Automatic two-dimensional layout using a rule-based heuristic algorithm

KIM, Yunyoung
Department of Marine Systems Engineering, Kyushu University

Gotoh, Koji
Department of Marine Systems Engineering, Kyushu University

Toyosada, Masahiro
Department of Marine Systems Engineering, Kyushu University

出版情報：Journal of Marine Science and Technology. 8 (1), pp.37-46, 2003-06. Springer
バージョン：
権利関係：

Automatic two-dimensional layout using rule-based heuristic algorithm

YUNYOUNG KIM, KOJI GOTOH and MASAHIRO TOYOSADA

Department of Marine Systems Engineering, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581, Japan

Abstract: The efficient layout strategies are introduced for an automatic nesting using a rule-based heuristic approach to improve the layout efficiency. Since a large portion of the complexity of the part layout problem results from the overlapping computation, geometric redesign techniques are also suggested to reduce the complexity of the problem as a fast and reliable mean for performing the overlapping computation. A new heuristic sliding technique is developed to find near-optimum layout location among all the feasible arrangements in such a manner that two or more arbitrary parts do not overlap or intersect. An identification method of pivoting point is suggested to calculate the boundary of the overlap region in fast computation time frame. The resource plate clipping using virtual memory, based on polygon clipping algorithm, is also proposed as a technique to reduce geometric conditions of the resource plate and overlap computation time by updating a new stock boundary of the resource plate for layout space of the next part after each part is placed. The aim of this paper is to develop the rule-based heuristic nesting system achieving a new automatic layout on the AutoCAD system. For this implementation, some of nesting examples are demonstrated. Consequently, the rule-based heuristic approach can be desirable in layout efficiency considering computational time for nesting problem, and proposed as a real-time layout simulation method in the industrial field.

Key words: Efficient layout strategies · Rule-based heuristic approach · Heuristic sliding · Geometric redesign techniques · Polygon clipping algorithm

1 Introduction

In two-dimensional nesting problem, the material saving is one of the most important factors to be considered, and it is well known that a well-nested part layout can result in a substantial saving of the resource material. Although each industry requires different functional constraints due to its own characteristics, one common goal is to minimise the wastage of resource plate by finding a most desirable layout of parts.

An important problem making the nesting problem difficult to handle is that the computation time required to obtain an optimal solution increases exponentially as the number of parts increases. Nesting is a peculiar engineering problem as there are no precise mathematical descriptions for the process, and there are an infinite number of possible solutions for most problems. Therefore, the nesting can be classified into the hard combinatorial optimisation problems termed NP-complete, and consequently cannot be solved exactly in the polynomial time. That is, it is difficult to find optimum solution for such layout problems within reasonable calculation time frame, because they include a lot of underlying combinatorial conditions. Moreover, since the nesting problem is non-differentiable, a deterministic gradient-based optimisation technique is not suitable for problems with non-differentiable objective functions or constraints, and several local minima[1].

Several commercial CAD/CAM packages, which include a nesting module for solving the layout problem, are very expensive in price. It also takes a lot of time to learn the modules of them. Therefore, since it is actually impossible to buy and use them in medium and small-sized shipbuilding or related industrial fields, the CAD/CAM systems using an interactive editing method is used at the present time. The most of these systems depend strongly on operator's experiences. In recent years, a number of successful nesting approaches have been developed to solve the problem of nesting of regular or irregular shapes by using the various deterministic, heuristic, probabilistic and evolutionary algorithms[2-9]. However, it seems that there are substantial difficulties in the practical application due to high varieties and complexity of the parts. As it were, the most of literatures have been reported the nesting problem with small number of parts.

According to above reasons, since a new user-friendly nesting system for an automatic cutting process in industrial area should be developed, this paper addresses to meet on its request. This system will enable to quickly prepare efficient layouts for generating a sequence of path commands for a numerically controlled cutter,

which can be a flame torch, plasma arc or laser beam.

Since a large portion of the complexity of the part layout problem results from the overlapping computation, the part geometric redesign techniques and efficient layout strategies are introduced to reduce the complexity of the problem as a fast and reliable mean for performing the overlapping computation. These proposed methods can easily handle parts and resource plate with widely varying geometrical shape. These methods for two-dimensional nesting problem are also used to simplify the shape identification algorithm and reduce the amount of processed data and run time, and consequently turn out to be reasonable in the considered application.

Consequently, the rule-based heuristic nesting can be desirable in layout efficiency considering computational time for nesting problem, and proposed as a real-time layout simulation method in the industrial field.

2 System configuration and geometric topology

The proposed nesting system is coded in the ARX/ADS structure[10] with C++ language for AutoCAD R2000 as a working package. The system is able to accept input files in DXF format, and describe the final arrangements in DXF format. This is to enable the system to be integrated with existing CAD/CAM systems found in the shipbuilding or various industrial fields. The system provides facilities for designer to input part geometry either by using parametric macros developed[11] or by converting part geometry insert from another drawing file. This system also supports some of environmental requirements for nesting and cutting process.

The first task of nesting procedure is to identify a representation modelling for defining part geometry with sufficient accuracy, and at the same time demanding less data storage space. An arbitrary two-dimensional part can be divided into entities such as line, arc and circle segments. The system stores coordinates of start and end point for lines, coordinates of centre point, start and end point for arcs, and centre point and radius for circles. A facility is also provided in this system to represent a polygon by a set of data points, which act as control points for the polygon. Once the existence of polygons in the part geometry is defined, the system converts the polygons into a polyline, subsequently treats as a number of line segments in computation. The approximation of the arc or circle is controlled by defining the possible a few number of mid points on the arc or circle.

Each designed part is described by specifying a reference point and a polyline as a sequence of points given relative to the reference point. The part with multiply formed polygons is handled by decomposing them into two or more simply formed polygons whose relative positions and orientations are fixed and connected. But, a database of these polygons is formed as a polyline component database for entities of the part.

There are two kinds of geometric database structures for parts in nesting procedure: first-database and second-database structure. The first-database structure describes the precise representation of part geometry of the input shape. This database structure will be used to describe the shape after the final placement has been decided. The second-database structure contains an approximate description of the actual shape that will be used during all the layout procedure. The goals of the data structures developed to achieve nesting are divided into three types: using available memory efficiently, providing for fast computation, and keeping structures simple and intuitive to allow class-module enhancements.

3 Part geometric redesign techniques

3.1 Polygonal clustering with layout angles

There are several aspects of the pairwise clustering problem that make it potentially quite difficult to handle. One of difficulties is the potentially unmanageable combinatorial problem associated with selecting: (1) a pair of shapes, (2) an absolute orientation for one shape, and (3) a relative orientation for the second shape with respect to the first. The other difficulty derives from the problem of finding the near-optimum clustering of two shapes with given orientations. The clustering technique presented here is a procedure that determines if it is possible to improve the clustering by rotating the target part neighbouring to the source part.

To obtain a good clustering of parts, the method of coinciding the edges of the part with the edge of the target boundary is generally used in the industrial field. Therefore, the parts should have a regular rotation angles: zero degree, 90°, 180° and 270°. To find out a suitable layout angle for clustering, it is achieved by finding the minimum circumscribing rectangular area between current part and previous part(s).

The procedure presented above can be considered to produce clusters of more than two shapes by approaching the multiple shape clustering problem as a multistage pairwise clustering problem. One could first obtain the near-optimal clustering of two parts and combine the two shapes to form one composite part. Then one would obtain the near-optimal clustering of the composite part with another shape and form the new composite part. The procedure could be repeated as many times as one wishes. This polygonal clustering technique is expected to be effective to shorten the calculation time and to improve scrap ratio.

3.2 Polygonal simplification technique

If the overall geometric structure of the polygonal curve should not change in the polygonal simplification technique of reducing the number of vertices, this technique makes a new polygon that quite closely resembles the original polygon. This main reason to be interested in polygon simplification is that it can be used to reduce memory allocated during nesting procedure, and then calculation time. Therefore, simplification is very useful in order to make data storage, transmission, computation, and display.

In practical shipbuilding, the parts having complicated shapes can be formed into simplified ones, when the formation does not affect the scrap ratio negatively. Small holes, cut-outs and scallops included in parts can be omitted to simplify their complicated shapes as shown in Fig. 1.

A two-dimensional part consists of one external feature and several internal features. In nesting procedure, only external feature information is considered for nesting, if the internal feature information can be neglected for placement. Therefore, the blank geometry representation can be applied for simplification technique, because the internal features may not affect the nesting process. Since this polygonal simplification technique has reduced database during nesting procedure, it is expected to be effective to shorten the calculation time.

The simplification technique is considered as follows[5]:

Case 1: When there are small holes in a hull structure part and no other parts can be placed in the holes, they are omitted from the part's shape.

Case 2: Cut-outs and scallops forming a hull structure part are omitted from the part's shape unconditionally.

Case 3: Arcs are replaced by straight lines, when they are included in geometrical elements of a part's shape.

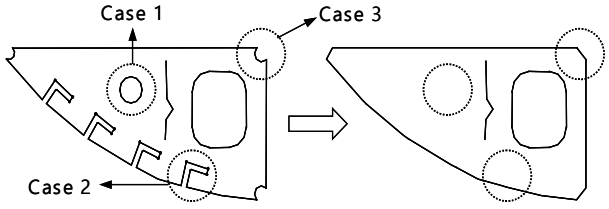


Fig. 1. Polygonal simplification technique

3.3 Polygonal offset technique

This function creates an offset of the selected entities for simplifying the part at a specified distance. The offset distance is the distance from the selected entities to the new entities. The important thing is to decide the side to which the offset must be made. In order to consider the kerf-width between parts for nesting procedure, in this paper, the parts are first expanded using this offset technique and then placed on the resource plate. This offset technique can also reduce the database of part.

In Fig. 2(b∼d), the middle vertex moves to the suitable distance (d') along with bisector direction of angle (θ) that three neighbouring vertices make. This moving distance (d') can be considered in three types of cases due to the shape of polygon. Here, the value h is the distance between parts. For inner-polygon, the direction of offset has reverse direction to outer-polygon.

Case 1: $0 < \theta \leq 84°$ (Fig. 2(b))

After shooting a ray along with bisector direction of angle (θ) that three neighbouring vertices make, the new vertices (g and f) -which are intersected at a point(e)- are obtained. The moving distance is the $d' = h/2$ as an expanding distance of part. Therefore, the expanding polygon $(A' - g - f - C')$ is formed. Since the line $\overline{B \cdot B'}$ is bigger than kerf-width $(h/2)$ remarkably, the layout efficiency is not good. So, that is why the new polygon $(A' - B' - C')$ is not adopted. The angle value (84°) is obtained from the trial and error simulation. But, since the offset result between 84° and 90° is almost same, the 90° can be also used.

Case 2: $\theta > 84°$ (Fig. 2(c))

The new vertex B', which is moved to the distance $d' = h/[2 \cdot \cos\{(180^\circ - \theta)/2\}]$ along with bisector direction of angle (θ), is formed and then new expanding polygon $(A' - B' - C')$ is generated.

Case 3: Arcs exist (Fig. 2(d))

Arc $(B - U - C)$ is moved to distance $d' = h/2$ along with bisector direction of angle (θ), and then new arc $(B' - U' - C')$ is generated. Here, if B' is expanded along with curvature of arc, new point e' -which is intersected with line $\overline{A' \cdot V}$ - will be generated. By similar procedure, new point f' is also generated. Therefore, a polygon $(A' - e' - B' - C' - f' - D')$ expanded can be simplified as a new polygon $(A' - e' - f' - D')$ by reducing the number of vertices.

Figure 3 shows the offset procedure for simplifying the part shape to reduce geometric information. Since the part moves along with non-overlapping direction while the reference point of part travels on the possible vertices of resource plate in section 4.2, the number of vertices of

resource plate is most important factor to reduce layout time. Therefore, this offset technique is also applied to resource plate.

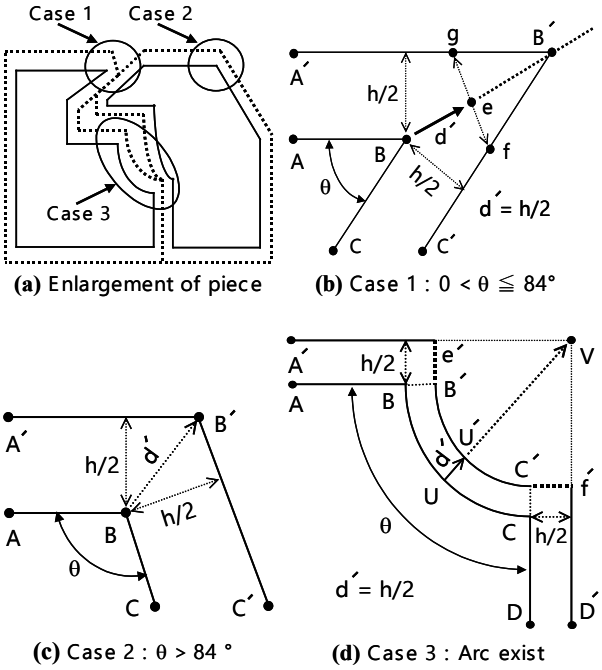


Fig. 2. Polygonal offset technique

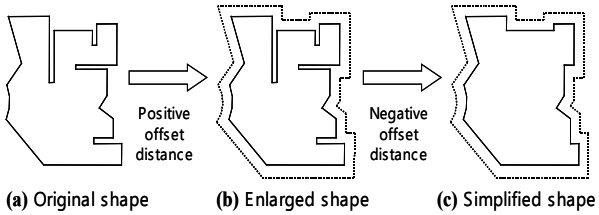


Fig. 3. Simplification of part due to offset technique

4 Efficient layout strategies

4.1 Polygon clipping algorithm

The process of clipping area occupied between two or more polygons is needed in nesting system for efficient layout of parts. A polygon clipping algorithm is adopted to classify the geometric topologies considering the analytical sets of the union, intersection, and difference between two or more polygons. That is, this algorithm checks the intersection among the parts, and is served as a useful tool for determination of movable distance of the parts in heuristic sliding technique.

Figure 4 is an example with overlapped shapes between two polygons A and B. Polygons A and B are a polygon consisting of one outer- and one inner-contour, respectively. The polygon clipping algorithm, which is based on the theory of Schutte[12], consists of the following steps:

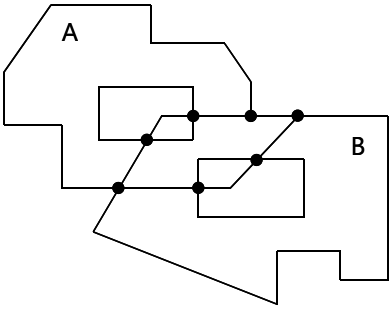


Fig. 4. Two overlapped polygons

Step 1. Calculating the intersections
The intersection points between two polygons A and B are calculated by intersecting all the edges of the first polygon with all the vertices of the second polygon. These intersection points are added to the polygons as vertices. The new created vertices do contain a link to the corresponding vertices on the other polygon as shown in Fig. 5. Schutte[12] called the resulting polygons augmented polygons.

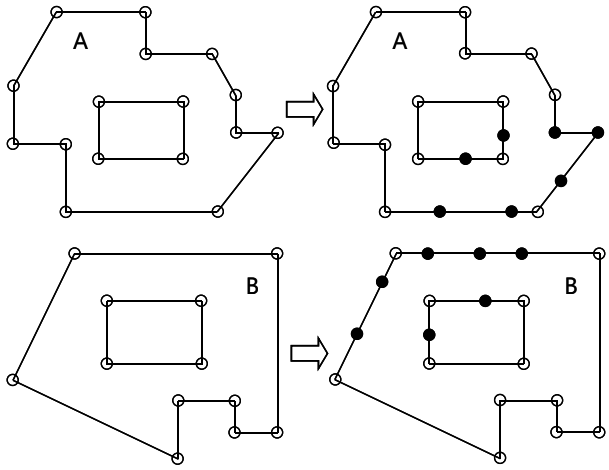


Fig. 5. Polygons before and after intersection calculation

Step 2. Labelling edges
All edges in the augmented polygons should be labelled "*Inside*, *Shared*, or *Outside*". Inside means that an edge is inside of the other polygon. An edge is Shared if both vertices of this edge are connected to the other polygon and if the two vertices they are connected to on the other polygon are the two vertices of one edge. Outside means that an edge is outside of the other polygon. Figure 6 shows the edge and contour labelling of the augmented polygons. Here, the continue lines are Outside, the dot lines are Inside and the dash lines are Shared.

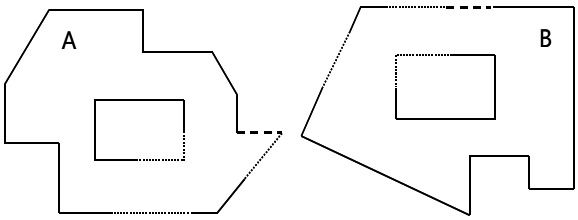


Fig. 6. Labelling for the edges of the augmented polygons

Step 3. Finding minimal polygons

All the edges from both augmented polygons are doubled in a forward and backward edge, which are called directed edges. Shared edges are doubled only once. Figure 7 shows the directed edges. For all these directed edges we search for the smallest clockwise oriented polygons. This is performed by following the directed edges such that at every intersection we proceed with that directed edge which starts at that intersection, and which has the smallest angle with the previous directed edge. A directed edge can only be part of one polygon. This also leads to one counter-clockwise polygon, which is the outer contour of the union. This polygon should be deleted from the set of minimal polygons.

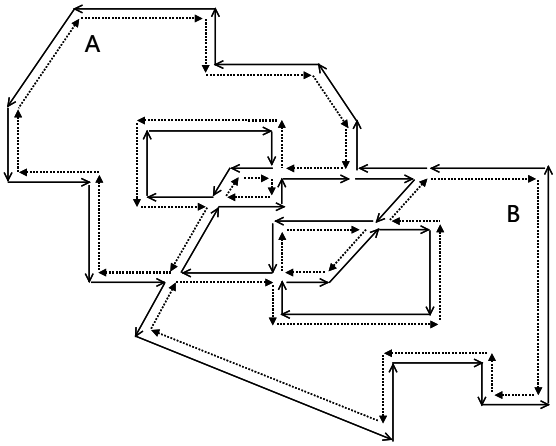


Fig. 7. Forward and backward edges

Step 4. Classifying minimal polygons

The set of minimal polygons found covers the whole area of the union of the two input polygons. Now we have to classify each of the polygons to be member one of the sets $A \cup B$ [Fig. 8(a)], $A \cap B$ [Fig. 8(b)], A-B [Fig. 8(c)] or B-A [Fig. 8(d)]. The classification of the minimal polygons is performed in a two-step process. First, we check the parenthood, which is defined as a relation between edges and input polygons and between minimal polygons and input polygons, of each minimal polygon. After that, we decide to which set the minimal polygon belongs, based on its parenthood.

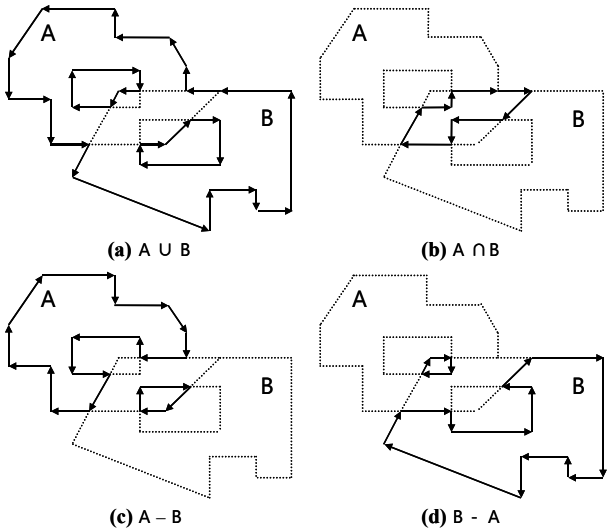


Fig. 8. Classifying the resulting polygons

4.2 Heuristic sliding technique

This section addresses a heuristic sliding technique for nesting of parts with various shapes. If any part is placed on the resource plate, the resource plate has new shape with/without overlapped area of the nested part by using polygon clipping algorithm. This sliding technique has been developed to find out all the feasible or optimal arrangements so that two or more arbitrary shapes may take positions that do not overlap.

The principle of the sliding technique can be explained as follows: Given a part and resource plate. If the part is slid like Fig. 9, the part has to travel along all vertices (boundary edges) of resource plate to find optimum layout vertex so that the travelling time will be tedious. Therefore, the travelling path on the resource plate must be reduced.

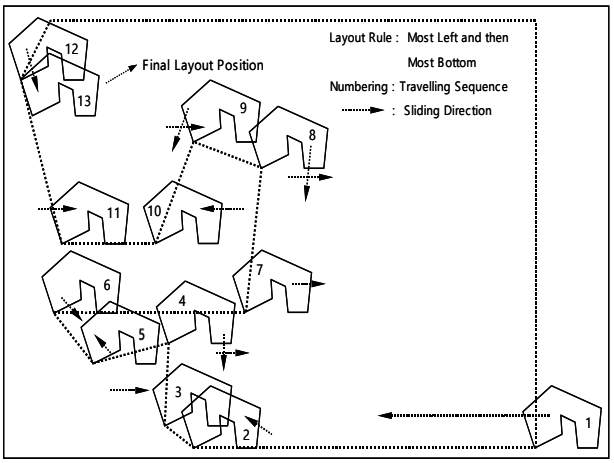


Fig. 9. Travelling sliding of a part

Because of this reason, a heuristic sliding technique is suggested as follows: First of all, the location and angle of all vertices on the resource plate are calculated in advance. With these data, the vertices are sorted by the nearest vertex data to x and y-coordinate, here x value has priority to y value. The reference point of part moves along suitable vertices of the irregular resource plate, which is formed by using polygon clipping algorithm, as shown in Fig. 10. If the moving part finds out the feasible or optimal layout vertex and then overlaps the resource plate, the sliding technique is used.

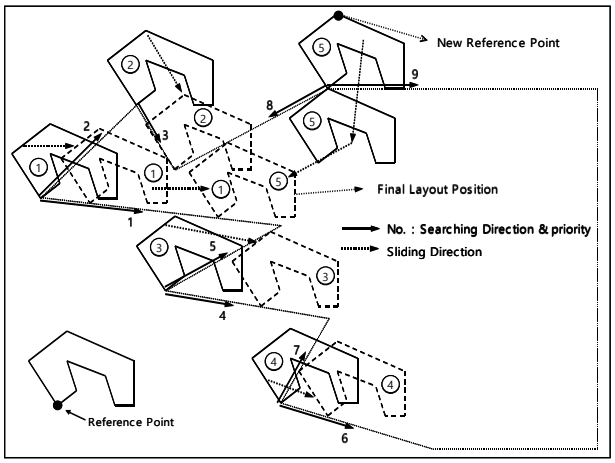


Fig. 10. Heuristic sliding of a part

The part has the priority of sliding order according to the following geometric rules. First, the part achieves sliding at most nearest vertex (part ① in Fig. 10) to global x-coordinate among suitable vertices of resource plate. The possible sliding directions here are 1 and 2, and direction 1 is prior for sliding, because the direction 1 is smaller than 2 to y-coordinate value. In this case, since an overlap happens due to geometrical shape of resource plate after sliding at this vertex, and at the same time this direction 1 has a longer distance for more sliding, the part has to do one more sliding along same direction 1. And then, the part will be located on a virtual layout space. The part ② achieves sliding at vertex that has second smallest x-coordinate value, but the overlap happens. By these iteration searches, parts ①, ③, ④, and ⑤ achieve a perfect sliding, and can be placed on a virtual layout space. Since part ⑤ has the x and y-directional sliding at starting vertex to find a layout space, the reference point of this part ⑤ is changed. The sliding along with direction 1 of part ①, which has the most smallest x-coordinate to the others, is decided as a final result for layout.

Consequently, if the boundary edge of part moving has bigger value than the boundary edge of resource plate, the sliding stops and the calculation is terminated. Since the part only travels on the necessary boundary edges of resource plate, the computation time can be reduced.

Before the actual nesting process, the system calculates the area of the parts and virtual layout spaces on the resource plate. After above step, the parts will be sorted by descending order, and virtual layout spaces of resource plate by ascending order. It selects a part with the largest area and a virtual layout space of resource plate with the smallest area (if not unusable space), and sets the orientation of the part at the possible sliding position. And then, the system calculates the longest distance (d) with same parallel direction along sliding side $(\overline{ab})$ in the overlapped area as shown in Fig. 11. In order to consider a tolerance (cutting gap), the parts are expanded and simplified first, and then placed during sliding process. The most left and then most bottom layout as a general rule is implemented in nesting system.

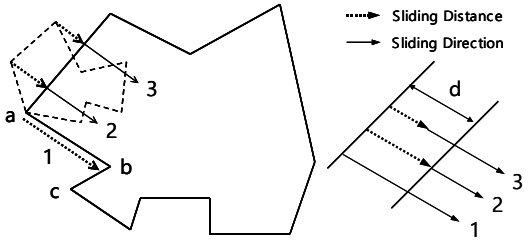


Fig. 11. Determination of movable distance for sliding

4.3 Identification of pivoting point

In order to find and calculate the boundary of the overlap region, it first has to identify whether the pivoting point of the placed part is inside or outside of moving part. This is a quite simple logic and can be identified using the following geometrical theory. The vector angles between a pivoting point and each vertices of moving part can be calculated. Here, the standard coordinate is absolute coordinate. The next step is to calculate angles, which are difference between starting and ending angle, of all entities vectors. Therefore, it can be calculated an accumulative angle of all entities vectors. The accumulative formulation to Hamiltonian cycling vector angle between all entities is

$$\beta = \sum_{i=1}^{n-1} \beta_i + \beta_n \tag{1}$$

where

n is the total number of vertices

$\beta_i = (\alpha_{v_{i+1}} - \alpha_{v_i})$, $\beta_n = (\alpha_{v_1} - \alpha_{v_n})$

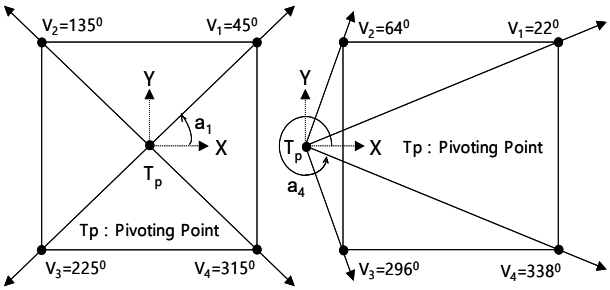
α_{v_i} is an angle between pivoting point and *i-th* vertex of moving part

If the value (β) of accumulative angle of all entities vectors is zero degree (0°), the pivoting point is outside of part, and 360° for inside of part. There are three types of constraint conditions to be considered in equation (1). For the convenience of understanding, simple examples are depicted in Fig. 12. The constraint conditions are

If $\alpha_{v_i} \geq 180°$ then $\alpha_{v_i} = \alpha_{v_i} - 360°$

If $-180° < \alpha_{v_i} < 180°$ then $\alpha_{v_i} = \alpha_{v_i}$

If $\alpha_{v_i} \leq -180°$ then $\alpha_{v_i} = 360° - \alpha_{v_i}$



(a) Case: Inner point **(b)** Case: Outer point

Case: Inner point		
Vector variable	Each vector angle	Vector angle
$V_2 - V_1$	135° – 45° = 90°	90°
$V_3 - V_2$	225° – 135° = 90°	180°
$V_4 - V_3$	315° – 225° = 90°	270°
$V_1 - V_4$	45° – 315° = –270° ⇒ 360 ° – 270° = 90°	**360°**
Case: Outer point		
Vector variable	Each vector angle	Vector angle
$V_2 - V_1$	64° – 22° = 42°	42°
$V_3 - V_2$	296° – 64° = 232° ⇒ 232° – 360° = –128°	–86°
$V_4 - V_3$	338° – 296° = 42°	–44°
$V_1 - V_4$	22° – 338° = –316° ⇒ 360° – 316° = 44°	**0°**

(c) Vector angles for each entity

Fig. 12. Illustration for identification of pivoting point

4.4 Computation of overlap area

An approach, vector graphic representation, in this paper is suggested to find the boundary of the overlap region and then computing its area. This approach considers the combination of the possible overlap areas between two or more shapes. In order to find the boundary of the overlap area, we first have to find all the intersection vertices between the shapes. The next step is to identify the boundary described by line segments that form the boundaries of the overlap region. This procedure is achieved by using "*finding minimal polygons*" of step 3 in polygonal crippling algorithm. Having found all the closed areas that constitute the overlap, their areas are computed using Green's theorem[13].

The boundary of each two-dimensional shape can be described by an ordered list of vertices that are connected by polyline. For overlap area of shape described by line segments with vertices, $\{(x_i, y_i); i = 1, \cdots, m\}$, the formulation is

$$\text{Overlap Area} = \frac{1}{2} \cdot \sum_{i=1}^{m} \{x_i \times (y_{i+1} - y_i) - y_i \times (x_{i+1} - x_i)\} \quad (2)$$

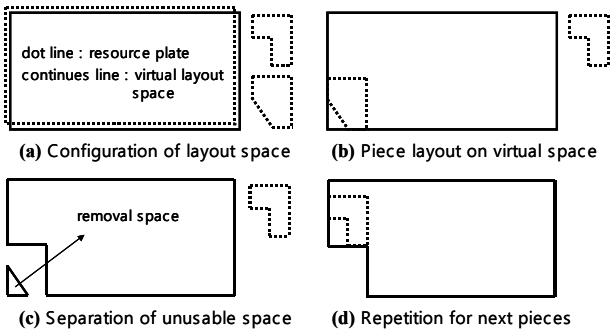
where x_i and y_i are x- and y-coordinate of *i-th* vertex, respectively, and m is total number of vertices.

4.5 Resource plate clipping using virtual memory

The resource plate clipping using virtual memory, which is very simple idea, is applied for rapid nesting system. This technique is an algorithm that whenever each of parts is placed, a new stock boundary of the resource plate is updated for layout space of the next part to reduce geometric conditions of the resource plate and overlap computation time.

This technique is explained by following procedure:

(1) First, make two memory structures (real and virtual) for layout space on resource plate (Fig. 13(a)).
(2) Select part to be nested on the resource plate, and then the part is placed using sliding function on virtual layout space (Fig. 13(b)).
(3) Calculate virtual spaces (areas) of the resource plate updated after the part is placed, and then separate virtual spaces. If there is a removal space, the space is skipped for next part to be nested (Fig. 13(c)).
(4) Repeat above steps for next parts (Fig. 13(d)).



(a) Configuration of layout space **(b)** Piece layout on virtual space

(c) Separation of unusable space **(d)** Repetition for next pieces

Fig. 13. Resource plate clipping using virtual memory

This technique is only method to shorten computation

time, and the real shape of the resource plate is shown on the screen.

5 Evaluation of objective function

The main objective of nesting is to minimise the engineering scrap derived from the area occupied by two-dimensional shapes. Since the sum over the area of the parts is a constant, we do not need to take account of them in objective function, and the scrap contribution is just up to the area occupied on the resource plate. The objective function is a workable length generated due to column layout boundary considering column of part placed to most right-side hand on resource plate shown in Fig. 14. The workable length is derived from an occupied length on the resource plate as a remnant length.

It is convenient to define a nesting efficiency parameter (η) as the non-dimensional ratio between the plate length and workable length. The nesting efficiency parameter (η), as an utilization ratio of resource plate, is very simple and given by the following equation:

$$\eta = (W_L \times 100) / P_L \tag{3}$$

where P_L is the length of resource plate and W_L is the workable length.

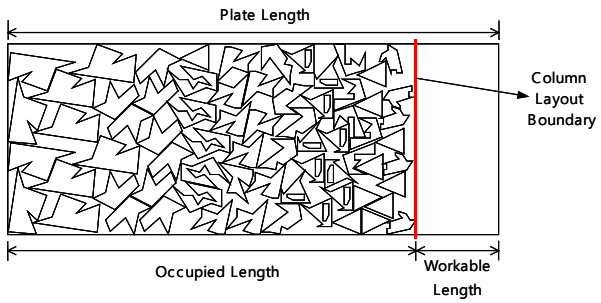


Fig. 14. Column layout boundary for objective function

6 Rule-based heuristic approach

The automatic nesting using a rule-based heuristic approach carries out fully at unsurpassed speed without any manual intervention, and also parts are even nested inside internal cut-outs. In the rule-based heuristic method, a set of rules is designed to try to take advantage of some characteristics of the shapes of the parts, placing earliest those with certain characteristics, pairing together parts with certain clustering features, etc. As it were, final arrangement is, in general, dependent on the placement configuration of the larger parts. Since the smaller parts can be located among the larger ones, they cannot affect the overall arrangement.

Before nesting process, the parts have to be sorted by area size from large to small. This sorting is to determine the sequence order of placing, and then the larger parts are placed first. After one part is placed, the next part to be placed should be the largest one in the unplaced parts list that can be positioned within the remaining region for placing. Therefore, the final layout of nesting is generally dependent on the placement configuration of the larger parts. The smaller ones cannot affect the overall layout because they can be located among the larger ones. The nesting procedure automatically achieves the arrangement of the parts in the selected task as clicking the icon of the parametric macros developed on the Auto-CAD system[11].

In practical industry, a tolerance and prescribed gap should be allowed considering the loss of the resource plate. The tolerance, which is a cutting gap as offset value, is only considered for kerf-width generating during cutting process in this paper.

The heuristic nesting system based on this rules achieves an automatic two-dimensional placement using "part geometric redesign techniques" and "efficient layout strategies". Figure 15 is a skeleton of the rule-based heuristic nesting system.

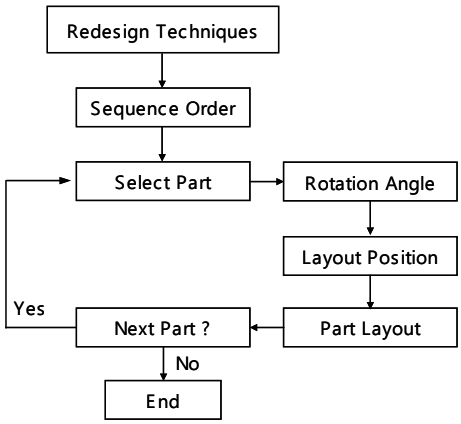
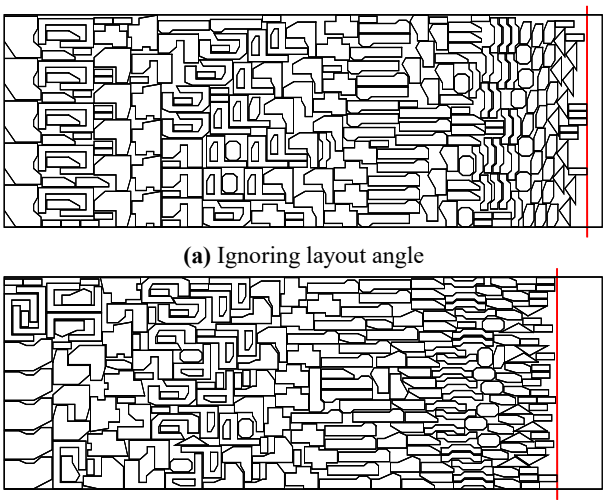


Fig. 15. Skeleton of the rule-based heuristic nesting

7 Numerical results

The layout performances using the rule-based heuristic approach are demonstrated with several nesting examples. All the examples here are performed on a personal computer with a 500 MHz Pentium CPU and 192 MB RAM. Figures 16 and 17 are the results of nesting with regular and irregular shapes generated from the rule-based heuristic approach, respectively. Figure 18 is the stripping layout of parts with irregular shapes. The stripping layout is the sequential placement procedure that the parts with same or very similar shape are

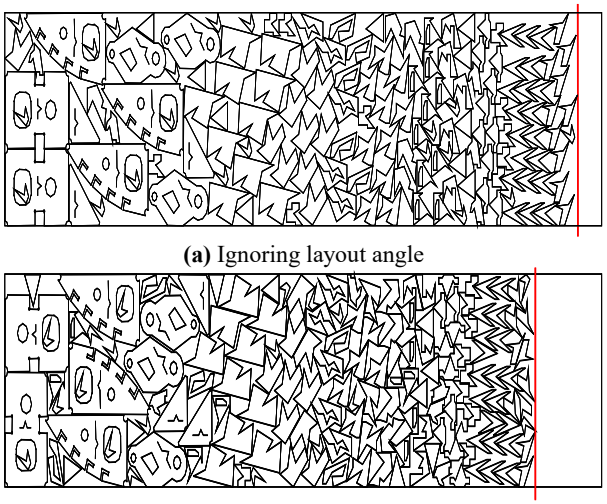
efficiently placed onto resource plate. For efficient layout of stripping nesting, the layout angle of parts is fixed at zero degree.



(a) Ignoring layout angle

(b) Considering layout angle

Fig. 16. Rule-based heuristic layout with regular shapes



(a) Ignoring layout angle

(b) Considering layout angle

Fig. 17. Rule-based heuristic layout with irregular shapes

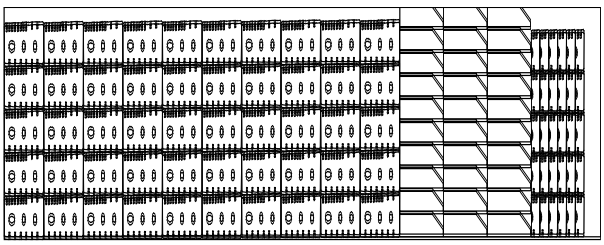


Fig. 18. Stripping layout of parts with irregular shape

Table 1 is the numerical results obtained from the rule-based heuristic approach for nesting with above examples. The offset value, which is the gap between two neighbouring parts, is adopted as 20*mm* for layout efficiency.

Table 1. Numerical results of the rule-based heuristic nesting

Shape type	No. of parts	Layout angle	Running time (Second)	Utilisation ratio (%)
Regular shapes	220	Ignoring	15.93	12.997
		Considering	57.56	17.445
Irregular shapes	156	Ignoring	53.48	7.522
		Considering	111.72	14.31
Stripping	110	0°	3.83	2.8

8 Concluding remarks

The proposed geometric redesign techniques for two-dimensional nesting problem can be used to simplify the shape identification algorithm and reduces the amount of processed data and run time, and consequently turns out to be reasonable in the considered application. Therefore, these techniques can easily handle parts and resource plate with widely varying geometrical shape.

The efficient layout strategies were also stated to achieve an effective nesting within fast computational time frame. The polygon clipping algorithm using virtual memory was a technique to reduce geometric conditions of the resource plate and overlap computation time. Moreover, since the part layout method due to travelling sliding needs a lot of times to find near-optimum vertex for layout, a heuristic sliding technique was suggested for the more efficient and rapid layout of nesting. This heuristic sliding plays a dominant role in speeding up the nesting process. Also, the identification method of pivoting point was suggested to calculate the boundary of the overlap region in fast time frame.

From the results of Table 1, the nesting using the rule-based heuristic approach considering layout angles has shown that computational time and total memory requirements are quite reasonable for practical problems, and the layout efficiency is also outstanding. Therefore, this nesting system is an effective tool for solving large size template layout problems. Moreover, the stripping nesting shows a good layout result.

Consequently, the nesting system using rule-based heuristic approach can be desirable in layout efficiency considering computational time for practical nesting problems, and proposed as a real-time layout simulation method in the industrial field.

References

1. Rinnooy Kan AHG, Timmer GT (1989) Global optimization, Handbooks in Operations Research and Management Science: Vol. I: Optimization, Nemhauser GL, Rinnooy Kan AHG, Todd MJ eds., North-Holland, Amsterdam, pp. 631-662.
2. Jain P, Fenyes P, Richter R (1992) Optimal Blank Nesting Using Simulated Annealing. Transaction of the ASME, J. of Mechanical Design, 114:160-165.
3. Han GC, Na SJ (1996) Two-Stage Approach for Nesting in Two-Dimensional Cutting Problems using Neural Network and Simulated Annealing. Proc. Instn. Mech. Engrs., Part B : J. of Engineering Manufacture, 210:509-519.
4. Fujita K, Akagi S, Hirokawa N (1993) Hybrid Approach for Optimal Nesting using a Genetic Algorithm and a Local Minimization Algorithm. In Proceedings of the 19th Annual ASME Design Automation Conference, 1:477-484, Albuquerque, NM, 19-22 September 1993, ASME, New York. (EI Jan 94) 91.
5. Doi K (1997) Automatic Nesting System by Use of Genetic Algorithm. ICCAS 1997, pp. 247-258.
6. Sha OP, Kumar R (2000) Nesting of Two-Dimensional Irregular Parts Within an Irregular Boundary Using Genetic Algorithms. Journal of Ship Production, 1680:232-242.
7. Ismail HS, Hon KKB (1995) The Nesting of Two-Dimensional Shapes Using Genetic Algorithms. Proc. Instn. Mech. Engrs. 209:115-124.
8. Ono T, Watanabe G (1997) Genetic Algorithms for Optimal Cutting. Evolutionary Algorithms in Engineering Applications, Dasgupta D, Michalewicz Z eds., Springer Inc., pp. 515-530.
9. Babu AR, Babu NR (1999) Effective Nesting of Rectangular Parts in Multiple Rectangular Sheets Using Genetic and Heuristic Algorithms. International Journal of Production Research, 37(7):1625-1643.
10. Autodesk, Objectarx developer's guide, http://usa.autodesk.com/adsk/item, Autodesk, Inc.
11. Kim Y, Park J (2000) Parametric Macro for Two-Dimensional Layout on the Auto-CAD System. Journal of Ship & Ocean Technology (SOTECH), 4(3):13-20.
12. Schutte K, (1995) An edge labeling approach to concave polygon clipping. ACM Transactions on Graphics, 7.
13. Eberhardt AC (1987) Calculating Precise Cross-Sectional Properties for Complex Geometries. Computers in Mechanical Engineering, pp. 32-37.