

Constrained Scheduling Optimization of Improved LLH Resources for Engineering Management

Lijun Liu

School of Architecture and Civil Engineering,
Jinggangshan University, Ji'an, 343000, China.
E-mail: lijun.ecard068@silomails.com

Luxia Ouyang

School of Architecture and Civil Engineering,
Jinggangshan University, Ji'an, 343000, China.
Corresponding author: ouyangluxia@jgsu.edu.cn

(Received on March 19, 2025; Revised June 10, 2025; Accepted on July 10, 2025)

Abstract

To investigate how to maximize the benefits of engineering projects through effective scheduling strategies in the context of limited resources, this study develops a high-level heuristic strategy that leverages deep reinforcement learning algorithms within a hyper-heuristic framework, complemented by enhancements to low-level heuristic methods. Experimental results demonstrate that the proposed super-heuristic algorithm achieves excellent convergence quality and efficiency, reaching a minimum convergence value of 0.31 and requiring only 32.16 seconds for computation. The super volume and inverse generation distance produced by this algorithm are 0.906 and 0.254, respectively. Additionally, the algorithm exhibits strong performance in uniformity and breadth evaluations, making it adept at handling various scenarios in multi-skill, resource-constrained project scheduling problems. In the realm of engineering project scheduling, the proposed method yields the highest solution quality, achieving a coverage rate of up to 0.954. It also demonstrates low levels of constraint violation and significant solution set entropy during the solving process, satisfying the requirements for constrained scheduling optimization. Furthermore, the deviation between the optimized solution and the ideal shortest project duration is less than 0.15, indicating improved efficiency in project duration optimization. The algorithm's success rate in consistently finding feasible solutions across multiple runs exceeds 90%. This study provides effective scheduling strategies and optimization methods for engineering project managers, equipping them to tackle the challenges posed by resource constraints in engineering projects.

Keywords- Genetic programming algorithm, Resource constrained project scheduling problem, Engineering management, Multi-objective optimization, Heuristic approach.

1. Introduction

An efficient management model is of vital importance in engineering project management, as it not only helps to enhance project execution efficiency and ensure project quality, but also strengthens risk management capabilities, laying a solid foundation for project success. In today's complex and ever-changing business environment, the success of a project often depends on the coordinated cooperation of multiple factors, including resource allocation, task allocation and the optimization of management models. Through the rational allocation of resources, project managers can ensure that all important tasks are executed in a timely manner, thereby taking into account the overall progress and quality of the project. However, the current existing engineering management model still has some significant deficiencies, especially the excessive reliance on human resources and the overly high expectations for workers' self-management ability, which leads to problems such as low work efficiency and soaring labor costs during the implementation process. Meanwhile, as many projects fail to reasonably consider and evaluate the cost of material consumption, this not only leads to the waste of resources, but also often results in budget overruns, hindering enterprises from achieving the goal of economic benefits (Van Eynde et al., 2024).

To address these challenges, enterprises are obligated to prioritize and consistently refine the project management model for engineering projects. They should endeavor to enhance the efficiency of resource utilization, expedite the project completion timeline, and ultimately attain sustainable development (Guo et al., 2023). Against this background, the Resource Constrained Project Scheduling Problem (RCPSP) has gradually come into the view of researchers. With refined scheduling strategies, it provides an effective solution for project management. RCPSP refers to the reasonable arrangement of project execution sequence and resource allocation under the premise of limited resources, in order to maximize resource utilization and minimize cost expenditure. Therefore, it provides engineering Project managers with a new tool to deal with the challenges of resource constraints, enabling more optimized Project Scheduling (PS) plans, improving project execution efficiency, and performing more efficiently in resource utilization at the same time (Bahroun et al., 2024; Zhang et al., 2024). However, the complexity of resource characteristics in actual engineering PS leads to many limitations of the traditional RCPSP model. The Multi-skill RCPSP model (MS-RCPSP) introduces the concept of skills and establishes a skill matching relationship between tasks and resources, which has stronger applicability. However, research on the super-heuristic algorithm (HHA) for solving MS-RCPSP remains insufficient. How to design appropriate heuristic rules to better adapt to the characteristics of MS-RCPSP still requires in-depth exploration (Vanhoucke and Coelho, 2024).

To address the challenge of resource constrained scheduling optimization in engineering management, this study utilizes the Deep Reinforcement Learning (DRL) algorithm as the High-Level Heuristic (HLH) strategy for HHA and optimizes the Low-Level Heuristic (LLH) method, thereby achieving the optimization of HHA. The innovation point of the research lies in the application of the improved DRL algorithm to the hyper-heuristic algorithm for the first time. By using the Deep Q-network (DQN) as the HLH strategy, it can effectively enhance the optimization ability of resource-constrained PS. The research also conducts systematic optimization for LLH strategies, combining Iterative Local Search (ILS) and Variable Neighborhood Descent (VND), and enhances the effect of local search through dynamic adjustment and multiple perturbation strategies. Finally, a MS-RCPSP model was studied and constructed, taking into account the diversity of resources and skill matching, to expand the application scope of the traditional RCPSP model.

The structure of the research content is mainly divided into four sections. The second section of the research reviews the literature related to MS-RCPSP, explores the research progress in this field in recent years, and discusses the application of different algorithms in engineering management. The third section will introduce in detail the HHA architecture designed for the MS-RCPSP problem, including HLH and the improved LLH, and explain its innovativeness. Next, the fourth section will demonstrate the rationality of the experimental design and parameter settings, analyze the experimental results, and verify the improvement of the model performance through statistical significance tests. Finally, the article will summarize the research conclusions, explore its practical application value for engineering project management, and explore the possible directions for future research.

2. Related Works

RCPSP is a classic problem in the field of project management, widely used in various project management such as logistics, production workshops, software development, etc. A large number of researchers have carried out study on it. Golab et al. (2023) proposed a method based on Evolutionary Convolutional Neural Network (ECNN) to optimize project duration. This method utilized a serial scheduling generation scheme, learned and automatically selected priority rules based on project parameters, and scheduled project activities. Compared with traditional metaheuristic methods, there was no need to generate multiple solutions or populations. The experiment outcomes indicated that this method outperformed the standard metaheuristic method in regard to performance (Golab et al., 2023). To explore the PS problem under multi-

skill resource constraints, Snauwaert and Vanhoucke (2023) proposed a new classification scheme to systematically sort out related issues, and developed a multi skill RCPSP data generation program to construct multiple artificial datasets. By comparing the benchmark dataset with the introduced practical case, and combining genetic algorithm and CPLEX12.6 to solve the mixed integer linear programming Equation, the influence of skills and labor scale on completion time was experimentally analyzed, verifying the practicality of the dataset and the complexity of multi skill PS (Snauwaert and Vanhoucke, 2023).

Regarding the replacement subgraph problem in RCPSP, the complex relationship between nesting and linking. Servranckx et al. (2024) tested and adjusted the satisfiability solver method to address specific constraints of RCPSP surrogate sub-graphs. Through computational experiments on small and large instances, the competitiveness of the algorithm with existing metaheuristic algorithms was verified, and its performance was compared with that of an exact mathematical solver. The results emphasized the applicability of the satisfiability solver (Servranckx et al., 2024). Pass Lanneau et al. (2024) proposed the concept of anchor solution as a new robust optimization method for RCPSP to solve the uncertainty of processing time. They reviewed the graph model for budget uncertainty and designed a dedicated heuristic algorithm. Numerical experiments showed that the proposed method performed well in terms of efficiency, verifying its effectiveness in addressing anchor robustness and adjustable robust RCPSP. Armas et al. (2024) first explored the application of quantum annealing technology to RCPSP. By analyzing various mixed integer linear programming Equations, the Equation with the highest quantum bit efficiency was selected and transformed into a quadratic unconstrained binary optimization model. The model was solved using a D-wave quantum annealer and compared with classical solvers. The results showed that quantum annealing had great potential in small and medium-sized instances.

There are also many studies on heuristic algorithms. In order to handle the issue of missile path planning in remote air to ground strikes, Xu et al. (2025) proposed the Monte-Carlo Hyper Heuristic Algorithm (MCHHA) to address the limitations of traditional metaheuristic algorithms and the difficulty in evaluating the performance of HHA. This method constructed a set of 18 LLH algorithms and designs high-order strategies to predict the validity of LLH algorithms using a discrete state action reward table. The results showed that the algorithm could effectively adapt to different path planning scenarios, and simulation comparisons verified its superiority and effectiveness. Deliktaş and Aydin (2023) aimed to minimize the number of workstations by maintaining a constant cycle time for simple assembly line balancing problems. The Artificial Bee Colony-Based Hyper Heuristic Algorithm (ABC-HHA) was used to solve the problem, optimizing the efficiency and idle rate of the assembly line. Parameter control and calibration were performed using the IRACE method, and model performance on benchmark problems was tested. The experiment outcomes indicated that the artificial bee colony super heuristic algorithm performed superior in 61 problem instances, surpassing the existing cutting-edge methods.

To solve the complex logistics scheduling problem of prefabricated modular cabin units on cruise ships, Li et al. (2024) designed a Self-learning Hyper Heuristic Algorithm Based on Genetic Algorithm (SLHHA-GA). This algorithm effectively avoided the infeasible solution problem of metaheuristic algorithms by optimizing LLH strategies and revealing the relationships between decision stages. Multiple numerical experiments and actual enterprise cases showed that the algorithm performed well in solving this problem, significantly improving optimization capabilities and stability, and reducing transportation time by 37% in actual cases. Zhao et al. (2023) proposed an Estimation of Distribution Algorithm-Based Hyper Heuristic (EDA-HH) for the scheduling problem of distributed assembly hybrid idle replacement flow shop. This algorithm used 10 heuristic rules to search the solution space and integrated distribution estimation algorithm as a high-level strategy to control LLH sequences. Development capabilities were enhanced through product and job destruction, as well as construction procedures. The experiment showed that the

algorithm outperformed its competitors statistically and effectively solved 810 large-scale problem instances.

Nabawy et al. (2024) proposed a neural risk model (INRM) method for infrastructure projects in Egypt to identify and evaluate the key risk factors affecting construction costs, with a focus on the probability of occurrence (POO) and the impact on costs (IOC). The research results showed that the coefficient of determination (R^2) of the developed INRM model revealed the best results, among which the best architectures in the training and test datasets were 0.872 and 0.777 respectively. Fan (2025) proposed an innovative hybrid soft computing method combining fuzzy logic and deep neural networks to obtain the prediction results of the relationship between key construction factors and construction quality in response to the significant impact of the project construction stage on the overall success. The prediction result of this model reached 96.08%. This research result enhanced project management practices and establishes effective construction management strategies, thereby improving the construction quality of the project. Zhang et al. (2024) put forward a novel approach rooted in graph convolutional neural networks and attention mechanisms. This method is specifically designed to dynamically capture and integrate project-related relationship information, thereby addressing complex relationship challenges in modern power grid project management that traditional methods struggle to model accurately. The experiments of this model on the public recommendation system dataset verified the effectiveness of the proposed method in the management of power grid projects.

Although the above-mentioned research has made certain progress in the RCPSP problem, there are still some deficiencies. Many studies focus on the single application of specific algorithms or technologies, lacking comprehensive consideration of scenarios constrained by multi-skill resources. Moreover, when dealing with dynamic changes and complex constraint conditions, the effects have not met expectations. Furthermore, traditional methods often lack the potential to effectively integrate machine learning techniques, which may lead to insufficient adaptability to complex problems. For example, although the existing hyper-heuristic algorithms have been applied to some extent, they have not been fully combined with the emerging deep learning and reinforcement learning methods to produce stronger optimization effects.

This research method introduces the improved DRL algorithm as a HLH strategy and optimizes the LLH algorithm. It can not only enhance the intelligent decision-making ability in the scheduling process, but also effectively improve the adaptability to multi-skill resource constraints. This comprehensive technical route enables the model to enhance scheduling accuracy and efficiency while dealing with resource constraints and time pressure, emphasizing the importance of establishing dynamic and flexible scheduling strategies in the field of engineering management. The introduction of the research method aims to fill the gap of the lack of comprehensiveness and adaptability in the current literature, in order to deal with the complex problems encountered in the scheduling of multi-skill resource-constrained projects. The integration of the enhanced DRL technology with the hyper-heuristic algorithm facilitates the derivation of more flexible and efficient scheduling strategies. This synergy significantly improves the model's responsiveness to dynamic and complex scenarios, thus propelling the state-of-the-art development in this research domain.

3. Design of HHA Architecture under MS-RCPSP

In the context of MS-RCPSP, the design of HHA architecture was carried out. The study first utilized an improved DRL algorithm to implement HLH optimization research, and then improved the design of LLH.

3.1 Design of Multi-Objective MS-RCPSP Model

MS-RCPSP is a complex scheduling problem in project management, which is an extension of RCPSP.

MS-RCPSP requires a reasonable arrangement of task execution time and sequence while meeting the priority relationships and resource constraints between project activities, while considering the various skills and proficiency levels of resources, in order to achieve the goal of project optimization (Qian et al., 2023; Wang et al., 2023). The engineering project includes j tasks T_j , which has a priority relationship between different tasks. The start and duration of the tasks are s_j and d_j , respectively. The project contains k resources R_r , which involve n skills S_k during the implementation process. Therefore, the cost C_{jr} of using resources to complete project tasks is calculated as shown in Equation (1).

$$C_{jr} = d_j \cdot c_r \quad (1)$$

In Equation (1), c_r represents the wage for the use of resources. Firstly, based on the characteristics of engineering management, a mathematical model of MS-RCPSP is constructed, and the optimization objective of MS-RCPSP is set to minimize project time and cost. The objective function is shown in Equation (2).

$$\begin{cases} \min_{c_{\max}} F_j, j \in J \\ \min_{Cost} = \sum_j x_j C_{jr} \end{cases} \quad (2)$$

In Equation (2), F_j represents the completion time of the task T_j , determined by s_j and d_j . In addition, the multi-objective MS-RCPSP model also needs to satisfy a series of constraints. The task processing constraints are shown in Equation (3).

$$\sum_r x_{jr} Q_{rk} \geq y_{jk} l_{jk}, \forall j \in J, k \in S, r \in R \quad (3)$$

In Equation (3), x_{jr} represents the decision variable that determines whether a task is allocated to resource R_r . Q_{rk} indicates the proficiency level of resource R_r mastering skill S_k . y_{jk} represents decision variable that determine whether resources should execute tasks. l_{jk} indicates the level of proficiency in the skills required to complete a task. Meanwhile, the constraint condition satisfies Equation (4).

$$\begin{cases} \sum_r x_{jr} = 1, \forall j \in J, r \in R \\ \sum_j y_{jn} \leq 1, \forall j \in J, r \in R, t \end{cases} \quad (4)$$

As shown in Equation (4), the number of tasks processed by different resources at once is less than 1, but all tasks should be processed. In addition, the task processing process meets the temporal requirements, that is, the next task can only be executed after the previous task is completed. To solve MS-RCPSP, appropriate algorithm design is required. The algorithm used in the study is HHA, which manages and manipulates a series of LLH methods through an HLH strategy to achieve optimization in the solution space. HHA belongs to a class of metaheuristic algorithms that do not rely on domain specific knowledge and use HLH strategies to guide the selection and combination of LLH to solve complex optimization problems (Cao et al., 2025; Dang et al., 2025). The basic structure of HHA is in **Figure 1**.

In **Figure 1**, the choice of HLH strategy is important as it immediately affects the search efficiency and resolution quality of HHA. Therefore, the study adopts the DQN in the DRL algorithm as the HLH strategy.

DQN is an algorithm that combines deep learning and reinforcement learning to solve reinforcement learning problems based on value functions. The core idea of reinforcement learning is that agents learn optimal strategies through interaction with the environment to maximize a certain cumulative reward. Markov Decision Process (MDP) is the standard mathematical model for reinforcement learning problems. MDP consists of state S , action a_t , strategies π , reward r , and the returns. The state represents the environmental state in which the intelligent agent is located and has Markov properties. The probability P of a certain state occurring is calculated using Equation (5).

$$P[s_{t+1} | s_t] = P[s_{t+1} | s_1, \dots, s_t] \quad (5)$$

In Equation (5), t represents time. Action a_t is the action that an intelligent agent can choose to execute in a given state. Strategy π is the mapping of intelligent agents from state to action. Reward is the immediate feedback obtained by an intelligent agent from the environment after performing an action, and the return is the sum of accumulated rewards, usually considering future rewards and discounting them. The calculation process is shown in Equation (6).

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \quad (6)$$

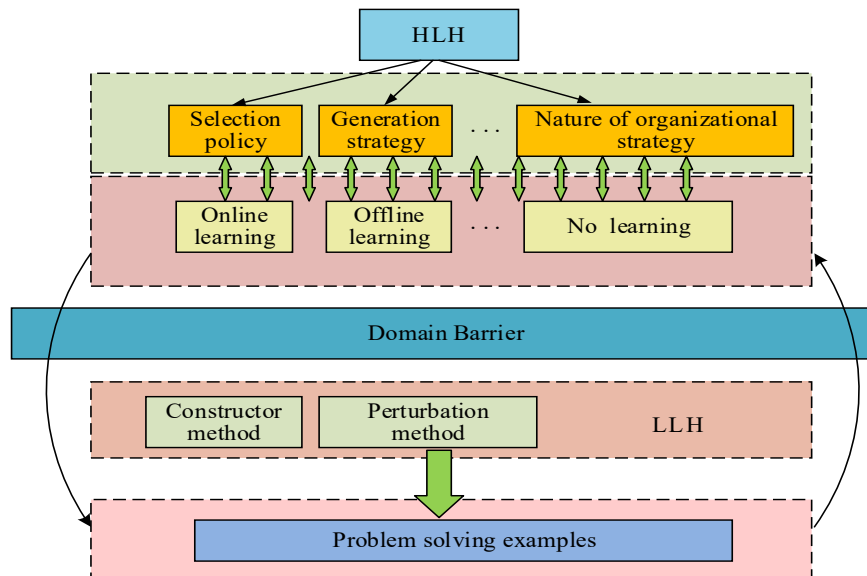


Figure 1. Schematic diagram of the basic framework of HHA.

In Equation (6), γ represents the discount factor. i represents time increment. Reinforcement learning, within the MDP framework, estimates the state action value function (AVF) by interacting with the environment, and updates strategies accordingly to maximize expected returns. The AVF, also known as the Q function, is expressed in Equation (7).

$$Q(s, a) = E[r + \gamma \max_a Q(s', a') | s_t = s, a_t = a] \quad (7)$$

In Equation (7), s' and a' represent the state and action at the next moment. The innovation points from Equation (5) to Equation (7) are mainly reflected in the combination of reinforcement learning and deep

learning to handle complex state spaces efficiently, thereby enhancing the intelligence of the decision-making process. These Equations construct a dynamic learning framework by defining the core elements of the MDP, including state transition probability, action selection strategy and Q-value function, enabling agents to continuously optimize strategies in the interaction with the environment. Especially the introduction of the Q-value update Equation (7) optimizes the value assessment of different states and actions. By using the deep neural network to approximate the Q function, the learning rate and accuracy of the model in the high-dimensional state space are greatly improved. The core of DQN is to use deep neural network approximation function $Q(s, a)$ to handle complex state spaces. The DQN algorithm framework is shown in **Figure 2**.

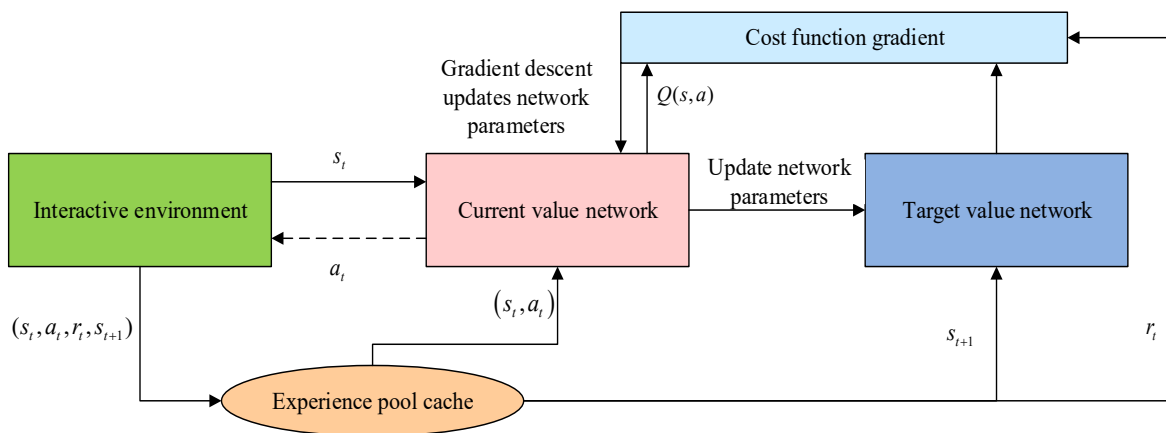


Figure 2. Schematic diagram of DQN algorithm framework.

As shown in **Figure 2**, DQN uses an experience replay buffer to store the experience data of the agent's interaction with the environment, namely the state action reward next state quadruple. Meanwhile, DQN introduces the target network to calculate the target Q value, thereby stabilizing the training process. The Q function update is shown in Equation (8).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \quad (8)$$

In Equation (8), α represents the hyperparameter learning rate. The action selection strategy is the basis for the DQN algorithm agent to select actions in a given state, which directly affects the convergence of the DQN algorithm. The traditional DQN algorithm adopts the ϵ -greedy strategy to balance exploration and utilization, but the exploration part of this strategy has strong randomness and lacks specificity, resulting in a significant decrease in the exploration efficiency of the agent. Moreover, random exploration fails to comprehensively cover all states. This is because it often results in agents being incapable of accessing isolated regions, thereby causing them to overlook potentially optimal actions and substantial rewards. In this regard, the study introduces the Upper Confidence Bound (UCB) algorithm to optimize action selection strategies. UCB is a heuristic search algorithm that considers both the average reward of known actions and exploration terms related to the number of times actions are selected, improving the probability of actions being explored. Therefore, UCB sets the UCB to be composed of the estimated average return and the exploration term, and the calculation of the UCB A_t is shown in Equation (9).

$$A_t = \arg \max_a \left[Q_t(a) + c \sqrt{\frac{\ln n}{N_t(a)}} \right] \quad (9)$$

In Equation (9), $Q_t(a)$ denotes the value estimation of the action. c represents positive real number. n indicates the total number of times an action is selected. $N_t(a)$ indicates the number of times the action was selected before time t . The innovation points from Equation (8) to Equation (9) are mainly reflected in the improved action selection strategy, in which the UCB algorithm is introduced to optimize the exploration efficiency of the agent in DRL. Equation (8) shows the update of the Q value and emphasizes the revenue-based learning mechanism. Meanwhile, the UCB strategy in Equation (9) enhances the agent's exploration ability in the solution space by balancing the average benefit of known actions with the possibility of exploring unknown actions. This action selection based on UCB not only accelerates the convergence process, but also ensures that the agent can search the solution space more comprehensively, thereby avoiding potential local optimal solutions and improving the performance of the algorithm in complex scheduling scenarios.

3.2 HHA Architecture Design Based on Improved LLH

After completing the construction of the multi-objective MS-RCPSP model and optimizing the DQN algorithm, LLH optimization design is carried out for HHA. The HHA architecture for multi-objective MS-RCPSP is shown in **Figure 3**.

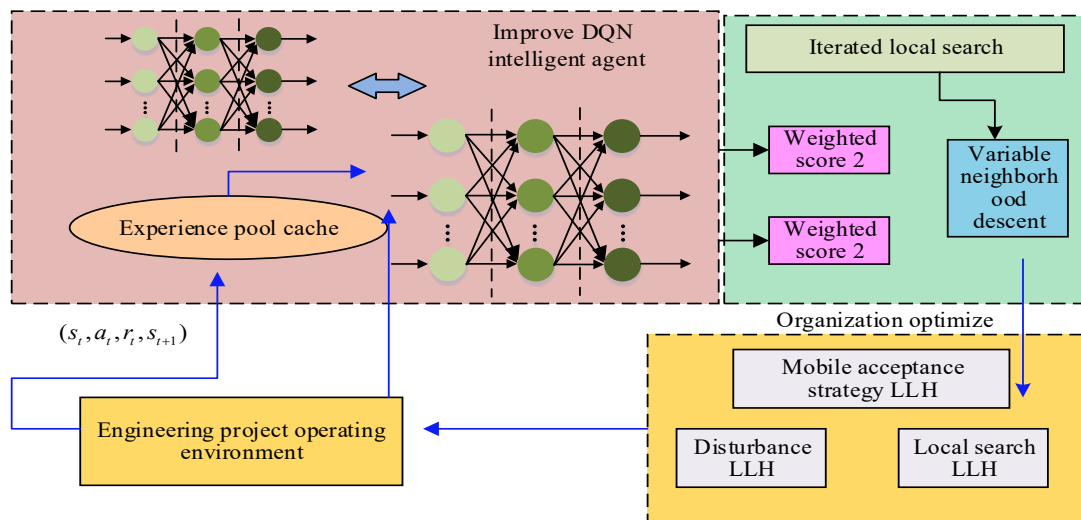


Figure 3. Schematic diagram of HHA architecture for multi-objective MS-RCPSP.

As shown in **Figure 3**, the study introduced ILS to optimize the organization of LLH and built an automatically designed HHA architecture. ILS is an exploratory local search method, whose core is to add perturbations to the local optimal solution obtained by local search, and then perform local search again. Meanwhile, by using perturbation strategies to escape from local optima and utilizing local search to search in a new solution space, a more optimal solution can be found (Bozorgi et al., 2023; Salem et al., 2023). ILS starts local search from the initial solution, obtains the local optimal solution, and then perturbs it to obtain a new solution. Then it starts a new local search from the new solution and finds a local optimal

solution again. Finally, based on the judgment strategy, it evaluates the new local optimal solution and decides whether to accept it as the new optimal solution. This process is repeated continuously until the termination condition is met. The search process and automatic design framework of ILS are shown in **Figure 4**.

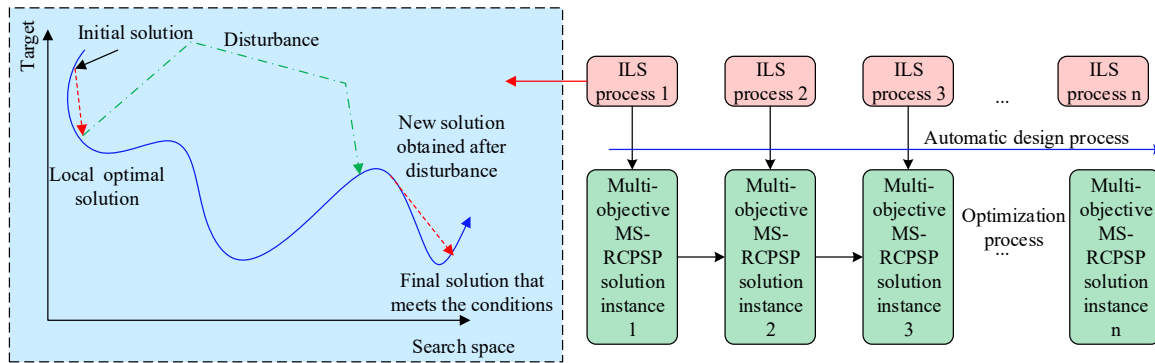


Figure 4. Schematic diagram of ILS search process and automatic design framework.

As shown in **Figure 4**, the ILS process mainly includes three main stages: disturbance, local search, and movement acceptance. The entire ILS process repeats these three stages continuously, forming different ILS cycles. At the end of each cycle, the automatic design framework determines whether to start a new round of ILS cycle based on the set restart conditions. The commonly used judgment criteria are shown in Equation (10).

$$p(\text{Accept}(\hat{s}, s', \text{history})) = \begin{cases} 1 & \text{if } f(s') < f(\hat{s}) \\ \exp\left(-\frac{f(s') - f(\hat{s})}{T}\right) & \text{otherwise} \end{cases} \quad (10)$$

In Equation (10), $p(\text{Accept}(\hat{s}, s', \text{history}))$ represents the probability function in simulated annealing. f represents the objective function. The innovation point of Equation (10) lies in its introduction of the mechanism of dynamic restart conditions, thereby enhancing the flexibility and adaptability of ILS. This Equation sets an evaluation criterion for the probability function in the simulated annealing algorithm, enabling it to automatically determine whether a new ILS loop needs to be initiated when a specific number of iterations or solution quality standards are reached. This dynamic judgment strategy promotes the effectiveness of the parsing process, allowing the algorithm to be more targeted and adaptable during the solution process, so as to continuously explore new solution spaces, thereby improving the quality of the overall solution and the search efficiency. Among them, in the local search stage, the algorithm used in the study is VND. VND is a refinement method that alternately searches in the solution space based on different neighborhood structures to find the local optimal solution of the problem. The VND algorithm first selects different neighborhood structures, constructs initial solutions, and sets termination criteria for the algorithm. Then, starting from the first neighborhood structure, the algorithm performs local search based on the search strategy defined by that structure. When a better solution cannot be found in the current neighborhood, it switches to the next neighborhood structure to continue searching. If a better solution is found within a certain neighborhood structure, the current solution will be updated. The operation process of VND is shown in **Figure 5**.

As shown in **Figure 5**, when the VND reaches the predetermined number of iterations or the quality of the solution meets a certain requirement, the algorithm terminates and outputs the optimal solution. Finally, the study investigated the LLH design of HHA architecture, including the design of Local Search LLH (LSLLH), Disturbance LLH (DLLH), and Mobile Acceptance Strategy LLH. LSLLH defines 11 types of neighborhoods, and LSLLH1 randomly inserts process codes from a string into other positions in the sequence. LSLLH2 selects the process code in the string and randomly changes it to another machine code. LSLLH3 adopts dual local search and combines the operations of LSLLH1 and LSLLH2 to change the encoding. The operation process of LSLLH4 and LSLLH5 is shown in **Figure 6**.

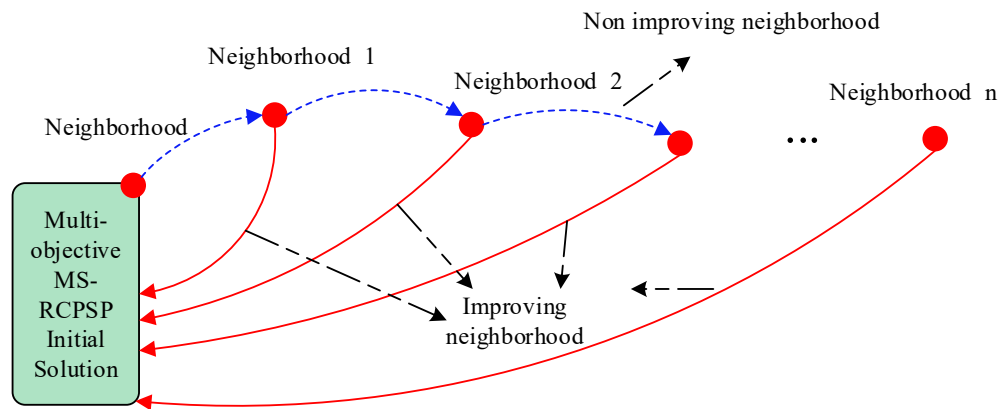


Figure 5. Schematic diagram of VND process.

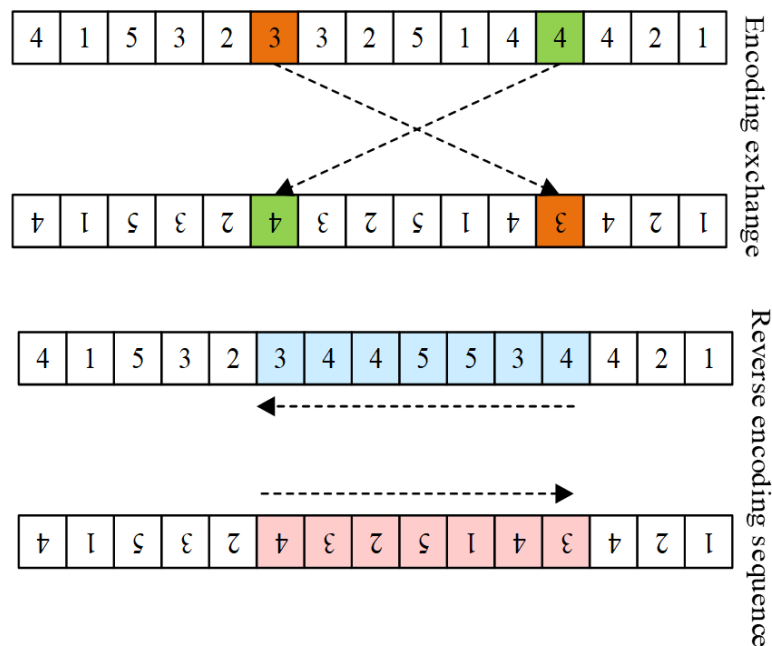


Figure 6. Schematic diagram of process code exchange and reverse operation.

As shown in **Figure 6**, LSLLH4 selects two process codes from the string and swaps them. LSLLH5 selects the process code in the string and flips it. LSLLH6 selects the process code in the string and adjusts the execution order of the process by moving the sub-sequence forward. LSLLH7 selects two process codes in the string and swaps them, while changing the corresponding machine codes. LSLLH8 changes the corresponding machine code simultaneously after operating LSLLH5. After operating LSLLH9 according to LSLLH6, the corresponding machine code is changed simultaneously. LSLLH10 moves the process of the machine with the highest workload to other optional machines. LSLLH11 repeatedly performs movement operations on the LSLLH2 neighborhood until the preset stopping condition is reached. 11 neighborhoods are used to support local search in VND algorithm. Six types of DLLH were set up in the study, and DLLH1 randomly scrambled any two process codes. DLLH2 and LSLLH4 operate similarly. DLLH3 operates similarly to LSLLH2. DLLH4 operates similarly to LSLLH7. DLLH5 requires randomly exchanging the process codes of two jobs in pairs. DLLH6 operates the same as LSLLH5. Finally, the study utilized simulated annealing to design the mobile acceptance strategy LLH.

The above content indicates that the combination of the UCB algorithm and DQN can demonstrate certain advantages in dynamic task scheduling and complex decision-making scenarios. The introduction of UCB enables agents to effectively balance exploration and utilization when choosing actions, optimizes the selection process of strategies, and realizes a more efficient search mechanism. By leveraging UCB, agents can not only make choices among known high-yield actions, but also intelligently explore unknown areas, providing opportunities for potential optimal solutions. This integration is capable of enhancing the convergence rate and solution quality of the DQN. It circumvents the issue of inadequate randomness inherent in conventional exploration strategies, thereby augmenting the algorithm's adaptability and efficiency within multi-skill resource-constrained PS scenarios. Consequently, the algorithm demonstrates heightened robustness and flexibility when confronted with complex problems.

4. Efficiency Evaluation and Application Effect Examination of Improved HHA Algorithm

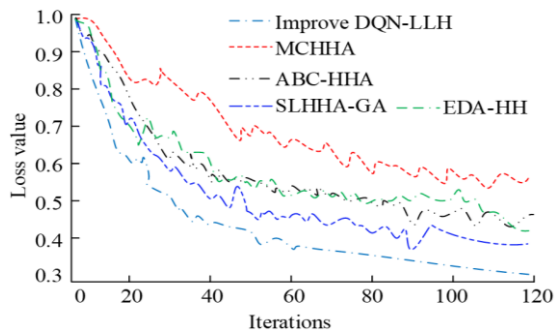
To confirm the validity of the improved HHA algorithm designed for solving multi-objective MS-RCPSP models, efficiency evaluation and application examination experiments were conducted, and the experimental results were discussed.

4.1 Improved HHA Algorithm Performance Testing

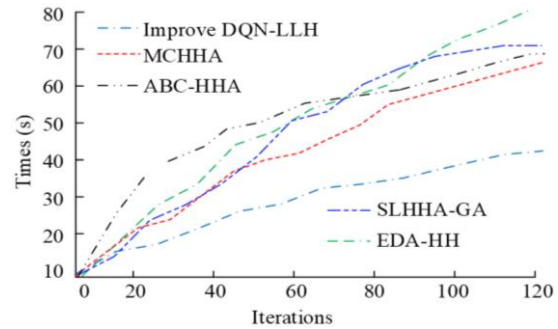
Firstly, a performance comparison analysis of the HHA algorithm was conducted. The experiment was based on the Windows 10 operating system, with an Intel Core i5-4200 4-core processor and a 2.50GHz clock speed. The image processor was GTX1080, the deep learning framework was Pytorch 1.7, and the programming language was Python 3.8. The dataset used in the experiment included intelligent Multi-objective PS Environment (iMOPSE) and PS Problem Library (PSPLIB). The iMOPSE dataset is a specialized dataset for multi-objective PS problems, which includes PS instances of various scales and complexities. The PSPLIB dataset covers different project sizes, resource types, and constraints, and can be extended to test the performance of the MS-RCPSP algorithm. The comparative algorithms included MCHHA from reference (Xu et al., 2025), ABC-HHA from reference (Deliktaş and Aydin, 2023), SLHHA-GA from reference (Li et al., 2024), and EDA-HH from reference (Zhao et al., 2023).

Firstly, the convergence characteristics and solving efficiency of different algorithms were compared, and the experimental results are shown in **Figure 7**. **Figure 7** shows the convergence trend of the proposed super heuristic algorithm under different iterations. In the experiment, with the increase of the number of iterations, the value of the objective function of the algorithm gradually decreased, showing good convergence. Especially within the first 50 iterations, the convergence speed was relatively fast, which indicated that the algorithm could quickly find a better solution in the initial stage. Next, although the

convergence rate slowed down somewhat, it remained stable and finally reached the lowest value of 0.31 after the 100th iteration. This trend not only proved the effectiveness of the algorithm, but also indicated that feasible optimization schemes can be found in a relatively short period of time, providing reliable support for the scheduling of engineering projects.



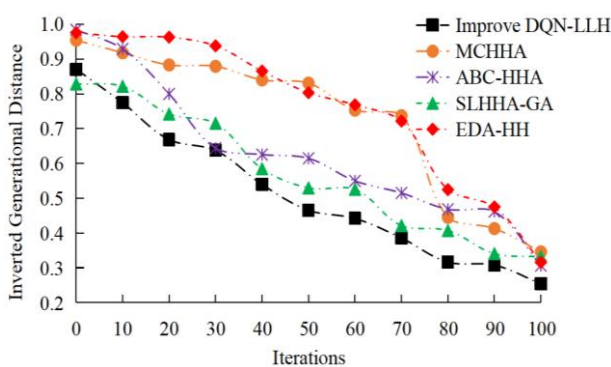
(a) Loss value comparison.



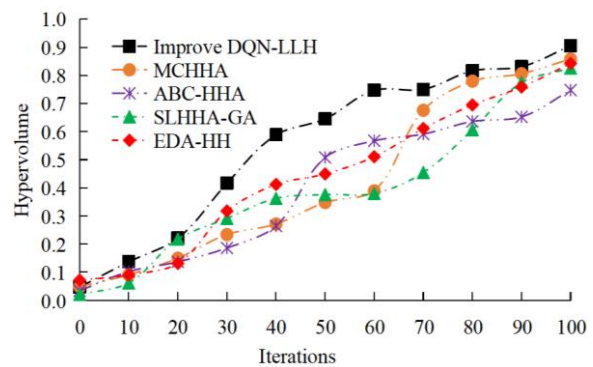
(b) Computational efficiency comparison.

Figure 7. Comparison of loss value and computational efficiency.

The results of Hypervolume (HV) and Inverse Generative Distance (IGD) for different HHAs are in **Figure 8**. The multi-objective optimization results shown in **Figure 8** demonstrate the balancing ability of the algorithm when solving multiple objective functions. In this figure, the horizontal axis represents different goals, while the vertical axis represents the corresponding optimization values. It can be observed that the super heuristic algorithm achieved a good trade-off between the two goals of HV and inverse generation distance. Especially in the index of HV (0.906), it indicated that the algorithm could find a relatively uniform distribution in a wide solution space. This indicated that through reasonable strategy design, the algorithm not only focused on a single objective but also effectively took into account multiple other objectives, which was conducive to achieving overall optimization.



(a) IGD comparison



(b) HV comparison

Figure 8. Comparison of IGD and HV indicators for various algorithms.

The evaluation results of the distribution of various HHA solutions are in **Figure 9**. Improved DQN-LLH performed well in Uniformity Performance (UP) evaluation, with a maximum UP value of 0.945 after 50

iterations of solving. However, the UP values of MCHHA, ABC-HHA, SLHHA-GA, and EDA-HH only reached 0.844, 0.769, 0.828, and 0.824. The improved DQN-LLH algorithm designed for the MS-RCPSPP problem did not tend towards certain specific regions during the search process, and could explore the solution space more comprehensively. In **Figure 9(b)**, there were significant differences among different HHAs in solving the Spacing Performance (SP) evaluation. The improved DQN-LLH had the smallest SP distribution range, with a minimum SP value of only 0.131. The improved DQN-LLH performed well in various situations in the MS-RCPSPP problem. In terms of the uniformity index, the high score performance reflected that the algorithm could effectively avoid the clustering phenomenon when exploring the solution space and ensure the diversity of the solutions. This feature is particularly important in PS with severe resource constraints, avoiding the decline in solution quality caused by local optimality.

To verify the influence analysis of experimental parameters on the model, the study set different parameters and made comparisons. The selection of the learning rate ranged from a smaller 0.01 to a larger 0.5 to explore the impact of different update speeds on the convergence of the model and ensure a more detailed and comprehensive search process. The setting of the discount factor (γ) took into account the importance of future rewards and was set at 0.9 and 0.95, enabling the algorithm to effectively balance long-term benefits and short-term benefits when optimizing the current decision. The variation of the exploration strategy parameter (c) further enhanced the flexibility of the model in different choices, enabling the agent to achieve a more effective balance between known results and new opportunities. The results are shown in **Table 1**. **Table 1** shows the influence of different parameter Settings on the model performance, which is specifically reflected in indicators such as the final loss value, inverse generation distance (IGD), HV, and uniformity performance (UP). When the learning rate varied from 0.01 to 0.1, the performance of the model improved significantly. Especially when the discount factor (γ) increased to 0.95 and was combined with appropriate exploration strategy parameters (c), both the final loss value and IGD decreased significantly, indicating that the convergence of the model and the quality of the solution improved. For example, when the learning rate was set to 0.01, γ was 0.95 and c was 3, the final loss value dropped to 0.265 and the IGD reached 0.328, which indicated that the effect of this parameter combination was optimal. Meanwhile, the changes in p -values also showed significant differences in model performance under different parameter Settings. The p -values of multiple combinations were lower than 0.05, indicating that the influence of these Settings on model performance was statistically significant. These analysis results provided a strong empirical basis for the subsequent optimization of model parameters and the improvement of scheduling efficiency.

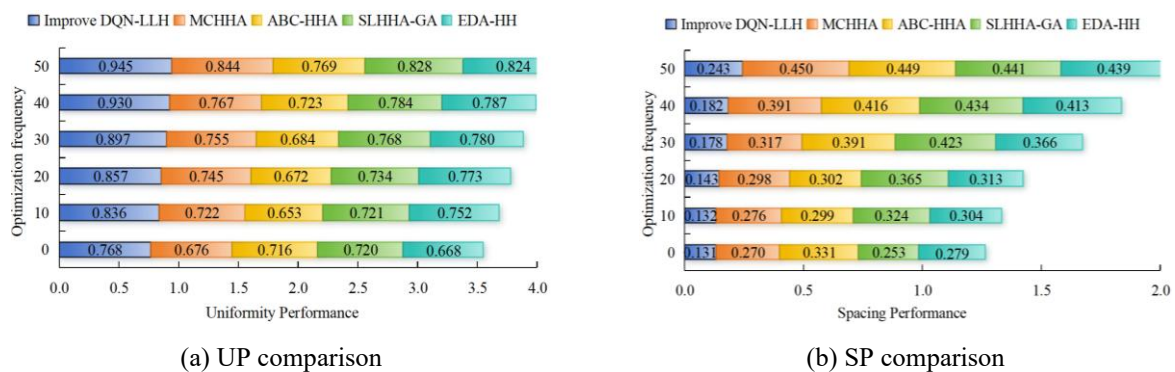


Figure 9. Comparison of SP and UP values of algorithms.

Table 1. The influence results of different parameters on the model.

Parameter setting	Final loss value	IGD	HV	UP	<i>p</i> value
Learning rate=0.01, $\gamma=0.9$, $c=1$	0.320	0.356	0.885	0.912	0.032
Learning rate=0.01, $\gamma=0.9$, $c=2$	0.290	0.342	0.895	0.925	0.028
Learning rate=0.01, $\gamma=0.9$, $c=3$	0.270	0.335	0.900	0.930	0.025
Learning rate=0.01, $\gamma=0.95$, $c=1$	0.315	0.360	0.880	0.910	0.030
Learning rate=0.01, $\gamma=0.95$, $c=2$	0.285	0.345	0.893	0.920	0.026
Learning rate=0.01, $\gamma=0.95$, $c=3$	0.265	0.328	0.905	0.935	0.023
Learning rate=0.1, $\gamma=0.9$, $c=1$	0.340	0.374	0.875	0.905	0.042
Learning rate=0.1, $\gamma=0.9$, $c=2$	0.315	0.352	0.888	0.912	0.040
Learning rate=0.1, $\gamma=0.9$, $c=3$	0.300	0.340	0.895	0.925	0.039
Learning rate=0.1, $\gamma=0.95$, $c=1$	0.335	0.365	0.883	0.915	0.041
Learning rate=0.1, $\gamma=0.95$, $c=2$	0.310	0.351	0.878	0.918	0.037
Learning rate=0.1, $\gamma=0.95$, $c=3$	0.280	0.330	0.892	0.930	0.036
Learning rate=0.5, $\gamma=0.95$, $c=1$	0.380	0.401	0.865	0.895	0.055
Learning rate=0.5, $\gamma=0.9$, $c=2$	0.355	0.385	0.872	0.902	0.053
Learning rate=0.5, $\gamma=0.9$, $c=3$	0.330	0.375	0.878	0.910	0.050
Learning rate=0.5, $\gamma=0.95$, $c=1$	0.370	0.395	0.870	0.900	0.052
Learning rate=0.5, $\gamma=0.95$, $c=2$	0.345	0.382	0.865	0.905	0.047
Learning rate=0.5, $\gamma=0.95$, $c=3$	0.320	0.370	0.872	0.912	0.045

4.2 Analysis of the Engineering Application Effect of Improving HHA Algorithm

Taking a large-scale infrastructure construction project in China as an example, the project included multiple sub-tasks and activities such as design, procurement, construction, and commissioning, each of which required specific types of resources and skills to complete. The project mainly consisted of 20 main tasks, each with a duration ranging from a few days to several months, depending on the complexity of the tasks and the required resources. The project was equipped with sufficient manpower, equipment, and materials, subject to constraints such as timing, resources, skills, and others.

The comparison of C-metric and Knee driven dissimilarity (KD) metrics for various algorithms is shown in **Figure 10**. In **Figure 10(a)**, the improved DQN-LLH performed well in the task scheduling process with a significant advantage over other methods in terms of the C-metric metric, with a maximum value of 0.954. High coverage means that the research design method could cover more solution space, increasing the possibility of accessing more solutions. As shown in **Figure 10(b)**, the algorithm performed well in KD values, with a minimum value of only 0.167, and the solved set could cover the inflection points of possible solutions well. The results showed that this algorithm had a prominent efficiency performance when optimizing the PS problem. The method proposed in the research could greatly improve the response speed of decision-makers and help project managers make more reasonable scheduling decisions in a timely manner.

The analysis of the Constraint Violation Degree (CVD) and Solution Set Entropy (SSE) of different algorithms on the MS-RCPSP problem is shown in **Figure 11**. As shown in **Figure 11(a)**, the improved DQN-LLH had the smallest CVD value during the solving process, fluctuating within the range of 0.1-0.25. The CVD values of other algorithms were all higher than 0.25. CVD measures the degree to which the solution set generated by the algorithm violates constraint conditions, indicating that the design of the study satisfies constrained scheduling optimization. Meanwhile, as shown in **Figure 11(b)**, the SSE values of the improved DQN-LLH fluctuate at the highest level, indicating good diversity in the solution set. The results showed that during the entire optimization process, the values of constraint violations remained within a very low range, proving that it had a high degree of reliability when dealing with the scheduling problem of resource constraints. This good control of constraints enabled the final output solution to not only meet the feasibility requirements but also conform to various restrictive conditions in the actual project execution.

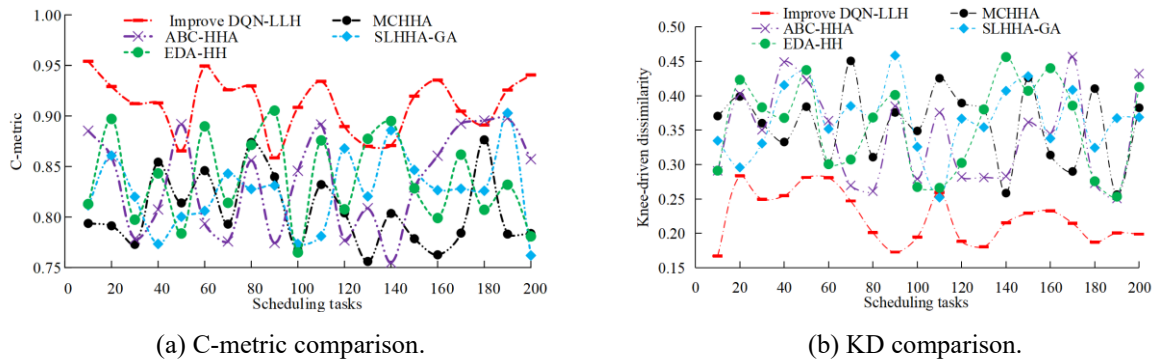


Figure 10. Coverage metrics and KD for different algorithms.

Finally, the project schedule deviation and algorithm success rate of different algorithms in engineering management scheduling were analyzed, as shown in **Figure 12**. As shown in **Figure 12(a)**, the deviation between the scheduling results of the improved DQN-LLH algorithm designed for research and the ideal shortest duration was the smallest, less than 0.15. In contrast, the deviation fluctuation range of other methods was too large, and the algorithm's effectiveness in optimizing project duration was insufficient. In **Figure 12(b)**, the success rate of the improved DQN-LLH algorithm was the highest, reaching over 0.90. The proportion of successfully finding feasible solutions in multiple runs was relatively high, indicating good stability. The results showed that the set optimization strategy and algorithm design had good robustness and stability, and could maintain a high problem-solving ability in numerous complex scheduling instances. This result not only highlighted the potential of the algorithm in practical applications, but also provided project managers with confidence when facing the ever-changing project environment.

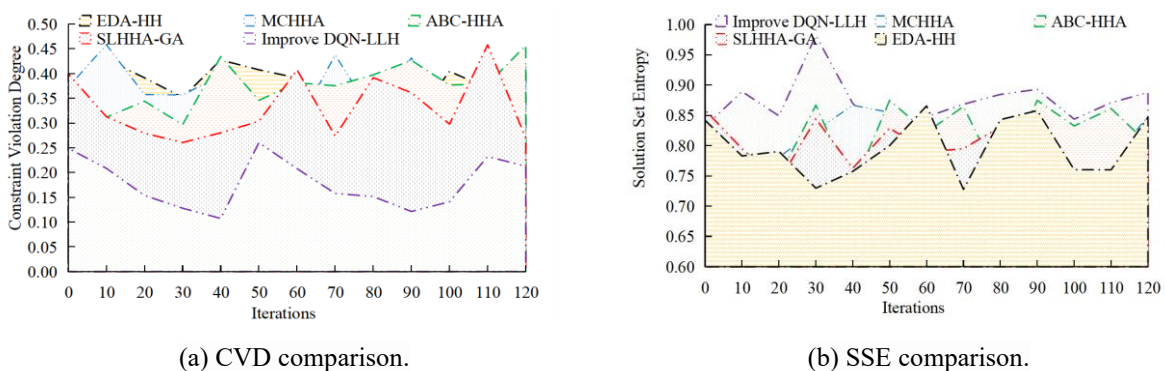


Figure 11. Comparison of CVD and SSE of different algorithms.

To further verify the universality of the improved DQN-LLH model, the study analyzed the model through actual engineering application scenarios. The experiment selected the cross-sea tunnel project in the Guangdong-Hong Kong-Macao Greater Bay Area as the verification case. This case included 120 construction tasks (earthwork excavation, support structure, mechanical and electrical installation, etc.), which had typical characteristics of multi-stage collaboration and dynamic allocation of multi-skill resources. The comparison algorithms included Non-dominated Sorting Genetic Algorithm III (NSGA-III), Multi-Objective Evolutionary Algorithm based on Decomposition with Differential Evolution (MOEA/D-DE), and Deep Reinforcement Learning Algorithm for Scheduling (DRLA-Schedule). The evaluation indicators were analyzed by using the project duration deviation rate (ΔT), total cost deviation rate (ΔC), convergence time and task scale expansion rate. The results are shown in **Table 2**.

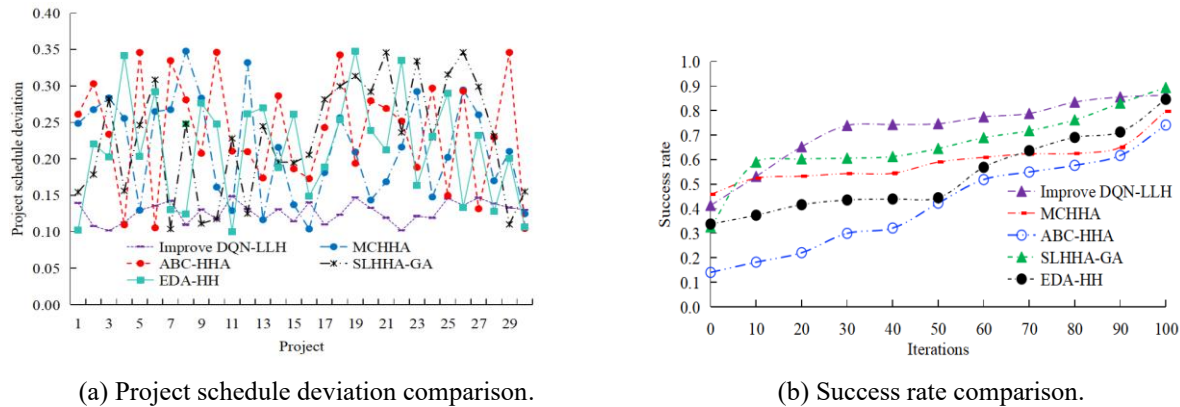


Figure 12. Comparison between project duration deviation and scheduling success rate.

Table 2 shows the performance of different algorithms in the practical application of cross-sea tunnel projects in the Guangdong-Hong Kong-Macao Greater Bay Area. The results showed that the improved DQN-LLH model performed well in both ΔT and ΔC , which were 1.28% and 2.15% respectively. Both were significantly superior to other comparison algorithms, such as NSGA-III (ΔT was 4.79%, ΔC was 5.62%) and MOEA/D-DE (ΔT was 3.95%, ΔC was 4.33%). Furthermore, the improved DQN-LLH also demonstrated significant advantages in terms of convergence time, taking only 856 seconds, which was lower than that of NSGA-III (1245 seconds) and MOEA/D-DE (1520 seconds), showing its potential in more efficient resource allocation and rapid scheme generation. Meanwhile, its task scale expansion rate reached 12.7%, further highlighting the scalability of the improved DQN-LLH in dealing with complex PS. This indicated that the model had strong universality and practicability, providing an effective scheduling solution for practical engineering.

Table 2. The application effects of different algorithms in real cases.

Algorithm	ΔT (%)	ΔC (%)	Time (s)	Scalability (%)
CPLEX	0	0	21850	N/A
Improve DQN-LLH	1.28	2.15	856	12.7
NSGA-III	4.79	5.62	1245	34.8
MOEA/D-DE	3.95	4.33	1520	28.3
DRLA-Schedule	2.14	3.02	932	18.5

5. Conclusion

To achieve optimal scheduling of the project, improve project execution efficiency, reduce costs, and ensure timely delivery, the improved DQN was studied as the HLH strategy and LLH was optimized to construct the MS-RCPSPP super heuristic algorithm. The experiment outcomes indicated that the improved DQN-LLH algorithm exhibited the fastest convergence speed, with a computation time of 32.16 seconds. In addition, the lowest IGD value converged to 0.254, the maximum HV value reached 0.906, the highest UP value reached 0.945, and the SP distribution range was the smallest. In practical application, the C-metric and KD indicators of the algorithm performed well, with a maximum C-metric value of 0.954 and a minimum KD value of only 0.167. Meanwhile, the algorithm performed well in solving the MS-RCPSPP problem with CVD and SSE values, satisfying constrained scheduling optimization. In the end, the algorithm designed for research had the smallest deviation between the scheduling results and the ideal shortest duration, and had the best stability in multiple runs. This study aimed to optimize resource scheduling in engineering management, reduce project costs and time consumption, improve overall project

efficiency, and promote sustainable development in the construction industry. However, the research design failed to fully consider various uncertain factors that might be encountered in actual engineering projects, such as the sudden shortage of resources and the dynamic adjustment of task priorities, which might affect the applicability and robustness of the model. In future research work, the study will enhance the adaptability and accuracy of the model by introducing more advanced machine learning techniques and big data analysis methods. For example, by combining graph neural networks with reinforcement learning, the complex relationships and dynamic changes among resources can be explored, thereby providing more real-time decision support. Additionally, the impact of incorporating uncertain factors during the scheduling process will be taken into account. These uncertain factors encompass changes in task priorities and fluctuations in resource availability. The objective is to augment the practicability and robustness of the model.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research was funded by the Science and Technology Project of Jiangxi Province Education Department, grant number GJJ2201658.

AI Disclosure

The author(s) declare that no assistance is taken from generative AI to write this article.

Reference

- Armas, L.F.P., Creemers, S., & Deleplanque, S. (2024). Solving the resource constrained project scheduling problem with quantum annealing. *Scientific Reports*, 14(1), 16784. <https://doi.org/10.1038/s41598-024-67168-6>.
- Bahroun, Z., As'ad, R., Tanash, M., & Athamneh, R. (2024). The multi-skilled resource-constrained project scheduling problem: systematic review and an exploration of future landscapes. *Management Systems in Production Engineering*, 32(1), 108-132. <https://doi.org/10.2478/mspe-2024-0012>.
- Bozorgi, S.M., Yazdani, S., Golsorkhtabamiri, M., & Adabi, S. (2023). A hyper-heuristic approach based on adaptive selection operator and behavioral schema for global optimization. *Soft Computing*, 27(22), 16759-16808.
- Cao, Y., Yu, J., Zhong, R., & Munetomo, M. (2025). Forecasting renewable energy and electricity consumption using evolutionary hyperheuristic algorithm. *Scientific Reports*, 15(1), 2565. <https://doi.org/10.1038/s41598-025-87013-8>.
- Dang, Y., Gao, X., & Wang, Z. (2025). A novel hyper-heuristic algorithm with soft and hard constraints for causal discovery using a linear structural equation Model. *Entropy*, 27(1), 38. <https://doi.org/10.3390/e27010038>.
- Deliktaş, D., & Aydin, D. (2023). An artificial bee colony based-hyper heuristic algorithm with local search for the assembly line balancing problems. *Engineering Computations*, 40(9/10), 2453-2482. <https://doi.org/10.1108/ec-02-2023-0075>.
- Fan, C.L. (2025). Predicting the construction quality of projects by using hybrid soft computing techniques. *Computer Modeling in Engineering & Sciences*, 142(2), 1995-2017. <https://doi.org/10.32604/cmes.2025.059414>.
- Golab, A., Gooya, E.S., Alfalou, A., & Cabon, M. (2023). A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): a new approach. *Decision Science Letters*, 12(2), 225-238. <https://dx.doi.org/10.5267/j.dsl.2023.2.002>.
- Guo, W., Vanhoucke, M., & Coelho, J. (2023). A prediction model for ranking branch-and-bound procedures for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 306(2), 579-595.

- Li, J., Dong, R., Wu, X., Huang, W., & Lin, P. (2024). A self-learning hyper-heuristic algorithm based on a genetic algorithm: a case study on prefabricated modular cabin unit logistics scheduling in a cruise ship manufacturer. *Biomimetics*, 9(9), 516. <https://doi.org/10.3390/biomimetics9090516>.
- Nabawy, M., & Mohamed, A.G. (2024). Risks assessment in the construction of infrastructure projects using artificial neural networks. *International Journal of Construction Management*, 24(4), 361-373. <https://doi.org/10.1080/15623599.2022.2156902>.
- Pass-Lanneau, A., Bendotti, P., & Brunod-Indrigo, L. (2024). Exact and heuristic methods for anchor-robust and adjustable-robust RCPSP. *Annals of Operations Research*, 337(2), 649-682. <https://doi.org/10.1007/s10479-023-05537-6>.
- Qian, L., Zhang, H., Qiu, J., Zhang, X., Fouad, H., & Altameem, T. (2023). Mobile multiple sink path planning for large-scale sensor networks based on hyper-heuristic artificial bee colony algorithm. *Journal of Nanoelectronics and Optoelectronics*, 18(3), 329-337. <https://doi.org/10.1166/jno.2023.3400>.
- Salem, R.B., Aimeur, E., & Hage, H. (2023). A multi-party agent for privacy preference elicitation. *Artificial Intelligence and Applications*, 1(2), 82-89. <https://doi.org/10.47852/bonviewaia2202514>.
- Servranckx, T., Coelho, J., & Vanhoucke, M. (2024). A genetic algorithm for the resource-constrained project scheduling problem with alternative subgraphs using a Boolean satisfiability solver. *European Journal of Operational Research*, 316(3), 815-827. <https://doi.org/10.1016/j.ejor.2024.02.041>.
- Snauwaert, J., & Vanhoucke, M.A. (2023). Classification and new benchmark instances for the multi-skilled resource-constrained project scheduling problem. *European Journal of Operational Research*, 307(1), 1-19. <https://doi.org/10.1016/j.ejor.2022.05.049>.
- Van Eynde, R., Vanhoucke, M., & Coelho, J. (2024). On the summary measures for the resource-constrained project scheduling problem. *Annals of Operations Research*, 337(2), 593-625. <https://doi.org/10.1007/s10479-023-05470-8>.
- Vanhoucke, M., & Coelho, J. (2024). A matheuristic for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 319(3), 711-725. <https://doi.org/10.1016/j.ejor.2024.07.016>.
- Wang, Z., Liu, J., & Zhang, J. (2023). Hyper-heuristic algorithm for traffic flow-based vehicle routing problem with simultaneous delivery and pickup. *Journal of Computational Design and Engineering*, 10(6), 2271-2287. <https://doi.org/10.1093/jcde/qwad097>.
- Xu, S., Huang, Z., Bi, W., & Zhang, A. (2025). A Monte Carlo hyper-heuristic algorithm with low-level heuristics reward prediction for missile path planning. *The Journal of Supercomputing*, 81(2), 374. <https://doi.org/10.1007/s11227-024-06771-w>.
- Zhang, J., Li, L., Demeulemeester, E., & Zhang, H. (2024). A three-dimensional spatial resource-constrained project scheduling problem: model and heuristic. *European Journal of Operational Research*, 319(3), 943-966.
- Zhang, Y., Bai, G., Gao, Z., Zhu, P., & Li, S. (2024). Modeling long-and short-term project relationships for project management systems. *IEEE Access*, 12(1), 72242-72251. <https://doi.org/10.1109/access.2024.3402448>.
- Zhao, F., Zhu, B., & Wang, L. (2023). An estimation of distribution algorithm-based hyper-heuristic for the distributed assembly mixed no-idle permutation flowshop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(9), 5626-5637. <https://doi.org/10.1109/tsmc.2023.3272311>.



Original content of this work is copyright © Ram Arti Publishers. Uses under the Creative Commons Attribution 4.0 International (CC BY 4.0) license at <https://creativecommons.org/licenses/by/4.0/>