Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076620

ARTICLE

Multi-Objective Optimization of Defective Multi-Inventory Mother-Plate Cutting

Changtian Zhang[1], Qi Zhang[1,*], Shujin Qin[2], Xiwang Guo[3], Bin Hu[4,*] and Wenjie Luo[1]

[1]College of Information Engineering, Shenyang University of Chemical Technology, Shenyang, China
[2]School of Information and Technology, Shangqiu Normal University, Shangqiu, China
[3]College of Information and Control Engineering, Liaoning Petrochemical University, Fushun, China
[4]Department of Computer Science and Technology, Kean University, Union, NJ, USA

*Corresponding Authors: Qi Zhang. Email: qizhang@syuct.edu.cn; Bin Hu. Email: bin.hu@kean.edu

Received: 23 November 2025; Accepted: 23 February 2026; Published: 08 May 2026

ABSTRACT: The increasing complexity of steel manufacturing and the rising demand for customized high-grade plates have intensified the need for efficient and defect-aware cutting optimization. In practical production, mother plates frequently contain multiple surface defects, and the cutting process is further constrained by delay-sensitive operations such as tool-change sequences and defect-tolerance requirements. To address these challenges, this study formulates the Defective Multi-Inventory Mother-Plate Two-Dimensional Cutting Stock Problem (DMMP-2CSP) as a multi-objective model that simultaneously maximizes cutting profit and minimizes tool changes under strict geometric and defect-avoidance constraints. We develop an Improved Multi-Objective Grey Wolf Optimizer (IMOGWO) featuring continuous random-keys encoding with hierarchical decoding to handle multi-plate, multi-defect layouts; a Large-Language-Model-guided Fourth-Leader Boost mechanism that adaptively mitigates stagnation through domain-informed auxiliary-leader generation; and an NSGA-II fusion module incorporating non-dominated sorting, crowding-distance control, and stochastic variation to balance exploration and exploitation throughout the search. Extensive experiments on industrial-scale datasets demonstrate that IMOGWO consistently produces well-distributed Pareto-optimal solutions, significantly improves cutting profit, reduces tool-change frequency, and achieves superior overall performance compared with classical Multiobjective Grey Wolf Optimizer, Multiobjective Particle Swarm Optimization, Multi-Objective Cuckoo Search, and Multi-Objective Snake Optimizer baselines.

KEYWORDS: Defective plate cutting; multi-objective optimization; metaheuristic optimization; Large-Language-Model (LLM)-assisted decision support; Pareto-based search

1 Introduction

Steel plates are indispensable to national defense and critical infrastructure projects, with widespread applications in demanding fields such as shipbuilding, bridge construction, nuclear power generation, and petrochemical equipment. This manufacturing landscape is increasingly characterized by customized high-performance requirements, frequent specification adjustments, and a distinct market imbalance—with overcapacity in commodity-grade plates contrasting with a scarcity of specialized grades. Accordingly, manufacturers face increasing pressure to improve material utilization, reduce costs, and promote sustainable practices—driving a shift toward intelligent, resource-efficient and waste-minimizing production systems [1].

Within the steel production workflow, the cutting of the mother-plate constitutes one of the most critical and resource-intensive stages. It serves as the essential link between upstream plate manufacturing and downstream customer order fulfillment. Cutting operations determine not only whether orders are satisfied but also how efficiently raw material is used. However, the combination of highly diverse specifications, multi-size demand patterns, and small-batch orders leads to substantial complexity in planning and layout design. Conventional cutting approaches frequently result in low material utilization, high scrap generation, and unnecessary tool-change operations, all of which increase cost, prolong processing time, and hinder sustainable production objectives. Formally, the core decision-making process can be modeled as the classic two-dimensional cutting stock problem (2CSP), a well-known variant of the two-dimensional bin-packing problem [2]. The pursuit of higher utilization and optimized layout design has thus become fundamental for intelligent, energy-efficient, and environmentally conscious steel manufacturing.

In real industrial settings, the complexity of cutting operations is further amplified by the presence of surface and internal defects in the rolled plates. Defects such as inclusions, crazing, patches, scratches, oxide scale, and micro-cracks arise commonly during casting, rolling, heat treatment, and surface finishing [3]. As illustrated in Fig. 1, these flaws reduce the usable area of each mother plate effectively and directly affect the yield of the material. Traditional cutting strategies typically address this issue by treating defective regions as completely unusable [4,5]. Although this avoids quality issues, it also leads to significant waste. However, in practice, many customer orders, depending on application requirements, tolerance specifications, and structural considerations, can accept certain types or severities of defects. Allowing such orders to be placed over tolerable defects increases the usable area of defective inventory, improves total yield, and aligns closely with sustainable manufacturing principles such as waste minimization, circular resource use, and improved material efficiency.

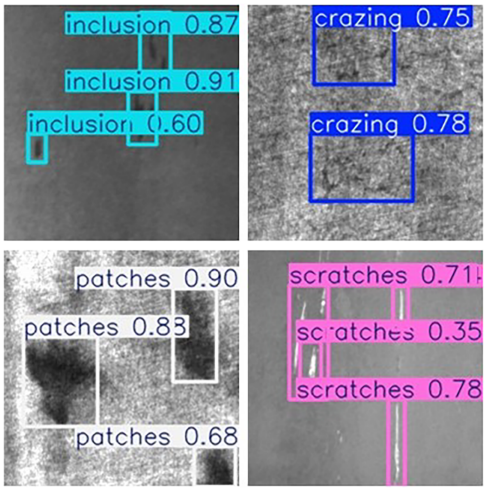


Figure 1: Representative photographs of typical surface defects in steel plates.

In industrial steel plate cutting, mother plates often contain heterogeneous defects that render parts of the material unusable—yet many customer orders can tolerate certain defect types or locations. Ignoring this tolerance leads to unnecessary waste and undermines sustainability. To address this gap, we introduce the Defective Multi-Inventory Mother-Plate Two-Dimensional Cutting Stock Problem (DMMP-2CSP): a realistic extension of the classical 2CSP that simultaneously (i) maximizes profit through improved material utilization by strategically placing items on defective plates according to order-specific defect acceptability, and (ii) minimizes tool-change operations to reduce energy use and maintain production

continuity. The problem integrates guillotine cutting, geometric feasibility, defect-avoidance/tolerance rules, and multi-inventory compatibility—posing significant combinatorial and practical challenges that motivate our methodological contributions.

Solving the DMMP-2CSP is computationally challenging. The 2CSP is NP-hard, and the additional complexities of defect modeling, multi-plate inventories, and multi-objective trade-offs render classical exact techniques such as branch-and-bound and column generation inefficient for industrial-scale instances [6]. Exhaustive search rapidly becomes impractical as the instance size increases. Stochastic and heuristic algorithms offer improved scalability but often struggle with premature convergence, limited exploration, and difficulty navigating feasible regions constrained by defects and guillotine operations. These limitations motivate the development of advanced metaheuristic strategies capable of delivering high-quality solutions under realistic industrial constraints.

To address these challenges, this work proposes an Improved Multi-Objective Grey Wolf Optimizer (IMOGWO). The algorithm integrates domain-specific knowledge of plate cutting with a hierarchical real-valued encoding-decoding strategy, a Large-Language-Model-guided Fourth-Leader Boost mechanism that enhances global exploration during stagnation, and NSGA-II-inspired diversity preservation operators. These innovations collectively improve convergence speed, population diversity, and robustness when solving highly constrained, defect-aware industrial cutting problems.

This study makes fundamental contributions at both the problem formulation and algorithmic levels. In modeling, we propose the DMMP-2CSP, which is novel in simultaneously integrating multiple defective plates with customer-specific defect-tolerance rules and elevating tool-change minimization to a primary objective alongside profit. Algorithmically, we develop the IMOGWO. Its core innovation is a deeply hybrid architecture that uniquely couples an LLM-guided stagnation-breaking mechanism, a problem-specific hierarchical decoder ensuring layout feasibility, and NSGA-II principles for diversity preservation. This framework demonstrates a practical and effective strategy for integrating AI-driven guidance into evolutionary multi-objective optimization for complex industrial problems.

The main contributions of this work are as follows:

1. A comprehensive multi-objective mathematical model for defective mother-plate cutting that explicitly incorporates geometric feasibility, dimensional constraints, defect tolerance rules, guillotine cutting structure, and tool-change costs, thereby reflecting real industrial operations.
2. An IMOGWO that integrates random-keys representation, LLM-guided exploration, and NSGA-II-based elitism to efficiently solve large-scale DMMP-2CSP instances under complex constraints.
3. Extensive experimental validation using three industrial-scale datasets synthetically generated from real production parameters of a collaborating steel manufacturer, demonstrating significant improvements in profit, material utilization, scrap reduction, and tool-change minimization compared with state-of-the-art baselines.

The remainder of this paper is organized as follows. Section 2 reviews related work on defective plate cutting, modeling approaches, and multi-objective optimization algorithms. Section 3 presents the formal mathematical formulation and parameter definitions. Section 4 details the architecture and implementation of the proposed IMOGWO algorithm. Section 5 reports computational experiments and comparative analyses. Finally, Section 6 concludes the paper and discusses directions for future research.

2 Literature Review

The defective multi-inventory mother-plate cutting problem (DMMP-2CSP) can be broadly situated within the family of two-dimensional bin packing and cutting stock problems, which have long served as

foundational abstractions for resource allocation, material utilization, and layout planning in manufacturing. At its core, the DMMP-2CSP can be formulated as a two-dimensional bin packing problem (2D-BPP), in which a set of rectangular items (order plates) is arranged within rectangular bins (mother plates) of fixed dimensions. The 2D-BPP is an NP-hard combinatorial optimization problem whose computational complexity remains extremely high; even instances with only a few dozen items may lack provably optimal solutions after extensive search. As a multidimensional extension of the one-dimensional bin packing model [7], the 2D-BPP provides an effective representation of industrial layout problems where geometric constraints, limited inventory resources, and spatial compatibility are central. Typical objectives involve minimizing the number of bins used or minimizing wasted area, ensuring both efficient resource utilization and high-performing layouts. To address layout optimization involving multiple inventory sizes, Furini and Malaguti [8] proposed a two-dimensional two-stage cutting stock model aimed at minimizing the total area of larger plates needed to satisfy production requirements.

In industrial cutting and nesting operations, cutting constraints strongly influence both production efficiency and material utilization. Guillotine cutting constraints, in particular, require each cut to traverse the entire material horizontally or vertically, splitting the plate into two independent sections. These rules are widely used across wood, glass, textile, and metal manufacturing due to their inherent operational stability and safety. Silva et al. [9] introduced one of the earliest integer programming formulations incorporating guillotine cutting constraints, establishing a foundation for subsequent quantitative studies. To improve solution quality on large-scale problem instances, Gardeyn and Wauters [10] developed a destroy-and-reconstruct heuristic complemented by a goal-driven improvement mechanism. Their method minimized total bin area under guillotine and rotation rules and demonstrated strong performance across numerous benchmarks.

Beyond the traditional assumptions of uniform material quality and geometric regularity, researchers have increasingly introduced spatial heterogeneity into bin packing models to better capture realistic industrial conditions. Zhou et al. [11] proposed a two-dimensional knapsack model with block-packing constraints, where the container is divided into discrete blocks representing irregular or defective regions. Items must be fully contained within permitted blocks, and the objective is to maximize total utilization. This formulation provides a more realistic approximation of material structures affected by internal damages, surface flaws, or nonstandard shapes.

In summary, traditional studies on the 2D-BPP and its variants have primarily emphasized geometric feasibility, dimensional constraints, and guillotine-cutting rules. However, relatively few works explicitly incorporate material defects, despite their prevalence in steel manufacturing. Surface or internal defects directly affect material usability, cutting feasibility, and product quality, rendering conventional idealized models insufficient for industrial deployment. Models that ignore defects often lead to significant scrap, reduced yield, and inefficiencies contrary to both economic objectives.

To capture these real-world complexities, the DMMP-2CSP extends classical bin packing and cutting stock models by including multiple fixed-size mother plates with distinct defective regions. Defects alter the effectively usable area of each plate and impose additional spatial and quality constraints within the layout process. As such, the DMMP-2CSP can be interpreted as a defective and multi-inventory extension of the classical 2D-BPP, grounded in operational realities and aligned with sustainable manufacturing goals involving waste minimization and efficient resource allocation., the DMMP-2CSP can be regarded as a defective and multi-inventory extension of the classical bin packing problem, reflecting the operational realities of industrial steel plate cutting.

The defective cutting problem, as a broader research category [12,13], concerns the cutting of raw rectangular materials containing one or more defects into smaller workpieces, with objectives often focused on minimizing waste or maximizing total recovered value. Numerous studies [14,15] have explored diverse

formulations and algorithms to accommodate industrial constraints, including defect acceptance tolerances, quality grading rules, and cost structures.

Among the earliest efforts to consider defect variation and its influence on cut-piece value were the dynamic programming models introduced by Gilmore and Gomory [16]. Their approach assessed the placement of items relative to defective regions in both one- and two-dimensional settings and influenced decades of subsequent research. Later extensions [17–19] refined feasibility tests, enhanced branching rules, and integrated preprocessing techniques. Neidlein et al. [20] proposed a branch-and-bound framework with greedy heuristics to address infeasibility arising from defect avoidance; their model accommodated a single rectangular defect and demonstrated strong solution quality. Afsharian et al. [21] expanded to multiple-defect scenarios via a dynamic programming–based heuristic that significantly improved efficiency and value recovery across industrial-scale instances.

To handle the complexity of large-scale layouts and quantity constraints, Martin et al. [22] developed a Benders decomposition approach for guillotine-based defective cutting. Their formulation balanced feasibility and computational efficiency and was later extended through a constraint programming model [23], which demonstrated state-of-the-art performance on benchmark datasets. Working under non-guillotine assumptions, Yao et al. [24] proposed a decomposition-based framework separating the original problem into a continuous master problem and an integer-based x-Check subproblem responsible for feasibility verification. Valid inequalities, preprocessing techniques, and cut-lifting procedures enhanced performance across all 5450 benchmark instances reported in the literature.

Further exploring defect-aware layout optimization, Yao et al. [25] investigated the two-dimensional strip packing problem with defects (2DSPP_D), aiming to minimize strip height under fixed-width constraints with defective regions. Their two-phase exact method combined integer programming and Benders decomposition, along with preprocessing, lower bounds, valid inequalities, and heuristic skyline-based search to produce highly competitive results. More recent research, such as Zhang et al. [26], has highlighted the importance of column generation and decomposition strategies for handling complex defective structures while maintaining high utilization and solution quality.

Beyond exact algorithms, recent works have increasingly recognized that large industrial cutting problems demand scalable and flexible optimization frameworks. Metaheuristics have thus been widely adopted due to their robustness and ability to explore complex search spaces with heterogeneous constraints. Grey Wolf Optimizer (GWO)–based techniques, in particular, have shown strong performance in solving complex layout and scheduling problems, including profit-maximizing human–robot collaborative disassembly [27], demonstrating the potential of GWO variants for large-scale combinatorial optimization.

The Multi-Objective Grey Wolf Optimizer (MOGWO), introduced by Mirjalili et al. [28], extends the standard GWO by incorporating a fixed-size external archive to preserve Pareto-optimal solutions. MOGWO simulates the hunting behavior and hierarchical leadership structure of grey wolves to drive multi-objective exploration, maintaining a robust balance between exploitation and exploration. Experimental studies show that MOGWO achieves competitive performance relative to other classical multi-objective algorithms.

Since its introduction, MOGWO has been successfully applied to various combinatorial, scheduling, and resource allocation problems, including manufacturing scheduling [29], distributed resource management [30], multiprocessor maintenance and diagnostic scheduling [31], fog-based IoT task scheduling [32], and electrochemical machining process optimization [33]. These studies highlight the algorithm's strong generalization ability and adaptability to heterogeneous constraints.

In recent years, multi-objective combinatorial optimization has gained importance in increasingly complex industrial systems, such as human–robot collaborative disassembly [34,35] and multi-factory remanufacturing and scheduling [36,37]. These efforts reflect a broader industrial trend toward intelligent, scalable, and sustainability-driven optimization frameworks capable of supporting the challenges of Industry 4.0 and sustainable production.

Finally, it is important to note the growing intersection between defective cutting and material efficiency. Although historically underexplored in cutting stock research, considerations such as minimizing scrap and reducing carbon-intensive raw material consumption have become increasingly important. Defect-aware optimization models such as DMMP-2CSP therefore represent not only a methodological advancement but also a strategic lever for achieving higher material utilization.

3 Problem Description and Modeling

3.1 Problem Description

To accurately represent the actual production process of steel plate cutting, the entire workflow is conceptualized as a three-stage hierarchical procedure, as illustrated in Fig. 2. In the first stage (transverse cutting), a cutting shear is used to perform a full-length transverse cut along the longitudinal direction of the mother plate, by which the plate is divided into several elongated strips having the same width as the original plate. These strips correspond to the primary shelves defined in the proposed model. In the second stage (longitudinal cutting), a blade holder equipped with disc shears is employed to cut each primary shelf along its width, thereby partitioning it into multiple independent rectangular regions, which represent the secondary shelves in the model. In the final stage (finishing cutting), manual operations or high-precision cutting equipment are applied to the secondary shelves to obtain finished plates that satisfy the dimensional and quality requirements of customer orders. Through this three-stage hierarchical process, comprising transverse full-length cutting, longitudinal strip cutting, and precision finishing, the structure of industrial steel plate cutting is faithfully reproduced, ensuring consistency between the mathematical model and real production practice.

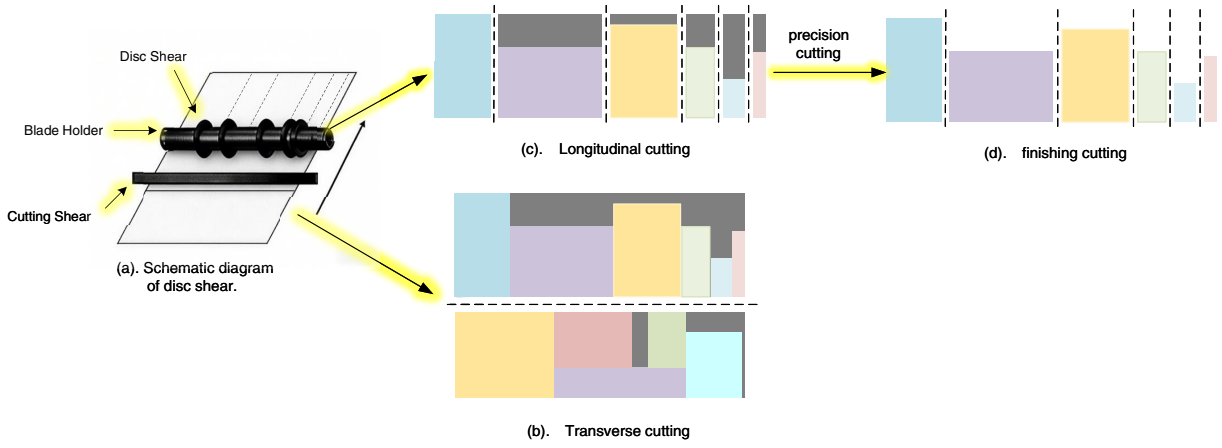


Figure 2: Enter caption.

A key industrial challenge addressed in this study is the presence of localized surface or internal defects, such as oxide scales, inclusions, cracks, and scratches, on mother plates. These defects introduce geometric complexity and often reduce usable area in traditional cutting plans. Conventional nesting strategies typically avoid defective regions entirely, which leads to substantial scrap and lowers material utilization. From

a sustainability standpoint, such waste is detrimental, as it increases raw-material consumption, energy expenditure in upstream steelmaking, and total carbon footprint.

However, in practice, not all order plates require defect-free surfaces. Many downstream applications allow for specific defect types or tolerances depending on structural, mechanical, or aesthetic requirements. Accounting for these defect-tolerance conditions significantly enhances the usable area of mother plates and reduces unnecessary scrap. To capture this industrially realistic behavior, the proposed model explicitly incorporates defect-avoidance constraints based on individual order-plate acceptance rules. As illustrated in Fig. 3, Order Plates I and II avoid Defect a, while Order Plate III is placed over Defect b due to its defect-tolerance specification.

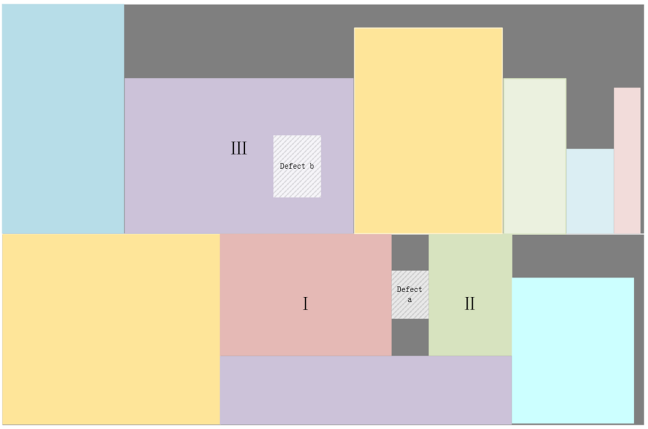


Figure 3: Schematic illustration of the defect-avoidance constraint.

The optimization problem in this study is centered on defective mother plates that contain localized surface or internal flaws such as oxide scales, inclusions, or cracks. The presence of these defects increases the geometric complexity of the layout process and often results in substantial material waste when conventional nesting strategies are applied. In actual industrial settings, however, not all order plates require completely defect-free surfaces; depending on their functional requirements and quality standards, some plates can tolerate specific types or degrees of defects. Accordingly, defect-avoidance constraints are incorporated into the proposed model to reflect such practical considerations. When an order plate cannot accept a particular defect, it is constrained to be placed entirely outside the corresponding defective region, as illustrated in Fig. 3, where Order Plates I and II avoid Defect a. Conversely, if a plate is defect-tolerant, placement over the defective region is permitted, as shown by Order Plate III accepting Defect b.

To develop a mathematically tractable and industrially implementable optimization model, several assumptions are adopted:

First, considering the limited quantity of mother plates in inventory and the sufficient, unrestricted demand for order plates, it is assumed that the inventory can meet production requirements, thus eliminating the need to account for stock shortages. Second, to avoid addressing issues related to damage, all inventory mother plates are assumed to remain effectively usable during the cutting process without becoming non-functional. The dimensions of all inventory mother plates and order plates are known and fixed; this assumption simplifies geometric constraints, allowing the model to focus on the efficient utilization of these fixed-size materials.

Regarding the layout of order plates, it is assumed that all order plates are rectangular and can be rotated by 90° as needed to enhance layout flexibility and material utilization efficiency. For defective regions on

inventory mother plates, their positions and sizes are assumed to be known and fixed, enabling the model to precisely handle and optimize around these defective areas. Considering that certain order plates in actual production can tolerate specific types or degrees of defects, it is assumed that some order plates possess defect tolerance, thereby avoiding excessive exclusion of defective regions and improving material utilization. The demand quantity for each type of order plate is assumed to be known and constant, simplifying demand forecasting and ensuring demand stability during production.

Furthermore, the cutting tools used in the process are assumed to have sufficient cutting capacity to successfully complete all cutting tasks without causing damage to inventory mother plates. To simplify the model, it is assumed that only one tool change occurs in each cutting stage, with the associated time and cost being known and fixed. Finally, the scrap recycling and disposal processes are assumed to be known, and the recycling value of scrap is fixed.

These assumptions mirror real industrial constraints while maintaining the tractability of the optimization model. They also support sustainability objectives by enabling fine-grained control over waste generation, defect utilization, and cutting feasibility.

1. Inventory plates are limited, and order plates are sufficient.
2. All inventory plates are usable.
3. Dimensions of inventory plates are fixed.
4. All order plates are rectangular.
5. Order plates can be rotated during layout.
6. Defective regions on inventory plates are fixed.
7. Order plates have defect tolerance.
8. Demand for all order plates is known and constant.
9. Cutting tools have sufficient performance.
10. No loss or damage occurs during cutting.
11. Tool changes occur once per cutting stage.
12. Scrap recycling and disposal are known.

3.2 Mathematical Model

The objective of this model is to determine the optimal cutting layouts for defective mother plates, including the assignment of order plates, cutting sequence design, and tool-change minimization. A mixed-integer nonlinear programming (MINLP) formulation is developed to maximize the total economic profit while minimizing the number of tool changes, subject to geometric non-overlapping, defect-avoidance, and guillotine-cutting constraints.

The model takes the following as inputs: (1) mother-plate inventory (dimensions, defect maps, defect types and positions); (2) order-plate specifications (dimensions, quantities, defect-tolerance rules); (3) economic parameters (unit selling price, mother-plate cost, scrap recycling price). The outputs of the model include: (1) cutting layout for each mother plate (primary and secondary shelf structures); (2) placement and rotation of each order plate; (3) total profit and tool-change count; and (4) scrap area and material utilization rate.

The parameters, sets, and constants used in the model are defined in Table 1, the auxiliary variables introduced for model linearization and geometric control are summarized in Table 2, and the continuous and integer decision variables describing cutting dimensions, positions, and sequencing are listed in Table 3.

Binary Decision Variables:

$$x_{sj} = \begin{cases} 1, & \text{if primary shelf } s \text{ is used on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$x_{s2sj} = \begin{cases} 1, & \text{if secondary shelf } s2 \text{ is used in primary shelf } s \\ & \text{of mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_{is2sj} = \begin{cases} 1, & \text{if order plate } i \text{ is placed in secondary shelf } s2 \\ & \text{of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$r_{isj} = \begin{cases} 1, & \text{if order plate } i \text{ is rotated } 90^\circ \text{ on primary shelf } s \\ & \text{of mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_{s-1,s,j} = \begin{cases} 1, & \text{if tool change required between primary shelves } s-1 \\ & \text{and } s \text{ of mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$z_{s,j} = \begin{cases} 1, & \text{if nonzero difference between primary shelves } s-1 \\ & \text{and } s \text{ of mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$u^{w,p}_{i,s,j} = \begin{cases} 1, & \text{if primary shelf } s \text{ on mother plate } j \text{ uses width } w_i \\ 0, & \text{otherwise} \end{cases}$$

$$u^{l,p}_{i,s,j} = \begin{cases} 1, & \text{if primary shelf } s \text{ on mother plate } j \text{ uses length } l_i \\ 0, & \text{otherwise} \end{cases}$$

$$u^{w}_{i,s2,s,j} = \begin{cases} 1, & \text{if secondary shelf } s2 \text{ in primary shelf } s \text{ on mother plate } j \text{ uses width } w_i \\ 0, & \text{otherwise} \end{cases}$$

$$u^{l}_{i,s2,s,j} = \begin{cases} 1, & \text{if secondary shelf } s2 \text{ in primary shelf } s \text{ on mother plate } j \text{ uses length } l_i \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha^{L}_{is2sjd} = \begin{cases} 1, & \text{if order plate } i \text{ is to the left of defect } d \\ & \text{in secondary shelf } s2 \text{ of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha^{R}_{is2sjd} = \begin{cases} 1, & \text{if order plate } i \text{ is to the right of defect } d \\ & \text{in secondary shelf } s2 \text{ of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha^{U}_{is2sjd} = \begin{cases} 1, & \text{if order plate } i \text{ is above defect } d \\ & \text{in secondary shelf } s2 \text{ of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha^{D}_{is2sjd} = \begin{cases} 1, & \text{if order plate } i \text{ is below defect } d \\ & \text{in secondary shelf } s2 \text{ of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

$$z^{\text{ord}}_{ii's2sj} = \begin{cases} 1, & \text{if order plate } i \text{ precedes order plate } i' \\ & \text{in secondary shelf } s2 \text{ of primary shelf } s \text{ on mother plate } j \\ 0, & \text{otherwise} \end{cases}$$

Objectives:

$$\max \Pi = \underbrace{\sum_{j\in J}\sum_{s\in S}\sum_{s2\in S_2}\sum_{i\in I} p_i\, a_i\, y_{is2sj}}_{\text{Sales}} - \underbrace{\sum_{j\in J} c_{\text{master}} A_j}_{\text{Mother plate cost}} + \underbrace{\sum_{j\in J} c_{\text{recycle}}\left(A_j - \sum_{s\in S}\sum_{s2\in S_2}\sum_{i\in I} a_i\, y_{is2sj}\right)}_{\text{Scrap}}. \tag{1}$$

$$\min\ TC_{\text{total}} = \sum_{j\in J}\sum_{s\in S:\, s\geq 2} \delta_{s-1,s,j}. \tag{2}$$

Constraints:

$$w^R_i = l_i, \quad l^R_i = w_i, \qquad \forall i \in I \tag{3}$$

$$W^{\text{eff}}_{isj} = (1 - r_{isj}) w_i + r_{isj} w^R_i, \qquad \forall i, s, j \tag{4}$$

$$L^{\text{eff}}_{isj} = (1 - r_{isj}) l_i + r_{isj} l^R_i, \qquad \forall i, s, j \tag{5}$$

$$w_{sj} = W_j\, x_{sj}, \qquad \forall s, j \tag{6}$$

$$\sum_{s\in S} l_{sj}\, x_{sj} \ \leq\ L_j, \qquad \forall j \tag{7}$$

$$q_{sj} + l_{sj} \ \leq\ L_j + M(1 - x_{sj}), \qquad \forall s, j \tag{8}$$

$$q_{s'j} \ \geq\ q_{sj} + l_{sj} - M\,(2 - x_{sj} - x_{s'j}), \qquad \forall j,\ \forall s' \neq s \tag{9}$$

$$l_{sj} = \sum_{i\in I}\left(w_i\, u^{w,p}_{i,s,j} + l_i\, u^{l,p}_{i,s,j}\right), \qquad \forall s, j \tag{10}$$

$$\sum_{i\in I}\left(u^{w,p}_{i,s,j} + u^{l,p}_{i,s,j}\right) = x_{sj}, \qquad u^{w,p}_{i,s,j}, u^{l,p}_{i,s,j} \in \{0,1\} \tag{11}$$

$$l_{s2sj} = l_{sj}\, x_{s2sj}, \qquad \forall s2, s, j \tag{12}$$

$$s_{s2sj} \geq s_{sj} - M(1 - x_{s2sj}), \quad s_{s2sj} + w_{s2sj} \leq s_{sj} + w_{sj} + M(1 - x_{s2sj}), \qquad \forall s2, s, j \tag{13}$$

$$s_{s2+1,sj} \geq s_{s2sj} + w_{s2sj} - M\,(2 - x_{s2sj} - x_{s2+1,sj}), \qquad \forall s2,\ s, j \tag{14}$$

$$\sum_{s2\in S_2} w_{s2sj} \ \leq\ w_{sj}, \qquad \forall s, j \tag{15}$$

$$\sum_{s2\in S_2} x_{s2sj} = k_{sj}, \qquad \forall s, j \tag{16}$$

$$w_{s2sj} = \sum_{i\in I}\left(w_i\, u^{w}_{i,s2,s,j} + l_i\, u^{l}_{i,s2,s,j}\right), \qquad \forall s2, s, j \tag{17}$$

$$\sum_{i\in I}\left(u^{w}_{i,s2,s,j} + u^{l}_{i,s2,s,j}\right) = x_{s2sj}, \quad u^{w}_{i,s2,s,j}, u^{l}_{i,s2,s,j} \in \{0,1\} \tag{18}$$

$$y_{is2sj} \leq x_{s2sj}, \qquad \forall i, s2, s, j \tag{19}$$

$$W^{\text{eff}}_{isj} \leq w_{s2sj} + M(1 - y_{is2sj}), \quad L^{\text{eff}}_{isj} \leq l_{s2sj} + M(1 - y_{is2sj}), \qquad \forall i, s2, s, j \tag{20}$$

$$\sum_{j\in J}\sum_{s\in S}\sum_{s2\in S_2} y_{is2sj} \le 1, \qquad \forall i \in I \tag{21}$$

$$pos_{is2sj} \ge q_{sj} - M(1 - y_{is2sj}), \qquad \forall i, s2, s, j \tag{22}$$

$$pos_{is2sj} + L^{\text{eff}}_{isj} \le q_{sj} + l_{sj} + M(1 - y_{is2sj}), \qquad \forall i, s2, s, j \tag{23}$$

$$pos_{i's2sj} \ge pos_{is2sj} + L^{\text{eff}}_{isj} - M\,(1 - z^{\text{ord}}_{ii's2sj}), \qquad \forall i \ne i', s2, s, j \tag{24}$$

$$pos_{is2sj} \ge pos_{i's2sj} + L^{\text{eff}}_{i'sj} - M\,z^{\text{ord}}_{ii's2sj}, \qquad \forall i \ne i', s2, s, j \tag{25}$$

$$\alpha^{L}_{is2sjd} + \alpha^{R}_{is2sjd} + \alpha^{U}_{is2sjd} + \alpha^{D}_{is2sjd} \;\ge\; (1 - \text{acc}_{id})\, y_{is2sj}, \qquad \forall i, s2, s, j, d \tag{26}$$

$$\text{(Left)}\;\; s_{s2sj} + w_{s2sj} \le w^{s}_{dj} + M\,(1 - \alpha^{L}_{is2sjd}), \qquad \forall i, s2, s, j, d \tag{27}$$

$$\text{(Right)}\; s_{s2sj} \ge w^{e}_{dj} - M\,(1 - \alpha^{R}_{is2sjd}), \qquad \forall i, s2, s, j, d \tag{28}$$

$$\text{(Down)}\; pos_{is2sj} + L^{\text{eff}}_{isj} \le l^{s}_{dj} + M\,(1 - \alpha^{D}_{is2sjd}), \qquad \forall i, s2, s, j, d \tag{29}$$

$$\text{(Up)}\quad pos_{is2sj} \ge l^{e}_{dj} - M\,(1 - \alpha^{U}_{is2sjd}), \qquad \forall i, s2, s, j, d \tag{30}$$

$$\Delta k_{s,j} = k_{sj} - k_{s-1,j}, \qquad \forall j,\; s \ge 2 \tag{31}$$

$$C_{s,j} = \sum_{s2\in S_2} \left(s_{s2sj}\, x_{s2sj} - s_{s2,s-1,j}\, x_{s2,s-1,j}\right), \qquad \forall j,\; s \ge 2 \tag{32}$$

$$\Delta k_{s,j} \le \Delta k^{+}_{s,j},\;\; -\Delta k_{s,j} \le \Delta k^{-}_{s,j},\;\; C_{s,j} \le C^{+}_{s,j},\;\; -C_{s,j} \le C^{-}_{s,j}, \qquad \forall j,\; s \ge 2 \tag{33}$$

$$\Delta k^{+}_{s,j}, \Delta k^{-}_{s,j}, C^{+}_{s,j}, C^{-}_{s,j} \ge 0, \qquad \forall j,\; s \ge 2 \tag{34}$$

$$(\Delta k^{+}_{s,j} + \Delta k^{-}_{s,j}) + (C^{+}_{s,j} + C^{-}_{s,j}) \;\ge \epsilon\; z_{s,j}, \qquad \forall j,\; s \ge 2 \tag{35}$$

$$(\Delta k^{+}_{s,j} + \Delta k^{-}_{s,j}) + (C^{+}_{s,j} + C^{-}_{s,j}) \;\le\; M\,(1 - z_{s,j}), \qquad \forall j,\; s \ge 2 \tag{36}$$

$$\delta_{s-1,s,j} = z_{s,j}, \qquad \forall j,\; s \ge 2 \tag{37}$$

The model adopts a bi-objective optimization framework that balances economic performance and production practicality. The primary objective (1) maximizes total profit Π by combining sales revenue from placed order plates, mother-plate costs, and scrap recycling revenue. This formulation provides a comprehensive economic evaluation. The secondary objective (2) minimizes total tool changes TC_{total} between adjacent primary shelves, thereby reducing setup times and improving continuity. These objectives capture real-world trade-offs between profitability and efficiency and can be reconciled using standard multi-objective techniques (e.g., weighted sum, lexicographic, or ϵ-constraint methods) as required.

Table 1: Parameters and sets.

Symbol	Description (Unit)
M	Big-M constant (sufficiently large, e.g., 10^6 mm)
ϵ	Small threshold for non-zero judgment (e.g., 0.1 mm)
p_i	Unit selling price of order plate i (CNY/mm^2)
c_{master}	Mother-plate cost per unit area (CNY/mm^2)
c_{recycle}	Scrap recycling price per unit area (CNY/mm^2)
I	Set of order plates
$I_R \subseteq I$	Set of rotatable order plates
J	Set of mother plates
S	Index set of primary (level-1) shelves for each $j \in J$
S_2	Index set of secondary (level-2) shelves for each (s, j)

(Continued)

Table 1 (continued)

Symbol	Description (Unit)
D	Set of defects on mother plates
w_i	Width of order plate i (mm)
l_i	Length of order plate i (mm)
w_i^R	Rotated width of order plate i (mm)
l_i^R	Rotated length of order plate i (mm)
W_j	Width of mother plate j (mm)
L_j	Length of mother plate j (mm)
$a_i = w_i l_i$	Area of order plate i (mm^2)
$A_j = W_j L_j$	Area of mother plate j (mm^2)
w_{dj}^s	Start x-coordinate of defect d on mother plate j (mm)
w_{dj}^e	End x-coordinate of defect d on mother plate j (mm)
l_{dj}^s	Start y-coordinate of defect d on mother plate j (mm)
l_{dj}^e	End y-coordinate of defect d on mother plate j (mm)
$\mathrm{acc}_{id} \in \{0,1\}$	1 if order plate i accepts defect d; 0 otherwise

Table 2: Auxiliary variables.

Variable	Meaning (Domain)
W_{isj}^{eff}	Effective width of order plate i considering rotation r_{isj} ($\mathbb{R}_+$)
L_{isj}^{eff}	Effective length of order plate i considering rotation r_{isj} ($\mathbb{R}_+$)
$\Delta k_{s,j}$	Difference in secondary shelf count between primary shelves $s-1$ and s on j ($\mathbb{Z}$)
$C_{s,j}$	Cumulative x-position difference of secondary shelves between $s-1$ and s on j ($\mathbb{R}$)
$\Delta k_{s,j}^+$	Positive part of $\Delta k_{s,j}$ for absolute value linearization ($\mathbb{R}_+$)
$\Delta k_{s,j}^-$	Negative part of $\Delta k_{s,j}$ for absolute value linearization ($\mathbb{R}_+$)
$C_{s,j}^+$	Positive part of $C_{s,j}$ for absolute value linearization ($\mathbb{R}_+$)
$C_{s,j}^-$	Negative part of $C_{s,j}$ for absolute value linearization ($\mathbb{R}_+$)

Table 3: Continuous decision variables.

Variable	Meaning (Domain)
w_{sj}	Width of primary shelf s on mother plate j ($\mathbb{R}_+$)
l_{sj}	Length of primary shelf s on mother plate j ($\mathbb{R}_+$)
s_{sj}	x-axis start position of primary shelf s on mother plate j ($\mathbb{R}_+$)
q_{sj}	y-axis start position of primary shelf s on mother plate j ($\mathbb{R}_+$)
w_{s2sj}	Width of secondary shelf $s2$ in primary shelf s of mother plate j ($\mathbb{R}_+$)
l_{s2sj}	Length of secondary shelf $s2$ in primary shelf s of mother plate j ($\mathbb{R}_+$)
s_{s2sj}	x-axis start position of secondary shelf $s2$ in primary shelf s of mother plate j ($\mathbb{R}_+$)
pos_{is2sj}	y-axis start position of order plate i in secondary shelf $(s2, s)$ of mother plate j ($\mathbb{R}_+$)
k_{sj}	Number of secondary shelves in primary shelf s of mother plate j ($\mathbb{Z}_+$)

Constraints (3)–(5) handle optional 90° plate rotations and compute effective dimensions accordingly.

Constraint (3) swaps width and length upon rotation ($w_i^R = l_i$, $l_i^R = w_i$, $\forall i \in I$).

Constraints (4) and (5) compute effective width and length as convex combinations controlled by the binary rotation indicator r_{isj}.

Constraints (6)–(9) enforce the first-stage guillotine cuts that form primary shelves.

Constraint (6) forces active primary shelves to span the full mother-plate width.

Constraint (7) limits their total length to the mother-plate length.

Constraints (8) and (9) arrange primary shelves sequentially along the y-axis without overlap using Big-M formulations.

Constraints (10) and (11) restrict primary-shelf lengths to valid order-plate dimensions $\{w_i, l_i\}$, ensuring downstream feasibility for secondary cuts.

Constraints (12)–(16) govern the second-stage guillotine cuts that create secondary shelves.

Constraint (12) aligns secondary-shelf length with its primary shelf.

Constraints (13) and (14) position secondary shelves sequentially along the x-axis without overlap.

Constraint (15) bounds total secondary-shelf widths by the primary-shelf width.

Constraint (16) counts secondary shelves for tool-change evaluation.

Constraints (17) and (18) set secondary-shelf width to the sum of (possibly rotated) dimensions of assigned plates, consistent with second-stage longitudinal guillotine cuts that permit side-by-side placement. The binary indicators $u^w_{i,s2,s,j}$ and $u^l_{i,s2,s,j}$ select exactly one dimension per plate.

Constraints (19)–(21) enforce placement logic: plates can occupy only active secondary shelves (19), must fit dimensionally (20), and are assigned at most once (21).

Constraints (22)–(25) control plate positioning and non-overlap along the y-axis within secondary shelves.

Constraints (22) and (23) confine plates to the shelf's vertical bounds.

Constraints (24) and (25) prevent overlaps via disjunctive sequencing enforced by $z^{\text{ord}}_{ii's2sj}$.

Constraints (26)–(30) enforce defect avoidance for non-tolerant plates.

Constraint (26) activates separation logic only when required.

Constraints (27)–(30) require the plate to lie entirely left, right, above, or below the defect using auxiliary binaries $\alpha^L, \alpha^R, \alpha^U, \alpha^D$.

Constraints (31)–(37) detect tool changes from pattern differences between adjacent primary shelves. Constraint (31) computes the secondary-shelf count difference, while (32) sums starting x-position differences.

Constraints (33) and (34) linearize absolute differences using positive/negative parts.

Constraints (35)–(37) activate the change indicator $\delta_{s-1,s,j}$ when the absolute sum exceeds threshold ϵ.

Collectively, these constraints ensure that the resulting cutting plan is feasible, implementable, defect-compliant, and consistent with real manufacturing rules. They also reflect sustainability considerations by promoting fine-grained control of scrap generation, enabling defect-aware utilization, and minimizing unnecessary tool changes to reduce operational waste.

4 Improved Multi-Objective Grey Wolf Optimization

This section presents the proposed IMOGWO, designed specifically to handle the discrete, defect-aware, and multi-constraint characteristics of the DMMP-2CSP. Although the original MOGWO offers a solid foundation for multi-objective search, its continuous search behavior, susceptibility to stagnation, and limited mechanisms for maintaining population diversity make it inadequate for complex industrial layout optimization. To address these limitations, IMOGWO incorporates a real-valued random-keys encoding with hierarchical decoding to translate continuous vectors into feasible defect-aware cutting layouts, integrates a Large-Language-Model–assisted fourth-leader boost mechanism to detect stagnation and inject guided exploration when necessary, and blends in NSGA-II principles, including non-dominated sorting, crowding-distance preservation, and genetic variation, to strengthen convergence stability and broaden Pareto-front coverage. Collectively, these enhancements yield a robust hybrid multi-objective metaheuristic well suited to sustainable steel-plate cutting, enabling improved defect-aware utilization, reduced scrap generation, and more efficient tool usage.

4.1 Encoding and Decoding Strategy

To apply the continuous search mechanism of MOGWO to the discrete DMMP-2CSP, we propose a real-valued encoding scheme coupled with a hierarchical decoding procedure.

Each individual is represented as a $2 \times n$ real-valued matrix, where n is the number of order plates. The first row encodes processing priorities (real numbers), and the second row encodes rotation probabilities in $[0, 1]$. During decoding:

- The processing sequence is obtained by sorting the priorities in ascending order.
- Rotation flags are binarized by rounding the corresponding probabilities (values ≥ 0.5 indicate 90° rotation).

An example is shown in Fig. 4.

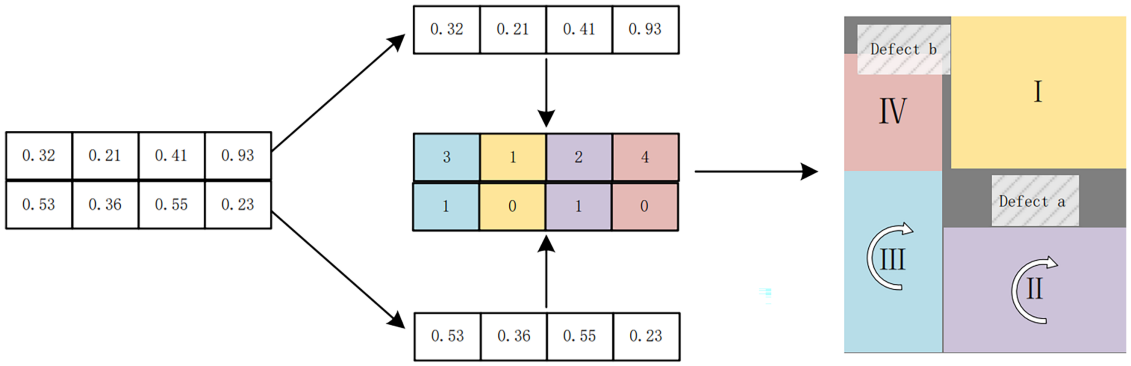


Figure 4: Schematic of the encoding and decoding process.

The decoding proceeds as follows. Order plates are processed sequentially according to the decoded order. Primary shelves are formed by cumulatively adding the effective length (length after rotation) of each plate until exceeding the mother-plate length would occur; the last added plate finalizes the current primary shelf height. The reference plate defining each primary shelf is placed therein.

Remaining plates are then placed using a best-fit strategy with bottom-left preference. For each plate, existing secondary shelves in all primary shelves are scanned. If a feasible position exists that satisfies geometric constraints and defect-avoidance rules (using the four-direction separation logic), the plate is

inserted; otherwise, a new secondary shelf is created from left to right in the earliest compatible primary shelf. Plates that cannot be placed in the current mother plate are postponed and reattempted on subsequent plates.

This encoding-decoding strategy effectively bridges the continuous search space of MOGWO. It serves as the essential link between upstream manufacturing and downstream customer order fulfillment. However, the combination of highly diverse specifications, multi-size demand patterns, and small-batch orders leads to substantial complexity in planning and layout design, with the discrete, highly constrained layout requirements of the DMMP-2CSP, consistently producing valid cutting plans.

The MINLP formulated in Section 3 provides a complete mathematical description of the DMMP-2CSP, but its size—comprising hundreds of variables and constraints reflecting spatial, rotational, and defect-avoidance requirements—makes direct solution intractable for industrial-scale instances. Instead, the MINLP serves as a formal feasibility benchmark. The IMOGWO algorithm operates in a continuous search space, and the hierarchical decoder described above acts as the bridge to the discrete feasible region: it translates a continuous-encoded candidate into a concrete cutting layout by sequentially placing plates while strictly enforcing all MINLP constraints. In this way, the decoder guarantees that every layout generated during the search is feasible, allowing the heuristic to explore only valid regions of the solution space.

4.2 LLM-Assisted Fourth-Leader Boost Mechanism

To mitigate premature convergence and maintain population diversity, the proposed IMOGWO introduces an adaptive fourth-leader boost mechanism. Whenever the search process becomes stagnant, an additional LLM-generated leader is injected to provide a new search direction. Stagnation is identified using a composite criterion that monitors both the size and a hash-based signature of the Pareto archive across consecutive iterations. As detailed in Algorithm 1, the variable `stagnation_count` tracks how many iterations the archive has remained unchanged, while `boost_timer` controls the duration of the boosted search phase. The Boolean flag `use_fourth` is activated whenever this boost phase is underway. If stagnation persists for at least P iterations and no boost is currently active, a new boost period of length B is initiated. The timer is then decremented each iteration until it expires. This mechanism enables the optimizer to trigger intensified exploration precisely when needed, thereby enhancing robustness and preserving global search diversity.

Algorithm 1: StagnationDetection—detect stagnation and activate LLM fourth leader

function StagnationDetection($\mathcal{A}^{t-1}$, $\mathcal{A}^{t}$, `stagnation_count`, `boost_timer`, P, B)
  Compute signature $\mathrm{Sig}(\mathcal{A}^{t})$
  Compute signature $\mathrm{Sig}(\mathcal{A}^{t-1})$
  if $|\mathcal{A}^{t}| = |\mathcal{A}^{t-1}|$ **and** $\mathrm{Sig}(\mathcal{A}^{t}) = \mathrm{Sig}(\mathcal{A}^{t-1})$ **then**
    `stagnation_count` ← `stagnation_count`+1
  else
    `stagnation_count` ← 0
  end if
  if `stagnation_count` ≥ P **and** `boost_timer` = 0 **then**
    `boost_timer` ← B
    `stagnation_count` ← 0
  end if
  `use_fourth` ← (`boost_timer` > 0)

(Continued)

Algorithm 1 (continued)

```
    if boost_timer > 0 then
        boost_timer ← boost_timer − 1
    end if
    return use_fourth, stagnation_count, boost_timer
end function
```

Once activated, the boost phase augments the conventional three-leader update rule by incorporating a fourth leader $\tilde{L}$, as defined in Algorithm 2. The auxiliary leader is generated on demand by a locally deployed *gpt-oss-120b* model (OpenAI, 120B parameters) running on a dedicated high-performance server equipped with six Intel RTX 5880 Ada GPUs, 120 GB RAM, and 2 TB SSD storage under Ubuntu 24.04.

Domain-specific knowledge is injected through the AnythingLLM framework, which integrates the theoretical foundations of multi-objective grey wolf optimization together with the DMMP-2CSP formulation presented in Section 3. This grounding enables the LLM to synthesize leader candidates that satisfy geometric feasibility, defect-avoidance constraints, and the bi-objective nature of the problem.

Whenever a boost is triggered, the LLM receives structured contextual information, including the current mother-plate inventory, defect maps, order-plate specifications, and the elite Pareto set, and produces a continuous random-keys chromosome representing the fourth leader. The generative behavior follows the prompt design illustrated in Fig. 5.

Structured Prompt Template for LLM-Guided Fourth-Leader Generation

System Instruction:
Act as a domain-informed auxiliary-leader generator embedded in the IMOGWO optimization loop. Your objective is to synthesize a feasible fourth leader that enhances search-space diversity and assists in escaping stagnation. All generated solutions must comply with the DMMP-2CSP formulation and the bi-objective profit–knife-change criteria.

Context Specification (Inputs):

- Number of order plates: {{n}}
- Mother-plate size: {{W}} × {{L}}
- Defect-map descriptors: {{defect_features}}
- Current leaders and fitness values: (L_1, L_2, L_3) with {{fitness_values}}

Generation Requirements: Produce a continuous random-keys matrix $\tilde{L} \in [0,1]^{2\times n}$ satisfying:

1. Preservation of geometric feasibility and defect-avoidance constraints.
2. Increased population diversity and improved exploration capability.
3. Compliance with the two-row encoding scheme – Row 1: sequencing priorities; – Row 2: rotation probabilities.
4. Output formatted as a Python-style list-of-lists of floats.

Expected Output Schema:

```
{
  "fourth_leader_matrix": [
      [0.12 0.85 0.33, ...],   # sequence row
      [0.50 0.00 0.25, ...]    # rotation row
  ],
  "comment": "Row 1: ordering; Row 2: rotation."
}
```

Figure 5: Structured prompt template used for LLM-guided fourth-leader generation.

Algorithm 2: FourthLeaderUpdate—position update with LLM-guided fourth leader

function FourthLeaderUpdate($X, \mathcal{A}^{t-1}, \mathcal{A}^{t}, a, sc, bt, P, B, n$)
  $(uf,\ sc,\ bt) \leftarrow$ StagnationDetection($\mathcal{A}^{t-1}, \mathcal{A}^{t}, sc, bt, P, B$)
  $N \leftarrow |X|$, $d \leftarrow$ dimension of X
  Initialize Xn as an $N \times d$ zero matrix
  for i = 1 to N **do**
    if $|\mathcal{A}^{t}| \geq 3$ **then**
      Select leaders $L_1, L_2, L_3 \in \mathcal{A}^{t}$
    else
      Randomly sample L_1, L_2, L_3 from X
    end if
    if uf **then**
      Query LLM $\rightarrow \tilde{L}$ ▷ $\tilde{L}$ should be $2 \times n$
      for row = 1 to 2 **do**
        $len \leftarrow$ length of $\tilde{L}[row]$
        if $len < n$ **then**
          Append *(n-len)* random values sampled from $U(0,1)$
        else if $len > n$ **then**
          Truncate $\tilde{L}[row]$ to first n elements
        end if
      end for
      $\mathcal{L} \leftarrow \{L_1, L_2, L_3, \tilde{L}\}$
      $au \leftarrow \min(2.5,\ 1.15a)$
    else
      $\mathcal{L} \leftarrow \{L_1, L_2, L_3\}$
      $au \leftarrow a$
    end if
    $w \leftarrow 1/|\mathcal{L}|$
    for j = 1 to d **do**
      $s \leftarrow 0$
      for each $L \in \mathcal{L}$ **do**
        Sample $r_1, r_2 \sim U(0,1)$
        $A \leftarrow 2aur_1 - au$
        $C \leftarrow 2r_2$
        $D \leftarrow |CL_j - X_{i,j}|$
        $s \leftarrow s + (L_j - AD)$
      end for
      $Xn_{i,j} \leftarrow w \cdot s$
    end for
  end for
  return $Xn,\ sc,\ bt$
end function

Algorithm 2 outlines the complete LLM-guided position update. To keep the pseudocode concise, variable names are shortened: sc tracks stagnation, bt records the remaining boost duration, and uf

indicates whether the boost phase is active. When uf = True, the algorithm queries the LLM to obtain $\tilde{L}$ and increases the exploration coefficient to au. Because the LLM may occasionally exhibit hallucinated behavior, the generated random-keys vector is post-processed to ensure the correct $2 \times n$ structure before being used in the update step. Updated positions are stored in Xn, while the accumulator s aggregates leader contributions dimension-wise. The routine first invokes the stagnation-detection module to refresh (sc, bt, uf), then performs position updates using either three or four leaders depending on the boost status. This mechanism allows IMOGWO to intensify exploration precisely when stagnation occurs while retaining computational efficiency.

4.3 Integration of NSGA-II Mechanisms

To enhance population diversity, maintain convergence stability, and balance exploration and exploitation within the Improved Multi-Objective Grey Wolf Optimizer (IMOGWO), core components of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) are incorporated. This hybrid strategy, referred to as *NSGA-II Fusion*, integrates non-dominated sorting, crowding-distance preservation, and genetic variation operators to address the limitations of classical GWO, such as uneven Pareto-front coverage and susceptibility to premature convergence in multi-objective environments.

During the iterative search process, wolves in standard MOGWO tend to cluster around the current non-dominated set, gradually reducing population diversity and limiting exploration of underrepresented regions. While GWO excels at global search via leader-driven position updates, it lacks structured selection pressure and variation operators. In contrast, NSGA-II Fusion introduces deterministic elitism and adaptive genetic operators, enabling IMOGWO to maintain a favorable trade-off between global diversification and local refinement.

Let the population at iteration t be

$$\mathcal{P}^t = \{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_N^{(t)}\}, \tag{38}$$

where N is the population size and each solution is represented as

$$\mathbf{x}_i^{(t)} = \left[x_{i,1}^{(t)}, x_{i,2}^{(t)}, \ldots, x_{i,d}^{(t)}\right], \tag{39}$$

denoting the i-th wolf in a d-dimensional random-keys search space.

Non-dominated sorting. A solution $\mathbf{x}_a$ is said to dominate $\mathbf{x}_b$ (denoted $\mathbf{x}_a \succ \mathbf{x}_b$) if

$$\begin{aligned} &f_m(\mathbf{x}_a) \le f_m(\mathbf{x}_b), \quad \forall m, \\ &\exists m : \ f_m(\mathbf{x}_a) < f_m(\mathbf{x}_b), \end{aligned} \tag{40}$$

where $f_m(\cdot)$ denotes the m-th objective (profit maximization and knife-change minimization). Solutions are grouped into hierarchical fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$, with $\mathcal{F}_1$ containing the non-dominated individuals.

Crowding distance. To preserve diversity within each front $\mathcal{F}$, the crowding distance for solution i is computed as

$$CD_i = \sum_{m=1}^{M} \frac{f_m(\mathbf{x}_{i+1}) - f_m(\mathbf{x}_{i-1})}{f_m^{\max} - f_m^{\min}}, \tag{41}$$

where $f_m^{\max}$ and $f_m^{\min}$ denote, respectively, the maximum and minimum objective values within the same front. Boundary solutions along each objective dimension are assigned $CD_i = \infty$ to ensure retention of extreme points.

Genetic variation operators. Two widely adopted NSGA-II operators are incorporated:

- **Simulated Binary Crossover (SBX).** For parents $\mathbf{x}_p$ and $\mathbf{x}_q$, offspring are generated according to

$$x_{c,j} = \begin{cases} \frac{1}{2}[(1+\beta)x_{p,j} + (1-\beta)x_{q,j}], & r < 0.5, \\ \frac{1}{2}[(1-\beta)x_{p,j} + (1+\beta)x_{q,j}], & \text{otherwise}, \end{cases} \tag{42}$$

where $r \sim U(0,1)$ and the spread factor β is computed as

$$\beta = \begin{cases} (2r)^{1/(\eta_c+1)}, & r \le 0.5, \\ [2(1-r)]^{-1/(\eta_c+1)}, & r > 0.5. \end{cases}$$

- **Polynomial mutation.** For solution $\mathbf{x}_i$, mutation modifies each variable as

$$x'_{i,j} = x_{i,j} + \delta_j(U_j - L_j), \tag{43}$$

where L_j and U_j are the lower and upper bounds, and

$$\delta_j = \begin{cases} (2r)^{1/(\eta_m+1)} - 1, & r < 0.5, \\ 1 - [2(1-r)]^{1/(\eta_m+1)}, & \text{otherwise}. \end{cases}$$

Population merging and selection.

In each iteration, the parent solutions, the GWO-derived offspring and the NSGA-II offspring are merged and selected according to the model.

$$\mathcal{P}^{t+1} = \text{Select}\left(\mathcal{P}^t \cup \mathcal{O}_{\text{GWO}} \cup \mathcal{O}_{\text{NSGA-II}}, N\right), \tag{44}$$

where the operator Select$(\cdot)$ performs non-dominated sorting followed by crowding-distance-based truncation, yielding a diverse and high-quality next-generation population.

Algorithm 3 summarizes the NSGA-II Fusion operator, which is invoked as a subroutine within the IMOGWO framework. The procedure takes the current population X and elite archive A as inputs, while the parameters η_c, η_m, p_c, and p_m govern the SBX crossover and polynomial mutation operators in Eqs. (42) and (43). A temporary set M collects GWO offspring, NSGA-II offspring, and parent solutions. Non-dominated sorting (Eq. (40)) and crowding-distance evaluation (Eq. (41)) are then applied to M, and the Select$(\cdot)$ operator (Eq. (44)) extracts the top N individuals to form the next-generation population. This module provides IMOGWO with structured elitism and diversity preservation while complementing GWO-driven exploration.

Algorithm 3: NSGA2Fusion—NSGA-II fusion operator for IMOGWO

```
function NSGA2Fusion(X, A, N, η_c, η_m, p_c, p_m)
    M ← ∅                                          ▷ Merged temporary set
    for i = 1 to N do
        Select leaders α, β, δ from A
        x_g ← GWO_Update(X_i, α, β, δ)
        Evaluate fitness of x_g
```

(Continued)

Algorithm 3 (continued)

Add x_g to M
end for
for $i = 1$ to N **do**
Select parents p,q via tournament selection
$x_c \leftarrow \mathrm{SBX}(p, q, \eta_c, p_c)$ ▷ Simulated Binary Crossover, Eq. (42)
$x_c \leftarrow \mathrm{Mutate}(x_c, \eta_m, p_m)$ ▷ Polynomial mutation, Eq. (43)
Evaluate fitness of x_c
Add x_c to M
end for
for $i = 1$ to N **do**
Add X_i to M
end for
Perform non-dominated sorting using dominance rule ▷ Eq. (40)
Compute crowding distance for all individuals ▷ Eq. (41)
$Xn \leftarrow \mathrm{Select}(M, N)$ ▷ Population selection, Eq. (44)
Clip all decision variables of Xn into $[0, 1]$
return Xn
end function

4.4 Overall Framework of IMOGWO

The complete IMOGWO framework is outlined in Algorithm 4. In each iteration, the algorithm first updates solution positions based on the grey-wolf leadership hierarchy, with the LLM-generated fourth leader introduced adaptively when stagnation is detected. This is followed by the integration of NSGA-II genetic operators and elitist selection to maintain population diversity and preserve high-quality non-dominated solutions. All candidate individuals are subsequently decoded into feasible cutting layouts and evaluated under the defect-aware multi-objective formulation. The external archive is then incrementally updated and pruned to retain only the most representative and well-distributed Pareto solutions.

Through this iterative process, IMOGWO achieves smooth and reliable transitions between continuous search operations and discrete layout feasibility, enhances exploration precisely when the search becomes stagnant, and ensures stable convergence through NSGA-II's elitism and crowding-distance mechanisms. By combining these components within a unified multi-objective search framework, IMOGWO consistently generates high-quality, defect-tolerant cutting plans that improve material utilization, reduce tool-change frequency, and contribute directly to more sustainable steel-plate manufacturing.

Algorithm 4: Overall framework of the proposed IMOGWO (with LLM Boost and NSGA-II Fusion)

Require: Population size N, maximum iterations $T_{\max}$, archive capacity $A_{\max}$, stagnation threshold P, boost duration B, crossover index η_c, mutation index η_m, crossover probability p_c, mutation probability p_m
Ensure: Final Pareto-optimal archive $\mathcal{A}$
Initialize population $X = \{X_1, \ldots, X_N\}$ using random-keys encoding
Decode and evaluate X
Initialize elite archive $\mathcal{A}$ via non-dominated sorting
$A_{\mathrm{prev}} \leftarrow \mathcal{A}$
$sc \leftarrow 0$, $bt \leftarrow 0$ ▷ stagnation counter & boost timer

(Continued)

Algorithm 4 (continued)

for $t = 1$ to $T_{\max}$ **do**
  $a \leftarrow 2 - 2t/T_{\max}$
  $(X_{\text{gwo}}, sc, bt) \leftarrow \text{FourthLeaderUpdate}(X, A_{\text{prev}}, \mathcal{A}, a, sc, bt, P, B)$ ▷ Algorithm 2
  Decode and evaluate X_{gwo}
  $X_{\text{nsga}} \leftarrow \text{NSGA2Fusion}(X, \mathcal{A}, N, \eta_c, \eta_m, p_c, p_m)$ ▷ Algorithm 3
  Decode and evaluate X_{nsga}
  $M \leftarrow X \cup X_{\text{gwo}} \cup X_{\text{nsga}}$
  Perform non-dominated sorting on M
  Compute crowding distance
  $X \leftarrow \text{Select}(M, N)$
  Clip all decision variables of X into $[0, 1]$
  $\mathcal{A} \leftarrow \mathcal{A} \cup X$
  Perform non-dominated sorting on $\mathcal{A}$
  Keep only first-front solutions in $\mathcal{A}$
  if $|\mathcal{A}| > A_{\max}$ **then**
    Sort $\mathcal{A}$ by descending crowding distance
    Truncate to $A_{\max}$ solutions
  end if
  $A_{\text{prev}} \leftarrow \mathcal{A}$
end for
return $\mathcal{A}$

5 Experiments and Analysis

5.1 Test Instances

To validate the efficiency of the proposed IMOGWO algorithm, a set of simulation experiments was constructed based on real-world steel plate cutting production parameters. The following sections outline the relevant parameter settings, including the number of order plates, dimensions of the mother plates, the number of defects on the plates, and rotation constraints for the order plates. All dimensions are provided in millimeters.

- Master Plate Dimensions and Inventory Quantity Generation: The dimensions of the master plates are generated within predefined ranges for defective inventory plates. Specifically, the length is randomly selected from the range [3000, 25,000], and the width is chosen from [600, 3000]. To better simulate real-world production scenarios, a specified number of data sets are generated. Each data set represents the dimensions of a specific inventory master plate, and the corresponding inventory quantity for each set is randomly selected from the integer range [1, 10].
- Master Plate Defect Generation: Each master plate is independently assigned defects. A random number of defects, ranging from 0 to 6, is generated for each master plate. Each defect is treated as a rectangular region, with its dimensions randomly selected from the range [100, 1000] for both length and width. There are six distinct types of defects, and defects may overlap on the same master plate. This defect generation process ensures variability and realism in simulating the potential imperfections encountered in steel plates during actual production.
- Order Plate Dimension Generation: The length of the order plates is randomly selected within the range [100, 1000], and the width is chosen from the range [200, 1000]. For each unique combination of length

and width, between 70 and 200 order plates are generated. This rule ensures a diverse set of order plates with varying dimensions, reflecting the complexity of real-world production requirements.

- Order Plate Defect Acceptance Generation: For each group of order plates, a uniform defect acceptance range is generated. The defect acceptance range is represented by a one-dimensional array of length 6, where each element (ranging from 0 to 6) corresponds to a defect type that the order plates in that group can tolerate. For example, if the acceptance array for a group of order plates is 1, 2, 0, 0, 0, 0, it indicates that the group can accept defect types 1 and 2. If the array is 1, 2, 0, 0, 0, 6, it means the group can accept defects 1, 2, and 6. If the array is 0, 0, 0, 0, 0, 0, it signifies that the group of order plates cannot accept any defects.

These datasets follow industrial parameter ranges but are synthetically generated for controlled evaluation. To intuitively present experimental data under the above parameter rules, Tables 4–6 provide concrete examples of master plate specifications, defect configurations, and order plate attributes, respectively, establishing a validated foundation for IMOGWO algorithm testing.

Table 4: Master plate specifications.

Master Plate ID	Length (mm)	Width (mm)	No. of Defects
1	12,000	1500	2
2	12,000	1500	3
3	18,000	2000	2
4	18,000	2000	4
5	15,000	2500	3
6	15,000	2500	2

Table 5: Defect details by master plate.

Master Plate ID	Defect Types	Defect Position (x,y)
1	1	(1000, 300)
1	3	(5000, 800)
2	2	(2500, 500)
2	4	(7000, 200)
2	6	(9500, 1100)
3	1	(3000, 400)
3	5	(12,000, 1500)
4	2	(5000, 800)
4	3	(9000, 1200)
4	4	(14,000, 600)
4	6	(16,000, 1800)
5	1	(2000, 600)
5	2	(6000, 1300)
5	5	(11,000, 1900)
6	3	(4500, 900)
6	4	(8500, 2200)

Table 6: Example data for order plates.

Order Plate ID	Length (mm)	Width (mm)	Accepted Defects	Quantity per Group
1	900	800	[1, 2, 0, 0, 0, 0]	80
2	800	1000	[2, 3, 0, 0, 0, 0]	120
3	700	600	[1, 4, 5, 0, 0, 0]	150
4	1000	900	[3, 0, 0, 0, 0, 6]	90
5	600	750	[1, 2, 3, 4, 0, 0]	180

To provide a clear illustration of the generated datasets, Tables 4–6 present concrete examples covering master-plate specifications, defect characteristics, and order-plate groups. These tables translate the abstract generation rules into representative instances, validating the correctness and industrial relevance of the evaluation data.

Together, these three tables translate abstract parameter rules into concrete, validated datasets, providing standardized and realistic inputs for subsequent IMOGWO performance evaluation and sensitivity analysis.

5.2 Parameter Settings for Metaheuristic Algorithms

This paper employs a multi-tiered comparison framework to rigorously evaluate the proposed IMOGWO algorithm. The experiments encompass five distinct categories of algorithms: (i) three classical multi-objective optimizers as baseline references; (ii) the original MOGWO algorithm; (iii) an ablation variant integrating only the LLM-guided Fourth-Leader Boost (MOGWO + LLM); (iv) an ablation variant integrating only the NSGA-II reproduction operator (MOGWO + NSGA-II); and (v) the full IMOGWO algorithm proposed herein. The selected comparison algorithms are listed as follows:

- *Multi-Objective Particle Swarm Optimization* (MOPSO) [38]: A swarm-intelligence-based algorithm that extends particle swarm optimization to multi-objective domains, maintaining an external archive of non-dominated solutions and leveraging leader selection to explore the Pareto front effectively.
- *Multi-Objective Cuckoo Search* (MOCS) [39]: A nature-inspired algorithm based on the cuckoo search meta-heuristic, adapted for multi-objective problems by integrating Lévy-flight moves, non-dominated sorting and elitism, thereby efficiently exploring complex Pareto fronts in multi-objective optimization tasks.
- *Multi-Objective Snake Optimizer* (MOSO) [40]: A multi-objective adaptation of the Snake Optimizer meta-heuristic, enhanced with non-dominated sorting, elitism preservation, and an external archive mechanism to maintain diversity and convergence along the Pareto front, effectively solving complex bi-objective and multi-objective optimization problems.

These algorithms are selected due to their strong representativeness and proven effectiveness in handling continuous and combinatorial multi-objective optimization problems. Together, they provide a comprehensive baseline for evaluating the convergence accuracy, diversity, and robustness of the proposed IMOGWO algorithm.

The key parameter settings for each algorithm were as follows. Wherever applicable, default parameters recommended in the respective original publications were adopted to ensure fair and consistent comparisons:

- *MOPSO:* The inertia weight was set to 0.5, while the cognitive and social learning factors (c_1 and c_2) were both set to 2. The external archive size was limited to 50 solutions, and the leader selection mechanism was based on crowding-distance sorting.
- *MOCS:* The discovery rate of alien eggs (p_a) was set to 0.25, and the Lévy-flight step size (α) to 1. A non-dominated sorting strategy and external archive of size 100 were employed to maintain Pareto diversity. Other control parameters followed the standard configuration proposed in the original MOCS publication.
- *MOSO:* The population size was set to 50, while the food quantity constant (c_1) and movement step constant (c_2) were set to 0.5 and 0.05, respectively. The temperature threshold was fixed at 0.6 to control the transition between exploration and exploitation.
- *MOGWO:* The population size and maximum iteration count were the same as those in IMOGWO. The exploration coefficient a was reduced linearly from 2 to 0, and no additional mechanisms such as LLM guidance or other enhancement strategies were applied.
- *MOGWO + LLM:* This variant retained the core MOGWO framework but integrated the LLM-guided Fourth-Leader Boost mechanism. The stagnation detection threshold P was set to 15, and the boost duration B to 5 iterations, consistent with the IMOGWO configuration.
- *MOGWO + NSGA-II:* This hybrid retained the MOGWO framework while integrating the NSGA-II reproduction operator (SBX crossover and polynomial mutation) to enhance diversity and Pareto-front exploration. Key parameters were set as: number of offspring N_c = `pop_size`, SBX crossover probability $p_c = 0.9$, polynomial mutation probability $p_m = 1/L$, with distribution indices $\eta_c = 15.0$ and $\eta_m = 20.0$.
- *IMOGWO:* The population size was set to 50, and the maximum number of iterations to 200. The stagnation detection threshold P was set to 15, and the boost duration B to 5 iterations. The exploration coefficient a linearly decreased from 2 to 0 over the entire iteration process. All other parameters followed the baseline MOGWO configuration.

To ensure fairness in benchmarking, all algorithms were configured with a uniform population size of 50 and a maximum of 200 iterations. The experiments were implemented in *Python 3.9.23* and executed using the *PyCharm 2023.2.5* integrated development environment (JetBrains Inc., Prague, Czech Republic). All computational tests were conducted on a workstation running *Windows 11* (Microsoft Corporation, Redmond, Washington, USA), equipped with a 12th Gen *Intel® Core™ i7-12700H* processor (Intel Corporation, Santa Clara, California, USA; base frequency 2.30 GHz) and 16 GB of RAM.

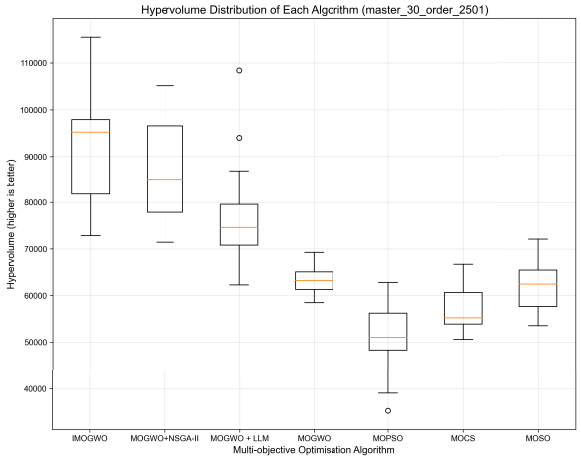
Each algorithm was run independently for 20 trials to account for stochastic behavior. The elite solution set obtained from each trial was recorded for statistical performance evaluation across all benchmark instances.

5.3 Computational Results and Discussion

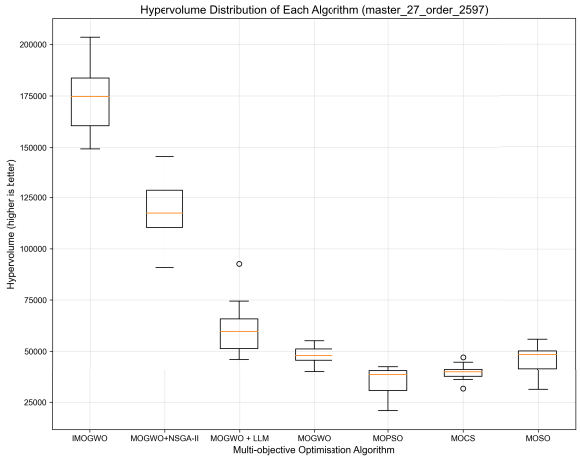
To evaluate the performance of the proposed IMOGWO algorithm, three datasets with varying scales were tested: Dataset 1 (30 master plates, 2501 order plates), Dataset 2 (27 master plates, 2597 order plates), and Dataset 3 (24 master plates, 2671 order plates). Hypervolume was adopted as the primary metric, quantifying both convergence and diversity of the Pareto front (larger values indicate better performance). Statistical significance was assessed using pairwise t-tests with $\alpha = 0.05$. Each algorithm was run independently 20 times (with minor exceptions due to runtime anomalies), and results are presented as mean hypervolume ± standard error, with the number of valid runs in parentheses.

To visually characterize the hypervolume variability across 20 independent runs, boxplots were constructed for each dataset, as shown in Fig. 6. The figure includes three subplots: (a) for Dataset 1 (30 master

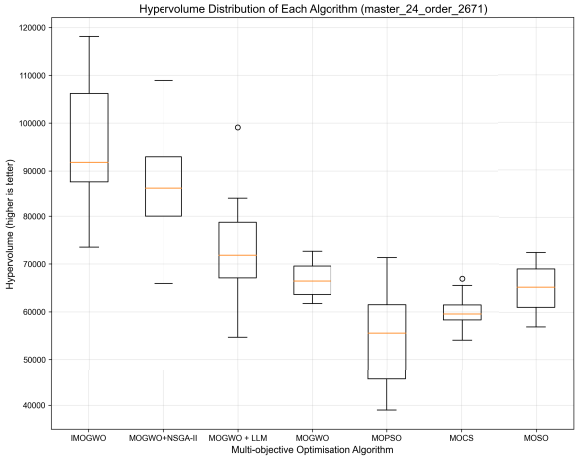
plates, 2501 order plates), (b) for Dataset 2 (27 master plates, 2597 order plates), and (c) for Dataset 3 (24 master plates, 2671 order plates). In each subplot, algorithms are ordered from left to right as follows: Improved Multi-Objective Grey Wolf Optimizer (IMOGWO), MOGWO with LLM-assisted Fourth-Leader Boost Mechanism (MOGWO + LLM), original MOGWO, MOPSO, and MOCS. Boxplots follow standard conventions: the box represents the interquartile range (IQR, 25th to 75th percentiles), the horizontal line within the box denotes the median, whiskers extend to 1.5 × IQR to indicate data spread, and individual points mark outliers beyond this range.



(a) Boxplots of Hypervolume for Dataset 1's Pareto Frontier Solution Set



(b) Boxplots of Hypervolume for Dataset 2's Pareto Frontier Solution Set



(c) Boxplots of Hypervolume for Dataset 3's Pareto Frontier Solution Set

Figure 6: Boxplots of hypervolume values for all algorithms across three datasets.

It is worth noting that benchmarking multi-objective metaheuristics often involves comparisons with exact or decomposition-based methods on small to medium-sized instances where near-optimal solutions can be obtained, as exemplified in prior studies such as the Flower Pollination Algorithm applied to electrochemical machining optimization [41]. However, the DMMP-2CSP is a strongly NP-hard bi-objective problem with complex geometric and defect constraints. Even for moderately sized instances, obtaining the

exact Pareto front via commercial solvers is computationally prohibitive. Therefore, in line with common practice in large-scale industrial optimization, we focus on evaluating IMOGWO against state-of-the-art metaheuristics on three realistically scaled datasets: Dataset 1 (30 master plates, 2501 order plates), Dataset 2 (27 master plates, 2597 order plates), and Dataset 3 (24 master plates, 2671 order plates).

Boxplot analysis across these datasets consistently confirms the superior performance of IMOGWO. In each case, IMOGWO achieves the highest median hypervolume, with its entire distribution (box and whiskers) positioned distinctly above all comparative algorithms—including MOGWO + NSGA-II, MOGWO + LLM, MOPSO, MOSO, and MOCS. Notably, no extreme outliers are observed for IMOGWO, indicating reliable operational stability. Its IQR, while slightly wider than those of some classical algorithms (e.g., MOGWO and MOCS), remains compact relative to its elevated performance, especially in the more complex configurations (Datasets 2 and 3). In contrast, other algorithms either cluster in lower hypervolume regions or exhibit greater variability, underscoring their limitations in handling the intricate constraints of defect-aware cutting.

Key experimental findings, detailed in Tables 7 and 8, reinforce these visual trends and firmly establish IMOGWO's superiority. The algorithm not only attains the highest mean hypervolume in every scenario but does so by substantial and statistically significant margins, highlighting its practical advantages.

Table 7: Hypervolume performance of algorithms across datasets (mean ± standard error, with number of runs in parentheses).

Algorithm	30 Master Plates	27 Master Plates	24 Master Plates
MOSO	62,080.48 ± 1156.0081	46,394.73 ± 1377.15	65,093.76 ± 1021.4820
MOCS	56,688.54 ± 978.13	39,480.34 ± 752.84	59,805.74 ± 766.98
MOPSO	51,318.07 ± 1556.02	35,487.44 ± 1436.50	54,420.06 ± 2105.25
MOGWO	63,472.85 ± 700.81	47,787.10 ± 898.48	66,486.13 ± 743.25
MOGWO + LLM	76,939.39 ± 2322.70	60,098.41 ± 2450.43	72,742.52 ± 2329.13
MOGWO + NSGA-II	86,793.68 ± 1156.0081	118,707.05 ± 2936.21	85,929.43 ± 2227.99
IMOGWO	92,787.94 ± 2572.83	173,455.65 ± 3402.05	95,256.28 ± 2777.18

Table 8: t-test statistics comparing IMOGWO with other algorithms (p-values < 0.0001 indicate statistically significant superiority).

Comparison	30 Master Plates	27 Master Plates	24 Master Plates
IMOGWO vs. MOSO	$t = 10.61$, $p = 0.0000$	$t = 33.74$, $p = 0.0000$	$t = 9.93$, $p = 0.0000$
IMOGWO vs. MOCS	$t = 12.78$, $p = 0.0000$	$t = 37.48$, $p = 0.0000$	$t = 11.99$, $p = 0.0000$
IMOGWO vs. MOPSO	$t = 13.43$, $p = 0.0000$	$t = 36.41$, $p = 0.0000$	$t = 11.42$, $p = 0.0000$
IMOGWO vs. MOGWO	$t = 10.72$, $p = 0.0000$	$t = 34.81$, $p = 0.0000$	$t = 9.75$, $p = 0.0000$
IMOGWO vs. MOGWO + LLM	$t = 4.46$, $p = 0.0001$	$t = 26.35$, $p = 0.0000$	$t = 6.05$, $p = 0.0000$
IMOGWO vs. MOGWO + NSGA-II	$t = 1.69$, $p = 0.0988$	$t = 11.87$, $p = 0.0000$	$t = 2.55$, $p = 0.0150$

While benchmarking multi-objective metaheuristics often involves comparisons with exact methods on small instances, the DMMP-2CSP is a strongly NP-hard problem where obtaining the exact Pareto front is computationally prohibitive, even for moderately-sized cases. Therefore, consistent with practices in large-scale industrial optimization, we evaluate IMOGWO against state-of-the-art metaheuristics on three realistically scaled datasets, which vary in structural complexity: Dataset 1 (30 master plates) represents the

largest inventory scale, Dataset 2 (27 master plates) presents a balanced configuration, and Dataset 3 (24 master plates) embodies the highest order density.

Across all datasets, IMOGWO demonstrates superior and robust performance. On the largest inventory (30 plates), IMOGWO's hypervolume leads by 20.6% over 'MOGWO + LLM' and 46.2% over baseline 'MOGWO'. Under the highest combinatorial complexity (24 plates, 2671 orders), it maintains a decisive advantage, outperforming the same variants by 31.0% and 43.3%, respectively. The balanced 27-plate dataset further confirms this scalability, where IMOGWO's performance surpasses 'MOGWO + LLM' and 'MOGWO' by factors of 2.9 and 3.6. Statistical validation solidly supports these results, with pairwise t-tests yielding $p < 0.0001$ for nearly all comparisons (e.g., $t = 11.99$ vs. MOCS on the 24-plate set).

This consistent superiority is attributed to IMOGWO's synergistic hybrid design. The LLM-guided boost effectively mitigates stagnation, as evidenced by the gains of the 'MOGWO + LLM' variant. However, IMOGWO's substantial additional improvements highlight the indispensable role of the NSGA-II fusion module, which ensures a well-distributed, high-quality Pareto front through non-dominated sorting and crowding-distance selection. This integrated architecture enables IMOGWO to navigate the intricate geometric and defect-aware constraints of real-world cutting problems more effectively than all baseline and ablated counterparts.

To objectively evaluate the LLM-guided leader's efficacy beyond empirical hypervolume gains, we compare its generated solution set against a purely random generator using the coverage metric $C(A, B)$. As summarized in Table 9, the LLM-guided set consistently dominates a large majority of random solutions, with $C(A, B)$ values of 0.95, 0.90, and 0.75 for the 30-, 27-, and 24-master-plate datasets, respectively. The observed decline on the most order-dense dataset (24 plates) is attributed to practical constraints: when processing exceptionally long solution arrays, the LLM occasionally encounters token limits or generation artifacts, prompting the algorithm to supplement incomplete outputs with random segments—thus diluting the measured coverage. Nevertheless, even under these conditions, the LLM-guided approach maintains a substantial advantage over undirected random search.

Table 9: Coverage (C) metric for LLM vs. random solution generators across datasets.

C index	30 Master Plates	27 Master Plates	24 Master Plates
C(A,B)	0.95	0.90	0.75

Fundamentally, these results confirm that the LLM provides directed, knowledge-informed exploration by leveraging embedded domain understanding of defect avoidance and cutting constraints, in contrast to the undirected exploration of a random strategy. The coverage metric thereby isolates and validates the intrinsic guidance capability of the LLM component.

Beyond solution quality, the practical applicability of IMOGWO in industrial settings hinges on its balanced computational efficiency. The runtime overhead introduced by the LLM component is well controlled: as shown in Fig. 7, the full IMOGWO framework incurs only a modest increase in execution time (approximately 2.3% on average) compared to its non-LLM variant, MOGWO + NSGA-II. This marginal increase is due to the on-demand activation of the LLM-guided boost mechanism, which is triggered only upon stagnation detection—typically 2–4 times per 200 iterations. In return, this lightweight overhead yields substantial gains in hypervolume and Pareto-front quality, making the trade-off highly favorable for time-sensitive production environments.

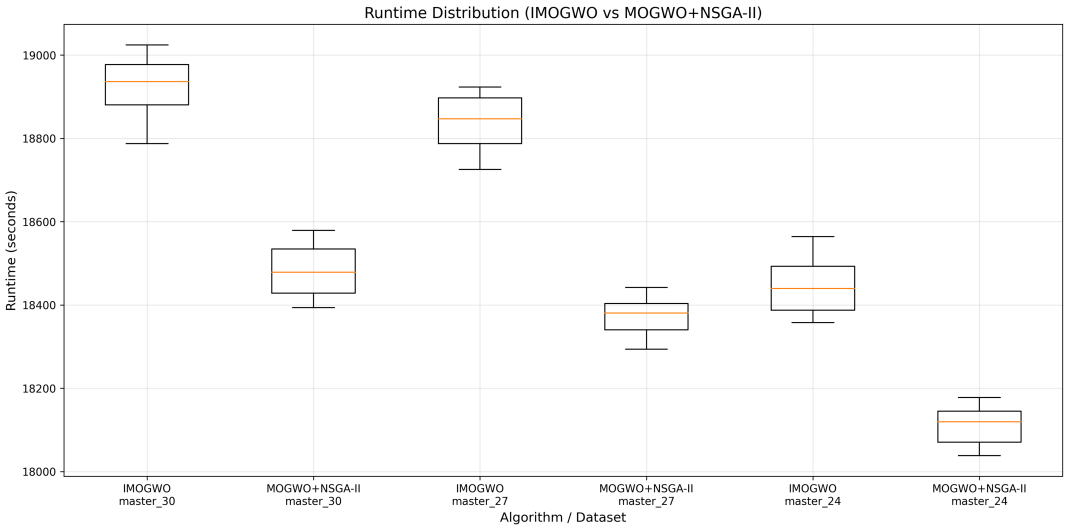


Figure 7: Enter caption.

The runtime of IMOGWO scales predictably with the number of master plates, as evidenced by the consistent distributions in Fig. 7. Defect density and archive size have limited impact, since defects are processed as geometric constraints within fixed plate areas and the archive is maintained via efficient non-dominated sorting. Moreover, the problem formulation naturally bounds the solution space: the total order-plate area must exceed the total master-plate area to ensure high utilization. To minimize the introduction of unreliable variables during comparative experiments, we adopted a single-core execution mode; in practical applications, however, multi-core parallel computing for fitness evaluation will significantly accelerate the algorithm's speed. Thus, the practical runtime remains within a manageable upper limit, ensuring reliable performance in real-world industrial batches.

6 Conclusions and Future Outlook

6.1 Research Conclusions

This study develops an Improved Multi-Objective Grey Wolf Optimizer (IMOGWO) for the Defective Multi-Inventory Mother-Plate Cutting Stock Problem. IMOGWO synergistically integrates an LLM-guided stagnation detection mechanism with NSGA-II's elitism operators, effectively balancing global exploration and local exploitation. This hybrid design directly addresses key limitations of conventional metaheuristics—specifically premature convergence and poor Pareto-front diversity—while introducing only a minimal and controlled runtime overhead.

The algorithm's effectiveness is concretely demonstrated across both optimization objectives. For the primary goal of maximizing cutting profit, IMOGWO consistently yields higher-quality layouts, translating to a significant improvement in material utilization compared to all baseline algorithms. For the secondary goal of minimizing tool changes, IMOGWO's solution sets include layouts that require substantially fewer tool switches, directly reducing production delays and operational costs. The quantitative superiority is validated by the significantly higher hypervolume achieved across all tested datasets (e.g., 2.89 times that of 'MOGWO + LLM' on the 27-plate set) and statistically confirmed by rigorous t-tests ($p < 0.0001$).

Beyond performance, IMOGWO demonstrates strong practical viability. Its runtime scales predictably with the number of master plates, and the LLM component is invoked sparingly (2–4 times per 200

iterations), ensuring that substantial solution quality gains come with a manageable and justifiable computational cost.

6.2 Limitations and Future Outlook

While demonstrating strong performance, the current work has limitations that chart a course for future research. Firstly, the evaluation is based on synthetic datasets with rectangular defects; validation on plates with irregular or continuous defect patterns would strengthen the method's industrial generality. Secondly, the problem model assumes static parameters; incorporating dynamic elements like real-time defect feedback or variable energy costs would enhance real-world fidelity.

A specific limitation concerns the reproducibility and practicality of the LLM component. The current implementation relies on a locally deployed large-scale model to handle the long-context prompts required for direct solution-array generation. This approach, while effective, imposes hardware dependencies and raises accessibility barriers. Substituting with an online model would introduce network latency, further impacting runtime. To overcome this, a promising direction is to evolve from using the LLM as a direct solution generator to leveraging it as a meta-heuristic designer. Future work will explore using LLMs to automatically generate or adapt heuristic rules and search strategies—creating an "LLM-as-a-meta-heuristic" framework—which could drastically reduce token consumption, lower hardware barriers, and enhance both reproducibility and adaptability for broader industrial deployment.

Finally, transitioning from simulation to physical pilot studies in collaboration with manufacturing partners will be essential to bridge the gap between algorithmic innovation and operational impact, solidifying the role of such hybrid AI methods in sustainable smart manufacturing.

Acknowledgement: None.

Funding Statement: This work was supported by the Liaoning Province Education Department Scientific Research Foundation of China under Grant No. JYTQN2023366, and the Doctoral Startup Project of Liaoning Provincial Department of Science and Technology under Grant No. 2025-BS-0433.

Author Contributions: Conceptualization, Qi Zhang, Shujin Qin, Xiwang Guo, Bin Hu and Changtian Zhang; Data collection, Wenjie Luo; Formal analysis, Changtian Zhang; Writing—original draft preparation, Changtian Zhang; Proofreading, Bin Hu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in this article and are available from the corresponding authors upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang Y, Zheng Z, Guo L, Yang Y, Zhang S, Liu X, et al. Advancements in production planning and scheduling within steel manufacturing: a review and its intelligent development. Int J Miner Metall Mater. 2025;32(10):2322–40. doi:10.1007/s12613-025-3188-5.
2. Peng G, Zhang B, Jiang S. A two-stage solution method for the design problem of medium-thick plates in steel plants. IET Collab Intell Manuf. 2024;6(1):e12091. doi:10.1049/cim2.12091.
3. Bahrami A, Kiani Khouzani M, Mokhtari SA, Zareh S, Yazdan Mehr M. Root cause analysis of surface cracks in heavy steel plates during the hot rolling process. Metals. 2019;9(7):801. doi:10.3390/met9070801.

4. Zhao S, Meng Y, Su L, Liu J, Tang L. Optimal plate design problem in steel production. Int J Prod Res. 2022;61(5):1575–90. doi:10.1080/00207543.2022.2043566.
5. Xu J, Yang W. Multi-objective steel plate cutting optimization problem based on real number coding genetic algorithm. Sci Rep. 2022;12(1):22472. doi:10.1038/s41598-022-27100-2.
6. Beasley JE. An exact two-dimensional non-guillotine cutting tree search procedure. Oper Res. 1985;33(1):49–64. doi:10.1287/opre.33.1.49.
7. Côté J-F, Haouari M, Iori M. Combinatorial benders decomposition for the two-dimensional bin packing problem. INFORMS J Comput. 2021;33(3):963–78. doi:10.1287/ijoc.2020.1014.
8. Furini F, Malaguti E. Models for the two-dimensional two-stage cutting stock problem with multiple stock sizes. Comput Oper Res. 2013;40(8):1953–62. doi:10.1016/j.cor.2013.02.026.
9. Silva FE, Alvelos F, Valério de Carvalho JM. An integer programming model for two- and three-stage two-dimensional cutting stock problems. Eur J Oper Res. 2010;205(3):699–708. doi:10.1016/j.ejor.2010.01.039.
10. Gardeyn J, Wauters T. A goal-driven ruin and recreate heuristic for the 2D variable-sized bin packing problem with guillotine constraints. Eur J Oper Res. 2022;301(1):432–44. doi:10.1016/j.ejor.2021.11.031.
11. Zhou SC, Li XP, Zhang KK, Du N. Two-dimensional knapsack-block packing problem. Appl Math Model. 2019;73(3):1–18. doi:10.1016/j.apm.2019.03.039.
12. Wäscher G, Haußner H, Schumann H. An improved typology of cutting and packing problems. Eur J Oper Res. 2007;183(3):1109–30. doi:10.1016/j.ejor.2005.12.047.
13. Scheithauer G. Introduction to cutting and packing optimization problems: modeling approaches, solution methods. Cham, Switzerland: Springer; 2018 [cited 2026 Jan 5]. Available from: https://link.springer.com/book/10.1007/978-3-319-64403-5.
14. Luo Q, Rao YQ, Du B. An effective discrete artificial bee colony for the rectangular cutting problem with guillotine in transformer manufacturing. Appl Soft Comput. 2024;159:111617. doi:10.1016/j.asoc.2024.111617.
15. Zhang H, Yao S, Liu Q, Leng J, Wei L. Exact approaches for the unconstrained two-dimensional cutting problem with defects. Comput Oper Res. 2023;160(4):106407. doi:10.1016/j.cor.2023.106407.
16. Gilmore PC, Gomory RE. Multistage cutting stock problems of two and more dimensions. Oper Res. 1965;13(1):94–120. doi:10.1287/opre.13.1.94.
17. Hahn S. On the optimal cutting of defective sheets. Oper Res. 1968;16(6):1100–14. doi:10.1287/opre.16.6.1100.
18. Scheithauer G, Terno J. Guillotine cutting of defective boards. Optimization. 1988;19(1):111–21. doi:10.1080/02331938808843323.
19. Iori M, Lima V, Martello S, Monaci M. Exact solution techniques for two-dimensional cutting and packing. Eur J Oper Res. 2021;289(2):399–415. doi:10.1016/j.ejor.2020.06.050.
20. Neidlein V, Vianna A, Arenales M, Morabito R. Two-dimensional guillotineable-layout cutting problems with a single defect: an and/or-graph approach. In: Operations Research Proceedings 2008. Cham, Switzerland: Springer; 2009. p. 85–90. doi:10.1007/978-3-642-00142-0_14.
21. Afsharian M, Niknejad A, Wäscher G. A heuristic dynamic programming based approach for a two-dimensional cutting problem with defects. OR Spectr. 2014;36(4):971–99. doi:10.1007/s00291-014-0363-x.
22. Martin M, Hokama P, Morabito R, Munari P. The constrained two-dimensional guillotine cutting problem with defects: an ILP formulation, a Benders decomposition and a CP-based algorithm. Int J Prod Res. 2020;58(9):2712–29. doi:10.1080/00207543.2019.1630773.
23. Martin M, Morabito R, Munari P. Two-stage and one-group two-dimensional guillotine cutting problems with defects: a CP-based algorithm and ILP formulations. Int J Prod Res. 2021;60(12):3642–61. doi:10.1080/00207543.2021.1876270.
24. Yao S, Zhang H, Liu Q, Leng J, Wei L. Combinatorial Benders' decomposition for the constrained two-dimensional non-guillotine cutting problem with defects. Int J Prod Res. 2024;62:8299–325. doi:10.1080/00207543.2024.2338194.
25. Yao S, Zhang H, Wei L, Liu Q. An exact approach for the two-dimensional strip packing problem with defects. Comput Ind Eng. 2025;200:110866. doi:10.1016/j.cie.2025.110866.
26. Zhang Q, Xing Y, Zhang C, Sun X, Hu B, Das A. Column generation algorithms for two-dimensional cutting problem with surface defects. Int J Artif Intell Green Manuf. 2025;1(2):23–35. doi:10.58837/chula.the.2018.1588.

27. Zhang Z, Dai S, Li C, Wang W, Parron J, Herrera E. Human–robot collaborative disassembly profit maximization via improved grey wolf optimizer. Int J Artif Intell Green Manuf. 2025;1(2):12–22.
28. Mirjalili S, Saremi S, Mirjalili SM, dos S Coelho L. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl. 2016;47:106–19. doi:10.1016/j.engappai.2016.10.013.
29. Lu C, Gao L, Li X, Xiao S. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. Eng Appl Artif Intell. 2017;57:61–79. doi:10.1016/j.engappai.2016.10.013.
30. Qin H, Fan P, Tang H, Huang P, Fang B, Pan S. An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint. Comput Ind Eng. 2018;128(2):458–76. doi:10.1016/j.cie.2018.12.061.
31. Abdelmaguid TF. Bi-objective dynamic multiprocessor open shop scheduling for maintenance and healthcare diagnostics. Expert Syst Appl. 2021;186(3):115777. doi:10.1016/j.eswa.2021.115777.
32. Seifhosseini S, Hosseini Shirvani M, Ramzanpoor Y. Multi-objective cost-aware bag-of-tasks scheduling optimization model for IoT applications running on heterogeneous fog environment. Comput Netw. 2024;240:110161. doi:10.1016/j.comnet.2023.110161.
33. Ghosh S, Acherjee B, Kuar AS. Application of grey wolf optimizer algorithm in electrochemical machining process optimization. In: Industrial automation and robotics. 1st ed. Boca Raton, FL, USA: CRC Press; 2022. p. 45–60. doi:10.1201/9781003121640-5.
34. Guo XW, Guo F, Qi L, Wang J, Liu SX, Qin S, et al. Modeling and optimization of multi–product human–robot collaborative hybrid disassembly line balancing with resource sharing. IEEE Trans Comput Soc Syst. 2025;12(1):1–14. doi:10.1109/TCSS.2025.3540070.
35. Zhou L, Guo X, Liu Q, Wang J, Qin S, Qi L. Multifactory remanufacturing process optimization considering worker scheduling. IEEE Trans Comput Soc Syst. 2025;12(1):43–56. doi:10.1109/TCSS.2025.3570435.
36. Guo XW, Chen L, Qi L, Wang J, Qin S, Chatterjee M, et al. Multi-factory disassembly process optimization considering worker posture. IEEE Trans Comput Soc Syst. 2025;12(1):15–28. doi:10.1109/TCSS.2025.3540565.
37. Zhou L, Zhu H, Akbari B. Multi-objective optimization of multi-factory remanufacturing process considering worker fatigue. Int J Artif Intell Green Manuf. 2025;1(2):36–50.
38. Coello Coello CA, Lechuga MS. MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02); 2002 May 12–17; Honolulu, HI, USA. p. 1051–6. doi:10.1109/CEC.2002.1004388.
39. Syberfeldt A. Multi-objective optimization of a real-world manufacturing process using cuckoo search. In: Yang XS, editor. Cuckoo search and firefly algorithm: theory and applications. Cham, Switzerland: Springer; 2014. p. 179–93. doi:10.1007/978-3-319-02141-6_9.
40. Li L, Song H. Research on electric vehicle charging station location based on multiple objective snake optimizer. In: Proceedings of the 2023 International Conference on Management Science and Engineering Management. Cham, Switzerland: Atlantis Press; 2023. p. 1419–27. doi:10.2991/978-94-6463-256-9_143.
41. Acherjee B, Maity D, Kuar AS. Optimization of correlated and conflicting responses of ECM process using flower pollination algorithm. Int J Appl Metaheur Comput. 2020;11(4):1–15. doi:10.4018/ijamc.2020100101.