IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 5, Ser. 2 (Sept. – October. 2023), PP 11-20
www.iosrjournals.org

Exhaustive Plane Packing Algorithm For Irregular Sheet Material

Chiang Ling Feng
Assistant Professor, Bachelor Degree Program of IOT Engineering and Applications, Yu Da University of Science and Technology, Taiwan

Abstract
The goal is to place patterns of different shapes within a specified rectangular area to minimize wasted space. This problem belongs to a combinatorial optimization problem known as the "two-dimensional bin packing problem," which is typically used to arrange flat objects with different shapes to minimize the filled space. The system employs optimization algorithms and collision checks to arrange patterns of different shapes by rotating and mirroring them to minimize wasted space. Its objective is to place as many of these shapes as possible within the given rectangular area while ensuring they do not overlap with each other. This is a challenging combinatorial optimization problem that often requires various optimization techniques and algorithms to find suitable solutions. This paper also analyzes and compares the results of two methods: randomly arranging the boards and consistently using the same pattern of boards for placement.

Date of Submission: 24-09-2023 Date of Acceptance: 04-10-2023

I. Introduce

Sheet material arrangement plays a pivotal role across various sectors, encompassing everyday life and industry. In furniture manufacturing, materials like wood, synthetic boards, and metal sheets must be strategically organized to minimize material wastage and enhance production efficiency. Similarly, in construction and renovation projects, large sheets such as gypsum boards, laminated sheets, and wooden boards require thought-out arrangement to cover surfaces like walls, ceilings, and floors, effectively reducing material waste and costs. This optimization is equally critical in the packaging industry, where it's imperative to arrange cardboard boxes, packaging materials, and boxes efficiently to minimize cardboard or paper usage and decrease packaging expenses.

In the textile sector, arranging fabrics of different sizes and shapes is essential to minimize waste and maximize yield, spanning clothing manufacturing, home textiles, and industrial textiles. The printing industry benefits from thoughtful sheet, label, and sticker arrangement, conserving paper and ink and improving printing efficiency. Meanwhile, the metal processing industry frequently involves arranging metal sheets for cutting, stamping, welding, and machining, a practice that significantly reduces material waste and enhances production efficiency.

Efficient arrangement extends to the food processing industry, where organizing food products to fit packaging containers, baking trays, or racks streamlines production and minimizes food wastage. The manufacturing of paper products, such as envelopes, cards, books, and paper boxes, necessitates proper paper and cardboard arrangement to curtail waste. In aerospace engineering, optimizing material arrangement and cutting is indispensable for crafting components for aircraft and spacecraft, thereby reducing costs and enhancing performance. In electronics manufacturing, closely arranging circuit boards and semiconductor wafers minimizes waste and elevates production efficiency.

These instances underscore the ubiquitous nature of sheet material arrangement across various domains. By judiciously managing materials, costs can be slashed, resource wastage curtailed, and production efficiency boosted.

As per our knowledge, the two-dimensional irregular (non-convex) blank cutting problem has received scant attention in prior research. In the literature, employing an envelope polygon represents a relatively straightforward approach for addressing the layout problem associated with two-dimensional free-form or irregular shapes [2-4]. The fundamental premise of free-form layout is to enclose irregular shapes with regular polygons and subsequently replace the original irregular shapes with regular ones. Irregular shapes can be enclosed with rectangles, circles, triangles, or other regular polygons. However, it's worth noting that while the envelope polygon simplifies the layout problem for irregular shapes, it can lead to wastage areas due to the enclosing process. Elkeran [5] introduced an envelope polygon method based on pairwise clustering, where shapes with analogous contour features are grouped, and then the envelope polygon approach is applied to the layout. Nonetheless, this method doesn't fundamentally mitigate wastage areas.

Various methodologies have been proposed to tackle related challenges across different domains. For example, Hopper and Turton devised a solution for the 2D strip packing problem [6]. Burke et al. suggested employing hill climbing and Tabu search techniques to address irregular strip packing [7]. Júnior et al. adopted a greedy bottom-left corner placement strategy and integrated genetic algorithms as a heuristic search engine [8]. Han et al. tackled the constrained two-dimensional irregular (convex) bin packing problem [9]. López-Camacho et al. introduced an adaptive approach to deal with the irregular (convex) bin packing problem [10]. Additionally, various optimization algorithms, such as simulated annealing, Tabu search, neural networks, genetic algorithms, and particle swarm optimization, have found applications in the realm of cutting and packing problems [8-11].

In the context of this paper, [5] proposed a layout strategy grounded in fuzzy matching. This strategy includes employing the longest common subsequence to identify geometrically similar features and utilizing the proposed layout algorithm to resolve collisions. Nevertheless, these approaches did not address the issue of selecting board patterns under multiple styles, which can influence the final reduction of board wastage.

This paper introduces a novel concept hinging on the disparity between the area of a polygon set and its convex hull area. The allocation of polygons undergoes rigorous feasibility tests, encompassing angle tests, boundary assessments, point inclusion evaluations, and polygon intersection analysis, to prevent overlaps and yield diverse polygon placements while optimizing the objective function. The paper's structure is delineated as follows: the second section outlines the principles of sheet layout, the third section deliberates on the results of minimum board area wastage derived from 500 simulations under diverse conditions and presents the layout outcomes while scrutinizing subtle layout nuances, and finally, the fourth section synthesizes conclusive insights pertaining to various conditions and methodologies.

II. Algorithm

Due to the complex concave-convex features of freeform shapes, it significantly increases the difficulty of determining the relative positions between shapes [16]. In the system, we first define patterns of different shapes, including quadrilaterals, triangles, and other shapes. Each shape is represented as a polygon composed of a set of coordinates. Arrangement optimization involves arranging these shapes within a specified rectangular area to minimize wasted space.

To achieve this goal, the program uses an optimization algorithm that attempts to place each shape in the rectangle at different angles and mirror orientations. Collision detection occurs when attempting to place a shape, where the program checks if it intersects with already placed shapes. This is achieved by constructing paths (Path) for the shapes and using intersection checks. If a shape intersects with any already placed shape, the placement attempt fails.

Rotation and mirroring involve rotating and mirroring each shape to find the best placement. It tries different rotation angles and mirror options to minimize wasted space. The system employs an optimization process that attempts to place each shape multiple times to find the best arrangement. If a suitable arrangement cannot be found within a specified number of attempts, the shape is skipped.

Once all shapes are placed, the program calculates the total area occupied by the placed shapes and the area of the unused rectangular region, thereby determining the percentage of waste. In [5], the material sheet is treated as a virtual rectangle, and its area calculation is as follows.

$$[max|_{k=1}^{M}\{x_{k\theta}\} - min|_{k=1}^{M}\{x_{k\theta}\}][max|_{k=1}^{M}\{y_{k\theta}\} - min|_{k=1}^{M}\{y_{k\theta}\}]$$

But in fact, this is incorrect. It will incorrectly reduce the loss area of the arrangement of the boards, and may even result in a negative ratio of board loss area. When calculating the area of the boards, the boards must be considered as polygons, and the area can be obtained based on the polygon area calculation method. The algorithm uses the shoelace formula: Consider a polygon on a plane, let(x_0, y_0), (x_1, y_1), ……., (x_{n-1}, y_{n-1}) represent the coordinates of the vertices sorted counterclockwise. The area of this polygon is denoted as *A* and can be calculated as the sum of the determinants of the *n* two-by-two matrices.

$$A = \frac{1}{2}\begin{vmatrix} x_0 & x_1 & x_2 \cdots x_{n-2} & x_{n-1} & x_n \\ y_0 & y_1 & y_2 \cdots y_{n-2} & y_{n-1} & y_n \end{vmatrix}$$

$$= \frac{1}{2}\left(\begin{vmatrix} x_0 & x_1 \\ y_0 & y_1 \end{vmatrix} + \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \cdots + \begin{vmatrix} x_{n-2} & x_{n-1} \\ y_{n-2} & y_{n-1} \end{vmatrix} + \begin{vmatrix} x_{n-1} & x_0 \\ y_{n-1} & y_0 \end{vmatrix}\right)$$

Regarding the arrangement of boards, it involves placing the given shape within a specified bounding box to minimize wasted space, while considering the possibilities of rotation and mirroring. This is executed based on the following principles and algorithms:

(1) Random Position Selection: The function first attempts to place the shape at random positions within the

bounding box. These random positions ensure diversity in multiple attempts.

(2) Loop for Rotation and Mirroring: Next, the function uses two nested loops to try different rotation angles and mirroring options. The outer loop increments the rotation angle in 5-degree steps, and the inner loop tries both mirroring options (True and False).

(3) Rotation and Mirroring of Shape: Within the inner loop, the function uses a function to calculate the shape after rotation and/or mirroring. This function rotates the coordinates of the shape by a given angle and applies mirroring based on the mirror parameter.

(4) Collision Detection: For each combination of rotation and mirroring, the function checks if the new shape intersects with any of the previously placed shapes. This is achieved by constructing the path of the shape and using path intersection checks. If the new shape intersects with any placed shape, it is marked as a collision.

Line segment *AB* intersects with line segment *CD*, so we can obtain two vectors *AC* and *AD*. Points *C* and *D* lie on opposite sides of *AB*, with vector *AC* in the counterclockwise direction from vector *AB* ($AB \times AC > 0$), and vector *AD* in the clockwise direction from vector *AB* ($AB \times AD < 0$), resulting in opposite-signed cross products. In other words, if the two endpoints *C* and *D* of line segment *CD*, when connected to one endpoint of another line segment (*A* or *B*, only one of them), form vectors and their cross product with vector *AB* yields opposite signs, it indicates that *C* and *D* are on opposite sides of the line *AB*. If the cross product yields the same sign, it means that both points of *CD* are on the same side of *AB*, indicating that they do not intersect. As for the formula for the cross product of two vectors *a* (*x1, y1*) and *b* (*x2, y2*), it is given by

$$a \times b = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = x_1 y_2 - y_1 x_2$$

(5) Angle Difference Calculation: If the new shape does not collide, the function calculates the angle difference between the new shape and the already placed shapes. This angle difference is computed by comparing the angles of the new shape with all the angles of the placed shapes (including mirrored cases). The goal is to find the rotation angle and mirror state that are closest to the already placed shapes.

(6) Selecting the Best Placement: Among all the combinations of rotations and mirror states, the function selects the combination with the smallest angle difference as the best placement. This best combination includes the rotation angle, mirror state, and the shape after rotation. The function repeats this process, making up to 100 attempts, to find a suitable placement. If a suitable placement cannot be found within the specified number of attempts, the function returns False.

(7) Success or Failure of Placement: Each shape is represented as a list of points, denoted as 'shape', representing a closed polygon. The list also defines the quantity of each shape to be placed. The rotation function is used to rotate a shape by a given angle (in degrees). It accepts a shape and a rotation angle as parameters and calculates the rotated shape using a rotation matrix. When attempting to place a shape within the bounding box, it first tries rotating the shape in 360-degree steps to find the best position. It randomly generates an x and y offset and starts rotating the shape from 0 degrees, incrementing by 5 degrees in each step, until it reaches 360 degrees. It checks whether the rotated shape is within the bounding box and does not intersect with already placed shapes. If a suitable position is found, it adds the shape to the 'placed_shape' list and

returns a message indicating successful placement. If a suitable position cannot be found after 100 attempts, it returns a message indicating that placement was not possible.

In summary, this function is a multi-step process that attempts to place shapes in a way that minimizes waste by randomly selecting positions, different rotation angles, and mirror options. The rotation formula for this process is described as follows [5].

$$\begin{bmatrix} x_{k\theta} \\ y_{k\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

Where (x_k, y_k) represents the coordinates of the *k*th vertex of the original sheet. $(x_{k\theta}, y_{k\theta})$ represents the coordinates of the *k*th vertex of the original sheet after rotating by an angle θ. This paper also considers collision detection and angle differences to select the best placement method. This method, after multiple attempts, will find a relatively better placement, but it may not necessarily find a global optimal solution.

Finally, we draw the result graph on the screen. First, we draw a closed polygon, which includes the four corners of the rectangle and a closed outline to mark the boundary of the rectangle. The coordinates of this polygon's points are specified in two lists, one for x-coordinates and one for y-coordinates. Then, using a for loop, we iterate through the shapes that have already been placed. We also check if a shape needs to be mirrored. If it does, we use list comprehension to reverse the shape's y-coordinates to achieve a mirrored display, flipping the shape horizontally. We also draw each of the placed shapes. If a shape needs to be mirrored, we reverse its y-coordinates during drawing to achieve the mirrored effect. Each shape is represented by a list containing all its coordinate points. Furthermore, we set the aspect ratio of the plot to ensure that the proportions of the shapes in the graph are maintained without distortion due to the size of the plot. Finally, all the placed shapes and the rectangular bounding box are displayed in one graph.

Based on the above analysis, it can be concluded that in the context of sheet layout optimization, the mathematical model for optimizing sheet arrangement is represented as:

$$\max \quad \eta = max \quad \frac{1}{L}\left\{\sum_{i=1}^{K_i} A_i\right\}$$

such that

$$\begin{cases} \max \quad x = max(x, \forall(x,y) \ in \ shape) \le L \\ max \quad y = max(y, \forall(x,y) \ in \ shape) \le W \\ 0 \le \sum_{i=1}^{K_i} A_i \le LW \\ 0 \le K_i \le M_{i,max} \end{cases}$$

III. Simulation Results

In this section, we conducted simulations for each condition 500 times and then determined the result with the lowest material wastage area. First, let's look at the results obtained without rotation or mirroring, as shown in Figure 1, where the minimum wastage area obtained is 44.52%. If we consider arranging the shapes with quadrilaterals first and allow adjacent edges to be closely connected, as shown in Figure 2, the results yield a minimum wastage area of 35.88%.

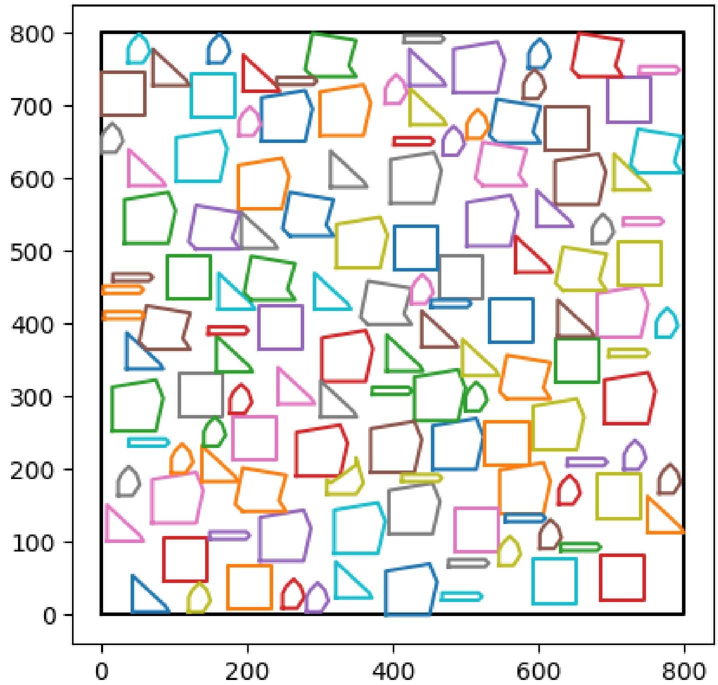

Figure 1: Layout diagram executed before rotation and mirroring

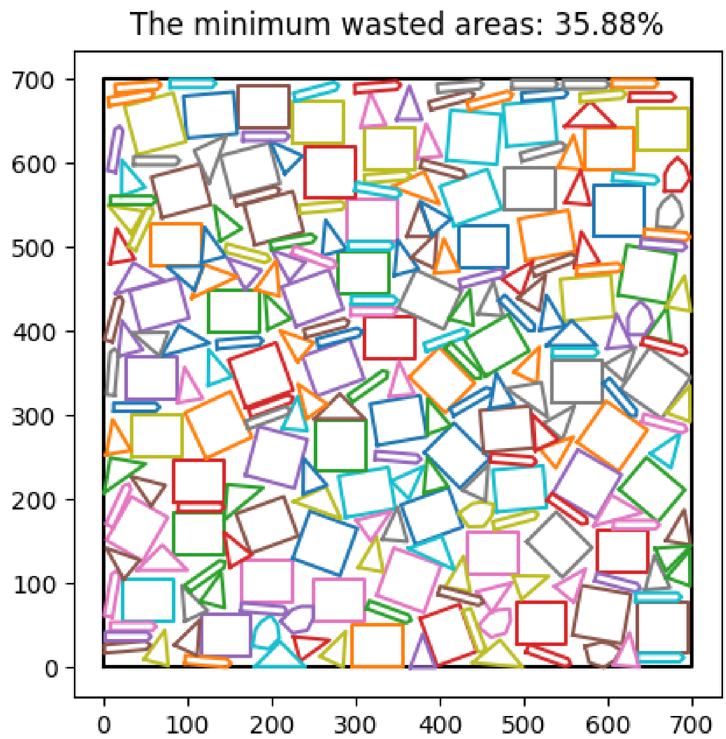

Figure 2: Layout diagram executed without rotation, but with specific panels placed first

When we incorporate a 360-degree rotation method, as shown in Figure 3, the results yield a minimum wastage area of 34.77%. However, when we introduce flipping and mirroring as shown in Figure 4, it appears that the results are worse, with a minimum wastage area of 36.98%.

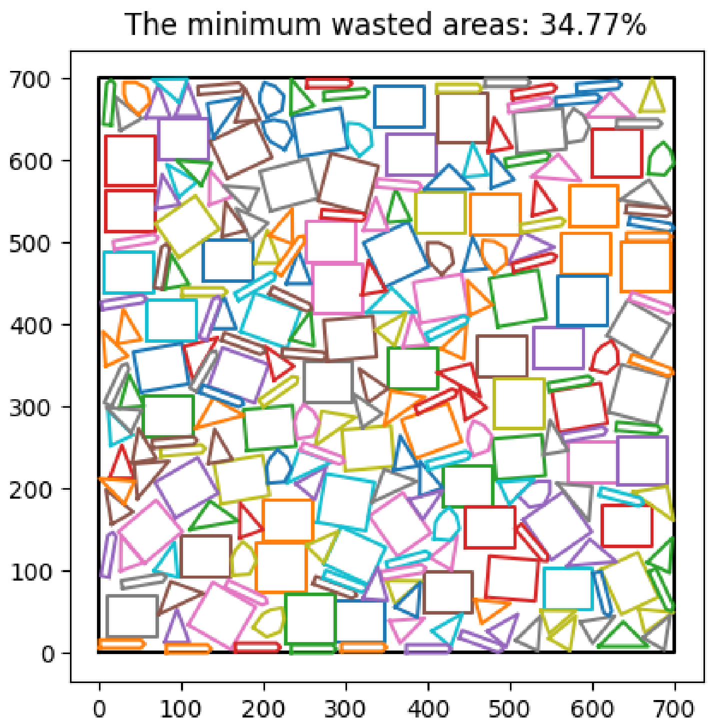

Figure 3: Layout diagram executed with rotation mechanism.

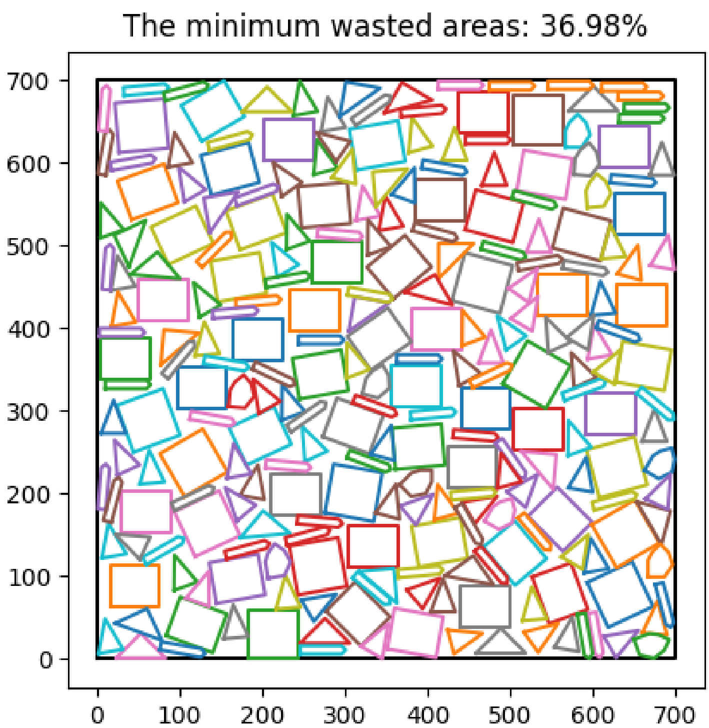

Figure 4: Layout diagram executed with flipping and mirroring mechanisms

From the simulations, it can be observed that because quadrilaterals have larger areas, if more quadrilaterals can be accommodated, the relative wastage area will be smaller. Polygons result in area wastage. Therefore, from Figure 2 to Figure 4, it can be seen that almost no pentagons, which occupy more board space, are placed. This is because most of the large areas are occupied by quadrilaterals, and there is not enough space left for pentagons. However, if we do not consider arranging quadrilaterals and triangles first and instead arrange the boards with a random seed, we will see that the result obtained, as shown in Figure 5, has a minimum wastage area of 34.97%. This value is close to what is shown in Figure 3. At the same time, from Figure 5, we can also see that quadrilaterals indeed contribute significantly to the efficiency of the layout. As for boards with larger and more irregular shapes, it seems that they waste more area after layout. Therefore, under the condition of minimizing board wastage, these boards will not be placed within the larger boards.

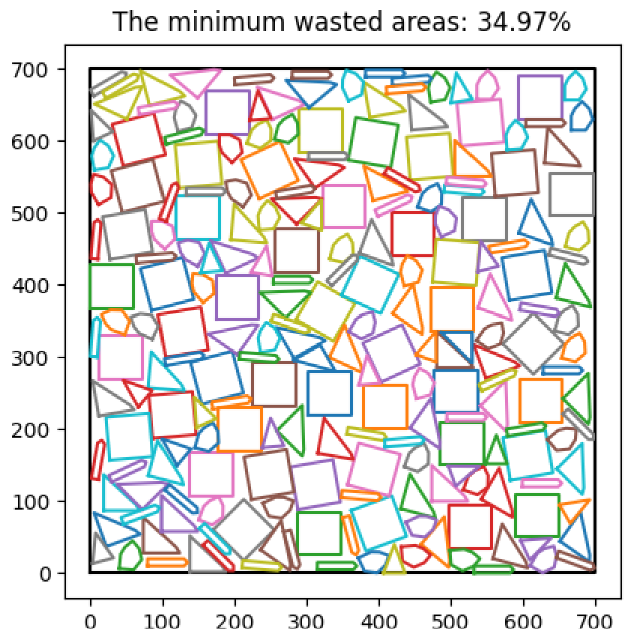

Figure 5: Layout diagram executed with random panel selection and rotation mechanism.

If the quadrilateral is removed, then under the mechanism of randomly selecting boards and rotating them, the layout of the boards executed will be as shown in Figure 6. It is evident that this performance will be better than when there is a quadrilateral present.

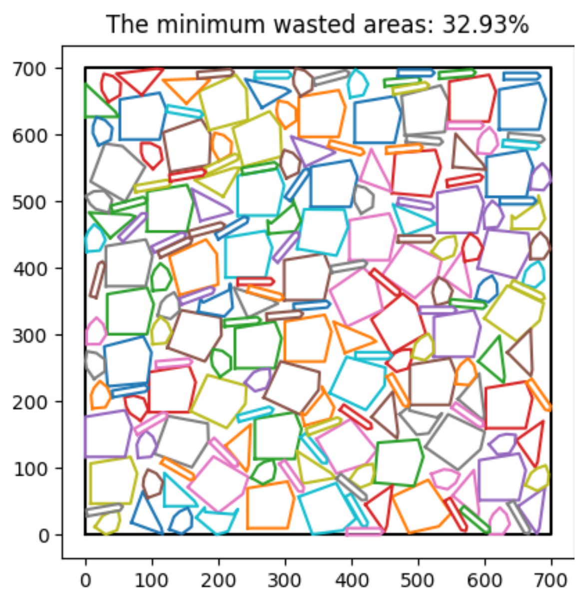

Figure 6: Arrangement Diagram of Pure Irregular Shaped Sheets Under Random Sheet Selection and Rotation Mechanism

To confirm whether there is a difference in efficiency between randomly selecting and arranging sheets and the method of continuously using the same type of sheet until it cannot be arranged, we simulated the method of continuously using the same type of sheet until it cannot be arranged. As shown in Figure 7, the minimum wastage area obtained from this method is 39.16%.

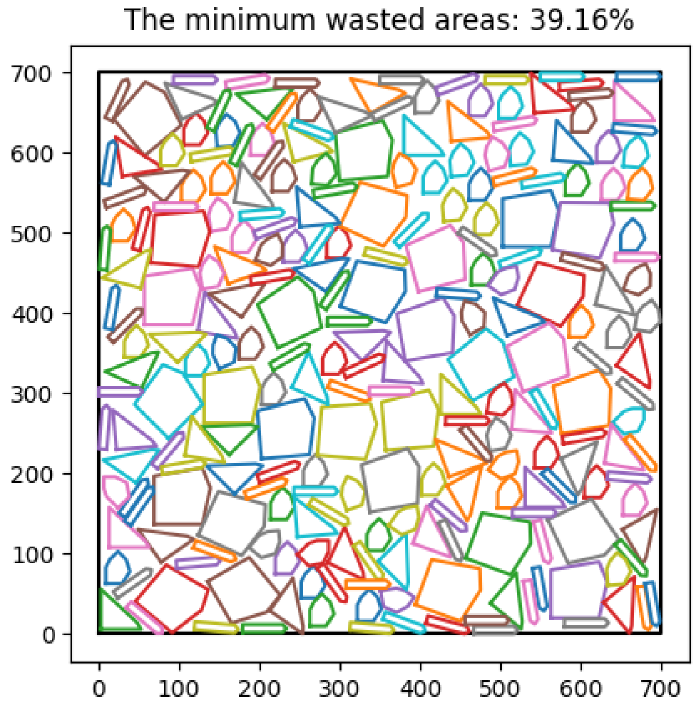

Figure 7: Layout diagram executed with the same style of panel selection and rotation mechanism for irregular-shaped panels.

IV. Conclusions

From the discussion in the previous section, it can be concluded that panels with similar areas, in the shape of quadrilaterals, suppress the arrangement of other polygons. Additionally, it is evident that randomly selecting and arranging panels yields better results than consistently using the same panel style. This is because pentagons, which have larger areas than quadrilaterals in this context, result in more wasted space due to their irregular shapes. In this paper, we propose two exhaustive search algorithms and a random search algorithm for the two-dimensional cutting stock problem, with constraints on the maximum quantities of each type of workpiece to be produced. The algorithms limit the size of the search by deriving and applying necessary conditions for the optimal cutting patterns. Node evaluation techniques are used to generate upper bounds during the search. The computational performance of the algorithms is demonstrated through testing on a large number of randomly generated problems with different tightness constraints. The results show that the algorithms are efficient procedures for solving medium-sized cutting problems.

References

[1]. Stephane Bonnevay , Philippe Aubertin , And Gérald GAVIN , "A Genetic Algorithm To Solve A Real 2-D Cutting Stock Problem With Setup Cost In The Paper Industry", Proceedings Of The 2015 Annual Conference On Genetic And Evolutionary Computation,

Pp. 807–814, July, 2015.

[2]. A. K. Sato, T. D. C. Martins, And M. D. S. G. Tsuzuki, "A Pairwise Exact Placement Algorithm For The Irregular Nesting Problem," International Journal Of Computer Integrated Manufacturing, Vol. 29, No. 11, Pp. 1177–1189, 2016.

[3]. Nicos Christofides, And Charles Whitlock, "An Algorithm For Two-Dimensional Cutting Problems" Operations Research, Vol. 25, No. 1, Pp. 30-44, Junary-Februrary, 1977.

[4]. Doraid Dalalah, Samir Khrais, And Khaled Bataineh, "Waste Minimization In Irregular Stock Cutting", Journal Of Manufacturing Systems, Vol. 33, No. 1, Pp. 27-40, January 2014.

[5]. Yan-Xin Xu, "An Efficient Heuristic Approach For Irregular Cutting Stock Problem In Ship Building Industry", Mathematical Problems In Engineering, Pp. 1-12, 2016

[6]. J. A. Bennell And J. F. Oliveira, "A Tutorial In Irregular Shape Packing Problems," Journal Of The Operational Research Society, Vol. 60, No. 1, Pp. S93–S105, 2009.

[7]. A. M. Gomes And J. F. Oliveira, "A 2-Exchange Heuristic For Nesting Problems," European Journal Of Operational Research, Vol. 141, No. 2, Pp. 359–370, 2002.

[8]. R. Alvarez-Valdes, F. Parreño, And J. M. Tamarit, "A Tabu Search Algorithm For A Two-Dimensional Non-Guillotine Cutting Problem," European Journal Of Operational Research, Vol. 183, No. 3, Pp. 1167–1182, 2007.

[9]. D. S. Liu, K. C. Tan, S. Y. Huang, C. K. Goh, And W. K. Ho, "On Solving Multiobjective Bin Packing Problems Using Evolutionary Particle Swarm Optimization," European Journal Of Operational Research, Vol. 190, No. 2, Pp. 357–382, 2008.

[10]. E. K. Burke, G. Kendall, And G. Whitwell, "A Simulated Annealing Enhancement Of The Best-Fit Heuristic For The Orthogonal Stock-Cutting Problem," INFORMS Journal On Computing, Vol. 21, No. 3, Pp. 505–516, 2009.

[11]. P. R. Pinheiro, B. Amaro Júnior, And R. D. Saraiva, "A Random-Key Genetic Algorithm For Solving The Nesting Problem," International Journal Of Computer Integrated Manufacturing, Vol. 29, No. 11, Pp. 1159–1165, 2016.

[12]. Baosu Guo, Jingwenm Hu, Fenghe Wu,And Qingjin Peng, "Automatic Layout Of 2D Free-Form Shapes Based On Geometric Similarity Feature Searching And Fuzzy Matching", Journal Of Manufacturing Systems, Volume 56, Pp. 37-49, 2020.

[13]. Sangmin Lee, Hyun-Gu Kahng, Taesu Cheong, Seoung Bum Kim, "Iterative Two-Stage Hybrid Algorithm For The Vehicle Lifter Location Problem In Semiconductor Manufacturing", Journal Of Manufacturing Systems, Vol. 51, Pp.106-119, April 2019.