Int. J. Production Economics 121 (2009) 148–161

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

# Active-set sequential quadratic programming method with compact neighbourhood algorithm for the multi-polygon mass production cutting-stock problem with rotatable polygons

M.T. Yu [a], T.Y. Lin [b,*], C. Hung [a]

[a] Department of Mechanical Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan, ROC
[b] Department of Mechanical Engineering, Chung Cheng Institute of Technology, National Defense University, Tahsi, Taoyuan 335, Taiwan, ROC

ARTICLE INFO

*Article history:*
Received 16 October 2006
Accepted 16 January 2009
Available online 15 May 2009

*Keywords:*
Cutting-stock problem
Material saving
Rotatable
Active-set
Sequential quadratic programming

ABSTRACT

The cutting-stock problem, which considers how to arrange the component profiles on the material without overlaps, can increase the utility rate of the sheet stock, and is thus a standard constrained optimisation problem. In some applications the components should be placed with specific orientations, but in others the components may be placed with any orientation. This study presents an overlap index and it is much more suitable for the active-set SQP method which can reduce the time spend for constraint consideration. Using this method, various object orientations can be considered easily and the number of object on the sheet stock can be improved by up to eight percent.


## 1. Introduction

The cutting-stock problem is a key consideration in many manufacturing industries, such as textile, garment, metalware, paper, ship building, and sheet metal industries. The material expense is a large part of manufacturing cost in these industries. For example: the fabric expense is approximate 50–60% of the manufacturing cost (Wong and Leung, 2008). The cutting-stock problem is a type of constrained optimisation problem. The objective of the cutting-stock problem is to place as many objects as possible on a sheet stock. The common constraint is that no object may overlap with another. Therefore, there are two key points for the cutting-stock problem, and it can thus be divided into two sub problems: how to consider the overlaps, and how to place the objects.

In the application viewpoint, the product of some industries may be mass production and the cutting-stock problem will be very complex if considering every object as an independent individual. For simplifying the problem, the components of a product may be placed as a cluster first, and then nest the cluster instead of the original components on the sheet stock. This "cluster-then-nest" strategy is useful for simplifying the mass production cutting-stock problem because the number of objects is reduced. The advantage of the cluster-then-nest strategy is not only for placing, but manufacture. The manufacturing time may decrease because the components of a product can be cut at the same time by using a die.

This study considers the mass production cutting-stock problem, and the cluster-then-nest strategy with an improvement strategy is used. As the classification by Wäscher, et al. (2007), the problem that will be solved in this study is an "identical item packing problem (IIPP)".

## 2. Literature review

The cluster-then-nest strategy is used by Cheng and Rao (1999). They used a sliding technique (Cheng and Rao, 1997) to place all objects—the components—of a product

* Corresponding author. Tel.: +886 3 3809870; fax: +886 3 3907306.
*E-mail address:* tsylin0912@hotmail.com (T.Y. Lin).

0925-5273/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2009.01.014

first, i.e. cluster, then integrated these objects into a cluster, and nested it by compact neighborhood algorithm (CNA). Finally, a Genetic Algorithm was used to adjust the position and orientation of the object to improve the result (2000).

Even the cluster-then-nest strategy is good for mass production, there are few literature discussing it. Therefore, the techniques used in the methods for placing different kinds of objects on one sheet stock may be used for generate cluster, and the technique for placing only one kind of object on the sheet stock may be used for nesting.

### 2.1. Methods for cluster

Bennell, et al. (2001) proposed a method to find the relative positions where one object contacts another object, and these positions can be shown as a polygon called a "no-fit polygon". The sliding technique (Cheng and Rao, 1997) can be used to get the no-fit polygon, but it does not consider the hole in an object. Thus, a sophisticated NFP calculation may get better no-fit polygon than the sliding technique. Several methods use the no-fit polygon to consider the overlaps and propose a strategy of placing objects (Dowsland et al., 2002) and deciding the placing sequence (Gomes and Oliveira, 2002). Even though the no-fit polygon is a popular technique for considering overlaps, the methods using the no-fit polygon cannot consider different placement orientations because the no-fit polygon is generated with a fixed orientation. In some applications, the objects can be placed only with some specific orientations because of the property of the material, such as the rolling direction and the pattern on the sheet stock surface. This is another constraint called "orientation constraint" in the cutting-stock problem, but the orientation constraint is not applicable in certain applications. The sheet stock utility rate will not be high when using these algorithms to solve cutting-stock problems without the orientation constraint because these algorithms do not consider the rotation of objects, and finding the best placing sequence also requires much calculation effort.

Jakobs (1996) used a two-stage method with a concept similar to the polygon approximation. In the first stage, the objects are enclosed by minimum rectangles, and then these rectangles are placed instead of the real object. The objects may be rotated when finding the minimum enclosed rectangle, but the objects will not be rotated in the next stage. In the second stage, the real objects are placed directly and the initial positions of real objects are set as the rectangle positions. There may be no gap between rectangles after the first stage, but there will be many gaps when considering the real objects. The real objects are moved downward to fill the gaps between them to reduce the height of the placing pattern. Dagli and Poshyanonda (1997) used a coordinate transfer matrix to find the minimum rectangle and represented the objects as binary matrices. They used artificial neural networks to learn to place the binary matrices instead of objects, and used a Genetic Algorithm to decide the placing sequence. These methods can consider different orientations of objects, but objects can rotate only when finding the minimum enclosed rectangle. The orientation cannot be adjusted to a better one in the placing process.

In some formulation methods the cutting-stock problem is treated as a standard optimisation problem, and uses the object positions and orientations as design variables. The overlaps with the overlap areas are considered directly. Ismail and Hon (1995) also used binary matrices to represent the objects, but the object profiles may become deformed if the binary matrices, instead of the real objects, are rotated. Also, the calculation effort increases when rotating the real objects. This is because the objects have to be transferred to binary matrices after every rotation, and using the binary matrices to determine the overlap area also requires increased computation effort. Petridis and Kazarlis (1994) used the real profiles instead of binary matrices and calculated the overlap of real objects when considering the overlaps. The object profiles will not be deformed by this method, but the rotation of objects is not considered.

### 2.2. Methods for nest

Kershner (1968) showed that no convex polygon with more than six sides can be nested without gaps, and there are only eight kinds of pentagons and three kinds of hexagons that can be nested without gaps. The cutting-stock problem may be changed to approximate an object as a special pattern because the nesting pattern is decided when the special pattern is chosen. Various methods based on polygon approximation have been proposed (Koroupi and Loftus, 1991; Yu and Tseng, 2005).

A review of the literature shows that the Genetic Algorithm, the artificial neural network, and other heuristic algorithms are commonly used methods for solving the cutting-stock problem. However, there are also many other algorithms, such as sequential linear programming (SLP) (Arora, 2004), traditional sequential quadratic programming (SQP) (Arora, 2004), and constrained steepest descent (CSD) (Arora, 2004), that can be used to solve this type of optimisation problem. These algorithms use the gradient or Hessian calculation of the cost function as the search direction. However, the cost function of the cutting-stock problem is usually the number of objects placed on the sheet stock or the area of the necessary sheet stock, neither of which is sensitive to the object position and orientation. When using the number of objects placed on the sheet stock as the cost function, the adjustment does not affect the cost function unless it makes the object out of the sheet stock. When using the area of the necessary sheet stock as the cost function, it will be calculated by multiplying the maximum upper bound of objects and the width of the sheet stock. At this time the position adjustment of objects will not affect the cost function unless it is the object with maximum upper bound. This will limited the efficient of the SQP method. In these two cases, the adjustment of the position or orientation of objects is not sensitive to the cost function. If the cost function does not relate to design variables sensitively, these algorithms cannot be used.

An algorithm called "active-set SQP method" (Lim and Arora, 1986) in this study, based on the traditional SQP method, reduces the calculation effort by reducing the considered constraints. For an inequality constraint that should be less than or equal to zero, the constraint is called "inactive" if its value is less than zero, i.e., the solution satisfies the constraint. The constraint is said to be "active" if its value equals to zero. Conversely, it is call "violated" if its value is larger than zero. The inactive constraints can be ignored in the iteration. Hence, the number of constraints that have to be considered is reduced.

This study presents a formulation, and solves it by using active-set SQP method for generating the cluster. The formulation considers the orientations of objects directly, and improves the sensitivity between the cost function and design variables. Thus, the problem can be solved by the traditional SQP method that is more efficient than the Genetic Algorithm. This study also presents an overlap index that considers the overlap amount instead of the overlap area. Thus, the calculation effort may be reduced and is well-suited for the active-set SQP method.

## 3. Method

A solving method for the cutting-stock problem can be divided into two parts: the overlap consideration method, and the placing strategy, which depends on the overlap consideration method. Therefore, the overlap consideration method will be introduced before introducing the placing strategy in this section.

### 3.1. Overlap consideration method

The objects of the cutting-stock problem cannot overlap one other. In traditional methods, the overlap area is calculated and the object positions are adjusted to reduce the overlap area until the total overlap area equals to zero when considering overlap directly. However, calculating the real overlap area is computation-intensive, so another index should preferably be used instead of the real overlap area. This study uses the maximum "depth" of two objects as the overlap index when considering the overlap.

When considering overlap between two objects, the "depth" means the distance from the vertex on one object to a point on the edge of the other object, and the maximum depth is the largest distance. The detailed calculation process is shown in Fig. 1 while the process can be explained with the example in Fig. 2. *A* and *B* are two objects and every vertex is numbered in a counter-clockwise order ($a_1 \sim a_8$ and $b_1 \sim b_8$, respectively). $O_A$ and $O_B$ are the centres of gravity (COG) of object *A* and object *B*, respectively. Objects are drawn by CAD software—CATIA, and the COG can be obtained automatically in the CATIA. As shown in the process (Fig. 1), the first step of finding the maximum depth is transferring the original coordinate to the coordinate system where the *y*-direction is parallel to the $\overrightarrow{O_A O_B}$ vector, which will be helpful for calculating depths. The next step initialises the maximum depth, and its value is determined by subtracting the *y*-value of $a_1$ from the *y*-value of $a'_1$, where $a'_1$ is the point that $a_1$ projects onto object *B* in the $\overrightarrow{O_A O_B}$ direction. The depth will be positive if the vertex is inside the other object, such as the depth of $a_1$. Similarly, if the vertex is outside the other object, the depth will be negative. Once the depth of $a_1$ is known, the depth of $a_2$ will be calculated, and is less than the depth of $a_1$ as shown in the figure. Thus, the maximum depth will not be updated. Only the depths of the object vertices have to be calculated when finding the maximum depth, and it is not necessary to calculate the depths of the points on the edge. It is obvious that the maximum depth will coincide with a vertex, because all edges are linear. If depth of a point is searched along an edge, it will be increased or decreased monotonously until the movement meets a corner. Thus, the maximum depth will coincide with a vertex. The next point is $a_3$ and its depth is negative because the *y*-value of $a_3$ is larger than $a'_3$. Other vertices on object *A* will also be considered one by one. Similarly, the depths of the vertices on object *B* are determined by subtracting the *y*-value of the projection point from the y-value of the vertex, and calculated one by one.

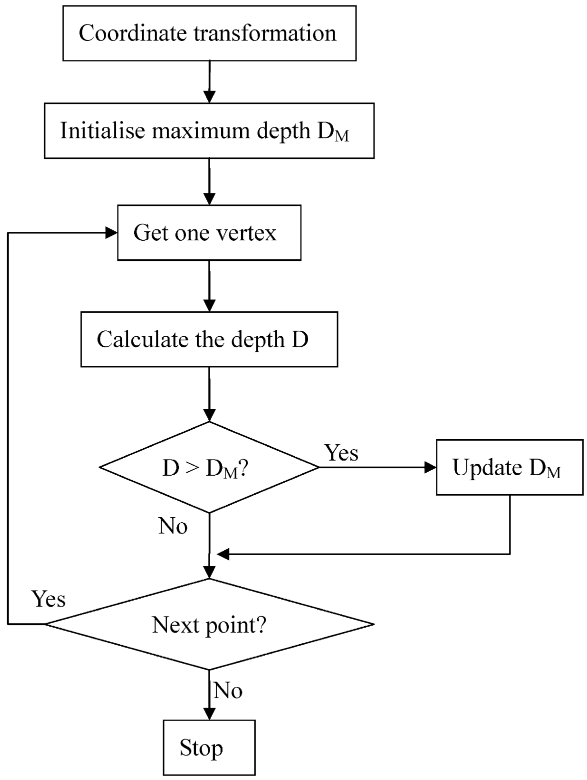


**Fig. 1.** Flowchart for finding the maximum depth.

By this way, the depth will be negative if the vertex is outside of the other object, and it is not necessary to check the vertex is inside the other object or not. The overlap index will be negative if there is a gap between two objects. A negative overlap index will be helpful for the active-set SQP method. The maximum depth will be negative but the overlap area is never negative. Therefore,

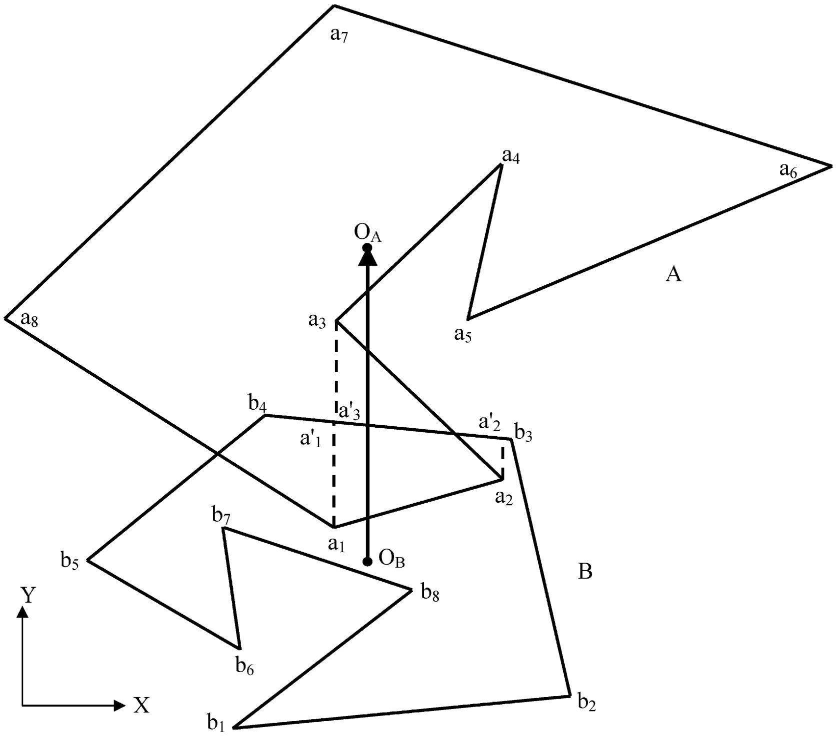


**Fig. 2.** Depths between two objects.

the maximum depth is more suitable for the active-set SQP method, which will be described in the next section.

### *3.2. Placing strategy*

The placing strategy in this study can be divided into three steps. The first step is to place the different objects as a cluster, and then to generate the nesting pattern according to the cluster. Finally, the third step is to adjust the orientation of the nesting pattern generated in the second step to maximise the sheet stock utility rate.

#### *3.2.1. First step—cluster*

The cutting-stock problem is formulated as a standard form of the constrained optimisation problem as follows:

$$\text{cost function}: \quad \text{minimise}\, f = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|\overrightarrow{O_iO_j}\| \tag{1}$$

$$\text{design variables}: \quad x_i, y_i, \theta_i \qquad i = 1 \sim N \tag{2}$$

$$\text{constraints}: \quad g_i = D_{Mjk} \le 0 \qquad i = 1 \sim \frac{N(N-1)}{2} \qquad j = 1 \sim (N-1) \qquad k = (j+1) \sim N \tag{3}$$

where $N$ is the number of objects; $\|\overrightarrow{O_iO_j}\|$ is the norm of vector $\overrightarrow{O_iO_j}$, i.e., the distance between $O_i$ and $O_j$; $x_i$ is the $x$-coordinate value of the COG of object $i$; $y_i$ is the $y$-coordinate value of the COG of object $i$; $\theta_i$ is the orientation of object $i$; $D_{Mjk}$ is the maximum depth between object $j$ and object $k$. The cost function is to minimise the summation of distances between any two objects, which means that the objects have to be as close as possible. This cost function (the distance summation) is more sensitive to the design variables than the number of objects or the necessary sheet stock area. Therefore, it is more suitable for the traditional SQP method. The constraints are that the maximum depths of any two objects cannot be larger than zero, i.e., one object can only be far away or just contact the adjacent object.

As shown in Eq. (3), there are many constraints when the number of objects is large, and reducing the number of constraints is important in order to decrease the calculation effort. Because of the nature of this problem, the constraints may not be reduced in physical ways, but they can be reduced using mathematical methods. If the solution satisfies the constraint, it will not be necessary to consider whether or not the constraint still exists. Therefore, the active-set SQP method is used to reduce the considered constraints in this study.

A constraint will never be inactive when using the overlap area as the constraint value because the overlap area is never less than zero. Thus, the number of constraints cannot be reduced. However, when using the maximum depth to consider overlap, the maximum depth will be negative if there is a gap between two objects, and therefore the constraint becomes inactive. These constraints will be ignored and the computation effort will be decreased.

After formulating the problem as a constrained optimisation problem, it can be solved by using the active-set SQP method. The SQP method is a numerical method for solving optimisation problem. The process of a numerical method is an iterative process of finding "search direction" and "step size".To solve the optimization problem by SQP method, the KKT conditions of the Lagrange function is used. The Lagrange function is defined as follows:

$$L(d,\mu) = f(d) + \mu^T g \quad (4)$$

where $\mu$ is the vector form of Lagrange multipliers, and $d$ is a collection of design variables, i.e. $x_i$, $y_i$, $\theta_i$. The numerical solving process of the KKT conditions is an iterative process of calculating the new solution $d^{(k+1)}$

$$d^{(k+1)} = d^{(k)} + \Delta d^{(k)} \quad (5)$$

where $k$ is the iteration number, and $\Delta d^{(k)}$ is the change of design variables. It is also the search direction of the SQP method. The SQP method defines a QP subproblem to calculate the search direction. The flowchart of SQP

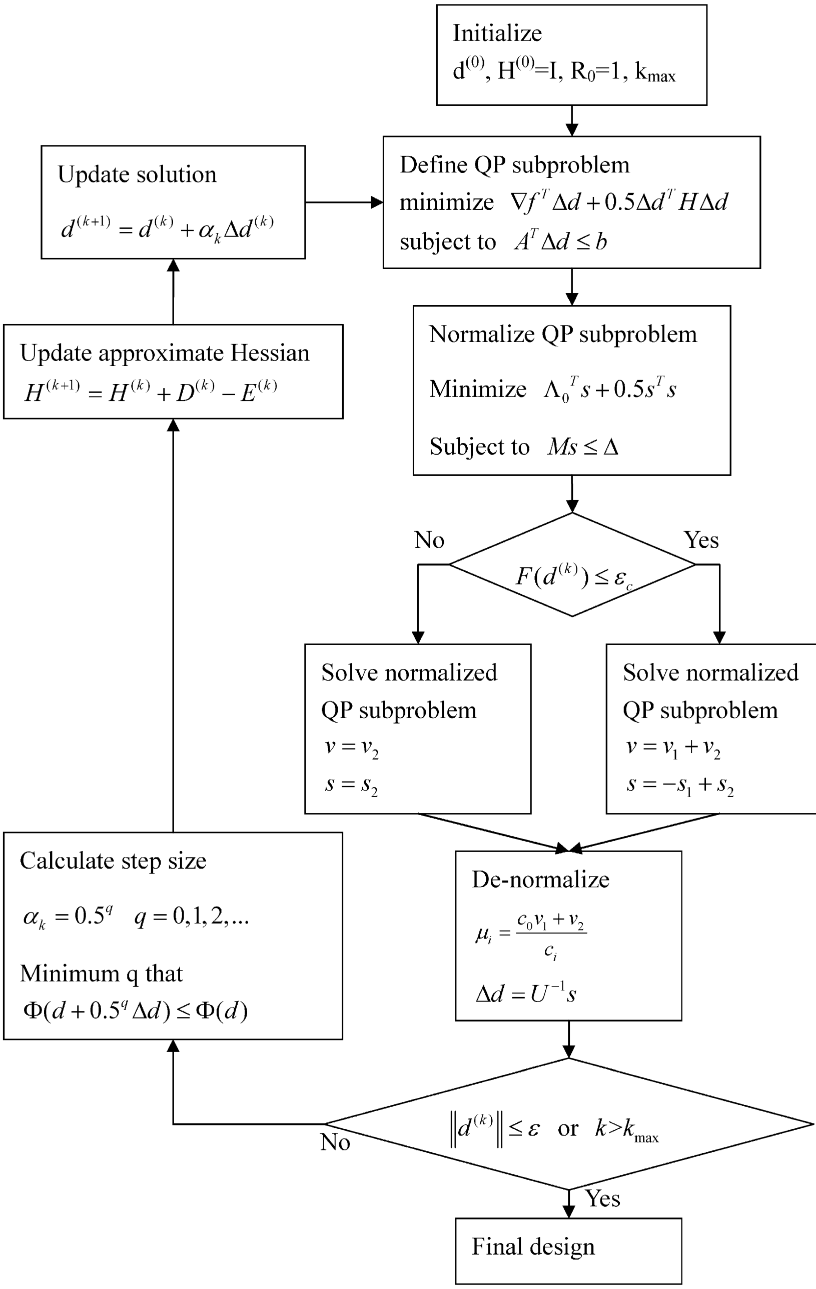


**Fig. 3.** The flowchart of SQP method.

method is shown in Fig. 3, and the process is described as follows.

(1) Select an initial solution, and the initial solution is $d^{(0)} = 0$ in this study. Set other parameters that will be described later.

(2) Define the QP subproblem with the active-set strategy. The QP subproblem is defined as follows:

$$\text{minimize} \quad \nabla f^T \Delta d + 0.5 \Delta d^T H \Delta d \tag{6}$$

$$\text{subject to} \quad A^T \Delta d \le b \tag{7}$$

where $H$ is the approximate Hessian matrix of Lagrange function,

$$A_{j_i} = \frac{\partial g_i'}{\partial d_j} \tag{8}$$

$$b_i = -g_i'(d). \tag{9}$$

When defining the QP subproblem, only the active and violated constraints need to be considered. The $g'$ is the vector form of active and violated constraints. This is the so-called active-set strategy.

(3) After the QP subproblem is defined, the problem will be normalised before solving (Arora, 1984). For normalising the QP subproblem, the $H$ is decomposed as

$$H = U^T U \tag{10}$$

where $U$ is an upper triangular matrix. A new variable $s$ is defined as

$$s = U \Delta d \tag{11}$$

and the problem can be normalised as:

$$\text{minimise} \quad \Lambda_0^T s + 0.5 s^T s \tag{12}$$

$$\text{subject to} \quad Ms \le \Delta \tag{13}$$

where

$$\Lambda_0 = \frac{U^{-1^T} \nabla f}{c_0} \tag{14}$$

$$c_0 = \| U^{-1^T} \nabla f \| \tag{15}$$

$$\Delta_i = \frac{-g_i'}{c_i} \tag{16}$$

$$c_i = \| U^{-1^T} \nabla g_i' \| \tag{17}$$

$$M = \frac{U^{-1^T} A}{c_0} \tag{18}$$

(4) Before solving the normalised QP subproblem, the maximum value of constraints is compared with a constant $\varepsilon_c$ (Arora, 1984). If the maximum value of constraints is less than the constant, i.e. the violation is not serious, the solution will focus on reducing the cost function. The solution is:

$$v = v_1 + v_2$$

$$s = -s_1 + s_2$$

where

$$v_1 = -B^{-1} M^T \Lambda_0 \tag{19}$$

$$v_2 = -B^{-1} \Delta \tag{20}$$

$$B = M^T M \tag{21}$$

$$s_1 = \Lambda_0 + M v_1 \tag{22}$$

$$s_2 = -M v_2 \tag{23}$$

If not, the solution focuses on correcting the constraints. The solution is:

$$v = v_2 \tag{24}$$

$$s = s_2 \tag{25}$$

(5) The solution of the original QP subproblem can be obtained as (Arora, 1984):

$$\mu_i = \frac{c_0 v_1 + v_2}{c_i} \tag{26}$$

$$\Delta d = U^{-1} s \tag{27}$$

(6) Check the stop condition. If $\|d^{(k)}\| \le \varepsilon$ or $k > k_{\max}$, stop the process and the current solution is the final solution. If not, continue the process. å is a small number close to zero, and $k_{\max}$ is the maximum iteration number set in the initialization step.

(7) Calculate the step size. The step size $\alpha_k$ is set as

$$\alpha_k = 0.5^q \quad q = 0, 1, 2, \ldots \tag{28}$$

$$\text{The minimum } q \text{ make } \Phi(d + 0.5^q \Delta d) \le \Phi(d) \tag{29}$$

is used to define $\alpha_k$ where

$$\Phi(d) = f(d) + R_k F(d) \tag{30}$$

$$F(d) = \max\{0, g_1'(d), \ldots, g_{p'}'(d)\} \tag{31}$$

$$R_k = 0.5 \left( R_{k-1} + \sum_{i=1}^{p'} \mu_i \right) \tag{32}$$

(8) Update the approximate Hessian matrix. The approximate Hessian matrix is updated by BFGS strategy that is described as follows:
Define three variables first.

$$z^{(k)} = \alpha_k H^{(k)} \Delta d^{(k)} \tag{33}$$

$$y^{(k)} = \nabla L(d^{(k+1)}, \mu^{(k)}) - \nabla L(d^{(k)}, \mu^{(k)}) \tag{34}$$

and

$$w^{(k)} = \theta y^{(k)} + (1-\theta) z^{(k)} \tag{35}$$

where

$$\theta = \begin{cases} 1, & \text{if } \Delta d^{(k)^T} y^{(k)} \geq \Delta d^{(k)^T} z^{(k)} \\ 0.8, & \text{otherwise} \end{cases} \tag{36}$$

Then, the approximate Hessian matrix is updated as

$$H^{(k+1)} = H^{(k)} + D^{(k)} - E^{(k)} \tag{37}$$

where

$$D^{(k)} = \frac{w^{(k)} w^{(k)^T}}{\alpha_k \Delta d^{(k)^T} w^{(k)}} \tag{38}$$

$$E^{(k)} = \frac{z^{(k)} z^{(k)^T}}{\alpha_k \Delta d^{(k)^T} z^{(k)}} \tag{39}$$

(9) Update the solution as

$$d^{(k+1)} = d^{(k)} + \alpha_k \Delta d^{(k)}, \tag{40}$$

and continue to define the QP subproblem.

All equations are shown above, and the derivation can be found in Arora (2004) and Liao(1990). There are some programs used the active-set SQP method to solve the constrained optimisation problem, such as MOST (Tseng) and IDESIGN (Arora).

*3.2.2. Second step—nest*

After the first step, the objects are placed at the position ($x_i$, $y_i$) and with the orientation ($\theta_i$), and are treated as a ckuster in this step. This step uses the CNA method (Cheng and Rao, 1999) to generate the nesting pattern. The detailed process is introduced with pentagons as an example as follows.

At first, two objects are in contact to each other at a point as shown in Fig. 4(a), and are called "stator" and "mover". The object $S$ in Fig. 4(a) is the stator and the object $M$ is the mover. The bottom-left vertex of the object is used to represent the object position, and is called a "reference point". Then $M$ slides on $S$ with a fixed orientation in a counter-clockwise direction. At beginning, $m_4$ contacts with $s_1$, and $M$ will moves along $\overrightarrow{s_1 s_2}$ until the contact vertex $m_4$ meet a corner or $M$ contacts $S$ on another point as shown in Fig. 4(b). The point $m_4$ meets a corner when it contacts with $s_2$, and then $M$ will moves along $\overrightarrow{m_5 m_4}$ until the contact vertex $s_2$ meet a corner or $M$ contacts $S$ on another point as shown in Fig. 4(c). The self-sliding process is complete when $M$ moves to the initial position, and the path of the bottom-left vertex of $M$ is recorded as a no-fit polygon as shown in Fig. 4(d). This process is called "self-sliding" because the two objects are identical and sliding relative to each other in the process.

After finding the no-fit polygon of self-sliding, the mover is removed and the stator is called object P0 as shown in Fig. 5(a). The reference point of another object (object P1) is put at an arbitrary position of the self-sliding no-fit polygon of P0 first. These two objects have their self-sliding no-fit polygons, and the right interaction point

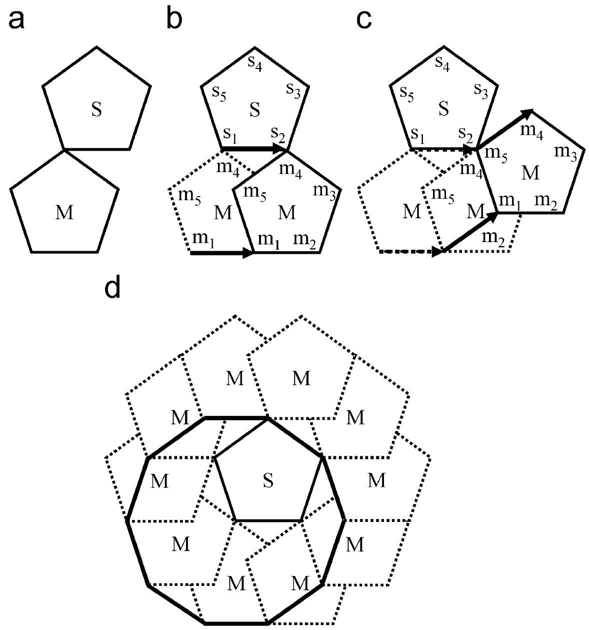


**Fig. 4.** Process of finding the self-sliding no-fit polygon: (a) two objects contact on a point; (b) first path of that the mover slides on the stator; (c) first two path of that the mover slides on the stator; (d) the self-sliding no-fit polygon.

of these two no-fit polygons is the position of object P2. The vector from the bottom-left vertex of object P0 to the object P1 is called the "first nesting vector", and the vector from the bottom-left vertex of object P0 to the object P2 is the second nesting vector as shown in Fig. 5(b). The third nesting vector is obtained by subtracting the first nesting from the second nesting vector. These three nesting vectors and their negative vectors will form a hexagon as shown in Fig. 5(c), which is called a "nesting crystal". Then, moving object P1 on the self-sliding no-fit polygon of P0 will result in different nesting vectors and different nesting crystals. The optimum nesting vectors are those vectors that cause the minimum nesting crystal area (Cheng and Rao, 1999).

*3.2.3. Third step—improvement*

Because the optimum nesting vectors in Fig. 5(b) might not be parallel to the sheet stock edges, there will be four corners that are not occupied by the nesting pattern after the second step. Rotating the nesting vector parallel to any axis will reduce this to two regions. Aligning three nesting vectors parallel to the $X$- and $Y$-axis, respectively, will result in six cases in this study to improve the nesting pattern.

After having introduced the methods a case study is given in the next section and the results are discussed.

## 4. Case study

The example used in this section was first introduced by Cheng and Rao (1997, 1999, 2000). The profiles of objects that will be nested are shown and numbered as in

Fig. 6. Objects 2 and 3 have the same profile; objects 7 and 8 also have the same profile. The coordinate values are shown with local coordinate systems in Table 1. The COGs are set as the origins of the local coordinate systems, and will be used to represent the positions of objects.

In the first step, the design variables are the coordinate values ($x_i$, $y_i$) and object orientations ($\theta_i$). There are 24 design variables in this case because there are eight objects and every object needs three design variables to represent its position and orientation. The cost function is

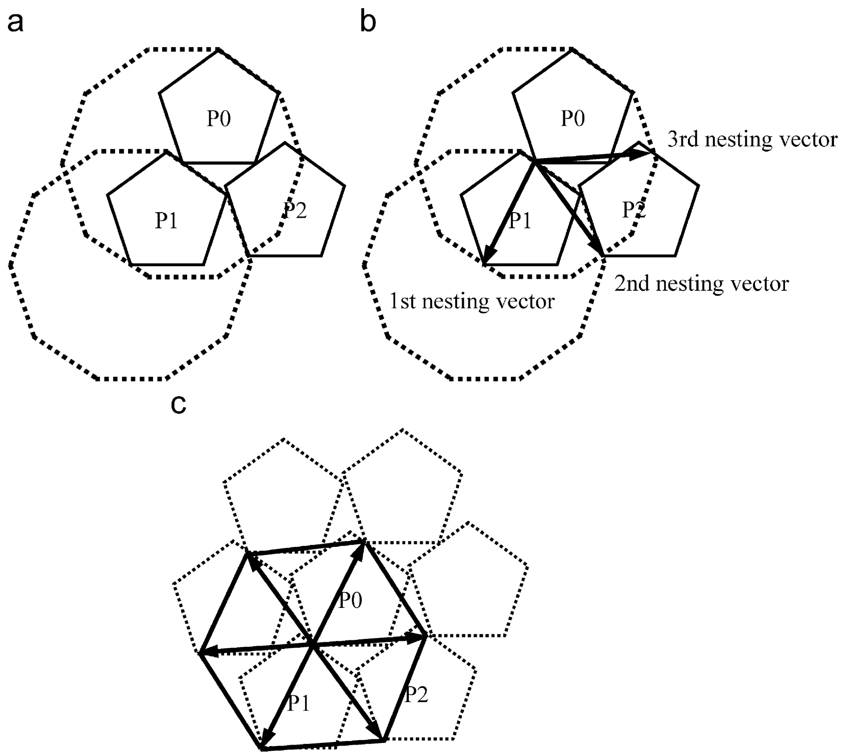


**Fig. 5.** The process of finding nesting vectors: (a) the object P1 on the self-sliding no-fit polygon of the object P0; (b) nesting vectors; (c) the nesting crystal.

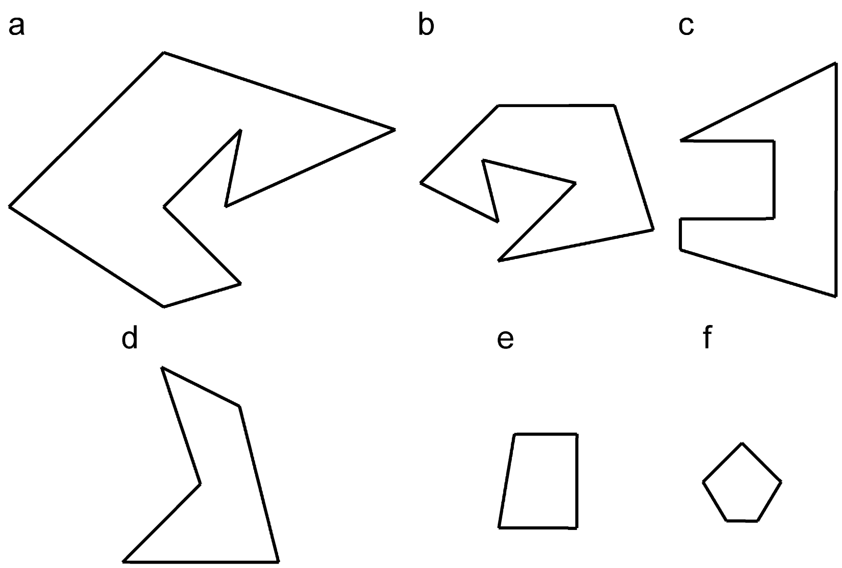


**Fig. 6.** Profiles of objects for cutting: (a) object 1; (b) objects 2 and 3; (c) object 4; (d) object 5; (e) object 6; (f) objects 7 and 8.

**Table 1**
Vertices of objects.

| Object no. | Vertices (x, y) |
|---|---|
| 1 | (−0.18, −1.77); (0.82, −1.47); (−0.18, −0.47); (0.82, 0.53); (0.62, −0.47); (2.82, 0.53); (−0.18, 1.53); (−2.18, −0.47) |
| 2 and 3 | (−0.7, −1.14); (1.3, −0.74); (0.8, 0.86); (−0.7, 0.86); (−1.7, −0.14); (−0.7, −0.64); (−0.9, 0.16); (0.3, −0.14) |
| 4 | (−1.32, −0.78); (0.68, −1.38); (0.68, 1.62); (−1.32,0.62); (−0.12, 0.62); (−0.12, −0.38); (−1.32, −0.38) |
| 5 | (−1.16, −0.94); (0.84, −0.94); (0.34, 1.06); (−0.66, 1.56); (−0.16, 0.06) |
| 6 | (−0.55, −0.58); (0.45, −0.58); (0.45, 0.62); (−0.35, 0.62) |
| 7 and 8 | (−0.2, −0.44); (0.2, −0.44); (0.5, 0.06); (0, 0.56); (−0.5, 0.06) |

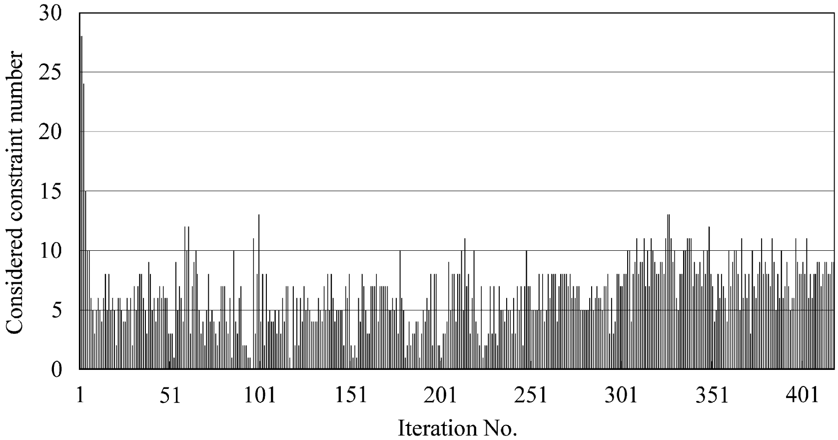


**Fig. 7.** Considered iteration number.

to minimise the summation distances between objects (as shown in Eq. (1)), and the 28 constraints are that no two objects may overlap (as shown in Eq. (3)). This means all objects have to be as close as possible without overlap. Even if there are many constraints in this case, the inactive constraints will be ignored in the active-set SQP solving process when using the maximum depth to consider the overlaps.

Deciding the initial solution is another problem in the optimisation process, and different initial solutions will lead to different local optimum solutions. The initial solution in this study is set by the concept of initially ignoring the constraints, and findings the best solution in the new unconstrained problem. Then the best solution of the unconstrained problem is set as the initial solution of the constrained problem. Thus, the initial solution is set as though all object positions are on the origin of the global coordinate system. This is because the best solution occurs when all objects overlap on the same position if the constraints are ignored. The object orientations are set as zero, i.e., the original orientation of objects, because they do not affect the cost function but can improve the constraints.

The object positions and orientation are adjusted to improve the constraints in the solving process, resulting in 418 iterations in this case. The number of considered constraints during the solving process is shown in Fig. 7. At the beginning, 28 constraints are used, i.e., all constraints are considered, but the number of considered constraints is later reduced. For example, there are six

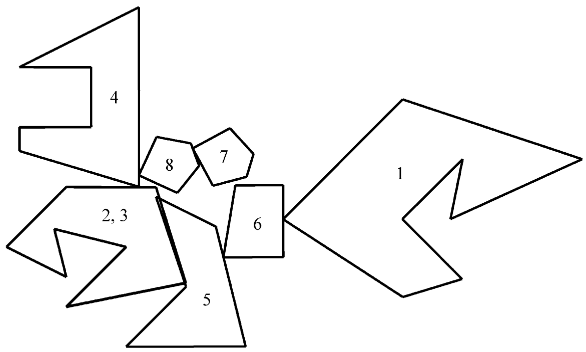


**Fig. 8.** The placing pattern of iteration 7.

**Table 2**
The data of the first step result.

| Object no. | X coord. value | y coord. value | Orientation (degree) |
|---|---|---|---|
| 1 | 0.305 | −2.808 | 6.922 |
| 2 | 2.723 | 1.155 | −49.388 |
| 3 | −1.196 | −0.911 | 52.126 |
| 4 | −0.138 | 0.985 | −3.821 |
| 5 | 1.737 | −0.717 | −11.513 |
| 6 | 0.973 | 0.707 | 5.641 |
| 7 | 0.451 | −0.904 | 92.524 |
| 8 | 1.088 | −0.346 | 106.560 |

pairs of objects that overlap one another in iteration 7. The placing pattern is shown in Fig. 8. This means that there are six considered constraint. The six pairs are objects 1 and 6, objects 2 and 3, objects 2 and 5, objects 3 and 5, objects 4 and 8, and objects 7 and 8. Objects 2 and 3 are almost overlapped completely, and look like one object in the figure. The maximum depths of the first, fifth, and sixth pair are very close to zero, and cannot be observed in the figure. If all constraints are considered in the solving process, there will be 11704 constraints ($28 \times 418$). But the summation of the number of considered constraints is 2670 by using the active-set SQP method with maximum depth overlap index. This amounts to a reduction of about 77.19% in this case.

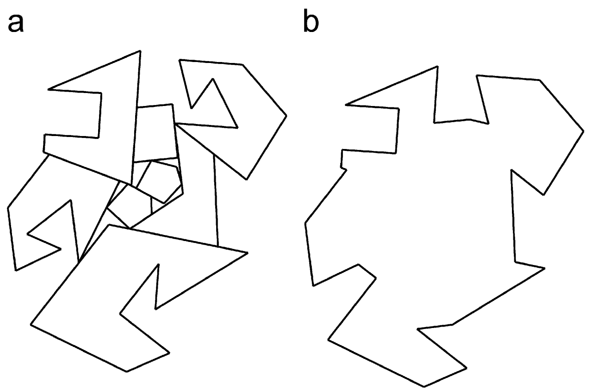


**Fig. 9.** Result of the first step: (a) placing pattern of the first step; (b) the cluster profile.

The result of the first step is shown in Table 2, and the *x*- and *y*-coordinate values are the position of the objects' COGs in the global coordinate system. The placing pattern of the first step is shown in Fig. 9(a), and the objects should be integrated before going into the second step. The profile of the cluster with a highly concave characteristic is shown in Fig. 9(b). This concave characteristic may be used in the second step.

The second step uses CNA to nest. The self-sliding no-fit polygon has to be found first. Then the position of the objects is adjusted to find the optimum nesting vectors as introduced above. The optimum nesting vectors are shown in Table 3. The nesting pattern with these nesting vectors has 60 clusters in a $50 \times 50$ sheet stock as shown in Fig. 10. $N_1$, $N_2$, and $N_3$ are the first, second, and third optimum nesting vectors, respectively. As shown in Table 3 and Fig. 10, the optimum nesting vectors are not parallel to the *X*- or *Y*-axis. Therefore, the nesting pattern can be improved in the third step.

In the third step, the three nesting vectors are aligned to the *X*- and *Y*-axis, respectively, as shown in Fig. 11. The

**Table 3**
Nesting vectors.

| | |
|---|---|
| First nesting vector | (1.917, −5.238) |
| Second nesting vector | (5.586, 1.060) |
| Third nesting vector | (3.669, 6.298) |

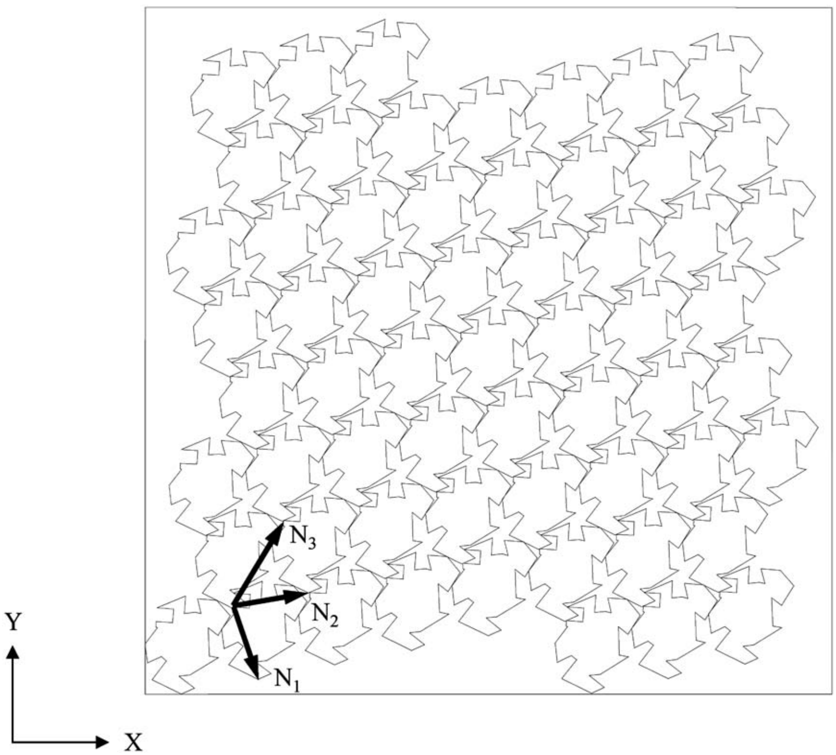


**Fig. 10.** The nesting pattern of the second step on the $50 \times 50$ sheet stock.

best case is when the third nesting vector is parallel to the $Y$-axis, resulting in 66 clusters in a $50 \times 50$ sheet stock.

For evaluating the proposed method, three kinds of sheet stocks were used, namely $50 \times 50$, $100 \times 100$, and $200 \times 200$. The cluster in the literature (Cheng and Rao, 2000) is shown in Fig. 12. Because the Genetic Algorithm is a controlled random method, the improvement step runs three times in every kind of sheet stock with the

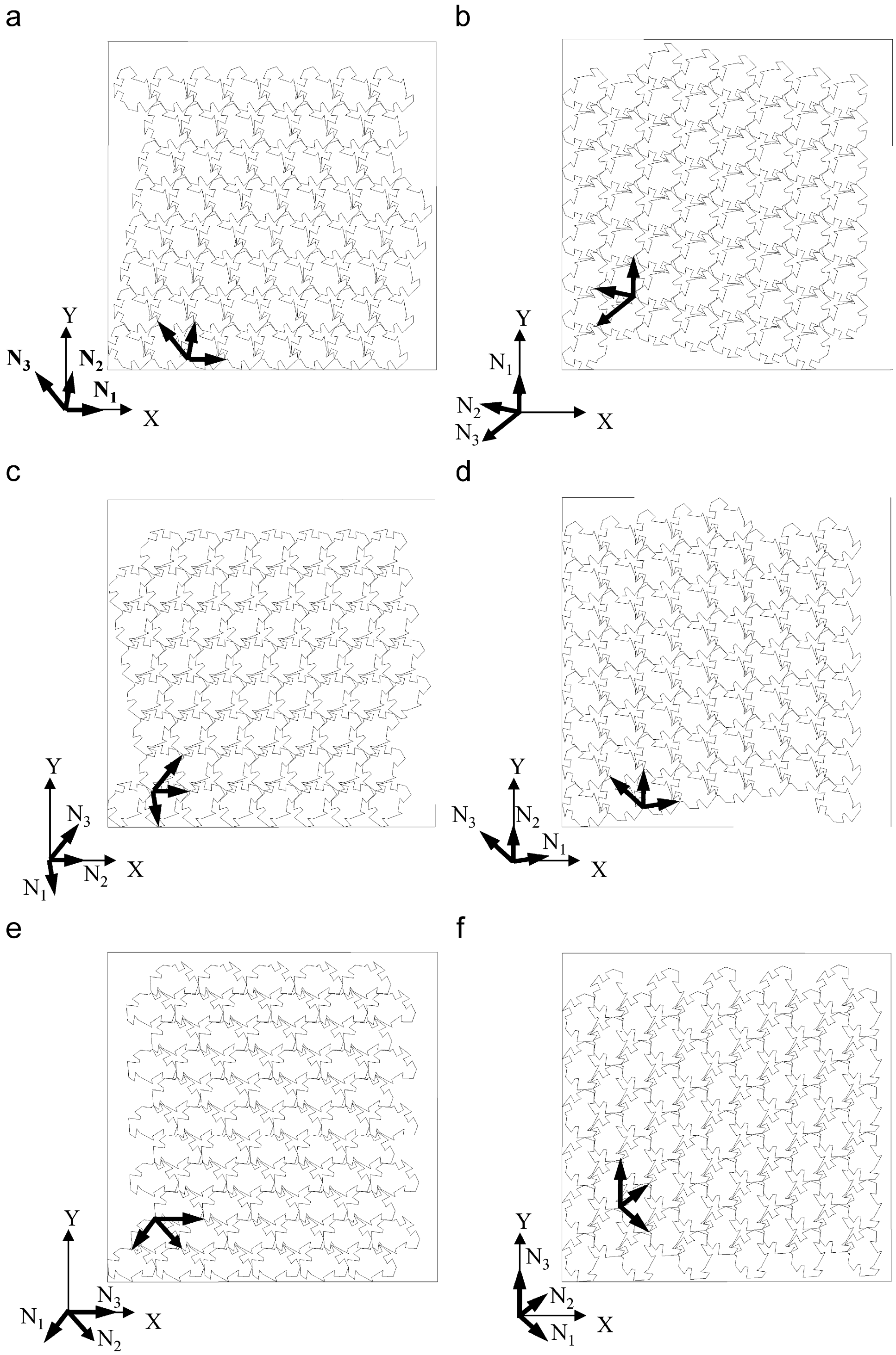


**Fig. 11.** Nesting patterns (a) $N_1$ parallels $X$-axis; (b) $N_1$ parallels $Y$-axis; (c) $N_2$ parallels $X$-axis; (d) $N_2$ parallels $Y$-axis; (e) $N_3$ parallels $X$-axis; (f) $N_3$ parallels $Y$-axis.

literature cluster, as shown in Table 4(a). The results of the proposed method with these sheet stocks are shown in Table 4(b). The comparison between these cases is shown in Table 4(c). After integrating the multi-polygon as a single object and nesting by CNA, there are 58 literature clusters in the $50 \times 50$ sheet stock, while the proposed method yields 60. Thus, the number of objects is improved by 3.45%. In the $100 \times 100$ and $200 \times 200$ sheet stock, the number of objects is improved by 5.28% and 4.91% respectively. However, after improving the nesting pattern by the Genetic Algorithm, there are up to 62 clusters in the literature method, while there are 66 clusters in this study after the third step. Thus, the number of objects is improved by 6.45%. The proposed method improves the number of objects in these cases by between three and six percent. Therefore, the proposed method has better results for rotatable objects and requires less calculation effort.

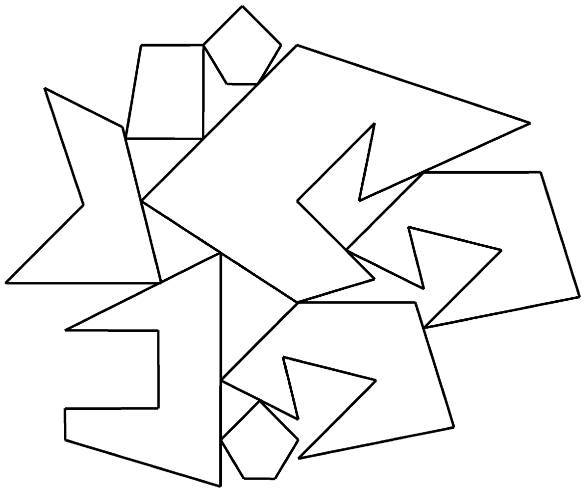

**Fig. 12.** The cluster of the literature.

For testing the efficiency of the method, this study uses other two cases called Dagli and Swim (http://paginas.fe.up.pt/~esicup/tiki-index.php). The clusters are shown in Fig. 13, and the information of the object can be found in the ESICUP website. The stock sizes are selected randomly, and the results are shown in Table 5 and 6. The results by using this method are better than the CNA with GA. Thus, the approach proposed in this study is a good method not only for one special case.

## 5. Conclusions

The cutting-stock problem is considered in many manufacturing industries, and the method for solving it can be divided into two parts: the overlap consideration method, and the placing strategy. The objects will not be rotatable if using a sliding technique to avoid overlap, while the calculation effort will be large if using real overlap area as the overlap index. Using binary matrices to represent objects can reduce the calculation effort, but the objects may be deformed after rotation. The placing strategies using different placing sequence in finding the best placement for the multi-polygon cutting-stock problem do not use gradient information or Hessian information. Therefore, this study proposes:

(1) Using the maximum depth as the overlap index instead of real overlap area to reduce the calculation effort.
(2) By using the maximum depth as the overlap index, the objects are not coded in binary matrices and they will not deform in different orientations.

**Table 4**
Results: (a) results of CNA with Genetic Algorithm; (b) results of the proposed method; (c) comparing the results.

| (a) | | | |
|---|---|---|---|
| The sheet stock size | $50 \times 50$ | $100 \times 100$ | $200 \times 200$ |
| CAN | 58 | 265 | 1140 |
| Improving run 1 | 60 | 269 | 1155 |
| Improving run 2 | 62 | 272 | 1150 |
| Improving run 3 | 62 | 277 | 1156 |

| (b) | | | | |
|---|---|---|---|---|
| The sheet stock size | | $50 \times 50$ | $100 \times 100$ | $200 \times 200$ |
| CAN | | 60 | 279 | 1196 |
| Paralleling *X*-axis | First nesting vector | 62 | 285 | 1215 |
| | Second nesting vector | 61 | 279 | 1226 |
| | Third nesting vector | 61 | 276 | 1216 |
| Paralleling *Y*-axis | First nesting vector | 62 | 286 | 1214 |
| | Second nesting vector | 62 | 280 | 1225 |
| | Third nesting vector | 66 | 284 | 1216 |

| (c) | | | |
|---|---|---|---|
| The sheet stock size | $50 \times 50$ | $100 \times 100$ | $200 \times 200$ |
| The best of proposed method | 66 | 286 | 1226 |
| The best of CNA with GA | 62 | 277 | 1156 |
| Improvement ratio | 6.45% | 3.25% | 6.06% |

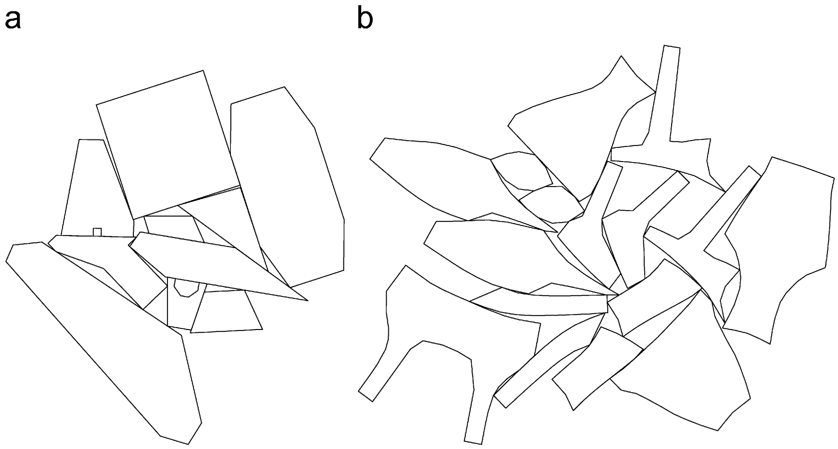


**Fig. 13.** The cluster of cases: (a) the cluster of case Dagli; (b) the cluster of case Swim.

**Table 5**
Results of case Dagli: (a) results of CNA with Genetic Algorithm; (b) results of the proposed method; (c) comparing the results.

| | | | | |
|---|---|---|---|---|
| (a) | | | | |
| The sheet stock size | | 851 × 1790 | 1681 × 1638 | 649 × 1490 |
| CNA | | 1042 | 1927 | 647 |
| Improving run 1 | | 1109 | 2030 | 687 |
| Improving run 2 | | 1099 | 2032 | 695 |
| Improving run 3 | | 1099 | 2026 | 687 |
| (b) | | | | |
| The sheet stock size | | 851 × 1790 | 1681 × 1638 | 649 × 1490 |
| CNA | | 1109 | 2045 | 689 |
| Paralleling *X*-axis | First nesting vector | 1131 | 2096 | 703 |
| | Second nesting vector | 1113 | 2060 | 690 |
| | Third nesting vector | 1124 | 2080 | 695 |
| Paralleling *Y*-axis | First nesting vector | 1146 | 2092 | 690 |
| | Second nesting vector | 1119 | 2082 | 687 |
| | Third nesting vector | 1153 | 2064 | 707 |
| (c) | | | | |
| The sheet stock size | | 851 × 1790 | 1681 × 1638 | 649 × 1490 |
| The best of proposed method | | 1153 | 2096 | 707 |
| The best of CNA with GA | | 1109 | 2032 | 695 |
| Improvement ratio | | 3.97% | 3.15% | 1.73% |

(3) The cutting-stock problem is formulated as a constrained optimisation problem, and can be solved by the active-set SQP method in the first step of the placing strategy. The active-set SQP method is a Hessian-based method and will improve the efficiency of searching for the optimum placing pattern.
(4) The summation of distances between objects is used as the cost function, and is more sensitive than using the necessary sheet stock area or the number of objects placed in the sheet stock. It is more suitable for the SQP method.
(5) The overlaps are the constraints of the cutting-stock problem, and the constraint consideration will be reduced by using the active-set SQP method with the maximum depth as the overlap index. This is because the constraints will be ignored if the overlap indices are less than zero, i.e., the maximum depths are less than zero and the constraints become inactive.
(6) The nesting pattern is improved easily in the third step of the placing strategy by aligning the three nesting vector with the *X*- and *Y*-axis, respectively, and choosing the best one as the final nesting pattern.

**Table 6**
Results of case Swim: (a) results of CNA with Genetic Algorithm; (b) results of the proposed method; (c) comparing the results.

| | | | | |
|---|---|---|---|---|
| (a) | | | | |
| The sheet stock size | | 144 202 × 124 029 | 48 817 × 71 275 | 69 834 × 95 666 |
| CAN | | 1259 | 223 | 451 |
| Improving run 1 | | 1277 | 230 | 462 |
| Improving run 2 | | 1282 | 229 | 471 |
| Improving run 3 | | 1277 | 230 | 468 |
| (b) | | | | |
| The sheet stock size | | 144 202 × 124 029 | 48 817 × 71 275 | 69 834 × 95 666 |
| CNA | | 1335 | 238 | 477 |
| Paralleling *X*-axis | First nesting vector | 1363 | 236 | 475 |
| | Second nesting vector | 1365 | 250 | 486 |
| | Third nesting vector | 1334 | 242 | 481 |
| Paralleling *Y*-axis | First nesting vector | 1336 | 243 | 484 |
| | Second nesting vector | 1330 | 249 | 484 |
| | Third nesting vector | 1337 | 237 | 497 |
| (c) | | | | |
| The sheet stock size | | 144 202 × 124 029 | 48 817 × 71 275 | 69 834 × 95 666 |
| The best of proposed method | | 1365 | 250 | 497 |
| The best of CNA with GA | | 1282 | 230 | 471 |
| Improvement ratio | | 6.47% | 8.7% | 5.52% |


## Acknowledgements

The authors would like to thank Professor C.H. Tseng of National Chiao Tung University for his helpful advice, and Dr. J.S. Luo, an associate professor of National United University, for his suggestions on this study.


## References


Arora, J.S., 1984. An algorithm for optimum structural design without line search. In: Atrek, E., Gallagher, R.H., Ragsdell, K.M., Zienklewiz, O.C. (Eds.), New Directions in Optimum Structural Design. Wiley, New York Chapter 20.

Arora, J.S., 2004. Introduction to Optimum Design, Second ed. Elsevier/ Academic Press, London.

Bennell, J.A., Dowsland, K.A., Dowsland, W.B., 2001. The irregular cutting-stock problem—a new procedure for deriving the no-fit polygon. Computers & Operations Research 28, 271–287.

Cheng, S.K., Rao, K.P., 1997. Quick and precise clustering of arbitrarily shaped flat patterns based on stringy effect. Computers & Industrial Engineering 33, 485–488.

Cheng, S.K., Rao, K.P., 1999. Concepts of neighborhood and universal compact yield towards achieving best pattern layout. International Journal of Production Research 37, 3643–3658.

Cheng, S.K., Rao, K.P., 2000. Large-scale nesting of irregular patterns using compact neighborhood algorithm. Journal of Materials Processing Technology 103, 135–140.

Dagli, C.H., Poshyanonda, P., 1997. New approaches to nesting rectangular patterns. Journal of Intelligent Manufacturing 8, 177–190.

Dowsland, K.A., Vaid, S., Dowsland, W.B., 2002. An algorithm for polygon placement using a bottom-left strategy. European Journal of Operational Research 141, 371–381.

Gomes, A.M., Oliveira, J.F., 2002. A 2-exchange heuristic for nesting problems. European Journal of Operational Research 141, 359–370.

Ismail, H.S., Hon, K.K.B., 1995. The nesting of two-dimensional shapes using genetic algorithms. Proceedings of the Institution of Mechanical Engineers Part B—Journal of Engineering Manufacture 209, 115–124.

Jakobs, S., 1996. On genetic algorithms for the packing of polygons. European Journal of Operational Research 88, 165–181.

Kershner, R.B., 1968. On paving the plane. The American Mathematical Monthly 75, 839–844.

Koroupi, F., Loftus, M., 1991. Accommodating diverse shapes within hexagonal pavers. International Journal of Production Research 29, 1507–1519.

Liao, W.C., 1990, Integrated software for multifunctional optimization. Master Thesis, National Chiao Tung University, Taiwan, ROC.

Lim, O.K., Arora, J.S., 1986. An active set RQP algorithm for engineering design optimization. Computer Methods in Applied Mechanics and Engineering 57, 51–65.

Petridis, V., Kazarlis, S., 1994. Varying quality function in genetic algorithms and the cutting problem. In: Proceedings of the IEEE Conference on Evolutionary Computation, pp. 166–169.

Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. European Journal of Operational Research 183, 1109–1130.

Wong, W.K., Leung, S.Y.S., 2008. Genetic optimization of fabric utilization in apparel manufacturing. International Journal of Production Economics 114, 376–387.

Yu, M.T., Tseng, C.H., 2005. A study of one type object nesting problem. In: Proceedings of the Eighth National Conference on the Design of Mechanisms and Machines, pp. 578–584.