

Original Paper

Task Decomposition for Cloud Manufacturing under Industrial Internet Using an Improved Genetic Algorithm

Yan Wang¹

¹ Business School, University of Shanghai for Science and Technology, Shanghai, China

Received: March 19, 2026

Accepted: April 22, 2026

Online Published: May 9, 2026

doi:10.22158/mmse.v8n2p236

URL: <http://dx.doi.org/10.22158/mmse.v8n2p236>

Abstract

In the industrial internet environment, task decomposition and reconfiguration are key links connecting user demands with manufacturing resources in a cloud manufacturing platform. To address the issues of traditional methods relying on process logic while neglecting enterprise capability differences, a capability matching-based multi-dimensional task decomposition method is proposed. A multi-dimensional task correlation model covering technological, capability, geographical, and cost correlations is constructed, followed by a multi-objective reconfiguration optimization model targeting efficiency, cost, and quality. For efficient solution, an improved genetic algorithm is designed (integer encoding, order-preserving crossover, heuristic mutation, and adaptive parameter adjustment). A case study on aero-engine impeller manufacturing demonstrates that, while maintaining the optimal makespan, the optimized solution reduces cost by 5.69% and improves quality by 0.33%, verifying the effectiveness of the proposed method.

Keywords

cloud manufacturing, task decomposition, capability matching, improved genetic algorithm

1. Introduction

With the in-depth development of industrial internet technologies and the transformation and upgrading of manufacturing paradigms, cloud manufacturing, as a service-oriented, on-demand collaborative manufacturing paradigm, is gradually becoming a key enabler for the socialized sharing and intelligent allocation of manufacturing resources (Zhang C, He K, Liu Q, et al., 2025). In a cloud manufacturing system, complex manufacturing demands submitted by users typically need to be decomposed into several independently executable subtasks (i.e., meta-tasks), and these subtasks are further rationally reorganized to form task units that can be explicitly assigned to different service providers. Task decomposition and reconfiguration thus serve as the "bridge" connecting user demands with

manufacturing resources, and their rationality directly determines the service efficiency, resource allocation quality, and overall coordination level of the cloud manufacturing platform.

Early research on task decomposition mainly originated from the fields of project management and complex system design, among which the most representative method is the Work Breakdown Structure (WBS). WBS emphasizes decomposing an overall task into manageable work units in a hierarchical, top-down manner, which facilitates scope definition, responsibility allocation, and schedule control (Globerson, 1994). However, this type of method is more suitable for task management scenarios within a single organization with relatively clear boundaries, and offers limited support for complex coupling relationships between tasks and subsequent resource coordination. With the advancement of complex product development and systems engineering research, scholars gradually extended task decomposition research from "hierarchical splitting" to "relationship modeling". Steward (Steward, 1981) proposed the Design Structure Matrix (DSM) method, which uses a matrix to describe dependencies among tasks, providing a structured tool for the decomposition and reconfiguration of complex systems. Browning (Browning, 2001) systematically summarized the application of DSM in system decomposition, integration, and reconfiguration, arguing that DSM can effectively support dependency identification, module partitioning, and organizational optimization for complex tasks. In the cloud manufacturing research stage, task decomposition has gradually shifted from structure splitting oriented to the task process itself to comprehensive studies integrating task granularity, task correlations, and the alignment between task decomposition and resource allocation. Wang et al. (2014) proposed a semantic modeling and description method for manufacturing tasks in cloud manufacturing systems, constructed a task semantic expression framework, and demonstrated that standardized task description can provide a foundation for subsequent task parsing and resource matching. Yi et al. (2015) proposed a task decomposition optimization method based on clustering algorithms. On the basis of preliminary task decomposition, they comprehensively considered inter-task correlations, task-resource matching, and resource competition to reorganize and optimize subtasks. The results show that this method improves the connection between task decomposition outcomes and resource allocation, and reduces the complexity of subsequent task-resource matching problems. Subsequently, Liu et al. (2017) proposed a cloud manufacturing task decomposition method based on Hierarchical Task Network (HTN), studied manufacturing task granularity analysis, task cohesion measurement, and task correlation measurement, and adopted a recursive decomposition algorithm for optimized task decomposition while considering task-resource matching during the decomposition process. In the most recent study, Zhang et al. (2026) further incorporated resource compatibility and resource competition into the task decomposition process, proposing a cloud manufacturing task decomposition method that considers resource compatibility and competition. This indicates that task decomposition research is gradually evolving toward integration with resource matching and subsequent configuration optimization.

Existing research on cloud manufacturing task decomposition suffers from the following main

limitations: First, most methods still follow the traditional single-dimensional decomposition approach, lacking systematic consideration of multi-enterprise collaboration characteristics in the industrial internet environment. Second, the correlation analysis among decomposed meta-tasks is insufficiently deep, making it difficult to effectively guide the rational combination and optimal allocation of tasks. Third, there is a lack of effective reconfiguration optimization mechanisms to adjust and improve the decomposition scheme based on actual constraints. To address these issues, this chapter proposes a cloud manufacturing task decomposition and reconfiguration optimization method for the industrial internet. Specifically, considering the complexity characteristics of cloud manufacturing tasks in the industrial internet context, a capability matching-based multi-dimensional task decomposition method is first proposed. Then, a meta-task correlation description model and a comprehensive correlation measurement method are constructed to quantitatively analyze the relationships among meta-tasks. Finally, a multi-objective reconfiguration optimization model considering efficiency, cost, and quality is established, and an improved genetic algorithm is employed to obtain a superior task organization scheme.

2. Meta-task Decomposition and Correlation Modeling

2.1 Cloud Manufacturing Task Decomposition Method

This paper proposes a capability matching-based multi-dimensional task decomposition method. This method breaks through the limitation of traditional WBS which solely relies on process logic for decomposition, and takes enterprise capabilities as an important basis for decomposition, thereby achieving effective alignment between task requirements and enterprise capabilities. The task decomposition follows a four-stage process: "requirement analysis – capability mapping – boundary determination – scheme generation". In the requirement analysis stage, the original manufacturing task is structurally described, clarifying key parameters such as technical requirements, quality standards, time constraints, and cost budgets. In the capability mapping stage, task requirements are matched with enterprise capabilities within the industrial network, identifying candidate enterprise sets that possess the corresponding capabilities. In the boundary determination stage, based on capability matching results and coordination cost considerations, the decomposition granularity and boundaries of meta-tasks are determined. In the scheme generation stage, a complete task decomposition scheme is formed.

The definition of meta-tasks follows three basic criteria: functional integrity, execution independence, and interface standardization. Functional integrity requires that each meta-task has a clear functional objective and verifiable output results. Execution independence ensures that a meta-task can be completed independently by a single enterprise or a tightly collaborating group of enterprises, without requiring real-time coordination with other meta-tasks. Interface standardization requires that information and material exchange between meta-tasks be conducted through standardized interfaces.

Suppose the original manufacturing task T can be decomposed into n meta-tasks, denoted as formula

(1):

$$T = \{T_1, T_2, \dots, T_n\} \quad (1)$$

Each meta-task T_i has the following four attributes, denoted as formulas (2) to (5) respectively.:

$$F_i = f_{i1}, f_{i2}, \dots, f_{im} \quad (2)$$

$$R_i = r_{i1}, r_{i2}, \dots, r_{ik} \quad (3)$$

$$D_i = \{t_{start}, t_{end}\} \quad (4)$$

$$Q_i = q_{i1}, q_{i2}, \dots, q_{ip} \quad (5)$$

Where f_{ij} denotes the functional requirement item, r_{ik} denotes the resource requirement item, D_i denotes the time window, and q_{ip} denotes the quality indicator.

2.2 Meta-task Correlation Description Model

Correlations among meta-tasks are a key factor affecting the effectiveness of task reconfiguration. In the industrial internet environment, multiple types of correlations exist among meta-tasks, including technological correlation, capability correlation, geographical correlation, and cost correlation. This paper establishes a multi-dimensional correlation description model to comprehensively characterize the complex relationships among meta-tasks.

Technological correlation is mainly reflected in the logical relationships of process flows. Some meta-tasks have strict sequential relationships and must be executed in a specific order; some meta-tasks have parallel relationships and can be carried out simultaneously; others exhibit complementary or conflicting relationships in technological implementation. Technological correlation can be represented by an adjacency matrix, as shown in formula (6).

$$R^{tech} = [r_{ij}^{tech}]_{n \times n} \quad (6)$$

When task i must be completed before task j , $r_{ij}^{tech} = 1$, otherwise, it is 0.

Capability correlation reflects the similarity of requirements for enterprise capabilities among different meta-tasks. If two meta-tasks require similar technical capabilities or equipment resources, they have a high capability correlation and are suitable to be undertaken by the same enterprise or by enterprises with similar capabilities. Capability correlation can be measured by the similarity of capability requirement vectors, as shown in formula (7).

$$r_{ij}^{cap} = \frac{F_i \cdot F_j}{\|F_i\| \cdot \|F_j\|} \quad (7)$$

Where F_i and F_j are the capability requirement vectors of tasks i and j , respectively, " \cdot " denotes the vector inner product, and $\|\cdot\|$ denotes the vector norm.

Geographical correlation considers the impact of the spatial distribution of meta-task execution locations on collaboration effectiveness. Tasks that are geographically close have natural advantages in material flow, information communication, and coordination management. Geographical correlation can be expressed as the reciprocal of distance, as shown in formula (8).

$$r_{ij}^{geo} = \frac{1}{1 + d_{ij}} \quad (8)$$

Where d_{ij} is the geographical distance between the execution locations of tasks i and j .

Cost correlation describes the potential cost synergy effect that may arise when meta-tasks are executed jointly. When multiple tasks are undertaken by the same enterprise or executed in close proximity, economies of scale and synergy effects can often be achieved, thereby reducing the overall cost. Cost correlation can be measured by the cost saving ratio, as shown in formula (9).

$$r_{ij}^{cost} = \frac{C_i + C_j - C_{ij}}{C_i + C_j} \quad (9)$$

Where C_i and C_j are the costs of executing tasks i and j independently, respectively, and C_{ij} is the total cost when they are executed jointly.

2.3 Meta-task Correlation Measurement Method

To quantitatively analyze the correlation strength among meta-tasks, this paper establishes a comprehensive correlation measurement model. This model performs a weighted combination of multi-dimensional correlations to obtain a comprehensive correlation index between meta-tasks.

The calculation formula for the comprehensive correlation strength is given by formula (10).

$$R_{ij} = w_1 \cdot r_{ij}^{tech} + w_2 \cdot r_{ij}^{cap} + w_3 \cdot r_{ij}^{geo} + w_4 \cdot r_{ij}^{cost} \quad (10)$$

where w_1, w_2, w_3, w_4 are the weight coefficients of each dimensional correlation, and they satisfy formula (11).

$$\sum_{k=1}^4 w_k = 1, \quad w_k \geq 0 \quad (11)$$

The determination of weight coefficients can be achieved by methods such as expert judgment, analytic hierarchy process, or principal component analysis. Different application scenarios can set different combinations of weights to reflect decision-makers' emphasis on various correlations.

The correlation matrix R is an $n \times n$ symmetric matrix, as shown in formula (12).

$$R = [R_{ij}]_{n \times n} \quad (12)$$

The eigenvalues and eigenvectors of this matrix can reveal the clustering structure and correlation patterns among meta-tasks, providing important references for subsequent reconfiguration optimization.

3. Meta-task Reconfiguration Optimization Model

3.1 Objective Function Design

The objective of meta-task reconfiguration optimization is to find the optimal task organization scheme while satisfying various constraints. Considering the characteristics of the industrial internet environment, this paper establishes a multi-objective optimization model including three main objectives: efficiency, cost, and quality.

The efficiency objective focuses on minimizing the overall execution time. In the task reconfiguration process, factors such as dependencies among tasks, resource constraints, and coordination time affect the overall efficiency. The efficiency objective function is shown in formula (13).

$$\min f_1 = \max_{i=1}^n (t_{start}^i + d_i) \quad (13)$$

Where t_{start}^i is the start time of task i , and d_i is the execution duration of task i .

The cost objective pursues the minimization of the total cost, including both direct execution cost and coordination management cost. The direct execution cost is directly related to the task assignment scheme, while the coordination management cost is related to the collaboration complexity among tasks. The cost objective function is shown in formula (14).

$$\min f_2 = \sum_{i=1}^n C_i^{exec} + \sum_{i=1}^n \sum_{j=i+1}^n R_{ij} \cdot C_{ij}^{coord} \quad (14)$$

Where C_i^{exec} is the execution cost of task i , C_{ij}^{coord} is the coordination cost between tasks i and j , and R_{ij} is the correlation strength.

The quality objective ensures that the reconfiguration scheme meets the expected quality requirements. The quality objective is measured by the weighted average of the quality indicators of each meta-task, as shown in formula (15).

$$\max f_3 = \frac{\sum_{i=1}^n w_i \cdot Q_i}{\sum_{i=1}^n w_i} \quad (15)$$

Where Q_i is the quality achievement degree of task i , and w_i is the importance weight of task i .

Normalization is performed through formulas (16) to (18), and the multi-objective optimization problem is transformed into a single-objective problem using formula (19).

$$f_{1'} = \frac{f_1 - f_1^{\min}}{f_1^{\max} - f_1^{\min}} \quad (16)$$

$$f_{2'} = \frac{f_2 - f_2^{\min}}{f_2^{\max} - f_2^{\min}} \quad (17)$$

$$f_{3'} = \frac{f_3^{\max} - f_3}{f_3^{\max} - f_3^{\min}} \quad (18)$$

$$\min F = \alpha_1 \cdot f_{1'} + \alpha_2 \cdot f_{2'} + \alpha_3 \cdot f_{3'} \quad (19)$$

Where $\alpha_1, \alpha_2, \alpha_3$ are objective weight coefficients, and they satisfy $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_i \geq 0$.

3.2 Constraint Formulation

Meta-task reconfiguration optimization needs to satisfy multiple constraints including technological constraints, resource constraints, time constraints, and capability constraints. Technological constraints ensure that the reconfiguration scheme complies with the basic requirements of the manufacturing process, as shown in formula (20).

$$t_{finish}^i \leq t_{start}^j, \quad \forall (i, j) \in P \quad (20)$$

Where P is the set of task pairs with precedence relationships.

Resource constraints ensure that at any time point, resource usage does not exceed the available capacity, as shown in formula (21).

$$\sum_{i \in T_t} r_{ik} \leq R_k^{\max}, \quad \forall k \in K, t \in [0, T_{\max}] \quad (21)$$

Where T_t is the set of tasks being executed at time t , r_{ik} is the demand of task i for resource k , and

R_k^{\max} is the maximum available capacity of resource k .

Time constraints ensure that the project can be completed on schedule, as shown in formula (22).

$$f_1 \leq T_{deadline} \quad (22)$$

Where $T_{deadline}$ is the project deadline.

Capability constraints ensure that each task is undertaken by an enterprise with the corresponding capabilities, as shown in formulas (23) and (24).

$$\sum_{j \in E_i} x_{ij} = 1, \quad \forall i \in T \quad (23)$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in T, j \in E_i \quad (24)$$

Where E_i is the set of enterprises capable of undertaking task i , and x_{ij} is a binary decision variable indicating whether task i is assigned to enterprise j .

4. Meta-task Reconfiguration Optimization Solving Based on Improved Genetic Algorithm

4.1 Basic Principles of Genetic Algorithm

Genetic algorithm is a classic global optimization method that simulates the natural evolution process, searching the solution space through genetic mechanisms such as selection, crossover, and mutation (Chaudhary et al., 2026). Genetic algorithm can encode problem solutions as chromosomes and gradually approach the optimal solution through population iterative evolution, and has thus been widely applied in discrete combinatorial optimization problems, especially in task scheduling and resource allocation (Teng, 2025; Liu, Zha, Yan et al., 2024).

The core idea of genetic algorithm is to represent problem solutions as chromosomes and, by simulating natural selection and genetic variation, evolve the population toward the optimal solution. The algorithm starts from a set of randomly generated initial solutions (population) and evaluates the fitness of each solution using a fitness function. Individuals with higher fitness have a greater probability of being selected to enter the next generation. Crossover operation enables information exchange between different individuals, while mutation operation maintains population diversity. After multiple generations of evolution, the best individual in the population represents the approximate optimal solution to the problem.

Although the basic genetic algorithm has advantages such as strong global search capability and good robustness, it has some shortcomings when solving the meta-task reconfiguration optimization problem. First, the encoding method of the basic genetic algorithm mostly uses binary encoding. For discrete optimization problems such as task assignment, binary encoding leads to excessively long code strings, affecting algorithm efficiency. Second, the crossover and mutation operations of the basic genetic algorithm are prone to generating infeasible solutions, especially in the presence of complex constraints, requiring additional repair mechanisms to ensure solution feasibility (Liang, Ban, Yu et al., 2023). Finally, the basic genetic algorithm is prone to premature convergence in the later stages of evolution, becoming trapped in local optima (Nekoueian, Servranckx, & Vanhoucke, 2025).

To address these problems, this paper improves the basic genetic algorithm in three aspects. First, integer encoding is adopted instead of binary encoding, directly using enterprise numbers to represent task assignment schemes, simplifying the encoding structure and improving algorithm efficiency. Second, genetic operators based on problem characteristics are designed, including an order-preserving crossover operator and a heuristic mutation operator, which improve search efficiency while ensuring solution feasibility. Third, an elite retention strategy and an adaptive parameter adjustment mechanism are introduced, both maintaining population diversity and accelerating convergence speed.

4.2 Encoding and Decoding Design

The choice of encoding directly affects the efficiency and performance of the genetic algorithm. For the meta-task reconfiguration optimization problem, it is necessary to determine which enterprise undertakes each meta-task, which is essentially a task-to-enterprise mapping problem. This paper adopts an integer-based direct encoding method, using an n -dimensional integer vector to represent the reconfiguration scheme, where n is the total number of meta-tasks. Let the chromosome be $X = [x_1, x_2, \dots, x_n]$, $x_i \in E_i$, where x_i denotes the enterprise number undertaking task i . For example, for a problem with six meta-tasks and five candidate enterprises, a possible chromosome is $X=[2,4,5,1,5,3]$, meaning task 1 is undertaken by enterprise 2, task 2 by enterprise 4, and so on. To ensure solution feasibility, each gene x_i in the chromosome must be selected from the set of enterprises E_i capable of undertaking task i . During the initialization phase, for each task i , an enterprise with the corresponding capability is randomly selected.

The decoding process of a chromosome converts the integer encoding into a concrete task assignment scheme. Given a chromosome X , the task assignment matrix A can be directly obtained, where a_{ij} indicates whether task i is assigned to enterprise j , as shown in formula (25).

$$a_{ij} = \begin{cases} 1 & x_i = j \\ 0 & x_i \neq j \end{cases} \quad (25)$$

Based on the task assignment matrix, the execution sequence of tasks can be further calculated. Considering the technical dependencies among tasks, this paper adopts topological sorting to determine a feasible execution sequence. For task pairs (ij) with precedence constraints, task i must be completed before task j . Through topological sorting, an execution sequence that satisfies all technical constraints can be obtained, and this sequence is used to calculate the objective function values.

4.3 Fitness Function Design

The fitness function is used to evaluate the quality of chromosomes and serves as the guiding basis for the evolutionary direction of genetic algorithms. The fitness function in this paper is based on the multi-objective optimization model established in Section 3.3. By normalizing and weighted summation, the multi-objective problem is transformed into a single-objective fitness evaluation. For a given chromosome X , its fitness value is the reciprocal of the objective function value, as shown in Formula (26). The range of the fitness value is $(0, 1]$, and the smaller the value, the higher the fitness of the individual.

$$Fitness = \frac{1}{1 + F} \quad (26)$$

4.4 Genetic Operator Design

Genetic operators are the core mechanisms driving the evolution of the genetic algorithm population, mainly including selection, crossover, and mutation operators. Given the characteristics of the

meta-task reconfiguration optimization problem, this paper designs specialized genetic operators to ensure solution feasibility and improve algorithm search efficiency.

The selection operator is used to determine which high-quality individuals can enter the next generation for reproduction. This paper adopts the tournament selection method, which is simple to implement and maintains appropriate selection pressure. The specific operation is as follows: randomly select k individuals from the current population (typically $k=3$ or $k=5$), compare their fitness values, and select the individual with the highest fitness into the mating pool. This process is repeated until a sufficient number of individuals accumulate in the mating pool. The advantage of tournament selection is that it does not require global sorting of the population, offering high computational efficiency, while the selection pressure intensity can be flexibly controlled by adjusting the tournament size k .

This paper designs a task-based partially mapped crossover operator. Given two parent chromosomes $P_1=[p_{11},p_{12},\dots,p_{1n}]$ and $P_2=[p_{21},p_{22},\dots,p_{2n}]$, two crossover points pos_1 and pos_2 ($pos_1 < pos_2$) are randomly selected, dividing the chromosome into three segments. The offspring chromosome C_1 inherits the middle segment $[pos_1, pos_2]$ from P_1 , and the genes at the two ends are selected from the corresponding positions of P_2 , but must satisfy the capability constraint shown in formula (27). Similarly, offspring C_2 is generated. The adaptive crossover probability is given by formula (28).

$$c_{1i} = \begin{cases} P_{2i} & p_{2i} \in E_i \\ RandomSelect(E_i) & otherwise \end{cases} \quad (27)$$

$$P_c(t) = P_{c_{max}} - (P_{c_{max}} - P_{c_{min}}) \cdot (t / MaxGen) \quad (28)$$

This paper adopts a heuristic mutation strategy, using task correlation information to guide mutation operations. For each gene in the chromosome, whether to perform mutation is determined by the mutation probability p_m . Specifically, for the enterprise x_i currently assigned to task i , the similarity between this enterprise and other alternative enterprises is calculated. The enterprise similarity is measured based on the overlap of the task sets undertaken by them, as shown in formula (29). The adaptive mutation probability is given by formula (30).

$$Sim(E_j, E_k) = \frac{|T_j \cap T_k|}{|T_j \cup T_k|} \quad (29)$$

$$P_m(t) = P_{m_{max}} - (P_{m_{max}} - P_{m_{min}}) \cdot (t / MaxGen)^2 \quad (30)$$

Where T_j and T_k are the sets of tasks that enterprises E_j and E_k are capable of undertaking, respectively. During mutation, roulette wheel selection is performed with similarity as the weight, meaning that enterprises with higher similarity are more likely to be selected. This heuristic mutation strategy helps maintain population diversity while guiding the search toward more promising regions.

To prevent the loss of excellent individuals during the evolutionary process, this paper introduces an elite retention strategy. The best few individuals with the highest fitness in each generation are directly copied to the next generation without participating in crossover and mutation operations.

4.5 Algorithm Flow

The complete execution flow of the improved genetic algorithm includes initialization, fitness evaluation, selection, crossover, mutation, and elite retention. The algorithm starts from a randomly generated initial population and gradually approaches the optimal solution through iterative evolution. The algorithm framework is shown in Figure 1.

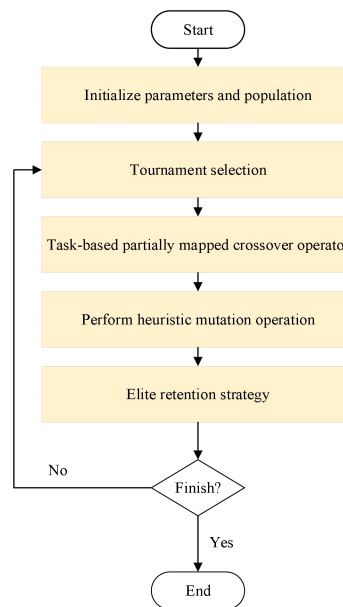


Figure 1. Improved Genetic Algorithm Framework

The pseudocode of the improved genetic algorithm is shown in Algorithm 1.

Algorithm 1: Improved Genetic Algorithm for Meta-task Reconfiguration Optimization Problem

Input: Meta-task set T , Enterprise set E , Correlation matrix R , Constraints

Output: Optimal reconfiguration scheme X^*

1: Initialize parameters: PopSize, MaxGen, Pc, Pm

2: Randomly generate the initial population $P(0) = \{X_1, X_2, \dots, X_{\text{PopSize}}\}$

3: Calculate the fitness value $\text{Fitness}(X_i)$ for each individual in the population

4: Find the current best individual $X^* = \text{argmax}\{\text{Fitness}(X_i)\}$

5: Set generation counter $t = 1$

6: **while** $t \leq \text{MaxGen}$ **do**

7: Generate the mating pool MatingPool using tournament selection

8: Offspring = \emptyset

```
9:  for i = 1 to PopSize/2 do
10:   Randomly select two parents P1, P2 from MatingPool
11:   if random() < Pc(t) then
12:     (C1, C2) = Partial Mapping Crossover (PMX)(P1, P2)
13:   else
14:     (C1, C2) = (P1, P2)
15:   end if
16:   Offspring = Offspring ∪ {C1, C2}
17: end for
18: for each individual X in Offspring do
19:   for each gene xi in X do
20:     if random() < Pm(t) then
21:       xi = Heuristic Mutation(xi, Ei)
22:     end if
23:   end for
24: end for
25: for each individual X in Offspring do
26:   Calculate Fitness(X)
27: end for
28: Sort Offspring according to fitness and retain the top EliteSize elite
    individuals
29: P(t) = Offspring
30: Xbest = argmax {Fitness(X) | X ∈ P(t)}
31: if Fitness(Xbest) > Fitness(X*) then
32:   X* = Xbest
33: end if
34: t = t + 1
35: end while
36: return the optimal reconfiguration scheme X* and its corresponding
    objective function value
```

5. Case Analysis and Verification

5.1 Case Background

To verify the effectiveness of the proposed meta-task decomposition and reconfiguration optimization method, an aero-engine impeller manufacturing task is selected as an example for analysis. The aero-engine impeller is one of the core components of an aircraft engine, and its manufacturing process involves multiple complex procedures, with extremely high requirements for machining accuracy,

material properties, and quality control. In the industrial internet environment, impeller manufacturing typically requires the collaboration of multiple enterprises with different professional capabilities. How to reasonably decompose the tasks and optimize the enterprise assignment scheme is of great significance for ensuring product quality, controlling production costs, and shortening the manufacturing cycle.

According to the process flow and technical requirements of impeller manufacturing, the entire manufacturing task is decomposed into six meta-tasks, namely: impeller blank forging, impeller precision milling, blade surface polishing, heat treatment strengthening, dynamic balance testing, and coating and encapsulation. There are clear technical dependencies among these meta-tasks, and they must be executed in a specific process sequence. Meanwhile, there are five enterprises in the industrial internet network that can undertake these tasks. Different enterprises possess different technical capabilities and resource conditions, and an optimization algorithm is needed to find the optimal task assignment scheme.

5.2 Data Preparation

1. Basic Attributes of Meta-tasks

According to the actual process requirements of impeller manufacturing, the basic attribute data of each meta-task are shown in Table 1.

Table 1. Basic Attributes of Meta-tasks

Task ID	Task name	Execution time (h)	Cost (ten thousand RMB)	Quality weight
T1	Impeller blank forging	24	5.0	0.15
T2	Precision impeller milling	36	8.0	0.25
T3	Blade surface polishing	18	4.0	0.20
T4	Heat treatment strengthening	48	6.0	0.15
T5	Dynamic balancing inspection	12	3.0	0.15
T6	Coating and encapsulation	24	5.5	0.10

The execution time reflects the duration requirement of each task. The cost data are based on market research and enterprise quotations. The quality weights are determined according to the impact of each process on the final product quality, and the sum of the weights is 1.0.

2. Technological Correlation Matrix

The technological correlation among meta-tasks is mainly reflected in the process sequence constraints. According to the process flow of impeller manufacturing, the technological correlation matrix is shown

in Table 2.

Table 2. Technological Correlation Matrix

	T1	T2	T3	T4	T5	T6
T1	0	1	0	0	0	0
T2	0	0	1	1	0	0
T3	0	0	0	0	1	0
T4	0	0	0	0	1	0
T5	0	0	0	0	0	1
T6	0	0	0	0	0	0

In the matrix, an element value of 1 indicates a technological dependency. For example, $r_{12}^{\text{tech}}=1$ means that task T2 cannot start until task T1 is completed. As can be seen from the matrix, the main process route is $T1 \rightarrow T2 \rightarrow \{T3, T4\} \rightarrow T5 \rightarrow T6$.

3. Candidate Enterprise Capabilities

There are five enterprises in the industrial network that can undertake the impeller manufacturing tasks. The capability distribution and geographical locations of each enterprise are shown in Table 3.

Table 3. Candidate Enterprise Capability distribution

Enterprise ID	Capable tasks	Geographical location	Enterprise expertise
E1	T1, T4	City A	Hot working
E2	T2, T3	City B	Precision machining
E3	T5, T6	City C	Inspection and surface treatment
E4	T1, T2, T4	City A	Comprehensive manufacturing
E5	T3, T5, T6	City B	Quality control

Different enterprises have different technical expertise. For example, E1 and E4 specialize in hot working, E2 focuses on precision machining, and E3 and E5 have advantages in inspection and surface treatment.

4. Geographical Distances Between Enterprises

Considering logistics costs and coordination convenience, the geographical distance matrix among enterprises is shown in Table 4.

Table 4. Geographical Distances Between Enterprises (kilometers)

	E1	E2	E3	E4	E5
--	----	----	----	----	----

E1	0	200	500	50	250
E2	200	0	400	180	80
E3	500	400	0	480	420
E4	50	180	480	0	200
E5	250	80	420	200	0

E1 and E4 are located in the same city A, relatively close; E2 and E5 are located in city B, relatively close; E3 is located in city C, relatively far away.

5. Coordination Cost Coefficient

When two tasks with strong correlation are undertaken by different enterprises, additional coordination costs are incurred. The coordination cost is related to the task correlation strength and the distance between enterprises. The basic coordination cost coefficient is set to 0.5 (ten thousand RMB per 100 kilometers).

6. Quality Achievement Degree

The quality achievement degrees of each enterprise for different tasks are shown in Table 5. The value range is [0, 1], with larger values indicating stronger quality assurance capability, where "-" indicates that the enterprise does not have the capability to undertake the corresponding task.

Table 5. Enterprise Quality Achievement Degree

	T1	T2	T3	T4	T5	T6
E1	0.90	-	-	0.85	-	-
E2	-	0.92	0.88	-	-	-
E3	-	-	-	-	0.90	0.87
E4	0.85	0.88	-	0.90	-	-
E5	-	-	0.90	-	0.93	0.90

7. Optimization Objective Weights

According to the preferences of enterprise decision-makers, the weights for the three optimization objectives are set as: efficiency objective $\alpha_1=0.4$, cost objective $\alpha_2=0.4$, quality objective $\alpha_3=0.2$. This weight setting reflects equal emphasis on efficiency and cost while also taking quality requirements into account.

5.3 Experimental Setup

To verify the performance of the improved genetic algorithm, the following experimental scheme was designed. The algorithm parameters were set as follows: population size PopSize=50, maximum

number of generations MaxGen=200, initial crossover probability $P_c^{\max}=0.9$, minimum crossover probability $P_c^{\min}=0.7$, initial mutation probability $P_m^{\max}=0.1$, minimum mutation probability $P_m^{\min}=0.01$, and elite retention ratio of 10%. To verify the stability of the algorithm, each set of experiments was independently run 30 times, and metrics such as the optimal solution, average solution, and convergence generation were recorded. All experiments were implemented in Python 3.8, and the experimental environment was a 64-bit Windows 11 operating system with an Intel(R) Core(TM) Ultra 7 155H 3.80 GHz processor and 16 GB of memory. The random assignment scheme and the improved genetic algorithm scheme were compared. The superiority of the proposed method was verified through comparative analysis.

5.4 Reconfiguration Optimization Results

After running the improved genetic algorithm, the optimal reconfiguration scheme for the impeller manufacturing task was obtained. The optimal task assignment scheme is shown in Table 6.

Table 6. Optimal Task Assignment Scheme

Task ID	Task name	Assigned enterprise	Enterprise expertise	Start time (h)	Completion time (h)
T1	Impeller blank forging	E1	Hot working	0	24
T2	Precision impeller milling	E2	Precision machining	24	60
T3	Blade surface polishing	E5	Quality control	60	78
T4	Heat treatment strengthening	E4	Comprehensive manufacturing	60	108
T5	Dynamic balancing inspection	E5	Quality control	108	120
T6	Coating and encapsulation	E5	Quality control	120	144

As can be seen from Table 6, the optimal scheme achieves a reasonable division of labor based on the professional capabilities of the enterprises. T1 is assigned to E1, fully utilizing its hot working expertise; T2 is assigned to E2, leveraging its precision machining advantages; three tasks, T3, T5, and T6, are centrally assigned to E5, reflecting E5's core role in quality control and surface treatment; T4 is assigned to E4, utilizing its comprehensive manufacturing capability. This assignment scheme not only follows the principle of specialized division of labor but also effectively reduces coordination costs by centrally assigning highly correlated subsequent tasks (T3, T5, T6) to the same enterprise. The total

completion time of this scheme is 144 hours, the total cost is 348,000 RMB, and the comprehensive quality achievement degree is 0.9095. Comparing the performance metrics of the random assignment scheme and the improved genetic algorithm scheme, the results are shown in Table 7.

Table 7. Performance Comparison of Different Schemes

Scheme	Total time (h)	Total cost (ten thousand RMB)	Quality achievement degree	Comprehensive fitness
random assignment	144	36.90	0.9065	0.9422
Improved Genetic Algorithm	144	34.80	0.9095	0.9422
Degree of improvement	0%	5.69%	0.33%	6.14%

As can be seen from Table 7, the improved genetic algorithm scheme outperforms the random assignment scheme in multiple dimensions. It is worth noting that the total completion time of both schemes is 144 hours, which is because the technical dependencies among tasks and the execution durations determine the lower bound of the critical path length. Given that the total time has already reached the theoretical optimum, the algorithm mainly improves the comprehensive performance by optimizing the cost and quality objectives.

In terms of total cost, the optimized scheme reduces the cost by 21,000 RMB, a decrease of 5.69%. By centrally assigning the three tasks T3, T5, and T6 to enterprise E5, the number of cross-enterprise coordinations in subsequent processes is reduced. In particular, the close linkage between T5 and T6, when undertaken by the same enterprise, avoids additional logistics and communication costs. Meanwhile, the assignment of T1 to E1 and T4 to E4 takes advantage of the geographical proximity of these two enterprises in city A (only 50 km apart), further reducing logistics costs.

In terms of quality achievement degree, the optimized scheme improves by 0.003, an increase of 0.33%. Although the improvement is relatively small, it should be noted that the quality capabilities of all enterprises are already at a high level (0.85–0.93). The quality improvement is mainly reflected in the optimal allocation of key processes: for example, T2 is assigned to E2, which has the highest quality achievement degree (0.92), and T5 is assigned to E5 (0.93). This ensures that the two processes that significantly affect final product quality—precision machining and dynamic balance testing—receive optimal guarantees.

Figure 1 shows the convergence process of the improved genetic algorithm. As can be seen from the figure, the algorithm exhibits a typical three-stage convergence characteristic. In the first 60 generations, a rapid ascent stage, the optimal fitness increases rapidly from the initial 0.9422 to 0.9933. In this stage, the algorithm performs extensive global search with relatively high crossover and

mutation probabilities. Generations 60–178 constitute a stable plateau period, where the optimal fitness remains at 0.9933, and the population performs local search and fine adjustment near the current optimal solution. After generation 178, a breakthrough from the plateau is achieved, reaching the ideal value of 1.0000, thanks to the random perturbation of the mutation operator and the maintenance of population diversity. The average fitness curve shows a continuous upward trend, gradually increasing from the initial 0.9422 to the final 1.0000, indicating that the overall population quality is constantly improving and the algorithm exhibits good global convergence.

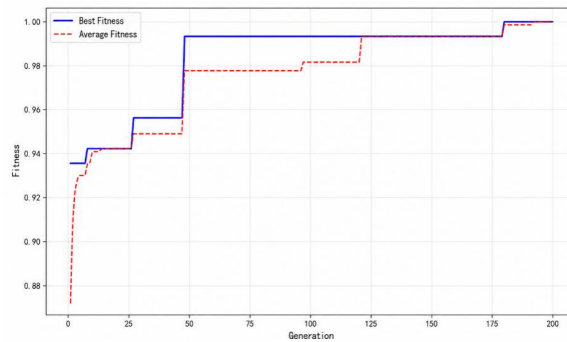


Figure 1. Genetic Algorithm Convergence Curve

Figure 2 shows the Gantt chart of the optimal task execution scheme. The timing arrangement of task execution and the enterprise allocation can be clearly seen from the figure. The critical path is T1 → T2 → T4 → T5 → T6, which determines the total project duration of 144 hours. As a subsequent parallel task of T2, T3 starts at the 60th hour, is executed simultaneously with T4, but finishes before T4. From the perspective of enterprise allocation, the scheme exhibits the characteristics of "specialized division of labor plus centralized collaboration": in the early stage, E1 and E2 undertake the forging and milling tasks, respectively, in which they have expertise; in the middle stage, T3 and T4 are executed in parallel; in the later stage, three tasks are continuously undertaken by E5, forming a smooth process.

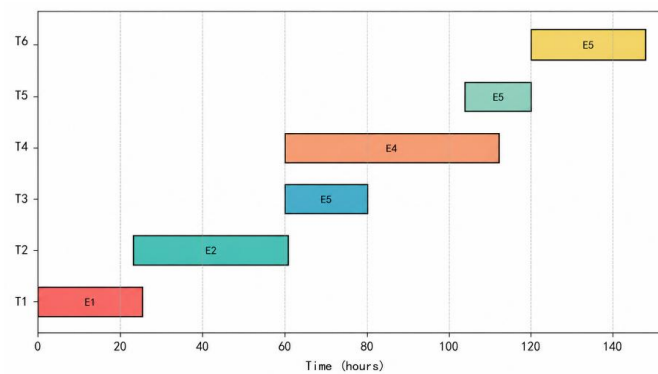


Figure 2. Task Execution Gantt Chart

From the Gantt chart, several important characteristics of task execution can also be observed. First,

there is no resource waiting time on the critical path, and tasks are tightly connected, indicating that resource allocation is reasonable. Second, the parallel execution of T3 and T4 reflects process flexibility: although both depend on T2, they have no dependency on each other and can be carried out simultaneously. Third, E5 continuously undertakes three tasks—T3, T5, and T6. Although there is a 30-hour gap between T3 and T5 (due to the need to wait for T4 to complete), this allocation avoids frequent cross-enterprise switching and reduces coordination complexity overall.

In summary, the experimental results verify the effectiveness of the improved genetic algorithm for the meta-task reconfiguration optimization problem. The algorithm can find high-quality solutions within a reasonable computation time, exhibits good convergence performance, and has practical application value. The optimal scheme will serve as the input for the next chapter's knowledge graph matching, to find specific sets of candidate resources for the six meta-tasks, which will then be used for subsequent service composition optimization, ultimately forming a complete cloud manufacturing resource allocation scheme.

6. Conclusions

This paper studies the problem of cloud manufacturing task decomposition and reorganization optimization. In response to the issues that traditional task decomposition methods mainly rely on process logic and are difficult to fully reflect the differences in enterprise capabilities and task acceptance boundaries, a multi-dimensional task decomposition method based on capability matching and a multi-objective reorganization optimization model are proposed. To describe the correlation characteristics among sub-tasks, a multi-dimensional correlation description model covering technical relevance, capability relevance, geographical relevance and cost relevance is constructed, and based on this, a multi-objective reorganization optimization model with efficiency, cost and quality as the objectives is established. Regarding the problems existing in the model-solving process, such as redundant encoding, difficulty in maintaining feasible solutions and easy falling into local optima, an improved genetic algorithm is designed. Through mechanisms such as integer encoding, order-preserving crossover, heuristic mutation, elite retention and adaptive parameter adjustment, the efficiency and solution quality of the solution are improved. Through an example of the manufacturing task of the fan blades for an aircraft engine, the results show that the optimized solution maintains the optimal construction period while reducing the cost by 5.69% and improving the quality by 0.33%, verifying the effectiveness of the method.

References

- Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 48(3), 292-306.
- Chaudhary, R., Verma, P., Salgotra, R. et al. (2026). Five decades of genetic algorithms: A systematic

- and bibliometric review (1975-2025). *Archives of Computational Methods in Engineering*.
- Globerson S. Impact of various work-breakdown structures on project conceptualization. *International Journal of Project Management*, 12(3), 165-171.
- Liang, J., Ban, X., Yu, K. et al. (2023). A survey on evolutionary constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 27(2), 201-221.
- Liu, M., Wang, Q., & Ling, L. (2017). Cloud manufacturing task decomposition method based on hierarchical task network. *Chinese Mechanical Engineering*, 28(08), 924-930.
- Liu, Z., Zha, J., Yan, J. et al. (2024). An improved genetic algorithm with an overlapping strategy for solving a combination of order batching and flexible job shop scheduling problem. *Engineering Applications of Artificial Intelligence*, 127, 107321.
- Nekoueiian, R., Servranckx, T., & Vanhoucke, M. (2025). A dynamic learning-based genetic algorithm for scheduling resource-constrained projects with alternative subgraphs. *Applied Soft Computing*, 180, 113316.
- Steward, D. V. (1981). The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, EM-28(3), 71-74.
- Teng, G. (2025). An improved genetic algorithm for dual-resource constrained flexible job shop scheduling problem with tool-switching dependent setup time. *Expert Systems with Applications*, 281, 127496.
- Wang, T., Guo, S., & Lee, C. G. (2014). Manufacturing task semantic modeling and description in cloud manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 71(9-12), 2017-2031.
- Yi, S., Tan, M., Guo, Z. et al. (2015). Manufacturing task decomposition optimization in cloud manufacturing service platform. *Computer Integrated Manufacturing System*, 21(08), 2201-2212.
- Zhang, C., He, K., Liu, Q. et al. (2025). Cloud manufacturing task decomposition method considering resource compatibility and competitiveness. *International Journal of Computer Integrated Manufacturing*.
- Zhang, C., He, K., Liu, Q. et al. (2026). Cloud manufacturing task decomposition method considering resource compatibility and competitiveness. *International Journal of Computer Integrated Manufacturing*, 39(2), 183-203.