

Optimization of precast Component Production Scheduling: A Comparative Study of Genetic Algorithm and Simulated Annealing

Junyong Liang^{1,2}, Shicen Liu¹, Meixue Lai¹, Yingmei Liao¹, Ziwei Wu¹, Yurui Yuan¹, Yong Liu² *

¹ School of Civil Engineering, Sichuan University of Science and Engineering, Zigong 643000, China

² Luzhou Key Laboratory of Intelligent Construction and Low-carbon Technology, Luzhou 646000, Sichuan, China

*Corresponding author: Yong Liu (Email: 232529047@qq.com)

Abstract: The production of precast components is a core link in the industrialization of construction, and the rationality of production scheduling directly determines project costs, construction progress, and resource utilization efficiency. Faced with challenges such as intricate production processes, multiple resource constraints, and the need for flexible response to market demands, traditional scheduling methods are increasingly incompetent. To address these issues, this study explores the application of two intelligent optimization algorithms—Genetic Algorithm (GA) and Simulated Annealing (SA)—in precast component production scheduling. Based on a systematic analysis of the production process and constraints of precast components, corresponding mathematical models are established for each algorithm. Through case studies using real production data from precast component manufacturers, the effectiveness of the two algorithms is verified. The results show that both GA and SA can significantly optimize the production sequence, shorten the maximum makespan, and improve equipment utilization. Specifically, GA achieves a makespan of 45 hours with a more efficient convergence rate, while SA effectively avoids local optima and obtains a makespan of 46.8 hours. This research provides a scientific and flexible scheduling solution for precast component manufacturers and enriches the application of intelligent algorithms in the construction industry.

Keywords: Precast Components; Production Scheduling Optimization; Genetic Algorithm; Simulated Annealing; Flow Shop Scheduling.

1. Introduction

With the acceleration of urbanization and the promotion of green construction, precast buildings have emerged as a key direction for the transformation and upgrading of the construction industry [1]. Precast components, as the core of precast buildings, are produced through standardized factory processes, which can reduce on-site construction pollution, shorten the construction period, and improve component quality [2]. However, the production of precast components involves six consecutive processes—mold assembly, reinforcement installation, concrete pouring, steam curing, demolding, and finishing—and faces constraints such as limited equipment capacity, sequential process requirements, and parallel operability of steam curing [3]. In addition, most enterprises still rely on empirical scheduling, leading to problems such as low equipment utilization, long production cycles, and poor adaptability to order changes [4].

Intelligent optimization algorithms have shown remarkable advantages in solving combinatorial optimization problems such as production scheduling. Among them, Genetic Algorithm (GA), inspired by biological evolution, has strong global search capabilities and is widely used in job sequencing and resource allocation [5]. Simulated Annealing (SA), based on the physical annealing process of solids, can avoid local optima through probabilistic acceptance criteria and is suitable for complex constrained scheduling problems [6]. Although existing studies have applied GA or SA separately to precast component production scheduling, few studies have compared their performance in the same production scenario.

This study aims to fill this gap by: (1) analyzing the

production characteristics and constraints of precast components; (2) establishing mathematical models for GA and SA tailored to the production process; (3) verifying the algorithms through case studies with real production data; and (4) comparing the optimization effects and applicability of the two algorithms. The research results can provide technical support for precast component manufacturers to improve scheduling efficiency and reduce production costs.

2. Literature Review

2.1. Global Research Progress

The research on precast component production scheduling has gradually deepened with the popularization of precast buildings. Before 2000, due to the late development of precast construction technology, related research was scarce, with only two relevant papers published between 1954 and 1998 [7]. From 2000 to 2017, as precast buildings gained traction, scholars began to explore scheduling models: Chan and Hu [8] constructed a Flow Shop Scheduling Model (FSSM) considering production constraints and optimized it using GA to minimize makespan and penalty costs. Benjaoran et al. [9] proposed an order-based flow shop scheduling model and analyzed the impact of mold quantity and process time on scheduling results.

After 2018, research on precast component production scheduling entered a period of rapid development, with an average of 10 related papers published annually [10]. Phanden et al. [11] compared the performance of GA and SA in job shop scheduling and found that both algorithms can effectively improve production efficiency. Zhang et al. [12] integrated SA into a co-evolutionary algorithm to solve the

integrated scheduling problem of distributed flexible job shops. In terms of multi-line scheduling, Yang et al. [13] constructed a flow shop scheduling model for multiple production lines, breaking the limitation of single-line scheduling research.

2.2. Domestic Research Status

Domestic scholars have focused on combining actual production constraints to optimize scheduling models. Wang Zhongyuan et al. [14] applied GA to optimize mold layout in precast component production and improved mold utilization. Yang Dinsheng [15] introduced the Metropolis criterion and verified the effectiveness of SA in solving production scheduling problems. Jing Huicheng et al. [16] proposed a hybrid algorithm combining SA and Particle Swarm Genetic Algorithm (PSG) to address the problem of GA easily falling into local optima. In addition, some studies have integrated digital technologies such as BIM and the Internet of Things into scheduling management to realize real-time monitoring and dynamic adjustment of production processes [17].

However, existing studies still have limitations: most focus on a single algorithm, lack comparative analysis of different algorithms in the same scenario; few studies fully consider the parallel operability of steam curing and the dynamic changes of production resources. This study addresses these issues by comparing GA and SA in precast component production scheduling and optimizing the algorithm design to adapt to complex production constraints.

3. Problem Statement

3.1. Production Process of precast Components

The production of precast components follows six sequential processes, as shown in Figure 1:

1. Mold Assembly: Building the basic framework for components, ensuring dimensional accuracy.
2. Reinforcement Installation: Tying steel bars to enhance the structural strength of components.
3. Concrete Pouring: Filling the mold with concrete to form the main shape of components.
4. Steam Curing: Accelerating concrete hardening in curing kilns, which can process multiple 5. components in parallel.
6. Demolding: Separating components from molds after curing.
7. Finishing and Repair: Polishing and repairing components to meet quality standards.

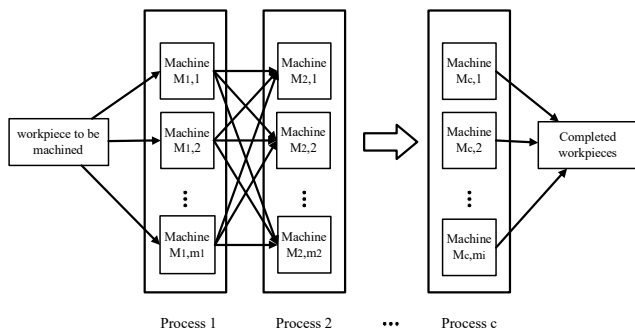


Figure 1. Schematic diagram of the precast component manufacturing process

3.2. Production Modes

There are two main production modes for precast components: Fixed Production: Suitable for small-batch, complex-shaped components. Production is carried out on

fixed pedestals, with molds fixed during processing. Flow-through Production: A highly automated mode suitable for large-scale standardized components. Processes are integrated into a production line, with continuous operation through automated equipment, featuring high labor specialization and production efficiency [18]. This study focuses on flow-through production, which belongs to the Hybrid Flow-shop Scheduling Problem (HFSP) due to the parallel operability of the steam curing process.

3.3. Key Constraints

During production scheduling, the following constraints must be satisfied:

Each component is processed only once in each process, with no repetition.

Except for steam curing, each machine can process only one component at a time.

A component can be processed on only one machine at any given time.

Once processing starts, it cannot be interrupted or preempted.

The processing time of each component in each process is fixed and not affected by scheduling order.

The sequence of processes is fixed; steam curing can start immediately after concrete pouring without waiting for machine availability.

4. Optimization Algorithms and Mathematical Models

4.1. Genetic Algorithm (GA)

4.1.1. Algorithm Principle

GA is an intelligent optimization algorithm inspired by Darwin's theory of evolution and Mendel's genetic laws. It searches for the optimal solution through operations such as selection, crossover, and mutation on a population of chromosomes (solutions) [19]. The core steps include population initialization, fitness evaluation, genetic operations, and population update.

4.1.2. Mathematical Model

The objective function is to minimize the maximum completion time (makespan) of all components:

$$\pi^* = \arg \min \{ f(\pi) = T_{n,m} \}$$

where:

$\pi = (\sigma_1, \sigma_2, \dots, \sigma_n)$ denotes the processing sequence of n components;

$T_{n,m}$ denotes the completion time of the n -th component in the m -th process.

Constraints:

Completion time of the first component in the first process:
 $T_{1,1} = t_{1,1}$

Completion time of the j -th component in the first process:
 $T_{j,1} = T_{j-1,1} + t_{j,1} (j=2, \dots, n)$

Completion time of the first component in the i -th process:
 $T_{1,i} = T_{1,i-1} + t_{1,i} (i=2, \dots, m)$

Completion time of the j -th component in the i -th process:
 $T_{j,i} = \max \{ T_{j-1,i}, T_{j,i-1} \} + t_{j,i} (j=2, \dots, n; i=2, \dots, m)$

where $t_{j,i}$ denotes the processing time of the j -th component in the i -th process.

4.1.3. Algorithm Design

Encoding: Process-based encoding is adopted, where each chromosome represents a component processing sequence.

For example, the chromosome [2, 4, 7, ..., 1] indicates that component 2 is processed first, followed by component 4, and so on.

Decoding: Generating executable scheduling plans based on encoding, including start time and completion time of each component in each process.

Fitness Function: $F=1/f(\pi)$, where a higher fitness value indicates a better solution.

Genetic Operations: Selection: Tournament selection is used to select individuals with high fitness for the next generation. Crossover: Two-point crossover is adopted to swap segments of parent chromosomes to generate offspring. Mutation: Swap mutation is used to randomly swap two genes in a chromosome to increase population diversity.

Avoiding Local Optima: Dynamically adjusting mutation probability (increasing when no optimal solution is found for multiple generations) and retaining elite individuals to ensure global search capability.

4.2. Simulated Annealing (SA)

4.2.1. Algorithm Principle

SA simulates the physical process of solid annealing: heating a solid to a high temperature to make particles in a disordered state, then cooling it slowly to reduce internal energy and reach a stable ground state [20]. The algorithm uses the Metropolis criterion to accept inferior solutions with a certain probability, enabling it to escape local optima and search for the global optimal solution.

4.2.2. Mathematical Model

The objective function is the same as GA: minimizing the maximum completion time:

$$\min K_{\max} = \max_i \left\{ \sum_{j=1}^m C_{\sigma(i),j} x_{ijt} \right\}$$

where:

x_{ijt} is a binary decision variable (1 if the i -th component starts processing in the j -th process at time t , 0 otherwise);

$C_{\sigma(i),j}$ denotes the completion time of the i -th component

in the j -th process.

Constraints:

Process sequence constraint:

$$\sum_{t=1}^T x_{ijt} \leq \sum_{t=1}^T x_{i(j+1)t} \quad (i=1, \dots, n; j=1, \dots, m-1)$$

Time constraint (excluding steam curing):

$$S_{\sigma(i),j} = \max\{C_{\sigma(i-1),j}, C_{\sigma(i),j-1}\}, C_{\sigma(i),j} = S_{\sigma(i),j} + P_{\sigma(i),j}$$

Steam curing constraint: $S_{\sigma(i),4} = C_{\sigma(i),3}, C_{\sigma(i),4} = S_{\sigma(i),4} + P_{\sigma(i),4}$ where $S_{\sigma(i),j}$ and $P_{\sigma(i),j}$ denote the start time and processing time of the i -th component in the j -th process, respectively.

4.2.3. Algorithm Design

Encoding: Task order-based encoding, where each solution is a permutation of component processing sequences.

Initial Solution: Randomly generate a component processing sequence as the initial solution.

Neighborhood Structure: Exchange the positions of two randomly selected components in the current solution to generate a new solution.

Metropolis Criterion: If the new solution is better ($\Delta f < 0$), accept it; otherwise, accept it with probability $e^{-\Delta f/T}$ (where T is the current temperature).

Cooling Strategy: Exponential cooling: $T_{k+1} = \lambda T_k$ (cooling coefficient $\lambda=0.98$).

Termination Condition: The algorithm terminates when the temperature drops to a threshold (T_{\min}) or the maximum number of iterations is reached.

5. Case Study and Results

5.1. Experimental Setup

Real production data from a precast component manufacturer in Zigong City is used for verification. The manufacturer produces 12 types of components, with 6 steam curing kilns. The processing time of each component in each process is shown in Table 1 (integrated from the two original studies).

Table 1. Processing time of each component (unit: hours)

No.	Assembling Molds	Reinforcement Installation	Concrete Pouring	Steam Curing	Demoulding	Tidying up and repairing
1	1.7	1.9	2.3	15	2.3	0.5
2	1.8	1.8	2.5	15	2.4	0.4
3	2.0	2.2	2.6	15	2.7	0.5
4	1.9	2.2	3.0	15	3.0	0.5
5	2.6	2.7	2.8	15	3.1	0.4
6	2.5	2.8	2.9	15	3.0	0.5
7	1.4	1.6	2.1	15	1.8	0.5
8	1.6	1.7	2.0	15	1.9	0.5
9	1.5	1.3	1.9	15	1.8	0.4
10	1.4	1.6	1.1	15	1.7	0.5
11	1.2	1.7	1.0	15	1.6	0.5
12	1.1	1.5	0.9	15	1.4	0.5

Algorithm parameters are set as follows:

GA: Population size = 200, number of iterations = 1000, crossover rate = 0.7, mutation rate = 0.05.

SA: Initial temperature $T_0=1000$, cooling coefficient $\lambda=0.98$, number of iterations per temperature = 200, minimum temperature $T_{\min}=10^{-3}$.

The algorithms are implemented using MATLAB R2016a, and the scheduling results are analyzed through Gantt charts and convergence curves.

5.2. Results Analysis

5.2.1. GA Optimization Results

After 1000 iterations, GA obtains the optimal processing sequence: 12-9-7-8-11-2-1-10-4-3-6-5. The Gantt chart (Figure 2) shows that the components are processed in an orderly manner, with minimal waiting time between processes. The maximum completion time is 45 hours.

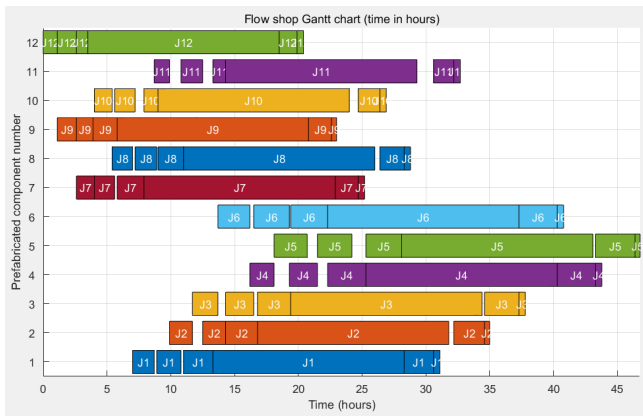


Figure 2. Gantt chart of GA-optimized production scheduling (makespan = 45 hours)

The convergence curve (Figure 3) indicates that GA converges rapidly in the early stage, with the makespan decreasing significantly. In the middle stage, it temporarily falls into a local optimum but escapes through dynamic mutation probability adjustment and elite retention, eventually converging to the global optimal solution.

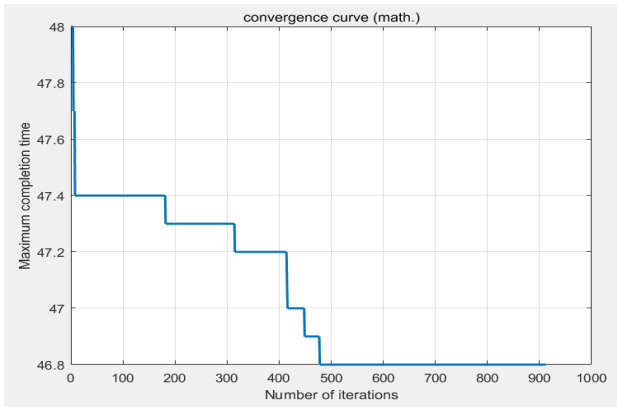


Figure 3. Convergence curve of the genetic algorithm

5.2.2. SA Optimization Results

SA obtains the optimal processing sequence: 10-12-9-7-1-8-5-6-3-11-4-2. The Gantt chart (Figure 4) shows that the components are arranged compactly, with high machine utilization. The maximum completion time is 46.8 hours.

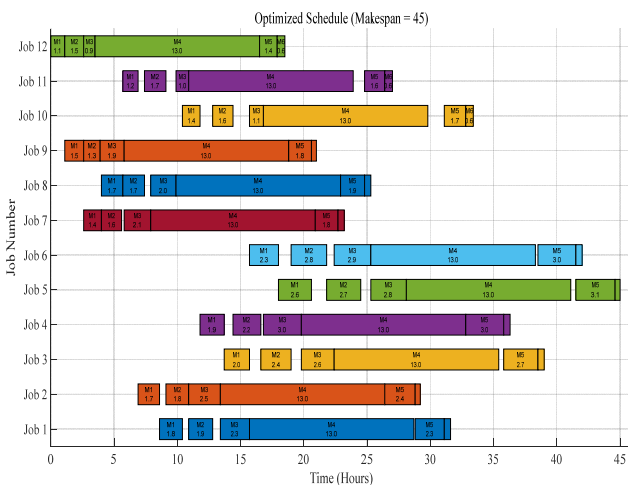


Figure 4. Gantt chart of SA-optimized production scheduling (makespan = 46.8 hours)

The convergence curve (Figure 5) shows that SA avoids local optima effectively through the Metropolis criterion. In

the early stage, the makespan decreases rapidly; in the later stage, as the temperature decreases, the acceptance probability of inferior solutions decreases, and the makespan stabilizes at the optimal value.

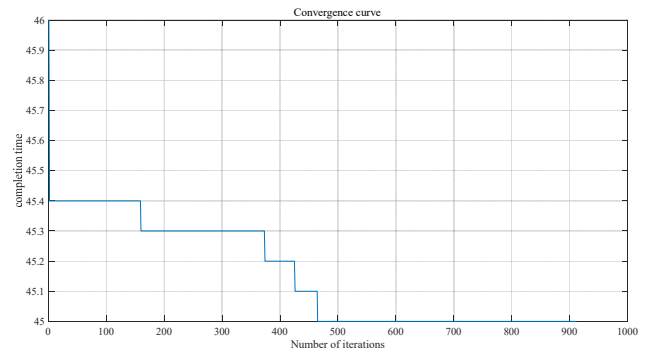


Figure 5. Convergence curve of the simulated annealing algorithm

5.2.3. Algorithm Comparison

The performance of GA and SA is compared in terms of makespan, convergence rate, and stability:

Makespan: GA achieves a shorter makespan (45 hours) than SA (46.8 hours), indicating higher optimization efficiency.

Convergence Rate: GA converges faster, reaching the optimal solution within 800 iterations, while SA requires more iterations to stabilize.

Stability: SA shows stronger stability in avoiding local optima, while GA is prone to local optima without improvement strategies.

5.3. Discussion

Both algorithms effectively optimize precast component production scheduling by rationalizing the processing sequence and resource allocation. GA is more suitable for scenarios requiring fast optimization and short production cycles, while SA is more suitable for complex production environments with multiple local optima. The difference in makespan is due to the different optimization mechanisms: GA uses global search through population evolution, while SA uses probabilistic search based on temperature cooling.

6. Conclusion and Future Work

This study applies Genetic Algorithm and Simulated Annealing to precast component production scheduling, establishes corresponding mathematical models, and verifies their effectiveness through case studies. The results show that both algorithms can significantly shorten the production cycle, improve equipment utilization, and solve the problems of traditional scheduling methods such as poor flexibility and low efficiency. GA has advantages in optimization efficiency and makespan, while SA excels in avoiding local optima.

Future research can be expanded in the following directions: (1) Consider more practical constraints such as raw material supply fluctuations, equipment failures, and dynamic order insertion; (2) Develop hybrid algorithms (e.g., GA-SA hybrid algorithm) to combine the advantages of both algorithms and improve optimization performance; (3) Integrate digital technologies such as BIM and IoT to build a visualized intelligent scheduling system, realizing real-time monitoring and dynamic adjustment of production processes; (4) Extend the research to multi-objective optimization considering cost, energy consumption, and carbon emissions to meet the

requirements of green and low-carbon production.

Acknowledgments

This research was Supported by a grant from Luzhou Key Laboratory of Intelligent Construction and Low-carbon Technology named Intelligent Scheduling Optimization of Production Resources for Prefabricated Concrete Components Based on Deep Reinforcement Learning (Project No.: 1) and the "College Student Innovation and Entrepreneurship Training Program Project" named Intelligent layout optimization of prefabricated concrete components in prefabricated buildings (Project No.: S202410622087).

References

- [1] Liang, W., Zhao, Y., & Yin, X. (2023). Precast production scheduling in off-site construction: Mainstream contents and optimization perspective. *Journal of Cleaner Production*, 405, 137054.
- [2] Ban, D. (2020). Research on the optimization method of precast concrete components production for assembled buildings [D]. Xi'an University of Architecture and Technology.
- [3] Wang, R., Ban, D., & Wang, Y. (2020). Research on precast concrete component production scheduling problem considering resource constraints. *Manufacturing Automation*, 42(12), 63-67.
- [4] Chen, J., Zhang, Y., & Gao, X. (2024). Production scheduling optimization of PC components based on improved genetic algorithm. *Mechanical Design and Manufacturing Engineering*, 53(1), 95-99.
- [5] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- [6] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- [7] Qin, X., Zhu, Q., & Han, J. (2023). Construction scheduling optimization for integrated precast component production-transportation-assembly. *Journal of Huaqiao University (Natural Science Edition)*, 44(3), 366-373.
- [8] Chan, W. T., & Hu, H. (2002). Production Scheduling for Precast Plants using a Flow Shop Sequencing Model. *Journal of Computing in Civil Engineering*, 16(3), 165-174.
- [9] Benjaoran, V., & Dawood, N. (2006). Intelligence approach to production planning system for bespoke precast concrete products. *Automation in Construction*, 15(6), 737-745.
- [10] Zhang, R., Feng, X., & Ma, G. (2023). Optimization of precast components production scheduling and transportation considering carbon emission based on NSGA-III algorithm. *Journal of Engineering Management*, 37(2), 153-158.
- [11] Phanden, K. R., Gupta, S., & Wolde, B. (2025). Optimisation of job shop scheduling problem using genetic algorithm and simulated annealing: a case study of manufacturing industry. *International Journal of System Assurance Engineering and Management*, (prepublish), 1-10.
- [12] Zhang, Z., Fu, Y., & Gao, K. (2024). A cooperative evolutionary algorithm with simulated annealing for integrated scheduling of distributed flexible job shops and distribution. *Swarm and Evolutionary Computation*, 85, 101467.
- [13] Yang, Z., Ma, Z., & Wu, S. (2016). Optimized flowshop scheduling of multiple production lines for precast production. *Automation in Construction*, 72, 321-329.
- [14] Wang, Z., & Chen, X. (2019). Research on optimization of mold arrangement in plate precast component production based on genetic algorithm. *Journal of Engineering Management*, 33(1), 45-49.
- [15] Yang, D. (2004). Solving production scheduling problems with simulated annealing algorithm. *Journal of Huanggang Normal College*, 24(3), 45-47.
- [16] Jing, H., & Wang, Y. (2020). Simulated annealing algorithm to optimize PSO-GA algorithm to solve flexible flow shop scheduling problem. *Small Microcomputer Systems*, 41(5), 996-999.
- [17] Yao, G., Qin, W., & Zhou, M. (2019). Optimization of BIM technology positioning for industrial production of PC components. *Journal of Civil and Environmental Engineering (in Chinese and English)*, 41(2), 140-146+166.
- [18] Deng, H., Liu, H., & Dong, Z. (2025). Optimized genetic algorithm for solving flexible flow shop scheduling problem. *Computer Knowledge and Technology*, 21(2), 15-18.
- [19] Chen, Z., Hu, Y., & Liu, J. (2024). Solving flexible job shop scheduling problem using improved genetic algorithm with reinforcement learning. *Science, Technology and Engineering*, 24(25), 10848-10856.
- [20] Tian, D., & Chi, H. (2006). Hybrid genetic algorithm and simulated annealing method. *Computer Engineering and Applications*, 42(22), 63-65.