

# Optimization of Asynchronous Parallel Tasks Scheduling with Multi-Resource Constraints

Zhang Xinyu<sup>1</sup>, Liu JinJian<sup>2</sup>, Zhang Guanwei<sup>1</sup>, Guo Wei<sup>1\*</sup>

<sup>1</sup>Tianjin University, School of Mechanical Engineering, Nankai District, Tianjin, 300072, China

<sup>2</sup>Tianjin Renai College, School of Digital Media and Design Arts, Jinghai District, Tianjin, 301636, China

E-mail: zxy2021201221@163.com

\*Corresponding author

**Keywords:** asynchronous parallel disassembling, improved genetic algorithm, disassembling task planning

**Received:** December 27, 2023

*Complex equipment disassembly tasks often require a group of people to complete, and the same time satisfying various resource constraints. This paper proposes an improved elite genetic algorithm (IGA) for the asynchronous parallel disassembly (APD) problem with the task priority, disassembly workspace interference and human resource constraints. The feasibility and effectiveness of the algorithm are verified through a hydraulic turbine hoisting and disassembling task instance. Results show that compared with classical algorithms (ACO&AGA) applied in APD, the proposed method is feasible and effective, and also has practical guiding significance for planning complex disassembling projects.*

*Povzetek: Članek predstavlja izboljšan elitni genetski algoritem za razstavljanje kompleksne opreme z asinhronimi nalogami, ki upošteva prioritete, medsebojne motnje in človeške vire.*

## 1 Introduction

As technology advances and develops, the scale and complexity of industrial equipment have become increasingly complex. For hydraulic turbines, vessels, and aerospace equipment, there are diverse maintenance procedures with many tasks, which are generally completed in separate project by assigned teams. As such, a significant research issue is the way to effectively execute these initiatives in an economical and effective manner<sup>[1]</sup>. Disassembling is one of the maintenance procedures. Unlike common ordinary disassembly scenarios, the disassembly tasks of complex equipment require multiple people to participate collaboratively in the disassembly, while also necessitating numerous specialized disassembly devices<sup>[2]</sup>.

Disassembly sequence planning (DSP), which takes into consideration the removal directions and other component qualities, chooses the sequence in which components are removed from products in an effort to minimize disassembly time, cost, or other considerations. It can be thought of as a difficult NP-hard computational optimization issue<sup>[3]</sup>. The DSP research mainly did from the perspectives of how to decide a disassembly mode, how to build a disassembly modelling and how to apply a selected planning method<sup>[4]</sup>. The disassembly mode can be built according to the particular disassembly scenarios,

such as sequential and parallel disassembly. Currently, there has been extensive research done on sequential disassembly sequence planning (SDSP), such as ACO<sup>[5]</sup>, genetic algorithms<sup>[6,7]</sup> and PSO<sup>[8]</sup> to solve SDSP problem. In recent years, parallel disassembly sequence planning (PDSP) has been mentioned more frequently, because large or complicated items may require long processing times due to the SDSP paradigm<sup>[4]</sup>. Based on the PDSP models, researchers have tried to find multiple ways to describe the disassembly precedence relationship of products. They extracted the precedence relationships between different parts in following ways: manual work<sup>[9,10]</sup>, graphs<sup>[3]</sup>, Petri nets<sup>[11]</sup>, matrix-based<sup>[12]</sup>. According to the latest related research, graph and matrix-based methods are primarily used to describe the priority order of tasks<sup>[13,18]</sup>. Graphs are a useful tool for describing the connections of priority and interaction between product components. However, processing disassembling restriction connections on computers can be done more easily with matrix-based techniques.

Aiming at solving the PDSP problem, many papers have given different disassembly objectives and planning methods. From the perspective of PDSP cost objective, it can be sub-divided into disassembly time and cost<sup>[4,14,15,17]</sup>. Some of the papers have proposed the concept of asynchronous parallel disassembly (APD). APD refer to allowing the workers to start the next disassembling task immediately after completing the current one, without

waiting for others <sup>[16,20]</sup>, which can save more time and improve effectiveness. Table 1 display the related research paper.

Table 1: Related work

Reference	Method	Findings	Limitation
Xing et al <sup>[13]</sup>	Collaborative ant colony algorithm to solve APD	The proposed method was capable of finding the ideal or nearly ideal solution in an acceptable amount of time.	It dependence on the max-min ant system, which could be susceptible to changes in the features of the problem and could not always provide the most effective solutions.
Xianjing et al <sup>[16]</sup>	A genetic algorithm with path reconnection strategy	A number of real-world scenarios demonstrate the algorithm's efficacy. A path reconnecting operator was introduced to reconnect with the elite solution acquired by the genetic algorithm iteration to improve the algorithm's local search capability.	Its capacity to handle intricate spatial constraints was limited, and it could not be scalable for large-scale disassembly issues.
Qiu et al <sup>[19]</sup>	The improved discrete NSGA-II (IDNSGA-II), which have good performance in multi-object APD modules.	As demonstrated by the ideal results, the IDNSGA-II algorithm's multifaceted optimized outcomes for APD series scheduling can be nearly identical to	It became more difficult to effectively realize the discrete numbers of disassembling sequence using the optimization algorithms already in

		the single object optimized results.	place, and computational complexity grew as product complexity increased.
Shao et al <sup>[21]</sup>	This paper presents GAILS, an efficient approach based on genetic algorithms (GA) and iteration local search (ILS) for handling JSSP and FJSSP.	Both JSSP and FJSSP might be processed efficiently by the proposed approach while meeting certain goals.	Its efficiency could be limited in unpredictable, real-time settings due to its possible inability to adapt to extremely dynamic production environments.
Peng et al <sup>[22]</sup>	Multi-Application Scheduling Algorithm (MASA)	It demonstrates show these learning improvements increase the network's capacity, and experimental results demonstrate that MASA performs superior to alternative neural scheduling methods and heuristics.	It is the possible susceptibility to the selection of hyperparameters that which can necessitate cautious adjustment for optimal outcomes in a range of application scenarios.
Mithila et al <sup>[23]</sup>	vector scheduling approach	The tests conducted in CloudLab using nodes distributed across several cloud sites show that incorporating latency into vector scheduling	Despite the benefits associated with modeled delay during communication, an overlay network configuration approach based on delay

		method improves performance and makes better use of available resources.	measures did not work well in a multi-cloud setting.
Francescutto et al <sup>[24]</sup>	Multi-resource Partial-ordering Flexible Job-shop Scheduling (MPF-JSS)	Tests performed on a collection of cases taken from a medium-sized semiconductor fault investigation laboratory show that method was able to locate schedules for 87 of the 91 cases that are thought to be examples from the real world.	The possibility of complexity of computation and limited resources may make expanding up to greater or more complicated production environments difficult.
Li et al <sup>[25]</sup>	Reinforcement learning (RL)based multi-method approach	The proposed approach produced outcomes that were 100% statistically superior to or comparable to those of its component algorithms.	Its reliance on precise specifications for belief degrees, which might be difficult to determine by in extremely dynamic and unpredictable project situations.
Feng et al <sup>[26]</sup>	deep reinforcement learning (RL) method	The efficiency of the proposed algorithm is demonstrated by the outcomes of the experiment.	the difficulties in transferring the performance of the deep reinforced learning model to carrier-borne aviation support activities in the actual world because of the unpredictability and dynamic nature of

			operating environments.
Li et al <sup>[27]</sup>	Grey Wolf Optimization algorithm, and particle swarm optimization	The findings of the sensitivity evaluation demonstrate the created fuzzy computing model's suitability for various ranges of confidence values.	It is believed that operating variables are exclusively represented by fuzzy programming, possibly ignoring intricate real-world variances.

The remaining of the paper is structured as follows: Section 2 give an overview of the disassembly workspace for APD and the previous connections of the APD process. In Section 3, a GA-based technique is proposed to obtain the parallel disassembly sequence. The outcomes of the techniques used in this paper are displayed in Section 4. Section 5 brings the current study to a close.

## 2 Problem statement

### 2.1 Problem representation

In addition to being limited by preceding events, the problem in the task planning is also constrained by multiple resources, which should belong to the parallel disassembly sequence planning (PDSP). Based on the related works above, we can focus on the specific problem as APD (asynchronous parallel disassembly) problem. And based on the constraint of the work space interference and the priority of the task for our problem, we give our methods to define the disassembling space interference matrix and priority task matrix.

#### (a) The definition of disassembling workspace

The Workspace for the project can be divided into 6 categories according to the contents and purposes of the occupied objects: component space, manpower space, equipment space, hazard space, protection space and temporary space <sup>[24]</sup>. Workspace interference refers to the situation where different workspaces occupy each other in actual scenes. Combined with the situation of complex equipment disassembling, in order to quickly obtain the results of the Workspace interference, we used the axial

bounding box to simplify the shape of the our disassembling workspace. According to the working scenario, we give the following definitions for the disassembling workspace:

- **Component space:** The components need to be dismantled in sequence according to the assembly direction and order. The collision interference with other components is not allowed during disassembling.
- **Disassembling operation space:** In the disassembling process, each task has a certain operation space. As shown in Figure 1, we take component **a**. as the bounding box, it sweeping along the task path  $S_k$ , and finally the disassembling operation space is formed as  $\Omega_2$ . Meanwhile, if the task is a manual task by workers, then we can take workers as a bounding box, and their movement paths as  $S_k$ , eventually forming a swept space.

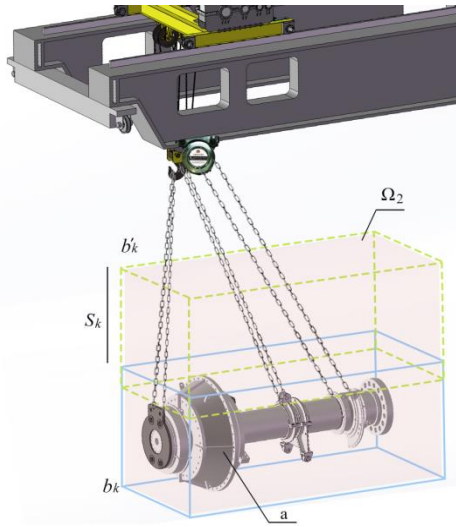


Figure 1: The form of the disassembling operation space.

- Protection space for lifting operations:** When the lifting operation task is executed, the space formed by projection from disassembling operation space onto the working plane, it is called the protection space, as shown in Figure 2,  $\Omega_3$ . From the scene we can see that a worker is carrying out task  $i$  while the crane is also conducting task  $j$ . Due to safety regulations for lifting, workers are not allowed to stay in spaces near where overhead lifting is underway. As shown in the diagram, the worker's disassembling operation space interferes with the protection space. Such a scenario would not be permitted under actual working conditions.

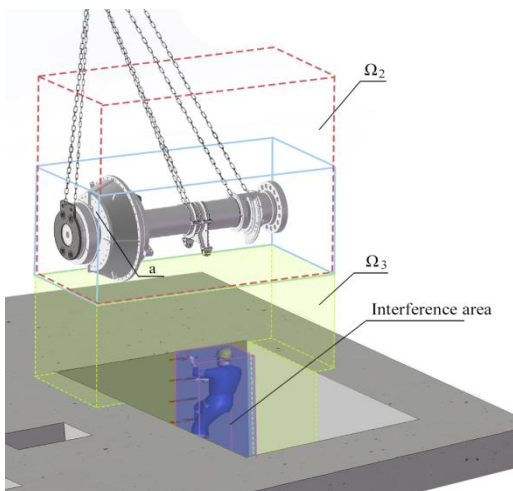


Figure 2: Schematic diagram of disassembling workspace interference.

Based on the above definition of the disassembling workspace, the workspaces be shown in Figure 3.

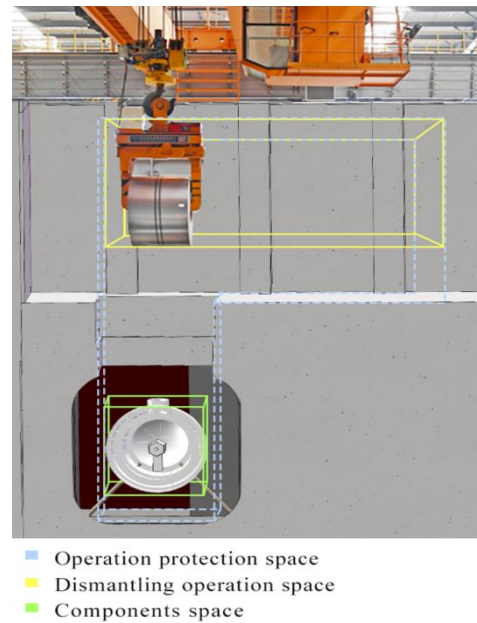


Figure 3. Schematic diagram of disassembling work space definition.

In order to facilitate the expression of the workspace interference, we constructs a disassembling space interference matrix  $\mathbf{C} = [C_{ij}]$ , which can encode the disassembling workspaces of each task with 0/1. Taking the working scene in Figure 3 as an example, the three types of workspaces do not interfere with each other, so the interference situation in the scene in the figure is recorded as 0.

$$C_{ij} = \begin{cases} 1, \text{Task } i \text{ has space interference with Task } j; \\ 0, \text{Task } i \text{ has no space interference with Task } j \end{cases} \quad (1)$$

### 2.2 Task priority matrix

The priority between tasks indicates the disassembling order [28]. For large-scale disassembling tasks, the order is defined according to logical regulations. For example, preparation work must be earlier than disassembling work, and follow by the hoisting task. We define a task priority matrix  $\mathbf{D} = [d_{ij}]$  to represent the priority relationship between the  $i, j$  task of the project.

$$d_{ij} = \begin{cases} 1, \text{Task } i \text{ is the preTask of Task } j; \\ 0, \text{Task } i \text{ is not the preTask of Task } j. \end{cases} \quad (2)$$

❖ **Notation:**

- $P$  — the disassembling project
- $N$  — the set of the tasks (we can set the start and end nodes of the virtual project to be denoted as  $O$  and  $\Theta$ .)
- $P_i, P_j$  — the tasks  $i, j$  in project,  $\forall i, j \in N$
- $n$  — the number of the tasks
- $K$  — the number of the whole workers
- $T_i$  — the time of  $P_i$
- $TS_i$  — the start time of  $P_i$
- $K_i$  — the requiring workers for task  $i$
- $Co_{ij} \in C$  — the disassembling space interference matrix for tasks  $i, j$
- $d_{ij} \in D$  — priority task matrix for tasks  $i, j$
- $t_{end}$  — the project completion time
- $\Gamma_t \subset N$  — the set of tasks being operated at time  $t$

❖ **Assumptions**

the mathematical model is established based on 5 assumptions:

- (1) The spatial requirements of the tasks are been given;
- (2) The supply of other tool-type resources for the tasks is sufficient;
- (3) Tasks must be carried out strictly according to the specified number of personnel. When the number of personnel does not meet the current task requirements, the task cannot start;
- (4) The skills and quality of the workers are basically the same;
- (5) The workers involved in task belong to the same trade.

❖ **Mathematical model**

$$\min t_{end} = P_{\Theta} \tag{3}$$

$$\text{s. t. } d_{ij}(TS_j - T_i - TS_i) \geq 0, \forall i, j \in N \tag{4}$$

$$Co_{ij} = 0, \forall i, j \in \Gamma_t \tag{5}$$

$$\sum_{i \in \Gamma_t} K_i \leq K, \forall i \tag{6}$$

The objective function (1) minimizes the project completion time  $t_{end}$ , namely the start time of the virtual end node. Constraint (2) represents the task precedence constraints, meaning a task's start time should not be earlier than the end time of its pre-tasks. Constraint (3) indicates that parallel tasks cannot have spatial interference at any time; constraint (4) requires that the number of workers assigned to the set of parallel tasks cannot exceed the maximum headcount limit at any time.

### 3 Improved elite genetic algorithm solution

#### 3.1 Coding

Different from the integer coding method adopted in related research on asynchronous parallel task planning<sup>[16, 26, 29]</sup>, we adopt real number coding to make the search space more sufficient. Encode the tasks to generate a random real number vector in the  $[0,1]$  interval as an individual, denoted as  $AL = (a_1, a_2, \dots, a_n)$ ,  $a_i \in [0,1]$ . According to the task priority matrix, the topological sorting method is used to generate a task sequence (Sequence of Tasks) that satisfies the task priority constraints from  $AL$ .

#### 3.2 Decoding

Since the asynchronous parallel tasks also contain multiple constraints, how to decode  $AL$  into a scheduling scheme the constraints become the key to solving the PSDP problem. We combine the idea of parallel topological sorting and designs a scheduling generation mechanism (SGS) based on multi-variable constraints, which innovatively incorporates multi-resource constraint conditions on the traditional serial scheduling generation mechanism (SSS). Traverse every task node in the  $AL$  task sequence, and judge the conditions of time, worker number, and work space interference in turn. Finally, update the task start time table and end time table. The value in the final task end time table is the completion time of the task sequence. The decoding steps are shown in Figure 4.

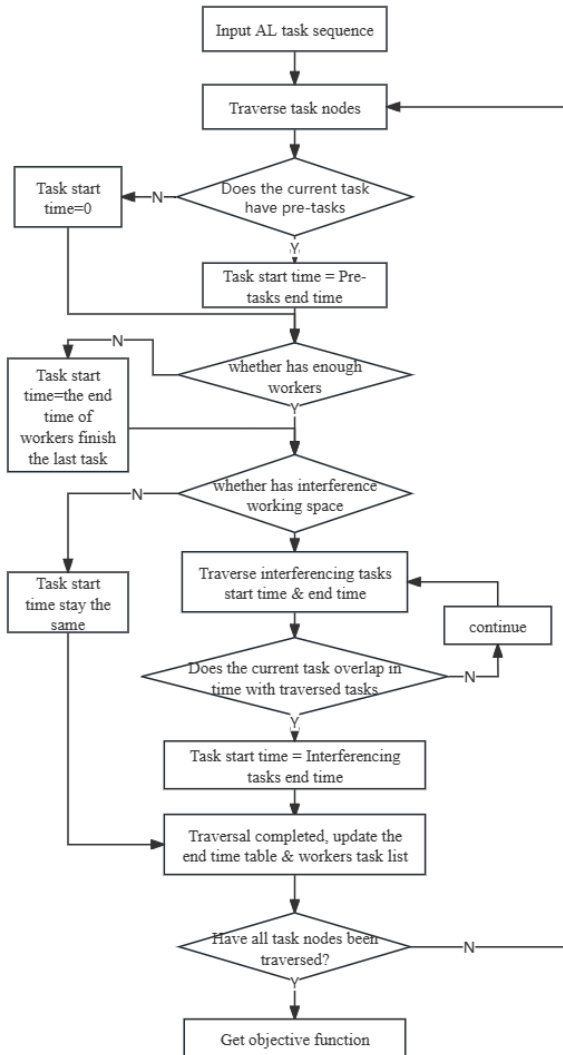


Figure 4: The workflow of decoding.

### 3.3 Genetic operator design

After obtaining the initialized population, the individuals in the population are selected, crossed, and mutated according to the fitness function. We defined the fitness as the objective function, shown in the following formula:

$$fitness = -t_{end} \quad (5)$$

$t_{end}$  indicates the last task completion time, which means that the shorter the task completion time of individuals with higher fitness in the population.

We choose the most widely used tournament selection method for genetic operators. In each generation population, two individuals are randomly selected for

comparison, and the individual with greater individual fitness remains in the next generation.

The crossover genetic operator pairs the individuals in the population, and exchanges part of the chromosome according to the crossover probability; the mutation genetic operator changes the gene values at some positions of the chromosome to other alleles. In the crossover operator, single-point crossover is adopted to exchange fragments between chromosomes at random positions. The mutation operator adopts single-point mutation, which adds a disturbance value to a random position of the chromosome to change the order of some tasks so that it can increase the diversity of the population. The adaptive crossover and mutation probability formulas defined in this paper are as follows:

$$p_c = k_c - \frac{f' - f_{avg}}{f_{max} - f_{min}} \lambda \quad (6)$$

$$p_m = k_m - \frac{f_i - f_{avg}}{f_{max} - f_{min}} \mu \quad (7)$$

$p_c$  is the adaptive crossover probability;  $k_c$  is the basic crossover probability;  $\lambda$  is the crossover factor;  $f'$  is the larger fitness value of the two individuals for crossover;  $k_m$  is the basic mutation probability;  $\mu$  is the mutation factor;  $f_i$  is the current individual fitness. When the population fitness tends to the local optimal value, appropriately reduce the crossover probability and increase the mutation probability to prevent premature convergence of the algorithm; when the fitness difference between chromosomes is relatively large or individuals in the population are dispersed in the solution space, the formula can appropriately increase the crossover probability and reduce the mutation probability, so that superior individuals can be retained to help the population conduct a more extensive search in the solution space.

### 3.4 Ant colony search operator

In order to further expand the search scope of the solution space, this paper designs an ant colony search operator, which compares the fitness of individuals in the population, and performs local search on suboptimal individuals and global search on inferior individuals through the calculation of state transition probability in the ant colony algorithm, so as to obtain higher quality populations. The operator adopts the pheromone search idea based on the ant colony algorithm. First, the fitness values of the new population after genetic manipulation are set as the initial pheromone, and the state transition probability of the population is calculated. The state transition probability formula is as follows:



$$p_i = -\frac{\tau_{max} - \tau_i}{\tau_{max}} \quad (8)$$

$p_i$  indicates the state transition probability of individual  $i$ ;  $\tau_{max}$  indicates the maximum pheromone value of the population;  $\tau_i$  is the current individual pheromone value.  $p_0$  is a constant for state transition probability. When the state transition probability  $p_i$  is less than  $p_0$ , a local search operation is performed; when the state transition probability  $p_i$  is greater than  $p_0$ , a global search operation is performed to generate new individuals. The search strategy formula is as follows:

$$temp = p_i + \varepsilon, \varepsilon \in [-\alpha\lambda, \alpha\lambda] \quad (9)$$

$temp$  is the temporary individual vector after searching,  $p_i$  is the population individual,  $\alpha$  indicates the local search step size,  $\lambda$  indicates the scaling coefficient. For the suboptimal individuals ( $p_i \geq p_0$ ) in the population,  $\lambda$  take a smaller value for local optimal search; for the inferior individuals ( $p_i < p_0$ ) in the population,  $\lambda$  take a larger value, with the purpose of global search for individuals. The obtained is processed for boundary conditions, and the solution value is defined within the encoding range of  $[0, 1]$ , and its fitness is calculated. If the fitness value is superior to  $p_i$ , the population individual is updated.

After the ant colony search is completed, the parent and offspring populations are retained to accelerate the convergence speed of the algorithm and improve the quality of the solution. In this paper, the individuals in the parent and offspring populations are sorted according to the fitness, and the top 50% of individuals with fitness are selected as the new population. Then the pheromone matrix is updated to record the current optimization trend.

### 3.5 Algorithm steps and processes

**Step 1:** Initialize the system, including population size, number of iterations, crossover probability, mutation probability and adaptive coefficients.

**Step 2:** Construct the initial solution to the problem, and generate an initial feasible solution population according to the constraint conditions.

**Step 3:** Judge the iterative circulation condition. Exit the operation if the maximum number of iterations is reached; otherwise, go to Step 4.

**Step 4:** Perform selection, crossover, mutation, ant colony search, and elite retention operations on the population

according to preset probabilities, respectively.

**Step 5:** Increment the number of iterations and jump to Step 3.

## 4 Application and discussion

### 4.1 Application

There are thirty-one tasks that need to be completed in order to disassemble a bulb cylindrical propellers generator unit. Four regular workers and one crane operator are assigned to complete the jobs. Table 3 provides basic details about the activities and people, and Figure 5 shows the workstation where tasks 1 through 10 are to be disassembled. The typical parameters presented in Table 2 are essential components of the project's requirements that are critical to defining its operational restrictions and framework. This disassembly project description, which is based on task specifics, worker assignments, and parameter settings, lays the foundation for effective planning and execution.

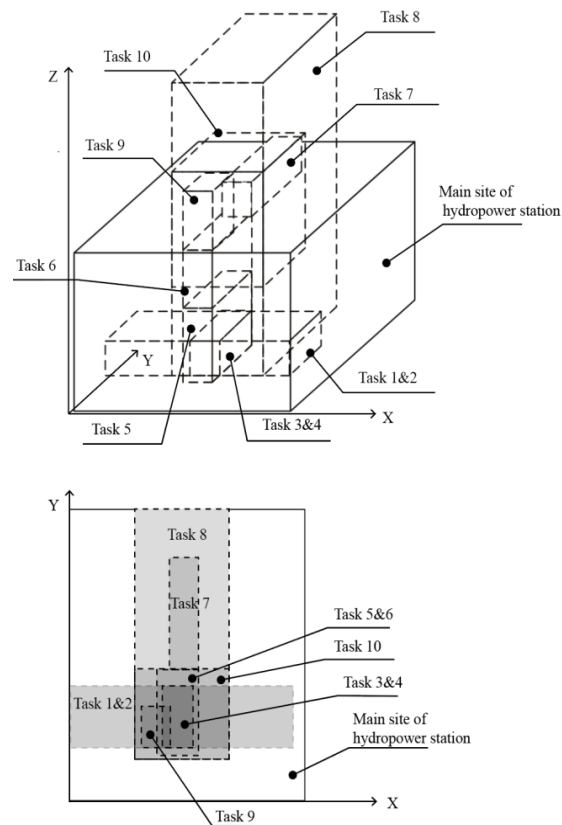


Figure 5: Disassembling work task space (Tasks 1-10) scene diagram

Table 2: Experimental parameter settings

NO.	Parameter
1	Population size = 8
2	Maximum number of iterations = 100
3	Basic crossover probability = 0.7
4	Basic mutation probability = 0.1
5	Crossover factor = 0.2
6	Mutation factor = 0.05
7	State transition constant = 0.2
8	Search step size = 0.2
9	Scaling coefficient [0.5, 2.5]

Table 3: Disassembling work task information

Num.	The description of tasks	Pre-tasks	Time (h)	workers	Space interference
1	Assist in draining the upstream channel and unblocking the pipeline	-	4	3	5,8,13,17,14,23,27,28
2	Clean up silt in upstream channel	-	6	3	3,5,6,12,13,14,15,17,23,24,27,28
3	Support marking for bulb head	1,2	2	2	4,5,7,8,13,14,15,17,23,24,27,28
4	Remove bulb head support bolts	3	2	3	5,8,13,14,17,23,24,27,28
5	Dismantle bulb head support	1,2,4	2	4	8,13,14,17,23,24,27,28

6	Remove bulb head assembly surface bolts	2	4	2	8,13,17,19,21,22,31
7	Prepare steel wires and four shackles for lifting	-	2	2	-
8	Auxiliary hook rises; the gantry crane moves to the pit	7	0.5	0	11,12,13,15,17,19,20,21,22,27,29
9	Remove shackles and connect thin steel wires	8	0.5	1	10,11,15,17,29
10	Descend auxiliary hook	9	0.5	0	11,12,15,17,27
11	Install four shackles	10	0.5	1	15,17,27,29
12	Main hook hangs rope	5	0.5	1	15,17,27,29
13	Install manual pulley block	5	0.5	1	15,17,22,24
14	Install jack under bulb head	5	0.5	1	17,23,28
15	Hang manual pulley blocks on two lifting points upstream of bulb head	14	0.5	1	17,22,24
16	Hang a pulley block on hanger ear of flow channel wall	-	0.5	1	16,17,23,27,30
17	Crane starts; gantry crane bears	6,11,12,13,15,16	0.5	0	19,21,22,23,27,29,30
18	Check stress status of each point	6,11,12,13,15,16	0.5	1	19,21,22,23,27,29,30
19	Remove remaining bolts between assemblies	17,18	1.5	2	21,22,27,30,31
20	Return tools and removed parts	17,18	0.5	1	23,27,30

	to corresponding positions				
21	Remove reserved bolts for bulb head	19,20	1	2	22,23
22	Pull the pulley block to separate bulb head from stator bonding surface	21	0.5	1	23
23	Crane lifts bulb head and places it on squared timbers	22, 27	1	0	27,28,30
24	Take measurements	23	0.5	1	27,30
25	Fabricate steel bracket	-	1	1	-
26	Prepare squared timbers	-	0.5	1	-
27	Lift squared timbers and steel bracket to bulb head placement	25,26,1 7	0.5	0	29
28	Full welding for steel bracket	23	0.5	1	30
29	Loosen main hook and remove steel wires	24	0.5	1	30
30	Lift bulb head bolts to installation room	29	0.5	0	31
31	Apply penetrating oil on flange surfaces between assemblies	23	0.5	1	-

## 4.2 Discussion

The IGA algorithm was executed to solve the case. After iterative convergence, the total duration of the optimal scheduling scheme generated is 25.5h. The Gantt chart is shown in the figure 6. No. 5 rows indicates the crane driver's task, and the other rows indicate ordinary workers. The width of the color blocks corresponds to the task time; the gap indicates that the worker is waiting, and the length of the gap corresponds to the time the worker waits before performing the next disassembling task. Our proposed method outperforms existing approaches such as

Collaborative ant colony algorithm to solve APD, A genetic algorithm with path reconnection strategy, The improved discrete NSGA-II (IDNSGA-II), genetic algorithms (GA) and iteration local search (ILS) for handling JSSP and FJSSP, Multi-Application Scheduling Algorithm (MASA), vector scheduling approach and Multi-resource Partial-ordering Flexible Job-shop Scheduling (MPF-JSS), demonstrating superior performance in the comparison of results.

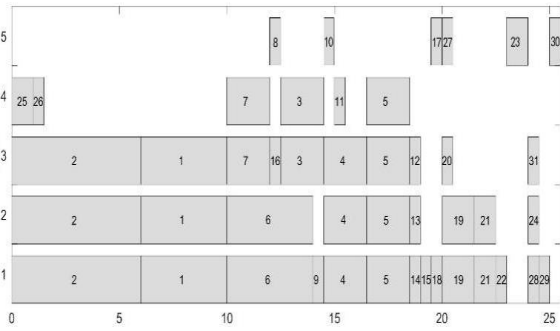


Figure 6: Gantt chart of optimal solution

### 4.3 Influence of resource constraints on solutions

Compare the results of this algorithm with two cases without considering human resource constraints and space interference constraints.

(1) With the same initial parameters, masking the human resource conditions, the results obtained are shown in Figure 7. Although it can be seen that the solving conditions are more relaxed, the task execution time is shorter, and the project end time is 19h, the figure shows that nearly 10 people are needed to complete the related task arrangements at the beginning of task planning, while only 4-5 people are needed to complete the project later in the schedule.

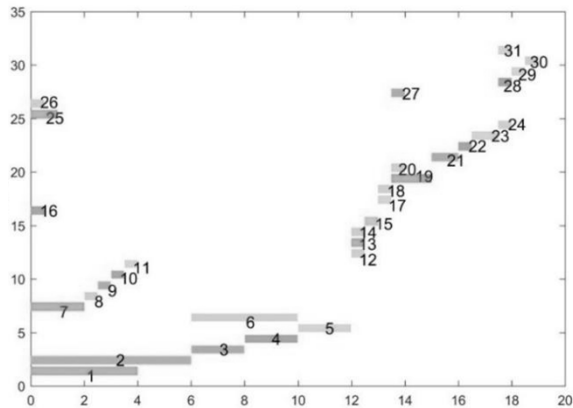


Figure 7: Disassembling task Gantt chart without considering personnel constraints

(2) With the same initial parameters, shielding the space interference constraints, the results obtained are shown in Figure 8. The task execution time is 25h, which is slightly shorter than the time planning of this algorithm. However, tasks 4 and 8 are planned to be carried out at the same time. According to the data in Table 2, there is a situation of workspace interference between tasks 4 and 8. While the crane is traveling to the pit and descending, the worker of

task 4 is working in the pit, which conflicts with our problem definition.

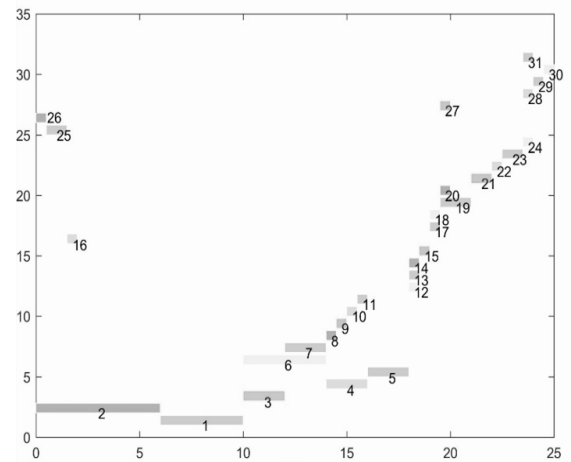


Figure 8: Disassembling task Gantt chart without considering spatial constraints

In summary, by introducing human resource and workspace constraints, a disassembling task sequence that is more in line with the actual situation can be obtained, which effectively avoids redundant personnel and space resource conflict issues in DSP.

### 4.4 Algorithm performance analysis

The improved elite genetic algorithm (IGA) proposed in this paper is compared experimentally with the traditional ant colony algorithm (ACO) and the adaptive genetic algorithm (AGA). Figure 9 display the comparison of algorithm. It can be seen that the local search strategy designed in this paper enables the algorithm to jump out of the local optimal value in terms of optimization, and compared with the ant colony algorithm, the improved genetic algorithm adopted in this paper can also find better quality optimal solutions faster.

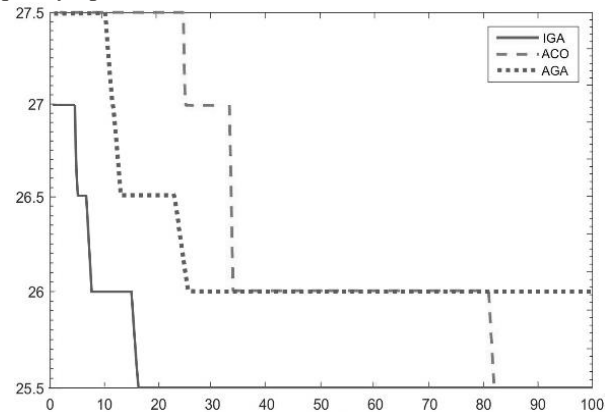


Figure 9: The algorithm comparison.

Precision is a fundamental parameter utilized in the fields of statistics to evaluate performance. Figure 10 depicts the comparative evaluation of precision in suggested and traditional methods. When compared to currently existing methods such as IGA and ACO, which have Precision values of 85% and 82%, respectively, the suggested AGA achieves a precision value of 90%.

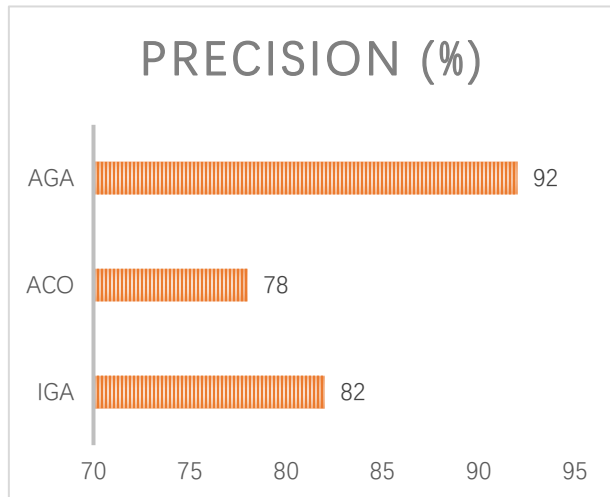


Figure 10: Precision recall

Recall is a performance metric used in data categorization that represents the proportion of actual positives that a model properly retrieves. The recall result is shown in Figure 11. When compared to currently existing methods such as IGA and ACO, which have Recall values of 88% and 85%, respectively, the suggested AGA achieves an accuracy value of 92%.

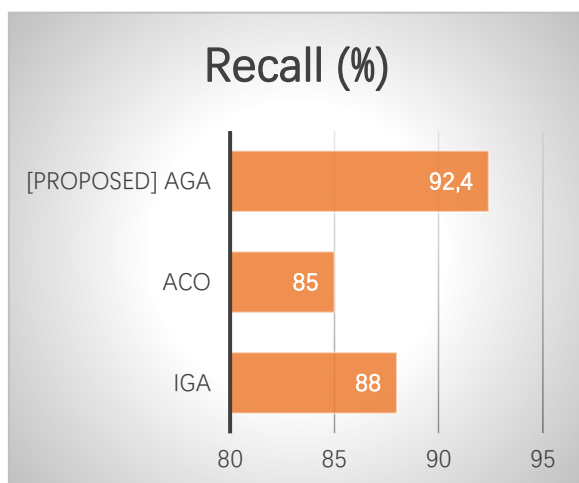


Figure 11: Recall

## 5 Conclusions

Aimed at the multi-resource constraint problem in asynchronous parallel task planning, this paper establishes a multi-resource constrained task scheduling mathematical programming model from the perspective of worker roles, working space, and limited personnel resources. An IGA algorithm is designed to solve this problem, and a decoding method based on parallel topological sorting is proposed to obtain the minimum project duration for tasks under multi-resource conditional constraints. On the basis of the elite genetic algorithm, combined with the advantage of local optimization of the ant colony algorithm, an ant colony local search operator based on pheromone is proposed to improve the search space and solution quality. The novel decoding technique and the proposed IGA algorithm have tremendous potential for a range of industrial uses. The optimum task scheduling mathematical model could be useful for industries like manufacturing, building, construction, and logistics that include complex project scheduling. The incorporation of a pheromone-based ant colony local search operator broadens the algorithm's industrial value by improving its adaptability to real-world scenarios and indicating its applicability in contexts that are dynamic and resource-intensive.

Through the hydraulic turbine hoisting and disassembling example, the effectiveness of the algorithm and improvement measures in this paper is verified by analyzing the key parameter settings. The experimental results show that the multi-resource constrained asynchronous parallel task planning proposed in this paper is more practical, effectively avoiding the safety threats and resource conflicts of tasks. The problems, models and algorithms studied in this paper can be customized for resource requirements according to the specific project characteristics, and have certain versatility and scalability. It has guiding significance for scheduling disassembling work tasks of large equipment. In future research, asynchronous parallel tasks can be further studied in terms of project scheduling costs. The proposed method may not be generalizable beyond the particular hydraulic turbine instance, and the algorithm's performance may be sensitive to changes in job parameters, requiring careful thought and modification for various disassembly tasks.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare no conflicts of interest

## Funding Statement

This study did not receive any funding in any form.

## References

- [1] Zhang Z, Yuan B, Zhang Z. A new discrete double-population firefly algorithm for assembly sequence planning [J]. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2016, 230(12): 2229-38.
- [2] Gong C. Instructional Design and Application of Hydraulic Turbine Based on MR (Mixed Reality) HoloLens 2; proceedings of the 2020 International Conference on *Information Science and Education (ICISE-IE)*, F, 2020 [C]. *IEEE*.
- [3] Smith S, Chen W-H. Multiple-target selective disassembly sequence planning with disassembly sequence structure graphs; proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, F, 2012 [C]. *American Society of Mechanical Engineers*.
- [4] Zhou Z, Liu J, Pham D T, et al. Disassembly sequence planning: Recent developments and future trends [J]. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2018, 233(5): 1450-71.
- [5] Zhang X. Single object selective disassembly sequence planning based on ant colony algorithm [J]. *Computer Integrated Manufacturing System*, 2007, 13(06): 0.
- [6] Shimizu Y, Tsuji K, Nomura M. Optimal disassembly sequence generation using a genetic programming [J]. *International Journal of Production Research*, 2007, 45(18-19): 4537-54.
- [7] Kongar E, Gupta S M. Disassembly sequencing using genetic algorithm [J]. *The International Journal of Advanced Manufacturing Technology*, 2006, 30(497-506).
- [8] Pornsing C, Watanasungsuit A. Discrete particle swarm optimization for disassembly sequence planning; proceedings of the 2014 IEEE International Conference on Management of Innovation and Technology, F, 2014 [C]. *IEEE*.
- [9] Xia K, Gao L, Li W, et al. Disassembly sequence planning using a simplified teaching-learning-based optimization algorithm [J]. *Sustainable Manufacturing and Remanufacturing Management: Process Planning, Optimization and Applications*, 2019, 319-43.
- [10] Agrawal D, Pande S. Automatic disassembly sequence planning and optimization [J]. *Dual Degree Thesis, IIT Bombay*, 2011,
- [11] Xia K, Gao L, Li W, et al. A Q-learning based selective disassembly planning service in the cloud based remanufacturing system for WEEE; proceedings of the International Manufacturing Science and Engineering Conference, F, 2014 [C]. *American Society of Mechanical Engineers*.
- [12] Chunming Z. Optimization for disassemble sequence planning of electromechanical products during recycling process based on genetic algorithms [J]. *International Journal of Multimedia and Ubiquitous Engineering*, 2016, 11(4): 107-14.
- [13] Xing Y, Wu D And Qu, L. Parallel disassembly sequence planning using improved ant colony algorithm. *The International Journal of Advanced Manufacturing Technology*, 2021, 113, 2327-2342.
- [14] SMITH S, HUNG P-Y. A novel selective parallel disassembly planning method for green design [J]. *Journal of engineering design*, 2015, 26(10-12): 283-301.
- [15] Ren Y, Tian G, Zhao F, et al. Selective cooperative disassembly planning based on multi-objective discrete artificial bee colony algorithm [J]. *Engineering Applications of Artificial Intelligence*, 2017, 64(415-31).
- [16] Xianjing S, Qiuhua T, Mingxing D. Asynchronous Parallel Disassembly Sequence Planning Based on Improved Genetic Algorithm [J]. *Industrial Engineering Journal*, 2022, 25(04): 151-7.
- [17] Kaijun C, Meijun Z, Li J, et al. Disassembly sequence planning for multi-people simultaneous operation [J]. *Computer Integrated Manufacturing Systems*, 2016, 22(12): 2767-77.
- [18] Aoyuan Z, Xiaofeng H, Yahui Z. Rescheduling of Multi-scenario and Multi-Objective Dynamic Changes of Ship Group Construction [J]. *Journal of Shanghai Jiaotong University*, 2023, 1-16.
- [19] Qiu L, Dong L, Wang Z, et al. Asynchronous parallel disassembly sequence planning method of complex products using discrete multi-objective optimization [J]. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2022, 236(11): 1466-82.
- [20] Guo X, Bi Z, Wang J, et al. Reinforcement Learning for Selective Disassembly System Optimization Problems: A Survey [J]. *International Journal of Network Dynamics and Intelligence*, 2023,
- [21] Shao X, Kshitij F S And Kim C S. Gails: an effective multi-object job shop scheduler based on genetic algorithm and iterative local search. *Scientific Reports*, 2024, 14(1), 2068.
- [22] Peng Q and Wang S. Masa: Multi-Application Scheduling Algorithm for Heterogeneous Resource Platform. *Electronics*, 2023, 12(19), 4056.

- [23] Mithila S P, Franz P And Baumgartner G. Scheduling Many-Task Applications on Multi-clouds and Hybrid Clouds. In Workshop on Asynchronous Many-Task Systems and Applications, 2023, 65-78. Cham: Springer Nature Switzerland.
- [24] Francescutto G, Schekotihin K And El-Kholany M M. Solving a multi-resource partial-ordering flexible variant of the job-shop scheduling problem with hybrid ASP. In *European Conference on Logics in Artificial Intelligence*, 2021, 313-328.
- [25] Li Y, Zhang X, Zeng T, Duan, J, Wu C, Wu D And Chen X. Task placement and resource allocation for edge machine learning: A gnn-based multi-agent reinforcement learning paradigm, 2023, *arXiv preprint arXiv:2302.00571*.
- [26] Feng H And Zeng W. Deep Reinforcement Learning for Carrier-borne Aircraft Support Operation Scheduling. In 2021 International Conference on Intelligent Computing, *Automation and Applications (ICAA)*, 2021, 929-935. *IEEE*.
- [27] Li B, Chen Q, Lau Y Y And Dulebenets M A. Tugboat Scheduling with Multiple Berthing Bases under Uncertainty. *Journal of Marine Science and Engineering*, 2023,11(11), p.2180.
- [28] Ren Y, Zhang C, Zhao F, et al. An asynchronous parallel disassembly planning based on genetic algorithm [J]. *European Journal of Operational Research*, 2018, 269(2): 647-60.
- [29] Bai-Lin L, Pan-Qi W, Zhi-Chao W, et al. Asynchronous parallel disassembly sequence planning base on improved gravitational search algorithm [J]. *Journal of Mechanical&Electrical Engineering*, 2023, 40(01): 113-21.