ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250512002657
Original scientific paper

Optimizing Two-Dimensional Cutting Algorithms for the TFT-LCD Industry using Multi-Objective Strategies

Yuan XU, Jin FANG, Guihua JIANG, Hong YE, Ke WANG, Jiaming CHANG, Leilei BO, Jingxuan WANG, Jiyang ZHU*

Abstract: Thin-film transistor (TFT) displays, celebrated for their high resolution and color accuracy, dominate the current display market. However, during the Thin-film transistor liquid crystal display (TFT-LCD) panel cutting process, reliance on traditional batch production methods persists among manufacturers. This approach results in several drawbacks, including excessive consumption of glass substrates, extended cutting paths, and increasing costs due to inefficient utilization of residual materials. To address these challenges, this study draws on concepts from machine learning to introduce a multi-objective adaptive (MOA) cutting stock algorithm. The algorithm employs a state selection matrix to optimize the sequence of blank selection at each cutting step and incorporates a blank cutting criterion to maximize the value of leftover materials. Experimental results indicate that the algorithm reduced production costs by approximately 15% on real-world data from a TFT-LCD factory.

Keywords: cutting cost; cutting stock problem; machine learning; material utilization rate; residual material value; state selection matrix

1 INTRODUCTION

Nowadays, thin-film transistor liquid crystal display (TFT-LCD) panels have become essential components in commercial display devices [1]. They achieve a significant leap in image quality while maintaining low cost and exceptional reliability [2]. The advancement of TFT-LCD manufacturing technology is pivotal for fostering the development of modern smart factories on a global scale. Glass substrates, serving as the primary raw materials for LCD production, are consumed in substantial quantities during the manufacturing process. Consequently, inefficiencies and material waste during the cutting phase of LCD panels significantly escalate production costs. To mitigate these issues, high-efficiency and precision cutting strategies are typically employed to enable large-scale, intelligent batch manufacturing [3]. However, many manufacturers continue to rely on traditional batch processing methods, where each glass substrate yields only one-panel size. This approach leads to excessive substrate consumption, prolonged cutting paths and non-recyclable waste, all of which contribute to heightened production costs.

This process can be effectively modeled as a two-dimensional rectangular cutting stock problem (RTD-CSP), which involves the optimal allocation of fixed-size raw sheets to produce a set of blanks that fulfill specific dimension and quantity requirements [4]. The RTD-CSP is a critical area of study within operations research, industrial engineering, and smart manufacturing, attracting substantial academic and practical attention. A stochastic multi-start algorithm was designed for core production in the transformer industry [5]. To minimize printing costs in the publishing sector, a nonlinear integer model and an efficient genetic algorithm were proposed by Mahdi [6]. Erkan [7] introduced a simulated annealing metaheuristic that integrates first-fit decreasing and increasing techniques to address cutting stock challenges in the wood manufacturing industry. Additionally, a beam search algorithm offers new insights for addressing the two-dimensional cutting and packing problems in the sheet metal industry [8].

The RTD-CSP is a classic non-deterministic polynomial (NP)-hard problem [9] characterized by high computational complexity, which makes it infeasible to find globally optimal solutions within reasonable timeframes. Only approximate local optima can realistically be achieved. Nevertheless, developing high-quality nesting layouts can markedly reduce material waste and improve utilization, resulting in significant economic benefits, which has attracted lots of attention. A two-stage homogeneous block nesting algorithm was introduced by Ji Jun [10], which generates rectangular cutting plans using implicit enumeration for optimal block selection and linear programming to minimize both sheet usage and computation time. To optimize layouts by maximizing net value (total blank value minus cutting costs) and enhance material efficiency while reducing operational expenses, Chen Yan [11] proposed a contour-parallel cutting strategy. Mixed-integer programming models were developed by Andrade [12] to conceptualize the residual usable area, aiming to maximize its value in two-phase cutting processes. Additionally, a decomposition-based heuristic was designed by Ayasandır [13] for multi-material scenarios, balancing solution quality and computational effort, although increased material diversity can lead to higher logistical costs. Lastly, Jin Xiaobo [14] improved traditional genetic algorithms by incorporating a "gene pool" operator and heuristic chromosome generation through block-based greedy methods, which enhances solution robustness but faces challenges with large-scale product diversity in practical applications.

In addressing such NP-hard problems, the concepts of machine learning are frequently applied. To resolve traffic scheduling issues in autonomous vehicle networks, an algorithm combining machine learning techniques with time- sensitive networking has been proposed [15]. A model integrating machine learning and greedy optimization has been introduced to tackle workflow tasks with unknown execution times [16]. A heuristic deep reinforcement learning approach offers a new perspective for optimizing reconfigurable intelligent surface (RIS)-assisted wireless networks, featuring faster convergence speeds [17]. Additionally, Niu Yun proposed a multi-objective optimization algorithm based on a decision tree classifier, utilizing an efficient location-routing strategy to balance total cost, carbon emissions, and the distance between residential areas and waste processing centers [18]. However, this approach has not yet been widely adopted in this field. Generally, using machine learning to solve NP-hard combinatorial problems faces two major challenges: firstly, the inherently discrete nature of combinatorial problems makes it difficult to map discrete outcomes with machine learning methods. Secondly, due to the large solution space, generating labels for supervised machine learning requires substantial computational effort, which can be nearly impossible [19].

Machine learning also has some applications in the field of two-dimensional cutting. In order to solve the cutting problem in the construction industry, reduce material waste and budget costs, a self-adjusting symbiotic biological search algorithm (SOS) is proposed to establish the residue optimization model [20]. Mariusz Kaleta [21] uses a simple feed-forward neural network population to solve the two-dimensional weakly homogeneous packing problem (2D-BPP) in the context of robotic packing by using the covariance matrix adaptive evolution strategy for black box optimization training.

Another significant challenge in practical production is managing production complexity, particularly concerning the cutting path. Optimizing the cutting path of rectangular parts shortens the total cutting path and reduces cutting cost, which has significant practical value for the production and processing of rectangular parts [22]. Moreover, the residual materials left after incomplete sheet utilization often exhibit varying shapes and sizes, resulting in resource wastage and raising environmental concerns [23].

To address these challenges, this paper presents a multi-objective adaptive (MOA) cutting stock algorithm, which enhances traditional genetic [24] and simulated annealing frameworks by integrating a state selection matrix inspired by machine learning model training. The matrix framework overcomes the issue of discrete solutions through training on the priority parameters for each blank selection. By directly using the profile utilization from each iteration's generated solution set as the label for subsequent training, it avoids extensive computations for label generation. During the iterative process, continuous model training within the matrix optimizes parameters to ensure that each round of solution generation and selection is optimal, thereby minimizing cutting losses. Additionally, the paper introduces a new blank cutting criterion designed to increase the value of residual materials. Experimental validation confirms the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows: Chapter 2 defines the RTD-CSP problem. Chapter 3 elaborates on the state selection matrix. Chapter 4 conducts a residual material analysis and introduces the blank cutting rules, enhancing the MOA algorithm. Chapter 5 presents the evaluation of experimental results, and Chapter 6 concludes the study.

2 PROBLEM DEFINITION

In this study, we address the problem faced by an enterprise possessing an unlimited supply of rectangular glass substrates with fixed dimensions, specifically length L and width W. The task at hand involves cutting $l_i \times w_i (i = 1,2,3,\dots M)$ types of blanks, each characterized by dimensions $d_i (i = 1,2,3,\dots M)$ and a specified demand N. The solution to this problem is represented by a final nesting scheme composed of ff sub-schemes. Our primary objective is to determine an optimal nesting configuration that minimizes the total number of glass substrates utilized, subject to the following constraints:

Boundary Compliance: Each blank must be fully contained within the boundaries of the substrate.

Non-Overlapping: Blanks must be arranged such that no two overlap.

Multi-Objective Optimization: Achieve a balanced trade-off between the residual material value and the associated cutting costs.

The proposed nesting scheme utilizes a segmented two-stage layout strategy guided by a state selection matrix. This approach ensures compliance with cutting constraints while optimizing material utilization, reducing production complexity, and enhancing the value of residual materials.

Residual materials can be effectively reused and generate value only if their dimensions satisfy a specific minimum size requirement, ensuring they are sufficiently large to accommodate at least one blank. Given that residual materials are not immediately demanded products, their value realization involves intrinsic risks, such as reprocessing costs and uncertain future demand. As a result, their value is lower compared to newly cut blanks or blanks of the same area. Let the value of a blank be denoted as $v_i = l_i \times w_i$ Consequently, the value of the residual material can be expressed as:

The mathematical model for this class of problems is established as follows:

$$V_i = \begin{cases} clw, & \text{if } 1 \geq l_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Parameter definition: l_m: the minimum blank length; c: the value coefficient of residual materials.

The mathematical model for this class of problems is established as follows:

$$\min Z = \sum_{k=1}^{N} \left(S + \lambda C_k \right) \cdot x_k - V_k \tag{2}$$

$$\text{s. t.} \quad \sum_{k=1}^{K} a_{ki} x_k = d_i, i = 1,...m; \tag{3}$$

For the k-th cutting pattern $(k = 1,\dots,N)$; S: cost of the glass substrate (a constant in this study, as it focuses on single-size substrate nesting); C_k: cutting path length for the k-th pattern; λ: weight coefficient for cutting costs; x_k: frequency of using the k-th pattern; a_{ki}: number of type- i blanks included in the k-th pattern.

3 STATE SELECTION MATRIX

Current industry-standard software and existing academic methods for addressing the RTD-CSP predominantly focus on a single criterion, namely material utilization rate, when selecting cutting patterns and blanks. This excessively simplistic approach often overlooks critical factors, leading to suboptimal solutions that fall short of approximating the global optimum. To overcome this limitation, the MOA algorithm proposed in this study introduces a state selection matrix framework. This framework integrates three interconnected processes: matrix initialization, wave propagation update, and dropout layer filtering. Matrix initialization sets up an initial configuration based on material utilization rates. The wave propagation update then dynamically adjusts weights through iterative learning, while dropout layer filtering refines solutions by selecting stable, high-value configurations and suppressing noise. During the matrix training phase, the algorithm progressively incorporates cutting costs and residual material value as secondary objectives, complementing the primary criterion of material utilization. This multi-objective integration provides quantitative reference criteria for selecting blanks during the simulated annealing iterations [25]. Consequently, the algorithm transitions from a single-objective blank selection process to a balanced multi-objective optimization paradigm, enabling a systematic exploration of the solution space and advancing towards global optimality.

3.1 Matrix Initialization

A rectangular block containing blanks of a single size on a substrate is defined as a homogeneous block, where all blanks are arranged in mutually parallel or perpendicular strips. As illustrated in Fig. 1, blocks labeled 1-4 and 5-6 represent two distinct homogeneous blocks of different sizes.

This study introduces the concept of a segment, which consists of multiple homogeneous blocks. Segments are classified into two types based on their alignment: *X*-directional segments, which comprise homogeneous blocks aligned horizontally from left to right, and *Y*-directional segments, which consist of homogeneous blocks arranged vertically from top to bottom.

To generate high-quality nesting solutions in the initial phase, this research employs a two-stage nesting method consisting of blank transformation and segment condensation to produce a comprehensive nesting scheme. In the blank transformation phase, the optimal nesting segment under current conditions is determined through computational optimization of all homogeneous block combinations within standardized dimensional constraints. Subsequently, during the segment condensation phase, this optimized segment is transformed into a comprehensive sheet layout diagram, which ultimately yields the mathematically-optimal two-dimensional cutting pattern for the raw material.

The initial schemes undergo preliminary screening to eliminate suboptimal candidates, thereby initializing the state selection matrix. In contrast to existing recursive three-block nesting techniques [26], this approach significantly reduces computational time and enhances combinatorial flexibility, enabling diverse and efficient nesting configurations.

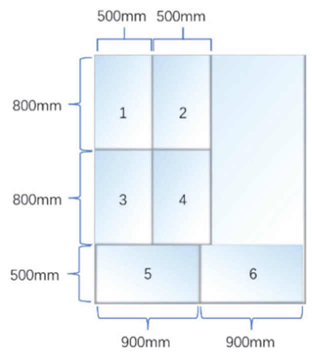

Figure 1 Schematic diagram of homogeneous blocks

3.1.1 Blank Transformation Phase

First, generate all possible combinations of homogeneous blocks based on the dimensions and quantity requirements of the blanks. During the blank transformation phase (Fig. 2), the area of each generated homogeneous block is considered as the item value within the knapsack problem formulation. This approach facilitates the calculation of the maximum allowable usage count for each type of blank:

$$num_j = d_i / blk_{ij} \tag{4}$$

Parameter definition: blk_{ij} : the number of type- i blanks contained in the j-th homogeneous block combination

Subsequently, decompose num_j (the total count of the j-th combination) into a sum of binary-weighted terms plus a residual term, expressed as $num_j = 2^1 + 2^2 + \ldots + 2^k + h$. Each partitioned term is referred to as a "package". For instance, if num_j = 12, it can be decomposed as: $num_j = 2^0 + 2^1 + 2^2 + 5$.

The homogeneous block packages are ultimately transformed into a constraint array w and a value array v . By applying the 0-1 knapsack solution method, the maximum value $r[i][j]$ achievable under constrained segment lengths is determined, representing the optimized value of the segment. This process enables the identification of optimal arrangements for both *X*-directional and *Y*-directional segments. The state transition equation is defined as follows:

$$r[i][j] = \max\left(r[i-1][j], j - w[i] \geq 0\, ?\, r[i-1][j-w[i]] + v[i] : 0\right) \tag{5}$$

Parameter definition: $r[i][j]$: the maximum total value achievable by placing the first i homogeneous blocks into a segment with a constraint length j; $w[i]$: the length of the constraint edge (the edge of the homogeneous block parallel to the segment's length) for the i-th homogeneous block package; $v[i]$: the value of the i-th package, calculated as the number of homogeneous blocks in the package multiplied by the area-based value *val* of each block.

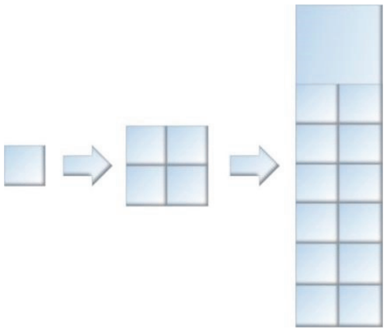

Figure 2 Blank transformation phase

3.1.2 Segment Condensation Phase

In the Segment Condensation Phase (Fig. 3), utilizing the X-directional segment arrangements obtained from the Blank Transformation Phase, the method calculates the maximum allowable usage count, constraint array w^* (which represents the dimensional limits of condensed segments), and value array (dimensional limits of condensed segments), and value array v^* (which encompasses aggregated efficiency metrics). These parameters are subsequently input into the state transition equation provided below to ascertain the maximum achievable value through the optimal embedding of X-directional segments into the substrate, adhering to length constraints:

$$s[i][j] = \max \left(s[i-1][j], j - w^*[i] \geq 0 \, ? \, s[i-1][j - w^*[i]] + v^*[i] : 0 \right) \quad (6)$$

Parameter definition: $s[i][j]$: the maximum value achievable by embedding the first i X-directional segment packages into a substrate sheet with a constrained length j;

$w^*[i]$: the constraint edge length (segment width of the X-directional segment) for the i-th X-directional package; $v^*[i]$: the value of the i-th X-directional package, calculated as the number of X-directional segments in the package multiplied by the area-based value *val*.

The maximum value for embedding Y-directional segments under constrained lengths into the substrate can be calculated in a similar manner by substituting the appropriate variables. Finally, the algorithm compares the $s[i][j]$ values of the optimal X-directional segment arrangement with those of the optimal Y-directional arrangement, selecting the configuration with the higher value as the optimal layout for the current segment on the substrate. The required number of sheets for the optimal layout is then increased, and the two phases, Segment Condensation and Blank Transformation, are iteratively applied until the demand d_i for each blank type is satisfied.

In summary, the proposed method initiates by partitioning the complete substrate into segments using guillotine cuts during the Segment Condensation Phase. Subsequently, these segments are further divided into homogeneous blocks in the Blank Transformation Phase. The optimal nesting of these homogeneous blocks forms efficient segments, whose arrangement contributes to the globally optimal nesting pattern. This approach ensures layouts remain within valid configurations while preserving combinatorial diversity by considering numerous feasible nesting possibilities. By concentrating on solving for optimal segments and nesting patterns directly, the method avoids computationally expensive evaluations of inefficient or impractical layouts, thus significantly reducing the time required to generate high-quality solutions.

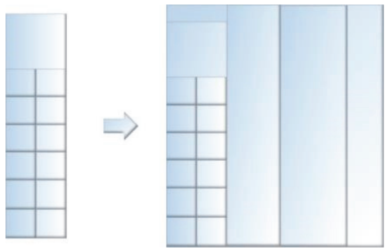

Figure 3 Segment condensation phase

3.1.3 Priority Initialization

Based on the derived nesting solutions, evaluate each solution to determine its material utilization rate. Let α% denote the threshold for excellent solutions. Retain solutions that achieve a utilization rate above α%, while cataloging the number of blanks used in solutions with material utilization rates below α%:

$$Ord_i = \sum_{k=1}^{L} x_k \times a_{ki} \quad (7)$$

Parameter definition: L: the number of solutions with a material utilization rate below α%.

A state selection matrix $P_{T \times M}$ is introduced, where T is the number of solutions with utilization rates above α%, and M is the number of blank types. P_{ij} represents the selection priority of blank Ord_j in the i-th solution. All elements in P are initialized to β. Starting from the last solution in the nesting sequence, the algorithm iterates backward and adjusts priorities based on the remaining stock of each blank Ord_j . If the current solution uses blank j, its priority P_{ij} is incremented; otherwise, it is decremented. The total priority adjustment for each blank type is recorded as:

We introduce a state selection matrix $P_{T \times M}$, where T is the number of solutions with utilization rates above α%, and M represents the number of blank types. P_{ij} signifies the selection priority of blank Ord_j in the i-th solution. All elements in P are initialized to β. Beginning from the last solution in the nesting sequence, the algorithm iterates backward, adjusting priorities based on the remaining stock of each blank Ord_j . If the current solution utilizes blank j, its priority P_{ij} is incremented; otherwise, it is decremented. The total priority adjustment for each blank type is recorded as:

$$I_i = \sum_{k=1}^{N} \left(a_{ki} / S \right) \times [\gamma \times (p+1)] \quad (8)$$

Parameter definition: S: the total remaining number of blanks; γ : the priority adjustment factor; p: the weighting coefficient for each solution, where solutions with higher utilization rates are assigned larger weights.

The formula dynamically adjusts priorities by incorporating both the current solution's utilization rate and the usage frequency of each blank. A higher utilization rate $P_{T\times M}$ or more frequent use of a blank results in a greater adjustment magnitude. Finally, I_i is evenly distributed across each entry $P_{ji}(j=1,2,\dots M)$, completing the initialization process.

3.2 Wave Propagation Update

The matrix initialization offers a preliminary reference standard for selecting blanks during solution generation. However, it lacks adaptive capability across successive iterations, resulting in initial criteria that become inadequate as selection dynamics shift with increasing iterations. To overcome this limitation, this paper introduces a wave-propagation update strategy.

The wave-propagation update mechanism (Fig. 4) includes two modes: positive sampling and negative sampling. This mechanism is primarily based on comparing the i-th solution P_s from the previous iteration ($i=1, 2, \dots, m$, where m is the number of solutions in that iteration) with the i-th solution N_s generated in the current iteration ($i=1, 2, \dots, n$, where n is the number of solutions in this iteration). The comparison criteria are as follows: N_s achieves a material utilization rate greater than or equal to P_s; N_s exhibits a lower Ck (e.g., cutting complexity) than P_s; N_s demonstrates a higher Vk (e.g., value metric) than P_s.

If condition 1 is satisfied, or if condition 1 is not met but conditions 2 and 3 are fulfilled, the current blank selection is considered reasonable, and positive sampling is applied to reinforce the preferred blanks. Otherwise, negative sampling is initiated to correct suboptimal selections.

Positive sampling is further categorized into rapid-growth and slow-growth modes. During the t-th iteration, blank priority adjustments follow predefined correction rules under the following circumstances:

$$P_{ij} \le \frac{\sum_{k=0}^{t} P_{ik} - P_{ji}}{t} + \theta \tag{9}$$

Parameter definition: t: the number of blank types used in the i-th iteration solution; θ : the priority adjustment threshold.

The rapid-growth method is employed to update priorities when the threshold condition is satisfied. This approach accelerates convergence towards the global optimal solution by avoiding suboptimal choices that adversely affect the overall nesting performance. The rapid-growth formula is defined as follows:

$$P_{ji} = P_{ij} + f \times a_{ij} \tag{10}$$

Parameter definition: f : the rapid-growth factor.

When the priority of a specific blank in the current solution surpasses those of other blanks by a predefined threshold, the slow-growth method is employed to decelerate update rates, thereby mitigating the risk of matrix overfitting. The slow-growth formula is defined as follows:

$$P_{ij} = P_{ij} + log_2\left(s \times a_{ij}\right) \tag{11}$$

Parameter definition: s: the slow-growth factor.

In summary, positive sampling reinforces existing high-quality nesting patterns by applying uniform positive adjustments to blank priorities. These adjustments are executed incrementally and are specifically targeted at the currently evaluated solution.

For negative sampling, the process follows 8 in Section 3.1, with a key distinction: after determining I_i, the adjustments are applied only to solutions identified for negative sampling.

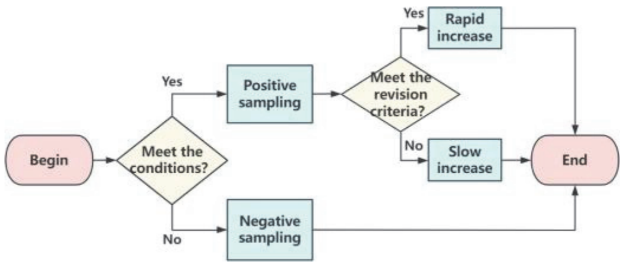

Figure 4 Wave propagation update

3.3 Dropout Layer Filtering

The selection priority is determined by a state selection matrix. This study employs a weighted stochastic selection algorithm to determine the cutting priority at each step, generating cutting patterns based on current blank inventory until stock depletion. This process yields a set of candidate patterns, from which one optimal pattern is selected for expansion while the remaining blanks from other patterns are recycled back into the blank inventory for subsequent pattern generation. To prevent early overfitting in the state selection matrix, a Dropout layer [27] is implemented at this pattern selection stage.

This approach addresses the issue where limited training samples and high parameter dimensionality can lead the wave-propagation update strategy to overly align the matrix parameters with the current training data as iterations progress. Although such alignment might temporarily enhance blank selection and material utilization in the current iteration, it risks degrading performance in subsequent iterations due to poor generalization.

The Dropout layer introduces stochasticity by considering the remaining stock of each blank type. Specifically, for the j-th blank in the i-th iteration, dropout activation occurs if its required usage exceeds the following threshold:

$$Ord_j \ge \delta \times S \tag{12}$$

Parameter definition: δ : the threshold for forcing the selection of this blank type.

This suggests that blank j has become the dominant component in the remaining stock for the current order. The manner in which blank j is incorporated into subsequent solutions will significantly impact the final material utilization rate of the entire nesting scheme. To address this, position P_{ij} in the state selection matrix is blocked, effectively freezing its value during updates. This compels the algorithm to prioritize solutions that include blank j in the current iteration.

4 MULTI-OBJECTIVE ADAPTIVE NESTING ALGORITHM

The integration of the state selection matrix markedly improves material utilization rates, cost efficiency, and the value of residual materials when compared to traditional standalone simulated annealing algorithms. This chapter elaborates on an advanced blank cutting rule designed to maximize the value of residual materials. This component refines the MOA algorithm (Fig. 5), contributing to a more thorough optimization of the nesting process.

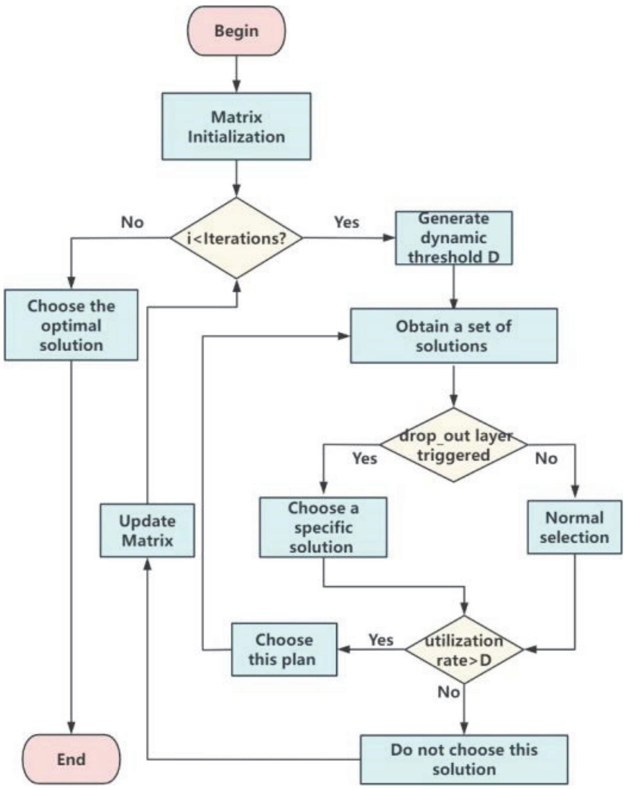

Figure 5 Multi-objective adaptive nesting algorithm

4.1 Residual Material Analysis

In the TFT-LCD cutting process, effectively managing leftover materials presents a significant challenge. The cutting of glass substrates often generates irregular shapes and off-cuts, resulting in substantial waste. This issue is compounded by the industry's stringent demands for precision and efficiency, which tend to prioritize the material utilization rate over the repurposing of residual materials. Consequently, these residuals accumulate, driving up material costs and increasing environmental impact. The lack of efficient strategies for utilizing these leftovers not only restricts resource optimization but also undermines efforts toward achieving sustainable manufacturing practices within the industry.

4.2 Enhanced Blank Cutting Criterion

To comprehensively evaluate the impact of each blank selection on subsequent operations, ensuring material optimization while maximizing residual value, this study introduces a novel blank cutting criterion. A fundamental assumption is that both blanks and sheets must satisfy the condition length ≥ width; otherwise, they are rotated to meet this requirement prior to cutting.

As illustrated in Fig. 6, there are four cutting configurations for positioning a blank on a sheet. The cutting method employed in this study adopts a guillotine-cutting approach, where each cutting path is designed to traverse from one edge of the sheet to the opposite edge. After each cut, two new residual sheets (Raw1 and Raw2) are recursively generated. To determine the optimal cutting direction (X-axis pass vs. Y-axis pass), a judge function is employed to quantify the downstream impact of the cut, defined as follows:

$$F = nS_r / W \times L + \lambda l_R w_R \tag{13}$$

Parameter definition: n: the number of blanks that can be cut from sheet Raw based on the current order quantity; S_r: the area of the selected blank; l_R, w_R: the length and width of the larger residual sheet generated after cutting.

The judge function evaluates the impact of X-axis and Y-axis cutting passes separately. The decision rule is: X-axis pass is selected if the judge value for X exceeds that of Y. Y-axis pass is chosen if the converse holds;

A random selection between the two is made if the values are equal.

Assuming that rotating a blank to satisfy Case 3 or Case 4 (as illustrated in Fig. 6) allows conversion to Case 1 or Case 2, priority is given to Cases 3 and 4 during cutting. When these cases are infeasible, the system defaults to Cases 1 and 2, utilizing a probabilistic exploration mechanism (e.g., an ε - greedy strategy) to avoid local optima.

This dual selection mechanism uniquely determines the optimal cutting configuration for the current sheet, effectively balancing the maximization of residual value with computational efficiency.

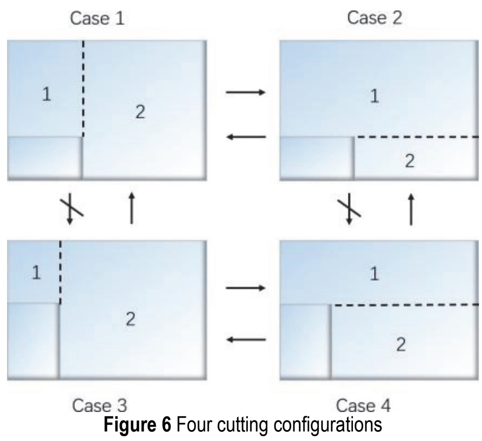

Figure 6 Four cutting configurations

5 EXPERIMENTAL DATA AND TEST RESULTS

To validate the effectiveness of the algorithm, comparative experiments were conducted using benchmark datasets from previous studies as well as real-world enterprise orders. The experiments were run on a personal computer equipped with an Intel(R) Core(TM) i7-11800H CPU @ 2.30GHz central processing unit, 32 GB random access memory, and Windows 10 operating system.

For standardized performance evaluation, we quantify $W_i (i = 1, 2...n)$ as the area of waste material within each sheet. A unified metric S_W is defined as follows:

$$S\_W_{jkm} = \sum W_i (wl_i \ge l_{\min}, ww_i \ge l_{\min}) \tag{14}$$

Parameter definition: j: the problem index; k: the nesting plan index for a specific problem; m: the algorithm type: m = 0: Algorithm from Reference [28]; m = 1: Proposed algorithm (this work). wl_i, ww_i: the length and the width of the W_i; $l_{\min}$: minimum length (or width) of blanks in the current blank library.

As illustrated in Fig. 7, a 9 × 9 sheet contains two 3 × 2 blanks and two 4 × 4 blanks, yielding $l_{\min} = 2$. Following (14), residual materials a and b are not classified as waste since both their length and width exceed $l_{\min}$. In contrast, residual c with a width of 1 (which is smaller than $l_{\min}$) qualifies as waste material. Consequently, the total waste area of this sheet equals the area of c, which is 4 square units.

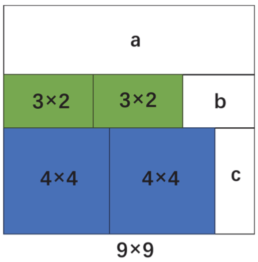

Figure 7 Schematic diagram of waste material classification criteria

$$\Delta = \frac{\sum_{k=0}^{n} S\_W_{jk0} - \sum_{k=0}^{n} S\_W_{jk1}}{\sum_{k=0}^{n} S\_W_{jk0}} \tag{15}$$

Clearly, a higher value of Δ indicates that compared to the method in Reference [28], the MOA algorithm generates less waste material, with residual pieces having higher utilization value and reuse potential, thereby enabling greater material savings in subsequent cutting operations.

5.1 Comparative Analysis of Cutting Costs

In the following experiments, the number of iterations of the state selection matrix is 5.

The experimental validation employed two case studies from Reference [28], with detailed parameters outlined in Tab. 1. This dataset is centered on small-scale sheet materials and limited blank quantities, which typify common small- size cutting challenges. While material utilization rates remain consistent across methods under these constraints, our analysis highlights significant differences in cutting costs and residual material value.

Fig. 8 provide a visual comparison of the nesting layouts generated by the proposed method and those from Reference [28] for Case 1 and Case 2, respectively. Regarding cutting efficiency, for Case 1, the cutting path length in Reference [28] is 328 units, whereas our proposed method reduces it to 291 units. Similarly, for Case 2, the cutting path length decreases from 641 units in Reference [28] to 587 units with the proposed approach.

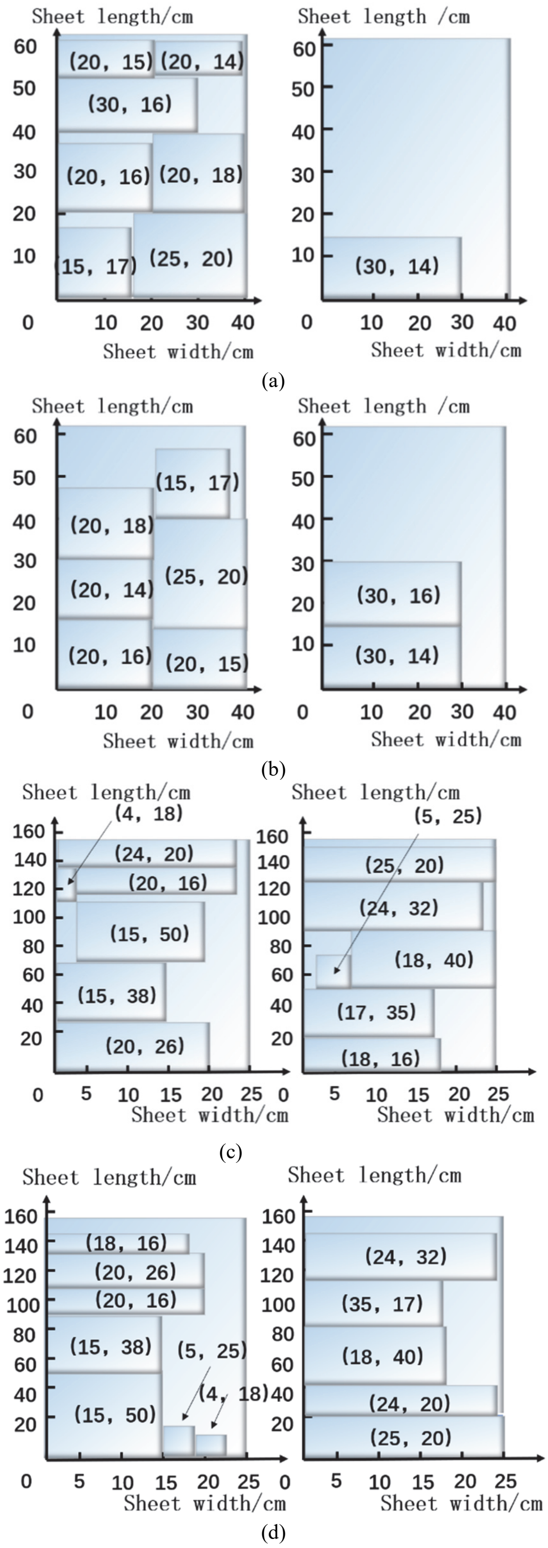

Figure 8 (a) Problem 1: Nesting layout strategy of the algorithm in reference [28]; (b) Problem 1: Nesting layout strategy of the MOA algorithm; (c) Problem 2: Nesting layout strategy of the algorithm in reference [28]. (d) Problem 2: Nesting layout strategy of the MOA algorithm

These reductions in cutting path length directly translate into shorter machining cycles, reduced operational complexity, and extended tool lifespan, collectively yielding substantial reductions in production costs.

Table 1 Two test cases

Problem	Sheet Size	Types of sheet	Problem
1	(40, 69)	8	(25, 20), (16, 20), (15, 20), (14, 20), (20, 18), (15, 17), (30, 16), (30, 14)
2	(25, 150)	12	(32, 24), (26, 20), (25, 20), (24, 20), (40, 18), (35, 17), (20, 16),(18, 16), (38, 15), (50, 15), (18, 4), (25, 5)

5.2 Residual Value Comparison

A comparative analysis of residual material value was conducted for the two cases from Reference [28]. The S_W metric, which measures the area of waste material, for both the nesting layouts from Reference [28] and the proposed algorithm is summarized in Tab. 2.

The optimization results demonstrate that both proposed schemes achieve improved material utilization, with Scheme 1 and Scheme 2 reducing waste area by 14.8% and 34.3% respectively. The nesting method in Reference [28] leads to fragmented residual regions that become non-reusable scrap. In contrast, our algorithm maintains structurally coherent residual blanks that are suitable for future orders.

Table 2 Test results

Item	Reference [28] Layout 1	Reference [28] Layout 2	Sum	Article Layout 1	Article Layout 2	Sum	Δ
1	265	140	405	45	300	345	14.8%
2	78	112	160	53	52	105	34.3%

5.3 Comparative Analysis of Profile Optimization Rates

In the following experiments, the number of iterations of the state selection matrix is 50, and the final result is the optimal solution selected from the 50 iterations as the final scheme.

This study examines instances 1-4 in Tab. 2 of Reference [29], which involve large-scale nesting problems characterized by oversized sheet materials and a high volume of blanks. The comparative analysis emphasizes the profile material utilization rates in relation to the data from the original study. Tab. 3 provides a summary of the key performance metrics for both approaches.

Table 3 Comparative data of sheet consumption between the proposed method and the approach in Reference [29]

	Instance1	Instance2	Instance3	Instance4
PSO	3634	1908	4665	4191
MOA	3633	1908	4521	4034
Optimization rate	0.02%	0	3.08%	3.75%

Among the four test cases, three exhibit higher profile optimization rates than those reported in Reference [29], with an average improvement of 1.66%. These results confirm that the proposed algorithm maintains robust performance even when applied to large-scale blank cutting problems, effectively balancing material utilization with computational efficiency.

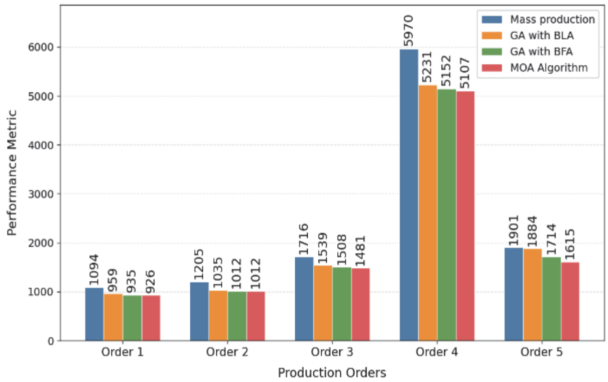

Figure 9 A comparative analysis of LCD sheet consumption conducted between the MOA algorithm and three methods from Reference [30]

As shown in Tab. 4, to evaluate the practical applicability of this algorithm in industrial production, we selected experimental orders 1-5 conducted in a 10th Generation facility within the TFT-LCD industry in Taiwan, as referenced in Tab. 7 of Reference [30], for comparison. The glass substrate dimensions for these orders were 3050 × 2850 mm. We compared the MOA algorithm with the batch production method, the genetic algorithm with bottom-left packing algorithm (GA-BLA) from Reference [31], and the genetic algorithm with best-fit algorithm (GA-BFA) from Reference [32] in terms of the number of sheets required.

The results, illustrated in Fig. 9, demonstrate significant improvements in sheet utilization for the MOA algorithm compared to batch production, GA-BLA and GA-BFA. Specifically, for Order1 to Order5, the MOA algorithm achieved material utilization increases of 15.3%, 16.0%, 13.7%, 14.5%, and 15.0%, respectively, over the industry-standard batch production method. These improvements highlight the algorithm's ability to reduce material waste and enhance efficiency in large-scale industrial applications through optimized nesting layouts.

To validate the practical performance of the algorithm, we analyzed real-world blank orders and glass substrate dimensions from a TFT-LCD manufacturer for the period from April to June. The glass substrates were categorized into three sizes: 2200 × 1870 mm, 2400 × 2160 mm, and 3080 × 2880 mm. Tab. 5 provides detailed parameters for eight test cases, including the number of blank types (k), their dimensions (L_i, W_i), and demand ranges (d_i).

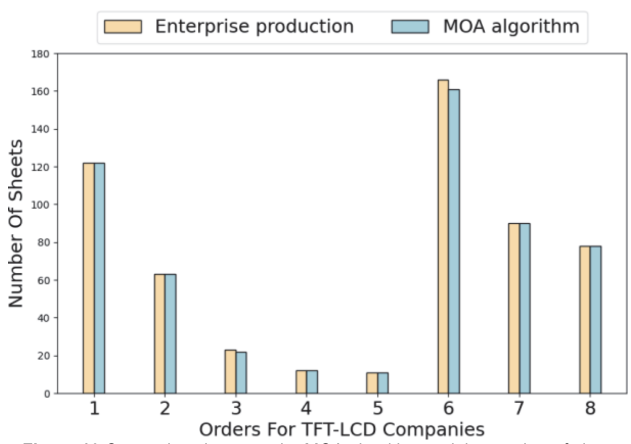

Figure 10 Comparison between the MOA algorithm and the number of glass substrates used in production by a TFT-LCD enterprise

The experimental results, shown in Fig. 10, confirm that the MOA algorithm consistently matches or surpasses the manufacturer's current production plans in minimizing glass substrate consumption. Specifically, the MOA algorithm achieves equal or reduced sheet usage while maintaining cutting feasibility. This demonstrates its effectiveness in reducing material waste in real-world manufacturing scenarios, offering enterprises tangible cost-saving opportunities through optimized nesting layouts.

Table 4 Comparison data with reference [28]

Order	Blank Size (L, W) / mm								
	(15, 20)	(90, 56)	(93, 60)	(99, 63)	(95, 66)	(98, 64)	(100, 62)	(108, 72)	(126, 82)
1	1000	1000	1000	2000	1000	1000	1000	1000	2000
2	2450	1670	1300	1120	1850	2450	1670	1300	1120
3	1500	2500	3000	2500	2000	1500	2500	3000	2500
4	7000	7000	7000	7000	7000	7000	7000	7000	7000
5	1150	1500	1250	1600	2500	1150	1500	1250	1600

Table 5 Comparison with company order data and results

Sheet size	k	L	W	d
2440 × 2000	12	[746, 1349]	[612, 952]	[3, 280]
3300 × 2600	12	[746, 1349]	[612, 952]	[3, 280]
2440 × 2000	13	[1252, 2400]	[358, 770]	[1, 26]
3300 × 2600	13	[1252, 2400]	[358, 770]	[1, 26]
3660 × 2600	13	[1252, 2400]	[358, 770]	[1, 26]
2440 × 2000	8	[312, 1932]	[496, 978]	[27, 260]
3300 × 2600	8	[312, 1932]	[496, 978]	[27, 260]
3660 × 2600	8	[312, 1932]	[496, 978]	[27, 260]

6 CONCLUSIONS

To address energy consumption issues in the cutting of glass substrates for TFT-LCD manufacturing, this study proposes an innovative MOA algorithm. Building upon the concept of machine learning, the algorithm incorporates a state selection matrix to optimize the choice of blanks at each cutting step and enhances blank-cutting criteria to maximize the value of residual materials. Experimental results demonstrate that the algorithm excels in terms of material utilization, cutting costs, and residual value. It has already been successfully implemented in operational factories, yielding substantial economic benefits and providing valuable solutions for the advancement of Industry 4.0 and smart factories.

Future research will expand this algorithm's application to other engineering fields, such as sheet metal cutting, glass processing, shipbuilding, vehicle manufacturing, furniture production, newspaper layout, and garment and leather cutting. Our ongoing work will focus on developing and refining algorithms that are more closely aligned with the practical needs of industrial production.

7 REFERENCES

[1] Sheng, Z. M., Gang, Y. J., & Liu, X. (2022). Mechanism study and improvement of vertical crosstalk in large-size high-resolution TFT-LCDs. *Chinese Journal of Liquid Crystals and Displays*, *37*(01), 37-44.

[2] Jung, J., Park, H., Jung, H. Y., Jung, S. E., Kim, S. G., Kim,T. H., … Lee, S. H. (2023). Recent progress in liquid crystal devices and materials of TFT-LCDs. *Journal of Information Display*, *25*(1), 121-142. https://doi.org/10.1080/15980316.2023.2281224

[3] Huang, Y. S., Wang, S. A., & Fang, C. C. (2017). Optimization of the LCD optical film cutting problem. *International Journal of Production Research*, *55*(15), 4411-4435. https://doi.org/10.1080/00207543.2016.1256507

[4] Adamowicz, M. & Albano, A. (2010). A solution of the rectangular cutting-stock problem. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 302-310. https://doi.org/10.1109/TSMC.1976.5408780

[5] Luo, Q., Du, B., Rao, Y., & Guo, X. (2022). Metaheuristic algorithms for a special cutting stock problem with multiple stocks in the transformer manufacturing industry. *Expert Systems with Applications*, *210*, 118578. https://doi.org/10.1016/j.eswa.2022.118578

[6] Mostajabdaveh, M., Salman, F. S., & Tahmasbi, N. (2022). Two dimensional guillotine cutting stock and scheduling problem in printing industry. *Computers & Operations Research*, *148*, 106014. https://doi.org/10.1016/j.cor.2022.106014

[7] Kokten, E. S. & Sel, Ç. (2022). A cutting stock problem in the wood products industry: a two-stage solution approach. *International Transactions in Operational Research*, *29*(2), 879-907. https://doi.org/10.1111/itor.12802

[8] Turbian, C. (2024). Two-Dimensional Cutting and Packing Problems in the Sheet Metal Industry.

[9] Yi, X. Y., Tong, Q. S., &Pan, W. P. (2015). A solving method of two-dimensional cutting for the rectangular blank. *International Forging Stamping Technology*, *40*(06), 150-154.

[10] Ji, J., Huang, D. H., Ng, F. F., & Cui, Y. D. (2021). Optimal Two-Section Layouts for the Two-Dimensional Cutting Problem. *Journal of Information Processing Systems*, *17*(2), 271-283. https://doi.org/10.3745/JIPS.01.0066

[11] Chen, Y., Zheng, X., Lu, S., & Hu, X. (2022). Optimization algorithm for rectangular parts cutting stock problem with cutting cost. *Computer Integrated Manufacturing System*, *28*(12), 3859.

[12] Andrade, R., Birgin, E. G., & Morabito, R. (2016). Two-stage two-dimensional guillotine cutting stock problems with usable leftover. *International Transactions in Operational Research*, *23*(1-2), 121-145. https://doi.org/10.1111/itor.12077

[13] Ayasandır, U. & Azizoğlu, M. (2024). Two-dimensional cutting stock problem with multiple stock sizes. *International Journal of Manufacturing Technology and Management*, *38*(2), 95-125. https://doi.org/10.1504/IJMTM.2024.137486

[14] Jin, X. B. (2023). A study of two-dimensional cutting stock problem based on improved genetic algorithm. *Yangtze River Information and Communication*, *36*(08), 79-80+85.

[15] Kwon, J. H., Kim, H. J., & Lee, S. (2024). Optimizing Traffic Scheduling in Autonomous Vehicle Networks Using Machine Learning Techniques and Time-Sensitive Networking. *Electronics*, *13*(14), 2837. https://doi.org/10.3390/electronics13142837

[16] Yang, Y., Shen, H., & Tian, H. (2024). Scheduling workflow tasks with unknown task execution time by combining machine-learning and greedy-optimization. *IEEE Transactions on Services Computing*, *17*(3), 1181-1195. https://doi.org/10.1109/TSC.2024.3351622

[17] Zhou, H., Erol-Kantarci, M., Liu, Y., & Poor, H. V. (2024). Heuristic algorithms for RIS-assisted wireless networks: Exploring heuristic-aided machine learning. *IEEE Wireless Communications*. https://doi.org/10.1109/MWC.010.2300321

[18] Niu, Y., Xu, C., Liao, S., Zhang, S., & Ao, J. (2024). Multi-objective location-routing optimization based on machine learning for green municipal waste management. *Waste Management*, *181*, 157-167. https://doi.org/10.1016/j.wasman.2024.04.001

[19] Tonnius, A. (2023). *Novel Machine Learning Algorithms for NP-hard Combinatorial Problems*. Doctoral dissertation, Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2023.

[20] Kaleta, M. & Śliwiński, T. (2025). Neural-Driven Constructive Heuristic for 2D Robotic Bin Packing Problem. *Electronics*, *14*(10), 1956. https://doi.org/10.3390/electronics14101956

[21] Cheng, M. Y., Fang, Y. C., & Wang, C. Y. (2021). Auto-tuning SOS algorithm for two-dimensional orthogonal cutting optimization. *KSCE Journal of Civil Engineering*, *25*(10), 3605-3619. https://doi.org/10.1007/s12205-021-0522-y

[22] Wu, D. J., Yan, C. P., Li, J., & Cao, W. D. (2018). Manufacturability-oriented rectangular parts cutting stock method. *Computer Integrated Manufacturing Systems*, *24*(06), 1374-1382. https://doi.org/10.1016/j.rcim.2003.10.006

[23] Qiulian, C. H. E. N., Renkun, S. O. N. G., & Yaodong, C. U. I. (2017). A Method for Three-Staged Two-Dimensional Guillotine Cutting with Usable Leftover. *Journal of Graphics*, *38*(1), 10. https://doi.org/10.11996/JG.j.2095-302X.2017010010

[24] Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, 380-384. https://doi.org/10.1109/COMITCon.2019.8862255

[25] Bao, Q. J. B.,Jiang, J. Q., & Song, C. Y. (2008). Optimal stock cutting based on particle swarm optimization and simulated annealing. *Computer Engineering and Applications*, (26), 246-248.

[26] Luo, Y. Z. & Pan, L. W. (2018). Three-block layout pattern and its generation algorithm of constrained cutting problem. *Forging & Stamping Technology*, *43*(10), 185-189.

[27] Wang, R., Yan, F., & Lu, J. (2021). Trend Prediction of New Crown Pneumonia on Dropout-LSTM Modeling. *Journal of University of Electronic Science and Technology*, *50*(03), 414-421.

[28] Huang, Y. H., Lu, H. C., & Wang, Y. C. (2020). Two-Dimensional Cutting Stock Problem in the Manufacturing. *Application of Decision Science in Business and Management*, *167*. https://doi.org/10.5772/intechopen.89376

[29] Ayadi, O., Masmoudi, M., Ben Ameur, M., & Masmoudi, F. (2017). A new PSO-based algorithm for two-dimensional non-guillotine non-oriented cutting stock problem. *Applied Artificial Intelligence*, *31*(4), 376-393. https://doi.org/10.1080/08839514.2017.1346966

[30] Lu, H. C. & Huang, Y. H. (2015). An efficient genetic algorithm with a corner space algorithm for a cutting stock problem in the TFT-LCD industry. *European Journal of Operational Research*, *246*(1), 51-65. https://doi.org/10.1016/j.ejor.2015.04.044

[31] Hopper, E. & Turton, B. (1999). A genetic algorithm for a 2D industrial packing problem. *Computers & Industrial Engineering*, *37*(1-2), 375-378. https://doi.org/10.1016/S0360-8352(99)00097-2

[32] Beasley, J. E. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations research*, *33*(1), 49-64. https://doi.org/10.1287/opre.33.1.49

Contact information:

Yuan XU
The Department of Medical Imaging,
The Affiliated Guangdong Second Provincial General Hospital of Jinan University,
Guangzhou, China

Jin FANG
The Department of Medical Imaging,
The Affiliated Guangdong Second Provincial General Hospital of Jinan University,
Guangzhou, China

Guihua JIANG
The Department of Medical Imaging,
The Affiliated Guangdong Second Provincial General Hospital of Jinan University,
Guangzhou, China

Hong YE
The Department of Internet, Anhui University,
Hefei, China

Ke WANG
The Department of Internet, Anhui University,
Hefei, China

Jiaming CHANG
The Department of Internet, Anhui University,
Hefei, China

Leilei BO
The Department of Internet, Anhui University, Hefei, China

Jingxuan WANG
The Department of Internet, Anhui University,
Hefei, China

Jiyang ZHU
(Corresponding author)
The Department of Internet, Anhui University,
Hefei, China
E-mail: zhujiyang_ahu@163.com