SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2026, Vol. 69, Iss. 5, 152204:1–152204:14
https://doi.org/10.1007/s11432-025-4821-6

Optimized stochastic resource allocation using graph neural networks

Qing WANG, Yujue WANG, Bin XIN*, Haoran WANG & Jia ZHANG

School of Automation, Beijing Institute of Technology, Beijing 100081, China

Received 3 July 2025/Revised 5 September 2025/Accepted 9 October 2025/Published online 23 March 2026

Abstract Stochastic resource allocation is essential in optimizing decision-making processes across various domains. This study examines a representative instance of stochastic resource allocation (heterogeneous resource allocation and task assignment), where efficient utilization and coordination of diverse resources are critical for enhancing system-level performance. Despite extensive research on resource allocation, challenges persist owing to the inherent complexity of the problem, particularly in modeling the intricate relationships among different resource types and tasks. To address these challenges, this study proposes a heterogeneous resource allocation graph neural network with greedy construction algorithm (HRAGNN-GCA), which integrates graph neural networks with a greedy algorithm. The model incorporates resource-task matching probabilities and resource collaboration factors within its message-passing mechanism and employs a greedy algorithm to efficiently construct allocation schemes. Experimental results indicate that compared with existing heuristic methods, the proposed approach achieves notable improvements in both allocation quality and computational efficiency. Furthermore, comprehensive evaluations across various problem scales and scenarios confirmed the effectiveness and generalization capability of HRAGNN-GCA, particularly in large-scale instances.

Keywords stochastic resource allocation, heterogeneous resource assignment, graph neural networks, greedy algorithm

Citation Wang Q, Wang Y J, Xin B, et al. Optimized stochastic resource allocation using graph neural networks. Sci China Inf Sci, 2026, 69(5): 152204, https://doi.org/10.1007/s11432-025-4821-6

1 Introduction

Stochastic resource allocation [1–3] is a problem in which the resources assigned to a task may fail probabilistically to complete the assigned tasks. The defining characteristic of stochastic resources is that they do not guarantee the successful completion of a task and their effectiveness can be quantified probabilistically. Accordingly, a stochastic resource allocation scheme must be determined based on the probability of the occurrence of relevant random events. Specifically, after a resource is allocated to a task, the successful completion of the task is a random event, typically represented by the probability of success.

Stochastic resource allocation plays a fundamental role in optimizing decision-making processes across various domains, including logistics, communication networks, and energy management [4, 5]. In numerous real-world scenarios, resource allocation must be performed under uncertainty, where factors, such as incomplete information and probabilistic constraints have a significant impact on the decision-making strategies [6]. Traditional deterministic optimization approaches often fail to effectively accommodate such uncertainties, highlighting the need for stochastic optimization techniques. These methods, which incorporate probabilistic modeling and adaptive strategies, have been widely applied to solve resource allocation problems [7, 8].

Among these, heterogeneous resource and task assignments represent a typical and challenging instance of stochastic resource allocation, where the efficient coordination of diverse resources is critical for enhancing the overall system performance [9]. In complex environments, such as smart grids, large-scale logistics, and intelligent service systems, the dynamic and uncertain nature of resource availability and task demands make high-quality allocation strategies particularly important [10].

Although classic resource task assignment problems have been studied extensively, various optimization methods have been proposed, including improved genetic algorithms (GA) [11–13], enhanced particle swarm optimization (PSO) [14, 15], hybrid techniques that combine multiple optimization approaches [16, 17] and other intelligent optimization strategies [18–20]. Incorporating heterogeneous resource collaboration into a traditional assignment

* Corresponding author (email: brucebin@bit.edu.cn)

framework significantly increases the complexity of the problem, thereby posing new challenges. Multi-objective optimization improvements, such as the MOEA/D algorithm [21, 22] and reinforcement learning methods [23, 24] have also been introduced to enhance the intelligence of assignment strategies. However, introducing resource heterogeneity and interactions further expands the solution space, making the problem more complex.

Since the mid-20th century, research on resource allocation problems has resulted in the development of various models and optimization algorithms, driving their dynamic formalization [25]. However, traditional methods often focus on simplified resource-task matching, while neglecting the complex cooperative relationships among diverse resource types. Bogdanowicz et al. were among the first to introduce multitype resource assignment modeling and formulated it as a combinatorial optimization problem involving multiple resource types and tasks [26]. Subsequent research has deepened this direction by introducing various solution strategies, including differential evolution algorithms [27], heuristic algorithms that incorporate marginal benefits [28, 29], reinforcement learning approaches [30, 31], and other heuristic methods [32]. The author previously proposed a fast heuristic algorithm (KCH) to obtain effective solutions for multistage S-WTA problems [33]. Although these methods have made progress in improving assignment efficiency and computational performance, challenges remain in scalability, computational efficiency, and adaptability to dynamic environments, necessitating further breakthroughs.

In recent years, graph neural networks (GNNs) have emerged as powerful tools for solving combinatorial optimization problems, owing to their ability to capture complex relational structures [34,35]. Representative models include graph convolutional networks (GCN) [36], graph attention networks (GAT) [37], and GraphSAGE [38], which have achieved remarkable success in areas, such as traffic prediction [39, 40], healthcare [41, 42], recommendation systems [43, 44], social network analysis [45, 46], and combinatorial optimization [47–50]. In resource task assignment scenarios, GNNs have been applied to scheduling and allocation problems [51–57], offering a data-driven optimization paradigm. However, heterogeneous resource assignment involves complex multitype interactions, making it crucial to model the probabilistic dependencies between resource availability, collaboration, and task demands. In addition, maintaining computational efficiency in large-scale instances remains a significant challenge when applying GNNs to such problems. Luo et al. [58] investigated the joint spectrum allocation and power-control problem for multi-UAV radar perception networks, employing GAT and mean aggregation for feature aggregation. However, the computation of attention coefficients resulted in a complex model that was unsuitable for real-time random resource allocation tasks with stringent requirements. Peng et al. [50] compared the expressive powers of Vertex-GNN and Edge-GNN by utilizing vertex features for end-to-end outputs to generate resource allocation results directly. However, they did not address the weighting of the initial matching probabilities in randomized resource allocation problems.

This study proposes a GNN-based approach for solving the heterogeneous resource assignment problem by leveraging a domain knowledge-infused specialized network model. Contrary to conventional GNN architectures, the proposed model explicitly incorporates resource-task matching probabilities and resource collaboration factors as attention weights to enhance information aggregation. Furthermore, an efficient allocation algorithm that combines GNN-based inference with a greedy optimization mechanism to achieve a balance between computational efficiency and solution quality is proposed. Experimental validation against various benchmark methods demonstrated that the proposed approach achieved superior performance in terms of both allocation quality and computational cost. The contributions of this study are as follows: (1) a probability-enhanced GNN model is proposed to solve the heterogeneous resource assignment problem, where probabilistic information is leveraged to guide the learning process, improving adaptability and generalization in resource task allocation tasks; (2) a hybrid learning-heuristic allocation algorithm is designed, integrating GNN-based inference with greedy optimization to fully exploit the advantages of both approaches, achieving a balance between solution quality and computational efficiency; and (3) a comprehensive evaluation is conducted to demonstrate the effectiveness of the proposed method across different scenarios.

2 Problem description and modeling

The heterogeneous resource assignment problem is an extension of traditional resource-task allocation problems involving the joint allocation of various heterogeneous resources to multiple tasks. Contrary to conventional assignment problems, this scenario requires consideration of both the matching between resources and tasks and the collaborative relationships among different resource types. The goal is to allocate diverse resources effectively to maximize the overall task completion performance under resource constraints.

This problem involves coordinating allocations among different types, performances, and capabilities of resources, and multiple tasks to achieve optimal system-level objectives. Complexity arises from the need to consider the

heterogeneity of resources, such as the effectiveness of certain resources for specific tasks, collaborative capabilities among resource types, and the importance or priority levels of different tasks. In addition, resource limitations further increase the difficulty of solving the problem. Next, the problem and mathematical modeling approach are introduced in detail.

2.1 Problem description and notation

The heterogeneous resource assignment problem focuses on the intelligent allocation of resources to maximize overall task completion benefits. The core challenge lies in establishing a coordinated allocation scheme among multiple tasks and resource types to maximize task performance while efficiently utilizing the available resources.

Assume that there are T tasks denoted by indices $1, 2, \ldots, t, \ldots, T$. In terms of resources, there are two types of resources (Resource $\mathcal{A}$ and Resource $\mathcal{B}$), denoted by indices $1, 2, \ldots, a, \ldots, A$ and $1, 2, \ldots, b, \ldots, B$, respectively. Here, t represents the t-th task, a represents the a-th Resource $\mathcal{A}$, and b represents the b-th Resource $\mathcal{B}$. The allocation should consider the matching probabilities between Resource $\mathcal{A}$ and tasks, the contribution probabilities between Resource $\mathcal{B}$ and tasks, and the collaboration relationships between Resource $\mathcal{A}$ and Resource $\mathcal{B}$. The solution process seeks to allocate resources to maximize the overall task performance. For clarity, the variables used in the modeling process are listed in Table 1.

The ultimate goal is to achieve effective resource allocation and improve the overall system performance by solving the allocation of Resource $\mathcal{A}$-Resource $\mathcal{B}$-task triplets. The following key assumptions are made during the problem-solving process [29].

Assumption 1. One of Resource $\mathcal{A}$ or Resource $\mathcal{B}$ can only be assigned to one task.

In real-world scenarios, a resource may potentially participate in multiple tasks simultaneously; for model simplification, such scenarios are treated as multiple independent resources at the same location, each assigned to a single task. A classic example of stochastic resource allocation is the weapon-target assignment problem. Bogdanowicz [59] mentioned that each sensor-weapon pair can be assigned to at most one target. Xin et al. [20] noted that every weapon can shoot only one target simultaneously, because the majority of actual weapons can shoot only one target once. In addition, a weapon capable of simultaneously engaging multiple targets can be treated as separate weapons. Similarly, when a resource needs to be allocated to multiple tasks, it can be considered having multiple independent resources for distribution.

Assumption 2. A single task can be supported by multiple Resource $\mathcal{A}$ and multiple Resource $\mathcal{B}$ simultaneously.

This assumption fully considers the complexity of real-world heterogeneous resource management scenarios and provides a reasonable basis for modeling. For example, in the weapon-target assignment problem, multiple missiles (resources) are required to intercept a high-value target (task).

2.2 Mathematical programming model

Based on the above description and analysis, a mathematical programming model is formulated to better understand and address the heterogeneous resource assignment problem.

The primary objective is to maximize task performance using an effective allocation strategy with limited resources. Accordingly, the following objective function is formulated:

$$J = \frac{\sum_{t=1}^{T} v_t \cdot \left(1 - \prod_{a \in \Phi_A^t} (1 - p_{at})\right) \cdot \left(1 - \prod_{b \in \Phi_B^t} (1 - q_{bt})\right)}{\sum_{t=1}^{T} v_t}, \tag{1}$$

where Φ_A^t represents the set of Resource $\mathcal{A}$ involved in all assignments for task t and Φ_B^t represents the set of Resource $\mathcal{B}$ involved in all assignments for task t.

The constraints specify that each resource can be assigned to at most one task, i.e., no single resource can be allocated to multiple tasks. The constraints are as follows:

$$\text{s.t.} \quad \begin{cases} b_i \neq b_j, \ \forall i \neq j, \\ a_i \neq a_j, \ \forall i \neq j. \end{cases} \tag{2}$$

In this context, the heterogeneous resource assignment problem can be considered an extension of traditional resource task assignment problems. As such combinatorial problems are generally NP-complete, the heterogeneous resource assignment problem inherits at least the same level of computational complexity.

Table 1 Variable descriptions.

Variable symbol	Variable meaning	Variable description
A	Total number of Resource $\mathcal{A}$	Resource $\mathcal{A}$ indices: $1, 2, \ldots, a, \ldots, A$
B	Total number of Resource $\mathcal{B}$	Resource $\mathcal{B}$ indices: $1, 2, \ldots, b, \ldots, B$
T	Number of tasks	Task indices: $1, 2, \ldots, t, \ldots, T$
Φ_A	Set of Resource $\mathcal{A}$ indices	$\{1, 2, \ldots, a, \ldots, A\}$
Φ_B	Set of Resource $\mathcal{B}$ indices	$\{1, 2, \ldots, b, \ldots, B\}$
Φ_T	Set of task indices	$\{1, 2, \ldots, t, \ldots, T\}$
$P = [p_{at}]_{A\times T}$	Resource $\mathcal{A}$-task matching probability matrix	p_{at} represents the probability that Resource $\mathcal{A}$ a supports task t
$Q = [q_{bt}]_{B\times T}$	Resource $\mathcal{B}$-task contribution probability matrix	q_{bt} represents the probability that Resource $\mathcal{B}$ b contributes to task t
$M = [m_{ab}]_{A\times B}$	Resource collaboration relationship matrix	m_{ab} indicates whether Resource $\mathcal{A}$ a and Resource $\mathcal{B}$ b can collaborate (1 if yes, 0 if no)
$V = [v_t]_{1\times T}$	Task importance matrix	v_t represents the importance of task t
$X = [x_n]_{N\times 3} = [b_n \quad a_n \quad t_n]_{N\times 3}$	Allocation result	x_n represents the n-th resource-task assignment, where b_n, a_n, t_n represent the indices of Resource $\mathcal{B}$, Resource $\mathcal{A}$, and task in the n-th assignment
$\Phi_A^t = \left\{a_1^t, a_2^t, \ldots, a_{n_t}^t\right\}$	Set of Resource $\mathcal{A}$ indices assigned to task t in the allocation result	a_i^t represents the index of the i-th Resource $\mathcal{A}$ allocated to task t, n_t represents the number of assignments related to task t
$\Phi_B^t = \left\{b_1^t, b_2^t, \ldots, b_{n_t}^t\right\}$	Set of Resource $\mathcal{A}$ indices assigned to task t in the allocation result	b_i^t represents index of the i-th Resource $\mathcal{B}$ allocated to task t
$G = (V, E)$	Undirected graph representation of the problem	V represents the set of nodes and E represents the set of edges
V	Set of nodes	Includes Resource $\mathcal{A}$ nodes, Resource $\mathcal{B}$ nodes, and task nodes, with a total of $A + B + T$ nodes
E	Set of edges	Includes Resource $\mathcal{A}$-task matching probability edges, Resource $\mathcal{B}$-task contribution probability edges, and collaboration relationship edges, with a total of $(A + B)T + AB$ edges

3 Algorithm design based on graph neural networks

To address the heterogeneous resource assignment problem, it is crucial to first represent it as a graph-based structure that allows the leveraging of graph-based algorithms for an effective solution. This approach helps manage the complexity introduced by the heterogeneity of resources and their interactions.

The heterogeneous resource assignment problem can be effectively represented using a graph structure, where nodes correspond to Resource $\mathcal{A}$, Resource $\mathcal{B}$, and tasks, and the edges represent the relationships between these entities, such as matching probabilities, contribution probabilities, and collaboration capabilities. This graph representation facilitates the application of GNN to analyze and improve resource allocation.

The heterogeneous resource allocation graph neural network with greedy construction algorithm (HRAGNN-GCA) consists of two key components.

Design of the GNN: A GNN tailored specifically for the heterogeneous resource assignment problem is designed to achieve more accurate feature extraction. This network aims to exploit the information of nodes and edges fully, thereby enhancing the representational capacity of the problem graph. By integrating the node and edge features, the network can effectively capture the complex relationships inherent in the resource-task assignment problem and compute the importance of the edges during the allocation process.

Solution construction: After processing with the GNN, the enhanced graph is fused with the original graph to fully utilize the edge information before and after processing. Subsequently, a greedy algorithm is employed to rapidly generate an optimal solution and improve the efficiency of the assignment decision-making process.

The overall process of the solution algorithm is shown in Figure 1.

This algorithmic strategy integrates the advantages of GNN in feature reconstruction with the efficiency of greedy algorithms in solution construction, thereby providing a robust framework for addressing the heterogeneous resource assignment problem. The proposed approach provides an effective solution to this problem by leveraging complementary strengths.

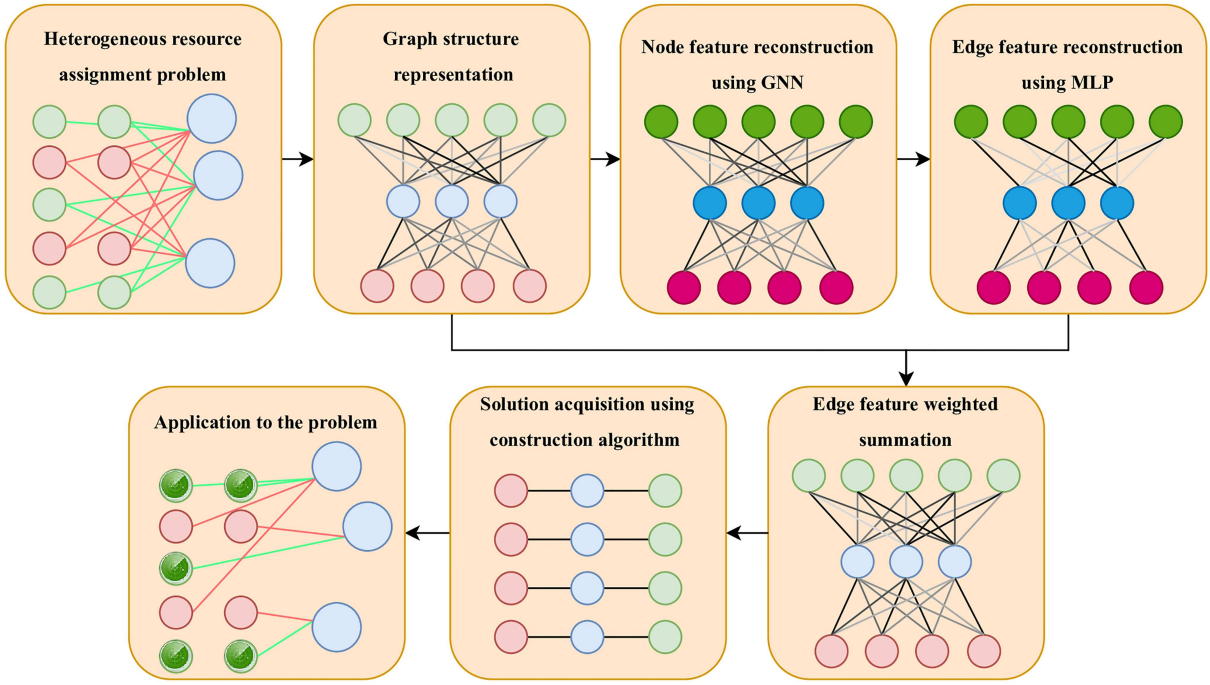

Figure 1 (Color online) Solution process based on the graph neural network algorithm.

3.1 GNN model for heterogeneous resource assignment

In this study, a specialized GNN model called the heterogeneous resource allocation GNN (HRAGNN) was designed for the heterogeneous resource assignment problem. This model is inspired by the graph isomorphism network (GIN) and attention mechanism. However, contrary to the traditional attention mechanism, which computes weighted values, the proposed approach directly utilizes the Resource $\mathcal{B}$-task contribution probability and Resource $\mathcal{A}$-task matching probability to determine the weight of the information transmission.

The node representation update formula for the model is as follows:

$$\mathbf{x}_i' = h_\Theta\left((1+\epsilon)\cdot\mathbf{x}_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\alpha_{ij}\mathbf{x}_j\right), \tag{3}$$

where $\mathbf{x}_i'$ represents the updated feature vector of node i; $\mathcal{N}(i)$ denotes the set of neighbors of node i; h_Θ represents the nonlinear transformation function for feature transformation, which in this case is implemented using an multilayer perceptron (MLP); ϵ denotes a learnable parameter; and α_{ij} is determined by the Resource $\mathcal{B}$-task contribution probability and the Resource $\mathcal{A}$-task matching probability, which control the weight of information propagation.

In the heterogeneous resource assignment problem, the interaction relationships among Resource $\mathcal{A}$, Resource $\mathcal{B}$, and the tasks are complex, and the manner in which information is propagated between nodes directly affects the final allocation decision. Traditional GNN models use uniform weights or learning-based attention mechanisms to aggregate neighboring information. However, the proposed model introduces a probability-based calculation for α_{ij} that eliminates the need for explicit weight learning. This approach aligns information propagation with the actual requirements of the resource-task assignment problem. For example, Resource $\mathcal{A}$ with a higher matching probability can guide information aggregation more effectively. Compelling another neural network to learn inherently known correlations not only increases model complexity and training burden but may also result in overfitting, especially when data are limited. By simplifying the model architecture and directly utilizing matching probabilities as α_{ij}, HRAGNN mitigates these challenges, allowing the model to focus its capacity on learning other essential transformations.

Compared with traditional GNN models, the proposed model more precisely captures the key factors in the heterogeneous resource assignment problem, making it more suitable for real-world application scenarios.

Table 2 Initial feature descriptions.

Type of nodes or edges	Feature representation	Feature description
Task node	$[v_t, 0, 0]$	Task nodes are encoded using one-hot encoding and multiplied by the task importance
Resource $\mathcal{B}$ node	$[0, 1, 0]$	Resource $\mathcal{B}$ nodes are encoded using one-hot encoding
Resource $\mathcal{A}$ node	$[0, 0, 1]$	Resource $\mathcal{A}$ nodes are encoded using one-hot encoding
Resource $\mathcal{A}$-Task matching probability edge	$[p_{at}, 0, 0]$	One-hot encoding is used and multiplied by the matching probability
Resource $\mathcal{B}$-Task contribution probability edge	$[0, q_{bt}, 0]$	One-hot encoding is used and multiplied by the contribution probability
Resource $\mathcal{A}$-Resource $\mathcal{B}$ collaboration relationship edge	$[0, 0, m_{ab}]$	One-hot encoding is used and multiplied by the collaboration relationship

3.2 Feature reconstruction process

First, the original features of the nodes in the graph are constructed. To differentiate between the different types of nodes and edges, one-hot encoding was used for the features. These nodes represent different entities, such as Resource $\mathcal{A}$, Resource $\mathcal{B}$, and tasks, whereas the edges signify their interactions, such as matching links or contribution relationships. Additionally, the corresponding attributes of the nodes and edges were considered. The initial features are presented in Table 2.

Next, the node features were reconstructed using the HRAGNN model. Subsequently, the edge features were reconstructed by computing the importance of each edge in the allocation process. This is achieved by concatenating the features of the edge and its two connected nodes, followed by a fully connected layer to determine edge importance. The importance of edges is central to the heterogeneous resource assignment process because it directly reflects the significance of the allocation relationships among resources and tasks. The detailed network architecture is shown in Figure 2.

3.2.1 *Node feature processing*

• First, the raw node features of the graph are processed. Initially, the node features are three-dimensional. An HRAGNN convolution layer is employed, which, based on the HRAGNN message-passing mechanism, aggregates the features of neighboring nodes and applies a nonlinear transformation through an MLP.

• In the message-passing mechanism, edge features are incorporated, where the probability information extracted from the edge features is used as attention coefficients to perform weighted aggregation of neighboring node information. This enhances the influence of edge information on node representations.

• The MLP in the HRAGNN convolution layer maps the aggregated three-dimensional node features to a 16-dimensional space, followed by a nonlinear transformation using the ReLU activation function. This process updates the node features to 16 dimensions.

3.2.2 *Edge feature reconstruction*

• After updating the node features, the enhanced node features are utilized to reconstruct the edge features. For each edge, the features of its two connected nodes are concatenated with its own edge features. Ultimately, each edge feature is represented as a 35-dimensional vector (32 dimensions from the concatenation of two node features and three dimensions from the original edge features).

• Finally, a two-layer MLP is applied to process the concatenated edge features. The Sigmoid activation function is used to predict the edge probability, thereby completing the edge prediction task. The importance of each edge is assessed, where values closer to 1 indicate higher importance.

Through the above process, the features of all nodes in the graph are first reconstructed. The importance of the edges is then calculated using the reconstructed node features. This measure of importance can be used to guide subsequent construction algorithms.

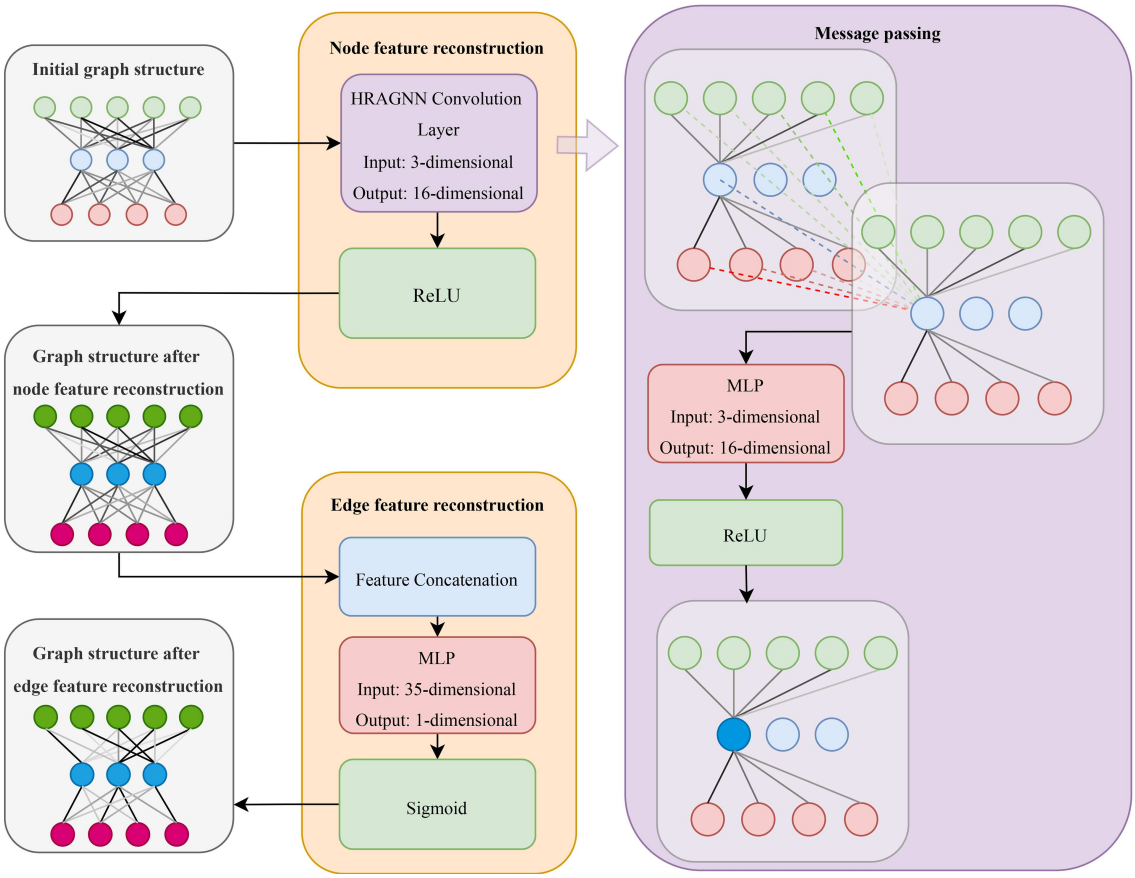

Figure 2 (Color online) Network structure.

3.3 Construction algorithm based on greedy algorithm

After calculating the importance of the edges, they are overlaid onto the edges of the original graph. This approach considers both the initial matching and contribution probabilities as well as the edge importance outputs from the GNN. Subsequently, for the overlaid graph, a greedy algorithm was employed to construct the allocation scheme. The specific process is as follows.

- **Step 1:** Sort the tasks in descending order based on their importance values, and then construct the allocation scheme sequentially.
- **Step 2:** Obtain all feasible resource-task chains for the current task, and calculate the evaluation value for each chain by summing the values of the two edges in the chain.
- **Step 3:** Select the chain with the highest evaluation value and store it in the allocation scheme.
- **Step 4:** Remove the nodes that have been allocated. If the task is not fully allocated, proceed to the next task and execute Step 2. If all tasks have been allocated, proceed to Step 5.
- **Step 5:** Once all tasks are allocated, output the final allocation scheme.

Using the above greedy algorithm, an efficient allocation scheme can be quickly generated from the graph, thereby significantly improving the solving efficiency.

3.4 Time complexity analysis

The algorithm proposed in this study consists of two main parts, as described in Subsections 3.2 and 3.3. Therefore, the overall time complexity is the sum of the time complexities of these two parts.

First, the feature reconstruction part described in Subsection 3.2 depends on the number of nodes and edges in the graph structure.

- **Node feature processing:** For the HRAGNN convolution layer, the time complexity is generally related to the number of nodes N and the number of neighbors for each node. In the worst case, where all different types of nodes are fully connected, the computational complexity for this layer is $O(B \times (A+T) + A \times (B+T) + T \times (B$

Table 3 Hyperparameter settings during training.

Hyperparameter	Value
Batch size	16
Optimizer	Adam
Learning rate	0.002
Weight decay	5×10^{-4}
Loss function	Binary cross-entropy loss
Number of epochs	100

$+A)$). Including the complexity of the ReLU activation function, which is $O(B + A + T)$, the total complexity is $O(A \times B + A \times T + B \times T)$.

• **Edge feature processing:** The reconstruction of each edge's features involves concatenating the features of two nodes and appending the original edge features. This process has a complexity of $O(E)$, which, in the worst case, becomes $O(B \times T + A \times T)$.

Thus, the overall time complexity of the feature reconstruction is $O(A \times B + A \times T + B \times T)$.

Second, the greedy algorithm mentioned in Subsection 3.3 primarily depends on the number of tasks T, Resource $\mathcal{A}$, and Resource $\mathcal{B}$:

• **Step 1:** The time complexity for sorting the tasks is $O(T \log T)$.

• **Step 2:** The evaluation of chains for each task involves checking all feasible Resource $\mathcal{A}$-Resource $\mathcal{B}$-task chains, with a time complexity of $O(A \times B)$ for a single task. Thus, the time complexity for Step 2 is $O(A \times B \times T)$.

In summary, the overall time complexity was $O(A \times (B + T) + B \times (A + T) + T \times (A + B)) + O(T \log T) + O(A \times B \times T)$. When the magnitudes of A, B, and T are similar and high, the overall time complexity is approximately $O(A \times B \times T)$.

4 Experiments and results analysis

The neural network model used during the inference stage must be trained. Comparative experiments and ablation studies are conducted to evaluate the performance of the model. After training, the model was employed to solve the heterogeneous stochastic resource allocation problem and was tested under different scenarios to assess the performance of the algorithm. The following sections describe these components and their corresponding results. Computational experiments were conducted on a computer equipped with an Intel(R) Core(TM) i7-14650HX CPU and 16 GB of RAM, running the Windows 11 operating system.

4.1 HRAGNN model experiments

Training data were obtained using an exact algorithm (EA) to derive an optimal allocation scheme for small-scale graph data. The dataset was designed to include scenarios with varying conditions, such as resource scarcity, resource balance, and resource abundance. Specifically, for scenarios with four units of Resource $\mathcal{B}$, five units of Resource $\mathcal{A}$, the numbers of tasks were set to two, three, four, five, six, seven, and eight, respectively, and 500 data samples were collected. This resulted in 3500 small-scale graph data samples and their corresponding optimal allocation schemes.

To validate the effectiveness of the HRAGNN model, 80% of the dataset was used for training, whereas the remaining 20% was used to evaluate model performance. The training performances of the different models were compared using identical training parameters. The goal of training is to predict the importance of edges not included in the allocation scheme as zero, whereas the edges included in the scheme are predicted with an importance value of one. During the training process, downsampling was performed on the edge data within the graph to balance the sample distribution, and the model was evaluated using the test set.

To ensure a comprehensive evaluation, HRAGNN was compared with multiple graph-based models, including GCN, GAT, GraphSAGE, GIN, GTN, and GraphBERT. Additionally, an ablation study was conducted using HRAGNN_without_alpha, a variant of HRAGNN that excludes probabilistic weighting, to assess the contribution of this component. Furthermore, MLP, which is a model without GNN-based feature extraction, was included to highlight the necessity of leveraging the graph structures for this task.

The hyperparameters used in the training process are listed in Table 3.

The loss curves for different models during training are illustrated in Figures 3(a) and (b), where Figure 3(a) represents the training loss and Figure 3(b) shows the validation loss.

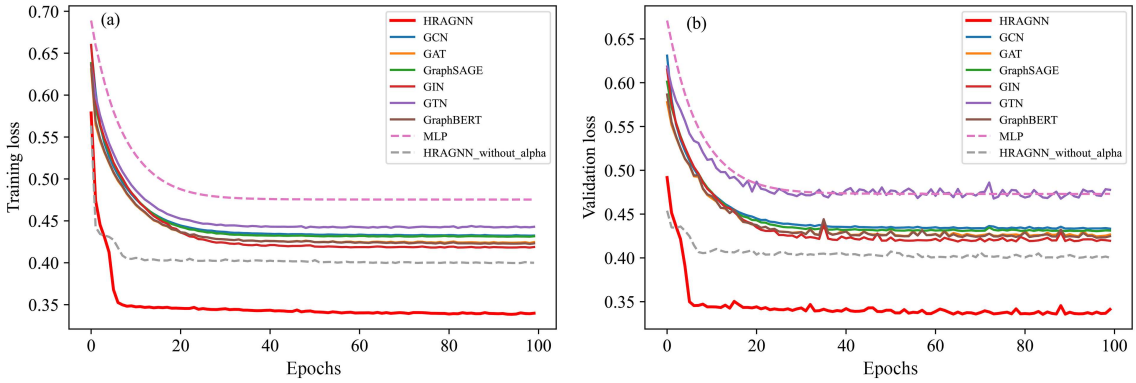

Figure 3 (Color online) Loss curves for different models. (a) Training loss; (b) validation loss.

Table 4 Model performance comparison. Bold indicates superior performance.

Model	Accuracy	Precision	Recall	F1-score	AUC	Training time (s)	Inference time (s)
HRAGNN	**0.8498**	**0.8664**	0.8271	**0.8463**	**0.9297**	186.8841	0.0063
GCN	0.7947	0.7807	0.8196	0.7997	0.8785	187.7563	0.0070
GAT	0.7957	0.8002	0.7882	0.7942	0.8826	197.8064	0.0085
GraphSAGE	0.7982	0.7762	0.8380	0.8059	0.8799	194.8057	0.0057
GIN	0.8000	0.7938	0.8105	0.8021	0.8855	192.7042	0.0063
GTN	0.7749	0.7288	**0.8757**	0.7955	0.8590	179.6394	0.0087
GraphBERT	0.7976	0.7998	0.7939	0.7968	0.8834	182.2251	0.0038
MLP	0.7717	0.7686	0.7775	0.7730	0.8545	187.6205	0.0079
HRAGNN_without_alpha	0.8104	0.7957	0.8354	0.8151	0.8965	172.2234	0.0099

As illustrated in Figure 3, the proposed HRAGNN model demonstrates exceptional performance in both training and validation loss convergence. Compared with conventional models and the method proposed by Luo et al. [58], HRAGNN achieves a faster convergence speed and lower loss values, indicating enhanced generalization capabilities. Furthermore, ablation studies highlight the critical role of the probabilistic weighting mechanism—its removal results in significant performance degradation. The MLP model also underperforms, highlighting the necessity for graph-based feature extraction.

A set of widely used metrics for binary classification was employed to evaluate the classification performance of each model comprehensively. The results are summarized in Table 4.

The results indicated that the proposed HRAGNN model outperformed all baseline methods across key classification metrics. It achieved the highest accuracy, precision, F1-score, and AUC, demonstrating its strong discriminative capability. In terms of computational efficiency, the training and inference times of HRAGNN remain competitive, ensuring its practical applicability. The proposed HRAGNN model was motivated by GIN and GAT. In (3), a learnable parameter is introduced to preserve the original central node information. An attention-based mechanism is adopted to aggregate neighbor information. However, the attention coefficients are determined by matching the probabilities and do not require explicit computation. This approach is based on the fact that the matching probability between resource allocation and a given task reflects the suitability of the resource for the task. Because the objective of employing an attention mechanism is to precisely capture this suitability, directly using the matching probability aligns more closely with practical requirements and expedites the inference process.

4.2 Algorithm evaluation

To comprehensively assess the performance of the HRAGNN-GCA under different scenarios, test instances were designed on three scales: small ($A = 5, B = 4$), medium ($A = 25, B = 20$), and large ($A = 60, B = 50$). Each scale was further subdivided into three scenarios: resource abundance ($T = \frac{3}{4}B$), resource balance ($T = B$), and high task pressure ($T = \frac{5}{4}B$). For each scenario, 100 instances were established, and the average performance across these 100 instances was used to evaluate the effectiveness of the algorithm. The details of the test scenarios are presented in Table 5.

To evaluate the performance of the HRAGNN-GCA algorithm comprehensively, four well-established algorithms were selected for comparison: the EA, GA, PSO, knowledge-based constructive heuristic (KCH), and deep Q-learning (DQN) [60]. The EA uses exhaustive search to solve small-scale problems and functions as a performance

Table 5 Example settings.

Example size	ID	Example description	A	B	T	Number of examples
Small scale	1	Resource abundant scenario	5	4	3	100
	2	Resource-task balanced scenario	5	4	4	100
	3	Task overload scenario	5	4	5	100
Medium scale	4	Resource abundant scenario	25	20	15	100
	5	Resource-task balanced scenario	25	20	20	100
	6	Task overload scenario	25	20	25	100
Large scale	7	Resource abundant scenario	60	50	38	100
	8	Resource-task balanced scenario	60	50	50	100
	9	Task overload scenario	60	50	62	100

Table 6 Experimental results of the evaluation value J (mean ± standard deviation). Bold indicates superior performance.

ID	EA	PSO	GA	KCH	DQN	HRAGNN-GCA
1	**0.6278±0.1069**	0.6160±0.1079	0.6273±0.1071	0.6112±0.1084	0.6200±0.1100	0.6215±0.1102
2	**0.5474±0.1026**	0.5253±0.1044	0.5468±0.1025	0.5323±0.0999	0.5350±0.1030	0.5399±0.1035
3	**0.5165±0.0923**	0.4897±0.0935	0.5161±0.0926	0.5042±0.0958	0.5044±0.0955	0.5046±0.0954
4	–	0.6338±0.0505	0.8312±0.0302	0.8647±0.0273	0.8735±0.0260	**0.8938±0.0248**
5	–	0.5290±0.0435	0.7448±0.0380	0.8161±0.0262	0.8224±0.0281	**0.8373±0.0280**
6	–	0.4493±0.0387	0.6698±0.0384	0.7567±0.0359	0.7522±0.0382	**0.7674±0.0388**
7	–	0.5156±0.0292	0.8277±0.0206	0.9338±0.0133	0.9446±0.0120	**0.9739±0.0063**
8	–	0.4167±0.0262	0.7328±0.0254	0.9015±0.0142	0.9050±0.0138	**0.9340±0.0112**
9	–	0.3579±0.0220	0.6531±0.0241	0.8609±0.0146	0.8650±0.0142	**0.8784±0.0138**

Table 7 Experimental results of the runtime (s). Bold indicates superior performance.

ID	EA	PSO	GA	KCH	DQN	HRAGNN-GCA
1	0.1184±0.0258	1.1366±0.0684	0.9638±0.0617	**0.0003±0.0005**	1.1032±0.0508	0.4039±0.0308
2	0.3704±0.0315	1.0899±0.0212	0.9266±0.0109	**0.0004±0.0005**	1.1231±0.0415	0.4059±0.0279
3	0.9135±0.0559	1.1057±0.0324	0.9411±0.0596	**0.0004±0.0005**	1.0937±0.0513	0.4025±0.0289
4	–	3.2029±0.1025	1.9159±0.0578	**0.0686±0.0280**	2.8024±0.1034	0.6848±0.0226
5	–	3.1696±0.0898	1.8900±0.0530	**0.0938±0.0290**	2.8534±0.0924	0.6904±0.0227
6	–	3.1668±0.0996	1.8591±0.0509	**0.1234±0.0294**	2.9044±0.1127	0.6931±0.0248
7	–	7.2812±0.1324	6.9803±0.1238	3.2027±0.1180	4.5062±0.1517	**1.0890±0.0709**
8	–	7.2700±0.1633	6.9628±0.1346	4.2500±0.1511	4.6064±0.1748	**1.0548±0.0484**
9	–	7.4200±0.1816	7.0202±0.1702	5.3565±0.2238	4.7517±0.2024	**1.0921±0.0512**

benchmark for these instances. However, owing to its unacceptable time complexity, it has only been used for small-scale problem comparisons. The GA and PSO are commonly used global optimization algorithms, whereas KCH is a heuristic based on marginal effects that can quickly generate good solutions.

The experimental results are summarized in Tables 6–8.

Table 6 presents the mean and standard deviation of the evaluation metrics for each algorithm across different scenarios, Table 7 reports the runtime under the same conditions, and Table 8 presents the results of the Wilcoxon rank-sum test to assess statistical significance at the 5% level. As shown in Table 6, HRAGNN-GCA exhibited a lower standard deviation in the evaluation metrics across multiple runs than the comparison algorithms. This demonstrates that the method not only achieves superior average performance but also exhibits greater stability and consistency, mitigating the performance fluctuations inherent in traditional heuristic algorithms caused by random initialization or random operations. This characteristic is crucial for the credibility and practicality of the method in real-world applications. As demonstrated by the time complexity analysis in Subsection 3.4, the computational efficiency of our method primarily stems from its polynomial-time GNN inference and the overall framework, which is dominated by a greedy allocation strategy. The runtime data presented in Table 7 for large-scale scenarios empirically corroborate this theoretical analysis, collectively validating the algorithm's efficiency and scalability when addressing large-scale randomized resource allocation problems.

The rapid decision-making capability of the solution method becomes particularly critical when addressing resource allocation problems with stringent real-time requirements. The Wilcoxon rank-sum test results further demonstrate that as the scenario scale expands, the advantage of HRAGNN-GCA over conventional methods be-

Table 8 Wilcoxon rank-sum test results for HRAGNN-GCA against other methods (p-value/significance).

ID	EA	PSO	GA	KCH	DQN
1	0.6583/No	0.6823/No	0.6725/No	0.5357/No	0.0123/No
2	0.5875/No	0.2307/No	0.6285/No	0.5332/No	0.0087/No
3	0.3731/No	0.2764/No	0.3904/No	0.9688/No	0.0054/No
4	–	0.0000/Yes	3.8377e−27/Yes	2.7865e−12/Yes	1.2548e−05/Yes
5	–	0.0000/Yes	1.3389e−29/Yes	3.8766e−07/Yes	9.8864e−06/Yes
6	–	0.0000/Yes	1.1329e−29/Yes	4.7800e−02/Yes	7.6543e−05/Yes
7	–	0.0000/Yes	0.0000/Yes	0.0000/Yes	0.0000/Yes
8	–	0.0000/Yes	0.0000/Yes	0.0000/Yes	0.0000/Yes
9	–	0.0000/Yes	0.0000/Yes	7.0711e−14/Yes	0.0000/Yes

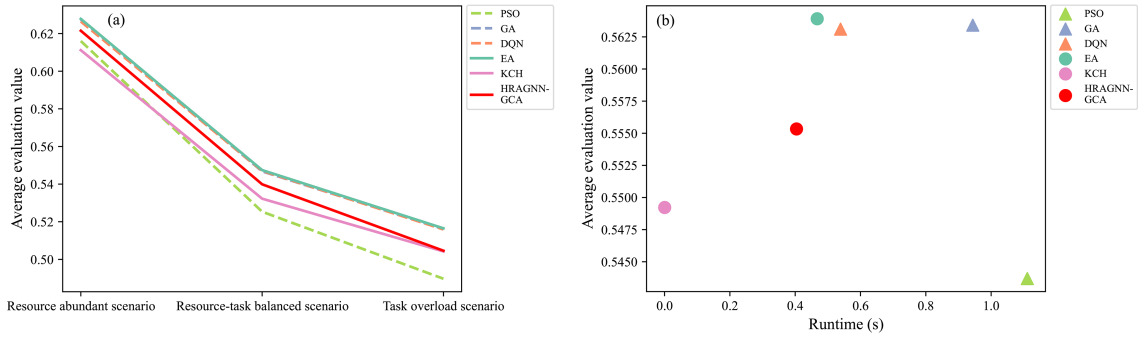

Figure 4 (Color online) Detailed experimental results in small-scale scenarios. (a) System efficiency with varying resource quantities; (b) runtime vs. system efficiency.

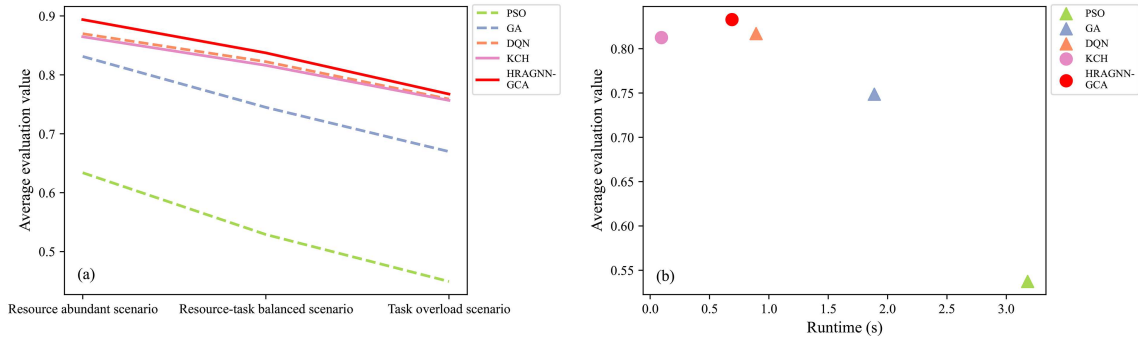

Figure 5 (Color online) Detailed experimental results in medium-scale scenarios. (a) System efficiency with varying resource quantities; (b) runtime vs. system efficiency.

comes increasingly pronounced. Moreover, the statistical findings effectively rule out the possibility that the superior performance observed on a small scale occurred by chance.

Figures 4–6 present the detailed experimental results for each algorithm across different scenarios, covering small, medium, and large scales. The graph on the left displays the average values of the evaluation function, where higher values indicate better performance. The right scatter plot illustrates the relationship between runtime and the evaluation value; specifically, points closer to the upper-left corner represent superior efficiency and performance.

To compare the results of the algorithms visually, the performance evaluation values and efficiencies across different scenarios are illustrated in Figure 7.

The observations reveal that, in small-scale instances, the GA results are very close to the optimal solution obtained by the EA, followed by the HRAGNN-GCA algorithm. However, in terms of computation time, the KCH algorithm outperformed the other algorithms.

In medium- and large-scale instances, the HRAGNN-GCA algorithm gradually showed its advantages, consistently achieving the best outcomes. While KCH maintained the fastest solution speed in medium-scale instances, as the problem size increased, HRAGNN-GCA surpassed it in terms of both solution quality and speed. The advantages of the GNN-GCA algorithm became more pronounced as the scale increased.

The superior performance of HRAGNN-GCA in large-scale instances may be attributed to the network's ability

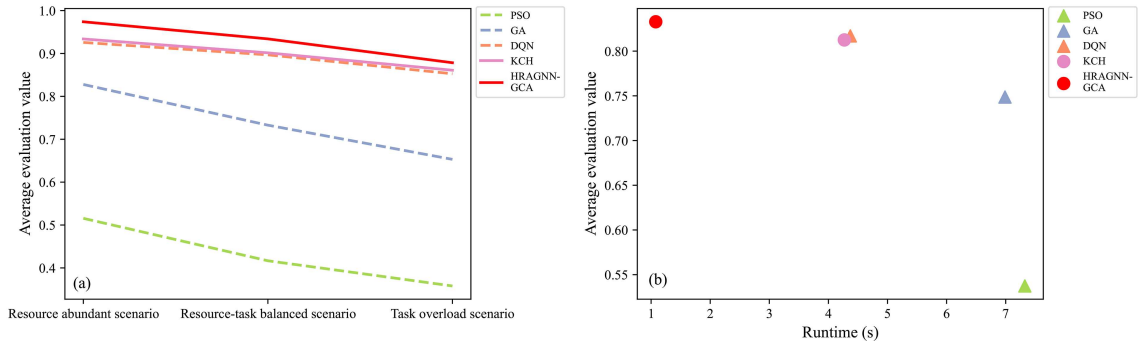

Figure 6 (Color online) Detailed experimental results in large-scale scenarios. (a) System efficiency with varying resource quantities; (b) runtime vs. system efficiency.

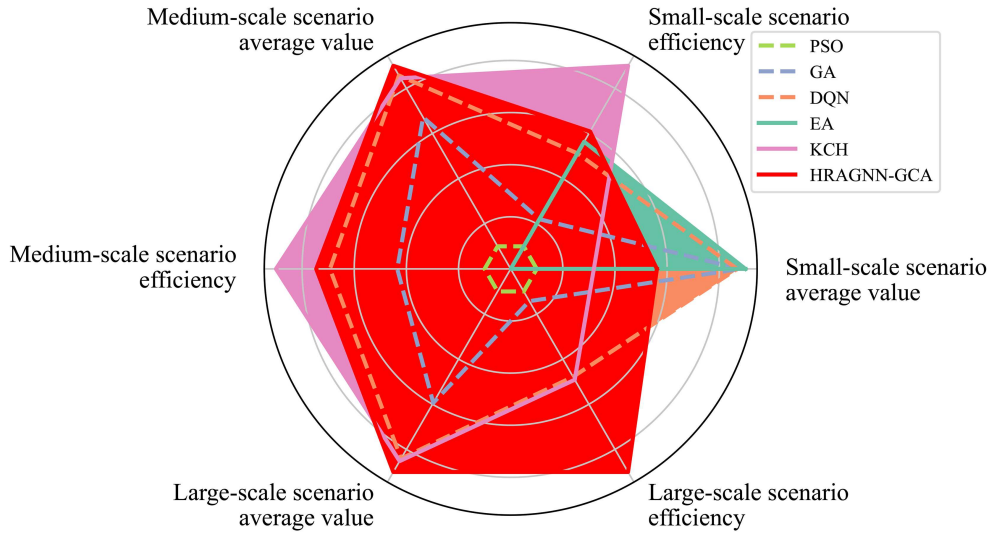

Figure 7 (Color online) Comparison of algorithm performance across scenarios.

to learn an appropriate method for reconstructing edges, allowing the results to reach relatively high levels. In addition, its parallel processing capability and rapid construction enable the algorithm to achieve the best runtime for large-scale instances. In contrast, traditional search algorithms, such as GA and PSO, experience performance degradation as the problem scale increases. As the problem size increases, the DQN algorithm requires exponentially increasing resources, resulting in low computational efficiency.

After training on small-scale samples encompassing diverse scenarios, HRAGNN-GCA demonstrated robust generalization capabilities for large-scale scenarios far exceeding the scope of the training data while maintaining exceptional resource allocation performance. Compared to traditional optimization algorithms (such as metaheuristics), the proposed approach exhibits significant advantages in scalability and practical applicability. Conventional methods often experience high computational complexity or convergence difficulties in large-scale problems, whereas our model achieves efficient inference owing to its generalization capability. This characteristic substantially enhances the adaptability and practical utility of the method for real-world large-scale stochastic resource allocation systems.

5 Conclusion

This study modeled a heterogeneous stochastic resource allocation problem and demonstrated the effectiveness of a GNN-based approach in solving it. The proposed HRAGNN-GCA method shows notable improvements in solution quality and computational efficiency compared with existing heuristic-based methods. It can be trained on small-scale instances and applied to large-scale instances. The experimental results confirm the competitive performance of this approach, particularly in large-scale scenarios involving complex resource-task relationships.

This method is broadly applicable to various resource allocation tasks under uncertainty, such as task scheduling,

energy management, and network resource optimization, where resource heterogeneity and probabilistic constraints must be considered.

Future research will focus on further optimizing the GNN architecture and exploring alternative neural network structures to enhance the solution quality in more complex and dynamic environments. Additionally, the integration of intelligent algorithms, such as reinforcement learning and metaheuristics, will be explored to further improve the allocation performance and adaptability.

Acknowledgements This work was supported in part by Beijing Natural Science Foundation (Grant No. 4252050), National Science Fund for Distinguished Young Scholars (Grant No. 62425304), and Basic Science Center Programs of National Natural Science Foundation of China (Grant No. 62088101).

References

1 Castanon D A, Wohletz J M. Model predictive control for stochastic resource allocation. IEEE Trans Automat Contr, 2009, 54: 1739–1750
2 Wang Y, Xin B, Chen J. An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources. IEEE Trans Cybern, 2021, 52: 11526–11538
3 Li J, Xin B, Pardalos P M, et al. Solving bi-objective uncertain stochastic resource allocation problems by the CVaR-based risk measure and decomposition-based multi-objective evolutionary algorithms. Ann Oper Res, 2021, 296: 639–666
4 Xu Y, Gui G, Gacanin H, et al. A survey on resource allocation for 5G heterogeneous networks: current research, future trends, and challenges. IEEE Commun Surv Tutorials, 2021, 23: 668–695
5 Aatabe M, Abbadi R E, Vargas A N, et al. Stochastic energy management strategy for autonomous PV-microgrid under unpredictable load consumption. IEEE Access, 2024, 12: 84401–84419
6 Li C, Grossmann I E. A review of stochastic programming methods for optimization of process systems under uncertainty. Front Chem Eng, 2021, 2: 622241
7 Chen Y, Xu J, Wu Y, et al. Dynamic task offloading and resource allocation for NOMA-aided mobile edge computing: an energy efficient design. IEEE Trans Serv Comput, 2024, 17: 1492–1503
8 Huang J, Lv B, Wu Y, et al. Dynamic admission control and resource allocation for mobile edge computing enabled small cell network. IEEE Trans Veh Technol, 2022, 71: 1964–1973
9 Cebrowski A K, Garstka J J. Network-centric warfare: its origin and future. In: Proceedings of US Naval Institute Proceedings, 1998. 124: 28–35
10 Paradis S, Benaskeur A, Oxenham M, et al. Threat evaluation and weapons allocation in network-centric warfare. In: Proceedings of 2005 7th International Conference on Information Fusion, 2005
11 Cheng Q, Chen D R, Gong J L. Weapon-target assignment of ballistic missiles based on Q-learning and genetic algorithm. In: Proceedings of 2021 IEEE International Conference on Unmanned Systems (ICUS), 2021. 908–912
12 Ruining L, Yan Z. Improved genetic algorithm for weapon target assignment problem. In: Proceedings of 2021 International Symposium on Computer Technology and Information Science (ISCTIS), 2021. 19–23
13 Huang J, Li X, Yang Z, et al. A novel elitism co-evolutionary algorithm for antagonistic weapon-target assignment. IEEE Access, 2021, 9: 139668
14 Kong L, Wang J, Zhao P. Solving the dynamic weapon target assignment problem by an improved multiobjective particle swarm optimization algorithm. Appl Sci, 2021, 11: 9254
15 Zhai H R, Wang W H, Li Q Z, et al. Weapon-target assignment based on improved PSO algorithm. In: Proceedings of 2021 33rd Chinese Control and Decision Conference (CCDC), 2021. 6320–6325
16 Xu Q Q, Li K Q, Yue Z Q, et al. Weapon target allocation based on GA-APSO algorithm. IEEE Access, 2024, 12: 164337
17 Fang F, He J F, Li Q W, et al. Weapon-target assignment based on improved particle swarm optimization for different allocation criteria. In: Proceedings of 2021 China Automation Congress (CAC), 2021. 6628–6633
18 Zhang J, Kong M, Zhang G, et al. Weapon-target assignment using a whale optimization algorithm. Int J Comput Intell Syst, 2023, 16: 62
19 Xin B, Chen J, Zhang J, et al. Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. IEEE Trans Syst Man Cybern C, 2010, 40: 649–662
20 Xin B, Chen J, Peng Z, et al. An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem. IEEE Trans Syst Man Cybern A, 2010, 41: 598–606
21 Yi X, Yu H, Xu T. Solving multi-objective weapon-target assignment considering reliability by improved MOEA/D-AM2M. Neurocomputing, 2024, 563: 126906
22 Wu X, Chen C, Ding S. A modified MOEA/D algorithm for solving bi-objective multi-stage weapon-target assignment problem. IEEE Access, 2021, 9: 71832–71848
23 Li S, He X, Xu X, et al. Weapon-target assignment strategy in joint combat decision-making based on multi-head deep reinforcement learning. IEEE Access, 2023, 11: 113740
24 Wang X, Zhang Y, Wang G. Target assignment for multiple stages of weapons systems using a deep Q-learning network and a modified artificial bee colony method. Comput Electrical Eng, 2024, 118: 109378
25 Ahuja R K, Kumar A, Jha K C, et al. Exact and heuristic algorithms for the weapon-target assignment problem. Operations Res, 2007, 55: 1136–1146
26 Bogdanowicz Z R, Coleman N P. Sensor-target and weapon-target pairings based on auction algorithm. In: Proceedings of the 11th WSEAS International Conference on Applied Mathematics, 2007. 92–96
27 Cao Z Q, Zhang Y J, Li Y, et al. Solving the dynamic sensor/weapon-target assignment problem by generation strategy optimization. In: Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control, 2022. 546–555
28 Li G, He G, Zheng M, et al. Uncertain sensor-weapon-target allocation problem based on uncertainty theory. Symmetry, 2023, 15: 176
29 Xin B, Wang Y, Chen J. An efficient marginal-return-based constructive heuristic to solve the sensor-weapon-target assignment problem. IEEE Trans Syst Man Cybern Syst, 2018, 49: 2536–2547
30 Li C, Xin B, He Y M, et al. Dynamic weapon target assignment based on deep Q network. In: Proceedings of 2023 42nd Chinese Control Conference (CCC), 2023. 1773–1778
31 Hu T, Zhang X, Luo X, et al. Dynamic target assignment by unmanned surface vehicles based on reinforcement learning. Mathematics, 2024, 12: 2557
32 Wang L J, Li J X, Wang G. Piecewise linearization and neighborhood-structure-based heuristic for the sensor-weapon-target assignment problem. In: Proceedings of 2024 China Automation Congress (CAC), 2024. 504–509
33 Wang Y P, Xin B, Chen J. Modeling and optimization of multi-stage sensor-weapon-target assignment. Control Theory Appl, 2019, 36: 1886–1895
34 Xia F, Sun K, Yu S, et al. Graph learning: a survey. IEEE Trans Artif Intell, 2021, 2: 109–127
35 Cappart Q, Chetelat D, Khalil E B, et al. Combinatorial optimization and reasoning with graph neural networks. J Mach Learn Res, 2023, 24: 1–61

36 Zhang S, Tong H, Xu J, et al. Graph convolutional networks: a comprehensive review. Comput Soc Netw, 2019, 6: 1–23
37 Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of ICLR, 2017
38 Hamilton W, Ying Z T, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of Adv. Neural Inf. Process., 2017
39 Chen C, Li K, Teo S G, et al. Gated residual recurrent graph neural networks for traffic prediction. AAAI, 2019, 33: 485–492
40 Jiang W, Luo J. Graph neural network for traffic forecasting: a survey. Expert Syst Appl, 2022, 207: 117921
41 Li Y, Qian B, Zhang X, et al. Graph neural network-based diagnosis prediction. Big Data, 2020, 8: 379–390
42 Zhang X M, Liang L, Liu L, et al. Graph neural networks and their current applications in bioinformatics. Front Genet, 2021, 12: 690049
43 Wu S W, Sun F, Zhang W T, et al. Graph neural networks in recommender systems: a survey. ACM Comput Surv, 2022, 55: 1–37
44 Gao C, Zheng Y, Li N, et al. A survey of graph neural networks for recommender systems: challenges, methods, and directions. ACM Trans Recomm Syst, 2023, 1: 1–51
45 Fan W Q, Ma Y, Li Q, et al. Graph neural networks for social recommendation. In: Proceedings of the World Wide Web Conference, 2019. 417–426
46 Fan W, Ma Y, Li Q, et al. A graph neural network framework for social recommendations. IEEE Trans Knowl Data Eng, 2020, 34: 2033–2047
47 Khalil E, Dai H J, Zhang Y Y, et al. Learning combinatorial optimization algorithms over graphs. In: Proceedings of Adv. Neural Inf. Process. Syst, 2017
48 Gasse M, Chételat D, Ferroni N, et al. Exact combinatorial optimization with graph convolutional neural networks. In: Proceedings of Adv. Neural Inf. Process. Syst, 2019
49 Pham T, Tran T, Phung D, et al. Column networks for collective classification. In: Proceedings of AAAI, 2017. 31: 2485-2491
50 Peng Y, Guo J, Yang C. Learning resource allocation policy: vertex-GNN or edge-GNN? Trans Mach Learn Comm Netw, 2024, 2: 190–209
51 Wang Q, Wang Y J, Xin B, et al. Dynamic weapon-target assignment optimization integrating deep reinforcement learning and graph neural networks. Control Theory & Appl, 2025, doi: 10.7641/CTA.2025.50065
52 Li Y, Zhang X, Zeng T, et al. Task placement and resource allocation for edge machine learning: a GNN-based multi-agent reinforcement learning paradigm. IEEE Trans Parallel Distrib Syst, 2023, 34: 3073–3089
53 Meng C, Tang M, Setayesh M, et al. Tackling resource allocation for decentralized federated learning: a GNN-based approach. IEEE Trans Mobile Comput, 2025, 24: 9554–9569
54 Luo Z, Bao Y, Wu C. Optimizing task placement and online scheduling for distributed GNN training acceleration in heterogeneous systems. IEEE ACM Trans Netwing, 2024, 32: 3715–3729
55 Hao X, She C, Lep Yeoh P, et al. Hybrid-task meta-learning: a GNN approach for scalable and transferable bandwidth allocation. IEEE Trans Wireless Commun, 2024, 23: 19820–19835
56 Ma Z Y, Xiong J, Gong H J, et al. Adaptive depth graph neural network-based dynamic task allocation for UAV-UGVs under complex environments. IEEE Trans Intell Veh, 2025, 10: 3573–3586
57 Wang J G, Xin B, Li G C. Combinatorial design of fast constructional algorithms for joint allocation off fire power and guidance resources. Sci Sin Inform, 2024, 54: 1458–1473
58 Luo J, Fei Z, Wang X, et al. GNN-based resource allocation for digital twin-enhanced multi-UAV radar networks. IEEE Wireless Commun Lett, 2024, 13: 3137–3141
59 Bogdanowicz Z R. A new efficient algorithm for optimal assignment of smart weapons to targets. Comput & Math Appl, 2009, 58: 1965–1969
60 Li C, Xin B, Wang D J, et al. Dynamic weapon target assignment based on deep Q network. In: Proceedings of 2023 42nd Chinese Control Conference (CCC), 2023. 1773–1778