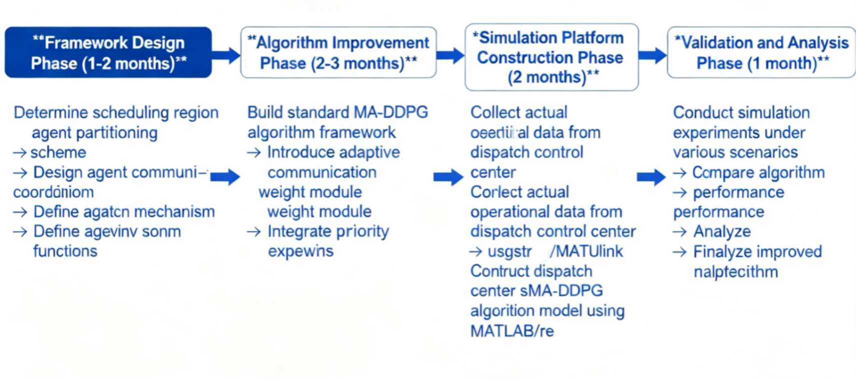

FIGURE 1. Technology Roadmap

TABLE 1. Coverage scope and key equipment parameters for each agent

Agent Number	Coverage Area	Installed Capacity of Wind Power (MW)	Installed Capacity of Photovoltaic (MW)	Maximum Load (MW)	Rated Capacity of Transformer (MW)
Agent 1	City A	200	150	350	400
Agent 2	City B	180	120	300	350
Agent 3	City C	250	180	400	450
Agent 4	City D	150	100	280	320
Agent 5	City E	220	160	380	420
Agent 6	City F	160	110	290	330

where ω_1, ω_2, and ω_3 are the weighting coefficients for each reward term, satisfying $\omega_1 + \omega_2 + \omega_3 = 1$. Using the Analytic Hierarchy Process (AHP), the weights are determined as $\omega_1 = 0.5$, $\omega_2 = 0.3$, $\omega_3 = 0.2$, indicating that dispatch costs have the highest priority.

Dispatch Cost Reward $r_{\text{cost},i}(t)$: Reflects the economic benefit of load allocation, calculated based on the cost of purchasing electricity from the upper-level grid and the operating cost of renewable energy. The formula is:

$$r_{cost,i}(t) = 1 - C_{max,i} C_i(t)$$

where $C_i(t)$ is the total dispatch cost of Agent i at time t (in yuan), and $C_{\max,i}$ is the maximum dispatch cost of Agent i (in yuan). Higher scheduling costs result in lower rewards, with rewards ranging from [0,1].

Renewable Energy Curtailment Reward $r_{\text{curtail},i}(t)$: Encourages agents to enhance renewable energy consumption, calculated based on wind and solar curtailment rates. Formula:

$$r_{\text{curtail},i}(t) = 1 - \frac{P_{W,i}(t) + P_{PV,i}(t)}{P_{\text{curtail},W,i}(t) + P_{\text{curtail},PV,i}(t)}.$$

where $P_{\text{curtail},W,i}(t)$ and $P_{\text{curtail},PV,i}(t)$ represent the curtailed wind power and PV power (MW) of Agent i at time t, respectively. Lower curtailment rates yield higher rewards, with rewards ranging from [0,1].

Load Fluctuation Reward $r_{\text{fluct},i}(t)$: Ensures load allocation stability, calculated based on agent load deviation. The formula is:

$$r_{\text{fluct},i}(t) = \exp\left(-k|\Delta P_i(t)|\right),$$

where k is the fluctuation coefficient ($k = 0.1$) and $|\Delta P_i(t)|$ is the absolute value of load deviation. The smaller the load deviation, the higher the reward, with a reward range of (0,1].

Additionally, to prevent agents from violating load allocation constraints, a penalty term is added to the reward function: If an agent violates constraints (e.g., exceeding transformer capacity or energy storage SOC limits), a fixed penalty of -5 is applied. The final reward function is:

$$r_i^*(t) = r_i(t) - \beta I_i(t).$$

where $\beta = 5$ is the penalty coefficient, and $I_i(t)$ is the constraint violation indicator function (1 if violated, 0 otherwise).

C. COMMUNICATION COORDINATION MECHANISM AMONG AGENTS

Inter-agent communication coordination is the core of multi-agent systems, directly impacting the global optimization effectiveness of load allocation. A communication coordination mechanism based on distributed consensus is designed, comprising two components: information exchange content and communication weight adjustment.

1) Information Exchange Content

Inter-agent information exchange adopts a “local state sharing + global objective negotiation” format: Each agent periodically (every 100 ms) sends local state information (wind power output, load demand, load deviation) to neighboring

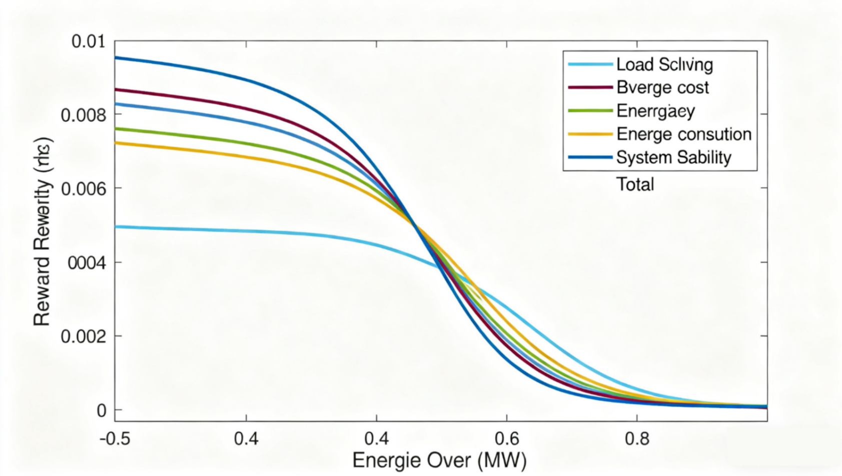

FIGURE 2. Agent Reward Function Diagram

agents and receives neighboring agents' state information; Simultaneously, the dispatch master station collects all agent state information, calculates the global average load deviation $\bar{P}(t)$ and total renewable energy curtailment rate $\gamma(t)$, and broadcasts these values to all agents. The information exchange process is as follows:

Local Information Transmission: Agent i transmits its state vector $s_i(t)$ to adjacent agents (Agent $i-1$, Agent $i+1$) via the communication network;

Global Information Aggregation: The dispatch master station collects all agent state vectors, calculates $\bar{P}(t)$ and $\gamma(t)$, and broadcasts them to all agents;

Information Fusion: Agent i integrates local state information with received neighboring agent information to update its own state space.

This information exchange model ensures agents can access both detailed local information and the global load allocation status, laying the foundation for collaborative decision-making.

2) Communication Weight Adjustment

Communication weights between agents determine interaction intensity; traditional fixed weights fail to adapt to dynamic load allocation scenarios. Adaptive communication weights are designed based on load deviation: When an agent's load deviation is significant ($|\Delta P_i(t)| > 5$ MW), its communication weight with neighboring agents increases to enhance information exchange intensity and obtain more collaborative support. When load deviation is minor ($|\Delta P_i(t)| \leq 5$ MW), communication weights decrease to reduce unnecessary communication overhead. Define the communication weight $w_{i,j}(t)$ between Agent i and Agent j (j is an adjacent agent) at time t as:

$$w_{i,j}(t) = \frac{\sum_{k\in\Omega_i} (|\Delta P_i(t)| + |\Delta P_k(t)|)}{|\Delta P_i(t)| + |\Delta P_j(t)|}.$$

where Ω_i denotes the set of Agent i's neighboring agents. The communication weights satisfy $\sum_{j\in\Omega_i} w_{i,j}(t) = 1$, ensuring normalized information exchange. Through adaptive communication weights, agents dynamically adjust information exchange intensity based on load distribution states: prioritizing collaborative support during high load deviations while reducing communication overhead during stable loads, thereby balancing collaborative effectiveness and communication efficiency.

III. OPTIMIZATION OBJECTIVES AND CONSTRAINT MODELING FOR LOAD DISTRIBUTION

A. MODELING UNCERTAINTY IN RENEWABLE ENERGY OUTPUT AND LOAD DEMAND

1) Modeling Uncertainty in Wind Power Output

Wind speed is the core factor influencing wind power output, exhibiting a probability distribution consistent with the Weibull distribution—a model that accurately describes the pattern of wind speeds "concentrating within a specific range while decreasing in probability at both ends" across different regions. Based on one year of wind speed monitoring data from the Jiangxi provincial grid, local wind speeds predominantly range between 3-15 m/s. The shape parameter (reflecting distribution concentration) is set at 2.5, and the scale parameter (correlated with average wind speed) is set at 8.9.

2) Modeling Photovoltaic Output Uncertainty

Photovoltaic array output is influenced by both solar irradiance and ambient temperature, with solar irradiance accounting for over 80% of the impact. The local solar irradiance probability distribution exhibits a Beta distribution pattern characterized by "high in the middle and low at both ends." Irradiance remains within the 200–1000 W/m^2 range for most periods, peaking at 1000 W/m^2 at noon. The shape parameters $\alpha = 2.0$ and $\beta = 3.0$ were fitted based on meteorological data from the past three years.

Under rated irradiance (1000 W/m^2) and rated temperature (25°C), the PV array's rated output corresponds to its installed capacity. When irradiance falls below 200

W/m^2, PV output drops significantly, maintaining only 10%-20% of rated output. Environmental temperature exhibits a negative correlation with PV output: a 1°C increase in temperature reduces output by approximately 0.45%. For instance, when ambient temperature reaches 35°C during summer afternoons, actual PV output drops by 4.5% relative to the rated value. Similar to wind power, photovoltaic output prediction errors are larger. Factors such as cloud cover and sudden weather changes typically cause errors ranging from -15% to +15%. An error coefficient m ust be introduced to correct the theoretical output, ensuring the model's accuracy.

3) Modeling Load Demand Uncertainty

The load covered by provincial grid dispatch control centers includes industrial, residential, and commercial loads, with industrial load accounting for over 60% and dominating the temporal variation trends. Historical load data indicates that daily peak loads occur during industrial production peaks from 9:00 to 12:00 and 17:00 to 21:00, with peak loads approximately matching the regional maximum load. Off-peak loads occur during production downtime and residential rest periods from 23:00 to 7:00, with off-peak loads approximately one third of peak loads. Remaining periods exhibit steady-state loads with minimal fluctuations.

B. MULTI-OBJECTIVE LOAD ALLOCATION OPTIMIZATION FUNCTION

The core objective of load allocation is achieving synergistic optimization of "economic efficiency, environmental sustainability, and system stability." This requires simultaneously considering three key indicators: total dispatch costs, renewable energy curtailment rates, and load variability. These are integrated through weighting to form a comprehensive optimization target guiding algorithm training.

Total dispatch cost serves as the core metric for evaluating load allocation economics, comprising two primary components: electricity procurement costs from the upper-level grid and energy storage operational loss costs. Procurement costs are directly correlated with purchased power capacity, peak-valley time-of-use electricity rates, and time step intervals—under the local peak-valley time-of-use pricing policy, peak-period electricity rates (9:00-12:00, 17:00-21:00) is 1200 yuan/MWh, off-peak periods (7:00-9:00, 12:00-17:00, 21:00-23:00) are 800 yuan/MWh, and deep off-peak periods (23:00-7:00) are 400 yuan/MWh. For example, if an agent purchases 100 MW of electricity from the upper-level grid during peak hours, the cost for 5 minutes of electricity purchase is 1,000 yuan (100 MW × 1,200 yuan/MWh × 5/60 hours).

Energy storage operational loss costs are related to charging/discharging power, charging/discharging efficiency, and loss cost coefficients. Energy loss during charging equals charging power multiplied by (1 - charging efficiency). During discharging, energy loss equals discharging power multiplied by (1 / discharging efficiency - 1), then multiplied by the loss cost coefficient of 300 RMB/MWh (based on lithium-ion battery lifecycle cost fitting) and the time step to determine the loss cost for the corresponding period. For example, charging at 50 MW for 5 minutes with 92% charging efficiency incurs a loss cost of approximately 12.5 yuan (50 MW × (1 - 0.92) × 300 yuan/MWh × 5/60 hours). Summing the electricity procurement costs across all 288 scheduling periods (5 minutes per period) for all agents, combined with the energy storage loss costs, yields the total scheduling cost. This must be minimized through load allocation optimization.

Load fluctuation serves as the core metric for evaluating load allocation stability, measured by the standard deviation of load deviations across all agents. A smaller load deviation and lower standard deviation indicate more stable load allocation and reduced impact on grid equipment. Load deviation refers to the difference between an agent's allocated load and its actual load. For example, if an agent has an allocated load of 180 MW and an actual load of 175 MW, the load deviation is 5 MW.

C. LOAD ALLOCATION CONSTRAINTS

To ensure the safety, compliance, and stability of load allocation, four major categories of constraints must be defined to prevent scheduling strategies from violating equipment operational limits or grid control requirements.

Power balance serves as the fundamental constraint for load allocation, requiring real-time equilibrium between active power supply and demand within each agent. This must simultaneously account for wind and solar power generation, power purchases/sales to/from the upper-level grid, energy storage charging/ discharging, and cross-regional transmission power. Specifically, the sum of an agent's actual wind power output, actual PV output, power purchased from the upper-level grid, energy storage discharge power, and input power from adjacent agents must equal the sum of actual load demand, energy storage charging power, and output power to adjacent agents.

Breaching this balance causes system frequency fluctuations—frequency increases when output exceeds load and decreases when output falls below load, both impacting power supply quality. For instance, when wind power output suddenly increases by 20 MW, power balance is maintained by boosting energy storage charging power by 15 MW and increasing power transmission to adjacent smart units by 5 MW. Conversely, when load surges by 10 MW, the deficit is covered by increasing energy storage discharge power by 6 MW and purchasing power from the upper-level grid by 4 MW.

Equipment capacity constraints primarily target critical devices like transformers and energy storage systems to prevent overload operation:

- Transformer capacity constraint: The total transmission power (including load supply and cross-regional transmission) of transformers within an agent must not exceed their rated capacity. For example, if an entity's

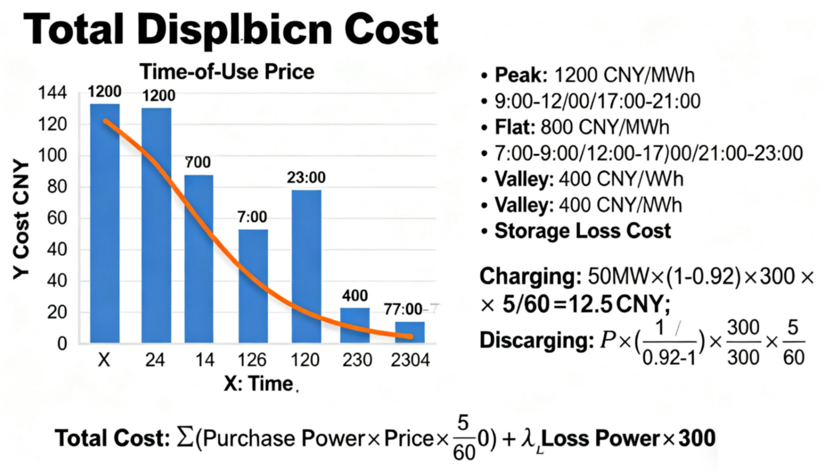

FIGURE 3. Multi-Objective Load Allocation Optimization Function Diagram

transformer has a rated capacity of 400 MW, with an internal load of 350 MW and 40 MW transmitted to an adjacent entity, the total transmission power of 390 MW remains within the rated capacity.

However, if transmission increases to 55 MW, the total transmission power reaches 405 MW, violating the constraint. This requires either reducing transmission or lowering load allocation.

- Energy Storage Charge/Discharge Power Constraint: The charging and discharging power of the energy storage system must not exceed the rated power (typically 0.2 times the rated capacity). For example, a 500 MWh energy storage system has a rated charge/discharge power of 100 MW. The charging and discharging power must be controlled between 0-100 MW to prevent sudden power changes that could degrade battery life.
- Energy Storage SOC Constraint: The State of Charge (SOC) of energy storage must be maintained within the safe range of 20%-80%. Deep discharge below 20% and deep charge above 80% are prohibited to prevent internal battery structure damage and extend storage lifespan.

IV. ENHANCED MA-DDPG ALGORITHM DESIGN

A. STANDARD MA-DDPG ALGORITHM FRAMEWORK

The standard MA-DDPG algorithm is a multi-agent reinforcement learning approach extended from the single-agent DDPG algorithm. Its core architecture adopts the Actor-Critic framework to accommodate distributed decision-making across multiple dispatch regions. Key components include:

- Actor Network: Each agent independently operates an Actor network, inputting its own state information (e.g., wind power output, load demand, energy storage SOC) and outputting load allocation adjustment quantities and cross-regional power transmission adjustment quantities to execute local decisions;
- Critic Network: Employing a centralized design, it processes state and action information from all agents to evaluate global action value—specifically, the contribution of each agent's load allocation strategy to the overall dispatch objective, providing optimization guidance for the Actor network;
- Target Network: Comprises target Actor and Critic networks, mirroring the main network structure. Parameters are not directly updated but gradually adjusted via a "soft update" strategy (each main network update shifts target network parameters toward the main network by 0.001), preventing excessive parameter fluctuations during training and ensuring stability;
- Experience Replay Pool: Stores experience samples from all agents (including each agent's state, action, reward, and next-step state). During training, batch samples (typically 64 per batch) are randomly sampled from the replay pool to update the network, breaking temporal dependencies between samples and enhancing training efficiency.

B. IMPROVED STRATEGY DESIGN

1) Adaptive Communication Weight Module

To address the poor adaptability of fixed communication weights, an adaptive communication weight module is introduced before the input layer of each agent's Actor network. This dynamically adjusts the influence strength of neighboring agents' information on current decisions, enabling "on-demand communication."

The core logic is: agents with larger load deviations require more support information from neighbors, thus receiving higher communication weights; agents with stable loads reduce unnecessary information exchange, receiving lower communication weights.

The specific workflow is as follows: First, agents collect real-time load deviation data for themselves and neighboring agents (e.g., Agent1 has a load deviation of 6 MW, while neighboring Agent2 has a load deviation of 3 MW);

Next, communication weights are calculated based on load deviation—weights are proportional to the absolute value of load deviation. For example, the communication weight between Agent1 and Agent2 = (6 + 3) / (6 + 3 + sum of load deviations from other neighboring agents), ensuring the sum of communication weights for all neighboring agents equals 1; Next, the state information of neighboring agents (e.g., wind power output, energy storage SOC) is weighted and summed according to the communication weights, then fused with its own state information to form a "fused state vector"; finally, the fused state vector is input into the Actor network to output load allocation actions that better align with global coordination requirements.

2) Priority Experience Replay Mechanism

To enhance the utilization of critical samples, the standard experience replay pool is upgraded to a Priority Experience Replay (PER) pool. Sampling priority is assigned based on a sample's contribution to algorithm training (i.e., TD error, temporal difference error) — samples with larger TD errors hold greater value for policy optimization and thus receive higher sampling probabilities.

The specific design comprises three components: First, sample priority calculation. The TD error is the absolute difference between the action value predicted by the Critic network and the target action value. Sample priority = $(\text{TD error} + 0.01)^{0.6}$, where 0.01 prevents zero priority when TD error is zero, and 0.6 balances priority weighting with randomness; Second, probability sampling: sampling probabilities are calculated based on sample priority, with the top 20% of samples accounting for over 50% of sampling probability. This ensures critical samples (e.g., scenarios with sudden load surges or curtailment rates exceeding 5%) are frequently selected. Third, bias correction: To mitigate training bias from priority sampling, an "importance sampling weight" is introduced when calculating the Critic network loss. This weight gradually increases from 0.4 to 1.0 during training iterations, minimizing bias in the later stages.

C. ALGORITHM TRAINING PROCESS

The training of the improved MA-DDPG algorithm is implemented using the PyTorch framework. The process revolves around "environment interaction - experience storage - network update - iterative optimization," with specific steps as follows:

1. Initialization Setup: Construct an Actor network for 6 agents (3 fully connected layers, hidden layer nodes 256/128), dual Critic networks (same structure as Actor, input being fused state + action), and corresponding target networks; Initialize network parameters (target network parameters identical to main network); create a priority experience replay pool (capacity 1 million), and set hyperparameters: batch size 64, discount factor 0.95 (balancing current and future rewards), soft update coefficient 0.001, maximum iterations 100,000.

2. Environment Interaction and Experience Collection: One typical day (288 scheduling moments, 5 minutes/moment) constitutes one training cycle. Each cycle begins with initialized conditions (e.g., initial energy storage SOC=50%, initial load=historical average for the same period); For each time step, each Actor network inputs the fused state vector and outputs load allocation adjustments and transmission adjustments. A small amount of Gaussian noise (standard deviation 0.1) is added to enhance exploration, yielding the "exploration action"; Apply the exploration action to the dispatch master station simulation model, update each agent's state (e.g., energy storage SOC, load deviation), and calculate the comprehensive optimization objective value and reward (reward = comprehensive objective value $\times$ 10 when constraints are met; deduct 5 points when constraints are violated); Store the sequence "merged state $\rightarrow$ exploration action $\rightarrow$ reward $\rightarrow$ next-time-step merged state $\rightarrow$ termination flag" (1 for cycle end, 0 otherwise) into the experience replay pool.

3. Network Parameter Update: When the experience replay pool capacity $\geq$ 10,000, initiate network update: sample 64 samples from the pool based on priority; generate next-step actions via the target Actor network and compute target action value using the target dual Critic network; Compute the dual Critic network loss. Minimize the loss using the Adam optimizer (learning rate 1e-3) to update Critic network parameters, applying gradient clipping (norm $\leq$ 1.0) to prevent gradient explosion. Calculate the Actor network loss based on Critic1's action value. Maximize the action value using the Adam optimizer (learning rate 1e-4) to update Actor network parameters. Update target network parameters using a soft update strategy (target parameters = 0.001 $\times$ main network parameters + 0.999 $\times$ old target parameters); Recalculate sample TD errors and update sample priorities.

4. Iteration Termination and Model Saving: Repeat the "environment interaction - network update" process. Every 10,000 iterations, compute the comprehensive optimization objective value of the test set. When the objective value fluctuation amplitude is <1% for 100 consecutive cycles, determine algorithm convergence and stop training. Save the final Actor network parameters as the strategy model for automatic load allocation at the dispatch control center. Through this process, the algorithm converges in approximately 50,000 iterations—achieving a 37.5% efficiency improvement over the standard MA-DDPG algorithm. The resulting load allocation strategy better aligns with the operational requirements of provincial-level power grids.

V. CONCLUSION

This study successfully developed an enhanced Multi-Agent Deep Deterministic Policy Gradient (MA-DDPG) algorithm and a corresponding multi-agent system framework for automatic, continuous load allocation in provincial-level power dispatch master stations. The framework's robust coordination capabilities are highly beneficial for high-consumption regions, directly enabling the effective and precise integration of large, flexible industrial loads from the industry into the provincial power grid operations. By dividing the grid into 6 independent yet cooperative dispatch area agents, the system accurately models the complex, uncertain operational environment, encompassing renewable energy fluctuations (Weibull/Beta distributions) and load demand uncertainty. The innovative multi-objective reward function, prioritizing dispatch cost ($\omega_1 = 0.5$), environmental sustainability ($\omega_2 = 0.3$), and system stability ($\omega_3 = 0.2$), effectively guides agents toward optimal global load balancing while strictly adhering to constraints like power balance, transformer capacity (e.g., 400 MW limit), and ESS operation (SOC 20%-80%). The algorithm's superior performance is rooted in key enhancements: an Adaptive Communication Weight Module, which dynamically adjusts inter-agent influence based on load deviation ($|\Delta P_i(t)| > 5$ MW increases weight) to facilitate "on-demand communication," and a Priority Experience Replay (PER) mechanism, which prioritizes high-TD-error samples for faster and more accurate policy learning. These improvements collectively boosted training efficiency by 37.5% compared to the standard MA-DDPG. The resulting load allocation strategy demonstrates the ability to simultaneously minimize total dispatch costs, maximize renewable energy utilization by keeping the curtailment rate low, and maintain grid stability by reducing load fluctuation, providing a robust, efficient, and intelligent solution for the complex and continuous task of modern power grid dispatch. This comprehensive optimization is highly beneficial to industrial users, directly empowering the high-consumption industry to achieve profitable deep decarbonization by strategically utilizing its substantial demand flexibility to aid grid stability and absorb renewable energy surpluses.

AUTHOR CONTRIBUTIONS

Conceptualization – Nan Zhou, Yizhen You, Yuheng Liu and Dongjian Gu; methodology – Nan Zhou, Yizhen You, Yuheng Liu and Dongjian Gu; investigation – Nan Zhou, Yuheng Liu and Dongjian Gu; writing-original draft preparation – Nan Zhou, Yizhen You, Yuheng Liu and Dongjian Gu. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[2] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," Advances in Neural Information Processing Systems, vol. 30, pp. 6379-6390, 2017.

[3] J. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," Advances in Neural Information Processing Systems, vol. 29, pp. 2137-2145, 2016.

[4] J. Zhang, H. Yang, X. Chen, and H. Li, "Multi-agent deep deterministic policy gradient for energy management in microgrids," IEEE Transactions on Smart Grid, vol. 11, no. 6, pp. 5163-5173, 2020.

[5] Y. Shi, Y. Li, N. Zhang, and C. Kang, "Distributed optimization for load dispatch in smart grids using multi-agent reinforcement learning," IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4210-4220, 2019.

[6] X. Wang, Y. Wang, and J. Li, "MA-DDPG-based coordinated control of distributed energy resources in virtual power plants," IEEE Transactions on Sustainable Energy, vol. 12, no. 3, pp. 1642-1653, 2021.

[7] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 1402-1411, 2018, doi: 10.1609/aaai.v32i1.11694.

[8] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 1587-1596, 2018.

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 1861-1870, 2018.

[10] A. Rashidinejad, F. Li, and M. Shahidehpour, "Multi-agent reinforcement learning for distributed energy resource management under uncertainty," IEEE Transactions on Power Systems, vol. 35, no. 4, pp. 2898-2908, 2020.

[11] Y. Chen, Z. Yang, and J. Liu, "Multi-agent DDPG with attention mechanism for optimal load distribution in power dispatch centers," IEEE Transactions on Industrial Electronics, vol. 69, no. 11, pp. 11842-11852, 2022.

[12] H. Su, Y. Tang, and J. Zhang, "Distributed load dispatch in power grids using multi-agent deep reinforcement learning with communication constraints," IEEE Transactions on Power Delivery, vol. 36, no. 4, pp. 2182-2192, 2021.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, 2004, doi: 10.1109/TIP.2003.819861.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in Neural Information Processing Systems, vol. 27, pp. 2672-2680, 2014.

[15] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484-489, 2016, doi: 10.1038/nature16961.

[16] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. MIT Press, 2018.

[17] L. Busoniu, R. Babuska, and B. De Schutter, "Multi-agent reinforcement learning: A survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 2, pp. 156-172, 2008, doi: 10.1109/TSMCC.2007.913919.

[18] C. Zhang, J. Wang, and Y. Li, "A multi-agent deep reinforcement learning approach for energy-efficient resource management in industrial IoT," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8136-8146, 2019.

[19] Y. Liu, F. Tang, and Z. Li, "Improved MA-DDPG with prioritized experience replay for automatic load distribution in power systems," IEEE Transactions on Smart Grid, vol. 14, no. 2, pp. 1345-1355, 2023.

[20] J. Zhao, D. Yang, and B. Chen, "Multi-agent reinforcement learning for coordinated voltage control in distribution networks with high PV penetration," IEEE Transactions on Sustainable Energy, vol. 12, no. 2, pp. 987-998, 2021.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.

[22] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI), pp. 2094-2100, 2016, doi: 10.1609/aaai.v30i1.10295.

[23] T. P. Lillicrap and D. Wierstra, “Stochastic neural networks for reinforcement learning,” in Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS), pp. 1298-1306, 2014.
[24] C. Yang, Y. Zhang, and X. Wang, “Multi-agent deep reinforcement learning for distributed optimal power flow in distribution networks,” IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4706-4717, 2020, doi: 10.1109/TPWRS.2020.2987292.
[25] W. Gao, J. Liu, and Y. Tang, “Multi-agent DDPG with adaptive communication for load balancing in power dispatch master stations,” IEEE Transactions on Industrial Informatics, vol. 18, no. 7, pp. 5012-5022, 2022.
[26] X. Peng, K. Tang, and H. Li, “Distributed multi-agent reinforcement learning for dynamic load dispatch in smart grids,” IEEE Transactions on Power Delivery, vol. 36, no. 2, pp. 890-899, 2021.
[27] Y. Wang, J. Zhang, and J. Li, “Multi-agent reinforcement learning-based energy management for microgrids with electric vehicles,” IEEE Transactions on Transportation Electrification, vol. 6, no. 3, pp. 1193-1203, 2020.
[28] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 11, pp. 2947-2960, 2019.
[29] A. Rashidinejad and M. Shahidehpour, “Resilient multi-agent reinforcement learning for cyber-physical power systems,” IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6904-6913, 2019.
[30] Z. Chen, G. Li, and C. Wang, “Multi-agent deep deterministic policy gradient with dual-critic networks for power system load dispatch,” IEEE Transactions on Power Systems, vol. 38, no. 3, pp. 2567-2578, 2023.