International Journal of Computer Integrated Manufacturing, Vol. 20, No. 8, December 2007, 741 – 756

Nesting of two-dimensional irregular parts: an integrated approach

S. Q. XIE*†, G. G. WANG† and Y. LIU‡

†Department of Mechanical Engineering, University of Auckland, Private Bag 92019,
Auckland, New Zealand
‡Department of Industrial and Manufacturing Engineering, The University of Manitoba, Canada

The present paper reports an intelligent computer-aided nesting (CAN) system for optimal nesting of two-dimensional parts, especially parts with complicated shapes, with the objective of effectively improving the utilization ratio of sheet materials. This paper also systemically reviews the nesting algorithms that were developed to perform various nesting tasks, and attacks the irregular part nesting problem by efficiently integrating and improving the performance of nesting algorithms such as the rectangular enclosure method, bottom-left nesting algorithms, heuristic algorithms and genetic algorithms. The CAN system has also been developed as a nesting algorithm test platform for researching and developing new nesting algorithms. Through this test platform, the limitations of existing nesting algorithms are investigated and problems such as nesting parts in spaces within a single part or between parts are also studied. Efforts have been devoted to improving the nesting efficiency of the existing algorithms and developing new nesting algorithms. Case studies are carried out in a sheet metal cutting company. The results show that the intelligent CAN system can effectively nest both regular and irregular parts, and greatly improve the utilization ratio of raw sheet material.

Keywords: Nesting algorithms; Irregular parts; No fit polygon (NFP); Heuristic algorithms

1. Introduction

In today's competitive manufacturing environment, manufacturing industries are working hard to improve productivity, as well as reduce cost and lead-time. For industries such as the sheet metal industry, cloth industry, glass and ship building industry, their manufacturing operations often involve cutting new two-dimensional (2D) parts from batches of sheet materials. An important issue faced by these industries is how to find the optimum layout of 2D parts with different shape and size within the available sheet, such that the material utilization can be maximized and the resulting material wastage can be minimized.

The irregular part nesting problem is a 2D cutting and packing problem where the small pieces to cut have irregular shapes. The main objective is to find a minimum-waste arrangement of the parts to be cut without overlap. The nesting irregular parts problem is characterized by the intrinsic difficulty of dealing with geometry, satisfaction of the no-overlapping and containment constraints, and complex computation. Currently, there are still lack of practical algorithms in industry to nest complex and multiplex parts, which impedes the realization of effective automatic nesting. Previous methods have shown good nesting results for nesting rectangular parts on one single sheet (Balas and Perregaard 2002, Shang and Song 2003). A few nesting algorithms have also shown promising results for nesting

---

*Corresponding author. Email: s.xie@auckland.ac.nz

International Journal of Computer Integrated Manufacturing
ISSN 0951-192X print/ISSN 1362-3052 online
http://www.tandf.co.uk/journals
DOI: 10.1080/09511920600996401

complicated parts (Cheng *et al.* 1999, 2000, Tay *et al.* 2002). However, to our knowledge, there is not a single nesting algorithm that can solve the complicated part nesting problem and be able to cope with various nesting requirements.

The present paper approaches the nesting problem from a different angle after systemically investigating the nesting algorithms that have been developed. Instead of using a single nesting algorithm, this research addresses the nesting problem by an integrated approach, where various nesting algorithms are investigated, improved and finally integrated. An intelligent computer-aided nesting (CAN) system is developed as a result of the above research.

2. Literature review

Tackling 2D nesting problems should consider the cutting processes, type of parts and shape and size of both parts and sheets. In the last few decades, different algorithms were proposed for nesting of parts of different shapes in sheets of different types. This section systemically reviews the nesting algorithms that were proposed to achieve different nesting tasks. The limitations of the algorithms are also reviewed. The nesting algorithms to be reviewed include rectangular parts nesting algorithms, enclosure algorithms, heuristic algorithms, pairwise and cluster algorithm, genetic algorithms, bottom-left (BL) nesting algorithms and hybrid algorithms.

2.1. *Rectangular parts nesting algorithms*

Most practical algorithms in early stages only focus on nesting simple rectangular parts. The method of cutting rectangular parts from one edge of the rectangle to the opposite edge, which is generally known as the 'guillotine cutting' (MacLeod *et al.* 1993). Several methods have been proposed to find an optimal solution to this problem. Roth *et al.* (1985) have developed methods to solve the rectangular cutting stock problem by using linear programming. Arvanitis (1999) proposed a recursive dynamic method, which is a kind of heuristic algorithm for an n-stage unconstrained guillotine cutting stock problem. Balas and Perregaard (2002) developed a tree search procedure in conjunction with a 0-1 integer programming method for solving the orthogonal non-guillotine cutting stock problem. Other researchers have efficiently combined heuristic algorithms, tree search methods and transportation algorithms to improve the utilization ratio of rectangular parts (Caroe and Tind 1997, Kaufmann 1990). However, most of these attempts do not work with columns of pieces of similar width and include features such as filling holes behind the current front face of the layout, and are restricted to rectangular parts and have difficulties for applications where irregular parts are involved. Also, the mathematical programming techniques that have been used for solving the rectangular cutting stock problem typically have high computational complexity (Bhadury and Chandrasekaran 1996, MacLeod *et al.* 1993, Martini and Soltan 1998).

2.2. *Enclosure algorithms*

The enclosure algorithms basically include three steps: embedding graphics in an enclosure, nesting parts and resuming parts. For the nesting of irregular parts, two types of enclosure algorithms, rectangular enclosure and hexagonal enclosure algorithms, were widely used (Grinde and Cavalier 1995, Cheng and Rao 2000). Traditionally, rectangular enclosure and hexagonal enclosure algorithms are widely used for generating this type of nesting (Cheng and Rao 2000). They can nest a great number of parts at a high speed. Unfortunately, these algorithms often lead to low material utilization ratio, or nesting efficiency. Traditional CAN algorithms always try to find minimum convex enclosures to approximate parts (Freeman and Shapira 1975). This minimum convex enclosure problem has been addressed in many articles. Placing irregular shaped parts into rectangular enclosures have been studied (Peach 1988). These rectangular enclosures are subsequently arranged on the sheet using heuristics. For a single polygon, all orientations that enable a side of the polygon to be parallel to the horizontal or vertical axis are found. For each orientation, the rectangular enclosure is calculated. There is an algorithm presented as inputting an arbitrarily closed curve and producing the minimum-area rectangular enclosure (Roth *et al.* 1985). The main result is that the rectangle of minimum area enclosing a convex polygon has at least one side collinear with one of the edges of the polygon (Grinde and Cavalier 1995).

The minimal-area rectangular enclosure algorithm was limited because it did not find the minimal-area convex enclosure. The minimal-area convex enclosure will lead to less waste for the same pair of enclosed features than the minimal-area rectangular enclosure. The minimal-area convex enclosure for two convex polygons (Lee and Woo 1988) was limited because it considered only convex polygons as inputs.

2.3. *Heuristic nesting algorithms*

There are different interpretations of heuristic algorithms in different fields. When used for nesting, heuristic algorithms work by generating different part orders guided by heuristics or in random to find the optimal. In heuristic algorithms, parts are often selectively paired or clustered by utilizing translations, insertion and rotations (Dumitrescu 1998). This process is referred as pairwising or clustering. For fixed orientations, they use the vertices of the NFP (no-fit polygon) of one shape with respect to the other as the basis of the search. Takahara *et al.* 2001 extended this concept by comparing each vertex to see if the enclosure can be made smaller. Clustering more than two shapes is more difficult because the ordering of clustering could influence the quality of solution. A simplified heuristic algorithm was used for multistage clustering (Cheng and Rao 2000). This algorithm only focuses on the limited nesting possibilities of part clusters.

Nesting two convex polygons was considered by Lee *et al.* (1988) and Wu (2001). The goal was to find the convex enclosure of minimal area containing the two parts. A fixed number of orientations were allowed, and with this restriction, the algorithms run in linear time to the total number of edges of the two polygons. Basically, one part was moved around the boundary of the other, and the area added to form the convex enclosure was maintained. The optimal position of two parts was acquired when the area was minimum. However, the algorithm works for convex parts. Nee *et al.* (1986) nested repetitive parts and grouped pairwise to form small nesting modules. These modules were further clustered using a rectangular nesting routine. Based on partial enumeration, simplified heuristics were utilized to develop algorithms for nesting of similar parts (Nee *et al.* 1986 and Toh *et al.* 1995).

2.4. Bottom-left nesting algorithms

The BL nesting algorithm known as the graphics are nested according to the column. After a column is fully nested so that no more graphics can be inserted, the nesting moves to the next column. If there are several parts that have similar width, the algorithm grants the nesting priority to them (Dowsland *et al.* 2002, Nandy and Bhattacharya 2003, Roy and Dasari 1998).

Early implementations normally involved one or more orders based on the dimensions of the parts, or a random sample of orders from which the best solution was chosen. (Dowsland *et al.* 2002, Nandy and Bhattacharya 2003, Roy and Dasari 1998). The advantages of this type of approach were its speed and simplicity when compared with more sophisticated methods that might be able to produce solutions with higher quality. As a result there were still many commercial environments where such single-pass placement policies were appropriate. Moreover in recent years interest has been boosted by implementations of modern heuristics, such as tabu search or genetic algorithms, which use a BL nesting policy as the basis of cost evaluation (Chen *et al.* 2003, Hopper and Turton 2001, Ramesh Babu and Ramesh Babu 2001). Usually, the BL nesting algorithms have to work with other searching algorithms and the performances of the algorithms depend very much on the searching algorithms.

2.5. Space searching algorithms

There are two important steps in nesting a part in an empty space between parts or a space within a single part, including finding the space and filling the space. Once the space is found, the algorithms involved in the second step are the same as nesting algorithms on irregular sheets. Heckmann and Lengauer (1998) presented an algorithm to solve the problems of nesting in spaces. They focused on selective data reduction, sequential part placement, and a topological part placement process. A set of intelligent nesting algorithms were exploited which imported the algorithms for irregular-shape-sheet stamping industry to nesting parts in spaces within a part (Mishra *et al.* 1999; Murty *et al.* 1983). Regarding the space area between parts and among a single part as an irregular sheet, the following research is also significant. Nee *et al.* (1986) indicated a graphics-based approach for part nesting on irregular sheets with internal bad patches. The parts were reported using a graphics routine and polygonalized. Probabilistic approaches had also been attempted to solve the irregular cutting stock problem. To find the final optimal placement, these algorithms had the common process producing one or more random placements of parts and using a repeating process on the sheet. However, for irregular parts, there are still lack of efficient space searching algorithms that can allocate empty spaces accurately among irregular parts.

2.6. Hybrid algorithms

The algorithms mentioned above can address some possible nesting problems and present relevant solutions. A comprehensive solution is, according to specific problems, to combine the advantages of nesting algorithms as hybrid algorithms. The irregular part layout problem has received increasing attention. Many articles as the theoretical supports of this algorithm are published (Alves *et al.* 1999; Lutfiyya *et al.* 1992, Dufour *et al.* 1997). All efforts on this problem to date are of a heuristic nature. It is possible to formulate the problem as a mathematical program (Grinde and Cavalier 1996). However, the problem is very difficult to solve optimally owing to the computational time required.

Targeting at a larger nesting scale, many genetic and heuristic algorithms show their advantages through translating, inserting or rotating methods (Hopper and Turton 1999; Hopper and Turton 2001). However, owing to their high computation demand, present CAN tool developers paid very little attention to these algorithms. Using current high-speed computers to solve calculation problems, the combination of genetic and heuristic algorithms can achieve practical and optimal nesting results.

As reviewed above, the nesting of irregular parts is a complicated research issue and involves nesting algorithms of various types. There is no single algorithm that could be used to solve the irregular parts nesting problems alone. Hence, the current paper investigates an integrated or hybrid approach to the irregular parts nesting problem. Nesting algorithms such as the rectangular enclosure method, BL nesting algorithms, heuristic algorithms, and genetic algorithms, are investigated, modified and efficiently integrated. New nesting algorithms are also proposed for nesting various types of parts. An intelligent CAN system is developed and used in a sheet metal cutting and a cloth cutting company in New Zealand.

3. The intelligent CAN system

This work presents an intelligent CAN system to support the integration of different nesting algorithms for nesting parts of irregular shape. The main structure of the intelligent CAN system is shown in figure 1. A standard interface is developed to convert CAD drawings or graphic information into the CAN system. One can then apply nesting algorithms and translate the nesting results into G-Codes for computer aided manufacturing (CAM) systems.

The intelligent CAN system efficiently integrates nesting algorithms with the purpose of utilizing sheet material efficiently. The nesting algorithms to be studied and integrated include the rectangular enclosure method, BL nesting algorithms, heuristic algorithms, and genetic algorithms. In particular, the algorithms are to maximize the amount of the parts to be nested or, to reduce waste in a large sheet with finite dimensions. The intelligent CAN mainly includes the following three components, which will be further described in this paper.

1. Foundations of CAN. The foundations for supporting the integration of nesting algorithms include the following modules: definition of basic regions, description of parts in graphical regions, internal and external contours, NFP, and definition of positions and region.
2. Investigating and integrating nesting algorithms for optimal nesting 2D irregular parts. As nesting algorithms are the key for the intelligent CAN system, there are various nesting algorithms being developed and integrated into the CAN system, such as an improved BL algorithm, a heuristic nesting algorithm and clustering algorithms. The hybrid algorithms are classified into three categories based on the quantity, size and shape of parts.
3. Integrating CAN with computer aided design (CAD), computer aided process planning (CAPP) and CAM systems. This mainly includes the development of interfaces, modules, and computer programs to integrate the CAN system with CAD, CAPP and CAM sub-systems. This includes a data interface with CAD, an interface with CAPP/CAM and a graphics and data exchange module.

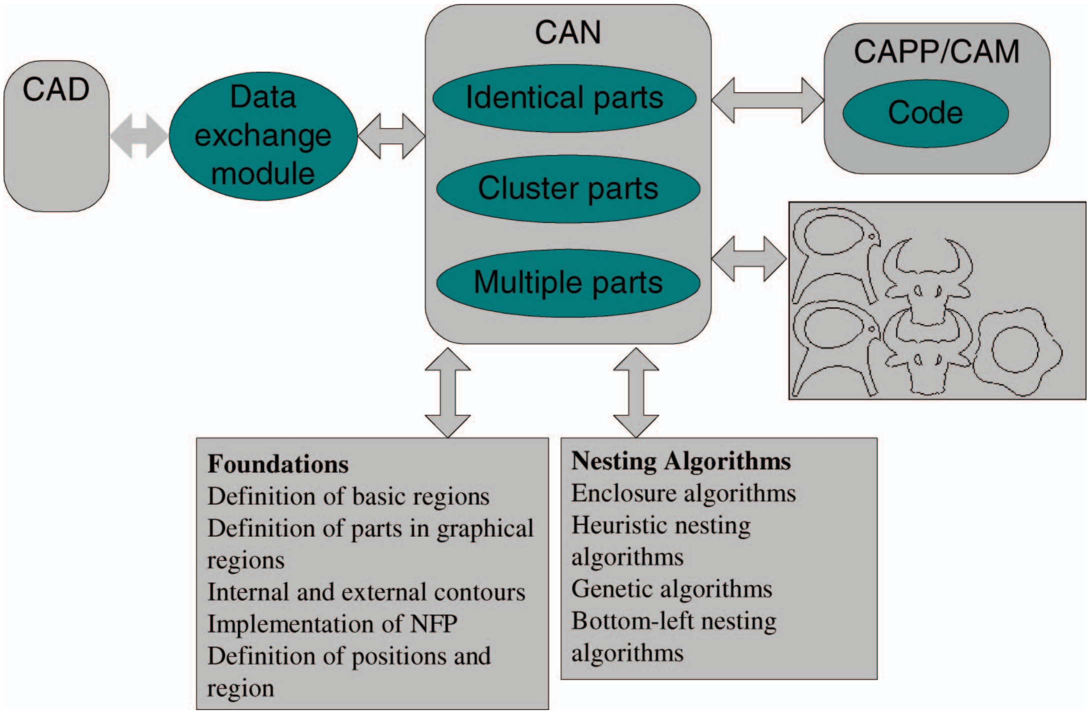


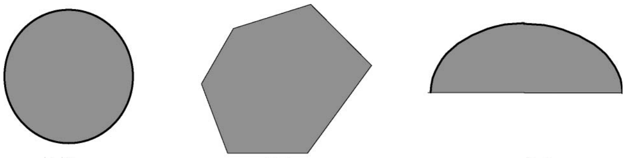
Figure 1. Structure of the intelligent CAN system.

4. Foundations to facilitate CAN

In any cutting or packing problem the layout geometric feasibility has to be tackled, i.e. pieces must not overlap each other and have to be completely within the sheet to be cut. In nesting problems, this geometric problem is particularly difficult owing to the irregularity and non-convexity of pieces to be placed. This section briefly discusses the modules that are developed to solve the abovementioned problems and provide foundations for supporting nesting and optimization. They include: definition of basic regions, description of parts in graphical regions, internal and external contours and NFP.

4.1. Definition of basic graphical regions

There are three basic shapes defined in this research as shown in figure 2. They include: (a) the circle region; (b) the polygon region; (c) the region formed by an arc and its opposite chord. These basic regions are modelled using an object-oriented methodology.



(a) (b) (c)

Figure 2. Three basic regions: (a) circle region; (b) polygon region; (c) region formed by an arc and the corresponding chord (arc-chord region).

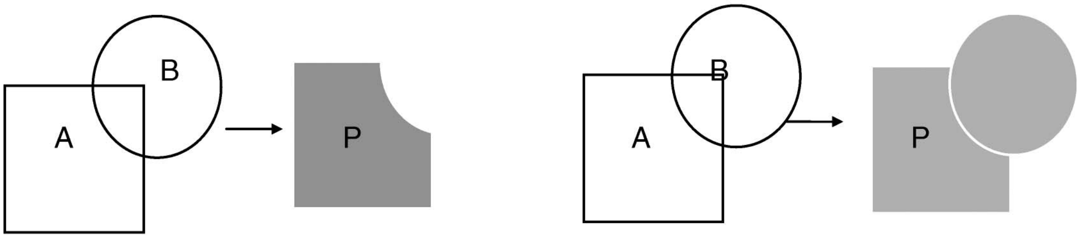
4.2. Description of graphical regions

A computer uses Boolean operations to describe any complex graphical region. Some definitions are given: (A and B are defined regions; P is the unknown region)

> The merger of regions (RGN_OR is represented with $\cup$)
> $P = A \cup B = \{P: P \in A \text{ OR } P \in B\}$ [figure 3(a)]
> The intersection of regions (RGN_AND is represented with $\cap$)
> $P = A \cap B = \{P: P \in A \text{ AND } P \in B\}$ [figure 3(b)]
> The subtraction of regions (RGN_DiFF is represented with -)
> $P = A\text{-}B = \{P: P \in A \text{ AND } P \in B\}$
> $P = B\text{-}A = \{P: P \in B \text{ AND } P \in A\}$ [figure 3(c)]
> The logic Exclusive OR of regions (RGN_XOR is represented with$\oplus$)
> $P = A \bullet B = (A \cup B)\text{-}(A \cap B)$ [figure 3(d)]

Normally, a graphic region includes not only convex arcs but also concave arcs. The subtracting operation is used in the concave arc area; while the adding operation is used in the convex arc area. The following example shows how a part is made up of the Boolean operation of the basic regions. As shown in figure 4(a), A is a convex arc and the

(a)A∪B = P (b)A∩B = P



(c)A - B = P (d)A•B = P

Figure 3. Boolean operation of regions.

adding operation should be used; C is a concave arc and the subtracting operation should be used.

Figure 5 shows a graphic representation of a more complex part using the above-mentioned principles. This part is the name badge of a manufacturing company.

4.3. *Internal and external contours*

Empty space searching is an important issue in irregular parts nesting. The empty space may come from either the internal space of a part or spaces among parts. Hence, it is important to identify internal and external contours.

A searching algorithm has been developed to identify internal and external contours. The algorithm starts from randomly drawing a line crossing the part involved at point A and point B. It is assumed that the line AB starts at point A and ends at point B. Three possibilities exist.

1. If line AB crosses the contour line of the part twice as shown in figure 6(a), there is no empty space in the area line AB crossed.
2. If line A crosses the contour line of the part more than twice and the number of times that the line AB crosses the contour lines is odd [as shown in figure 6(b)], it is then regarded that line AC, DE and FB lie in the internal area of the part.
3. If line A crosses the contour line of the part more than twice and the total number of crossover is even, it can then be concluded that line CD and EF lie in the external area of the space.

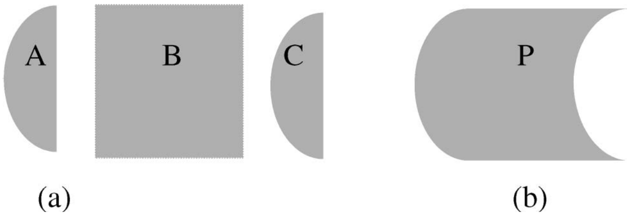


Figure 4. (a) Basic regions A, B and C; (b) region P formed by the Boolean operations of basic regions A, B and C.

Figure 5. Graphical representation of a name badge.

4.4. *No fit polygon*

The NFP determines all arrangements that two arbitrary polygons may take such that the polygons touch but do not overlap. Figure 7 illustrates this concept for two parts A and B. The contour of polygon NFP_{AB} is the trace of point R_B when part B slide around part A. As the two parts are not contacting the point $R_B = u - v$ lies well outside NFP_{AB}. Point R_B can be moved a distance D_x in the horizontal direction before it contacts NFP_{AB}. This is reflected in the fact that part B could be moved the same distance before contacting part A. A more detailed examination of the figure reveals that if point R_B is moved to a point inside the NFP. The translation of part B by the same distance and direction will lead to the overlapping between two parts.

In the case of a bottom-left nesting, there are basic principles to determine NFP of two parts. According to the above analysis, the following principles are drawn:

(a) if the reference point of part B is placed in the interior of NFP_{AB}, then B intersects A;
(b) if the reference point of part B is placed on the boundary of NFP_{AB}, then B contacts A;

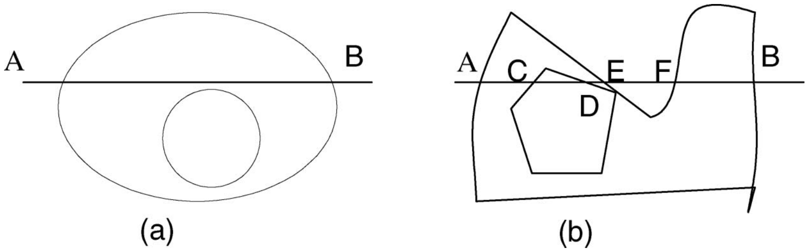


Figure 6. Illustration on how to identify the internal drawings and external contours.

(c) if the reference point of part B is placed in the exterior of NFP_{AB}, then B does not intersect or contact A.

5. Integrated nesting algorithms

As found from the literature survey there is no single nesting method that can solve the irregular part nesting problem alone, this work first categorizes parts to-be-nested into three basic types, and then develops integrated nesting algorithms according to the part type. These three types include:

(a) *identical parts*: parts having the same shape and size, and one is an identical copy of the other;
(b) *collectable parts*: parts of different shapes but can fit together and be sufficiently represented by a number of part clusters of simple shape(normally a rectangle);
(c) *multiple parts*: parts cannot easily be represented by any identical cluster.

Table 1 shows the nesting algorithms that are used to nest the above-mentioned three types of parts. For example, enclosure algorithms and bottom-left nesting algorithms are used to nest identical parts. Besides the three common nesting algorithms used for all three types of parts, there are four more nesting algorithms being used for nesting multiple parts. These include the clustering algorithm, random nesting algorithm, heuristic searching algorithm and the heuristic filling algorithm. Heuristic algorithms, which always try to find the compact neighbourhood between parts through translation, rotation and insertion of parts, have advantages over the bottom-left algorithm. The hybrid algorithms efficiently combine the advantages of these nesting algorithms. The following sections discuss the algorithms that have been developed to nest different types of parts under different situations.

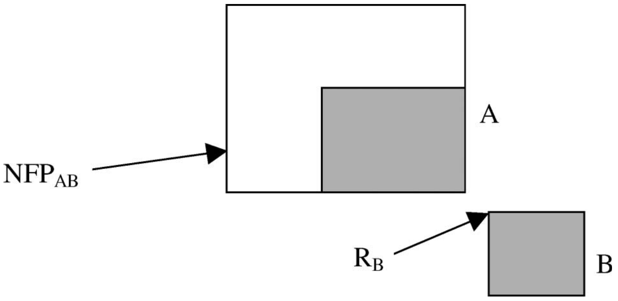


Figure 7. Illustration on no-fit-polygon NFP_{AB}.

Table 1. Algorithm selection for different part types.

	Identical parts	Collectable parts	Multiple parts
Enclosure algorithms	Yes	Yes	Yes
Bottom-left nesting algorithm	Yes	Yes	Yes
Dynamic allocation algorithm	Yes	Yes	Yes
Clustering algorithm		Yes	Yes
Random nesting algorithm			Yes
Heuristic searching algorithm			Yes
Heuristic filling algorithm			Yes

5.1. *Identical parts nesting*

In this research, three nesting methods are proposed and the traditional bottom left nesting algorithm is modified to improve the utilization ratio of identical parts nesting.

The basic principle of identical part nesting is:

1. The part is rotated in 0–180° to find the exact degree with the smallest rectangular enclosure as illustrated by figure 8(b). Assuming the initial orientation as shown in figure 8(a).
2. If the utilization ratio is not able to reach the maximum utilization (Cheng *et al.* 1999) after the rotation, identical parts are pair wised. The pairwise is the combination of a part and its 180° rotated copy.

5.1.1. Single-column and pairwise nesting algorithm. Generally nesting algorithms consist of single-column nesting algorithm and pairwise nesting algorithm as shown in figure 9. Comparing these two methods, the simple single column nesting should be used if it meets the requirement and it has a high utilization ratio. Otherwise, the pairwise nesting should be applied, particularly for parts with similar shape to a triangle. The pairwise is then treated as a single part in the nesting process, the single-column nesting algorithms is the application of the bottom-left algorithms on identical parts. For example, figure 10 shows a layout designed by using the single column nesting algorithm. As defined in section 4, parameters x and y record the positions of part 1. The parameter R_{gn} is to write down the region data of parts in the programme.

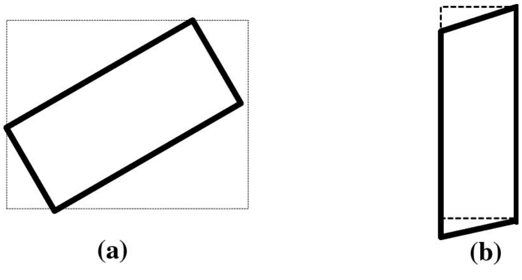


Figure 8. Process of generating a rectangular enclosure. (a) Initialization; (b) after rotation.

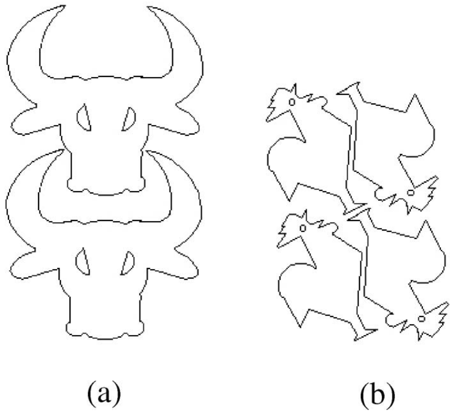


Figure 9. Two basic methods: (a) single-column nesting and (b) pairwise nesting.

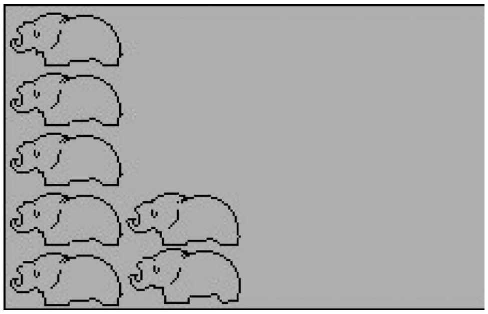


Figure 10. Layout of identical parts.

The parameter d_y is the length of vertical translation. The process of producing a layout of identical parts includes the following main steps:

Step 1: Pick the enclosure of a part as part 1.
Step 2: Copy part 1 as part 2.
Step 3: Allocate part 2 onto part 1.
Step 4: Remove rectangular enclosures.
Step 5: Move part 2 towards part 1 until contacting part 1.
Step 6: Record the displacement of part 2 and final position of two parts.
Step 7: Delete all dynamic parameters.

A traditional BL algorithm can reach a dense and fast layout when it is applied to identical parts. After the BL nesting, normally, there is an empty space between the top boundary of sheet block and the top of a column, which is too small to insert the next ordered rectangular enclosure into. Obviously, this space could mean a low utilization ratio of the layout, and it should be minimized.

The modified algorithm still uses the BL nesting algorithm to nest parts column by column. The first column is chosen as an example. This BL nesting algorithm is changed to allow parts to rotate in the range from 0 to 360°. Once in each nesting layout, the top right corner of one column just contacts the top boundary of the sheet, the software program records the utilization ratio of this layout. The above limiting factor may produce many different layouts; however, the final layout comes from comparing their utilization ratio and then finding the best column. Copy this best column until the required number of parts is reached as shown in figure 11(b). The difference between the modified BL nesting algorithm and the ordinary BL nesting algorithm can be found by comparing figures 11(a) and (b). There is a space between the top boundary of the sheet block and the top of a column as shown in figure 11(a). In figure 11(b), the layout for the identical parts from a manufacturing company is optimized after the rotation. The utilization ratio increases from 66% to 81%.

5.1.2. Dynamic allocation algorithm.

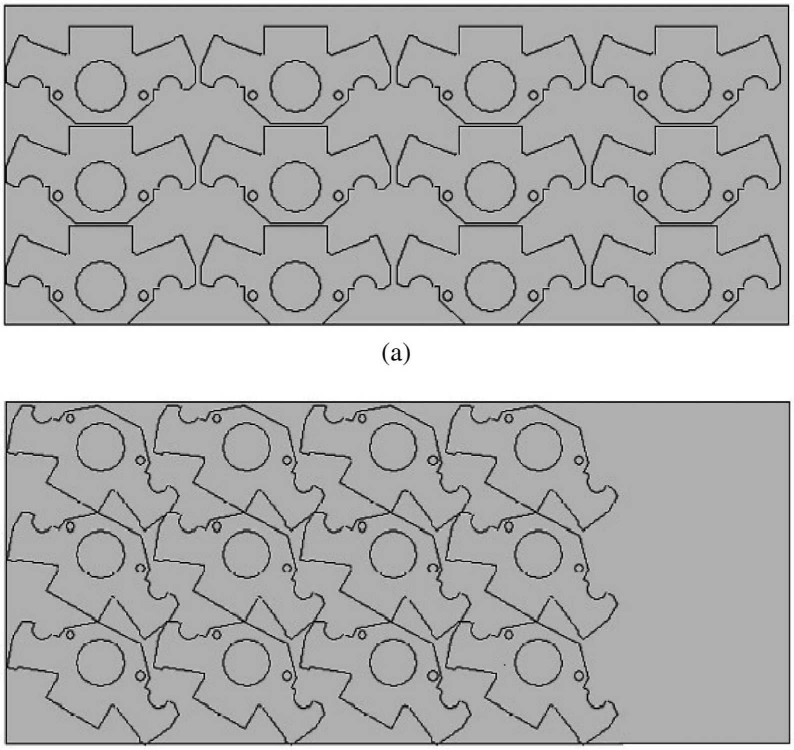
Traditional enclosure algorithms usually produce large gaps after resuming the nested enclosures to real parts. Dynamic allocation algorithm is studied and used in the nesting of all three different types of parts. The purpose of developing this algorithm is to reduce the gap between parts by further moving nested parts. This algorithm includes the following steps.

1. Position part 1 and its rectangular enclosure.
2. Locate the top-left corner of the rectangular enclosure of part 1 at point (x, y).
3. Locate the enclosure of part 2 on (x,y) without overlapping with part 1.
4. Move downward part 2 a step with an adjustable step size without overlapping.
5. Build the second graphic region.

As shown in figure 12, the gap between the two parts 'd' is fine-tuned by the program with a default value of 1 mm. This gap should be determined by the nesting algorithms. Figure 12(b) shows the correct gap between two parts. Comparing with figure 12(a) and (b), the gap in figure 12(a) is greater than (b). Hence, by using dynamic allocation algorithm, more compact pattern as shown in figure 12(b) can be obtained and thus material wastage is reduced.

5.2. *Collectable parts nesting*

In manufacturing processes, there are often parts with different shapes. To improve productivity and increase nesting speed of collectable parts, a new clustering algorithm has been adopted (Cheng and Rao 2000). The clustering algorithm is only engaged when parts have a small variety of shapes and certain product scales between each other. To reduce the complexity of the nesting process, some parts are arranged in certain compacted neighbourhood as a cluster, and then the part clusters are transacted as parts having


(a)

(b)

Figure 11. Comparison of bottom-left nesting algorithms. (a) Ordinary bottom-left nesting algorithm; (b) modified bottom-left nesting algorithm.

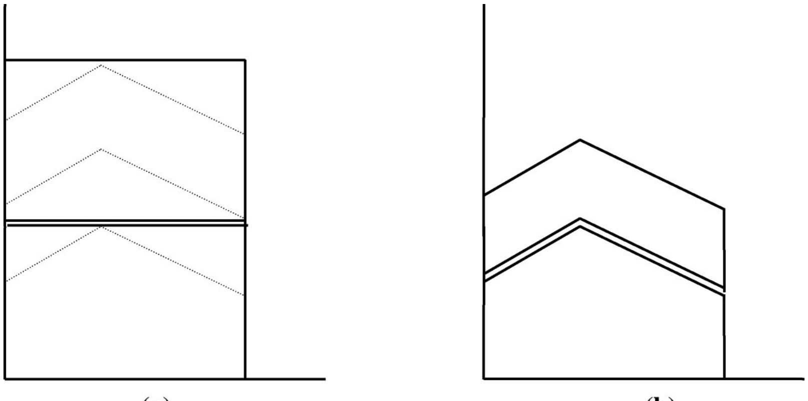

(a) **(b)**

Figure 12. Comparison of two results of algorithms. (a) Result of the enclosure algorithm; (b) result of the dynamic allocation algorithm.

identical shape. In particular, when the number of parts in a cluster is two, a pair of parts is named a pairwise. A sub-problem of the nesting problem is investigated here to efficiently nest collectable parts. When parts are rotated, the algorithmic approach should take advantage of the added flexibility. The heuristic algorithm is imported to produce pairwise or clusters.

Given polygons A and B, the nesting process of a pairwise can be solved as follows: determine if A can be translated and/or rotated to fit inside B; and if so, report such a placement of A and B. For clustering more than two shapes, the heuristic algorithm is more useful. First, setting the number of parts line up with the scale. The heuristic algorithm can easily find the best nesting positions of them under the same functionality as producing a pairwise. The detailed steps of the algorithm are as follows:

1. Sort out the parts into a chronological order.
2. Pair the first two parts.
3. Merge the new pair.
4. Record the positions of two parts.
5. Replicate steps 2 and 3 until finished.
6. Record the utilization ratio according to the various positions.
7. Reorder these parts, then replicate the above steps under all possible combinations.
8. Compare all nesting efficiencies to find the best part cluster.

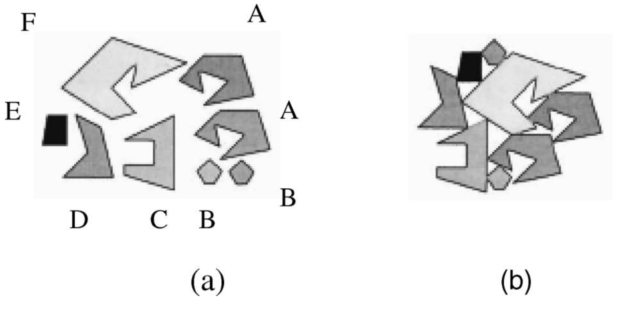
As shown in figure 13(a), there are five parts with different shapes, which can be formed as a part cluster. Figure 13(b) shows a part cluster that is nested by the above heuristic algorithm. To find the best part cluster, the replications in steps 7 and 8 work under two principles: increasing the rectangularity and decreasing the area of the enclosure.

When clustering collectable parts, the utilization ratio will be calculated in real time as shown in step 6. The goal of comparing the utilization ratio in real time is to find a cluster that has the highest utilization ratio. The collectable parts after clustering can then be simplified to an identical parts nesting task. The part cluster is regarded as a single identical part. The nesting algorithms for identical parts are then employed for nesting the part clusters.

5.3. *Multiple parts nesting*

There are three types of algorithms used for nesting multiple parts: optimized bottom-left nesting algorithms, heuristic searching and nesting algorithms and hybrid algorithms.

5.3.1. Bottom-left nesting algorithm. The integration of the BL algorithm and the heuristic searching algorithm has many advantages in nesting identical or similar width parts. However, there are drawbacks in traditional nesting algorithms. For example, for the two parts shown in figure 14, the common nesting results by following the BL method are shown in figure 15(a).



(a) (b)

Figure 13. A part cluster generated by the heuristic algorithm: (a) six parts; (b) part cluster of the six parts (Cheng and Rao 2000).

For the common BL nesting algorithm, there are two possible solutions in early research. Once applying the rectangular enclosure algorithms, the heuristic searching algorithm will find the best way to fill small parts between the big part and the sheet boundary through comparing, inserting and sliding parts. But the complexity of the heuristic searching algorithm requires more time. On the other hand, remounting parts into the sheet has an advantage while utilizing the heuristic algorithms. However the algorithm cannot find the most optimal layout. It always positions the biggest part in the bottom-left corner, and then adds a similar width part on the top of it. This algorithm may present a poor layout as shown in figure 15(a) but the better result is the one shown in figure 15(b). Their nesting efficiencies in the relevant rectangular enclosures are 76% and 91% respectively.

The BL nesting algorithm has been modified in this research. The BL nesting algorithm was the core of the early research in this project. It is only efficient for a group of similar width rectangular parts or parts in rectangular enclosure, but not for more complicated parts. Normally, the width difference is defined less than 10%. The modified algorithm focuses on optimizing a layout column by column, which can easily find the best allocation of parts in each column. The steps of the modified BL nesting algorithm are as follows:

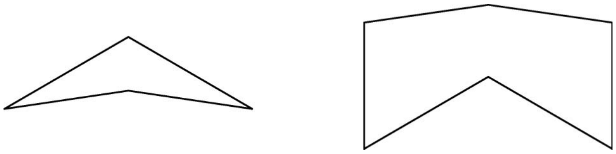
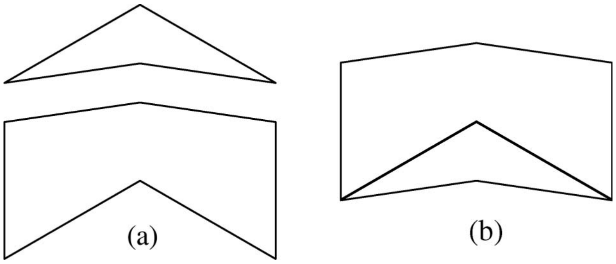


Figure 14. Two sample parts involved in the bottom-left algorithm.



(a) (b)

Figure 15. (a) Layout under early algorithms; (b) optimal layout.

1. Choose parts that have the similar width, from the program records.
2. Settle bigger parts in the bottom-left corner.
3. Allocate parts on the top of the nested part.
4. Moving these parts towards the bottom and left until the rectangular enclosure is minimized.
5. Replicate the above steps until the parts fully fill the column.
6. Remount this column.
7. Re-nest this column in all possible orders.
8. Refill the column.
9. Nest untreated parts according step 1.
10. Nest the next column until finished.

In step 6, the remounting algorithm is developed to minimize the distance between parts and sheet boundaries by rearranging nesting sequences. A new algorithm called random nesting algorithm is imported in the nesting process, which will be explained in the following section.

5.3.2. Heuristic searching algorithm. The heuristic searching algorithm is developed to find empty spaces among nested parts, which are always found between nested parts and even within single parts. It is hence important to locate these spaces and fill them with parts of suitable size. For example, as shown in figure 16, a small rectangle is inserted into the two nested parts: a bull head and a rabbit. The heuristic searching algorithm is developed to achieve the above task. As the parts are allocated one by one from the bottom left corner of a sheet, this algorithm is carried out in the following two directions:

(a) the space between a nested part and the part on its left side;
(b) the space between a nested part and the part below it.

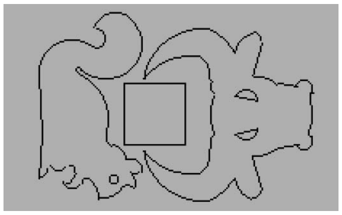


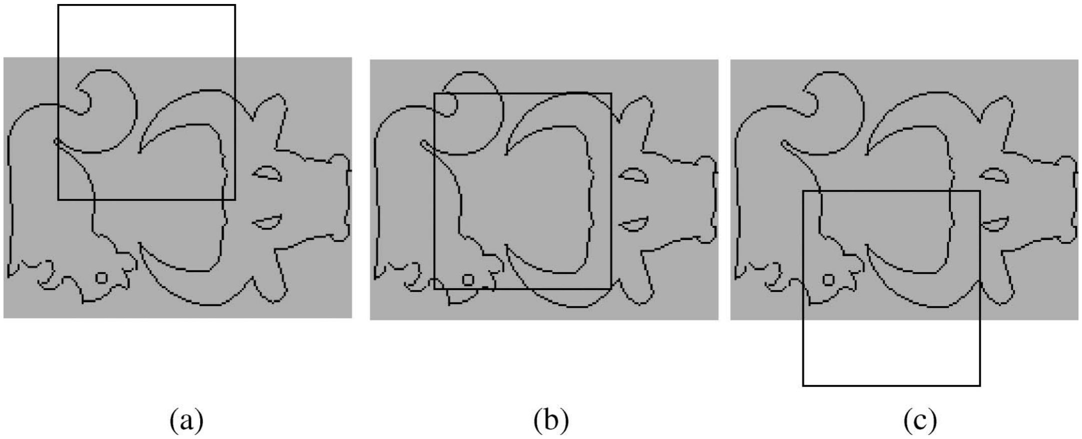
Figure 16. Nesting parts in empty spaces.

The heuristic searching is to locate a searching square in a position so that the empty spaces are enclosed. The following definitions are made so that the heuristic searching algorithm can be used. The area of the rectangular enclosure of the largest nested parts determines the area of the predefined searching square. As shown in figure 17, this searching square is allocated in three positions. The searching algorithm will then perform the following steps. First, the start point is randomly selected, if searching space is failed, the algorithm moves the rectangle to a new position. It will continue searching until an empty space is found. This is determined by the fact that the nested parts are all merged in a used region called UsedRgn, which does not contain the region of the empty space. It is assumed that the region of the searching square is RectRgn; the space region is defined SpaceRgn. The Boolean calculation of regions is

$$\text{SpaceRgn} = \text{RectRgn} - \text{UsedRgn} \quad (3)$$

If heuristic searching algorithm finds spaces where parts can be inserted, a part will be inserted by the following work three steps:

(a) shrinking the searching square until it is totally inside the empty space;



(a) (b) (c)

Figure 17. Three positions of the searching square.

(b) regarding this square as a new sheet, identifying a small part, rotating it to a suitable angle;
(c) inserting it into the rectangle.

5.3.3. Random nesting algorithms. The random nesting algorithm under heuristic principles randomly grants all parts a number of nesting order. In certain order, it generates a layout, and then records the utilization ratio. The parts are successively placed on the sheet under the overlapping judgement. If the part overlaps any of the previously placed parts, it is moved until it no longer overlaps. The part is moved as close as possible to the centre of mass of the previously placed parts. This ensures that the part's final position is contacting, but not overlapping, a previously placed part. Once all of the parts have been placed and moved, those that remain completely on the sheet constitute an actual layout. The least nesting efficient layouts are deleted and replaced with the next random layout. This process is replicated until the condition is met. Normally, the condition is the times of the generations.

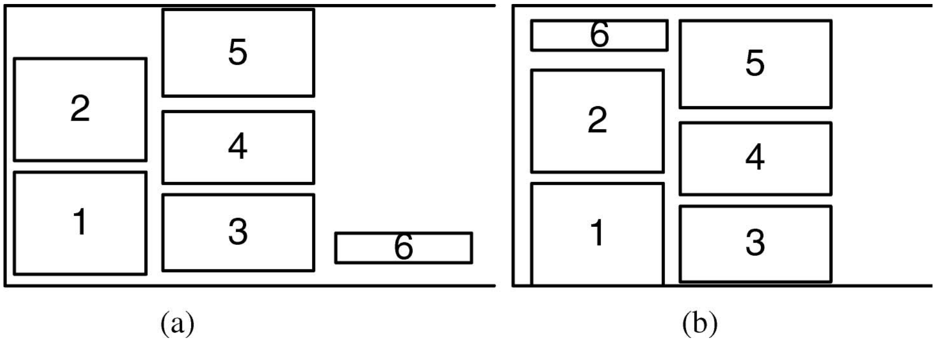
The heuristic nesting algorithm is often integrated with the BL nesting algorithm. In figure 18, six ordered parts are involved in random nesting algorithm. It assumes that parts 1–5 have already been placed and part 6 is about to be placed using the relevant bottom-left definition. Orderly, parts can only be placed to the right of the current nesting front [shown in figure 18(a)]. Obviously, another nesting order, 1, 2, and 6, 3, 4, 5 can avoid the empty space between part 2 and the boundary of the sheet. The random nesting algorithm finds out the best nesting result as shown in figure 18(b) after different nesting orders are tried several times.

5.3.4. Heuristic filling algorithms. The heuristic searching algorithm can insert multiple parts into an empty space. It needs efficient algorithms to guide the filling process. The heuristic filling algorithm is utilized in the bottom-left nesting process. A time key is defined in this process when the area of the empty space equals the total area of parts to be nested. The time key pauses the BL nesting to allow inserting parts into spaces between parts or in each single part. After the process randomly grants the orders to the parts, the heuristic algorithm nests one part at a time. The first shape will be nested at a certain position along the boundary of the irregular-shaped enclosure where the heuristic algorithm gives its optimal location and the degree of rotation. A new boundary of the stock is then extracted, that is from the original simulation enclosure minus the nested shape. The next shape is then nested in this newly evolved boundary using the heuristic algorithm. The new boundary is then extracted again. This forms a new boundary for the subsequent shape to be nested. Different layouts generated from each nesting order are recorded to compare the utilization ratio of these layouts to find out the best. The flowchart as shown in figure 19 summarizes the boundary nesting approach.

The other optimization that the algorithm requires is that: if a real part has an initial allocation in the centre of the space that does not overlap any existing parts including excluded boundary in the sheet, then the algorithm translates it towards other parts in the sheet until it contacts with a previously allocated part. The algorithm can produce an allocation of all inserters with higher efficiency after nesting plenty of generations such as 1000 times. The algorithm did not include any of the options described in the above sections. These algorithms could clearly be adapted here just as in the rectangular sheet problem, and even the nesting on the irregular sheet, which would presumably increase the efficiency here as well.

5.4. The integrated approach

The single sheet algorithm could be applied iteratively to arrive at a layout for the entire problem, but this is generally not optimal. By integrating the nesting discussed algorithms, the hybrid algorithms start with identifying the quantity and proportion of the shapes. The structure of nesting process in a single rectangular sheet is shown in figure 20. When iteratively applying the single sheet algorithm, the objective at every step is to minimize waste on the current sheet.



(a) (b)

Figure 18. Nesting in empty spaces produces the bottom-left allocation.

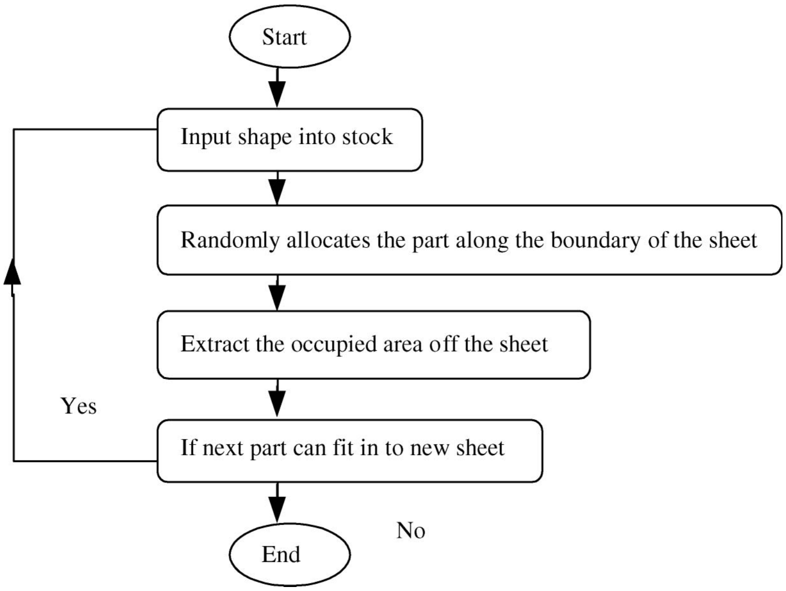


Figure 19. Flowchart of the heuristic filling algorithm.

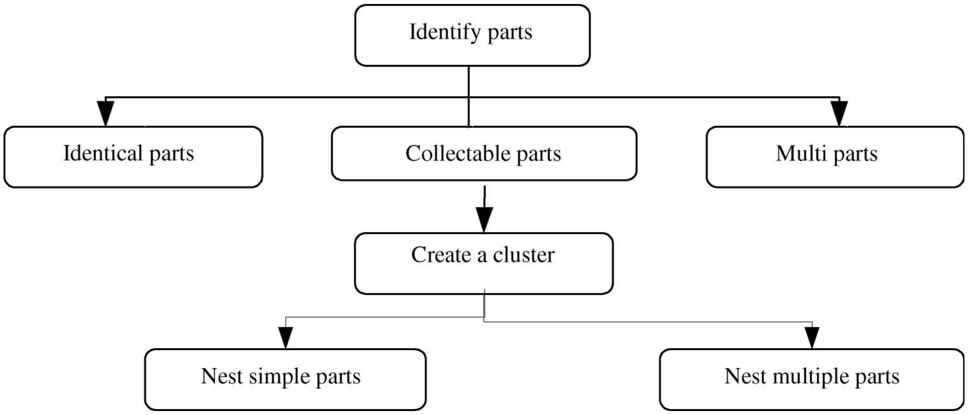


Figure 20. Structure of the nesting process of the single rectangular sheet nesting.

Heuristic algorithms have advantage in nesting complicated irregular parts; BL algorithms can quickly find out near optimal layouts when nesting similar width parts. The combination of these algorithms produces hybrid algorithms – the core of the intelligent CAN system. For multiple parts, the nesting process includes the follow steps:

1. settle parts into an enclosures and apply the pair-wise algorithm;
2. find out parts or clusters of similar width;
3. order part or cluster in descending size;
4. BL nest similar width parts;
5. random order nest parts;
6. heuristically search spaces;
7. heuristically fill spaces.

If there are still some un-nested parts after step 4, the system will proceed to step 5. Steps 6 and 7 are not operated until the total area of the left parts equals to the area of spaces in the nested parts. The last two steps replicate until all parts or clusters are nested.

6. Test examples and discussions

An intelligent CAN system has been developed as a result of the above research. This system has been developed as a software platform for testing the performance of different nesting algorithms. A screen copy of the system is shown in figure 21. Standard interfaces are designed to integrate this system with product design software packages, process planning and CAM software packages.

The intelligent CAN system has been integrated with a CAD/CAPP/CAM software package (Xie *et al.* 2001). The CAN/CAPP/CAM system provides part information to the intelligent CAN system and read the nesting layout for process planning. As shown in figure 23 (see below) the output of the intelligent CAN system is sent to a CAPP/CAM software package. Auxiliary cut-in and cut-out paths are automatically added for cutting purposes. The process planning and cutting path planning methods have been discussed by Xie *et al.* (2001). The final output of the CAN/CAPP/CAM system is NC code for various types of machines.

The system has been used for sheet metal and cloth cutting companies in Auckland, New Zealand. Case studies have been carried out to test the algorithms presented in section 5 in a sheet metal cutting company. The test results show that the intelligent CAN system has greatly improved the utilization ratio. Two nesting examples are selected from the sheet metal cutting company, as shown in figures 22 and 23.

> Example 1: 38 parts are nested on a 300 × 320 mm sheet while the gap between two adjacent parts is set to 5 mm. The utilization ratio on this sheet is 80.12% (this can be improved when the number of small size parts increases).

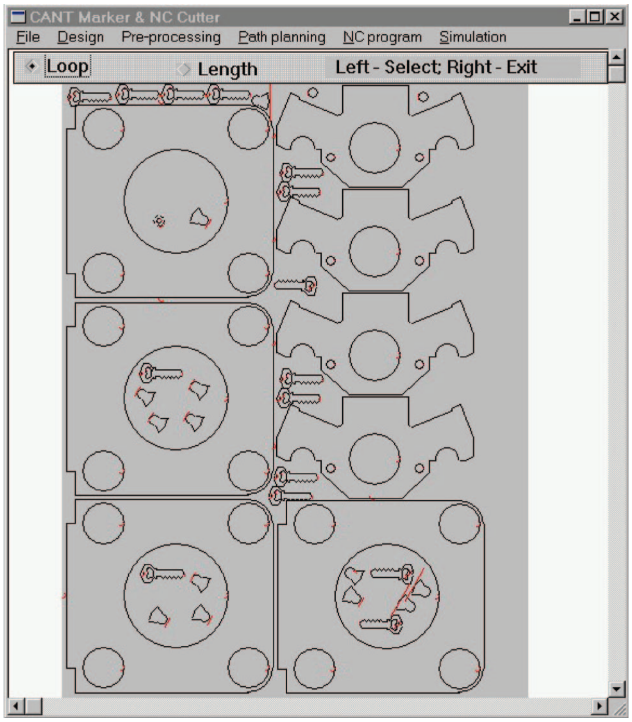


Figure 21. A screen copy of the intelligent CAN system.

> Example 2: having the same gap as the above, 43 parts are nested on a 500 mm × 450 mm sheet. The utilization ratio is 71.26%.

In figure 22, 'L'-shaped polygons, 'chickens', some triangles, and other parts cover most of the available area in the rectangular sheet. This shows that the pairwise algorithms are capable of nesting efficiently. In addition, other complicated polygons have found suitable positions in this sheet. It indicates that the heuristic nesting algorithms improve the result of the simple bottom-left nesting algorithm. The layout has proved that the random nesting algorithm successfully combines with the similar width-based BL nesting algorithm.

In figure 23, the paired 'rabbits' also show efficient layouts. Convex, concave shapes, and squares are nested on the sheet. Similar width polygonal parts are nested column by column, and the two biggest irregular parts, the name badges of the manufacturing company, are nested in the bottom-left corner. This result shows that the algorithms are capable of nesting similar width parts. Two badges produced several spaces within the parts and between the parts' contours and the sheet boundary. This is extremely important to this research as other studies have usually avoided this type of situation by ignoring the existing empty spaces. If the enclosure is represented by a polygon rather than a rectangle, the utilization ratio can be further improved. The present result shows the ability of the intelligent CAN system to find the empty spaces and to efficiently nest a variety of parts in these spaces.

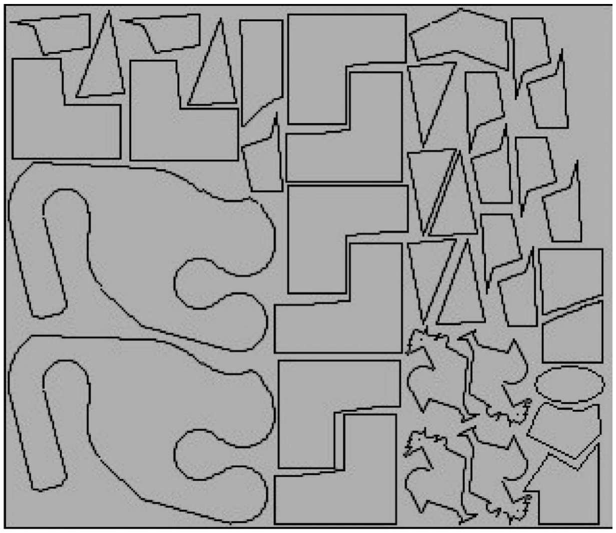


Figure 22. Example 1.

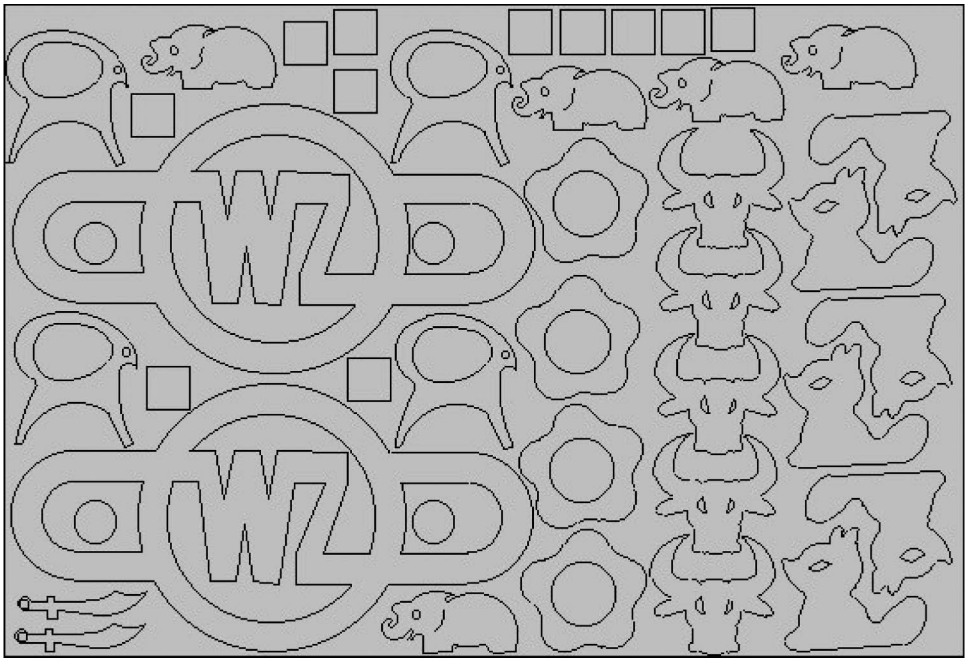


Figure 23. Example 2.

7. Conclusion

The current paper presented an intelligent CAN system for optimal nesting of irregular parts after systemically reviewing the recent developments of nesting algorithms for various nesting tasks. An integrated approach to the nesting problem was introduced, in which nesting algorithms such as the rectangular enclosure method, BL nesting algorithms, heuristic algorithms and genetic algorithms are investigated and integrated. These nesting algorithms are efficiently integrated to nest both regular and irregular parts. Improvements have also been made to traditional nesting algorithms and new nesting algorithms are proposed such that the utilization ratio is improved.

As a result of the above research, an intelligent CAN system has been developed to support the nesting of parts with complicated shapes. This intelligent CAN system can generate optimal or near optimal layouts for the nesting of irregular parts, with the support of hybrid algorithms, such as heuristic algorithms and bottom-left algorithms.

This system has been used in several manufacturing companies with the integration of a CAPP/CAM software package and has helped them improve their product utilization ratio. The optimal nesting results from the system are directly converted to CNC codes for various types of cutting systems. Supported by the developed algorithms, this intelligent CAN system has excellent application prospects of enabling optimal nesting in the 2D irregular parts industry. This intelligent CAN system was originally designed for laser cutting process. It has been further developed for nesting of parts produced with other cutting processes, e.g. flame, plasma, water jet cutting. The system has also been further extended to cutting processes with constraints such as cloth cutting and woodcutting.

References

Alves, J.C., Ferreira, J.C., Oliveira, J.F., Ferreira, J.S., Matos, J.S. and Albuquerque, C., Fafner – accelerating nesting problems with FPGAs. *Proceedings of the 7th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCMM 1999)*, 21–23 April 1999, Napa Valley, CA, United States, pp. 168–177.

Arvanitis, K.G., An algorithm for adaptive pole placement control of linear systems based on generalized sampled-data hold functions. *J. Franklin Inst.*, 1999, **336**, 503–521.

Balas, E. and Perregaard, M., Lift-and-project for mixed 0-1 programming: recent progress. *Discrete Appl. Math.*, 2002, **123**, 129–154.

Bhadury, J. and Chandrasekaran, R., Stock cutting to minimize cutting length. *Eur. J. Opl Res.*, 1996, **88**, 69–87.

Bruss, F.T. and Delbaen, F., Optimal rules for the sequential selection of monotone subsequences of maximum expected length. *Stoch. Proc. Appl.*, 2001, **96**, 313–342.

Caroe, C.C. and Tind, J., Cutting-plane approach to mixed 0-1 stochastic integer programs. *Eur. J. Opl Res.*, 1997, **101**, 306–316.

Chen, Y., Tang, M., Tong, R. and Dong, J., Packing of polygons using genetic simulated annealing algorithm. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/J. Comput-Aided Des. Comput. Graphics*, 2003, **15**, 598–603.

Cheng, S.K. and Rao, K.P., Concepts of neighbourhood and universal compact yield towards achieving best pattern layouts. *Int. J. Prod. Res.*, 1999, **37**, 3643–3658.

Cheng, S.K. and Rao, K.P., Large-scale nesting of irregular patterns using compact neighborhood algorithm. *J. Mater. Process. Technol.*, 2000, **103**, 135–140.

Cheng, Y.F., Guo, S.J. and Lai, H.Y. Dynamic simulation of random packing of spherical particles. *Powder Technol.*, 2000, **107**, 123–130.

Dowsland, K.A., Vaid, S. and Dowsland, W.B., An algorithm for polygon placement using a bottom-left strategy. *Eur. J. Opl Res.*, 2002, **141**, 371–381.

Dufour, J., McBride, R., Zhang, P. and Cheng, C.-K. Building block placement tool. *Proceedings of the 1997 Asia and South Pacific Design Automation Conference, ASP-DAC*, 28–31 January 1997, Chiba, Japan, pp. 271–276.

Dumitrescu, A. An approximation algorithm for cutting out convex polygons. *Configurable Computing: Technology and Applications*, 2–3 November 1998, Boston, MA, United States, pp. 823–827.

Freeman, H. and Shapira, R., Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM 18*, 1975, **7**, 409–413.

Grinde, R.B. and Cavalier, T.M., New algorithm for the minimal-area convex enclosure problem. *Eur. J. Opl Res.*, 1995, **84**, 522–538.

Grinde, R.B. and Cavalier, T.M., Containment of a single polygon using mathematical programming. *Eur. J. Opl Res.*, 1996, **92**, 368–386.

Heckmann, R. and Lengauer, T., Computing closely matching upper and lower bounds on textile nesting problems. *Eur. J. Opl Res.*, 1998, **108**, 473–489.

Hopper, E. and Turton, B., A genetic algorithm for a 2D industrial packing problem. *Comput. Ind. Engng*, 1999, **37**, 375–378.

Hopper, E. and Turton, B.C.H., An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *Eur. J. Opl Res.*, 2001, **128**, 34–57.

Kaufmann, M., Linear-time algorithm for routing in a convex grid. *IEEE Trans Comput-Aided Des. Integr. Circuits Systems*, 1990, **9**, 180–184.

Lee, H.-C. and Woo, T.C., Determining in linear time the minimum area convex hull of two polygons. *IIE Trans.*, 1985, **20**, 338–345.

Lutfiyya, H., McMillin, B., Poshyanonda, P. and Dagli, C., Composite stock cutting through simulated annealing.' *Math. Comput. Modelling*, 1992, **16**, 57–74.

MacLeod, B., Moll, R., Girkar, M. and Hanifi, N., An algorithm for the 2D guillotine cutting stock problem. *Eur. J. Opl Res.*, 1993, **68**, 400–412.

Martini, H. and Soltan, V., Minimum Convex partition of polygonal domains by guillotine cuts. *Discrete Comput. Geom.*, 1998, **19**, 291.

Milenkovic, V.J. and Daniels, K., Translational polygon containment and minimal enclosure using mathematical programming*1. *Int. Trans. Opl Res.*, 1999, **6**, 525–554.

Mishra, V.N., Singh, V., Prasad, B. and Ojha, S.P., Optical dispersion curves of a light waveguide with a guiding region cross-section resembling a double convex lens. *Optik (Jena)*, 1999, **110**, 397–399.

Murty, R.L., Reddy, P.J. and Prasad, A.S., *Mechanics of Wobble Broaching*, 1983, **2**, 123–131.

Nandy, S.C. and Bhattacharya, B.B., On finding an empty staircase polygon of largest area (width) in a planar point-set. *Comput. Geom.: Theory and Applications*, 2003, **26**, 143–171.

Nee, A.Y.C., Seow, K.W. and Long, S.L. Designing Algorithm for nesting irregular shapes with and without boundary constraints. *CIRP Annls 1986: Mf Technol., Annls Int. Instn Prod. Engng Res.*, 1986, 107–110.

Peach, R.C. Rigorous two-dimensional equations for the analysis of contoured crystal resonators. *Proceedings of the Annual Frequency Control Symposium*, 1–3 June 1988, Baltimore, MD, USA, pp. 38–44.

Ramesh Babu, A. and Ramesh Babu, N., A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. *Comput.-Aided Des.*, 2001, **33**, 879–891.

Roth, J., Hashimshony, R. and Wachman, A., Generating layouts with non-convex envelopes. *Build. Environ.*, 1985, **20**, 211–219.

Roy, U. and Dasari, V.R., Implementation of a polygonal algorithm for surface-surface intersections. *Comput. Ind. Engng*, 1998, **34**, 399–412.

Shang, K.H. and Song, J.-S., Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Mgmt Sci.*, 2003, **49**, 618–638.

Takahara, S., Kusomoto, Y. and Miyamoto, S., An adaptive meta-heuristic approach using partial optimization to non-convex polygons allocation problem. *10th IEEE International Conference on Fuzzy Systems*, 2–5 December 2001, Melbourne, Australia, pp. 1191–1194.

Tay, F.E.H., Chong, T.Y. and Lee, F.C., Pattern nesting on irregular-shaped stock using Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, 2002, **15**, 551–558.

Toh, S.K.H., Loh, H.T., Nee, A.Y.C. and Lee, K.S., A feature-based flat pattern development system for sheet metal parts. *J. Mater. Process.Technol.*, 1995, **48**, 89–95.

Wu, S.Y., A cutting plane approach to solving quadratic infinite programs on measure spaces. *J. Global Optim.*, 2001, **21**, 67–87.

Xie, S.Q. and Tu, Y.L., Liu, J.Q. and Zhou, Z.D., An integrated and concurrent approach for compound sheet metal cutting and punching. *Int. J. Prod. Res.*, 2001, **39**, 1095–1112.