



ELSEVIER

Journal of Materials Processing Technology 103 (2000) 135–140

Journal of
**Materials
Processing
Technology**

www.elsevier.com/locate/jmatprotec

Large-scale nesting of irregular patterns using compact neighborhood algorithm

S.K. Cheng, K.P. Rao*

*Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong*

Abstract

The typical nesting technique that is widely used is the geometrical tilting of a single pattern or selected cluster step by step from the original position to an orientation of 180° , i.e. orthogonal packing. However, this is a blind search of best stock layout and, geometrically, it becomes inefficient when several pattern entities are involved. Also, it is not highly suitable for handling patterns with a range of orientation constraints. In this paper, an algorithm is proposed which combines the compact neighborhood algorithm (CNA) with the genetic algorithm (GA) to optimize large-scale nesting processes with the consideration of multiple orientation constraints. © 2000 Elsevier Science S.A. All rights reserved.

Keywords: Cutting stock problem; Nesting; Compact neighborhood algorithm; Genetic algorithm; Orientation constraints

1. Introduction

The cutting stock problem is of interest to many industries like garment, paper, ship building, and sheet metal industries. Gilmore and Gomory [7] have initiated the research work to solve the rectangular cutting stock problem by using linear programming. For the irregular case, Adamowicz [1] attempted to use a heuristic approach which divides the problem into two sub-problems, called clustering and nesting. Clustering is to specify a collection of patterns that fit well together before nesting onto a given stock. Nesting of patterns or clusters can be broadly divided into two broad categories, namely, small-scale and large-scale. The difference between them is the level of duplication of the cluster on the given stock. For small-scale nesting, we only need to find the inter-orientation relationship between the selected cluster and the given stock [4]. However, the problem becomes more complicated for large-scale nesting since the inter-space relationship between the duplicated clusters should also be considered. Traditionally, two basic techniques are popularly used for generating this type of nesting: “hexagonal approximation” and “orthogonal nesting”.

A typical pattern, shown in Fig. 1a, with both concave and convex features, is selected to explain these techniques. The

pattern contour is plotted with the help of a digitizer, as shown in Fig. 1b, and has an area (A_p) of 74.44 sq. units. In the “hexagonal approximation” suggested by Dori and Ben-Bassat [5], the pattern is first approximated using a convex polygon which is further approximated by another convex polygon with fewer number of entities until an hexagonal enclosure is obtained, as shown in Fig. 1c. The hexagon is then paved on a given stock with no overlapping of the former [6]. The resultant layout generated by use of this technique is given in Fig. 1e. It is readily evident that the technique is not highly efficient due to the poor approximation performance, especially in the case of highly irregular patterns. Another problem is that the pattern or cluster can assume two positions only (0 or 180°), with no exploitation or consideration of other permissible range of orientations.

In the second technique, used by Nee [9], the nesting process is achieved by approximating a single pattern/cluster by a rectangle as shown in Fig. 1d. This rectangle is then duplicated in an orthogonal way, resulting in the layout shown in Fig. 1f. This technique can be easily applied when there are no- or partial-orientation constraints, i.e. the single pattern or cluster can rotate within a certain range while fitting it on the stock. Like the hexagonal approximation, the main disadvantage of this approach is that the algorithm’s performance is highly dependent on the shape of patterns. Moreover, in the case of multiple orientation constraints, the

* Corresponding author. Tel.: +852-2788-8409; fax: +852-2788-8423.
E-mail address: mekprao@cityu.edu.hk (K.P. Rao)

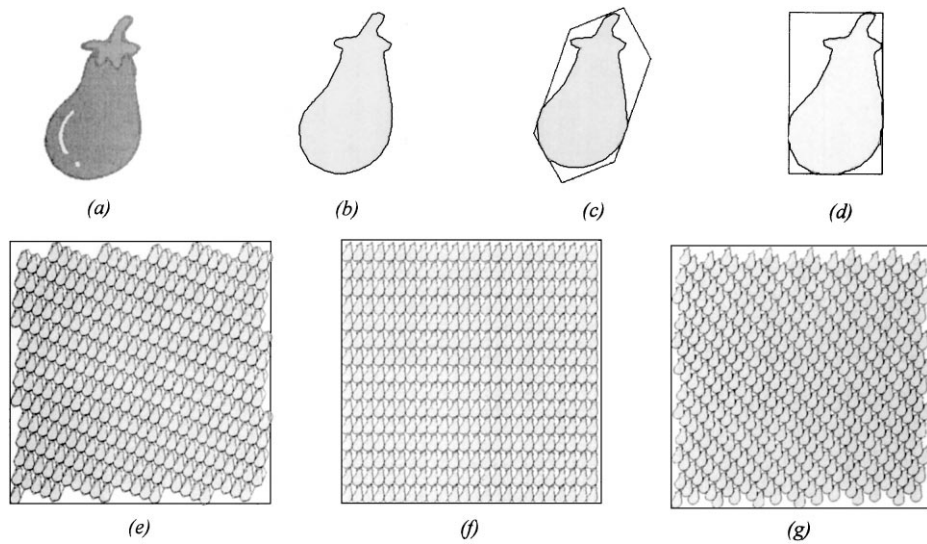


Fig. 1. (a) The chosen flat pattern for demonstrating the working principle of CNA algorithm; (b) pattern contour obtained by digitizer; (c) hexagonal approximation; (d) orthogonal approximation; (e) layout generated by using hexagonal approximation yielding a stock utilization of 60.05%; (f) layout generated by using orthogonal approximation yielding a stock utilization of 67.14%; and (g) layout generated by using CNA yielding a stock utilization of 74.10%.

time taken to estimate a suitable rotation angle for the patterns is always much longer.

In order to increase the accuracy and speed of nesting, Cheng and Rao [4] proposed a compact neighborhood algorithm (CNA) that considers the relationship between the number of neighbors and the sharing space between them. Fig. 1g shows a typical layout generated using CNA which normally yields higher packing density when compared with the orthogonal and hexagonal approximations. However, CNA, in its present form, has been mainly designated for nesting of patterns with the consideration of full orientation constraints, and is not ideal for situations where more freedom is available in the orientation of patterns.

This study is aimed at improving the flexibility of CNA by incorporating the available freedom in the orientation of patterns and a genetic algorithm (GA) that follows natural rules to optimize the generated layouts. The new technique is translated into a computer program written in C++

object-oriented language. The new algorithm can handle the problem of nesting two-dimensional flat patterns of any shape containing line segments and arcs. With the help of a typical example, the enhanced capabilities of CNA and the associated computer program will be demonstrated in this paper.

2. Description of compact neighborhood algorithm (CNA)

A CNA [4] tracks the characteristics of the evolving neighborhoods when the patterns are moved to form different arrangements, as summarized schematically in Fig. 2a–c. As the shear displacement increases, the upper and lower neighbors tend to collapse due to the change in crystallization directions. Finally, a most compact structure and a numerical value for material yield, called “universal compact utilization (UCU)”, can be obtained. No matter

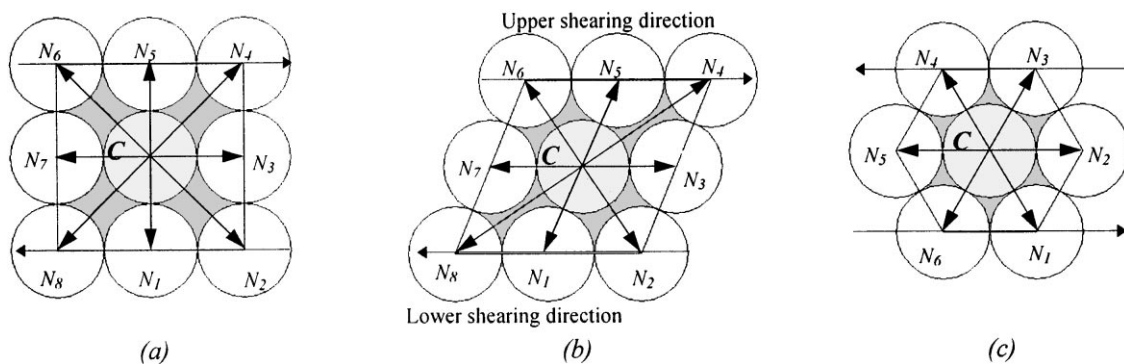


Fig. 2. Typical neighborhood structures for circular patterns — (a) formation of orthogonal packing unit cell with $N_n=8$ and $A_u=16r^2$; (b) shearing of layers leading to sheared orthogonal packing; and (c) best compact structure with hexagonal packing unit cell with $N_n=6$ and $A_u = 6\sqrt{3}r^2$, where A_u is the area of a unit cell, r the radius of circular pattern and N_n is the number of neighbors to construct the unit cell.

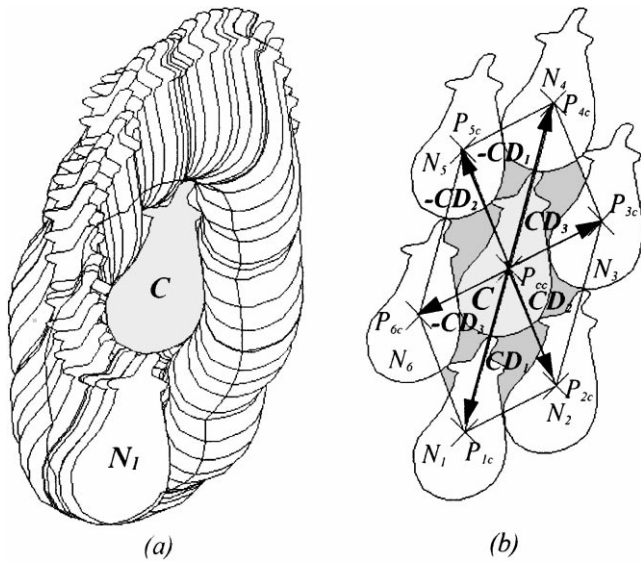


Fig. 3. (a) Steps involved in the generation of self-sliding path to create a neighborhood; and (b) optimal neighborhood structure with hexagonal packing unit cell with a UCU of 83.07%.

whether the pattern can be rotated or not, UCU indicates the upper limit of yield that may be possible with any chosen stock and hence can be regarded as an index for stopping criteria in the nesting process.

The main steps involved in finding the compact neighborhood are: (1) generating a “self-sliding path” or a no-fit-polygon (NFP) [1], as shown in Fig. 3a, which guides the relative movement between two patterns with the consideration of no overlapping; and (2) defining the crystallization directions, as shown in Fig. 3b, that provide essential data for building the whole neighborhood by filling the given stock during large-scale nesting.

3. Proposed algorithm for large-scale nesting

The proposed techniques of enhancing the capabilities of CNA by taking advantage of a genetic algorithm are dealt in this section. A flat pattern can be divided into entities of line segments and arcs. Polygonal representation methods [2] can be used to represent both concave and convex arcs as sets of straight lines. The actual number of lines is dependent on the required accuracy level. Also, clearance or offset generation is an essential step that contributes towards the success of CAD/CAM technology. An algorithm to generate the required offset, called “three point island tracing (TPIT)” technique [2], is incorporated in the present nesting system.

3.1. CNA for large-scale nesting

In the previous section, we have already mentioned the basic steps involved in obtaining the best compact neighborhood, as shown in Fig. 3b. Our next concern is the determination of the best position to place the first pattern and

expand this structure to fill the entire stock. For nesting of patterns with full orientation constraint, it is only necessary to decide a “nesting vector CD_n ” that defines where the neighborhood should be translated around the given stock. However, in the case of nesting of patterns with limited or no orientation limitations, the problem becomes more complicated due to an increase in the possible combinations that we need to consider. In this case, the first step — which is global with or without orientation limitations — is to translate the neighborhood to an arbitrary position inside the given stock, i.e. defining a vector CD_n . Afterward, a “nesting angle θ_n ” is to be determined so that a good orientation is selected for the neighborhood to grow. All the required geometrical operations are summarized in Fig. 4.

It is critical to optimize CD_n and θ_n which can finally lead to a most compact neighborhood structure. It is believed that there are no unique mathematical steps to calculate these parameters for any type of stock. In addition, we cannot accept an exhaustive search because of the constraints posed on computation time, especially in the case of nesting of patterns with too long a computation time, especially while nesting patterns with many entities and concave features. Hence, in this study, a recent popular optimization technique, called GA, is applied. The main principle is provided in the following section.

3.2. GA for optimizing layouts

GA [8] maintains a population of candidate problem solutions. Based on their performance, the fittest of these solutions not only survive, and, analogous to sexual reproduction, exchange information with other candidates to form a new generation. Before starting any genetic operation, one needs to define the fitness function and the coding method. As mentioned earlier, the goal in nesting of patterns is to reduce the scrap by fitting the clusters together so that they occupy a minimum area. To represent the compactness of a particular layout, one can be confident that the most direct way is to relate it with the stock yield

$$f(x, y) = p = \frac{y}{x} \quad (1)$$

where x is the area of the given stock and y the total area of the patterns that could be cut out from the given stock.

Coding can directly and indirectly influence the optimization process. This is because our main concern is how to fix the translation position (i.e. nesting vector CD_n) and the degree of rotation (i.e. nesting angle θ_n). They are thus selected as the coding parameters that guide the properties (i.e. corresponding to natural chromosomes) for exchange in the genetic operators of cross-over and mutation.

3.3. The genetic operators

As proposed by Holland et al. [8], the GA aims at optimizing the solution by mimicking nature’s evolutionary

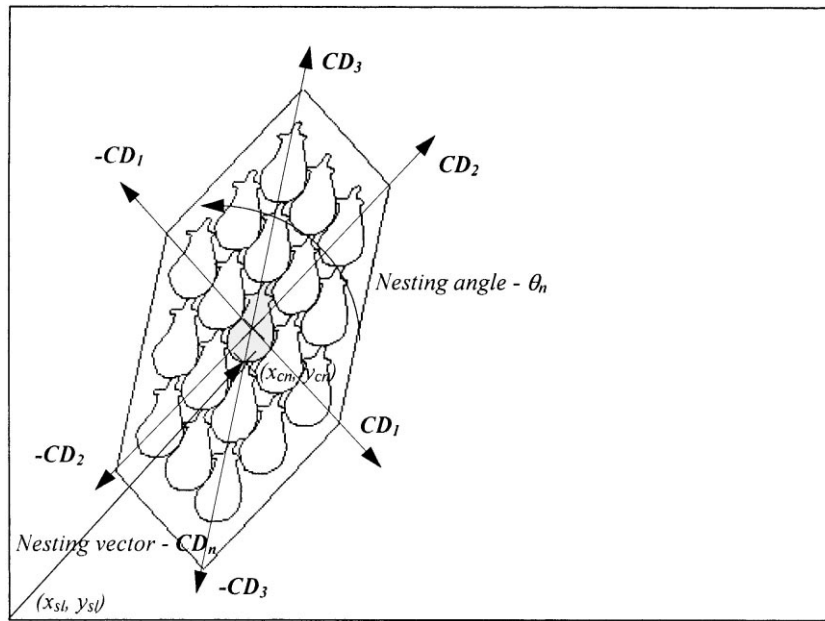


Fig. 4. Translation of the neighborhood to a pre-defined position with nesting vector CD_n and subsequent rotation involving a nesting angle θ_n .

process. Like human beings a typical GA contains the following genetic operators.

3.3.1. Initialization

At the beginning, a population of i genetic solutions (i.e. layouts) are generated by randomly selecting values for all the parameters (i.e. nesting angle θ_n and nesting vector CD_n). In real industrial applications, there are usually situations that limit the rotation of patterns freely. For example, in the design of progressive dies, there are many restricted orientation regions as a result of the cost of setting corresponding pilots, limitations of bending angle for subsequent sheet metal operations, and the like. Fig. 5 shows

two typical patterns with different restricted and unrestricted orientation regions. As a result, the positions that these patterns could assume are limited to two limited ranges in the present cases, as shown in the same figure. Hence, after the random generation of a nesting angle θ_n , the following inequality check has to be satisfied:

$$\theta_{pj,kl} \leq \theta_n \leq \theta_{pj,ku} \quad \text{with } j, k \in I, 1 \leq j \leq N \quad \text{and} \quad 1 \leq k \leq C_{pj} \quad (2)$$

where N and C_{pj} are the number of patterns to be nested (i.e. p_1, p_2, \dots, p_N) and the number of restricted orientation regions of pattern p_j , respectively.

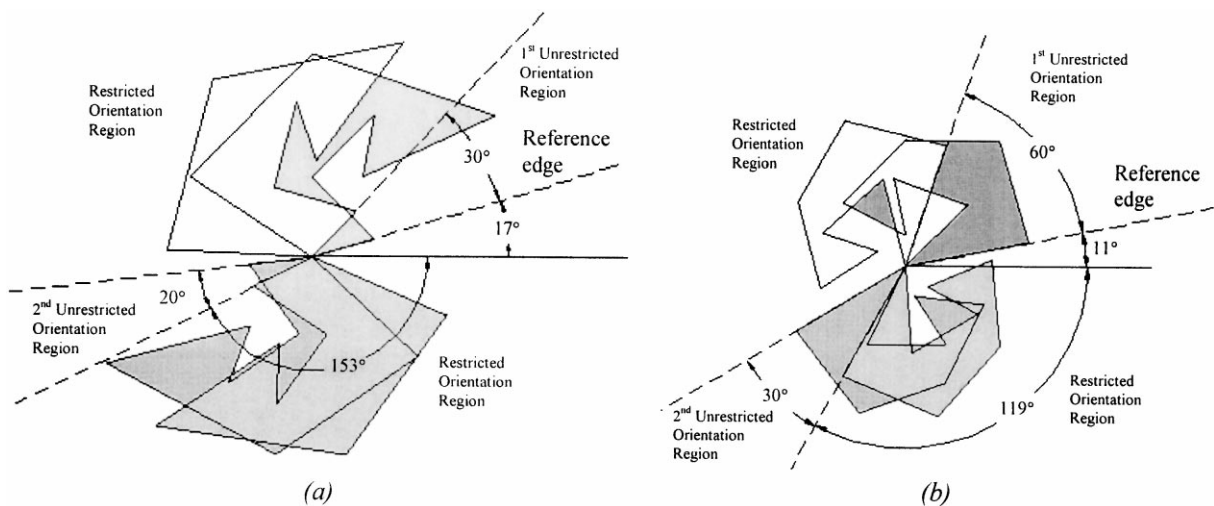


Fig. 5. Rotation constraints with respect to the lower bound of the first unrestricted region — (a) stationary pattern (p_1): $\theta_{p1,1l}=0^\circ, \theta_{p1,1u}=30^\circ, \theta_{p1,2l}=170^\circ, \theta_{p1,2u}=190^\circ$; and (b) movable pattern (p_2): $\theta_{p2,1l}=0^\circ, \theta_{p2,1u}=60^\circ, \theta_{p2,2l}=200^\circ, \theta_{p2,2u}=230^\circ$.

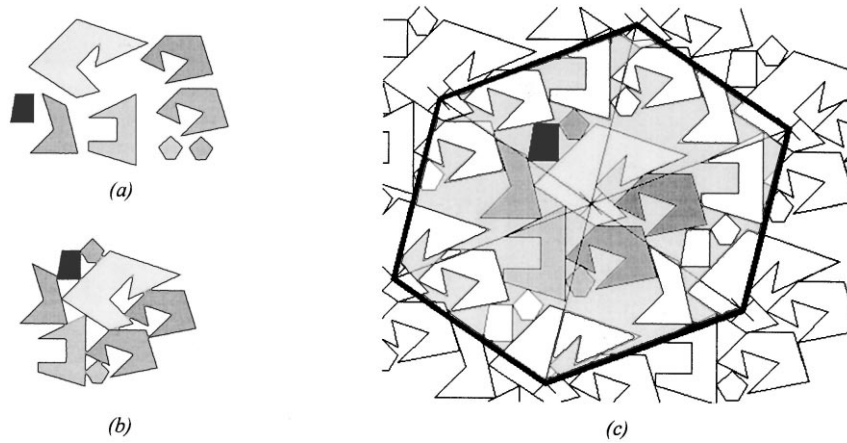


Fig. 6. Proposed strategy for solving the cutting stock problem with multiple patterns — (a) patterns to be clustered and nested; (b) cluster generated by using sliding technique; and (c) an hexagonal packing unit cell generated by CNA.

3.3.2. Fitness calculation

By following the fitness function, each generated layout (i) is assigned with a fitness value (p_i) that decides the layout compactness. This value is used in the subsequent operations for determining the candidate that should be selected in the step of reproduction.

3.3.3. Reproduction

According to the assigned fitness values p , every individual has a specific chance to be selected for reproduction in the subsequent weighted random selection process. It is important that better individuals, i.e. higher values of p , have better chance of being selected than less suitable individuals.

3.3.4. Cross-over

This is the most important step that determines members in the next generation. The reproduced individuals, say layouts i and j , are crossed by using the cross-over operator, which is referred to taking average of their nesting angles θ_n or vectors CD_n . Each new value of θ_n is subject to the inequality condition (Eq. (2)) to check the availability.

3.3.5. Mutation

In fact, there are many approaches for implementing this operator. In the current system, it is basically achieved through an algorithm that rotates the neighborhood randomly by 15° with a probability of $1 - 0.5(p_i + p_j)$.

3.4. Large-scale nesting with multiple patterns

For the cutting stock problem with multiple patterns, the algorithm mentioned above is still valid. To decrease the huge search space, a clustering process is first operated to collect the patterns. For example, the eight patterns shown in Fig. 6a are collected together to form a cluster shown in Fig. 6b by using the stringy sliding technique [3]. After clustering, a grouping process is operated to remove all internal boundaries so that a much faster computation speed can be achieved during the subsequent steps. By repeating the same techniques mentioned above, the most compact neighborhood structure, shown in Fig. 6c, can be finally obtained, which is now suitable for large-scale nesting.

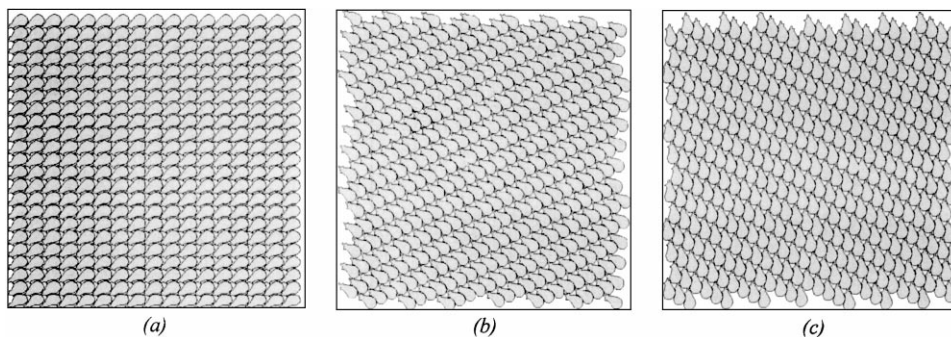


Fig. 7. (a) The layout generated by orthogonal approximation with a stock utilization of 72.8%; (b) the best layout among the initial population using CNA with a stock utilization of 73.9%; and (c) the best layout obtained after five generations using CNA with a stock utilization of 74.3%.

4. Case study

To test the efficiency of the proposed techniques for use in large-scale nesting, the results obtained are compared with those obtainable with the traditional nesting technique, i.e. orthogonal approximation, used by Nee [9] in terms of stock utilization (% SU). A stock size of 200×200 units is used here. Supposing that we do not have any orientation limitations imposed on patterns while nesting, a typical layout generated by using orthogonal approximation and two other layouts obtained by the proposed technique are shown in Fig. 7a–c, respectively. As illustrated, the proposed technique can give a better solution than the traditional one even with the initial population. Besides, it has been found that the solution can quickly reach the most optimal stage within an acceptable number of generations. This has been particularly true in those cases when the size of stock is large when compared with that of patterns. The use of the GA to optimize the values of nesting vector CD_n and nesting angle θ_n becomes more important as the size of stock decreases.

5. Conclusions

An attempt has been made to improve the effectiveness of the clustering and nesting processes, with a rigorous consideration of orientation constraints of the patterns. A CNA, which was recently proposed for generating stock layouts when there are full orientation constraints, i.e. the nested patterns could not be rotated at all, is further developed to solve the problem of multiple orientation constraints. CNA, with its concept of UCU, guides the user to attain near optimum layouts. By using CNA in conjunction with a GA, the best solution can be mathematically obtained, and the technique is highly suitable for large-scale nesting purposes.

The feasibility of the proposed algorithm has been verified in terms of the stock utilization.

Acknowledgements

The authors would like to acknowledge the support received from the City University of Hong Kong, through Strategic Grant No. 7000442, and the Materials Research Centre.

References

- [1] M. Adamowicz, The optimum two-dimensional allocation of irregular, multiple-connected shapes with linear, logical and geometric constraints, Ph.D. Thesis, Department of Electrical Engineering, New York University, 1969.
- [2] S.K. Cheng, K.P. Rao, Automatic generation of clearance for flat patterns, in: Proceedings of the Seventh International Manufacturing Conference in China, Harbin, Vol. 2, 1995, pp. 191–197.
- [3] S.K. Cheng, K.P. Rao, Quick and precise clustering of arbitrarily shaped flat patterns based on stringy effect, *Comput. Ind. Eng.* 33 (1997) 485–488.
- [4] S.K. Cheng, K.P. Rao, An intelligent stock layout technique for irregular patterns with multiple orientation constraints, in: Proceedings of the Fourth International Conference on Manufacturing Technology in Hong Kong, Hong Kong, November–December 1997 (CD-ROM; Computer Integrated Manufacturing Section).
- [5] D. Dori, M. Ben-Bassat, Circumscribing a convex polygon by a polygon of fewer sides with minimal area addition, *Comput. Vision Graphics Image Process.* 24 (1983) 131–159.
- [6] D. Dori, M. Ben-Bassat, Efficient nesting of congruent convex figures, *Image Process. Comput. Vision* 27 (1984) 228–235.
- [7] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem — Part 1, *Oper. Res.* 9 (1961) 724–746.
- [8] J.H. Holland, K.J. Holyoak, R.E. Nisbett, P.R. Thagard, *Induction: Processes of Inference, Learning and Discovery*, MIT Press, Cambridge, MA, 1986.
- [9] A.Y.C. Nee, A heuristic algorithm for optimum layout of metal stamping blanks, *Ann. CIRP* 33 (1984) 317–320.