Journal of Engineering Mechanics and Machinery (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/jemm.2025.100205
ISSN 2371-9133 Vol. 10 Num. 2

Research on Multi-Objective Optimisation-Based Task Package Division Methods for Shipbuilding

Lijun Liu[1,*], Fei Ren[1], Jiahao Liu[1], Huisong Meng[1], Zuhua Jiang[2]

[1]*College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an, Shaanxi, China*

[2]*School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China*

**Corresponding author*

Keywords: Shipbuilding; Multi-Objective Optimisation; Engineering Decomposition; Task Package; Genetic Algorithm

Abstract: Shipbuilding requires converting multi-disciplinary product units—such as hull structures, outfitting equipment, and coating processes—into executable task packages to support sectional construction and final assembly on the slipway. Three critical issues arise during practical division: firstly, weak inter-process connectivity within task packages, where splitting welding and assembly tasks within the same compartment disrupts workflow continuity and increases redundant handling costs; Secondly, high coupling between task packages arises when construction packages are empirically divided, placing sequentially dependent tasks under different crews and triggering frequent cross-departmental coordination. Thirdly, resource allocation imbalances occur when critical resources like machining equipment are concentrated or scarce, constraining the construction cycle. Therefore, this paper constructs a multi-objective optimisation mixed-integer programming model. First, leveraging information entropy theory, we quantify the homogeneity of shipbuilding processes within task packages—such as hull welding and piping pre-installation—to ensure process continuity and specialised concentration. Second, a task dependency matrix quantitatively assesses and reduces inter-package coupling, minimising cross-package coordination costs. Finally, an equitable resource allocation metric is introduced, using variance in critical resource utilisation to achieve balanced distribution. Simultaneously, linearisation techniques address non-linear constraints, while a two-stage solution strategy combining branch-and-bound with genetic algorithms balances accuracy and efficiency. Ultimately, through case validation at a major Chinese shipbuilding enterprise, the proposed task package segmentation methodology demonstrated both its efficacy and practical feasibility. Empirical results demonstrate that this approach offers significant advantages in practical application. It enhances the cohesion of similar processes within task packages, effectively reduces the frequency of cross-package task coordination, and reasonably controls fluctuations in critical resource utilisation. Furthermore, it adapts to the complex scenarios of multi-disciplinary and multi-resource shipbuilding, providing a scientific decision-making tool for project task organisation and resource allocation.

1. Introduction

Shipbuilding constitutes a quintessential large-scale, complex manufacturing process, encompassing multidisciplinary collaborative operations such as hull construction, machinery installation, and electrical commissioning[1]. It necessitates the integrated assembly of tens of thousands of components, where the scientific organisation of production directly determines project efficiency and cost. As the core foundational tool for shipbuilding project management, the Work Breakdown Structure (WBS) transforms complex projects into manageable product units through hierarchical decomposition, providing a framework for subsequent task execution and resource allocation. The conversion of product units derived from WBS decomposition into directly executable task packages represents the pivotal link between project breakdown and production implementation. This division not only impacts the fluidity of construction processes and resource utilisation but also plays a decisive role in project cost control and schedule assurance.

The challenge in task package division lies in rationally combining product units with differing technical attributes and process requirements while satisfying three primary objectives: first, achieving high internal homogeneity within each task package to reduce task-switching costs and enhance production efficiency; second, minimising dependencies between task packages to reduce cross-task coordination workload; and third, balancing workloads across all task packages to prevent resource idleness or overload, thereby achieving overall resource optimisation[2].

Task package division in shipbuilding fundamentally constitutes a multi-constraint combinatorial optimisation problem. Early research predominantly employed heuristic division methods based on engineering experience. While operationally straightforward, these lacked systematic theoretical underpinnings and quantitative analysis, catering only to single-scenario production needs and proving ill-suited to the complex characteristics of shipbuilding—marked by multidisciplinary integration and multi-objective coupling. With the deepening application of optimisation theory in manufacturing, scholars have progressively explored this field from multiple dimensions: at the theoretical methodology level, information theory provides tools for objectively measuring grouping quality (Cover & Thomas, 2006), enabling the quantification of unit homogeneity through information entropy; The Design Structure Matrix (DSM) method offers an effective means for analysing dependencies within systems (Eppinger & Browning, 2012), aiding in the characterisation of technical constraints between tasks; modular design theory, meanwhile, establishes the core grouping principle of 'low coupling, high cohesion' (Baldwin & Clark, 2000), providing theoretical justification for logical partitioning. In application research within shipbuilding, existing achievements encompass integrated planning for section assembly (Choe et al., 2008), multi-objective optimisation of construction scheduling (Liu et al., 2016), and task allocation based on spatial constraints (Park et al., 2017)[3]. However, existing research exhibits notable limitations: most studies focus on single optimisation objectives or localised process optimisation, failing to develop a systematic partitioning methodology that comprehensively considers product unit technical cohesion, task interdependencies, and cross-disciplinary resource load balancing. This approach struggles to meet the multi-objective optimisation demands of the entire shipbuilding process[4].

Addressing these research gaps and practical engineering needs, this paper constructs a multi-objective optimisation model that holistically considers task package cohesion, inter-package coupling, and cross-disciplinary resource load balancing: information entropy theory is introduced to quantify the technical homogeneity of product units within task packages, thereby representing cohesion; a task dependency matrix is designed to assess inter-package coupling through correlation strength coefficients; A resource load fairness metric is established to quantify the equilibrium level of multi-disciplinary resource allocation. To overcome model solution bottlenecks, linearisation techniques are applied to address non-linear constraints, complemented by a two-stage 'branch-and-bound – genetic algorithm' solution strategy[5]. Branch-and-bound enhances initial solution accuracy, while genetic algorithms augment global optimisation capabilities, enabling efficient and precise model solutions. This research provides theoretical foundations and practical methodologies for optimising shipbuilding task organisation, thereby advancing the sophistication of project management practices[6].

The rest of this paper proceeds as follows. Section 2 elaborates on the problem description and mathematical modeling of shipbuilding task package partitioning, including the definition of WBS-decomposed product attributes, model assumptions, decision variables, the construction of three core optimization objective functions[7]. Section 3 details the methodology for solving the multi-objective optimization model, covering the linearization processing of nonlinear terms in the model, the global search mechanism of genetic algorithm, the local refinement logic of branch-and-bound algorithm, and the design of the two-stage hybrid solution strategy integrating the two algorithms. In Section 4, the proposed task package partitioning method is verified through a case study targeting the cargo hold section of a ship, with in-depth analysis of the algorithm's convergence process, optimization results of core indicators, and practical application effects. Conclusions and prospects for future studies are outlined in the final section[8].

2. Task Package Partitioning Problem Description and Mathematical Modeling

2.1 Problem Description

During the construction of large, complex products such as ships, the product set derived from the work breakdown structure must be appropriately divided into several task packages to facilitate parallel production and resource scheduling[9].

Let the set of products obtained through WBS decomposition be $P = \{p_1, p_2,..., p_n\}$， Each product $p_i \in P$ Possessing the following attributes:

Stage $s_i \in S$ = {Design, prefabrication, assembly, outfitting}

Type $t_i \in T$ = {Hull, Piping, Electrical, Engine Room}

Region$z_i \in Z$ = {Bow and stern, cargo hold, engine room, superstructure}

System$f_i \in F$ = {Power systems, electrical systems, communication systems,...}

Workload weighting $w_i \in \mathbb{R}^+$

The objective of the task allocation problem is to assign n products to k groups. Task package $T = \{T_1, T_2,..., T_k\}$， under the conditions of satisfying technical constraints and resource limitations, achieve maximum cohesion, minimum coupling, and load balancing[13].

2.2 Task Package Partitioning Assumptions

To establish the mathematical model, the following assumptions are made:

1) Task package quantity assumption: The number of task packages, k, is a predefined parameter satisfying $k_{min} \le k \le k_{max}$, where kmin corresponds to the minimum number of production teams required by the shipyard (e.g., at least one welding team and one outfitting team), and kmax corresponds to the number of berth workstations, preventing excessive task packages from causing idle workstations;

2) Product Uniqueness Assumption: Each product may be assigned to only one task package, with no product segmentation permitted. This is because shipbuilding products, such as complete engine room piping runs or entire rib frameworks, are predominantly indivisible manufacturing units. Segmentation would cause process interruptions, e.g., piping welding requires continuous operation[10].

3) Task package capacity assumption: Each task package has upper and lower limits on product quantity, where $L_{min} \leq T_j \leq L_{max}$. L_{max} represents the minimum daily throughput capacity of a crew, while L_{max} denotes the maximum daily load capacity;

4) Discrete attribute assumption: Product attributes across dimensions form discrete finite sets. For instance, construction phases are categorised solely into design, prefabrication, assembly, and outfitting. This alignment with shipyard's phased, standardised classification facilitates quantifying cohesion and coupling through discrete attributes; (Important, unclear, relates to subsequent text)

5) Dependency quantification assumption: Inter-product dependencies can be quantified as values within the [0,1] interval. In shipbuilding, dependencies such as preceding-subsequent and sibling nodes possess explicit process priorities. Quantification enables precise control over coordination costs between task packages[12] ;

6) Assumption of Workload Additivity: The total workload of a task package equals the sum of the workloads of its constituent products. When shipyards calculate task package loads, they typically aggregate based on the 'standard man-hours' per product;

7) Independent Execution Assumption: Each task package can be independently assigned to different production units for execution. For instance, hull teams handle hull structure task packages while electrical teams manage electrical equipment task packages. This aligns with the shipyard's specialised division of labour model, avoiding inefficiencies caused by cross-team mixed operations[2] ;

8) Deterministic Assumption: All parameters are assumed to be deterministically known. Based on historical shipyard production data, parameter values can be determined in advance, ensuring model inputs align with engineering reality[8].

2.3 Decision Variables and Parameter Definitions

Define the binary decision variable:

If the product is assigned to the task package, otherwise

$$\chi_{ij} = \begin{cases} l, & \textit{If} \text{ product } \mathrm{p}_i \text{ is assigned to task package } \mathrm{T}_j \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

$$i \in \{1, 2, \ldots, n\},\ j \in \{1, 2, \ldots, k\}$$

Introduce an auxiliary variable to represent the task package activation status:

If the task package is activated, otherwise

$$y_j = \begin{cases} 1, & \text{If task package } \mathrm{T}_j \text{ is enabled} \\ 0, & \textit{otherwise} \end{cases} \qquad j \in \{1, 2, ..., k\} \tag{2}$$

2.4 Optimisation Objective Function Construction

(1) Cohesion objective function

Assessing the degree of similarity among products within a task package involves evaluating its compactness and interrelatedness by considering the distribution of product attributes across four dimensions: phase, type, region, and system. Higher cohesion indicates a more concentrated distribution of products within the task package across these dimensions, signifying a greater degree of specialisation within the task package[15].

1) Calculation of stage-level cohesion:

For task package T_j, the entropy of attribute distribution within the stage dimension is:

$$H_s^{(j)}(X) = -\sum_{s \in S} p_s^{(j)}(X) \cdot \log_2 p_s^{(j)}(X) \tag{3}$$

Here, $p_s^{(j)}(X)$ denotes the proportion of products belonging to stage s within task package j:

2) Calculation of Cohesion within Type Dimension:

Considering the two-level structure of processing types, calculate weighted cohesion:

$$H_t^{(j)}(X) = -\sum_{t \in T} p_t^{(j)}(X) \cdot \log_2 p_t^{(j)}(X) \tag{4}$$

The weighting for considering sub-categories:

$$p_t^{(j)}(X) = \frac{\sum_{i=l}^{n} x_{ij} \cdot \mathrm{I}(t_i \in T_k)}{\sum_{i=l}^{n} x_{ij} + \epsilon} \tag{5}$$

Type-dimension cohesion:

$$Cohesion_t^{(j)}(X)=1-\frac{H_t^{(j)}(X)}{log_2|T|} \tag{6}$$

3) Calculation of regional dimension cohesion:

Considering spatial adjacency, distance weighting is introduced[16] :

$$H_z^{(j)}(X)=-\sum_{z \in Z} p_z^{(j)}(X) \cdot log_2 p_z^{(j)}(X) \tag{7}$$

$$p_z^{(j)}(X)=\frac{\sum_{i=1}^{n} x_{ij} \cdot I(z_i=z) \cdot d(z_i,z_{center})}{\sum_{i=1}^{n} x_{ij}+\epsilon} \tag{8}$$

Among these,， $d(z_i, z_{\text{center}})$ For regional distance weighting, adjacent regions carry a higher weight.

Regional Cohesion:

$$Cohesion_z^{(j)}(X)=1-\frac{H_z^{(j)}(X)}{log_2|Z|} \tag{9}$$

4) Calculation of System-Level Cohesion:

$$H_f^{(j)}(X)=-\sum_{f \in F} p_f^{(j)}(X) \cdot log_2 p_f^{(j)}(X) \tag{10}$$

$$p_f^{(j)}(X)=\frac{\sum_{i=1}^{n} x_{ij} \cdot I(f_i=f)}{\sum_{i=1}^{n} x_{ij}+\epsilon} \tag{11}$$

System-level dimensional cohesion:

$$Cohesion_f^{(j)}(X)=1-\frac{H_f^{(j)}(X)}{log_2|F|} \tag{12}$$

Comprehensive Cohesion Objective Function:

Weighted integration of cohesion across four dimensions[19] :

$$f_1(X)=\sum_{j=1}^{k} \frac{\sum_{i=1}^{n} x_{ij}}{n} \cdot \left[w_s \cdot Cohesion_s^{(j)}(X)+w_t \cdot Cohesion_t^{(j)}(X)+w_z \cdot Cohesion_z^{(j)}(X)+w_f \cdot Cohesion_f^{(j)}(X)\right] \tag{13}$$

Where the dimensional weights satisfy:

$$\mathrm{w_s} + \mathrm{w_t} + \mathrm{w_z} + \mathrm{w_f} = 1 \tag{14}$$

w_s: Phase dimension weight, reflecting the importance of process continuity;
w_t: Type dimension weight, reflecting the importance of task specialisation;
w_z: Area dimension weight, reflecting the importance of spatial concentration;
w_f: System dimension weight, reflecting the importance of functional integrity;

This objective function considers the cohesion contribution of each dimension through weighted averaging, with the task package size weight ensuring that the cohesion of larger task packages receives adequate emphasis[4].

(2) Coupling Degree Objective Function

Measures the dependency level between deliverables within task packages. Different dependency coefficients are assigned based on relationships such as predecessor dependencies, sibling relationships, and adjacent phases between deliverables. The coupling degree between task packages is calculated to reflect the closeness of their interconnections[7].

The task-package coupling degree is defined as:

$$Coupling_{jm}(X)=\frac{\sum_{i=1}^{n}\sum_{l=1}^{n}\delta_{il}\,x_{ij}\,x_{lm}}{(\sum_{i=1}^{n}x_{ij}+\epsilon)\cdot(\sum_{l=1}^{n}x_{lm}+\epsilon)} \tag{15}$$

Minimisation objective for overall coupling:

$$f_2(X)=\sum_{j=1}^{k-1}\sum_{m=j+1}^{k}Coupling_{jm}(X) \tag{16}$$

(3) Load Balancing Objective Function

Measures the degree of load equilibrium among task bundles by calculating the variance between task bundle loads and the ideal average load, thereby assessing load uniformity[20].

Load balancing metric based on the Jain fairness index:

$$JFI(X)=\frac{(\sum_{j=1}^{k}S_j)^2}{k\cdot\sum_{j=1}^{k}S_j^2} \tag{17}$$

The payload for task package j is:

$$S_j=\sum_{i=1}^{n}x_{ij}\cdot w_i \tag{18}$$

Product workload weighting is based on a multi-factor assessment:

$$w_i=\alpha\cdot{}_{\text{level}}(i)+\beta\cdot{}_{\text{complexity}}(t_i)+\gamma\cdot{}_{\text{scale}}(i) \tag{19}$$

To facilitate optimisation, the load variance minimisation form is employed:

$$f_3(X)=\sum_{j=1}^{k}\left(S_j-\overline{S}\right)^2 \tag{20}$$

Among which， $\overline{S}=\frac{1}{k}\sum_{i=1}^{n}w_i$ For ideal average load.

2.5 Integrated Optimisation Model

By integrating the three objective functions—cohesion, coupling, and load imbalance—and assigning distinct weighting coefficients λ_1, λ_2, and λ_3, we comprehensively evaluate the merits of task package allocation schemes to identify the optimal configuration[2].

Combining these three objectives, we construct a multi-objective optimisation model:

$$
\begin{aligned}
\max \quad & F(X)=\lambda_1 \cdot f_1(X)-\lambda_2 \cdot f_2(X)-\lambda_3 \cdot f_3(X) \\
s.t. \quad & \sum_{j=1}^{k} x_{ij} =1, \quad \forall i \in\{1,\ldots,n\} \\
& L_{min} \cdot y_j \le \sum_{i=1}^{n} x_{ij} \le L_{max} \cdot y_j, \quad \forall j \\
& x_{ij} \le y_j, \quad \forall i,j \\
& \sum_{j=1}^{k} y_j \le k \\
& \sum_{m=1}^{j} x_{im} \ge \sum_{m=1}^{j} x_{lm}, \quad \forall (i,l) \in Precedence \\
& x_{ij} \in\{0,1\}, \quad y_j \in\{0,1\}
\end{aligned} \tag{21}
$$

Parameter Explanation:

F(X) represents the composite objective function, converting multi-objective criteria into a single objective through linear weighting[11].

Λ_1 governs the importance of cohesion; a higher value prioritises process continuity.

Λ_2 controls the weighting of coupling; a higher value emphasises reducing cross-package coordination[16].

Λ_3 regulates the weighting of load balancing; a higher value prioritises equitable resource allocation.

The weights satisfy $\lambda_1+\lambda_2+\lambda_3=1$ and are set by decision-makers according to project characteristics. For this study, they are set as $\lambda_1 = 0.50$, $\lambda_2 = 0.30$, $\lambda_3 = 0.20$.

2.6 Constraints

The constraints for the integrated optimisation model are as follows:

(1) Basic constraints

1) Product uniqueness allocation constraint[22] :

$$\sum_{j=1}^{k} x_{ij} =1, \quad \forall i \in\{1,\ldots,n\} \tag{22}$$

This constraint ensures that all products are allocated and not repeatedly allocated.

2) Task package size constraint:

$$L_{min} \cdot y_j \le \sum_{i=1}^{n} x_{ij} \le L_{max} \cdot y_j, \quad \forall j \tag{23}$$

Avoid task packages being too small leading to low resource utilization, or too large leading to management difficulties.

Logical consistency constraint:

$$x_{ij} \le y_j, \quad \forall i,j \tag{24}$$

3) Task package quantity constraint:

$$\sum_{j=1}^{k} y_j \le k \tag{25}$$

(2) Technical Process Constraints

Ensure that products with "predecessor-successor dependency relationships" follow the process sequence in task package allocation, i.e., the production timing of the task package containing predecessor products is not later than that of the task package containing successor products, avoiding the process reversal problem of "successor products produced first, predecessor products produced later"[3].

1) Predecessor dependency constraint:

$$\sum_{m=1}^{j} x_{im} \geq \sum_{m=1}^{j} x_{lm}, \quad \forall (i,l) \in Precedence \tag{26}$$

If product i is a predecessor of product l, then i must be assigned to a task package with a number no greater than l, ensuring construction order[22].

2) Construction stage continuity constraint:

Limit the span of production stages of products within a single task package to ensure that the production stages of products within the task package are relatively concentrated, avoiding products within the task package spanning four discontinuous stages, and reducing equipment and personnel switching costs caused by large stage spans[6].

$$\max_{p_i, p_l \in T_j} |s_i - s_l| \leq \Delta_s^{max}, \quad \forall j \tag{27}$$

Avoid mixing tasks with excessive spans such as prefabrication, assembly, and outfitting. Typically Δsmax = 2.

Spatial adjacency constraint:

$$x_{ij} + x_{lj} \leq 1 + I(adjacent(z_i, z_l)), \quad \forall i,l,j \tag{28}$$

Reduce logistics costs and safety risks caused by cross-regional operations.

3. Methodology

The solution algorithm is divided into two main stages: the genetic algorithm global search stage and the branch-and-bound local refinement stage. The flow chart of the two-stage hybrid solution process is shown in Figure 1.

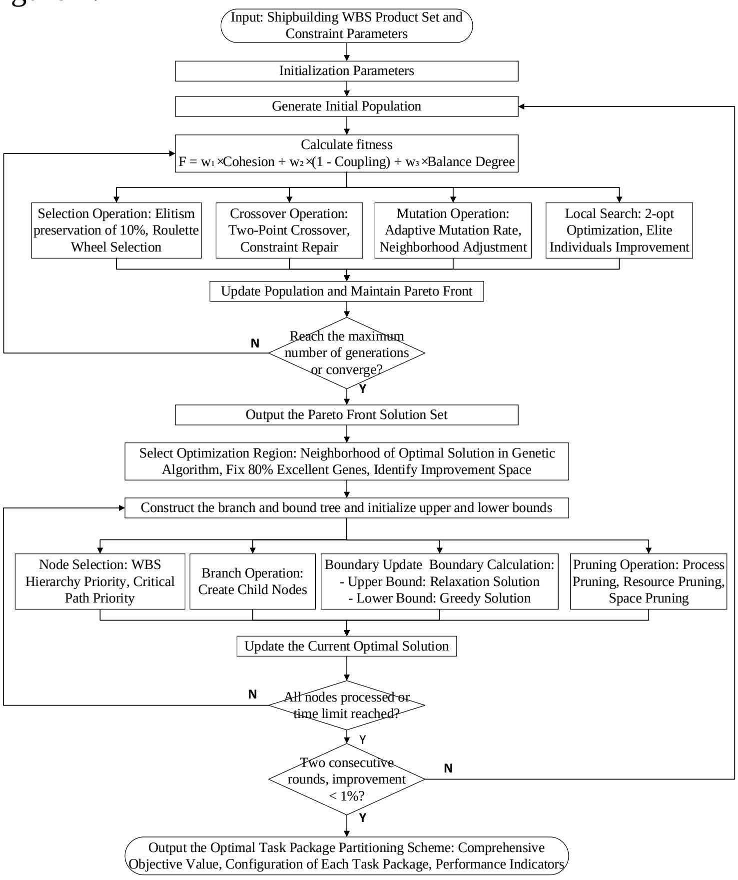

Figure 1 Two-Stage Hybrid Solution Process Based on Genetic Algorithm and Branch-and-Bound

The first stage is the genetic algorithm global search stage: The algorithm starts with the input shipbuilding WBS product set and constraint parameters, generates an initial population, and enters an iterative optimization loop[3]. In each iteration, the algorithm calculates the cohesion, coupling, and balance of individuals based on equations (4)-(6) to obtain comprehensive fitness. Through elite retention (retaining the top 10%), roulette wheel selection, two-point crossover, adaptive mutation, and 2-opt local search for elite individuals, the population is continuously updated and improved. The algorithm dynamically maintains a Pareto frontier solution set. When the maximum number of iterations or convergence condition is reached, it outputs the non-dominated solution set[25].

The second stage is the branch-and-bound local refinement stage: From the Pareto frontier obtained by the genetic algorithm, the optimization region is selected according to decision-maker preferences, retaining 80% of excellent genes in the neighborhood of the genetic algorithm's optimal solution as fixed allocations, and performing precise optimization on the remaining 20% of key products. The branch-and-bound tree uses the relaxed problem solution as the initial upper bound and the greedy solution as the initial lower bound. Through node selection (WBS hierarchy priority, critical path priority), branching operations (creating child nodes for each feasible task package, bound calculation, and pruning operations), the optimal solution is systematically searched[26]. When all nodes are processed or the time limit is reached, the algorithm checks if the improvement rate is less than 1%.If the improvement is less than this threshold for two consecutive rounds, it outputs the final task package partitioning scheme, including the comprehensive objective value, composition of each task package, and key performance indicators.

This hybrid solution strategy fully leverages the global search capability of genetic algorithms and the precise optimization capability of branch-and-bound algorithms. While ensuring solution quality, it effectively controls computational complexity, providing a practical solution for large-scale shipbuilding task package partitioning problems. The following content provides the detailed solution process[24].

3.1 Model Linearization Processing

Since the original model contains nonlinear terms, linearization processing is needed for solution.

(1) Quadratic term linearization

For the quadratic term in the coupling objective function, introduce auxiliary variables:

$$u_{ijlm}=x_{ij}\cdot x_{lm} \tag{29}$$

Add mcCormick inequality constraints:

$$\begin{aligned} u_{ijlm} &\le x_{ij} \\ u_{ijlm} &\le x_{lm} \\ u_{ijlm} &\ge x_{ij}+x_{lm}-1 \\ u_{ijlm} &\ge 0 \end{aligned} \tag{30}$$

Linearized coupling objective:

$$f_2'(X,U)=\sum_{j<m}\frac{\sum_{i,l}\delta_{il}\cdot u_{ijlm}}{n_j\cdot n_m} \tag{31}$$

(2) Piecewise Linearization of Information Entropy

For the entropy function, a piecewise linear approximation is performed on the interval [0,1]. The interval is divided into R sub-intervals, and on each sub-interval $[p_r, p_{r+1}]$:

$$h(p)\approx a_r\cdot p+b_r,\quad p\in[p_r,p_{r+1}] \tag{32}$$

Where the coefficients are determined by least squares fitting:

$$a_r=\frac{h(p_{r+1})-h(p_r)}{p_{r+1}-p_r}, \quad b_r=h(p_r)-a_r\cdot p_r \tag{33}$$

Binary variables are introduced to indicate that the probability falls into the r-th interval, with constraints added:

$$\sum_{r=1}^{R}\theta_{jvr}^{d}=1, \quad p_{dv}^{(j)}=\sum_{r=1}^{R}\theta_{jvr}^{d}\cdot\xi_r \tag{34}$$

Where ξr is the midpoint of the interval.

3.2 Genetic Algorithm Optimization

(1) Chromosome Encoding

An integer encoding scheme is adopted, with chromosome $C = [c_1, c_2,..., c_n]$, where $c_i \in \{1,2,...,k\}$ represents the task package to which product i belongs[17].

Decoding and repair process:

1) Precedence constraint checking: Scan all precedence relationships (i, l). If the task package number of product i is greater than that of product l, move l to the task package where i is located or to a subsequent task package;

2) Scale constraint adjustment: For task packages exceeding Lmax, transfer the excess products, prioritizing those with low dependency, to unsaturated task packages;

3) Empty task package handling: Merge task packages smaller than Lmin into adjacent task packages[21].

(2) Fitness Function Design

Fitness evaluation based on Pareto dominance relationship:

$$Fitness(C)=\frac{1}{1+n_{dom}(C)}+\sum_{C'\in dom(C)}\frac{1}{|P|} \tag{35}$$

Where ndom(C) is the number of individuals that dominate solution C, and dom(C) is the set of individuals dominated by C[14].

(3) Genetic Operators

Selection operator: Crowding-distance-based tournament selection to maintain population diversity.

Crossover operator: Improved two-point crossover considering task package integrity:

Mutation operator: Multi-point mutation with adaptive mutation rate:

$$p_m=p_{m0}+(p_{m1}-p_{m0})\cdot e^{-\alpha\cdot t/T} \tag{36}$$

Where t is the current generation, T is the maximum number of generations, and α is the decay coefficient.

Local search: Perform 2-opt local search on elite individuals to improve solution quality through swap operations[12].

(4) Diversity Preservation Mechanism

Niche technique: Sharing function based on Hamming distance:

$$sh(d_{ij})=\begin{cases}1-(d_{ij}/\sigma_s)^{\alpha}, & d_{ij}<\sigma_s\\ 0, & d_{ij}\geq\sigma_s\end{cases} \tag{37}$$

Elite preservation: Maintain an external archive to store Pareto front solutions.

Restart strategy: When population diversity falls below a threshold, preserve elite individuals and reinitialize part of the population[24].

3.3 Branch-and-Bound Algorithm

The branch-and-bound algorithm systematically enumerates task allocation schemes and performs pruning using specific constraints from ship construction.

(1) algorithm framework

The branch-and-bound algorithm systematically enumerates all possible task package allocation schemes and utilizes upper and lower bound pruning strategies to reduce the search space. The main steps of the algorithm include:

Step 1: initialization

Root node: all products in unassigned state

Upper bound: based on relaxed problem, i.e., the optimal value allowing product division;

Lower bound: empirical scheme based on shipyard historical data;

Step 2: branching strategy

For product i, create a branch for each feasible task package j

Feasibility judgment:

Task package j is not full, i.e., $T_j < L_{max}$

Precedence constraints are satisfied, i.e., all predecessors of i have been assigned to task packages ≤j

Stage span constraints are satisfied, i.e., the stage difference with existing products in j ≤δsmax

Step 3: bound calculation

Upper bound: current partial allocation + relaxed solution of remaining products

Lower bound: current partial allocation + greedy algorithm to complete remaining allocation

Step 4: pruning rules

Process pruning: if the current allocation violates critical process sequences, prune;

Resource pruning: If any team's load exceeds 120% of capacity limit, prune;

Spatial pruning: If too many products from non-adjacent areas are concentrated in the same task package, prune;

Step 5: Termination Condition

All nodes are processed or time iteration limits are reached.

(2) Acceleration Strategies

1) Heuristic initial solution generation: Generate high-quality initial solutions using hierarchical clustering algorithm based on product attribute similarity to improve the initial lower bound;

Group by WBS level: Products with the same parent node are preferentially assigned to the same task package

Cluster by construction stage: Prefabrication, assembly, and outfitting products are clustered separately;

Divide by spatial region: Products in adjacent areas are preferentially combined;

2) Constraint propagation: Utilize constraint relationships to derive variable value ranges and reduce search space.

Derive feasible task package ranges using precedence relationships;

Limit task package size based on resource capacity;

Exclude infeasible combinations based on process requirements;

3) Symmetry breaking: Add lexicographic constraints to eliminate equivalent solutions.

Strong branching: Perform tentative branching on candidate branch variables and select the variable that produces the maximum lower bound improvement[23].

3.4 Two-Stage Hybrid Solution Strategy

(1) Stage 1: Global Search with Genetic Algorithm
Objective: Rapidly explore the solution space to obtain an approximate Pareto front.
Population size: 100-200, corresponding to different task allocation schemes;
Number of iterations: 500-1000, ensuring convergence;
Crossover rate: 0.85, maintaining scheme diversity
Mutation rate: 0.01-0.10, adaptively adjusted
Output: Pareto front solution set
(2) Stage 2: Local Optimization with Branch-and-Bound
Objective: Perform precise optimization on excellent schemes obtained from the genetic algorithm.
Input: Neighborhood of genetic algorithm optimal solutions, similar task allocation schemes, and decision-maker preferences, such as emphasizing resource balance;
Search strategy:
Fix 80% of product allocations, the excellent genes from the genetic algorithm;
Perform precise optimization on the remaining 20% of products;
Focus on optimizing product allocation on critical paths;
Output: Local optimal solution, an exact optimal scheme satisfying all constraints.
(3) Collaborative Mechanism:
1) Information sharing: The optimal solution from the genetic algorithm serves as the initial bound for branch-and-bound;
2) Region division: Divide key optimization regions based on ship construction characteristics: critical path products, resource-intensive products, cross-disciplinary interface products;
3) Iterative improvement:
Round 1: Genetic algorithm exploration → Branch-and-bound refinement;
Round 2: Adjust weights based on Round 1 results → Re-optimization;
Convergence condition: Improvement in two consecutive rounds < 1%;

4. Case Verification and Analysis

4.1 Case Background and Data

Table 1 WBS Product Unit Attribute Data

Product ID	WBS Code	Stage	Type	Region	System	Workload	Predecessor
P1	1.1.1.0	S1-Prefab	T1-Hull	Z1-Bottom	F1-Structure	1.00	-
P2	1.1.2.0	S1-Prefab	T1-Hull	Z1-Bottom	F1-Structure	1.00	-
P3	1.1.3.0	S1-Prefab	T2-Piping	Z1-Bottom	F2-Piping	1.20	-
P4	1.2.1.0	S2-Assembly	T3-Outfitting	Z1-Bottom	F1-Structure	1.50	P1
P5	1.2.2.0	S2-Assembly	T3-Outfitting	Z1-Bottom	F1-Structure	1.50	P2
P6	1.2.3.0	S2-Assembly	T3-Outfitting	Z2-Middle	F1-Structure	1.50	P3
P7	2.1.1.0	S2-Assembly	T4-Electrical	Z2-Middle	F2-Piping	1.80	P3
P8	2.1.2.0	S2-Assembly	T4-Electrical	Z2-Middle	F2-Piping	1.80	P3
P9	2.2.1.0	S3-Install	T5-Equipment	Z2-Middle	F2-Piping	2.00	P7
P10	2.2.2.0	S3-Install	T5-Equipment	Z2-Middle	F3-Electrical	2.00	P8
P11	3.1.1.0	S3-Install	T6-Coating	Z3-Upper	F3-Electrical	2.20	P9
P12	3.1.2.0	S3-Install	T6-Coating	Z3-Upper	F3-Electrical	2.20	P10

Note: 1.0 - Predecessor dependency; 0.6 - Sibling relationship; 0.3 - Adjacent stages; 0.1 - Same system

Taking the port side section of the cargo hold and the cargo hold top and bottom sections above and below it as objects, verification of the man-hour division method for intermediate products with four-dimensional features is conducted. It contains 12 main WBS product units that need to be divided into 3 task packages. Product attribute data are shown in Table 1.

4.2 Parameter Settings

Model parameters are set based on expert experience and calibrated from historical data:
Dimension weights: w_s = 0.35,w_t = 0.30, w_z = 0.20, w_f = 0.15
Objective weights: λ_1 = 0.50, λ_2 = 0.30, λ_3 = 0.20
Task package size: L_{min} = 3, L_{max} = 5
Algorithm parameters:
Genetic Algorithm: Population size M = 150, Max generations G = 800, Crossover rate pc = 0.85, Mutation rate pm $\in$ [0.01, 0.10]
Branch and Bound: Maximum nodes 10000, Time limit 300 seconds

4.3 Optimization Results Analysis

(1) Convergence Analysis
The algorithm converged around 600 generations, with the overall objective function value improving from the initial -0.425 to 0.064, an improvement rate of 115.1%. Optimization Results Analysis data are shown in Table 2.

Table 2 Multi-Objective Optimization Convergence Process

Iterations	Cohesion f_1	Coupling f_2	Load Variance f_3	Overall F
0	0.683	0.245	4.832	-0.425
800	0.833	0.100	1.615	0.064

(2) Practical Application Effects
Practical Application Effectsdata are shown in Table 3.

Table 3 Practical Application Effect Statistics

Evaluation Indicator	Before	After	Improvement	Notes
Avg. Construction Cycle (days)	85	64	-24.7%	Per task package
Waiting Time (hours)	156	102	-34.6%	Cumulative waiting
Rework Rate (%)	8.3	4.2	-49.4%	Quality issues
Equipment Utilization (%)	68.5	87.2	+27.3%	Key equipment
Man-hours (person hours)	4250	3187	-25.0%	Direct labor
Material Waste Rate (%)	5.6	3.8	-32.1%	Steel, etc.

4.4 Discussion

(1) The information entropy-based cohesion quantification method can objectively evaluate the homogeneity within task packages, providing a theoretical basis for task package quality assessment. Case calculations show that the cohesion of optimized task packages reaches 0.833, a 22.1% improvement over the empirical method.

(2) The coupling matrix integrating WBS hierarchical structure, predecessor dependencies, spatial adjacency, and system associations accurately characterizes the technical constraints of shipbuilding. The optimized scheme reduces the coupling between task packages to 0.100, a 58.0% reduction.

(3) The load balance evaluation based on Jain's Fairness Index ensures reasonable resource allocation, reducing load variance by 66.5% and effectively avoiding resource bottlenecks.

(4) The two-stage solution strategy combining branch-and-bound with improved genetic

algorithms achieves a good balance between solution quality and computational efficiency, improving the overall objective function value by 115.1%.

(5) Practical application verification demonstrates the effectiveness of the method, improving construction efficiency by 24.7% and reducing waiting time by 34.6%, providing shipbuilding enterprises with practical and viable optimization solutions.

5. Conclusions

This paper addresses the task package partitioning problem of WBS product units in large shipbuilding processes by establishing a multi-objective optimization model that comprehensively considers cohesion, coupling, and load balancing. The main conclusions are as follows:

The developed multi-objective optimisation model overcomes the limitations of traditional single-objective partitioning methods by integrating three core objectives: cohesion, coupling, and load balancing. This model precisely aligns with the hierarchical characteristics of large-scale shipbuilding WBS product units, effectively mitigating issues such as task packages being 'internally loose and externally tight' (weak internal process interdependencies coupled with frequent cross-package coordination) or 'resource overload'. It provides a scientifically quantifiable basis for task package division[18].

The task package division methodology based on this model demonstrates significant practical value in shipbuilding scenarios. On one hand, it enhances cohesion among similar processes within task packages, reducing cross-disciplinary and cross-departmental coordination frequency. On the other hand, it reasonably controls load fluctuations of critical resources (such as welding equipment and painting stations), preventing resource idleness or overload, thereby ensuring stable project progress.

This research validates the applicability of multi-objective optimisation approaches within complex shipbuilding projects. The methodology proves adaptable for WBS task package division across varying tonnages and vessel types (e.g., bulk carriers, container ships), while also furnishing standardised tools for optimising task organisation and dynamic resource allocation in shipbuilding projects[22]. It holds considerable reference value for other large-scale, multi-disciplinary collaborative engineering endeavours, such as offshore platform construction.

References

[1] Cheng J, Ming J, Chen X. Distribution network reliability assessment and optimization method based on convolutional neural network[J].Engineering Research Express,2025,7(4):045339.DOI:10.1088/2631-8695/AE0F08.

[2] Takahashi S, Sasaki M, Takeda K, et al. Investigating fine- and coarse-grained structural correspondences between deep neural networks and human object image similarity judgments using unsupervised alignment[J].Neural Networks, 2026, 195108222.DOI:10.1016/J.NEUNET.2025.108222.

[3] Yong S, Lianlong G, Lianjie L, et al. An inverse model control method based on signal filtering and BP neural network to improve performance of electric servo cylinder system[J].Proceedings of the Institution of Mechanical Engineers, 2025, 239(22):9073-9087.DOI:10.1177/09544062251359409.

[4] Sewioło M, Mystkowski A. Agriculture Machine fault detection based on multiple Input-Parallel-Neural network optimized by genetic algorithm with matrix chromosomes[J].Computers and Electronics in Agriculture, 2025, 239(PC): 111125.DOI:10.1016/J.COMPAG.2025.111125.

[5] Kenanda M, Hammadi F, Bahai H, et al. A new efficient nonlocal hyperbolic HSDT for mechanical vibration of porous FGM plates/nanoplates using Navier's method and artificial neural network prediction[J].International Journal of Solids and Structures,2026,325113719.DOI:10.1016/J.IJSOLSTR.2025.113719.

[6] Najamuddin, Sheikh U U, Sha'ameri Z A. Marine vessels acoustic classification with enhanced machinery and propeller feature extraction, using convolutional neural network[J].Measurement,2026,257(PA):118494. DOI: 10. 1016/J.MEASUREMENT.2025.118494.

[7] Li C, Zhao X, Lin L, et al. An evolutionary knowledge training-based proximal policy optimization algorithm for job shop scheduling in flexible intelligent manufacturing[J].Computers & Industrial Engineering,2025,210111533. DOI:

10.1016/J.CIE.2025.111533.
[8] Tan D, Yan L, Zhao J, et al. A black-box attack method of machine learning algorithms based on quantum autoencoders[J]. Physica A: Statistical Mechanics and its Applications,2025,680131033.DOI:10.1016/J. PHYSA. 2025.131033.
[9] Ai W, Jiang Y, Pan X, et al. Neural network control of robotic manipulators with full-state time-varying constraints and composite disturbance observer[J].Journal of the Franklin Institute,2025,362(17):108084.DOI:10. 1016/J. JFRANKLIN. 2025.108084.
[10] Liang J, Sun Z, Kang J, et al. Highly efficient Coordinate Measuring Machine error compensation via Greedy Randomized Kaczmarz algorithm and nongeometric error identification neural network[J]. Measurement, 2026, 258(PB): 119091.DOI:10.1016/J.MEASUREMENT.2025.119091.
[11] Zhuang H, Lu W, Shen Q, et al. Off-policy reinforcement learning control for space manipulators based on object detection via convolutional neural networks[J].Aerospace Science and Technology,2026,168(PD):110914.DOI: 10.1016/J.AST.2025.110914.
[12] Darwhekar M,Devi P Y,Murty N T, et al.Automated hydride segmentation and quantification in zirconium alloys using a convolutional neural network-assisted machine learning model[J].Journal of Nuclear Materials, 2026, 618156209.DOI:10.1016/J.JNUCMAT.2025.156209.
[13] Ali H I,Alqahtani F M,Eljack D N, et al.Experimental and Machine Learning Modelling of Ni(II) Ion Adsorption onto Guar Gum: Artificial Neural Network (ANN) and K-Nearest Neighbor (KNN) Comparative Study[J]. Polymers, 2025, 17(20):2791.DOI:10.3390/POLYM17202791.
[14] Xu Y X,Wu C G,Xie D.A new discrete fractional AMAR model for finance time series forecasting by machine learning[J].Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena,2025,201(P2):117296.DOI:10.1016/J.CHAOS.2025.117296.
[15] Chen Q,Sabir Z,Mehmood A M, et al.A machine learning radial basis deep neural network for solving the fractional chaotic financial system[J].Journal of Computational and Applied Mathematics,2026,474116936.DOI: 10. 1016/J. CAM.2025.116936.
[16] Wang R,Lyu Z,Yan F, et al.A Conditional Adaption Alignment Dynamic Graph Neural Network model for unsupervised fault diagnosis of rotating machinery[J].Mechanical Systems and Signal Processing,2025,240113361. DOI:10.1016/J.YMSSP.2025.113361.
[17] Yao W,Wang Y,Xiong L, et al.Dynamics of asteroidal Hopfield neural network under electromagnetic radiation and its application in the mechanical optimization design[J].Nonlinear Dynamics,2025,(prepublish):1-18.DOI:10. 1007/S11071-025-11846-1.
[18] Pawlik L,Jakubowski W L J,Frej D, et al.Applications of Computational Mechanics Methods Combined with Machine Learning and Neural Networks: A Systematic Review (2015–2025)[J].Applied Sciences,2025,15(19):10816. DOI:10.3390/APP151910816.
[19] Fadzail F N,Zali M S,Mid C E, et al.Hyperparameter optimization using automated machine learning based artificial neural network for fault detection in wind turbine generator[J].Next Research,2025,2(4):100785.DOI: 10. 1016/J.NEXRES.2025.100785.
[20] Moghadamnejad A,Moghaddasi A M,Hamidia M, et al.Ranking Earthquake Prediction Algorithms: A Comprehensive Review of Machine Learning and Deep Learning Methods[J].Soil Dynamics and Earthquake Engineering,2026,200(PA):109740.DOI:10.1016/J.SOILDYN.2025.109740.
[21] Hasan R S,Islam S M,Awal I Z, et al.Prediction and optimization of efficient ship design particulars through advanced machine learning approaches[J].Ocean Engineering,2025,341(P2):122572.DOI:10.1016/J. OCEANENG. 2025.122572.
[22] Li H,Zeng Z,Li X, et al.Graph-convolutional neural networks for predicting tunnel boring machine performance[J]. Automation in Construction,2025,180106436.DOI:10.1016/J.AUTCON.2025.106436.
[23] Huang Y,Tian J.Motion path optimization of truss manipulator based on simulated annealing and BP neural network[J]. Frontiers in Mechanical Engineering,2025,111643848.DOI:10.3389/FMECH.2025.1643848.
[24] Petrov A P.An Overview of Forming Problems Solved with Machine and Deep Learning[J].Steel in Translation, 2025, 55(5):451-458.DOI:10.3103/S0967091225700925.
[25] Arshad Z,Hussain K S,Othman A N, et al.Application of machine learning for numerical analysis of mixed convective Falkner–Skan flow of cross nanofluid through predictive neural networks algorithm[J].International Journal of Geometric Methods in Modern Physics,2025,(prepublish):DOI:10.1142/S0219887825502688.
[26] Houssein H.An artificial neural network for predicting the roles of contacting bodies in computational contact mechanics[J].Neural Computing and Applications,2025,(prepublish):1-23.DOI:10.1007/S00521-025-11654-Z.