IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 April 2024, accepted 20 April 2024, date of publication 29 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394510

RESEARCH ARTICLE

Scheduling Model for Prefabricated Component Assembly, Production, and Transportation Stage Based on GA for Prefabricated Buildings

ZHIPENG HUO[1], XIAOQIANG WU[1], AND TAO CHENG[2]
[1]School of Architecture Engineering, Yan'an University, Yan'an 716000, China
[2]Design Department, China-shun International Engineering Design Company Ltd., Xi'an 716000, China

Corresponding author: Zhipeng Huo (hzp13399119900@163.com)

This work was supported in part by Yan'an Science and Technology Plan Project, Application and Research of Prefabricated Building, under Grant SL2022SLZDCY-005; in part by Yan'an University 2023 Research Special Project, Experimental Study on the Performance of Coal Gangue Insulation Material in Assembly Building in Northern Shaanxi Province, under Grant 2023JBZR-001; and in part by Yan'an Science and Technology Plan Project, Application and research of polypropylene fiber coal gangue insulation material in prefabricated low-carbon buildings, under Project 2023SLZDCY-206.

ABSTRACT With the development of prefabricated buildings in China, the demand for prefabricated components is also increasing. The construction schedule of prefabricated components has heterogeneity and timeliness, which makes the traditional scheduling models not applicable. In order to control the construction process and reduce costs, research is conducted on controlling the construction process of prefabricated components in prefabricated buildings. This study divides the construction process into three stages according to the construction characteristics of prefabricated buildings. The scheduling models of these three stages are established, namely assembly, production, and transportation stages scheduling models. The scheduling model of the three stages is related to each other through the duration constraints. In addition, an improved genetic algorithm is developed to solve the scheduling model of the assembly stage. Then an improved particle swarm optimization is designed to solve the scheduling model in the production and transportation stages. The results show that the minimum duration of the assembly phase was 8 days. The duration and cost of the production phase cannot be minimized at the same time. The minimum carbon emission duration and transportation cost in the transportation phase are 93.8 hours and 22516 yuan, respectively. The improved genetic algorithm tended to flatten out after nearly 180 iterations. The maximum running time of the improved particle swarm algorithm on the training set is 4.23s, the maximum hyper volume is 0.736, and the maximum anti generation distance is 2.35×10^{-3}. The scheduling models of different stages and corresponding solving algorithms are effective and provide technical support for the construction process control of assembly parts. The technical contribution of this study is to optimize the genetic algorithm based on weed invasion algorithm and improve the local search ability of genetic algorithm. Then, the differential evolution algorithm is used to improve the particle swarm optimization algorithm and continuously generate new particles to replace the optimal position.

INDEX TERMS Prefabricated buildings, prefabricated components, genetic algorithm, scheduling model, particle swarm optimization.

I. INTRODUCTION

Prefabricated buildings have the advantages of fast construction speed, saving labor and materials, high quality, and green environmental protection, making them the focus of future development in the construction industry. In prefabricated buildings, various resources used during the construction process should be reasonably integrated and then an optimal construction scheduling plan should be obtained [1], [2], [3]. The Prefabricated Component (PC) construction process

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

control is the core work of prefabricated building construction scheduling. Regarding the scheduling problem of prefabricated building construction, the current research mainly focuses on the production, transportation, and on-site assembly of PC. In solving Multi-Objective Optimization (MOO) problems, common methods include conventional mathematical methods such as weighted sum method, objective programming method, and ε- constraint method. However, these methods also have certain shortcomings, such as relying on the experience of decision-makers, pre-determining the expected values of each Objective Function (OF), and being sensitive to the shape of the search space [4], [5]. With the advancement of computer technology, more intelligent optimization algorithms are being used to solve construction scheduling models for prefabricated buildings. In recent years, an improved Grey Wolf intelligence algorithm is proposed to optimize the prefabricated production scheduling. Some papers have also proposed a radial basis function fuzzy logic neural network algorithm to solve resource scheduling problems in prefabricated building construction [6], [7]. However, these algorithms also have certain shortcomings, such as being prone to getting stuck in local optima and taking longer computation time. Therefore, in order to control the PC construction process in prefabricated buildings, shorten the construction period, reduce costs, and overcome the tendency of the scheduling models to fall into local optimality, different scheduling models are constructed from the three stages of the PC construction process, namely, the assembly stage scheduling model, the production stage scheduling model and the transportation stage scheduling model. In addition, the Invasive Weed Optimization (IWO) is used to improve the Genetic Algorithm (GA) for solving the scheduling model in the assembly stage. The differential evolution algorithm is used to improve the Particle Swarm Optimization (PSO) to solve the scheduling model in the production and transportation stages. The research aims to construct a scheduling model for PC in prefabricated buildings at different stages, improve the performance of scheduling model solving, assist in the scheduling of PC assembly, production, and transportation on construction sites, shorten the construction period, and reduce costs. There are two main innovations in the research. The first combines IWO and GA. The second combines differential evolution algorithm and PSO. The study has four parts. The first part reviews prefabricated building scheduling models. The second part constructs the scheduling model for different stages and designs the solving algorithm. The third part analyzes the scheduling model performance at different stages. The fourth part summarizes the conclusions, shortcomings, and future prospects.

II. RELATED WORKS

With the support of national policies, China has made significant progress in the technology and research related to prefabricated buildings. More scholars are conducting research on prefabricated building scheduling models. Ruan et al. designed an improved model for the production scheduling of PC, taking into account resource constraints. There were 13 constraints in this model. Enterprise decision coefficients were added to the OF. The model had good applicability, which could well meet diverse production conditions and business needs [8]. Zhang et al. designed an elastic cost trade-off model for prefabricated buildings to plan the supply chain of PC in emergency situations. A multi-objective PSO was designed to solve the model. Before constructing this model, the elastic supply chain plan was designed. The model could effectively solve the elasticity problem of the supply chain during the planning phase [9]. Podolski et al. designed a corresponding production planning model to address the manpower allocation in the production of precast concrete components. Simulated Annealing (SA) and taboo search algorithm were combined to solve the model. This model took into account unexpected situations during the production process. The production scheduling model was effective, which could effectively allocate human resources in the production of prefabricated concrete components [10]. Du et al. designed a dynamic flow shop scheduling model for PC production to address the frequent dynamic demand fluctuations that occurred on construction sites. A multi-objective GA was constructed to solve the model. The model took into account demand fluctuations, such as early delivery and order cancellations. The results indicated that the model effectively planed production scheduling under demand fluctuations [11].

More researchers are using intelligent optimization algorithms to solve scheduling models. Gebreyesus et al. designed a deep reinforcement learning method to solve the workshop scheduling of dynamic jobs. The gate mechanism was introduced to regulate the learning feature flow. In addition, the study also designed a gate-based graph pooling mechanism. This method was superior to existing heuristic algorithms and deep reinforcement learning [12]. Zhang et al. designed a GA to solve the ship scheduling model. The calculation function of tidal level variation over time was constructed. Finally, the model performance was validated on real ship data. The constructed method was significantly more efficient than existing methods, which had good reusability [13]. Wu et al. constructed a dynamic scheduling strategy and heuristic algorithm from the perspective of dynamic priority to solve imaging task scheduling. The algorithm performance and strategy was validated through experiments. This algorithm had obvious advantages. It was significantly better than the comparison algorithm in terms of solving speed and accuracy [14]. Meng et al. designed a parameter optimized dynamic priority scheduling algorithm based on improved reinforcement learning. This algorithm improved the action step size and reward function of the reinforcement learning algorithm, and accelerated the online learning speed. The constructed algorithm reduced the number of iterations in reinforcement learning and optimized scheduling cost [15].

In summary, there is rich research on prefabricated building scheduling models and scheduling model solving algorithms. However, the current research also has certain shortcomings,

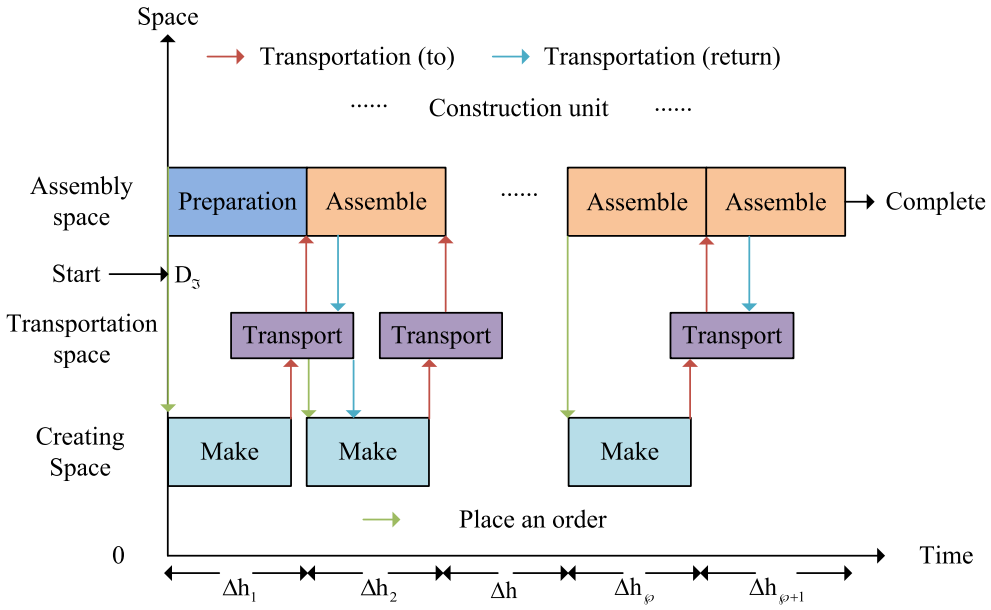


FIGURE 1. A diagram of the three phases working together.

such as some algorithms being prone to falling into local optima and taking longer computation time. Therefore, the scheduling model is established for the assembly, production, and transportation stages of PC in prefabricated buildings. An improved GA is developed to solve the assembly stage scheduling model. Then an improved PSO is also designed to solve the scheduling model for the production and transportation stages. The research aims to improve the performance of scheduling model solving algorithms and assist in the scheduling of PC assembly, production, and transportation on construction sites.

III. METHODOLOGY

In this chapter, based on collaborative work in different stages, a scheduling model for the assembly, production, and transportation stages in prefabricated buildings is constructed. The scheduling models for these three stages are associated with time constraints. For the scheduling model in the assembly stage, an improved GA is designed to solve it. The optimized PSO is applied to solve the scheduling model for the production and transportation stages.

A. SCHEDULING MODEL CONSTRUCTION

The PC construction control is the focus of prefabricated building construction. For the scheduling optimization of PC, it has three stages. Scheduling modeling is carried out for each stage. These three stages are assembly, manufacturing, and transportation. The optimization objectives for these three stages are determined in the study. The scheduling model is solved based on improved GA and PSO. The models are solved separately to obtain the optimal scheduling solution that meets the requirements of each stage. The three stages can be closely connected. To control the construction of PC, the construction process of PC is first divided. It mainly includes four aspects, namely determining the construction plan, producing PC, transporting and storing PC, and assembling PC [16]. Therefore, the construction process includes three stages: assembly, fabrication, and transportation. The collaborative work diagram of these three stages is shown in Figure 1.

In Figure 1, $\Delta\hbar$ represents the time required for a construction unit. $\wp$ represents the number of construction units. $\not{\lambda}_{\Im}$ represents the type and quantity of PC required for the $\Im$-th construction unit. The work positions, working hours, and scheduling nature of the three stages are not consistent, but they all serve the same construction goal. Before constructing scheduling models for different stages, a dimensionality reduction approach is adopted to lag the timeline and move different stages to the same time period for analysis. The optimization objectives vary in different stages. The assembly stage is to minimize the construction period to the greatest extent possible. The assembly period determines the deadline for the production and transportation stages. The optimization goal during the production phase is to minimize production cost and duration. The optimization goal during the transportation stage is to minimize transportation cost and minimize carbon emissions from transportation vehicles. The scheduling model of three stages is correlated through duration constraints. Therefore, the construction scheduling process mainly has three steps. The first step is to obtain the scheduling plan that meets the shortest construction period. The second step is to select the optimal scheduling plan for the production stage. The third is to select the optimal scheduling plan for the transportation stage. Before proceeding with the first step, the construction process and required resources for the assembly phase should be obtained first. The specific construction process during the assembly stage is shown in Figure 2 [17], [18].

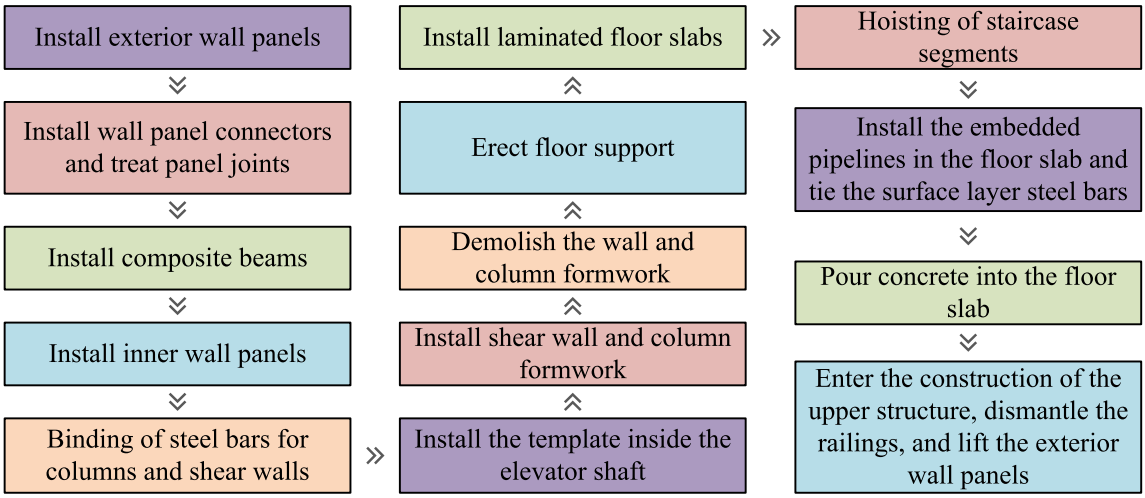


FIGURE 2. Specific construction processes during the assembly phase.

From Figure 2, the first step in the construction process during the assembly phase is to install the exterior wall panels, which are prefabricated insulation walls. The second step is to install the wall panel connectors and process the panel joints. The third is to install the composite beam. The fourth is to install the inner wall panel. The fifth is to tie the steel bars of the columns and shear walls. The sixth step is to install the formwork inside the elevator shaft. The seventh step is to install the shear wall and column formwork. The eighth step is to remove the wall column formwork. The ninth step is to set up the floor support, and the tenth step is to install the stacked floor slab. The eleventh step is to lift the staircase segment, and the twelfth step is to install the embedded pipelines in the floor slab and tie the surface layer steel bars. The thirteenth step is to pour concrete into the floor slab. The fourteenth step is to continue the construction of the previous structure, dismantle the railings, and lift the exterior wall panels. The OF in the assembly phase scheduling model is shown in equation (1).

$$\min S_{n+2} = \min \sum d_i \tag{1}$$

In equation (1), d_i represents the time required for activity i. $n+2$ represents the total number of activities. The tight preceding activity constraint relationship between activities i and j is shown in equation (2).

$$S_j + d_j \leq S_i, S_i \in P(i) \tag{2}$$

In equation (2), $P(i)$ represents the set of all preceding activities of activity j. S_j represents the start time of activity j. S_i refers to the start time of activity i. d_j refers to the time required for activity j. The resource constraint of the activity is shown in equation (3).

$$\sum_{i \in A(t)} rik \leq R_k \tag{3}$$

In equation (3), k represents the k-th renewable resource. rik refers to the amount of resource k used in activity i at t. rk is the amount of renewable resource k consumed. $A(t)$ refers to the collection of all ongoing activities at t. R_k represents the upper capacity limit of the renewable resource k. The discrete numerical constraint for time is shown in equation (4).

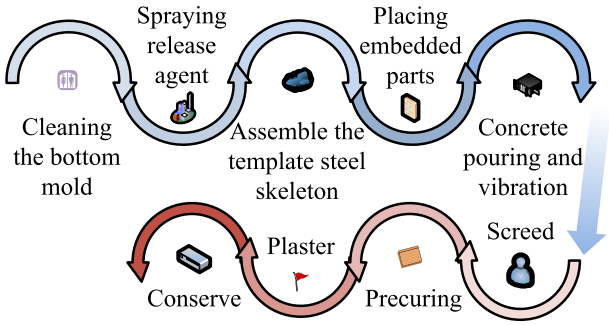


FIGURE 3. The production process of prefabricated components.

$$t = 0, 1, \ldots, Sn + 2 \tag{4}$$

The non-negative start time constraint for all activities is shown in equation (5).

$$Si \geq 0, \quad i = 1, 2, \ldots, n + 2 \tag{5}$$

After determining the assembly schedule for the standard layer, the production cycle required for each layer of assembled components can also be determined. The scheduling problem during the production phase can be summarized as follows. Under the constraint of project schedule, the production time is minimized and the resource cost is minimized. In addition, the scheduling problem during the production phase also is a multi-modal time cost trade-off model. Before constructing the optimal scheduling model for the PC production stage, the production process of PC is first analyzed, as shown in Figure 3 [19], [20].

From Figure 3, the first step in the production of PC is to clean the bottom mold first. The second step is to spray release agent, and the third step is to assemble the template

steel reinforcement framework. The fourth is to place the embedded parts, and the fifth is to pour and vibrate the concrete. The sixth is to scrape flat, and the seventh step is to perform pre-curing. The eighth step is to plaster, and the ninth step is to maintain. The OF in this stage is shown in equation (6).

$$\begin{cases} \min \sum_{j=1}^{n+2} d_{jm} \\ \min \sum_{j=1}^{n+2} \sum_{m \in M_j} c_{jm} x_{jm} + \sum_{j=1}^{n+2} r \cdot d_{jm} \end{cases} \tag{6}$$

In equation (6), the first OF represents minimizing the duration. The second OF represents cost minimization. x_{jm} represents the decision variable. m represents the activity selection mode. M_j represents the number of execution modes. r represents the indirect cost. c_{jm} represents the direct cost of the activity. The decision variable constraints of the production scheduling model are shown in equation (7).

$$\sum_{m \in M_j} x_{jm} = 1, \quad j = 1, \ldots, n+2 \tag{7}$$

The tight constraint relationship of the activity is shown in equation (8).

$$S_i + \sum_{m \in M_j} d_{jm} x_{jm} \leq S_j, \quad \forall j < i, j > \in A(t) \tag{8}$$

The completion time of the project satisfies the project deadline, as shown in equation (9).

$$S_{n+2} \leq \alpha \tag{9}$$

In equation (9), α represents the project deadline. The constraint that the start time of each activity is non-negative is shown in equation (10).

$$S_j \geq 0, \quad \forall j < V \tag{10}$$

In equation (10), V represents the range of j. The value constraint of decision variables is shown in equation (11).

$$x_{jm} \in \{0, 1\}, \quad \forall m < M_j \tag{11}$$

After completing PC production, they need to be transported. Transportation needs to be completed before the assembly of the next layer, which must also meet the time window restrictions on the construction site. Carbon emissions are generated during transportation. There is a positive correlation between carbon emissions and total transportation time. Therefore, the overall optimization goal during the transportation phase can be summarized as follows. Under the constraint of transportation time, transportation cost is minimized and carbon emissions are minimized. The OF in the transportation phase is displayed in equation (12).

$$\begin{cases} \min \text{cost} = \sum_{b=1}^{d} \sum_{g=1}^{h} r_o x_{bg} + x_{bg} L_g \\ \min \text{time} = \sum_{b=1}^{d} \sum_{g=1}^{h} Time_{bg} x_{bg} \end{cases} \tag{12}$$

In equation (12), the first OF refers to the minimum transportation cost. The second OF represents the shortest total time for carbon emissions. b represents the type of PC. d is the total types of PC. g represents the transport vehicle. h is the total transport vehicles. r_o represents the vehicle start-up cost. L_g represents the round-trip transportation cost of the vehicle. $Time_{bg}$ is the transportation round-trip time. x_{bg} represents the total transport vehicles carrying out transportation tasks. The round-trip transportation fee L_g for vehicles is shown in equation (13).

$$L_g = \beta \times \frac{Og \times \chi \times r_{\text{oil}}}{100} \ (g = 1, 2, \ldots, h) \tag{13}$$

In equation (13), β represents the transportation cost adjustment coefficient. χ represents the transportation distance. r_{oil} is the price of diesel. Og represents the fuel consumption of the transportation vehicle. The transportation round-trip time $Time_{bg}$ is shown in equation (14).

$$Time_{bg} = \varepsilon \frac{2\chi}{\ell_g} \ (b = 1, 2, \ldots, d; g = 1, 2, \ldots, h) \tag{14}$$

In equation (14), ε represents the speed adjustment coefficient. ℓ_g represents the transportation speed of the vehicle. The transportation time $Time_{ug}$ for PC is shown in equation (15).

$$Time_{ug} = Time_{bg} + t_z + t_y \tag{15}$$

In equation (15), t_z represents the loading time. t_y represents the unloading time. The constraint condition that each component's transportation volume should meet during the transportation phase is shown in equation (16).

$$\sum_{g=1}^{h} x_{bg} \geq \varphi_b \tag{16}$$

In equation (16), φ_b represents the required quantity of PC. The constraint on the total transport vehicles is shown in equation (17).

$$\sum_{b \in A(t)} x_{bg} \leq \gamma_g \tag{17}$$

In equation (17), γ_g refers to the total transport vehicles.

B. ALGORITHM DESIGN FOR SOLVING

In the previous chapter, a scheduling model is constructed and a solution algorithm is designed for the assembly, production, and transportation stages in the construction control of PC. In this chapter, the research will focus on model construction and algorithm design for the assembly, production, and transportation stages in the construction control of prefabricated building components. For the solution of the assembly stage scheduling model, the GA is selected for the study. The main process of GA is shown in Figure 4.

From Figure 4, the first step of the GA process is to create the initial population. The second is to evaluate the fitness value. The third is to determine whether the termination

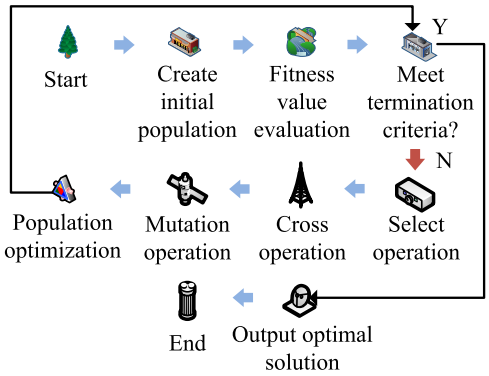


FIGURE 4. Specific construction processes during the assembly phase.

criterion is met. If it is met, the optimal solution is output. The process ends. Otherwise, it proceeds to the fourth step. The fourth is to perform a selection operation. The fifth is to perform a crossover operation. The sixth step is to perform a mutation operation. The seventh is to perform population optimization, and then return to the third step. The GA, as a heuristic search algorithm, can solve optimization problems, which has strong global search ability and low requirements for the OF. However, the GA also has poor local search ability and susceptibility to premature convergence, which cannot guarantee the convergence [21], [22]. Therefore, GA has been improved. The specific improvement measure is to introduce the IWO algorithm. The IWO, as a biomimetic swarm intelligence optimization algorithm, has simple structure, strong robustness, and powerful local and global search performance. The main steps of the IWO algorithm are population initialization, growth and reproduction, spatial diffusion, and competitive exclusion [23], [24]. The main process of the improved GA based on IWO is shown in Figure 5.

From Figure 5, the first to seventh steps of the GA process improved by the IWO are consistent with the standard GA process. The eighth step is to output the current optimal population. The ninth is to determine whether the maximum iterations have been reached. If it is determined to be negative, then seed reproduction is carried out. If it is determined to be positive, it continues to determine whether the termination criteria are met. The tenth step is to conduct spatial diffusion. The eleventh step is to determine whether the maximum population size has been reached. If it is negative, it returns to seed reproduction. If the result is positive, it competes for exclusion. The twelfth step is to determine whether the stop condition is met. If it is determined as no, the process returns to seed reproduction. Otherwise, the process continues to determine whether the termination criterion is met. Based on the characteristics of the assembly stage scheduling model, the study adopts natural number coding to uniformly number each process. To make the fitness value of each chromosome represent the quality of the solution, the study transforms the OF from minimization to maximization. The transformed maximization objective is shown in equation (18).

$$Fit = \frac{1}{f\ (x)} \tag{18}$$

In equation (18), $f\ (x)$ represents the minimum completion time. For the selection operation, the study adopts a roulette wheel strategy. The probability calculation of fitness value is shown in (19).

$$p\ (x_a) = \frac{f\ (x_a)}{\sum_{a=1}^{N} f\ (x_a)} \tag{19}$$

In equation (19), $f\ (x_a)$ represents the individual fitness value. N represents the number of individuals. The solution of scheduling models in the production and transportation stages both belong to multi-objective optimization problems. Therefore, it is necessary to select appropriate algorithms for solving. In multi-objective optimization problems, a non-dominated solution is usually obtained, which is the Pareto optimal solution set. When solving multi-objective optimization problems, the Pareto optimal solution is determined by comparing the advantages and disadvantages of the solutions, and then selecting these advantageous solutions [25], [26]. All surfaces formed by the Pareto optimal solution are called Pareto optimal surfaces, also known as Pareto frontiers. Therefore, appropriate algorithms should be used to evaluate and select the Pareto optimal solution. PSO is an intelligent algorithm that originates from the foraging behavior of birds, which is simple and fast [27]. The PSO algorithm explores problems according to the velocity and direction of any particle. The particle velocity in the PSO algorithm is shown in equation (20).

$$\begin{aligned}\partial_{CD}\ (\theta+1) = \partial_{CD}\ (\theta) + \phi_1 \cdot \omega_1 \cdot [\xi_{CD}\ (\theta) - \psi_{CD}\ (\theta)] \\ + \phi_2 \cdot \omega_2 \cdot [\Theta\ (\theta) - \psi_{CD}\ (\theta)]\end{aligned} \tag{20}$$

In equation (20), ∂_{CD} represents the running speed of the particle. θ is a particle. ϕ_1 and ϕ_2 are both acceleration constants. ω_1 and ω_2 are both random numbers between [0,1]. ξ_{CD} represents the optimal direction of a particle during its motion. ψ_{CD} refers to the particle direction. Θ refers to the best position for the entire group. The position update is shown in equation (21) [28].

$$\psi_{CD}\ (\theta+1) = \psi_{CD}\ (\theta) + \partial_{CD}\ (\theta+1) \tag{21}$$

The PSO is displayed in Figure 6.

The algorithm proposed in Figure 6 is particle swarm optimization. From Figure 6, the first step of the PSO algorithm is to initialize the parameters. The second is to update the position and velocity of particles. The third is to calculate the fitness value of particles. The fourth is to determine whether the current fitness value outdoes the individual's best direction value. If the judgment result is positive, the individual optimal is updated. The process enters the next step. Otherwise, it directly enters the next step. The fifth is to determine whether the current optimal fitness is better than the optimal position value of all particles in the whole. If it is determined to be positive, the global optimum is updated. The process enters the next step. Otherwise, it directly enters the next step. The sixth step is to determine whether the maximum iterations have been reached. If it is determined to be

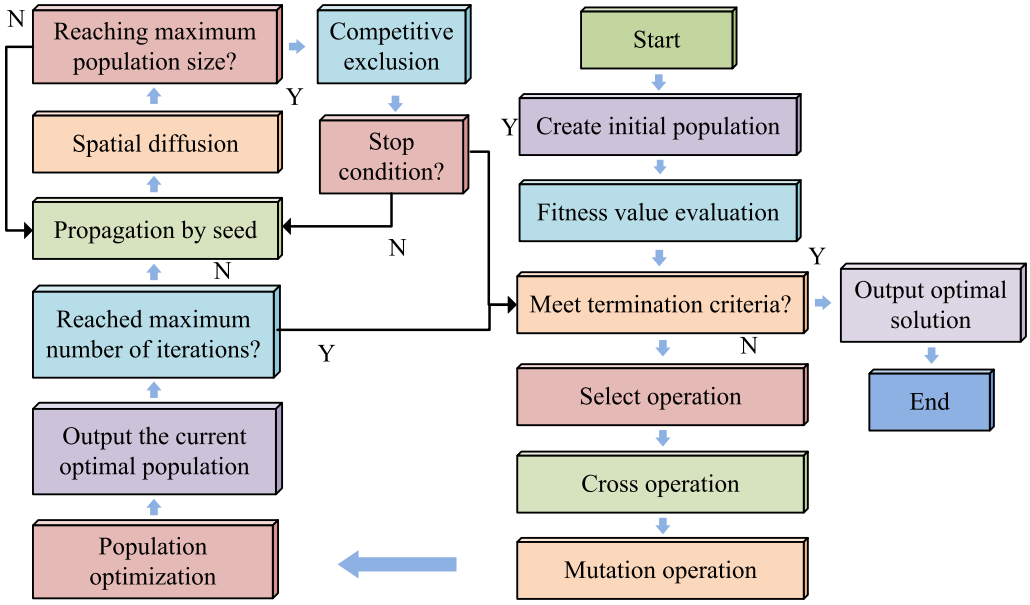


FIGURE 5. The main process of GA improved by IWO.

```
procedure PSO
    for each particle θ
        Initialize velocity ∂_CD and position ψ_CD for particle θ
        Evaluate particle θ and set Pbest θ =ψ_CD
    end for
    Gbest=min (Pbest θ )
    while not stop
        for θ =1 to N
            Update the velocity and position of particle θ
            Evaluate particle θ
            if fit(ψ_CD )<fit(Pbest θ )
                Pbest θ = ψ_CD ,
            if fit(Pbest θ ) < fit(Gbest)
                Gbest=Pbest θ ,
        end for
    end while
    print Gbest
end procedure
```

FIGURE 6. The main process of PSO algorithm.

yes, the process ends. Otherwise, it returns to the second step. The contribution of PSO algorithm is its ability to quickly find the optimal solution. Especially when facing complex situations, the algorithm has good performance in speed and accuracy, as well as high robustness and applicability. Differential Evolution (DE) can solve for the minimum value [29]. The individual generation in the DE algorithm is shown in equation (22).

$$\Phi_{EF} = \Phi_{\min} + \text{rand}\,(0, 1) * (\Phi_{\max} + \Phi_{\min}) \tag{22}$$

In equation (22), $\Phi_{\min}$ represents the lower limit of the search space. $\Phi_{\max}$ represents the upper limit of the search space. *rand* (0, 1) represents a random number between [0,1]. The crossover strategy of the DE algorithm is quadratic term crossover, as shown in equation (23) [30].

$$\Upsilon_{EF} = \begin{cases} \Omega_{EF} & if\ \left(rand_{\eta}(0, 1) \le CR\right) \\ \Phi_{EF} & otherwise\ (\eta = 1, 2, 3, \ldots, \mu) \end{cases} \tag{23}$$

In equation (23), Ω_{EF} represents the new vector of mutated individuals. *CR* is a crossover factor. μ represents the vector dimension. The PSO algorithm has fast convergence speed and high efficiency in the early stage. However, there is also a lack of local optima in the later stage. The main reason for this problem is that there are no new particles to replace the

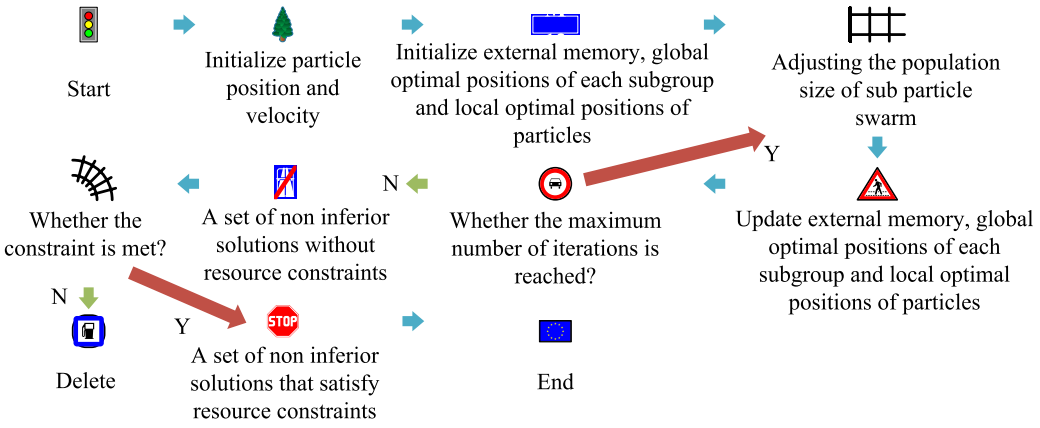


FIGURE 7. The main process of the DEPSO algorithm.

optimal position [31]. The DE algorithm can mutate particles, continuously iterate at the best position, and generate new particles to replace them. Therefore, PSO is combined with DE to solve the scheduling models for the production and transportation stages (DEPSO). The DEPSO is displayed in Figure 7.

The algorithm proposed in Figure 7 is a DEPSO algorithm that combines particle swarm optimization and differential evolution algorithms. From Figure 7, the first step in the DEPSO is to initialize the particle position and velocity. The second is to initialize the external memory and initialize the global optimal position and local optimal position of each sub-group. The third is to adjust the population size. The fourth is to update the external memory, the global optimal position of each sub-group, and the local optimal position of particles. The fifth is to determine whether the iterations have reached its maximum. If it has reached its maximum, it returns to the third step. Otherwise, it is placed in the non-inferior solution set without resource constraints. The sixth is to determine whether the constraint is met. If it is not, it will be directly deleted. If it is satisfied, it is placed in a non-inferior solution set that satisfies the resource constraints. The process ends. The contribution of the DEPSO algorithm is to overcome the local optimization problem faced by the PSO algorithm in the later stage, improve the overall and local retrieval ability of the PSO algorithm, select the optimal particles, and avoid the singularity of the population.

IV. RESULTS

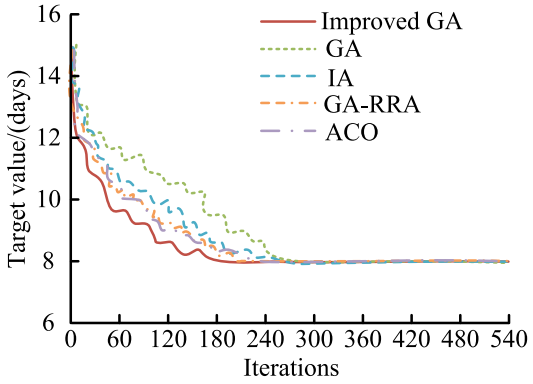
In this chapter, performance validation is conducted on scheduling models and solving algorithms at different stages. The experimental environment and implementation parameters are set. The optimal solutions for scheduling models at different stages are solved. In addition, the study also analyzes evaluation indicators and comparative algorithms.

A. IMPROVED GA ALGORITHM

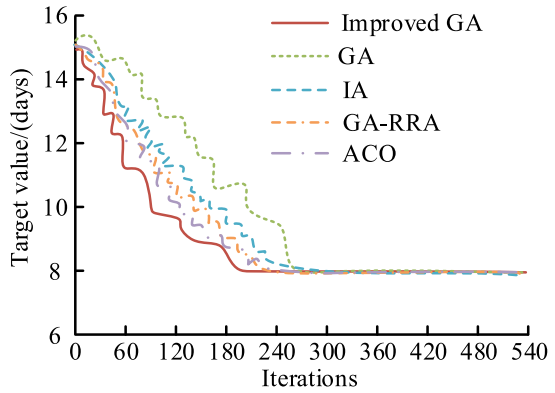
To verify the performance of different scheduling model solving algorithms, comparative experiments are conducted. For the performance verification of the improved GA, traditional GA, Immune Algorithm (IA), Hybrid Algorithm combining GA and Remaining Rectangle Algorithm (GA-RRA), and Ant Colony Optimization (ACO) are selected. The comparison indicators include the convergence speed and accuracy. The dataset is IEEE CEC 2019 MMOP, with 70% used as the training set and 30% used as the testing set. The operating system is Windows 11, with an Intel Core i5-12600 processor, a maximum memory of 128GB, 6 cores, and 12 threads. The convergence speed of different algorithms is shown in Figure 8.

From Figure 8 (a), the improved GA flattened out after nearly 180 iterations. The traditional GA, IA, GA-RRA, and ACO iterated nearly 250, 247, 210, and 235 iterations respectively, and then tended to flatten out. In Figure 8 (b), on the testing set, the improved GA tended to flatten out after nearly 197 iterations. The traditional GA, IA, GA-RRA, and ACO iterated nearly 266, 253, 220, and 240 iterations respectively, and then tended to flatten out. From this, the improved GA has faster convergence speed and better performance. The accuracy comparison of different algorithms is shown in Figure 9.

In Figure 9 (a), on the training set, the maximum values of the improved GA, traditional GA, IA, GA-RRA, and ACO were 99.53%, 92.31%, 94.54%, 97.65%, and 95.47%, respectively, while the minimum values were 98.37%, 91.82%, 92.13%, 97.65%, and 95.47%, respectively. The accuracy of the improved GA has always been higher than other comparative algorithms. From Figure 9 (b), on the testing set, the maximum accuracy values of the five algorithms were 99.71%, 92.68%, 95.01%, 97.47%, and 95.76%, respectively, while the minimum values were 98.26%, 92.03%, 92.57%, 95.07%, and 91.86%, respectively. The accuracy of the improved GA also has advantages on the testing set. From this, the improved GA has the highest accuracy and superior performance. To better validate the performance of the improved GA, other algorithms are selected for comparison in the study. The newly added comparative algorithm are the Distributed Co-evolutionary Memory Algorithm (DCMA) designed by Zhang et al. and the Multi-Objective Evolutionary Algorithm based on Fuzzy Correlation Entropy



(a) Comparison of convergence speed on the training set



(b) Comparison of convergence speed on the testing set

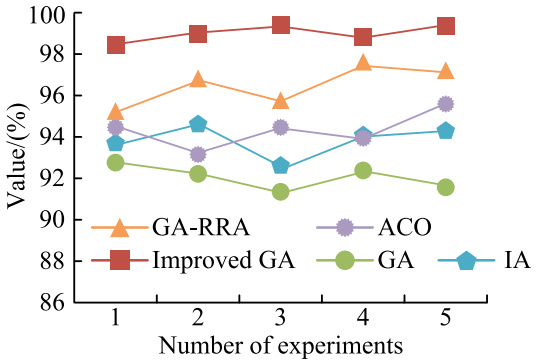
FIGURE 8. Comparison of convergence speeds of different algorithms.

TABLE 1. Comparison of MSE results of different algorithms on training and testing sets.

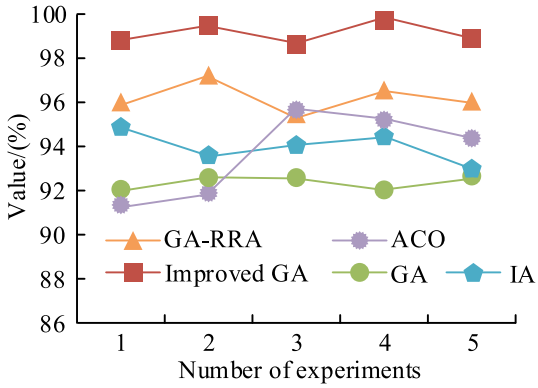
Algorithm	Training set Number of experiments				Testing set Number of experiments			
	1	2	3	4	1	2	3	4
GA	1.31	1.48	1.26	1.37	1.38	1.51	1.22	1.39
IA	1.27	1.17	1.24	1.29	1.31	1.15	1.26	1.18
GA-RRA	1.01	0.98	0.95	1.07	0.99	0.94	1.05	0.91
ACO	0.98	0.95	0.88	0.96	0.94	0.99	0.97	0.91
DCMA	0.84	0.76	0.82	0.73	0.81	0.72	0.75	0.69
MOE-FCE	0.77	0.69	0.71	0.66	0.64	0.70	0.65	0.68
Manuscript	0.35	0.31	0.28	0.33	0.27	0.32	0.24	0.35

(MOE-FCE) designed by Li et al. [32], [33]. The comparison indicator is Mean Squared Error (MSE). The comparison results are shown in Table 1.

From Table 1, the maximum and minimum values of MSE on the training set were 1.48 and 0.28, respectively, which appeared in the GA and the improved GA. On the testing set, the maximum and minimum values of MSE were 1.51 and 0.24, respectively, which also appeared in the GA and the improved GA. In addition, the MSE values of ACO, DCMA,



(a) Comparison of accuracy on the training set



(b) Comparison of accuracy on the testing set

FIGURE 9. Comparison of accuracy of different algorithms.

TABLE 2. Comparison of runtime of different algorithms.

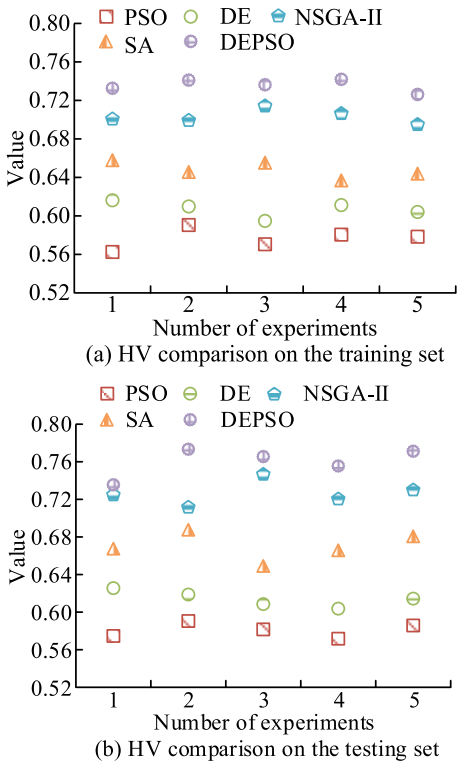
Algorithm	Training set Number of experiments				Testing set Number of experiments			
	1	2	3	4	1	2	3	4
PSO	40.17s	20.0 5s	30.1 1s	33.4 5s	41.2 3s	20.6 8s	31.5 4s	34.6 8s
DE	23.7 6s	20.0 7s	27.3 4s	28.2 2s	24.3 2s	21.4 5s	26.7 8s	28.8 9s
NSGA-II	11.9 8s	9.78 s	10.3 7s	11.6 6s	12.1 4s	9.66 s	10.5 5s	11.7 8s
SA	18.7 9s	18.8 3s	18.5 7s	17.9 6s	19.0 1s	18.9 6s	17.5 4s	18.6 5s
DEPSO	3.82 s	4.01 s	4.23 s	3.95 s	3.97 s	3.81 s	4.15 s	4.09 s

and MOE-FCE were less than 1 on both the training and testing sets. From this, the designed improved GA has better performance, followed by the DCMA and MOE-FCE.

B. DEPSO ALGORITHM

To verify the DEPSO performance, the evaluation indicators selected in the study include the runtime, Hyper-Volume (HV), and Inverse Generation Distance (IGD). The comparative algorithms include PSO, DE, Non-dominated Sorting GA-II (NSGA-II), and SA. The experimental environment is consistent with the performance verification of the improved GA, so it will not be repeated. The running time of different algorithms is displayed in Table 2.

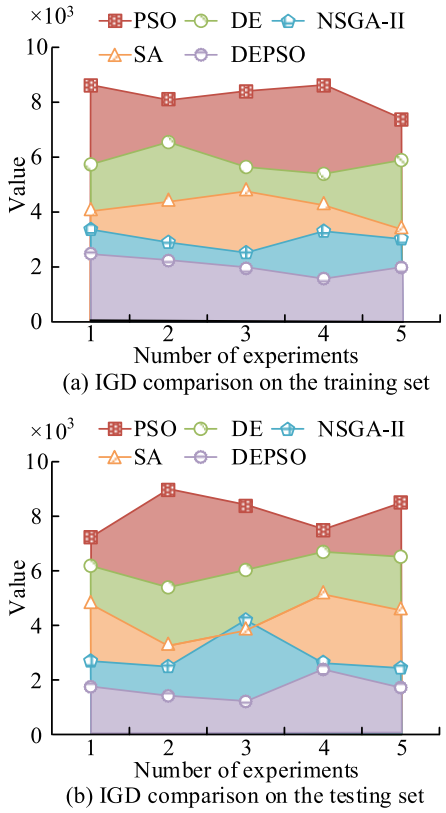
For the running time in Table 2, on the training set, the maximum values of PSO, DE, and NSGA-II were 40.17s,



(a) HV comparison on the training set

(b) HV comparison on the testing set

FIGURE 10. HV comparison of different algorithms.

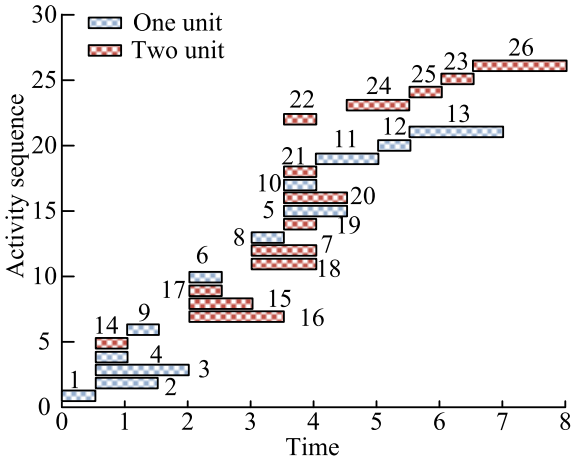


(a) IGD comparison on the training set

(b) IGD comparison on the testing set

FIGURE 11. Comparison of IGD using different algorithms.

28.22s, and 11.98s, respectively, and the minimum values were 20.05s, 20.07s, and 9.78s, respectively. The maximum



1(14):Surveying and setting out
2(15):Binding of cast-in-place steel bars
3(16):External wall hoisting
4(17):Composite beam and slab support
5(18):Template support
6(19):Sealing and grouting
7(20):Hoisting of laminated beam and slab
8(21):Reinforcement binding
9(22):Balcony panel support
10(23):Attachment parts for climbing frame
11(24):Binding steel bars on the board surface
12(25):Balconies and stairs
13(26):Concrete pouring and curing

FIGURE 12. Gantt chart for progress arrangement during assembly phase.

running time of SA and DEPSO was 18.83s and 4.23s, respectively, and the minimum values were 17.96s and 3.82s, respectively. On the testing set, the five algorithms were 41.23s, 28.89s, 12.14s, 19.01s, and 4.15s, while the minimum values were 20.68s, 21.45s, 9.66s, 17.54s, and 3.81s, respectively. From this, the DEPSO has the shortest running time and better performance. The HV comparison of different algorithms is shown in Figure 10.

From Figure 10 (a), the maximum HV values of PSO, DE, and NSGA-II were 0.582, 0.613, and 0.719, and the minimum values were 0.568, 0.598, and 0.705. The maximum HV values for SA and DEPSO were 0.657 and 0.736, and the minimum values were 0.637 and 0.721. In Figure 10 (b), on the testing set, the maximum HV values of the five algorithms were 0.591, 0.634, 0.744, 0.687, and 0.765, respectively, while the minimum values were 0.571, 0.603, 0.717, 0.645, and 0.732, respectively. A high HV value indicates good performance. The HV of the DEPSO is significantly higher than other comparative algorithms. From this, the DEPSO has superior performance. The IGD for different algorithms is shown in Figure 11.

From Figure 11 (a), on the training set, the maximum IGD values of PSO, DE, and NSGA-II were 8.56×10^{-3}, 6.21×10^{-3}, and 3.52×10^{-3}, respectively, and the minimum values were 7.93×10^{-3}, 5.77×10^{-3}, and 2.64×10^{-3}, respectively. The maximum IGD values of SA and DEPSO

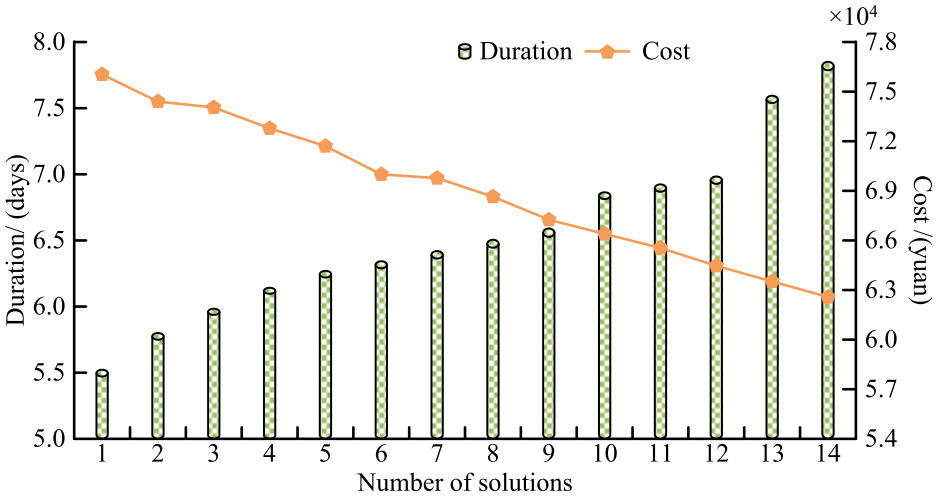


FIGURE 13. The OF value of the Pareto optimal solution during the production phase.

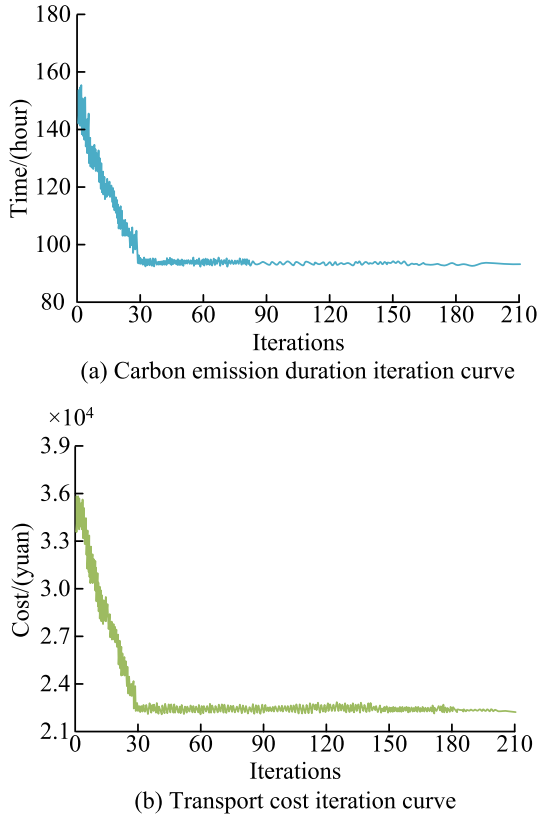
were 4.58×10^{-3} and 2.35×10^{-3}, respectively, and the minimum values were 3.62×10^{-3} and 1.47×10^{-3}, respectively. In Figure 11 (b), on the testing set, the maximum IGD values of the five algorithms were 8.79×10^{-3}, 6.52×10^{-3}, 4.05×10^{-3}, 4.77×10^{-3}, and 2.17×10^{-3}, respectively, while the minimum values were 7.24×10^{-3}, 5.24×10^{-3}, 2.18×10^{-3}, 3.27×10^{-3}, and 1.29×10^{-3}, respectively. A small IGD value indicates that the overall performance, including convergence and distribution performance, is good. The IGD value of the DEPSO is the smallest in both the training and testing sets. It can be inferred that the DEPSO performs better.

C. INSTANCE VERIFICATION

To verify the effectiveness of scheduling models at different stages, engineering cases are conducted at different stages. The selected engineering example for the study is an affordable housing project in a certain area, which consists of two units, with a total of eighteen floors. In addition, the exterior walls, beams, slabs, stairs, and balconies of the affordable housing are all prefabricated. The main resources provided on the construction site include one tower, three surveyors, twelve formwork hoisting workers, five ordinary workers, eleven steel reinforcement workers, seven grouting workers, and seven concrete workers. The PCs used in this affordable housing project have three complete production lines. The exterior wall panel production line is number one, the interior wall and laminated panel production line is number two, and the fixed formwork production line is number three. The types of PCs on the three production lines are diverse and comprehensive, with a reasonable layout, which can meet the production and construction progress requirements of the affordable housing project example. To solve the scheduling model for the assembly stage, the study adopts the Windows 11 operating system and MATLAB 2018a software. The processor is Intel Core i7 6800K, with a maximum memory of 128GB. The population size of the GA is 85. The schedule of the assembly phase is shown in Figure 12.

From Figure 12, the minimum construction period for the assembly phase was 8 days. On the first day, the measurement and layout of the first unit were carried out, including binding cast-in-place steel bars, hoisting the exterior wall, supporting composite beams and slabs, and the measurement and layout of the unit 2. The next day, the first unit adopted cast-in-place structural steel bars, external wall hoisting, and balcony slab support reinforcement. On the third day, the unit 1 was filled with joints, grouted, and the unit 2 was reinforced with cast-in-place structural steel bars, stacked beams, plate supports, and external wall hoisting. On the fourth day, the first unit was subjected to formwork erection, lifting of composite beams and slabs, binding of steel bars, pre-embedding of water and electricity pipelines, and installation of climbing accessories. The second unit involved external wall hoisting, formwork installation, joint sealing, grouting, composite beam and slab hoisting, reinforcement, water and electricity pipelines, and balcony slab support. On the fifth day, one 1 was carried out template support and binding of steel bars on the board surface. The lifting and binding of steel bars on the surface of the stacked beam and slab were completed in unit 2. On the sixth day, the hoisting balcony and stairs of unit 1 were poured and cured with concrete, as well as the binding of steel bars on the surface of unit 2 and the hoisting of the balcony and stairs. On the seventh day, the concrete pouring and curing of Unit 1, as well as the installation of climbing frame attachments and concrete pouring and curing of unit 2, were completed. On the eighth day, the concrete pouring and curing of unit 2 were completed. The constructed assembly stage scheduling model is effective. The OF value of the Pareto optimal solution during the production phase is shown in Figure 13.

From Figure 13, there were 14 optimal solutions in the scheduling model solution during the production phase. The construction period for solutions 1 to 10 was 5.5, 5.8, 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, and 6.8 days, respectively. The costs were 75969, 74791, 73949, 72997, 71963, 69571, 69437, 68117, 67241, and 66515 yuan, respectively. The construction period from solutions 11 to 14 was 6.9, 7, 7.6, and



(a) Carbon emission duration iteration curve

(b) Transport cost iteration curve

FIGURE 14. Iterative curve of carbon emission duration and transportation cost during the construction phase.

7.8 days, respectively, with cost of 65339, 64471, 63593, and 62877 yuan. The shortest construction period and the lowest cost have not been achieved together. There is an inverse relationship between construction period and cost. Therefore, according to different preferences, corresponding construction schedule arrangements are formulated. The iterative curve of carbon emission duration and transportation cost during the construction phase is shown in Figure 14.

From Figure 14 (a), the change in carbon emission duration decreased with the increase of iteration times. The carbon emission duration curve tended to flatten after nearly 30 iterations, with a corresponding total duration of 93.8 hours. In Figure 14 (b), the change in transportation cost also decreased with the increase of iteration times. When the transportation cost curve tended to flatten, the corresponding number of iterations was also nearly 30. The corresponding transportation cost was 22516 yuan. There is a positive relationship between carbon emission duration and transportation cost. When carbon emissions are minimized, the corresponding transportation cost is also minimized. Therefore, there is only one optimal solution. The duration of carbon emissions is 93.8h, and the transportation cost is 22516 yuan, respectively.

V. CONCLUSION

To control the construction process of PC, a scheduling model for the assembly, production, and transportation stages of PC was designed. The solving algorithms for these scheduling models are designed. The results showed that the minimum duration of the assembly phase was 8 days. The minimum carbon emission duration and transportation cost during the transportation phase were 93.8h and 22516 yuan, respectively. The duration and cost of the production phase couldn't be minimized together. Corresponding construction schedules should be formulated based on different preferences. After nearly 180 iterations on the training set, the improved GA tended to flatten out. The traditional GA, IA, GA-RRA, and ACO all iterated nearly 250, 247, 210, and 235 iterations respectively before stabilizing. From this, the improved GA performed better. On the training set, the maximum runtime of the DEPSO was 4.23s, while the maximum runtime of the PSO, DE, NSGA-II, and SA were 40.17s, 28.22s, 11.98s, and 18.83s, respectively. The maximum HV value of the DEPSO was 0.736, while the maximum HV values of the other four algorithms were 0.582, 0.613, 0.719, and 0.657, respectively. The maximum IGD value of the DEPSO was 2.35×10^{-3}, while the maximum IGD values of the other four algorithms are 8.56×10^{-3}, 6.21×10^{-3}, and 3.52×10^{-3} and 4.58×10^{-3}, respectively. It can be inferred that the DEPSO performs better. The study mainly analyzes the construction scheduling for prefabricated buildings under stable and unchanged environmental conditions. Future research can involve the impact of different factors on construction scheduling in uncertain environments. In addition, the study analyzes the construction scheduling of prefabricated buildings from three stages. Future research can merge and optimize these three stages.

REFERENCES

[1] W. Dan and L. Jinghan, "Development and application of risk assessment model for prefabricated building construction," *CSSJ*, vol. 32, no. 10, pp. 69–75, Apr. 2022, doi: 10.16265/j.cnki.issn1003-3033.2022.10.2119.

[2] M. Ma, K. Zhang, L. Chen, and S. Tang, "Analysis of the impact of a novel cool roof on cooling performance for a low-rise prefabricated building in China," *Building Services Eng. Res. Technol.*, vol. 42, no. 1, pp. 26–44, Jan. 2021, doi: 10.1177/0143624420960276.

[3] Y. Yuan, S. Ye, and L. Lin, "Process monitoring with support of IoT in prefabricated building construction," *Sensors Mater.*, vol. 33, no. 4, p. 1167, Apr. 2021, doi: 10.18494/sam.2021.3003.

[4] H. Zare and M. Hajarian, "Determination of regularization parameter via solving a multi-objective optimization problem," *Appl. Numer. Math.*, vol. 156, pp. 542–554, Oct. 2020, doi: 10.1016/j.apnum.2020.05.021.

[5] M. M. M. Farahi, M. Ahmadi, and B. Dabir, "Model-based multi-objective particle swarm production optimization for efficient injection/production planning to improve reservoir recovery," *Can. J. Chem. Eng.*, vol. 100, no. 3, pp. 503–520, Jun. 2021, doi: 10.1002/cjce.24158.

[6] H. Wang, H. Wang, and Y. Li, "Production decision rescheduling of prefabricated building parts subject to interference from the arrival of new orders," *Int. J. Ind. Eng.*, vol. 27, no. 5, pp. 791–809, Jan. 2020, doi: 10.23055/ijietap.2020.27.5.6547.

[7] X. Yu, "A RBF fuzzy logic neural network algorithm for construction resource scheduling," *J. Intell. Fuzzy Syst.*, vol. 41, no. 4, pp. 4937–4945, Nov. 2021, doi: 10.3233/jifs-189980.

[8] M. Ruan and F. Xu, "Improved eight-process model of precast component production scheduling considering resource constraints," *J. CIVIL Eng. Manage.*, vol. 28, no. 3, pp. 208–222, Feb. 2022, doi: 10.3846/jcem.2022.16454.

[9] H. Zhang and L. Yu, "Resilience-cost tradeoff supply chain planning for the prefabricated construction project," *J. Civil Eng. Manage.*, vol. 27, no. 1, pp. 45–59, Jan. 2021, doi: 10.3846/jcem.2021.14114.

[10] M. Podolski, "Effective allocation of manpower in the production of precast concrete elements with the use of metaheuristics," *J. Civil Eng. Manage.*, vol. 28, no. 4, pp. 247–260, Mar. 2022, doi: 10.3846/jcem.2022.16383.
[11] J. Du, P. Dong, and V. Sugumaran, "Dynamic production scheduling for prefabricated components considering the demand fluctuation," *Intell. Autom. Soft Comput.*, vol. 26, no. 4, pp. 715–723, Jan. 2020, doi: 10.32604/iasc.2020.010105.
[12] G. Gebreyesus, G. Fellek, A. Farid, S. Fujimura, and O. Yoshie, "Gated-attention model with reinforcement learning for solving dynamic job shop scheduling problem," *IEEJ Trans. Electr. Electron. Eng.*, vol. 18, no. 6, pp. 932–944, May 2023, doi: 10.1002/tee.23788.
[13] B. Zhang and Z. Zheng, "Model and algorithm for vessel scheduling optimisation through the compound channel with the consideration of tide height," *Int. J. Shipping Transp. Logistics*, vol. 13, nos. 3–4, p. 445, Jan. 2021, doi: 10.1504/ijstl.2021.10036556.
[14] J. Wu, F. Lu, J. Zhang, J. Yang, and L. Xing, "Design of task priority model and algorithm for imaging observation problem," *J. Syst. Eng. Electron.*, vol. 31, no. 2, pp. 321–334, Apr. 2020, doi: 10.23919/JSEE.2020.000010.
[15] S. Meng, Q. Zhu, F. Xia, and J. Lu, "Research on parameter optimisation of dynamic priority scheduling algorithm based on improved reinforcement learning," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 16, pp. 3171–3178, Aug. 2020, doi: 10.1049/iet-gtd.2019.1468.
[16] X. Yan, H. Zhang, and W. Zhang, "Intelligent monitoring and evaluation for the prefabricated construction schedule," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 3, pp. 391–407, Feb. 2023, doi: 10.1111/mice.12838.
[17] M. Almashaqbeh and K. El-Rayes, "Minimizing transportation cost of prefabricated modules in modular construction projects," *Eng., Construct. Architectural Manage.*, vol. 29, no. 10, pp. 3847–3867, Dec. 2022, doi: 10.1108/ecam-11-2020-0969.
[18] X. Ouyang, Y. Peng, S. Pan, and P. Ouyang, "Information visualization method for intelligent construction of prefabricated buildings based on P-ISOMAP algorithm," *Int. J. Emerg. Electr. Power Syst.*, vol. 24, no. 1, pp. 73–89, Nov. 2022, doi: 10.1515/ijeeps-2022-0118.
[19] S. Fasching, T. Huber, M. Rath, and J. Kollegger, "Semi-precast segmental bridges: Development of a new construction method using thin-walled prefabricated concrete elements," *Struct. Concrete*, vol. 22, no. 3, pp. 1561–1573, Mar. 2021, doi: 10.1002/suco.202000474.
[20] A. Yudina, S. Sychov, and A. Gaido, "Construction system for the erection of prefabricated buildings out of factory-made modules," *Archit. Eng.*, vol. 5, no. 2, pp. 32–37, Jun. 2020, doi: 10.23968/2500-0055-2020-5-2-32-37.
[21] J. Cao, H. Yan, W. Li, D. Li, and Y. Wang, "Optimization of stator ventilation structure of high-speed railway traction motor based on the genetic algorithm," *IET Electr. Power Appl.*, vol. 17, no. 3, pp. 281–292, Mar. 2023, doi: 10.1049/elp2.12263.
[22] Y. Wang, S. Liu, H. Wu, C. Zhang, J. Xu, and H. Yu, "On-demand optimize design of sound-absorbing porous material based on multi-population genetic algorithm," *e-Polymers*, vol. 20, no. 1, pp. 122–132, Mar. 2020, doi: 10.1515/epoly-2020-0014.
[23] S. S. Sathiyadhas and M. C. V. S. Antony, "A network intrusion detection system in cloud computing environment using dragonfly improved invasive weed optimization integrated shepard convolutional neural network," *Int. J. Adapt. Control Signal Process.*, vol. 36, no. 5, pp. 1060–1076, May 2022, doi: 10.1002/acs.3386.
[24] V. S. Gutte and D. Parasar, "Sailfish invasive weed optimization algorithm for multiple image sharing in cloud computing," *Int. J. Intell. Syst.*, vol. 37, no. 7, pp. 4190–4213, Jul. 2022, doi: 10.1002/int.22717.
[25] J. Sun, Z. Zhang, L. Yang, and J. Zheng, "Multi-view hand gesture recognition via Pareto optimal front," *IET Image Process.*, vol. 14, no. 14, pp. 3579–3587, Oct. 2020, doi: 10.1049/iet-ipr.2019.0924.
[26] A. I. Jabbr, H. Gaja, and U. O. Koylu, "Multi-objective optimization of operating parameters for a H2/diesel dual-fuel compression-ignition engine," *Int. J. Hydrogen Energy*, vol. 45, no. 38, pp. 19965–19975, Jul. 2020, doi: 10.1016/j.ijhydene.2020.05.071.
[27] S. Grassi and L. Pareschi, "From particle swarm optimization to consensus based optimization: Stochastic modeling and mean-field limit," *Math. Models Methods Appl. Sci.*, vol. 31, no. 8, pp. 1625–1657, Jul. 2021, doi: 10.1142/s0218202521500342.
[28] C. L. Tsai and G. H. Fredrickson, "Using particle swarm optimization and self-consistent field theory to discover globally stable morphologies of block copolymers," *Macromolecules*, vol. 55, no. 12, pp. 5249–5262, Apr. 2022, doi: 10.1021/acs.macromol.2c00042.
[29] O. Hachana, "Accurate PEM fuel cells parameters estimation using hybrid artificial bee colony differential evolution shuffled complex optimizer," *Int. J. Energy Res.*, vol. 46, no. 5, pp. 6383–6405, Apr. 2022, doi: 10.1002/er.7576.
[30] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar, "Real-time human detection and counting system using deep learning computer vision techniques," *Artif. Intell. Appl.*, vol. 1, no. 4, pp. 221–229, Oct. 2022, doi: 10.47852/bonviewaia2202391.
[31] B. Zhang, J. Song, S. Zhao, H. Jiang, J. Wei, and Y. Wang, "Prediction of yarn stsrength based on an expert weighted neural network optimized by particle swarm optimization," *Textile Res. J.*, vol. 91, nos. 23–24, pp. 2911–2924, Dec. 2021, doi: 10.1177/00405175211022619.
[32] G. Zhang, B. Liu, L. Wang, D. Yu, and K. Xing, "Distributed co-evolutionary memetic algorithm for distributed hybrid differentiation flowshop scheduling problem," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1043–1057, Oct. 2022, doi: 10.1109/TEVC.2022.3150771.
[33] W. Li, L. He, and Y. Cao, "Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10721–10734, Oct. 2022, doi: 10.1109/TCYB.2021.3069184.

ZHIPENG HUO was born in Yulin, Shaanxi, Han Nationality, in 1988. He received the bachelor's degree in civil engineering and the master's degree in structural engineering from Kunming University of Technology, in 2009 and 2012, respectively.

Since 2014, he has been a Teacher with Yan'an University. He has published more than ten papers, presided over, and participated in more than ten scientific research projects of various types. His research interest includes the application and research of assembly building.

XIAOQIANG WU was born in Yan'an, Shaanxi, Han Nationality, in 1977. He received the master's degree in structural engineering from Xi'an University of Architecture and Technology, in 2005.

Since 2005, he has been a Teacher with Yan'an University. He has published more than 30 papers, presided over, and participated in more than ten scientific research projects of various types. His research interest includes the application and research of assembly building.

TAO CHENG was born in Shangluo, Shaanxi, Han Ethnicity, in 1998. He received the bachelor's degree in civil engineering from Yan'an University, in 2020.

Since 2020, he has been employed with China-shun International Engineering Design Company Ltd. He has participated in more than 100 actual engineering projects.