AN EDGE MATCHING APPROACH FOR TWO-DIMENSIONAL IRREGULAR SHAPED CUTTING STOCK PROBLEMS

DERYA AKBULUT

JUNE, 2012

Title of Thesis : **An Edge Matching Approach for Two Dimensional Irregular Shaped Cutting Stock Problems.**

Submitted by : **Derya AKBULUT**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTINOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Levent KANDİLLER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Sciences.

Asst. Prof. Dr. Engin TOPAN	Prof. Dr. Levent KANDİLLER

Examination Date: 27.06.2012

Examining Committee Members

Prof. Dr. Ömer KIRCA	(METU)
Prof. Dr. Levent KANDİLLER	(Çankaya Üni.)
Asst. Prof. Dr. Abdül Kadir GÖRÜR	(Çankaya Üni.)
Asst. Prof. Dr. Nureddin KIRKAVAK	(Çankaya Üni.)
Asst. Prof. Dr. Engin TOPAN	(Çankaya Üni.)

STATEMENT OF NON PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by thesis rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Derya AKBULUT

Signature

Date : 27.08.2012

ABSTRACT

AN EDGE MATCHING APPROACH FOR TWO DIMENSIONAL IRREGULAR SHAPED CUTTING STOCK PROBLEMS

AKBULUT, Derya

M.Sc., Department of Industrial Engineering

Supervisor: Levent KANDİLLER, Ph.D.

Co-Supervisor: Engin TOPAN, Ph.D.

June 2012, 81 pages

In this thesis, a two-dimensional irregular shape cutting stock problem is considered, in which a number of irregular shaped pieces are cut out of rectangular stock sheets so that waste of stock to be minimized. This is an operational problem commonly observed in metal cutting and textile industries. In the literature, there are many algorithms proposed to find optimal or suboptimal solutions for the problem. Since the problem is NP-hard, heuristic approaches predominate among the solution methodologies. In this study, a non-linear mixed integer mathematical model formulation is developed and it is tested for small sized problems. For larger scale problems, an edge matching approach is proposed to generate cutting patterns. The approach is based on positioning of the pieces in such a way that their most fitting edges are aligned together or to the borders. In this way, the total scrap and the cutting operations are minimized. On the contrary to the most of the solution methodologies in the literature, the method enables rotating pieces by any angle during the alignment process and further more it is applicable for irregular shaped stock materials. The developed procedure is tested against the traditional cutting stock approaches using benchmark test problems reported in the literature. It is found that our procedure outperforms for a large portion of these benchmark problems.

Keywords*:* Irregular shape cutting stock problem, edge matching.

ÖZ

İKİ BOYUTLU VE DÜZGÜN OLMAYAN ŞEKİLLİ STOK KESİM PROBLEMLERİ İÇİN KENAR EŞLEME YAKLAŞIMI

AKBULUT, Derya

Yüksek Lisans, Endüstri Mühendisliği Ana Bilim Dalı

Tez Yöneticisi: Prof. Dr. Levent KANDİLLER

Yardımcı Tez Yöneticisi: Yrd. Doç. Dr. Engin TOPAN

Haziran 2012, 81 sayfa

Bu tez kapsamında, iki boyutlu düzgün olmayan şekiller için stok kesim problemi çalışılmıştır. Söz konusu problem, metal ve tekstil endüstrisinde yaygın olarak karşılaşılan, dikdörtgen şeklindeki bir hammaddeden düzensiz şekildeki parçaları en az artık malzemeye yol açacak şekilde keserek çıkartmaya dayanmaktadır. Literatürde, problem için optimal ya da yaklaşık-optimal pek çok çözüm algoritması bulunmaktadır. Problemin NP-Zor oluşundan dolayı, sezgisel yöntemler literatürde hakim durumdadır. Bu çalışmada, problem doğrusal olmayan karmaşık tamsayılı bir formulasyonla modellenerek küçük boyutlu problemler için test edilmiştir. Büyük boyutlu problemler içinse, bir kenar eşleme yaklaşımı önerilmiştir. Kesim kalıpları oluştururken parçalar kenarları birbirine yaslanacak şekilde konumlandırılmış, böylelikle artık malzemenin yanında kesim uzunluğunun, dolayısıyla işleme zamanının, en azlanması hedeflenmiştir. Kenar yaslama işlemi, birçok çözüm yönteminin aksine, parçaların istenen açıyla döndürülmesine izin vermekte ve ayrıca düzensiz şekilli stok malzemeleri için de kullanılabilmektedir. Geliştirilen yaklaşım, literatürde sunulan problem setleri kullanılarak mevcut çözüm yöntemleriyle kıyaslanmış ve bir çok problem için daha iyi sonuçlar verdiği görülmüştür.

Anahtar Kelimeler: Düzensiz şekilli stok kesim problemi, kenar eşleme.

ACKNOWLEDGEMENTS

First and foremost I would like to thank my advisors Prof. Dr. Levent Kandiller and Asst. Prof. Dr. Engin Topan for their motivating guidance throughout my M.Sc. study, for their being accessible no matter how busy schedule they had and for their patience during all stages of thesis writing.

I am grateful to my examining committee members, Prof. Dr. Ömer Kırca, Asst. Prof. Dr. Nureddin Kırkavak, and Asst. Prof. Dr. Abdül Kadir Görür for accepting to be on my committee and for their valuable advice and suggestions on my research.

I would also like to thank Asst. Prof. Dr. Abdül Kadir Görür, for his time, interest and guidance during the coding process of my algorithm.

I would like to thank all my colleagues at the IE department, especially Funda Karabak, Atıl Kurt, Miray H. Aslan Yıldırım and İpek Seyran Topan for the technical and moral support that they had provided to me.

I would like to thank all my FRIENDs for their invaluable friendships and all the support that they had provided to me whenever I need throughout my M.Sc. study.

Finally, I would like to thank my family, Nazım Akbulut, Şükran Akbulut and Çağım Akbulut, for every moment that they spent for me throughout my whole life, and for all the supports that they had provided to me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
2DCSP	Two Dimensional Cutting Stock Problem
BL	Bottom Left
BP	Best Published
CCW	Counter Clockwise
CW	Clockwise
C&P	Cutting and Packing
CPU	Central Processing Unit
ESICUP	Euro Special Interest Group on Cutting and Packing
GHz	Gigahertz
LAF	Largest Area First
LPF	Longest Perimeter First
NFP	No Fit Polygon
NP	Non-deterministically Polynomial
RAM	Random Access Memory

CHAPTER 1

INTRODUCTION

In industrial cutting process there exists a large stock material which can be in the form of metal sheets, fabric rolls, glass plates, paper rolls, etc. and smaller parts to be generated by cutting that stock material into pieces. The objective is to determine a cutting pattern for the pieces such that the scrap of the stock material used is minimized. A cutting pattern stands for the location information for the pieces on the stock material, and shows where to cut each piece. During this placement operation, the pieces are tried to be packed on the stock material. Thus, cutting stock problems are dually related with the packing problems.

In manufacturing environment, it is not always possible to generate a cutting pattern that utilizes the raw material with 100 percent efficiency. This means some portion of the raw material is generally scrapped. Sometimes, the cost of these scrap amounts reaches up to unacceptable levels. The design of cutting patterns directly affects the scrap amount as well as the unit cost of the produced pieces. Hence, determining these patterns becomes a critical decision for the manufacturers. The problem is a widely studied research topic. The first known formulation for a cutting stock problem was developed by Kantorovich (1939). Since then, many researchers have worked on several different aspects of the problem.

In our study, a mixed integer non-linear programming model formulation is presented for the problem, which is one of the first in the literature. The model minimizes the stock length for a given width, to maximize the material utilization. Furthermore, for larger scale problems, a heuristic approach is proposed since it becomes intractable to solve the problem by using non-linear integer programming solvers. Our heuristic approach is based on the proper alignment of the edges of different pieces together to have higher utilizations. By using the approach, the length of cutting paths obtained by cutting patterns is reduced. Using cutting path lengths as a performance measure is also new to the literature. Finally, the cutting patterns generated by our approach are tested using the benchmark problems in the literature, and the results are presented. The results of the experiments reveal that our edge matching heuristic yields quite satisfactory results.

The organization of the manuscript is as follows: Section 2 describes the related work about the problem in the literature. In Section 3, the problem is defined and the mathematical formulation of the problem is presented. This section includes development of optimal solution techniques and lower bound calculations. In Section 4, the proposed heuristic approach is presented. Section 5 presents the performance results obtained by the procedure. Finally, in Section 6, the conclusion and future extensions are presented.

CHAPTER 2

LITERATURE REVIEW

In this section, a literature review for the cutting and packing problems are presented within three main streams. In Section 2.1, based on the typology suggested by Dyckhoff (1990), a classification for the cutting and packing problems is presented. Using this classification, a discussion about the position of our study in the literature is provided. Section 2.2 is devoted to introduce critical decisions of *irregular shape pattern generation* algorithms, to provide a better understanding on the problem. Section 2.3 focuses on the *solution procedures* reported in the literature. These are based on different combinations of decisions that are described in Section 2.2

2.1 Classification of the Problem

The cutting stock problems are classified under the generic title of “Cutting and Packing” problems (C&P). The trim loss minimization problem, marker marking problem, bin or strip packing problem, pallet or container loading problem, nesting problems are all among the special topics that belong to the area of C&P. This classification is based on the typology suggested by Dyckhoff (1990). Dyckhoff considers the main topic as C&P problem, and positions subclasses with respect to geometric combinatorics. He presents a survey on C&P problems and classifies the studies as in Figure 1. According to this classification, cutting problems and packing problems are dual of each other. Due to this duality, there is a natural connection between these two streams of research.

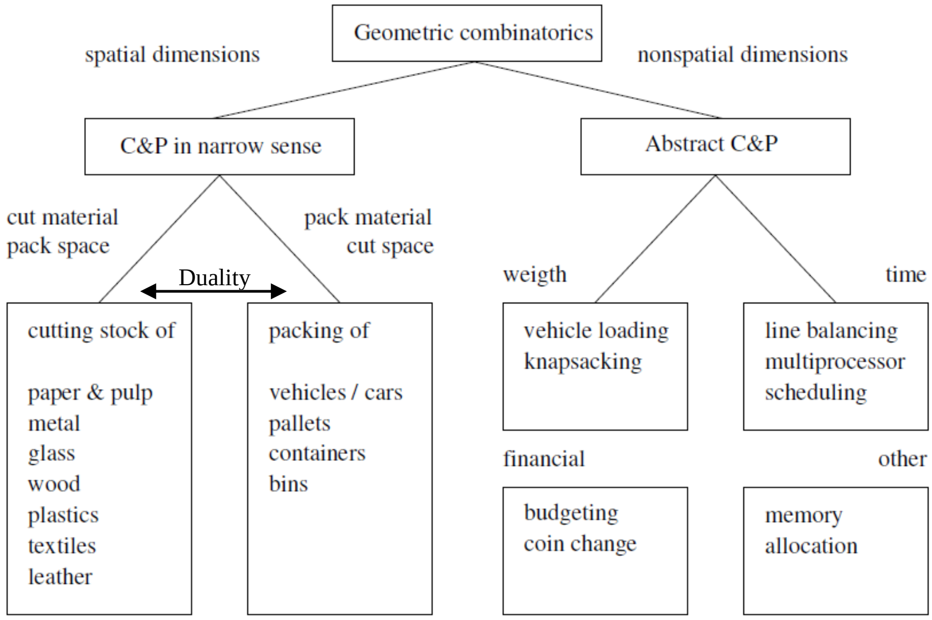


Figure 1 Phenomology of C&P problems, (Dyckhoff, 1990)

Dyckhoff (1990) groups the existing cutting and packing problems in the literature, under four main characteristics as follows:

1. *Dimensionality*
   (1) One-dimensional
   (2) Two-dimensional
   (3) Three-dimensional
   (N) N-dimensional with N > 3.
2. *Kind of Assignment*
   (B) All large objects and a selection of small items
   (V) A selection of large objects and all small items
3. *Assortment of large objects*
   (O) One large object
   (I) Many identical large objects
   (D) Different large objects
4. *Assortment of small items (shapes)*
   (F) Few items of (different figures)
   (M) Many items of many different figures
   (R) Many items of relatively few different (non-congruent) figures
   (C) Congruent figures

Later, Wäscher et al (2007) presents an improved typology. Although the main frame is based on four characteristics of Dyckhoff (1990), some modifications are made to Dyckhoff's classification to represent more details about the structure of the problem. They propose a list of abbreviations for the problem types which currently used for indexing related papers in database of ESICUP which is a special interest group on cutting and packing. These abbreviations are reported in Table 1.

Table 1 C&P problem abbreviations of Wäscher, G. et al (2007)

Abbreviation	**Problem Type**
BPP	Bin Packing Problem
IIPP	Identical Item Packing Problem
MBSBPP	Multiple Bin Size Bin Packing Problem
MHKP	Multiple Heterogeneous Knapsack Problem
MHLOPP	Multiple Heterogeneous Large Object Placement Problem
MIKP	Multiple Identical Knapsack Problem
MILOPP	Multiple Identical Large Object Placement Problem
MSSCSP	Multiple Stock Size Cutting Stock Problem
ODP	Open Dimension Problem
RBPP	Residual Bin Packing Problem
RCSP	Residual Cutting Stock Problem
SBSBPP	Single Bin Size Bin Packing Problem
SKP	Single Knapsack Problem
SLOPP	Single Large Object Placement Problem
SSSCSP	Single Stock Size Cutting Stock Problem

Note that these abbreviations are not sufficient to completely describe the problems since it does not include the information about small items geometry. For example, a two-dimensional rectangular object placement problem is represented as "2D rectangular SLOPP" by Wäscher et al (2007). For cutting stock problems, classification with respect to the small item geometries is illustrated in Figure 2.

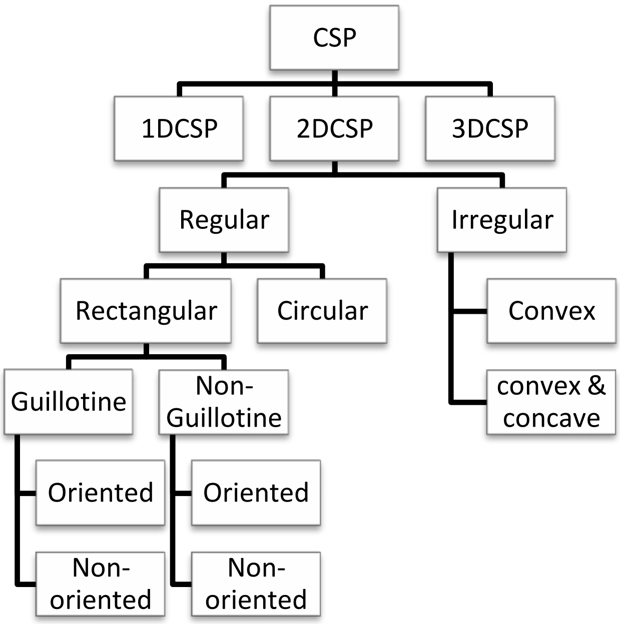


Figure 2 Classification of the problem with small item's geometry

Following the classification in Figure 2, the problem studied in this thesis is an irregular shape two-dimensional cutting stock problem (irregular 2DCSP). The term irregular refers to both convex and non-convex polygons. The stock material in this thesis is assumed to be in the form of a rectangular shape, having predetermined widths. In this situation, the cutting problem reduces to the length minimization of the stock, which is classified as a 2D irregular ODP according to the typology of Wäscher et al (2007). Therefore, the cutting patterns are generated without using the whole length of the stock material. In contrast to most of the studies in the literature, the rotation of pieces is completely allowed. Note that in textile industry, there are some pattern restrictions such as the importance of texture or fiber direction. Therefore, our method is more appropriate for plasma cutting and laser cutting of metal pieces in metal cutting industry, which is the main focus of this study.

2.2 Critical Decisions on Pattern Generation

There are some critical decisions to be made for irregular shape pattern generation, which directly affect the quality of the solution. These are placement order decision, rotation restriction and overlap prevention.

2.2.1 Placement Order

For most of the algorithms reviewed in the literature, a cutting pattern is generated by placing each piece on the stock one by one, following a placement order. This placement order is vital on the cutting performance. A common approach is to use a greedy algorithm operating on the order of the pieces. Although this reduces the size of the solution space, it also reduces the quality of the solution as expected. Therefore, it is important to select an appropriate ordering criterion to yield patterns with high performance. Oliveira et all (2000) propose a range of criteria for ordering of pieces. These are as follows:

- decreasing area
- decreasing length
- decreasing width
- decreasing irregularity
- increasing irregularity
- random order

Dowsland and Dowsland (1995) place items under a random ordering policy. Qu and Sanders (1987) study the performance of two different ordering policies; first one is to sort the pieces by descending order of length and the other is to use the height (width) instead of the lengths. Gomes and Oliveira (2002) study alternative criteria to generate the order of items to be placed. They later proposed *random weighted length*, which generates the sequences by selecting the pieces with a probability of selecting a piece is proportional to its length (Gomes and Oliveira, 2006).

Apart from studies which practice a random ordering policy, sorting the items in descending order of sizes is very common in the literature, since it becomes harder to find a room for larger pieces when they are left behind. Besides, considering area as a measure yields better results than using a single dimensional measure, e.g. length or width. Among all these ordering policies, none of them guarantees dominance in terms of solution quality.

2.2.2 Rotation Restriction and Orientation Decision

The 2D irregular shape cutting stock problem is more complex than the ones with regular shape. To be able to reduce the complexity, rotation of the pieces is generally not allowed (Ono and Watanabe 1997), or it is restricted by limited rotation angles and/or mirror-reflection poses of the pieces are used. Milenkovic (1999) studies the problem in two different directions, in which (i) any rotation angle is allowed, (ii) only rotation angles of 22.5^o and multipliers are allowed. He shows that the setting that allows any rotation angle yields quite better results in comparison with the setting that limits the rotation angle (Milenkovic, 1999).

Although there are alternative rotation constraints imposed, 90^o or 180^o angles are common in the literature. Gomes and Oliveira (2002) can be referred as an example for studies only allowing a rotation angle of 180^o.

The rotation restrictions can be reasonable in practice depending on the nature of the problem. For instance, in textile industry it is not possible to change the position of a shirt piece because of the fiber texture. However, for cases where rotation is practically possible for the problem, omitting or limiting the rotation angle may yield poor results. Furthermore, for the cases which rotation is not allowed, it is also critical to decide which orientation of the shape should be assumed as the initial pose. On the contrary, when any rotation of the piece is allowed, the resulting cutting pattern does not depend on the initial pose of the shape. Since rotation is allowed in any angle for our study, resulting cutting is independent of the initial orientations of the shapes.

2.2.3 Overlap Prevention

Since, it is not possible to have a feasible cutting pattern when two shapes are overlapped; overlap prevention becomes critical. Nevertheless, the overlap prevention is not an easy issue in case of irregular shapes. As explained in detail in the following subsections, one of the main approaches in the literature is No Fit Polygons (NFP). The other overlap prevention approaches are based on geometrical and trigonometric considerations such as combinations of line intersections and point in polygon tests or specialized procedures as polygon clipping.

No Fit Polygons

NFP is a common method used for preventing overlaps in irregular shape cutting and packing problems. Basically NFP searches for all possible arrangements of two polygons such that shapes are in touch but it is not possible to move them closer. The NFP method is firstly introduced by Adamowicz and Albano (1976) which is based on the “shape envelop” concept that is used in Art (1966). Later, Mahadevan

(1984) presents an algorithm to build NFPs. Cunninghame-Green (1989) proposes an approach to generate NFP_{AB} by tracing polygon ‘B’ around a fixed polygon ‘A’. Burke and Kendall (1999), Ribeiro et al (1999), Whitwell (2004) and many other authors prefer the tracing approach to generate NFPs. NFP of two shapes is illustrated in Figure 3.

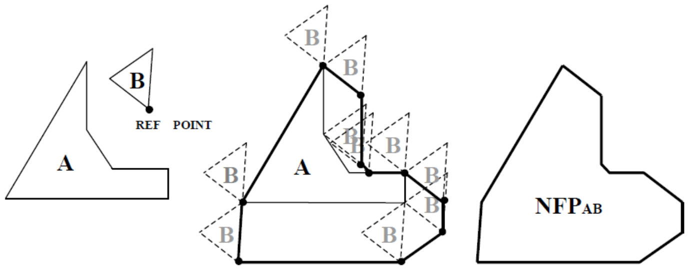


Figure 3 The no-fit polygon of two shapes A and B (Whitwell, 2004)

Whitweel (2004) reviews the methods to generate NFP and categorizes them. The extended study is later presented by Burke (2005). In general, NFP method necessitates fixing up the orientation of the two shapes before creating the NFP of two polygons. Therefore, the method is not suitable for a full rotation allowance and it is not possible to use in our edge matching approach since the orientations of the pieces are fixed.

Trigonometric Approach

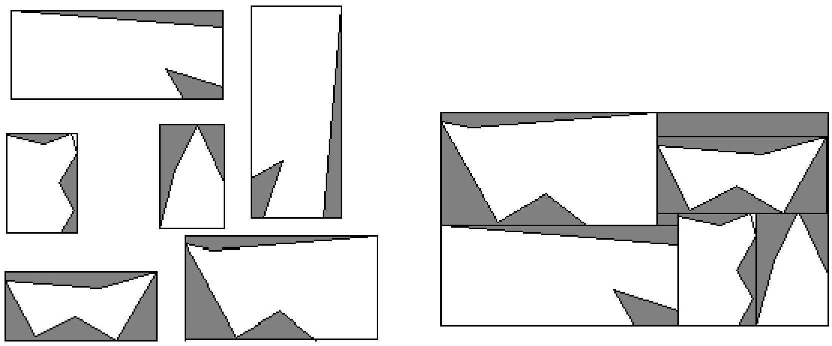
Trigonometric approaches can detect overlaps by considering the shapes as a combination of geometric elements, e.g., line segments. Dowsland et al. (2002) state that using trigonometric methods requires extensive computational effort. Whitweel (2004) discusses the pros and cons of trigonometric approaches with respect to the use of NFP. He states that trigonometric approaches are more robust with respect to the NFP, and that the implementation of NFPs is more complex. And he concludes that trigonometric approaches are preferred due to implementation ease in software applications although the use of NFPs is quicker than trigonometric calculations. In this thesis, trigonometric approach is being used to be able to detect the overlaps since it allows rotation in any angle.

2.3 Solution Procedures

The two-dimensional cutting stock problems are known to be NP-Hard (Garey and Johnson, 1979). Therefore, heuristic methods are prevalent in the literature. In this section, heuristic methods developed for irregular shape 2DCSP are introduced. And a recent study on mathematical modeling formulation of the problem is also presented.

2.3.1 Minimum Enclosing Boundary

The earliest solution methods for irregular shape 2DCSP are based on determining the smallest enclosing boundary rectangle for each irregular shape and then generating the final layout by solving a rectangular 2DCSP. Haims (1966) is the first researcher who tried to generate a cutting pattern for the irregular shapes using their minimum boundary rectangles. Later on, Haims and Freeman (1970) extends the method to have more compacted enclosing boundaries. Although the method is computationally efficient since it reduces the problem to a regular shape 2DCSP, it yields large amounts of scrape accumulated by each enclosing boundary generated. Nevertheless, there is a waste for almost all the pieces when the enclosing boundaries of the shapes are used, and the total scrap is accumulated in large amounts. Therefore, the quality of the solutions generated is generally not satisfactory in terms of raw material utilization. Figure 4-a, illustrates five pieces and their enclosing boundaries. Figure 4-b depicts a solution generated by using a minimum enclosing boundary method. Adamowicz and Albano (1976) proposed a treshold to be able have a control on the quality of the solutions. They were combining two same pieces such that one of them is 180^{o} rotated to have a more regular shape if the treshold is not satisfied with the single piece.



a) 5 pieces in their enclosing boundaries b) Layout for rectangular boundaries

Figure 4 Illustration of enclosing boundary

Apart from rectangular shapes, other geometries are also used for enclosing boundaries in the literature. Using a similar method, Dori and Ben-Bassat (1984) compute the minimum enclosing hexagons for convex shapes and then tessalete the entire stock material with these hexagons.

2.3.2 Bottom Left Approach

The bottom left (BL) approach which is also named as the left-most policy relies on placing a piece at the left most available position. This method is generally combined with other methods to be able to generate more efficient solutions. Jacobs (1996) uses minimum enclosing rectangles for each shape to decide their orientations and then places the shapes using a bottom left strategy. Gomes and Oliveira (2002) propose a method to generate sequences for placement order of the pieces using bottom left approach. Burke et al. (2005) presented an improved BL algorithm using hill climbing and tabu search techniques.

2.3.3 Compaction – Separation Algorithms

Compaction and separation algorithms can be implemented as just compaction, just separation or as a hybrid of both. In pure compaction algorithm, the pieces are first positioned on the stock sheet without any overlap. The resulting layout is a trivial solution. Then this solution is improved by shifting the pieces towards a given direction until they hit any other piece. At the end of the shifting process, a compact version of the first trivial solution is obtained. Figure 5 illustrates the compaction process.

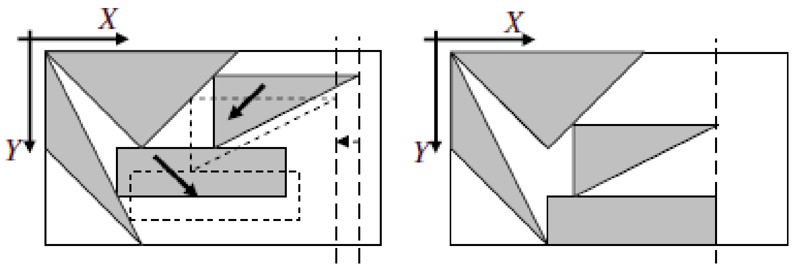


Figure 5 Compaction example (Gomes and Oliveira, 2006)

In separation, pieces are positioned on the stock sheets by allowing overlaps. The resulting layout will be an infeasible cutting pattern. Then, by shifting the pieces towards a given direction, the pieces are separated from each other until the overlaps are eliminated. The process should continue until a feasible layout with no overlaps is obtained. Figure 6 illustrates the separation of pieces.

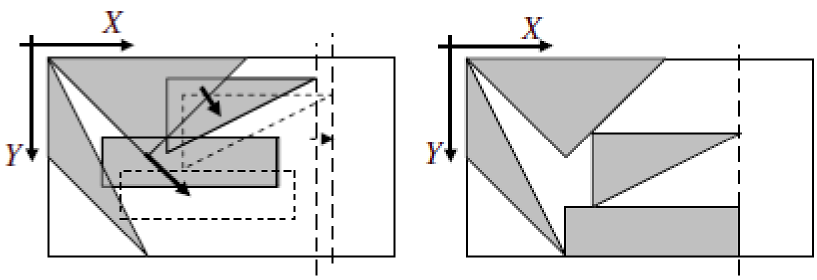


Figure 6 Separation example (Gomes and Oliveira, 2006)

To compact the cutting layout by shifting the pieces is not an easy task. Li and Milenkovic (1993) study the complexity of compaction problem and stated that compaction problem is NP-Hard. Gomes and Oliveira (2006) presents an LP model for the compaction problem. This model is a generalization of the previously studied models in the literature (Li and Milenkovic, 1995, Stoyan et al., 1996, Bennell and Dowsland, 2001). It minimizes the length of the stock material with respect to the given constraints. The reader may refer to Gomes and Oliveira (2006) for detailed information about the model.

After implementing separation algorithm, quality of the solution can be further improved if the separated layout is compacted again. This hybrid algorithm is an iterative and consecutive implementation of compaction and separation algorithms (Gomes and Oliveira, 2006).

2.3.4 Metaheuristics

Due to the discrete combinatorial structure of irregular shaped 2DCSP, it is a common approach in the literature to apply metaheuristic to obtain high quality solutions for large problem instances within acceptable time limits. Metaheuristics provide an efficient procedure for searching solution space for the problem. This section only includes some examples on metaheuristic implementations for the problem.

One of the most implemented metaheuristics on this problem is genetic algorithms (GA). İsmail and Hon (1995) studies genetic algorithms for 2D nesting. Bounsaythip and Maouche (1996) use a genetic approach to solve a cutting stock problem, in textile industry. Ono and Watanabe (1997) apply genetic algorithms to obtain an efficient cutting pattern for a 2DCSP in which all shapes are polygon, either convex or concave. Petridis et al. (1998) uses varying fitness functions in the genetic optimization.

There are also tabu search (Benell and Downsland, 1999 – 2001; Blazewicz et al., 2004; Burke et al., 2005) and simulated annealing implementations (Theodoracates and Grimsley, 1995; Heckman and Lengauer, 1995; Burke and Kendall, 1999; Gomes and Oliveira, 2006) in this problem enviroment.

A common characteristic of studied metaheuristics is that higher computation times are required to achieve high quality solutions in terms of raw material utilization.

2.3.5 Mathematical Programming

In the literature, heuristic methods have predominance over exact solution procedures in solving irregular C&P problems. But still, there is limited number of studies to reach optimal solutions, based on exact formulation of the problems. As an example for the case, Gomes and Oliveira (2006) use a linear programming model formulation in the compaction phase of their study while the main search space is traced by simulated annealing algorithm. In a recent study, Chernov et. al. (2010) present a mathematical model for packing of irregular objects. Since our study also proposes a mathematical model formulation for the problem, the study of Chernov et al. is discussed in detail. The main idea of their model is representing the irregular pieces as the combinations of primitive geometric shapes such as circles, rectangles and regular polygons. They call these combined shapes as phi-objects. In Figure 7, there is a phi object represented as the unions and subtractions of basic geometric shapes.

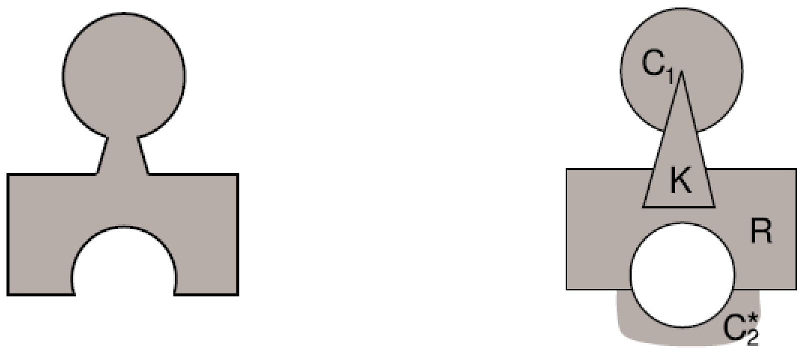


Figure 7 An example of a composed phi-object $\boldsymbol{C_1} \cup \boldsymbol{K} \cup (\boldsymbol{R} \cap \boldsymbol{C_2^*})$ (Chernov et al., 2010)

Geometric parameters of the phi-objects are specified with respect to the analytical equations of those geometric shapes. For example, a circle is defined with its center, C, and radius, r, using the analytic equation $(C, r) = \{(x, y): x^2 + y^2 \leq r^2\}$.

In order to adjust the position and orientation of a piece, a translation vector is introduced as $v = (v_1, v_2)$, and a rotation angle is defined as $\theta \epsilon [0, 2\pi)$. Trigonometric relations are used to redefine the coordinates after changes in position.

To prevent the overlaps, phi-functions are formulated for couples of primitive shapes (e.g., circle-circle, rectangle-rectangle and circle-rectangle). Phi-functions calculate the distance between these couples and they are denoted by $\phi^{T_i T_j}$. In the model, lower limits on the distances that prevent the overlaps are used as constraints on the phi-functions. In general, the value of a phi-function is positive if there is no intersection, zero if pieces are touching, and negative if they are overlapping. It is also possible to introduce an offset distance that can be used as lower limit instead of zero. In this way, imposing an offset constraint prevents the pieces to get closer than a desired distance.

In terms of phi-functions and phi-objects, the authors formulate the model as follows,

Min F(u_o)

Subject to:

$$\phi^{T_0 T_i} \geq 0 \qquad \forall i$$

$$\phi^{T_i T_j} \geq 0 \qquad \forall i \neq j$$

Where item i=0 represents the stock material. Since phi-functions are generally in quadratic form the resulted model is non-linear. Therefore, it is required to provide a feasible initial layout for the problem to obtain a good starting solution. In this model, there is a general representation for the constraint that prevents overlaps. However, each phi-function is unique, hence, it is required to construct phi-functions of each shape pair before running the model.

By the use of proper phi-functions in mathematical formulation, this model is capable of generating solutions for a wide range of objects including non-convex and/or curved shapes, with continuous translations and rotations for each object.

In our mathematical model formulation which is presented in Section 3.2, phi-functions are not used. Instead, vertex coordinates are used to introduce the shapes so the constraint equations are standard for all shapes.

CHAPTER 3

PROBLEM DEFINITION AND MATHEMATICAL MODEL

In this section we propose an exact mathematical model formulation for the irregular shaped 2DCSP, and a solution procedure to solve this problem. First, the problem is defined in Section 3.1. Then, in Section 3.2, a mathematical model formulation is presented for the defined problem.

3.1 Problem Definition

There are N irregular shape polygons each having different number of vertices. The problem is to position the pieces on a rectangular stock sheet with a given width such that there is no overlap and the stock length used is being minimized. The quality of the layout generated which is called cutting pattern is measured by a ratio of total area of the shapes to the area of stock sheet used as shown in Equation 1. The ratio gives the quality of the solution, that is to say efficiency of the cutting pattern. Since the stock width, (*W*), is constant for the problem, minimizing the length of the stock, (*L*), is equivalent to maximizing the efficiency, (E), for stock material utilization. A sample cutting pattern is illustrated in Figure 8.

$$Max\ E = \frac{\sum_i^n A_i}{WL} \equiv Min\ L \qquad (1)$$

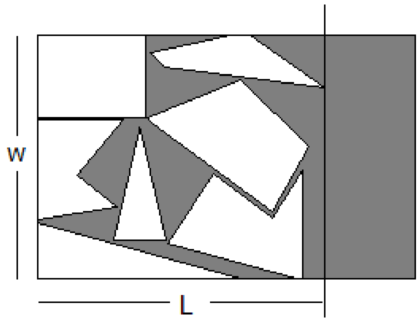


Figure 8 Sample cutting pattern

3.2 The Mathematical Model Formulation

In this subsection, we propose a mixed integer non-linear programming model for the irregular 2DCSP. The model can be described as minimizing the length of the stock material ensuring that all pieces are placed on the stock sheet without any overlaps. The constraints guarantees positioning of every pieces on the stock while at the same time preventing the overlaps.

3.2.1 Positioning Constraint

The shapes of the pieces (not necessarily convex) are stated by Cartesian coordinates of the vertices in two-dimensional space. Since these coordinates also indicate the position of the pieces as well as the shapes, it can be said that there exist initial positions for each pieces. To change the position and orientation of a piece, two main actions are required, shifting the piece along x and y axis, and rotating the piece. Figure 9 illustrates these actions. In the figure, point 'O' is the reference point for the piece which is used as rotation base. For the sake of simplicity, the point 'O' is taken as origin point (0,0). First, the piece is rotated around the point 'O' by α. Then, by shifting the piece as Cx and Cy along the x and y axis respectively, the base point O is moved to the point (Cx,Cy).

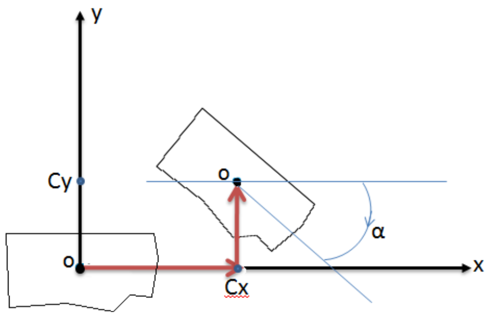


Figure 9 Defining the placement of a piece

Our formulation requires inputting initial (x,y) coordinates for each piece. Let x_i and y_i denote the coordinates for a vertex of piece i. In the case that the origin is taken as rotation reference point of the piece, trigonometrically, the final coordinates can be computed as,

$$\cos\left[\tan^{-1}\frac{y_i}{x_i}+\alpha\right]\sqrt{(x_i{}^2+y_i{}^2)}+Cx=nX_i \qquad \text{for all } i$$

$$-\sin\left[\tan^{-1}\frac{y_i}{x_i}+\alpha\right]\sqrt{(x_i{}^2+y_i{}^2)}+Cy=nY_i \qquad \text{for all } i$$

where

Cx: Replacement of the reference point of the piece along x-axis

Cy: Replacement of the reference point of the piece along y-axis

α: Rotation angle for piece (around reference point which is origin)

nX_i: Final x coordinate of vertex i for piece

nY_i : Final y coordinate of vertex i for piece

For a rectangular stock material, assuming that lower left corner is the origin, following decision variables are introduced to define the maximum x and y coordinates.

X_{max}: Maximum of x coordinates of piece

Y_{max}: Maximum of y coordinates of piece

MaxX: Length of stock material

W: Width of stock material

Then the following inequalities are to ensure that any piece is positioned within the stock material boundaries.

$MaxX \geq X_{max} \geq nX_{\mathrm{i}}$ for all i

$W \geq Y_{max} \geq nY_{\mathrm{i}}$ for all i

3.2.2 Overlap Prevention Constraint

The overlaping of two pieces (equivalently shapes) can occur in three different ways as it is shown in Figure 10.

(a) A vertex of a shape falls on the other shape (also called point in polygon), but no line intersection occurs.

(b) The edges intersect, but no point in polygon situation occurs.

(c) Both line intersections and point in polygon situation occur.

For each of these three cases it can be said that the two shapes are overlapping.

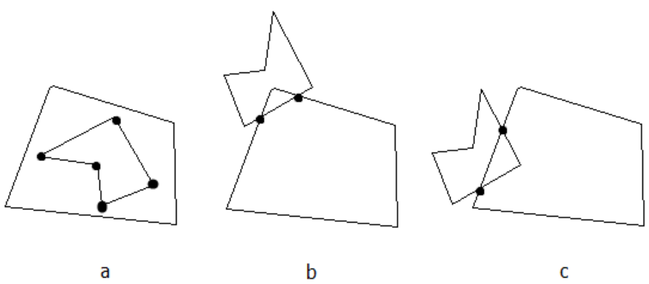


Figure 10 Overlap position

To guarantee that there is no overlap between two shapes, (i) each edge pairs must be checked whether there exists a line intersection and (ii) the vertices must be controlled for a point in polygon situation as illustrated in Figure 10-a.

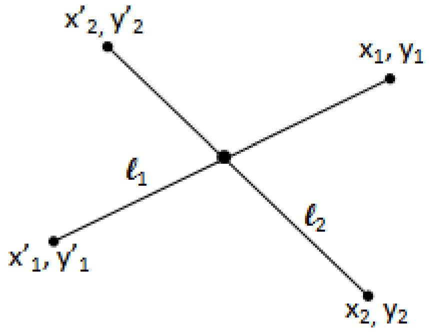


Figure 11 Intersection of two line segments

To determine whether two shapes overlap, one should check the following three possible cases for any two line segments: (i) they can be parallel, (ii) they can intersect, or (iii) they neither intersect nor are parallel. Let l_1 and l_2 are two line segments as it is illustrated in Figure 11 and let (x_1, y_1) and (x'_1, y'_1) be end points of l_1. Similarly, let (x_2, y_2) and (x'_2, y'_2) are end points of l_2. To check whether these two line segments intersect, first, one should check whether they are parallel or not by using the following equation.

$$d = (x_2 - x'_2)\,(y_1 - y'_1) - (x_1 - x'_1)\,(y_2 - y'_2) \qquad (2)$$

If d = 0, this implies l_1 and l_2 are parallel, hence there is no intersection. However, if d ≠ 0; then two different p parameters must be calculated using;

$$p_1 = \frac{(x_k - x_l)\,(y_i - y_k) - (x_i - x_k)\,(y_k - y_l)}{d} \qquad (3)$$

$$p_2 = \frac{(x_i - x_k)(y_j - y_i) - (x_j - x_i)(y_i - y_k)}{d} \qquad (4)$$

If $0 \leq p_1 \leq 1$ and $0 \leq p_2 \leq 1$; then l_1 and l_2 intersects on a point along the segments. If $p = 0$ or $p = 1$; then this implies that the intersection point is on the end point of the line segments. Since the intersection on the vertices do not mean overlap for the shapes, it can be said that when $p \geq 1$ or $p \leq 0$ there is no edge intersection which causes an overlap (Bildirici, 2003).

Checking possible edge intersection can detect overlaps for the cases shown in Figure 10-b and 10-c. Nevertheless, it is not sufficient to conclude that there is no overlap for shapes. To guarantee preventing overlaps, the vertices of the shape must be considered for point in polygon situation that is illustrated in Figure 10-a. Note that, if a vertex of any polygon lays in any another polygon without line intersections between the two polygons, all other vertices of the former polygon also lays in the latter polygon. Thus, if it is guaranteed that there are no line intersections between two polygons, one should still check whether one polygon is inside the other one. However, it is sufficient just to check a single vertex from each polygon for point in polygon situation, instead of checking all vertices one by one.

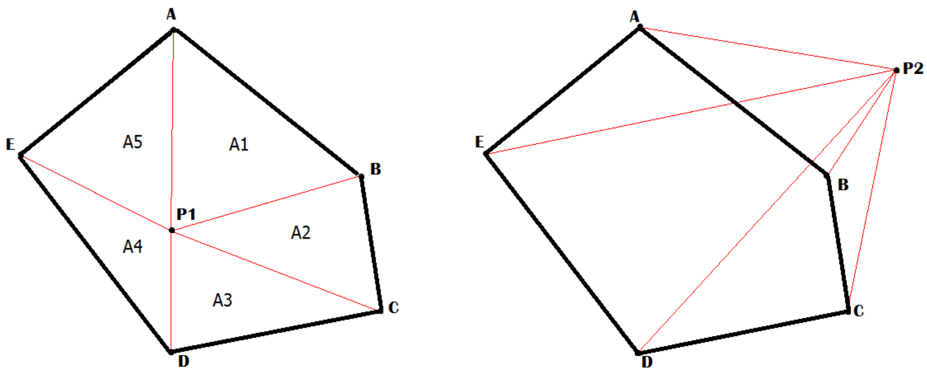


Figure 12 Point in the polygon

In order to check whether a point is inside a polygon or not, we apply the following method. Figure 12 illustrates same convex polygon twice. In the first figure, there is a point P1, inside the polygon ABCDE; and in the second one there is a point P2 outside the polygon ABCDE. For both shapes, the points are connected to the vertices of the polygon with lines. Then, for each of the figure the following triangles are obtained: ABP1, BCP1, CDP1, DEP1, and EAP1 from the first one and ABP2, BCP2, CDP2, DEP2, and EAP2 from the second one. Note that, for a convex polygon, the summation of the areas of these triangles must be equal to the area of the polygon if point P is inside that polygon. Otherwise, the total area of the triangles must exceed the area of the polygon. For P(x_1, x_2), A(x_2, y_2), B(x_3, y_3) the area of triangle ABP can be computed as follows;

$$\text{A1} = \text{A(ABP)} = \left|\frac{1}{2}(x_1\, y_2 + x_2\, y_3 + x_3\, y_1) - (x_2\, y_1 + x_3\, y_2 + x_1\, y_3)\right| \qquad (5)$$

This area control may not work for non convex polygons. Therefore, the method that we propose is valid for non-convex shapes or the concave shapes which cannot be positioned as in the case of Figure 10-a.

3.2.3 Mixed Integer Non linear Model Formulation

Using the findings in previous, a non linear mathematical formulation is presented in this section. For N pieces, each having V_i vertices where $i \in N$, our decision variables and parameters are as follows:

Decision variables:

α_i: Rotation angle for piece i (around origin.) $\forall i \in N$

Cx_i: Shift among x-axis for piece i $\forall i \in N$

Cy_i: Shift among y-axis for piece i $\quad \forall i \in N$

$Xmax_i$: Maximum of x coordinates of piece i $\quad \forall i \in N$

$Ymax_i$: Maximum of y coordinates of piece i $\quad \forall i \in N$

$MaxX$: global maximum of x coordinates

nX_{ij}: final x coordinate of vertex j for piece i $\quad \forall i \in N,\ \forall j \in V_i$

nY_{ij}: final y coordinate of vertex j for piece i $\quad \forall i \in N,\ \forall j \in V_i$

$P1_{ijkm}$: intersection control parameter btw jth edge of piece i and mth edge of piece k

$\forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

$P2_{ijkm}$: intersection control parameter btw jth edge of piece i and mth edge of piece k

$\forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

$Z1_{ijkm}$: 0 or 1 $\quad \forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

$Z2_{ijkm}$: 0 or 1 $\quad \forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

$Z3_{ijkm}$: 0 or 1 $\quad \forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

$Z3_{ijkm}$: 0 or 1 $\quad \forall i,j,k,m \quad \forall i,k \in N;\ \forall j \in V_i;\ \forall j \in V_j;\ i < k$

Note that, Z variables are also used to check intersections in coordination with P values.

Parameters:

x_{ij} : initial x coordinate for jth vertex of piece i $\quad \forall i,j \quad i = 1,\dots,N \quad j = 1,\dots,V_i + 1$

y_{ij} : initial y coordinate for jth vertex of piece i $\quad \forall i,j \quad i = 1,\dots,N \quad j = 1,\dots,V_i + 1$

A_i : Area of shape i $\quad \forall i,\ i = 1,\dots,N$

W: Stock width

M: A large number

Given those decision variables and parameters, the mixed integer non-linear model can be formulated as followings.

Minimize Z = *MaxX* (6)

subject to:

$$\cos\left[\tan^{-1}\frac{y_{ij}}{x_{ij}}+\alpha_i\right]\sqrt{\left(x_{ij}{}^2+y_{ij}{}^2\right)}+Cx_i=nX_{ij} \qquad \forall\, i,j \tag{7}$$

$$-\sin\left[\tan^{-1}\frac{y_{ij}}{x_{ij}}+\alpha_i\right]\sqrt{\left(x_{ij}{}^2+y_{ij}{}^2\right)}+Cy_i=nY_{ij} \qquad \forall\, i,j \tag{8}$$

$$\frac{(nX_{km}-nX_{k,m+1})(nY_{ij}-nY_{km})-(nX_{ij}-nX_{km})(nY_{km}-nY_{k,m+1})}{(nX_{i,j+1}-nX_{ij})(nY_{km}-nY_{k,m+1})-(nX_{km}-nX_{k,m+1})(nY_{i,j+1}-nY_{ij})}=P1_{ijkm} \tag{9}$$

$$\frac{(nX_{ij}-nX_{km})(nY_{i,j+1}-nY_{ij})-(nX_{i,j+1}-nX_{ij})(nY_{ij}-nY_{km})}{(nX_{i,j+1}-nX_{ij})(nY_{km}-nY_{k,m+1})-(nX_{km}-nX_{k,m+1})(nY_{i,j+1}-nY_{ij})}=P2_{ijkm} \tag{10}$$

$$P1_{ijkm}\le M\left(1-Z1_{ijkm}\right) \tag{11}$$

$$P1_{ijkm}\ge 1-M\left(1-Z2_{ijkm}\right) \tag{12}$$

$$P2_{ijkm}\le M\left(1-Z3_{ijkm}\right) \tag{13}$$

$$P2_{ijkm}\ge 1-M\left(1-Z4_{ijkm}\right) \tag{14}$$

$$Z1_{ijkm}+Z2_{ijkm}+Z3_{ijkm}+Z4_{ijkm}\ge 1 \tag{15}$$

$$Xmax_i \ge nX_{ij} \qquad \forall\, i,j \tag{16}$$

$$Ymax_i \ge nY_{ij} \qquad \forall\, i,j \tag{17}$$

$$MaxX \ge Xmax_i \qquad \forall\, i \tag{18}$$

$$W \ge Ymax_i \qquad \forall\, i \tag{19}$$

$$\sum_{j=1}^{n-1}\left|\frac{1}{2}\left(\mathrm{n}X_{ij}\,nY_{i,j+1}+nX_{i,j+1}\,nY_{km}+nX_{km}\,nY_{ij}\right)-\left(nX_{i,j+1}\,nY_{ij}+nX_{km}\,nY_{i,j+1}+nX_{ij}\,nY_{km}\right)\right|$$

$\ge A_i$ (20)

For all the equations above (9) to (20);

$$\forall i,j,k,m \qquad i,k=1,\ldots,N;\quad j=1,\ldots,V_i;\ \ m=1,\ldots,V_j;\ \ i<k$$

In the model, Equation (6) stands for the objective function where *MaxX* denotes the used length of the stock material. Value of MaxX is computed in constraint set (18). Equations (7) and (8) are to compute the new coordinates after shift and rotation Equations (9) and (10) compute the *p* parameters which are discussed in Section 3.2.2. Constraints from (11) to (15) are to prevent edge intersections ensuring that there is no line intersection. Constraints (16) and (17) is to find maximum of x and y coordinates respectively for the shape i. Inequality (19) ensures that the shapes do not exceed the width of the shape. Final constraint (20) checks point in polygon condition using Equation (5) and ensure that a vertex of a shape cannot be positioned within the boundary of another shape.

3.3 Model Verification and Results

Our mathematical model is tested for problem sizes up to 7 pieces. For the test data, the pieces are selected such that it is not possible to cover a piece with another one as in Figure 10-a. This implies that the 20^{th} constraint in the formulation can be omitted. For the data set with two pieces, the pieces shown in Figure 13 are used.

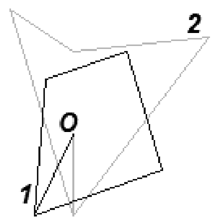


Figure 13 Two piece data set

The initial coordinates of the pieces are given in Tables 2 and 3. The initial values for decision variables Cx, Cy and α angle are all taken as zero for the initial positions. The model is written on an Excel sheet and it is solved by using Excel Solver. The screenshot of the related sheet can be seen in Figure 13.

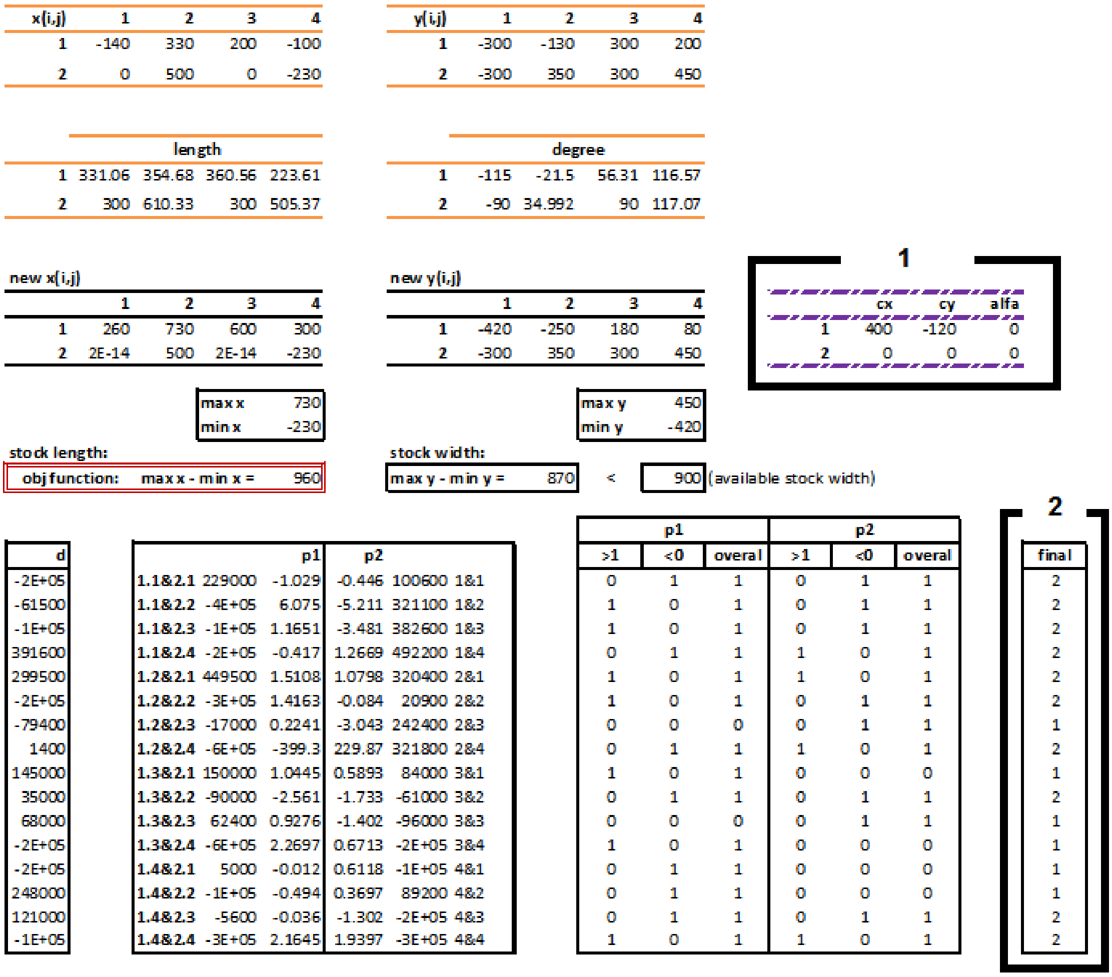
Table 2 Initial x coordinates for the pieces

x(i,j)	1	2	3	4
1	-140	330	200	-100
2	0	500	0	-230

Table 3 Initial y coordinates for the pieces

y(i,j)	1	2	3	4
1	-300	-130	300	200
2	-300	350	300	450

As it is illustrated in Figure 14, initial coordinates of the pieces are inserted in the sheet. The positioning variables C_x, C_y and α values are also given as a part of the initial feasible solution.



x(i,j)	1	2	3	4
1	-140	330	200	-100
2	0	500	0	-230

y(i,j)	1	2	3	4
1	-300	-130	300	200
2	-300	350	300	450

	length			
1	331.06	354.68	360.56	223.61
2	300	610.33	300	505.37

	degree			
1	-115	-21.5	56.31	116.57
2	-90	34.992	90	117.07

new x(i,j)

	1	2	3	4
1	260	730	600	300
2	2E-14	500	2E-14	-230

new y(i,j)

	1	2	3	4
1	-420	-250	180	80
2	-300	350	300	450

1

	cx	cy	alfa
1	400	-120	0
2	0	0	0

max x	730
min x	-230

max y	450
min y	-420

stock length:

obj function: max x - min x = 960

stock width:

max y - min y = 870 < 900 (available stock width)

d
-2E+05
-61500
-1E+05
391600
299500
-2E+05
-79400
1400
145000
35000
68000
-2E+05
-2E+05
248000
121000
-1E+05

		p1	p2		
1.1&2.1	229000	-1.029	-0.446	100600	1&1
1.1&2.2	-4E+05	6.075	-5.211	321100	1&2
1.1&2.3	-1E+05	1.1651	-3.481	382600	1&3
1.1&2.4	-2E+05	-0.417	1.2669	492200	1&4
1.2&2.1	449500	1.5108	1.0798	320400	2&1
1.2&2.2	-3E+05	1.4163	-0.084	20900	2&2
1.2&2.3	-17000	0.2241	-3.043	242400	2&3
1.2&2.4	-6E+05	-399.3	229.87	321800	2&4
1.3&2.1	150000	1.0445	0.5893	84000	3&1
1.3&2.2	-90000	-2.561	-1.733	-61000	3&2
1.3&2.3	62400	0.9276	-1.402	-96000	3&3
1.3&2.4	-6E+05	2.2697	0.6713	-2E+05	3&4
1.4&2.1	5000	-0.012	0.6118	-1E+05	4&1
1.4&2.2	-1E+05	-0.494	0.3697	89200	4&2
1.4&2.3	-5600	-0.036	-1.302	-2E+05	4&3
1.4&2.4	-3E+05	2.1645	1.9397	-3E+05	4&4

p1			p2		
>1	<0	overal	>1	<0	overal
0	1	1	0	1	1
1	0	1	0	1	1
1	0	1	0	1	1
0	1	1	1	0	1
1	0	1	1	0	1
1	0	1	0	1	1
0	0	0	0	1	1
0	1	1	1	0	1
1	0	1	0	0	0
0	1	1	0	1	1
0	0	0	0	1	1
1	0	1	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1

2

final
2
2
2
2
2
2
1
2
1
2
1
1
1
1
2
2

Figure 14 Excel sheet for the two piece instance

By the use of those initial coordinates and positioning variables, lengths and degrees of the angles for each edge are computed and finally new x and y coordinates are derived using the Equations 7 and 8. The differences between maximum and minimum of x and y coordinates are also computed to see stock lengths and widths respectively. In the given example, the stock width is limited by 900 units, while stock length is minimized as an objective. Within the sheet, the first group illustrated as 'P1' and 'P2' are computed for each edge pairs using Equation 9 and 10. Second 'P1' and 'P2' group that include binary values correspond to the equations through Equation 11 to Equation 14. The column 'Final' is obtained by solving Equation 15 in our model. This column is the one to be introduced as '*constraints'* for the Excel Solver and set to be greater than or equal to 1. The resulting solution is feasible if the values in this column are greater than zero.

To run the model, the cells that include Cx, Cy and α values are selected as decision variables to Excel Solver. All other values e.g new coordinates, P values and the column *'final'* are computed, depending on these decision variables.

When the model is run, it is observed that the model cannot converge to a solution if all positioning variables are taken as zero. This is due to the non-linearity of the model. Therefore our model requires an initial feasible solution to converge to the optimal solution. The solution that the model yields also depends on the initial solution since Excel Solver searches for a local optimum solution.

To show the sensitivity of our method to the initial feasible solution, our model is tested using three different initial solutions as depicted in Table 4. Given the initial positions of the pieces, the respective layouts are obtained as shown in Figure 15.

Among three different initial solutions, minimum stock length has been recorded as 936.418 units using the initial solution shown in Figure 15-c.

Table 4 The resulting solutions for the two piece instance

		cx	cy	alfa	cx	cy	alfa	Stock Length	improvement
a	**Initial**	0.0000	0.0000	0.0000	500.0000	0.0000	0.0000	1140.000	~ %9.8
	Final	0.0005	0.0000	0.0024	387.3111	0.0000	0.0028	1027.287	
b	**Initial**	500.0000	-120.0000	0.0000	0.0000	0.0000	0.0000	1060.000	~ % 9.8
	Final	395.7700	-120.0000	-9×10^{-4}	0.0004	0.0000	-0.0030	955.745	
c	**Initial**	500.0000	-10.0000	-30.0000	0.0000	0.0000	0.0000	1053.205	~ % 11
	Final	386.5000	-10.0000	-28.8600	0.0005	0.0000	-0.0040	936.418	

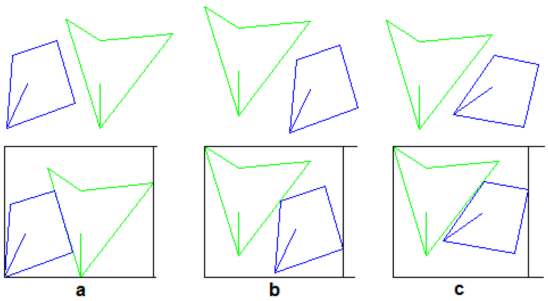


Figure 15 Results for three different initial solutions, respectively

Based on the observations, it is seen that the model is capable of compacting the layout under the given constraints. However, it should be noted that, Excel Solver stops searching for optimal solution when two pieces touches each other. Thus, results are strongly dependent to the initial solution and might be further away from the global optimum. However, this situation prevents to have the position in Figure 10-a for the pieces, so the model can compact the layout for both convex and non-convex polygons without using the 20th constraint. Moreover, it is not required to write edge intersection constraint for each edge pairs and the model is relaxed. For example, for the given initial of two pieces which are illustrated in Figure 16, it is enough to use edge intersection constraints for the edge pairs [c,1], [b,1], and [d,1].

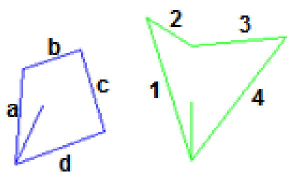


Figure 16 Example for initial positions of two pieces

To verify the model, 5 more instances are considered where the number of pieces varies from 3 to 7 pieces. Data sets are generated by adding a new piece to previous data set. Properties of these pieces are presented in Appendix A. Figure 17 illustrates the given initial cutting patterns and final solutions. Values of the decision variables are also reported in Appendix B.

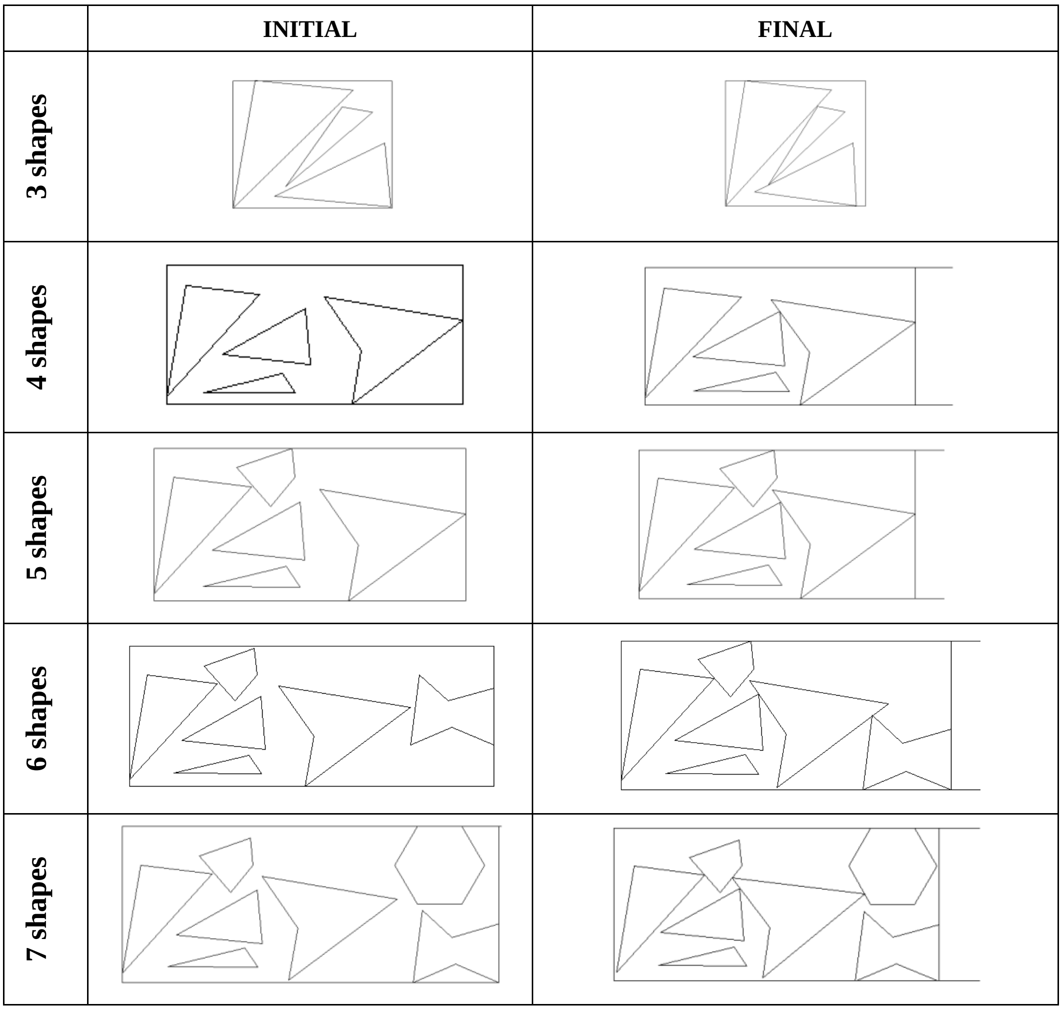


Figure 17 Resulting cutting patterns of the mathematical model

According to the results, it is seen that the solver compacts the pieces towards to center. Thus, the pieces which are positioned around the center of the pattern do not move. In general, position changes are caused by the linear motions through x and y axis and it stops when the moving piece hit to another piece. Rotation angle slightly changes to remove infeasibilities. Since the source of compaction is mainly caused by the outer positioned pieces, it is not expected to obtain good solutions for large scale problems. As it is illustrated in Table 5, for the tested data sets, the lengths of stock materials are shorten 12% on the average

Table 5 Length of initial and final cutting patterns

INSTANCES	INITIAL	FINAL	% Imp.
3 shapes	130	117.39	0.10
4 shapes	320	280.98	0.12
5 shapes	325	284.33	0.13
6 shapes	420	362.24	0.14
7 shapes	420	368.85	0.12
Average	**323**	**282.76**	**0.12**

CHAPTER 4

PROPOSED SOLUTION METHODOLOGY

The model presented in Chapter 3 becomes intractable in generated solutions for large scale problems with acceptable time limits. Thus, we propose a heuristic method based on an edge matching approach for the problem, and we test it compare to the benchmark results in the literature. In order to provide a comparison tool for new introduced data sets, we also propose a method to generate lower bounds. In this chapter, study o n lower bounds is presented in Section 4.1, and our heuristic method is presented in Section 4.2.

4.1 Lower Bounds

As it is mentioned above, our mathematical model formulation is describing the problem but it is not possible to employ the methodology directly for generating solutions for large scale problems in acceptable time limits. Therefore, it is not possible to compare the performance of a heuristic method with the optimal solution for real life problems. Thus, the only way to measure the quality of a heuristic solution seems to check the efficiency and to compare the result with previously reported solution methodologies in terms of the related data sets reported in the literature. In this section, lower bounds for irregular shaped 2DCSP are studied to be able to have an idea about the quality of the solution, which is measured in terms of the gap between the probable best solution and our solution.

4.1.1 Generation

One trivial lower bound for the stock length can be obtained by dividing the total area of the shapes to the stock width. This bound is equal to optimal solution only when it is possible to locate the pieces on stock material with zero scrap. But since this is not generally possible in practice, the gap between actual solution and this LB is expected to be high.

Therefore, to obtain a tighter bound, an alternative lower bounding scheme is considered. According to this scheme, the irregular shapes are represented by triangular and rectangular sub-pieces and then these sub-pieces are fed to the optimal solver to generate a cutting pattern that provides the lower bound. The method consists of the following steps:

Step 0: List the vertices of the shape.

Step 1: Select the next most upper vertex and draw a horizontal line.

Step 2: For that line, determine the most left touching and most right touching points to the shape.

Step 3: If the whole line segment within these two points are inside the shape go to Step 4. Else, go back to Step 1.

Step 4: Draw two vertical lines to the bottom, starting from the most left touching and most right touching points.

Step 5: Check the points where you hit by the vertical lines and select the one which has higher y coordinate.

Step 6: Starting from selected coordinate, draw a horizontal line towards the other vertical line. This will create a rectangle within two vertical and two horizontal lines.

Step 7: Compute the area of the rectangle and record. Then go back to Step 1 if there are still vertices unprocessed in the list. Go to step 8 if the bottom vertices are already processed in Step 1.

Step 8: Take the rectangle with the largest area. And consider the remaining parts as distinct irregular shapes and repeat the process until the shape is completely divided into rectangles and triangles.

The obtained largest rectangle and remaining irregular shapes are illustrated in Figure 18. According to the figure, P1 is the point on the 4^{th} visit to Step 1. As mentioned in Step 2, P2 is the most left coordinate and P3 is the most right coordinate for the drawn horizontal line on point P1. From the sketched vertical lines P5 is the one which is selected by Step 5. The line segment between P4 and P5 is sketched in Step 6.

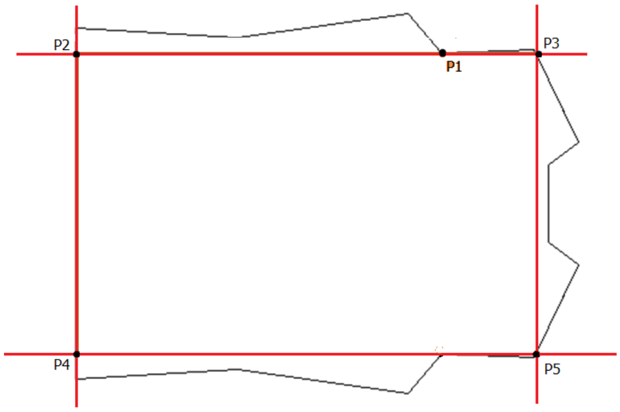


Figure 18 Making rectangles

After applying the steps of the lower bound algorithm for the remaining parts until the entire shape is represented in terms of rectangles and triangles, the shape in Figure 19 is obtained.

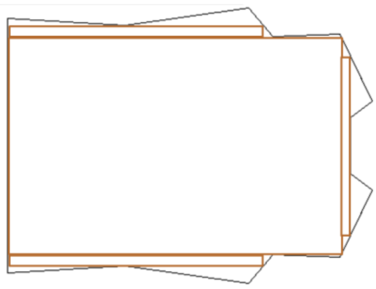


Figure 19 The shape divided in rectangles and triangles

The main idea of obtaining these rectangular sub-pieces is to use them as data set of a 2DCSP. The stock length of obtained cutting pattern yields a lower bound for irregular shaped problem for two main reasons:

1. The triangular parts are trimmed; hence, the area of each shape is reduced.
2. Splitting the shape into pieces breaks a natural constraint which enforces these rectangles to be placed in the same position relative to each other. So it becomes a relaxation for the problem.

The effect of the first item can be reduced by adding the total area of the triangle shapes to the solution obtained by 2DCSP. If we convert this additional area to a rigid rectangle which has the same width with the stock material and include it to the problem as a new piece, the lower bound will be improved.

4.1.2 Verification

To verify the method, benchmark datasets are used. For each data set, the shapes are divided into rectangles as stated above. Solving the cutting stock problems with these rectangular shapes, the lower bounds reported on Table 6 are obtained. Since the irregular shapes are split into pieces, the size of the problem increases considerably in terms of number of pieces. For data sets *shapes* and *shirts*, a feasible result is not reached within 3 hr run. The column named '*unrepresented*' shows the percentage of the area which belongs to the remaining triangles when the rectangles are trimmed. In the table, GAP stands for the relative gap between the best reported length for the given data sets, and the lower bounds that is obtained by rectangular 2DCSP. As the undefined area gets smaller, the represented area of the shape will increase. Thus, as the undefined area becomes larger, GAP also becomes larger.

Table 6 Lower bound test

	Number of pieces					Resulted % GAP
	Original		Rectangular form			
Data sets	**Type**	**Total**	**Type**	**Total**	**Unrepresented**	
Jacobs1	25	25	18	39	0.02%	18.30%
Mao	9	20	20	40	16.59%	11.28%
Marques	8	24	12	40	9.72%	6.47%
Albano	8	24	15	46	11.85%	2.12%
Shapes 0-1	4	43	8	133	2.00%	-
Blaz	7	28	13	52	24.21%	18.23%
Dagli	10	30	19	57	20.95%	18.40%
Fu	12	12	4	6	37.12%	2.40%
Jacobs2	25	25	15	29	37.71%	10.23%
Poly1a	15	15	16	16	53.75%	36.93%
Poly2a	15	30	16	32	53.75%	39.18%
Shirts	8	99	12	134	16.48%	-

When the results are considered, it is seen that the performance of the lower bounds strongly depends on the data. According to the results, we can say that higher gaps may be observed when

- the area of the rectangle parts gets smaller
- the number of the rectangular pieces increases
- when the represented portion for the shapes increases

Thus, higher gaps do not guarantee that the cutting pattern can be that much improved. However, lower gaps indicate a lower possibility to improve the generated cutting pattern. In other words, lower gaps are much more conclusive for the result.

4.2 Edge Matching Approach

The proposed methodology to generate a good feasible solution is inspired by a puzzle reassembly idea. It is a fact that, the maximum efficiency of a cutting pattern is %100 and it can be achieved only if the shapes are placed side by side with perfectly matching edges and, hence with no spaces between any two pieces. Figure 20-b illustrates an example of a cutting pattern which has %100 of efficiency. As it is seen in Figure 20-b, the edges for all pieces are completely matching each other. Figure 20-a illustrates a contrary case where the edges are not aligned to each other. In this case, for the same stock width, a higher length is required in order to position all these 5 pieces.

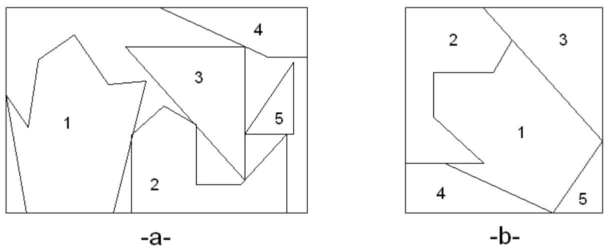


Figure 20 Cutting pattern examples for 5 pieces

The ideal efficiency of %100 can be achieved only if a perfect match exists between pieces. Therefore, it may be possible to state that there is a positive relation between edge matches and efficiency. Kopardekar and Mital (1999), make a study on extracting and generalizing human intuitive in laying out parts and they reported that operators perform higher utilizations when edge matches of pieces are maximized.

The result of their study also points out this relation. The solution methodology proposed in this thesis relies on generating cutting patterns by considering ‘edge match-efficiency’ relation observed in theory as well as in practice.

Thus, the cutting patterns are constructed by positioning the pieces like they are pieces of an apictorial puzzle. Note that, the pieces do not necessarily form a solid with perfect matches as in puzzles in cutting stock problems. There may be some spaces between pieces even under the cutting pattern with optimal stock material utilization.

Basically, the idea of the edge matching approach is to position the pieces in such a way that its highest fitting edges are aligned with a fitting edge of another piece or a fitting boundary of the stock material. It is usually required to rotate piece in specific angles in order to align them. In a continuous space, it is not easy to determine the position which yields the maximum fit. To approximate the continuous rotation and formulate the problem as a discrete combinatorial problem, two assumptions are made.

1. The pieces will be positioned one by one and the position remains same to the end of the construction phase.
2. Each piece should be positioned such that at least one vertex of the piece is aligned with a boundary vertex or with a vertex of previously inserted piece.

Note that, although each insertion step of algorithm aims at maximizing edge matches, the overall objective of the algorithm is still to maximize the efficiency of stock material usage. This efficiency is tried to be accomplished by minimizing the length of the stock material used, given that the stock width is constant.

Based on the assumptions, final position of a piece is determined in each iteration. Hence, the inserted piece is assumed to be cut out off the stock material at the end of iteration. Then, the remaining part of the stock material becomes the new boundary of the stock material to be considered for next iteration. Each of these iterations consists of three main steps: ‘preparation’, edge matching’ and ‘boundary update’. For an *N-piece* problem, it is expected to end up with *N* iterations provided that the size of the stock material (area of the boundary) is large enough to place all those *N* pieces.

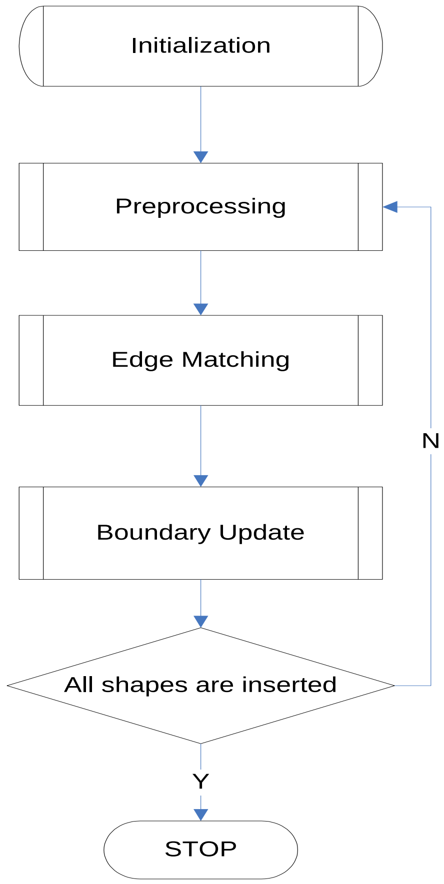


Figure 21 Main steps of a single iteration of the algorithm

As it is illustrated in Figure 21, there is an initialization phase before starting to iterate the algorithm. Initialization and iteration phases are explained in the following subsections. The summary of the algorithm is presented in Appendix C.

4.2.1 Initializing Phase

This phase inputs the stock material and pieces in different types of shapes to be cut off. The Cartesian coordinates are used to define the vertices of the pieces and stock material. For each shape, the number of vertices and the total number of pieces in the form of that shape are also used as input parameters at the initialization phase. The calculations of the area, the lengths of edges, and the interior angles of each shape type are computed.

4.2.2 Preprocessing Phase

This preprocessing phase consists of three main steps: shape smoothing, insertion order determination and increment amount determination. The first step is smoothing operation for each shape, which is required to reduce the computational effort. Then, an insertion order is determined for the pieces followed by the edge matching. Finally, a limit on the length increment is determined to control the maximum length of stock these three steps are explained further below.

Smoothing Shapes

Shape smoothing is considered for two reasons. The first reason is to decrease the complexity of the problem. The second reason is to prevent some infeasible situations. Thus, we have two types of smoothing operations. For both of the smoothing types, the shapes are reformed. In order to have an acceptable reformation on the shape, the change in the area of a shape is limited within specified tolerance limits (taken as %3 of the actual area). Figure 22 illustrates the result of a smoothing operation of type 1. Vertex 3 of the shape is deleted and the number of vertices is reduced to five with an area increment within tolerance limit of 1% of actual area.

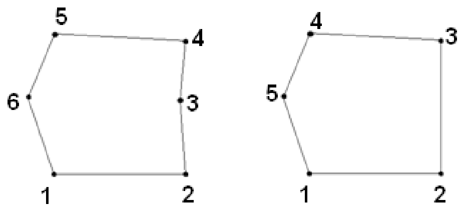


Figure 22 Smooth type 1 – Decreasing the complexity of the shape

If the vertex makes the shape concave, it can safely be deleted without decreasing the total area. If it is not concave, the shape will be smaller when the vertex is deleted which is not allowed. For convex shapes, it is definitely not possible to make such a kind of reduction.

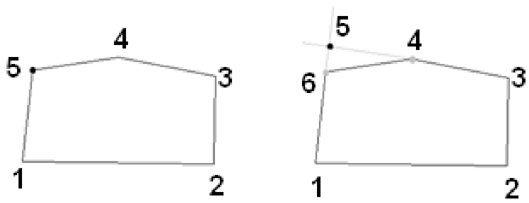


Figure 23 Smooth type 2 – Introducing a new vertex with a proper interior angle.

Figure 23 illustrates the smoothing operation of type 2. Due to the nature of the algorithm, if all interior angles of a shape are greater than the interior angles of the boundary, a feasible position cannot be obtained for the shape. In this case, a new vertex is introduced to get a smaller angle that fits into the boundary. Since this is done to control infeasibility, the increment on the area of shape is not considered. As it is shown in the Figure 23, vertex 5 is created on the intersection points of the extensions for edges [5,1] and [4,5], hence, the total number of the vertices is increased to six.

Insertion Order

In our algorithm, pieces are positioned one by one and the position is fixed to the end of stock material. Thus, the insertion order plays an important role in the performance of the overall algorithm. One of the most commonly used insertion criteria is the placement of the largest piece first. Using this criterion, the shapes are ordered with respect to their areas in descending order, but placing the piece with longest perimeter is also employed in this study. As an alternative method, it is also tested to merge pieces before inserting them. Since, the pieces get larger when they are merged; they are ordered with respect to the areas.

Although, there is only one single stock material at the beginning of the problem, any piece insertion at each iteration may split the stock material into different regions as it is shown in Figure 24. In this case, more than one region should be considered while inserting the next selected piece. In this study, when there is more than one region, the boundary of the smallest region is selected first.

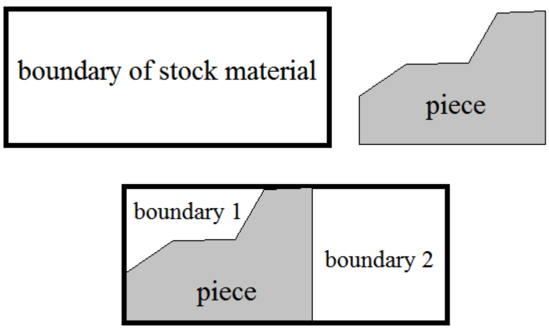


Figure 24 Two different boundaries emerge after positioning the piece

Limiting the Stock Length Increases

Although the approach provides maximum edge matching, the main focus is still minimizing the stock length. Therefore, a limit for the increment of the length is considered to hold the pieces on possible left most position on the stock. In the case that this limit is exceeded, a shorter edge fit must be preferred to have control on stock length.

4.2.3 Edge Matching Phase

Edge matching is the main step of the algorithm since it determines the position of each piece on the stock material. In order to determine the position with the maximum edge match, there are many issues to be considered. First of all, a piece must remain within the boundaries after being positioned. This will be controlled through overlap checks. Therefore, the edge matching phase determines final location and orientation of the piece that yields the maximum edge matches.

Overlap Check

Overlap control is made using trigonometric computations. When a piece is inserted, each edge is checked whether there is an intersection in such a way that none of the edge should intersect with the edges of boundary. The method for controlling edge intersection was described before. However, point in polygon test is considered in a different way than the mathematical model formulation in Chapter 3. For the edge matching method, each piece must be positioned inside the boundaries. A position is feasible only if there is no overlap between a piece and the boundary. Thus, the overlap check is routinely performed during any edge matching iteration to see if the resulted position is feasible.

Positioning with Edge Match

Each piece should be positioned in such a way that at least one vertex of the piece should be match with a boundary vertex or with the vertices of previously inserted pieces. Figure 25 illustrates the positioning procedure with this assumption. First of all, the piece is moved onto boundary such that a vertex of that piece is matched with a vertex of the boundary as it is shown in Figure 25-a. This is called as a ‘vertex match’. Then the piece is rotated clockwise (CW) direction over the matching vertex until one of the edge of the piece hits the boundary as in Figure 25-b. Then an alternative position is obtained by rotating the piece in the counter clockwise (CCW) direction till the edge of piece hits the boundary as in Figure 25-c. For each rotation an ‘edge fit’ is observed.

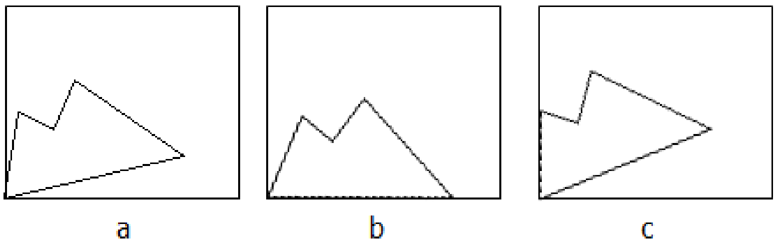


Figure 25 Illustration for positioning a single piece

For both positions in Figure 25-b and 25-c, the overlap check is performed. If there is no overlap, the length of fit along edges is calculated and recorded. In this way, all possible vertex match combinations are obtained and edge fit lengths are computed. The position which provides the longest length of edge fit is set as the final position for that shape. For instance in Figure 25, there are 4 vertices of boundary and 5 vertices of the piece that yields 4x5 = 20 possible matching combinations. Note that some of the matches will not be feasible due to the overlaps. The longest length fit for our example is illustrated in Figure 25-b.

In order to predetermine some of the infeasible cases before carrying out the positioning operations, interior angles for the matching points are computed. If the interior angle of the piece is larger than the interior angle of the boundary, there is no need for further feasibility check.

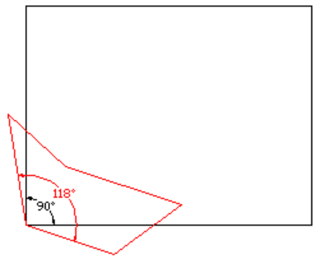


Figure 26 Overlap predetermined using interior angles

Figure 26 illustrates this kind of an overlap. Since the interior angle of the piece is 118^{o} and it is greater than the interior angle of the boundary which is 90^{o}, there is no feasible rotation to position the piece over the illustrated point. This means that, there is no need for overlap checks or length of edge fit for that point. Smoothing type 2, which was mentioned before, is made for any piece which has no vertices in feasible angles.

All vertices are numbered in CCW direction. Let *i* denote the i^{th} vertex of the piece to be inserted and *j* denote the j^{th} vertex of boundary. Our edge matching algorithm can be stated as below:

S1. Let:

i: vertex of piece   i = 1 to N

j: vertex of boundary   j = 1 to M

S2. For all i and j

Set maximum length to zero

If interior angle on i^{th} vertex is smaller than the interior angle on j^{th} vertex do:

i. Move the i^{th} vertex of piece onto j^{th} vertex of boundary

a. Compute the α angle between the edge |i, i+1| and edge |j, j+1|

1' Rotate the piece CCW over vertex i, by α.

2' Check for overlaps. If there is any overlaps go to '**b**'.

3' Compute the length of edge fit between the boundary and the piece

4' If the length is greater than the maximum length, update the maximum length and record the values of α, i and j as maxα, max_i and max_j, respectively.

b. Compute the α angle between the edge |i, i-1| and edge |j, j-1|

1' Rotate the piece CW over vertex i, by α.

2' Check for overlaps. If there is any overlap change 'j''.

3' Compute the length of edge fit between the boundary and the piece

4' If the length is greater than the maximum length, update the maximum length and record the values of α, i and j as maxα, max_i and max_j, respectively.

S3. Move max_i^{th} vertex of the piece to the max_j^{th} vertex of the boundary and rotate the piece as maxα^o.

When all possible vertex matches are tested, the algorithm returns the rotation angle with max_i and max_j values that yields the maximum edge fit.

4.2.4 Boundary Update Phase

As it is mentioned before, once a piece is inserted within the boundary, the position of that piece is preserved throughout the cutting pattern generation. Hence, the remaining part of the boundary can be used as a feasible region for the remaining pieces. Thus, when a shape is positioned, the vertices of the boundary are updated for the next iteration. In Figure 27, the shaded regions illustrate the area within the boundaries before and after insertion of a shape. At the beginning, the vertices of the boundary are defined by A-B-C-D-A. As it is seen, after the piece is positioned, the vertices of the boundary are updated as A-B-C-D-h-g-f-e-A since only this region is feasible to position a new piece. Note that, when an irregular shape is inserted on the stock, the boundary also becomes an irregular shape itself. In this way, of the algorithm generates solutions also for irregular shaped stock materials.

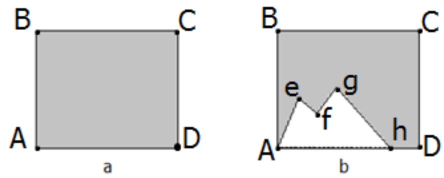


Figure 27 Boundary update

To update the boundary, Weiler-Atherton clipping algorithm is used (Weiler and Atherton., 1977). According to the algorithm

- A vertex on the boundary, which is not on the edge of the piece, is selected.
- Then, the vertices are visited in CCW direction until one of the edges of the pieces is hit.
- The points visited before hitting the piece are updated as vertices of the boundary.
- After hitting a point on the edge of the piece, switch to the edges on the piece and traverse in reverse direction.
- As soon as hitting an edge, switch to the boundary and continue traversing vertices of it in reverse direction.

As a summary, while traversing vertices of a shape, the algorithm evolves by traversing vertices of the piece and boundary in reverse directions until returning to the starting point. For Figure 27-b, assuming that starting vertex is C, the steps of algorithm will be traced as follows:

- Vertex -C- is not on the edge of piece, record it
- Move to vertex -B-
    - It is not on the edge of piece, record it
- Move to next vertex -A-
    - It is also on the edge [A,h] of piece
    - Move back to starting point of that edge, i.e 'A', record it
    - Switch to traversing the piece, change your direction
- Move to point -e-
    - It is not on the boundary, record it
- Move to vertex -f-
    - It is not on the boundary, record it
- Move to vertex -g-
    - It is not on the boundary, record it
- Move to vertex -h-
    - It is also on the edge [D,A] of the boundary
- Move back to starting point of that edge, i.e, -D-, record it
    - Switch the boundary, change your direction.
- Move to the vertex -C-, record it
- Since vertex -C- is the starting point, STOP.

The updated boundary is: C-B-A-e-f-g-h-D-C.

Creating Boundary Generator Lists

The boundary updating process seems to be a bit complicated in the first glance. It requires using two different lists for the boundary and the piece, named as *boundary generator lists*. To create a boundary list, start form a vertex of the boundary and record all points that you visit until you reach the starting point. For the piece list, the same procedure is carried out. The lists which belong to the situation shown in Figure 27 are given as follows:

Boundary list: A - h - D - C- B

Piece list: A – h – g – f – e

In the algorithm, doubly linked lists are used as data structures to build up and store boundary generator lists. Each node of these link lists includes the vertex id, the vertex type indicator (to indicate if it belongs to boundary or piece), the address of the next node in the list, the address of the previous node in the list, indicator to see if there is a link to other list (Vertices ‘A’ and ‘h’ are links for other lists since they are common for both lists), the address of the vertices on the other list (if there is a link to other list), and finally an indicator to see if the node is visited or not. The flow chart of the algorithm that creates a boundary list is given in Appendix C.

Traversing Boundary Generator Lists

The following procedure is used to obtain a loop, which will correspond to a new boundary after placing a piece on a stock. The procedure is based on Weiler-Atherton clipping algorithm, which was just mentioned above. The procedure iterates as follows: First start with the first element in boundary list. If the same element also exists in the other list, switch to the other list and change your traversing direction and continue in the opposite direction. Otherwise, continue with the next element until you reach the starting point. The resulting loop corresponds to the new boundary. The boundary loop obtained for the example in Figure 27 is illustrated in Figure 28.

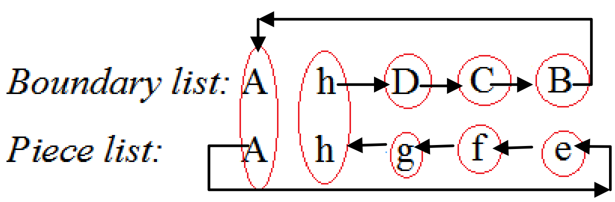


Figure 28 Boundary generation using lists

As a result, according to Figure 28, updated boundary is : A-e-f-g-h-D-C-B.

When there is a unique boundary loop to be obtained, the algorithm works in this way. However, when there is more than one boundary after a piece is positioned, the Weiler-Atherton clipping algorithm needs some modifications. These modifications require vertex analysis for both boundary and the piece.

Vertex and Case Analysis

With the help of boundary generator lists, it is possible to generate the boundaries when more than one region emerges after inserting a piece on the stock. For example, in Figure 29, the shaded region illustrates the piece. Since the piece splits up the boundary into two different regions, two different boundaries will be generated for the next iteration.

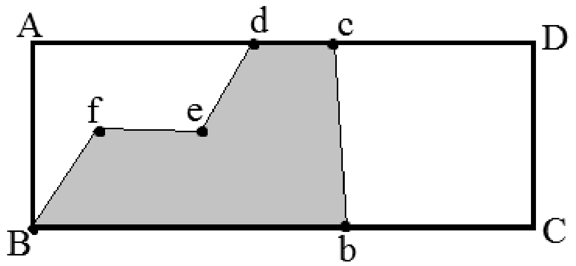


Figure 29 A Piece that causes more than one boundary

Using the methods described before, the boundary generator lists for the situation in Figure 29 are constructed and traversed as it is illustrated in Figure 30.

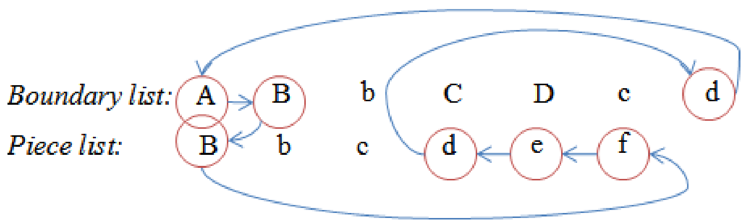


Figure 30 Generation of the first boundary from Figure 28

In Figure 30, vertex A is the start point. Following the arcs, the first boundary is determined as A-B-f-e-d-A. When the visited vertices in the list are deleted, the remaining vertices are listed as it is shown in Figure 31. When vertex -b- is selected as a new starting point, a loop is created with only two different vertices. Since it is not possible to generate a closed region using less than three vertices, vertex C is used as the new starting point instead of vertex b.

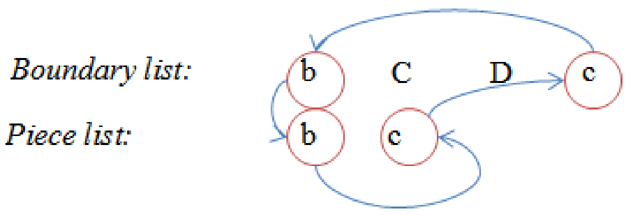


Figure 31 Loop with no boundary

When C is selected as the starting point, the loop shown in Figure 32 is generated. Hence, the second boundary becomes C-D-c-b. Since all the elements in the lists are visited at the end of this iteration traversing the lists is terminated.

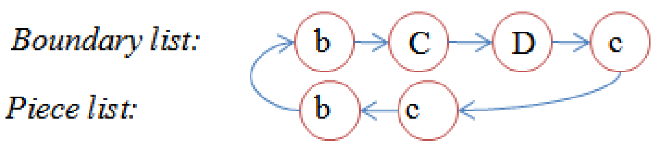


Figure 32 Second boundary generated by the lists

Unfortunately, the method still requires additional modifications to correct the boundaries in the case that there are common vertices for two different boundaries. Figure 33 illustrates this situation, where point A is a common vertex for both boundary 1 and boundary 2. In the first run, the boundary will be generated using the lists. Then, vertex A is going to be deleted from the lists since it is visited.

However, it is also part of boundary 2, and it is not possible to generate that boundary without vertex A. Therefore, to prevent this situation, vertices are analyzed for following information.

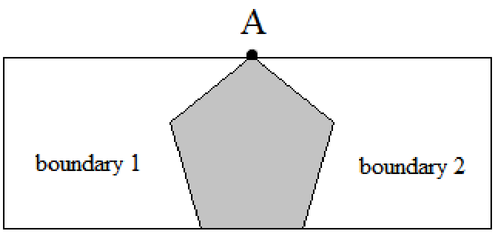


Figure 33 Common vertex for two different boundaries

1. Vertex of a piece is positioned on a boundary edge or not.
2. Vertex of a piece positioned on a boundary vertex or not.
3. Vertex id, which denotes the boundary edge which the piece vertex belong to, if the answers for previous statements are yes.

The vertices of the boundaries are also analyzed in the same manner.

The characterization of the vertices requires some basic distance comparisons:

If *a vertex i is positioned on an edge [j, j+1) then;*

distance between i and j must be equal to the distance between i and j+1

and

id for vertex i is j (since it is belong to the edge [j, j+1).

The flow charts for the vertex analysis can be seen in Appendix C. Using the results of the vertex analysis, the case analysis are performed to decide if there exists a shared vertex or not. Considering the possible relative positions of boundaries and pieces, 9 cases are determined. These cases are illustrated in Figure 34.

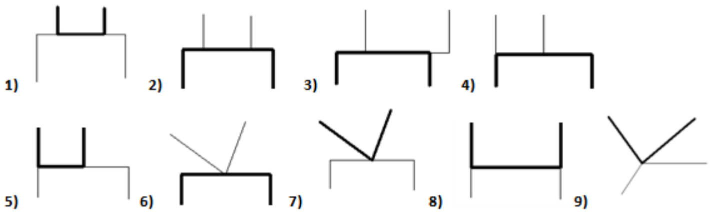


Figure 34 Possible cases of relative positions

In the figure, bold lines refer to boundary edges and other lines refer to piece edges. For the cases 6, 7 and 9, there are shared vertices. This type of vertices should be written twice in the boundary generator lists. In this way, when the elements of the first boundary are deleted, the vertex still will be remaining to generate second boundary. This analysis for boundary and piece are performed separately, and the results of vertex analysis are used as input. Flow charts for the case analysis can be seen in Appendix C.

When the case analysis is performed, the boundary creator lists are generated and then they are traversed to obtain the updated boundaries. As the boundaries are updated, our boundary traversing algorithm terminates. If there are still pieces to be positioned, preparations for new iterations are run.

4.3 Computational Study for Edge Matching Approach

In this section, the results of edge matching approach are presented. Our algorithm is coded in Visual Basic Studio 10.0 environment, using ANSI C language. A personal computer with Intel® Core™ i5 CPU M 520 @ 2.40 GHz processor and 4 GB RAM is used to run the algorithm. The algorithm is tested on the data sets that are obtained from ESICUP. Three different insertion order rule is tested for data sets: largest area first, longest perimeter first and merged pieces. Performance measures and computational results for each insertion order rule is presented in the following sections.

4.3.1 Performance Measures

In this study, stock material's utilization rate of the cutting patterns are taken as a performance measure and called as 'efficiency type I'. Higher values for efficiency type I, yields smaller scrap amounts for the cutting patterns. Since the raw material cost is the main cost item in cutting stock processes, using this performance measure is used in the literature. Thus, stock material utilization is the common performance measure in the literature.

Besides of the raw material cost, another important performance indicator is the machining cost for the cutting stock processes. Machining cost depends on the duration spent to cut off the pieces from the stock material. This duration can be expressed in terms of cutting path lengths like in the case of Chinese Postman problem. Figure 35 illustrates two different cutting patterns each consists of two regular hexagons, and their cutting paths respectively. In Figure 35-a, hexagons should be fully traced by the cutting tool in order to carve the piece out of the stock sheet. In Figure 35-b, there are two matched edges for the hexagons (one between the pieces and one with the stock sheet edge). Thus, the cutting tool can carve out the pieces from stock sheet just by cutting 10 edges for the case in Figure 35-b where as it needs to cut 12 edges in Figure 35-a.

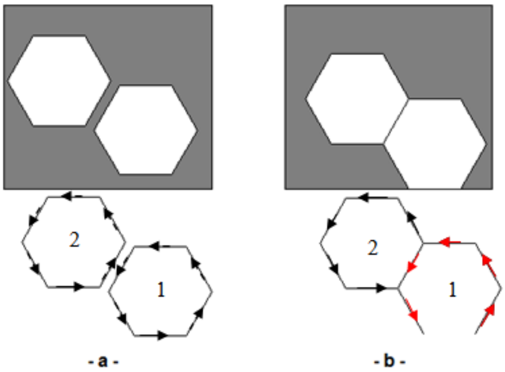


Figure 35 Cutting patterns with different cutting paths

In the edge matching approach, it is tried to position the pieces with matching edges. Although the main objective is minimizing the scrap rate, edge matching provides shorter cutting paths for the generated patterns. Since shorter cutting paths provides higher efficiencies in terms of machining time, another performance measure for this study is taken as the cutting path efficiency and called as 'efficiency type II'. This performance measure is the ratio of cutting path length to the total perimeter for all pieces. Note that, the value of efficiency type II cannot reach to 100% since it is not possible to obtain the length of cutting path length as zero.

$$\text{Efficiency Type II} = 1- \frac{(\text{Sum of perimeters for all pieces})-(\text{sum of matching edge lengths})}{(\text{sum of perimeters for all pieces})} \qquad (21)$$

During this study, CPU times are not taken as a performance measure and they are not used as a comparison tool for benchmarks. The main reason to that is smoothing process of the algorithm. Although smoothing process is embedded in the algorithm, it is still not automated enough. For some pieces of the data sets, manual pre-smoothing operations may be required. Addition to that the merging of the piece is done manually. Since the duration of these manual adjustments cannot be taken into account, it is not thought to be fair to use CPU times for pattern generation runs which are all less than 5 min.

4.3.2 Data Set

In order to evaluate the results of the algorithm, generated cutting patterns are to be compared with the results of selected benchmark problems with respect to the stated performance measures. To the best of our knowledge, the best published results for the selected benchmark problems are illustrated in Table 7.

Table 7 Best published results for selected benchmark problems

	Problem Specifications				Best Published (BP)					
	Total # of Pieces	Stock width	Total Area	Total perimeter	Length	Efficiency 1	Length of Cutting Path	Efficiency 2	Source	LB
1-Albano	24	4900	42658241.46	132687.60	9957.40	0.87	105854.50	0.20	a	9746.30
2-Blaz	28	15	323.97	396.76	25.84	0.84	314.40	0.21	a	21.13
3-Dağlı	30	60	3005.28	1230.20	58.20	0.86	868.49	0.29	a	47.49
4-Dighe2	10	100	10000.00	1349.20	100.00	1.00	454.02	0.66	a	100.00
5-Fu	12	38	1082.95	480.45	31.33	0.91	264.56	0.45	a	30.58
6-Jacobs1	25	40	391.98	404.63	11.50	0.85	257.65	0.36	b	9.40
7-Jacobs2	25	70	1331.00	845.81	24.70	0.77	595.33	0.30	b	22.17
8-Marques	24	104	7196.80	2277.83	78.00	0.89	1753.26	0.23	b	72.95
9-Shapes0	43	40	1595.80	1588.81	60.00	0.66	1273.61	0.20	a	-
10-Shapes1	43	40	1595.80	1588.81	56.00	0.71	1200.61	0.24	a	-
11-Poly1a	15	40	410.00	357.24	13.3	0.77	310.841	0.13	b	8.39
12-Poly2a	30	40	820.00	714.48	27.09	0.76	610.8011	0.15	b	16.48
13-Shirts	99	40	2160.12	1854.38	62.21	0.87	1419.91	0.23	a	-

a) Oliveira and Gomes (2006); b) Burke et. al. (2005)

In Table 7, first five columns illustrate the properties of the data sets such as total number of pieces, area of the pieces, width of the stock material and total perimeters of the pieces. The remaining columns denote the current best results. As it is seen, values for 'efficiency type I' are close to 1. However, the maximum of 'efficiency type II' is just 0.66. This is due to the fact that it is not possible to reach 100% efficiency for type II since it is almost impossible to have the cutting path length as zero.

4.3.3 Largest Area First (LAF)

As it is mentioned before, insertion order for the pieces has a vital importance on the generated cutting pattern since the algorithm is greedy. One approach to order the pieces in the data set with respect to their areas such that the largest piece is first to be positioned on the stock material. LAF's results are illustrated in Table 8.

Table 8 Results of the algorithm - LAF

	Length	Efficiency 1	Length of Cutting Path	Efficiency 2	GAP 1	GAP 2
1-Albano	11000.00	0.79	99188.14	0.25	-0.10	0.06
2-Blaz	30.59	0.71	315.17	0.21	-0.18	0.00
3-Dağlı	60.00	0.83	844.04	0.31	-0.03	0.03
4-Dighe 2	100.00	1.00	454.02	0.66	0.00	0.00
5-Fu	31.64	0.90	250.10	0.48	-0.01	0.05
6-Jacobs 1	12.00	0.82	246.31	0.39	-0.04	0.04
7-Jacobs 2	26.00	0.73	570.84	0.33	-0.05	0.04
8-Marques	81.00	0.85	1898.39	0.17	-0.04	-0.08
9-Shapes 0	58.00	0.69	1089.25	0.31	0.03	0.14
10-Shapes 1	58.00	0.69	1089.25	0.31	-0.04	0.09
11-Poly1a	13.5	0.76	255.96	0.28	-0.02	0.18
12-Poly2a	27.78	0.74	523.60	0.27	-0.03	0.14
13-Shirts	61.00	0.89	1273.69	0.31	0.02	0.10

In Table 8, GAP 1 and GAP 2 denotes the relative differences for efficiency type I and efficiency type II, respectively, based on the best published results:

$$GAP\ 1 = \frac{\text{Stock Length}_{\text{BP}} - \text{Stock Length}_{\text{LAF}}}{\text{Stock Length}_{\text{BP}}} \tag{22}$$

$$GAP\ 2 = \frac{\text{Length of Cutting Path}_{\text{BP}} - \text{Length of Cutting Path}_{\text{LAF}}}{\text{Length of Cutting Path}_{\text{BP}}} \tag{23}$$

When the values for efficiency type I are inspected, it is seen that the worst case among the data sets has a negative gap of 10% for *Albano*. The results for *Shapes 0* and *Shirts* are better than the best published results with positive gaps of 3% and 2% respectively. On the average, the stock material utilization for the edge matching approach is 4% less with respect to the best published results when the largest piece is selected first.

According to the results of GAP 2, it is seen that for all data sets except Marques, generated cutting patterns have shorter cutting path lengths with respect to the cutting paths lengths of the benchmark results. This means that, edge matching approach provides cutting patterns which are more efficient in terms of machining times. Resulting cutting patterns are illustrated in Figure 36.

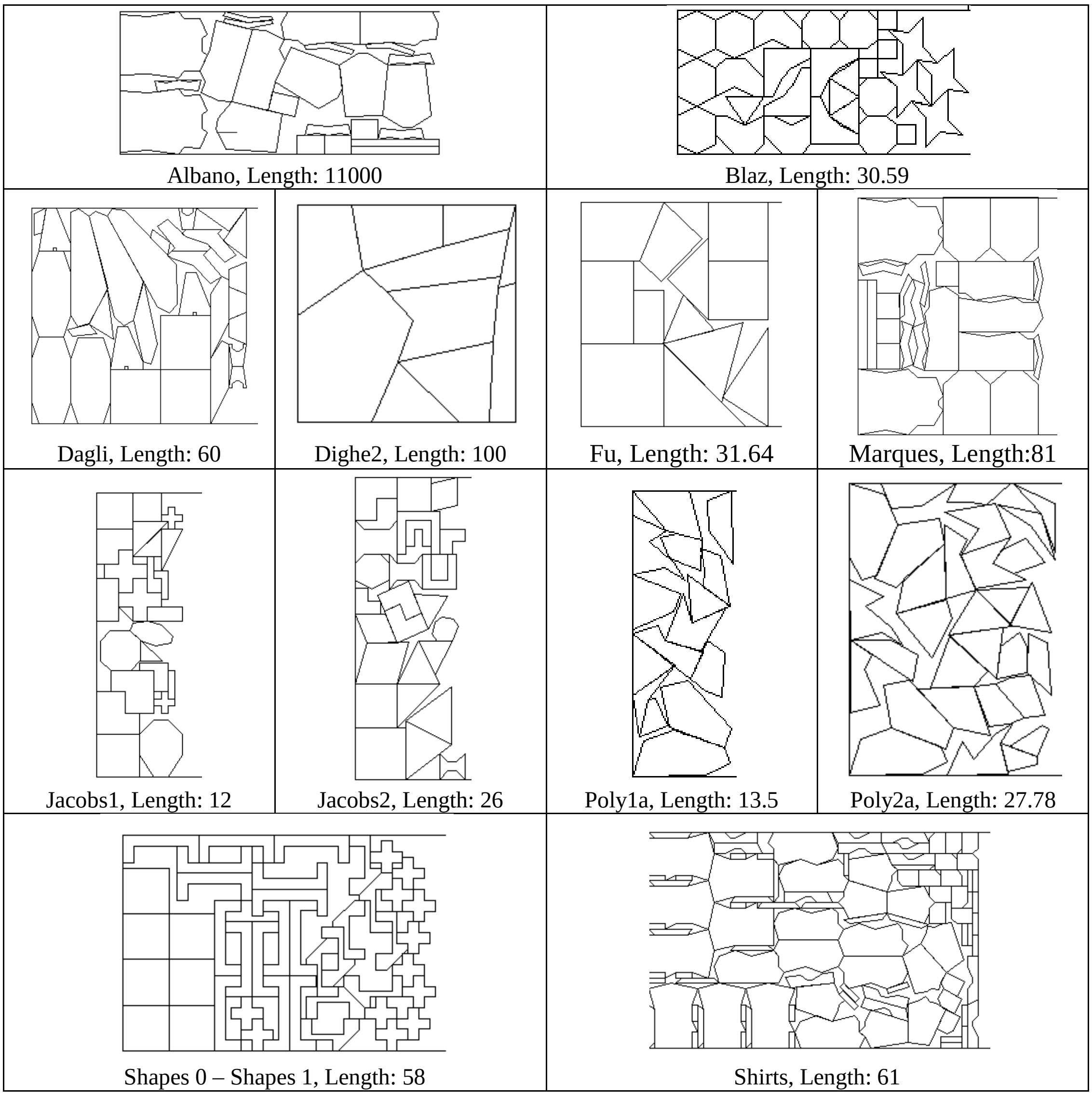


Figure 36 Generated cutting patterns – LAF

4.3.4 Longest Perimeter First (LPF)

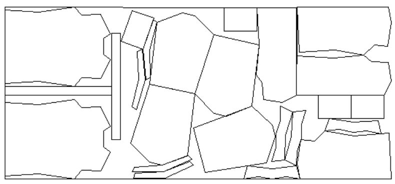
In the edge matching approach, the algorithm searches for the position which provides maximum length of edge match. A piece that have a long perimeter may have chance to maintain a longer match. Thus, as another insertion order rule, the pieces are ordered with respect to the length of their perimeters and the one with the longest perimeter is selected to be positioned first. In this section the results for the insertion rule "longest perimeter first" (LPF) is presented. Table 9 illustrates the summary of the results.

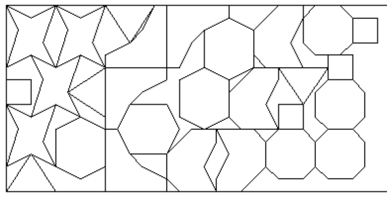
Table 9 Results of the algorithm - LPF

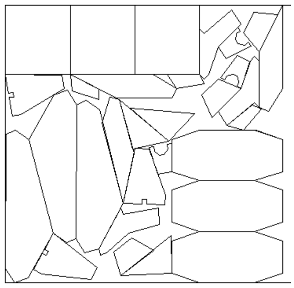
	Length	Efficiency 1	Length of Cutting Path	Efficiency 2	GAP 1	GAP 2
1-Albano	11000.00	0.79	109849.83	0.17	-0.10	-0.04
2-Blaz	30.00	0.72	312.11	0.21	-0.16	0.01
3-Dağlı	60.00	0.83	805.00	0.35	-0.03	0.07
4-Dighe 2	100.00	1.00	454.02	0.66	0.00	0.00
5-Fu	33.00	0.86	255.92	0.47	-0.05	0.03
6-Jacobs 1	11.28	0.87	211.47	0.48	0.02	0.18
7-Jacobs 2	25.10	0.76	564.35	0.33	-0.02	0.05
8-Marques	83.00	0.83	1636.59	0.28	-0.06	0.07
9-Shapes 0	55.49	0.72	1073.65	0.32	0.08	0.16
10-Shapes 1	55.49	0.72	1073.65	0.32	0.01	0.11
11-Poly1a	14	0.73	238.55	0.33	-0.05	0.23
12-Poly2a	27	0.76	504.29	0.29	0.00	0.17
13-Shirts	61.00	0.89	1273.69	0.31	0.02	0.10

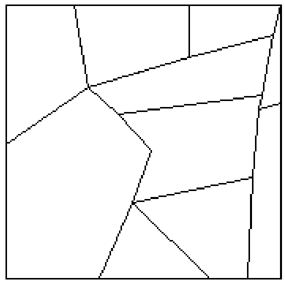
According to the GAP 1 results, 5 data sets out of 13 are better than the best published results. The result for *Dighe 2* is the same with the best published result. Note that the data set of *Dighe 2* is the only data set that reaches the 100% stock material utilization, thus it is not possible to improve that solution. The worst result in terms of GAP 1 is obtained from the data set of *Blaz* with a gap of 16%. On the average the stock material utilization for generated patterns is 3% less according to the best published results.

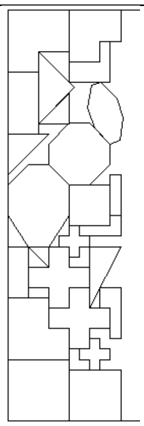
When GAP 2 is considered, except the data set of *Albano*, all of the generated patterns are better than the patterns of the best published results in terms of cutting path length. On the average, the cutting path is 9% sorter in the edge matching approach with LPF. Generated cutting patterns are presented in the Figure 37.

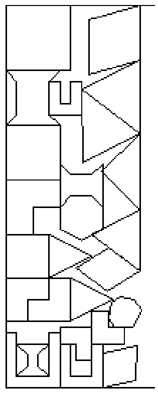

Albano, Length: 11000

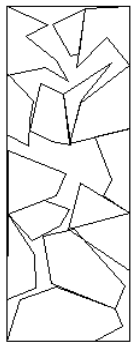

Blaz, Length: 30

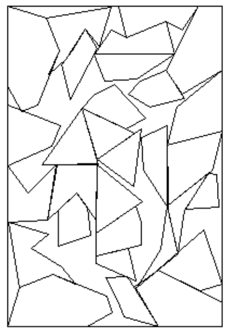

Dagli, Length: 60

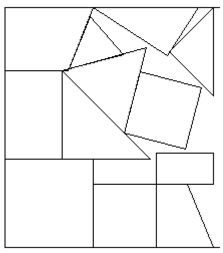

Dighe2, Length: 100

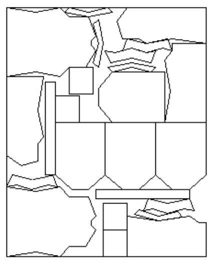

Jacobs1, Length: 11.28

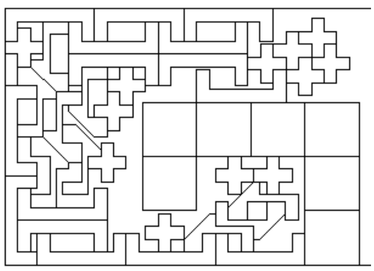

Jacobs2, Length: 25.1


Poly1a, Length: 14


Poly2a, Length: 27


Fu, Length: 33


Marques, Length: 83


Shapes 0 – Shapes 1, Length: 55.49

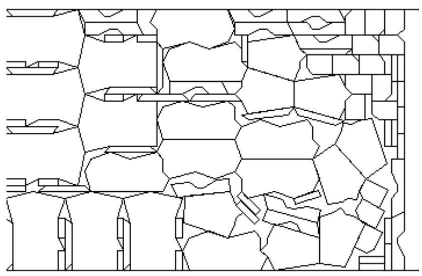

Shirts, Length: 61

Figure 37 Generated cutting patterns for LPF

4.3.5 Computational Result for Merged Pieces

In this section, effect of merging the pieces is analyzed. Here, the term '*merge*' stands for combining two identical pieces to form a new single piece. With this merge operation, the total numbers of the pieces decreases and the complexity of the problem reduce for the given data set. Briefly, with the help of merging some pieces together, a new data set with less but larger pieces can be derived.

There are infinitely many relative positions for two pieces to be merged. In order to simplify the process, three main cases for positions are considered to merge two identical pieces. These cases can be described as follows;

1. Fix the initial position for one of the pieces and rotate the other one by 180^o
2. Fix the initial positions for both of the pieces
3. Fix the initial position for one of the pieces and use the mirror reflection of the other piece.

For each case, the pieces are brought together in four different ways. Figure 38 illustrates these ways for the first case where one of the pieces is rotated by 180^o.

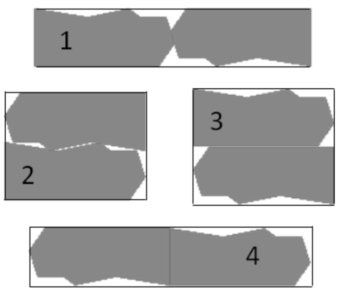


Figure 38 Alternative combining positions for a sample piece from Albano.

As it is seen in the Figure 37, the rotated piece is positioned at the left, at the bottom, at the top and at the right side of the other pieces respectively. For all these four positions, the one which provides the minimum enclosing rectangle is to be selected.

After the decision for the position is made for each case, they are compared to between each other. Figure 39 illustrates an example for a sample piece from *Albano*. Figure illustrates the 180^o rotated case, same positions case and mirror reflection case in 39-a, 39-b, and 39-c, respectively. In the case that the area of enclosing rectangle for combined shapes is equal to the summation of the areas for separate enclosing areas of these shapes, the merge operation is not done. In other words, enclosing rectangles of each piece must be overlapped to approve the position. In Figure 39-b and 39-c there is no overlap so only the position in 39-a is a valid position to merge the pieces. If there are more than one valid position, than the one with the smallest enclosing rectangle is selected.

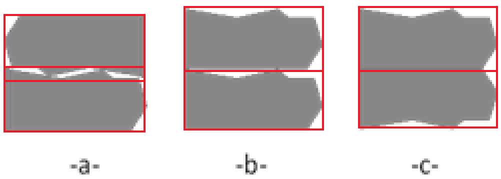


Figure 39 Three cases for a sample piece from Albano

Among the selected benchmark problems which are tested, some data sets do not include duplications of same piece. Since merge is considered for identical pieces in this study, only the data sets of *Albano, Blaz, Dagli, Marques, Shapes* and *Poly 2a* is are used here.

Table 10 Number of merged pieces in each cases

	Albano	Blaz	Dagli	Marques	Shapes	Poly 2a	Total
180^o rotated	5	2	6	5	1	14	33
Same position	1	0	0	2	0	0	3
Reflection	0	0	0	0	0	0	0
None	2	5	2	2	3	1	15
Total	**8**	**7**	**8**	**9**	**4**	**15**	**51**

In Table 10, amounts of merged piece types are presented for each data set. As it is illustrated in the table, there are 8 different shapes of pieces for Albano and 5 of them are resulted better when they are positioned as in case 1. Similarly, most of the pieces form up preferable new pieces when they are merged as in case 1. For these 7 data sets, 15 pieces out of 51 do not constitute a valid position to merge them. For the remaining 36 pieces, 33 of them are provided in case 1.

According to these results, the data sets of the selected benchmark problems are updated then the algorithm is compiled using LAF. The reason to select the LAF is that the areas of the pieces increase as they are combined and it is difficult to find a room on the stock sheet for those large pieces.

The results of the runs are summarized in the Table 11. As it is seen, for all the data sets except *shapes 0*, value of *efficiency 1* is lower than the results of LAF and LPF. On the average, merging the pieces performs 14% worse than BP results in terms of *efficiency 1* values.

Table 11 Computational results for merged pieces

	Types of Pieces	Total Number of Pieces	Length	Efficiency 1	Length of Cutting Path	Efficiency 2	GAP 1	GAP 2
Albano	8	19	11512.98	0.76	106753.60	0.20	-0.16	-0.01
Blaz	7	24	32.97	0.66	340.78	0.14	-0.28	-0.08
Dagli	18	22	63.69	0.79	832.49	0.32	-0.09	0.04
Marques	9	17	83.81	0.83	1875.98	0.18	-0.07	-0.06
Shapes0	5	29	57.97	0.69	1147.02	0.28	0.03	0.12
Shapes1	5	29	57.97	0.69	1147.02	0.28	-0.04	0.05
Poly2a	15	16	36.79	0.56	574.10	0.20	-0.36	0.07

As it is mentioned before; '*Each piece should be positioned such that at least one vertex of the piece should be aligned with a boundary vertex or with a vertex of previously inserted piece.'* When the pieces are merged together, this rule is omitted for them. Thus, the problem is relaxed in a manner. However, another constraint is involved, such that the merged pieces must stay together on the stock material. According to the resulting cutting patterns with merged pieces, it is seen that relaxation effect of merging the pieces is negligible with respect to the tightening effect of it. Generated cutting patterns for merged pieces are illustrated in the Appendix D.

4.3.6 Evaluation of Computational Results

When the results in the previous subsections are analyzed, we have the overall picture as summarized in Table 12. Last two columns of the table present the source for the best efficiency among the results. According to these results, best performance on efficiency 1 and efficiency 2 have the same source for 7 data sets. In other words, for 7 data sets, maximum stock material utilization is obtained when maximum cutting path efficiency is provided, or vice versa.

Table 13 illustrates the average values of efficiencies and the relative gaps to the best published results of the selected benchmark problems. According to the results, merging the pieces performs worst with an average efficiency 1 value of 71% and average efficiency 2 value of 0.23. LPF performs slightly better than LAF according to the average values. On the other hand, LPF performs better than best published results for 5 data sets, where LAF can beat just for one data set.

Table 12 Summary of efficiencies for cutting patterns

	BP		LAF		LPF		Merge		Source of Best Result	
	Eff. 1	Eff. 2	Eff. 1	Eff. 2	Eff. 1	Eff. 2	Eff. 1	Eff. 2	Eff. 1	Eff. 2
1-Albano	0.87	0.20	0.79	0.25	0.79	0.17	0.76	0.20	BP	LAF
2-Blaz	0.84	0.21	0.71	0.21	0.72	0.21	0.66	0.14	BP	BP-LAF-LPF
3-Dağlı	0.86	0.29	0.83	0.31	0.83	0.35	0.79	0.32	BP	LPF
4-Dighe 2	1.00	0.66	1.00	0.66	1.00	0.66	-	-	BP-LAF-LPF	BP-LAF-LPF
5-Fu	0.91	0.45	0.90	0.48	0.86	0.47	-	-	BP	LAF
6-Jacobs 1	0.85	0.36	0.82	0.39	0.87	0.48	-	-	LPF	LPF
7-Jacobs 2	0.77	0.30	0.73	0.33	0.76	0.33	-	-	BP	LAF-LPF
8-Marques	0.89	0.23	0.85	0.17	0.83	0.28	0.83	0.18	BP	LPF
9-Shapes 0	0.66	0.20	0.69	0.31	0.72	0.32	0.69	0.28	LPF	LPF
10-Shapes 1	0.71	0.24	0.69	0.31	0.72	0.32	0.69	0.28	LPF	LPF
11-Poly1a	0.77	0.13	0.76	0.28	0.73	0.33	-	-	BP	LPF
12-Poly2a	0.76	0.15	0.74	0.27	0.76	0.29	0.56	0.20	LPF	LPF
13-Shirts	0.87	0.23	0.89	0.31	0.89	0.31	-	-	LAF-LPF	LAF-LPF

Table 13 Average values for the performance measures - I

	Average Efficiency 1	Average GAP 1	Worst GAP 1	Average Efficiency 2	Average GAP 2	Worst GAP 2
LAF	0.80	-0.04	-0.18	0.33	0.06	-0.08
LPF	0.81	-0.03	-0.16	0.35	0.09	-0.04
Merge	0.71	-0.14	-0.36	0.23	0.02	-0.08
BP	0.83	-	-	0.28	-	-

In order to make a deeper comparison, the respective sources for the best published results are analyzed with respect to efficiency I values. In Table 14, results of LAF and LPF are compared to the methods in the literature. As it is seen, all 4 methods are slightly differs from each other, in terms of Efficiency 1 measure.

Table 14 Average values for the performance measures - II

	Average Efficiency 1	Average GAP 1	Worst GAP 1
LAF	0.80	-0.04	-0.18
LPF	0.81	-0.03	-0.16
BP (a)	0.84	-0.02	-0.04
BP (b)	0.81	-0.02	-0.15

(a) Oliveira and Gomes (2006); (b) Burke et. al. (2005)

Relation between Efficiency I and Efficiency II

Matching the edges to generate cutting patterns with higher material utilization was the main point of this study. Thus we expected to see a positive relationship between material utilization and cutting path efficiencies which is provided by edge matches. In order to see the relation between Efficiency 1 and Efficiency 2, results are compared for total of BP, LAF and LPF. Using these 39 results (13 results from each), the plot in Figure 40 is obtained. Pearson correlation coefficient for Efficiency 1 and Efficiency 2 is obtained as 0.65, indicating that the efficiencies are positively related, however this is not a strong relationship.

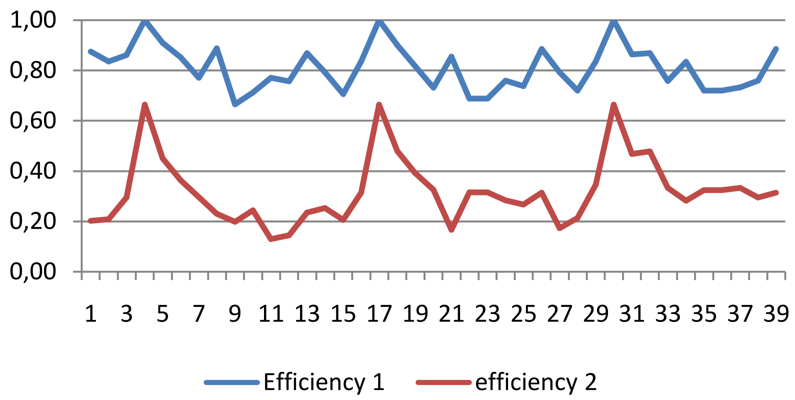


Figure 40 Efficiencies for BP, LAF and LPF

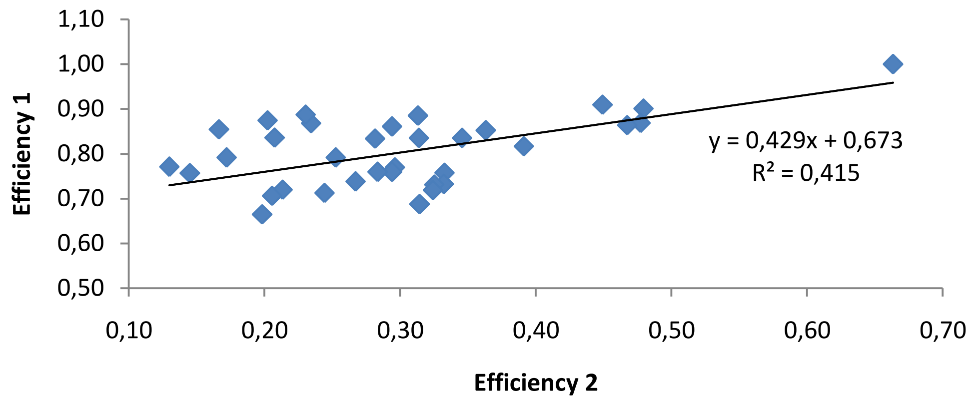


Figure 41 Efficiency 1 vs. Efficiency 2

In Figure 41, the scatter plot also illustrates the positive relation between efficiencies. However, according to the R-square coefficient value of 0.4153, the linearity seems to be weak. Especially for the *efficiency 2* values below 0.3, linearity do not exists (R-square value of 0.0032). Figures 42 and 43 illustrate the same plot for the *efficiency 2* values above 0.25 and for the *efficiency 1* values above 0.80, respectively. According to the R-square values, linear relationship between efficiencies is slightly stronger for higher values.

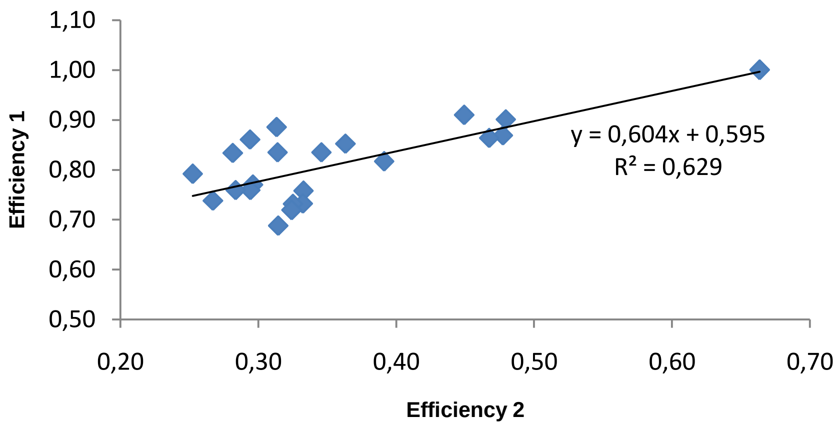


Figure 42 Efficiency 1 vs. Efficiency 2 | efficiency 2 > 0.25

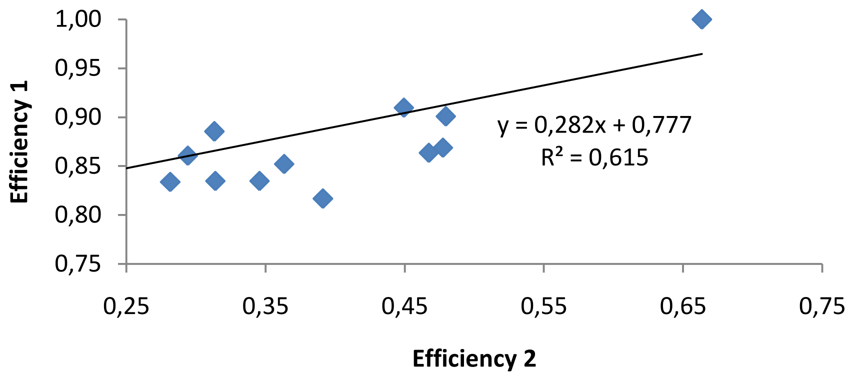


Figure 43 Efficiency 1 vs. Efficiency 2 | efficiency 1 > 0.80

Table 15 illustrates the values of the parameters such as total number of pieces, average number of vertices, average length of perimeter and etc. for each data set. The effect of these parameters on the correlation of efficiency I and efficiency II measures is also analyzed during this study. When the effect of parameters are considered separately and the effect of each factor is observed ignoring the other factors, it is seen that the results about the relationship between efficiency values and any of these parameters are inconclusive. On the other hand, one may expect the correlation between efficiency values to be weaker when the data set is more irregular. In order to obtain a numerical measure for the irregularity of a data set, a ratio is computed under the given assumptions.

1. As the number of different pieces and average number of vertices increase, the irregularity also increases.
2. As the total number of pieces and ratio of stock width to the average length of perimeter increase, the irregularity decreases.

Based on the assumptions, the ratio of irregularity (RI) for each data set is computed as follows.

$$\mathrm{RI} = \frac{(\text{Number of different pieces})\text{x}\ (\text{Average number of vertices})}{(\text{Total number of pieces})\text{x}\left(\dfrac{\text{Stock width}}{\text{Average length of perimeter}}\right)} \tag{24}$$

Table 15 Parameters of data sets

Data Sets	Stock width	Number of different pieces	Total number of pieces	Average number of vertices	Total number of vertices	Average Length of perimeter	Stock width / average perimeter	Norm. RI
Albano	4900	8	24	6.83	164	5528.65	0.89	0.38
Blaz	15	7	28	7.5	210	14.17	1.06	0.25
Dagli	60	10	30	7.3	219	41.01	1.46	0.23
Dighe2	100	10	10	4.7	47	134.92	0.74	1.00
Fu	38	12	12	3.58	43	40.04	0.95	0.58
Jacobs1	40	25	25	5.16	129	16.19	2.47	0.30
Jacobs2	70	25	25	5.36	134	33.83	2.07	0.38
Marques	104	8	24	7.08	170	94.91	1.1	0.31
Shapes0	40	4	43	8.75	376	36.95	1.08	0.08
Shapes1	40	4	43	8.75	376	36.95	1.08	0.08
Poly1a	40	15	15	4.6	69	23.82	1.68	0.41
Poly2a	40	15	30	4.6	138	23.82	1.68	0.18
Shirts	40	8	99	6.63	656	18.73	2.14	0.00

For the normalized RI values which are greater than 0.30, the scatter plot for Efficiency I and Efficiency II values are illustrated in Figure 44. Based on the observations, any strong relationship between the parameters of the data set and efficiency values cannot be figured out. Thus, it is not possible to generalize a conclusive result about the effect of interaction between those stated parameters on the correlation between efficiency values.

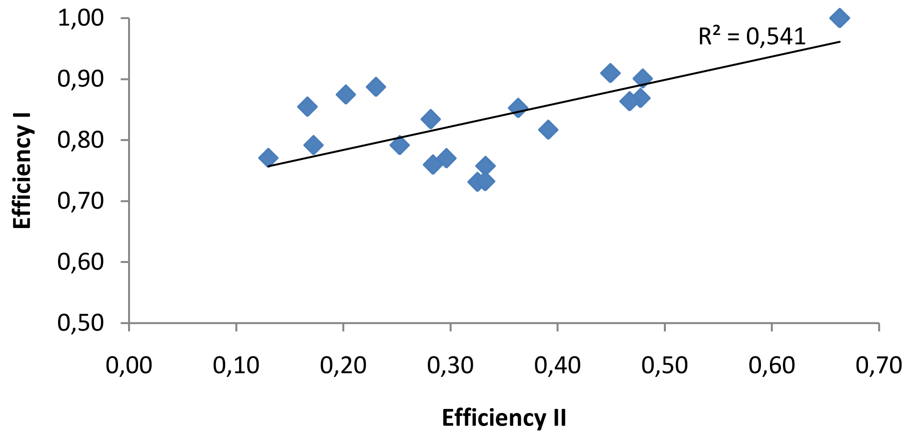


Figure 44 Efficiency I vs Efficiency II - RI > 0.30

As it is mentioned before, it is not possible to have the efficiency II measure as 100% since the length of cutting path cannot be zero. During this study, maximum value for efficiency type II is observed as 66% for the data set which also provides maximum of efficiency type I value as 100%. Note that, the average efficiency II value is 34%. Although this study stands for the irregular shaped pieces, we also compute efficiency II measures for three different rectangular 2DCSP benchmark problems.

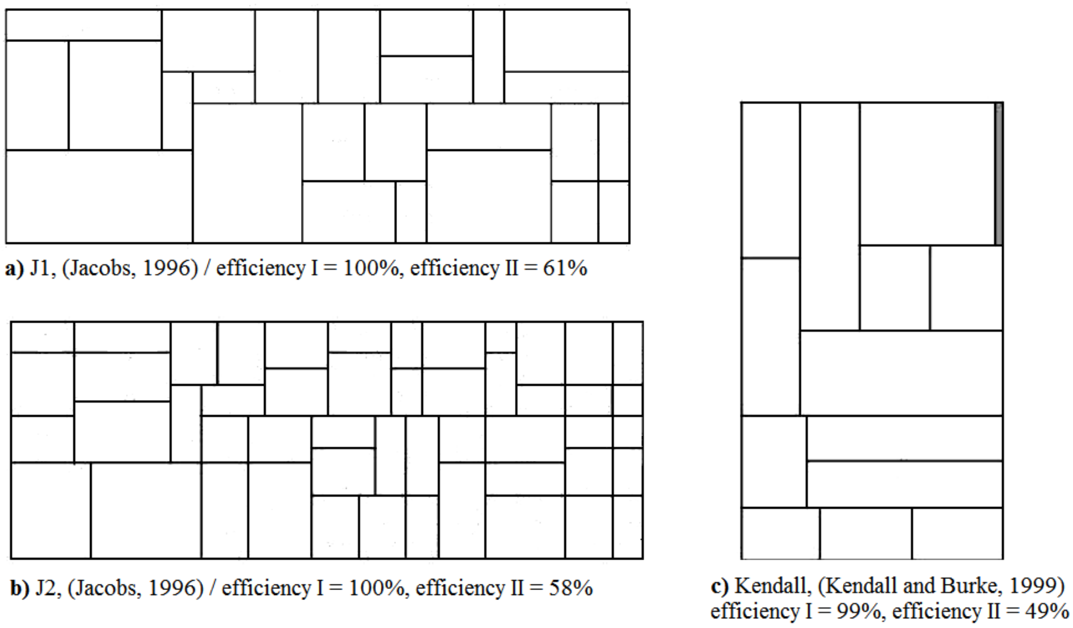


Figure 45 Data sets for rectangular 2DCSP and their optimal solutions

As it is seen in Figure 45, the edges of rectangular pieces are all touching each other in their optimal cutting patterns. Due to these edge matches, efficiency II values are above the average of efficiency II measures that we compute for irregular 2DCSP benchmark problems. Thus, it can be stated that there is also a positive relationship between efficiency I and efficiency II measures in rectangular 2CSP.

CHAPTER 5

CONCLUSION AND DISCUSSION

In this thesis, we study an irregular shaped two-dimensional cutting stock problem, which is widely observed in practical applications. Our goal is to get cutting patterns efficient in terms of material usage for the problem. To achieve this goal, we propose a mathematical model, a lower bound and a heuristic method inspired by puzzle reassembly idea. The motivating point of the idea is that a cutting pattern yields 100% efficiency only if all the edges of the shapes are in touch with each other such that there is no un-utilized space left on the stock sheet.

Before the implementation of the edge matching approach, the related works for the problem is reviewed. During this literature search, it is realized that there is no comprehensive study to generate the optimal solution for the problem, due to the high complexity of the problem. To form a new method for the evaluation of the results for irregular shaped 2DCSP, first, a mixed integer non-linear mathematical model is formulated for the problem. Then, for larger scale problems where exact formulation becomes intractable, a lower bounding scheme is developed to test the performance of the heuristic solutions. Finally, we propose a new heuristic procedure based on an ‘edge matching approach’. To implement the heuristic, an algorithm is developed and coded to generate cutting patterns. The algorithm has the following properties.

- The shape for the stock material is not restricted except for circular shapes. Thus, the algorithm can also be used for scrapped sheets and plates.
- Rotating the pieces is allowed in any angle, and this angle is not determined by searching. Instead, rotation angle is directly computed to align the pieces together. Thus, initial orientation of a piece does not affect its final position of it. Nevertheless, this also implies that the algorithm is not suitable for the problems with rotation angle constraint.
- Pieces are positioned within the stock material one by one and the place of a piece does not change once it is positioned.
- On the generated cutting pattern, the pieces touch each other. Thus, it is generally not possible to compact the layout by shifting the pieces.
- Positioning the pieces with the edge matching approach reveals a new performance measure for the problem. Since the length of the cutting paths is also optimized when the pieces are aligned together, generated cutting patterns are also expected to be efficient in terms of machining times.

In order to test the performance of the algorithm, some benchmark problems are used. Three different methods; 'largest piece first' (LAF), 'longest perimeter first' (LPF), and 'merging the pieces' are compared for benchmark problems. LPF performed best both in terms of stock material utilization and cutting path efficiencies.

There are some further improvements that may be implied to the algorithm in order to obtain higher matches. First of all, two pieces are connected to each other by their vertices. It may also be allowed to connect the vertex of a piece to the midpoint on the edge of another piece. This will increase the range of search space, providing a chance to obtain better results.

As it is described before, length increments are limited by the algorithm. However, any analysis on alternative increment limitations was not performed during the study. It should be wise to consider this issue, and propose a systematic method to determine length increment limits, to have a better performance. Addition to that, when the same edge fit length is maintained by more than one position, the one with minimum x coordinate is selected by the algorithm. The effect of this tie-breaker rule may also be analyzed. Also a tie-breaker rule can be studied for the placement order of the pieces which have equal areas or equal lengths of perimeter. In our study, the last introduced piece is used first in such equality. However it is possible to use the largest area first rule for perimeter equalities or longest perimeter first rule for the area equalities.

Although some rules are determined for the smoothing process, any analysis on area increment tolerances was not performed during the study and it is taken as 3% of actual area. However, it is observed that allowing higher amount of area increments provides better results for some cases. Addition to that, decreasing the complexity is one of the main reasons to smooth the pieces, but the effect on computational time is not tested. Thus, smoothing operation on the pieces is another issue to be studied further.

For the implementation of the algorithm in the industry, some further issues may be required to be considered such that the offset distances between pieces to have valid cutting processes. In this study, the pieces are aligned together; assuming that diameter of cutting tool is negligible. The offset distances must be added to the area of the pieces for real life implementations.

There are several future research directions that can be undertaken. The edge matching approach provides a cutting pattern with a single construction phase, but no improvement is performed on the pattern. The main reason for that is closely touching pieces on the layout. As it is mentioned before, it is generally not possible to find space to shift the pieces toward the axis. Thus, it is not much possible to compact and improve the layout further. However, our mathematical model formulation can still be applied to generated patterns, in order to compact them. As a future study, it is a very interesting topic to study on the improvement phase by using the mathematical model proposed in this work.

As another further extension, the pieces with curved edges may also be included in the problem. In current conditions, the inputs for the geometry of a shape are just the Cartesian coordinates of the vertices. Curved edges may be considered as arcs, using their origin and radius besides the end point coordinates. Line-curve and curve-curve intersections must be considered to prevent overlaps for the case.

REFERENCES

1. **Adamowicz, M., Albano, A.** (1976), "Nesting Two-Dimensional Shapes in Rectangular Modules", *Computer Aided Design*, 27-33, vol 8.

2. **Art Jr., R.C.** (1966), "An Approach to the Two-Dimensional Irregular Cutting Stock Problem.", Technical Report 36.Y08, IBM Cambridge Scientific Centre.

3. **Bennell, J.A., Dowsland, K.A.** (1999), "A Tabu Tresholding implementation for the irregular stock cutting problem.", *International Journal of Production Research*, 4259 – 4275, vol. 37, no: 18.

4. **Bennell, J.A., Dowsland, K.A.** (2001), "Hybridising Tabu Search with Optimisation Techniques for Irregular Stock Cutting". *Management Science*, pp. 1160–1172, vol. 47 (8).

5. **Bildirici, I.Ö.** (2003), "Mekansal Veri Analizinde Point in Polygon Testi", TUJK 2003 Yili Bilimsel Toplantisi: CBS ve Jeodezik Ağlar Çalistayi, Seminer Bildirileri (24-26 Eylül 2003), 94-103.

6. **Blazewicz, J.,** et al. (2004), "Parallel Tabu Search Approaches for Two-Dimensional Cutting". *Parallel Processing Letters*, 23–32, vol. 14.

7. **Bounsaythip, C. and Maouche, S.** (1996), "A Genetic Approach to a Nesting Problem", Proceedings of the Second Nordic Workshop of Genetic Algorithms, 89-104.

8. **Burke, E. K., Kendall, G.** (1999), "A Comparison of Meta-Heuristic Algorithms for Clustering Rectangles." *Computational Industrial Engineerin,* 383–386, vol. 37(1-2).

9. **Burke, E.K.** et al. (2005), "A New Bottom-Left-Fill Heuristic for the Two-Dimensional Irregular Packing Problem.", *Operations Research,* 587-601, vol. 54, no: 3.

10. **Chernov, N.** et al. (2010), "Mathematical Model and Efficient Algorithms for Object Packing Problem.", *Computational Geometry,* 535 – 553, vol. 43.

11. **Cunninghame – Green, R.** (1989), "Geometry, Shoemaking and the Milk Tray Problem.", *New Scientist,* Agust, 50 – 53,

12. **Dori, D., Ben-Bassat, M.** (1984), "Efficient Nesting Of Congruent Convex Figures.", CACM (27), 1984, 228-235.

13. **Downsland, K.A., Dowsland, W.B.** (1995), "Solution Approaches to Irregular Nesting Problems.", *European Journal of Operational Research,* 506 – 521, vol. 84, no: 3.

14. **Dowsland K.A.,** et al. (2002), "An Algorithm for Polygon Placement Using Bottom Left Strategy.", *European Journal of Operational Research,* 371 – 381, vol. 141.

15. **Dychoff, H. (**1990), "A Typology of Cutting and Packing Problems.", *European Journal of Operational Research*, 145–159, vol. 44.

16. **Garey, M., Johnson, D.S.** (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness." Freeman, New York.

17. **Gomes, A.M., Oliveira, J.F.** (2002), "A Two-Exchange Heuristic for Nesting Problems." *European Journal of Operational Research,* 359–370, vol. 141.

18. **Gomes, A.M. and Oliveira, J.F**. (2006), “Solving Irregular Strip Packing Problems by Hybridizing Simulated Annealing and Linear Programming.”, *European Journal of Operational Research*, 811-829, vol. 171.

19. **Haims, M.J.** (1966), “On Optimal Two-Dimensional Allocation Problem.”, Ph.D. dissertation, Dept. of Elec. Engineering, New York University, Bronx, Technical Report, 136 – 400.

20. **Haims, M.J., and Freeman, H.** (1970), “A multistage solution of the template-layout problem.” *IEEE Trans. Syst. Science and Cybernetics*, 145-151, vol. SSC-6, no: 2.

21. **Heckmann, R., Lengauer, T**. (1995). “A Simulated Annealing Approach to the Nesting Problem in The Textile Manufacturing Industry.”, *Annals of Operations Research*, 103–133, vol. 57.

22. **Ismail, HS & Hon KKB.** (1995), “The Nesting of Two-dimensional Shapes using Genetic Algorithms.”, *Proceeding of The Institution of Mechanical Engineers*, 115-124, Part B, vol. 209.

23. **Jakobs S.** (1996), “On Genetic Algorithms for the Packing of Polygons.”, *European Journal of Operational Research,* 165 – 181, vol. 88.

24. **Kantorovich, L.V.** (1960), “Mathematical Methods of Organizing and Planning Production.”, *Management Science,* 366 – 422, vol. 6. (Russian trans. in 1939)

25. **Kopardekar, P., Mital, A.** (1999), “Cutting Stock Problem: A Heuristic Solution Based on Operator’s Intuitive Strategies.”, *International Journal of Computer Integrated Manufacturing*, 371 – 382, vol. 12/4.

26. **Li, Z., Milenkovic, V.J.** (1993). “The Complexity of the Compaction Problem.”, In: Lubiw, L., Urrutia, J. (Eds.), Proceedings of the Fifth Canadian Conference on Computational Geometry, University of Waterloo, Ont., Canada, 7–11.

27. **Li, Z., Milenkovic, V.** (1995). “Compaction and Separation Algorithms for Non-Convex Polygons and Their Applications.”, *European Journal of Operational Research*,539–561, vol. 84, no: 3.

28. **Mahadevan, A.** (1984), “Optimization in Computer-Aided Pattern Packing.”, Ph.D. dissertation, North Carolina State University.

29. **Milenkovic, V.J.** (1999), “Rotational Polygon Containment and Minimum Enclosure Using Only Robust 2D Construction.”, *Computational Geometry,3 - 19*, vol. 13.

30. **Oliveria, J. F.** et al. (2000), “TOPOS – A New Constructive Algorithm for Nesting Problems.”, *OR Spectrum*, 263 – 284, vol. 22.

31. **Ono, O. and Watanabe, G.** (1997), “Genetic Algorithms for Optimal Cutting.”, Evolutionary Algorithms in Engineering Applications, 515–530.

32. **Petridis V.** et al. (1998), “Varying Fitness Functions in Genetic Algorithm Constrained Optimization: The Cutting Stock and Unit Commitment Problems.”, *IEEE Transactions on System*s, 629 – 640, vol. 28.

33. **Qu, W., Sanders, J.** (1987), “A Nesting Algorithm for Irregular Parts and Factors Effecting Trim Losses.” *International Journal of Production Research*, 381 – 397, vol. 25, no: 3.

34. **Ribeiro, C.** et al. (1999), “Applying Constraint Logic Programming to the Resolution of Nesting Problems.”, *Pesquisa Operacional*, 239 – 247.

35. **Stoyan, Y.G.** et al. (1996). “Mathematical Model and Method of Searching for a Local Extremum for the Non-Convex Oriented Polygons Allocation Problem.”, *European Journal of Operational Research,* 193–210, vol. 92, no: 1.

36. **Theodoracates, V. E.; Grimsley, J. L**. (1995), “The Optimal Packing of Arbitrarily-Shaped Polygons Using Simulated Annealing and Polynomial-Time Cooling Schedules.”, *Computer Methods in Applied Mechanics and Engineering*, 53-70, vol. 27.

37. **Wäscher, G.** et al. (2007), “An Improved Typology of Cutting and Packing Problems.”, *European Journal of Operational Research,* 1109 – 1130, vol: 183.

38. **Weiler, K., Atherton, P**. (1977), “Hidden Surface Removal Using Polygon Area Sorting.”, Proceedings of the SIGGRAPH, 214-222.

39. **Whitwell, G.** (2004), “Novel Heuristic and Metaheuristic Approaches to Cutting and Packing.”, Ph.D. Thesis, School of Computer Science and Information Technology, University of Nottingham.

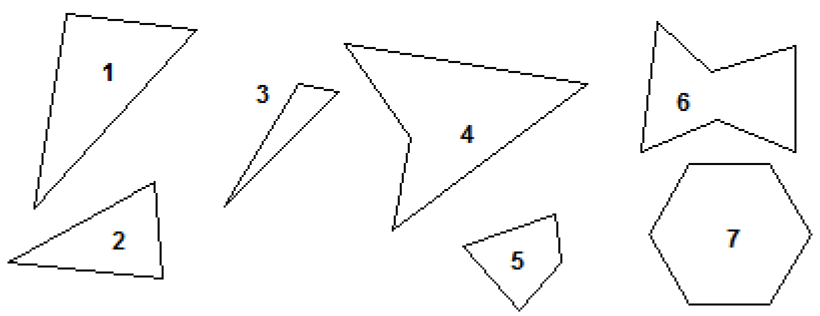
APPENDICES

APPENDIX A - PIECES USED IN MATHEMATICAL MODEL

Cartesian Coordinates for the Pieces

Piece No							
1	x	-30	70	-10			
	y	-70	40	50			
2	x	35	30	-60			
	y	-35	25	-25			
3	x	9	80	55			
	y	-65	5	10			
4	x	-80	40	-110	-70		
	y	-82.5	7.5	35.5	-25		
5	x	0	25	22	-35		
	y	0	30	60	40		
6	x	-30	7	55	55	3	-30
	y	-30	-10	-30	35	20	-50
7	x	-25	25	50	25	-25	-50
	y	-43.3013	-43.3013	0	43.3013	43.3013	0

N shape data set includes the first N pieces given above.



APPENDIX B - Results for 3 - 7 shapes

Microsoft Excel 12.0 Answer Report
Worksheet: [3 shapes test run.xlsx]set1
Target Cell (Min)

Cell	Name	Original Value	Final Value
M15	obj function:	130	117.3858956

Adjustable Cells

Cell	Name	Original Value	Final Value
N11	cx	-5	0.34699432
O11	cy	0	-0.472700317
P11	alfa	0	0.00017407
N12	cx	60	54.65300567
O12	cy	-34	-33.65285914
P12	alfa	0	-3.062552585
N13	cx	0	0
O13	cy	15	10.56979413
P13	alfa	-20	-20

Microsoft Excel 12.0 Answer Report
Worksheet: [4shapes test run set1.xlsx]set1
Target Cell
(Min)

Cell	Name	Original Value	Final Value
M19	obj function:	320	280.9827221

Adjustable
Cells

Cell	Name	Original Value	Final Value
T13	cx	0	0.000624251
U13	cy	0	0
V13	alfa	0	
T14	cx	90	80
U14	cy	0	0
V14	alfa	0	0
T15	cx	60	60
U15	cy	-10	-10
V15	alfa	-45	-45
T16	cx	250	210.9842888
U16	cy	5	5
V16	alfa	0	8.17145E-05

Microsoft Excel 12.0 Answer Report
Worksheet: [5shapes test run set1.xlsx]set1
Target
Cell(Min)

Cell	Name	Original Value	Final Value
M19	obj function:	325	284.3325701

Adjustable Cells

Cell	Name	Original Value	Final Value
T13	cx	0	0
U13	cy	0	0
V13	alfa	0	0
T14	cx	80	78.35067555
U14	cy	0	0
V14	alfa	0	0
T15	cx	60	60
U15	cy	-10	-10
V15	alfa	-45	-45
T16	cx	250	219.7335093
U16	cy	5	5
V16	alfa	0	8.17142E-05
T17	cx	0	0.000624251
U17	cy	20	20
V17	alfa	0	-0.000435811

Microsoft Excel 12.0 Answer Report
Worksheet: [6shapes test run set1.xlsx]set1
Target
Cell(Min)

Cell	Name	Original Value	Final Value
M22	obj function:	420	362.2398246

Adjustable Cells

Cell	Name	Original Value	Final Value
T14	cx	0	5
U14	cy	0	0
V14	alfa	0	0
T15	cx	80	80
U15	cy	0	0
V15	alfa	0	0
T16	cx	60	60
U16	cy	-10	-10
V16	alfa	-45	-45
T17	cx	250	220
U17	cy	5	5
V17	alfa	0	0
T18	cx	0	0.000267606
U18	cy	20	20
V18	alfa	0	-0.000186824
T19	cx	320	262.2301249
U19	cy	0	-50

V19	alfa	0	-0.000160007

Microsoft Excel 12.0 Answer Report
Worksheet: [7shapes test run set1.xlsx]set1

Target Cell (Min)

Cell	Name	Original Value	Final Value
M23	obj function:	420	368.8450236

Adjustable Cells

Cell	Name	Original Value	Final Value
T15	cx	0	0
U15	cy	0	0
V15	alfa	0	0
T16	cx	80	80
U16	cy	0	0
V16	alfa	0	0
T17	cx	60	60
U17	cy	-10	-10
V17	alfa	-45	-45
T18	cx	250	217.8787111
U18	cy	0	0
V18	alfa	0	0
T19	cx	0	0.000342118
U19	cy	0	0
V19	alfa	0	-0.000238843
T20	cx	320	268.8454935
U20	cy	-50	-50
V20	alfa	0	-6.37201E-05
T21	cx	320	257.592069
U21	cy	50	50
V21	alfa	0	0

APPENDIX C - FLOW CHARTS

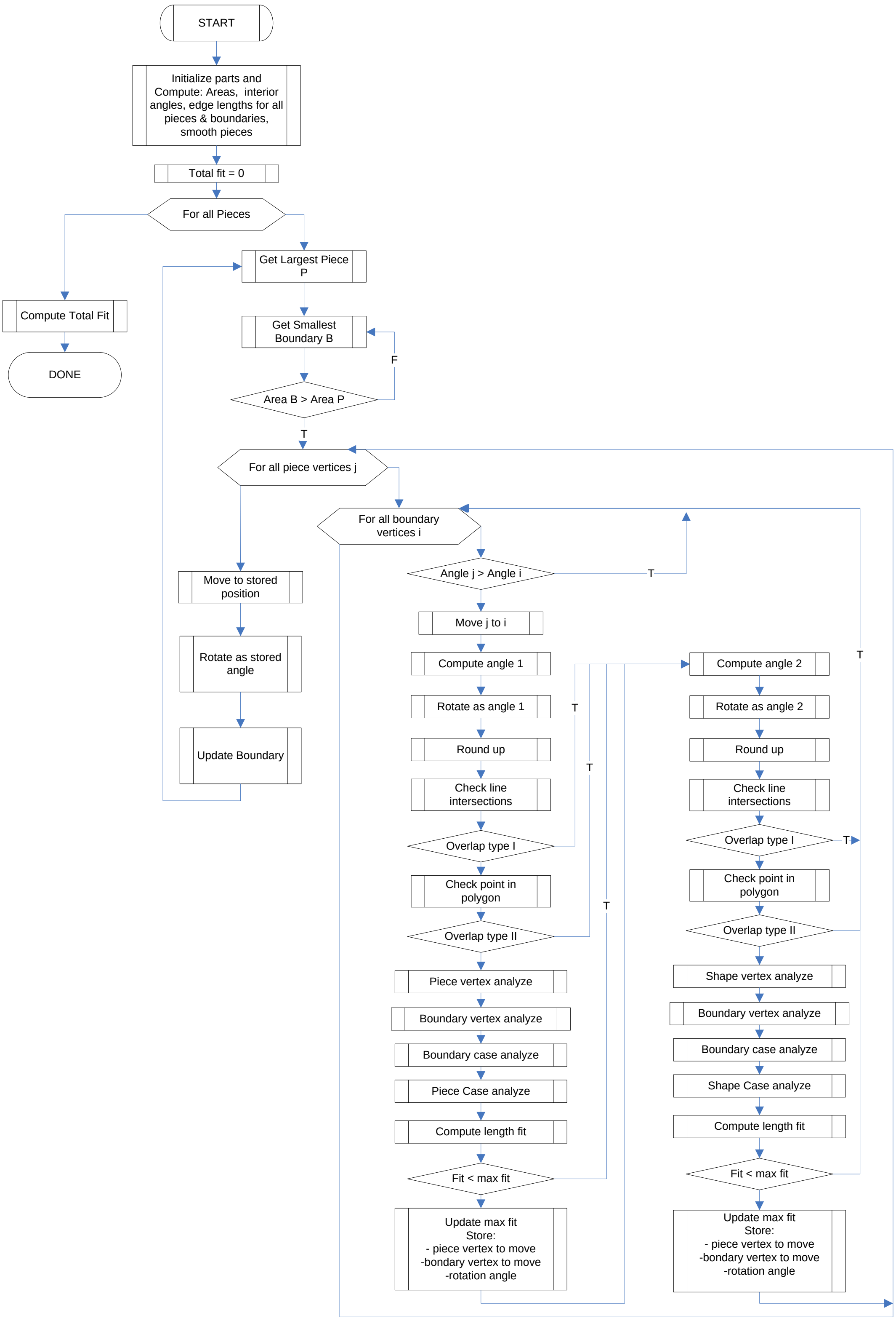


Chart 1 Flow Chart of Main Algorithm

Explanations:

```
Item.b;//vertex id from boundary
Item.s;//vertex id from piece
Item.type;//1 if there is a link to other list
```

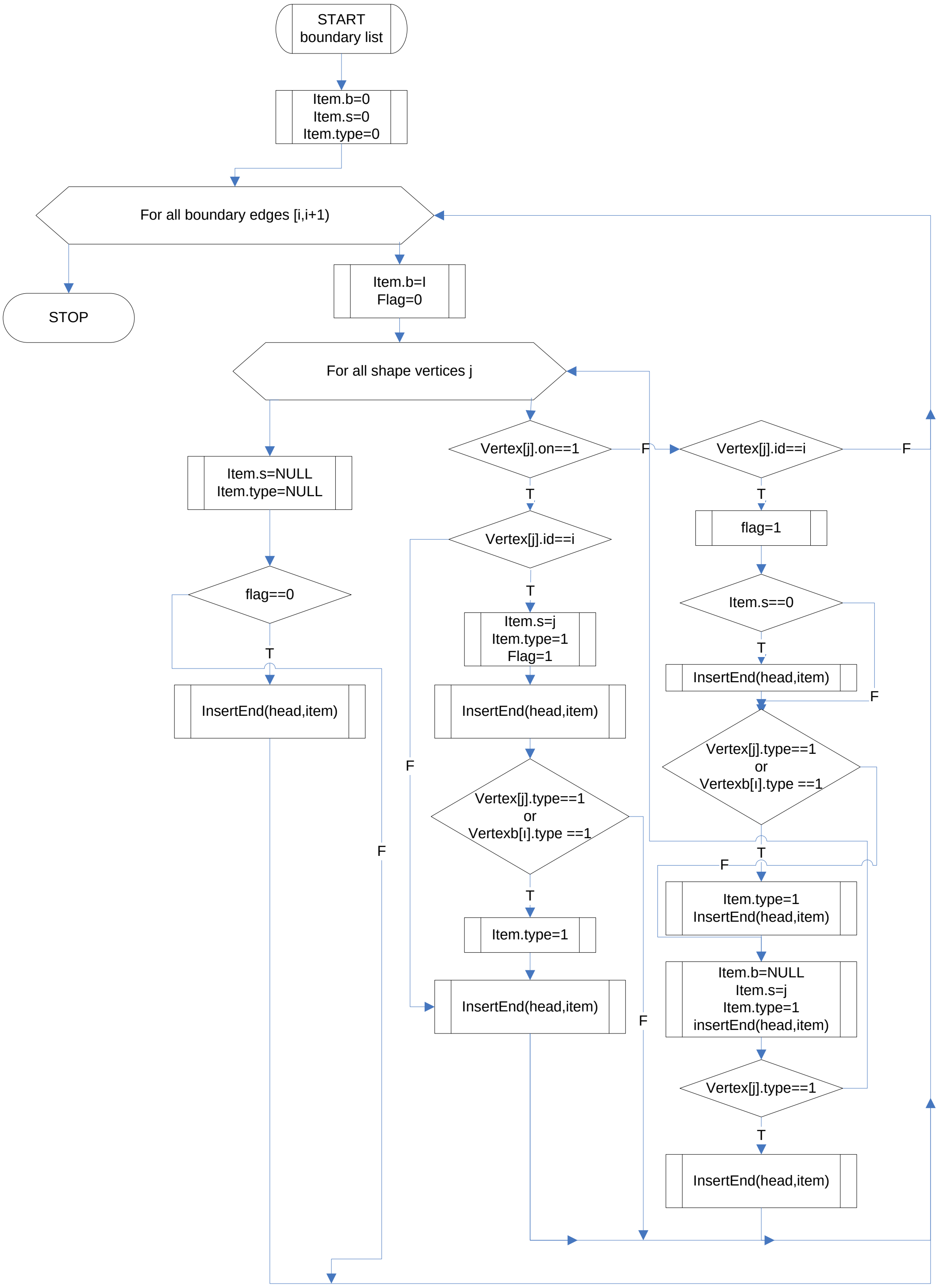


Chart 2 Flow Chart for Creating Generator list

```
Vertex.on; //1 if piece vertex is positioned on a boundary vertex
Vertex.line; 1 if piece vertex is positioned on a boundary edge
Vertex.id; //id of boundary edge that vertex of piece belongs to
```

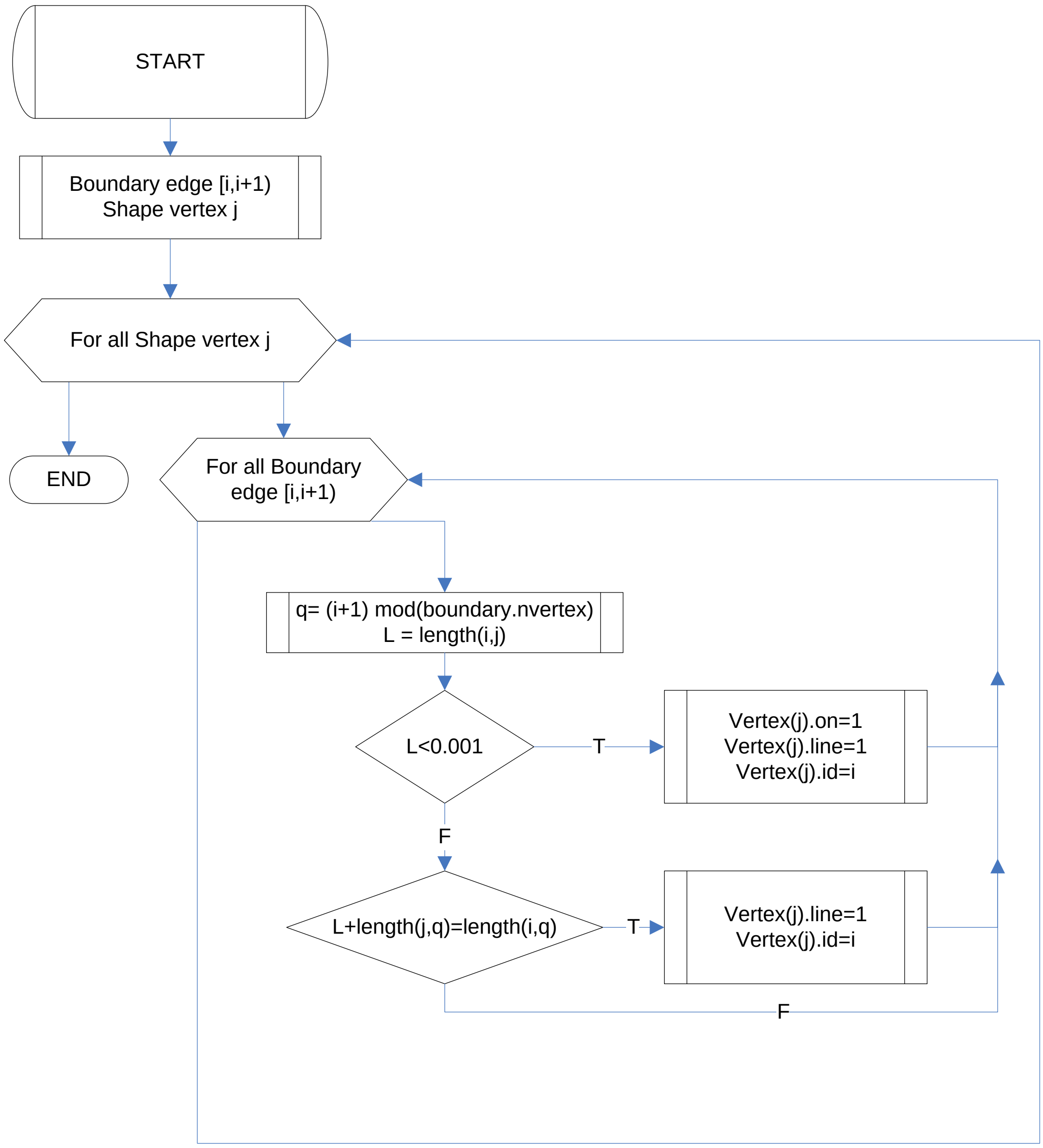


Chart 3 Flow Chart of Vertex Analysis for a Piece

```
Vertexb.on; //1 if boundary vertex is positioned on a piece vertex
Vertexb.line; //1 if boundary vertex is positioned on a piece edge
Vertexb.id; //id of piece edge that vertex of boundary belongs to
```

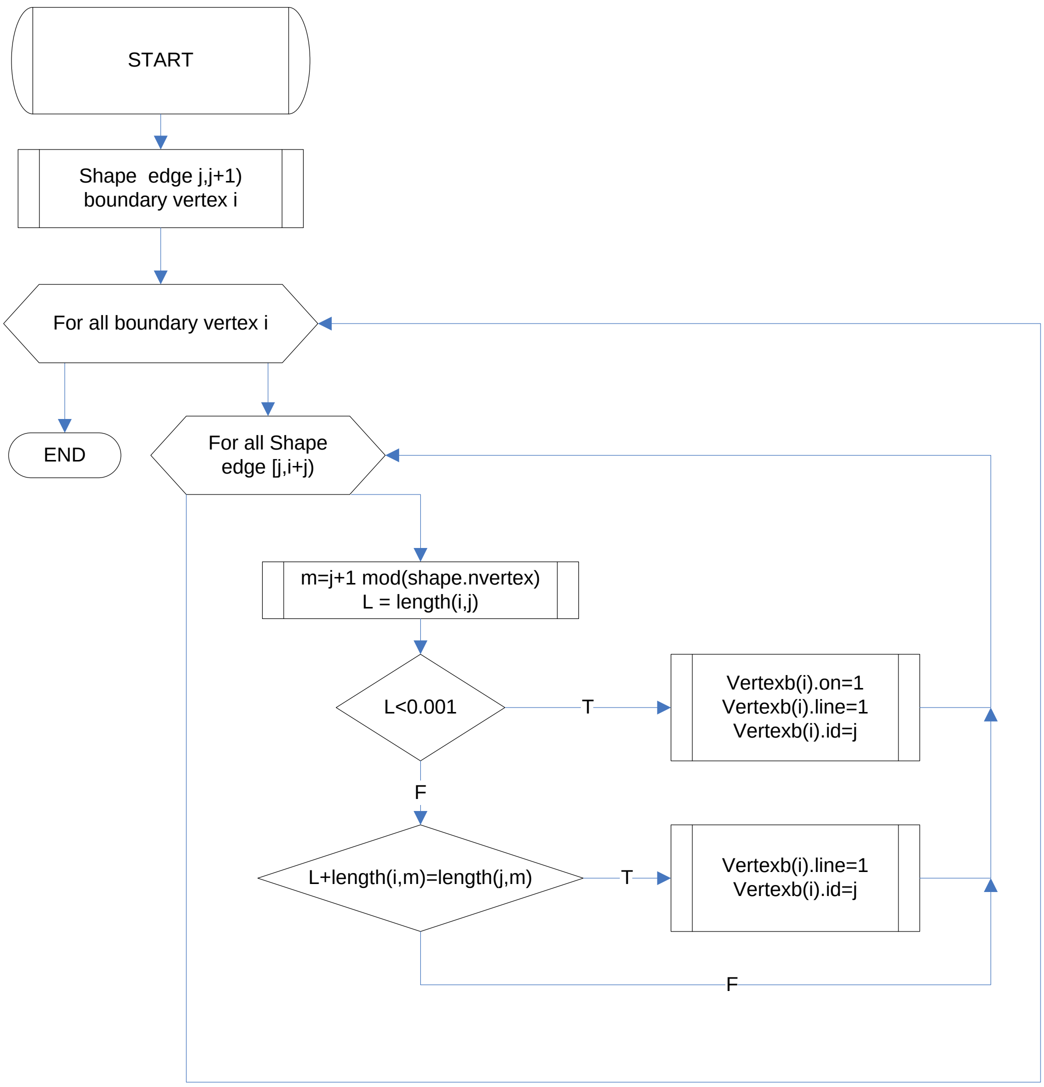


Chart 4 Flow Chart of Vertex analysis for the Boundary

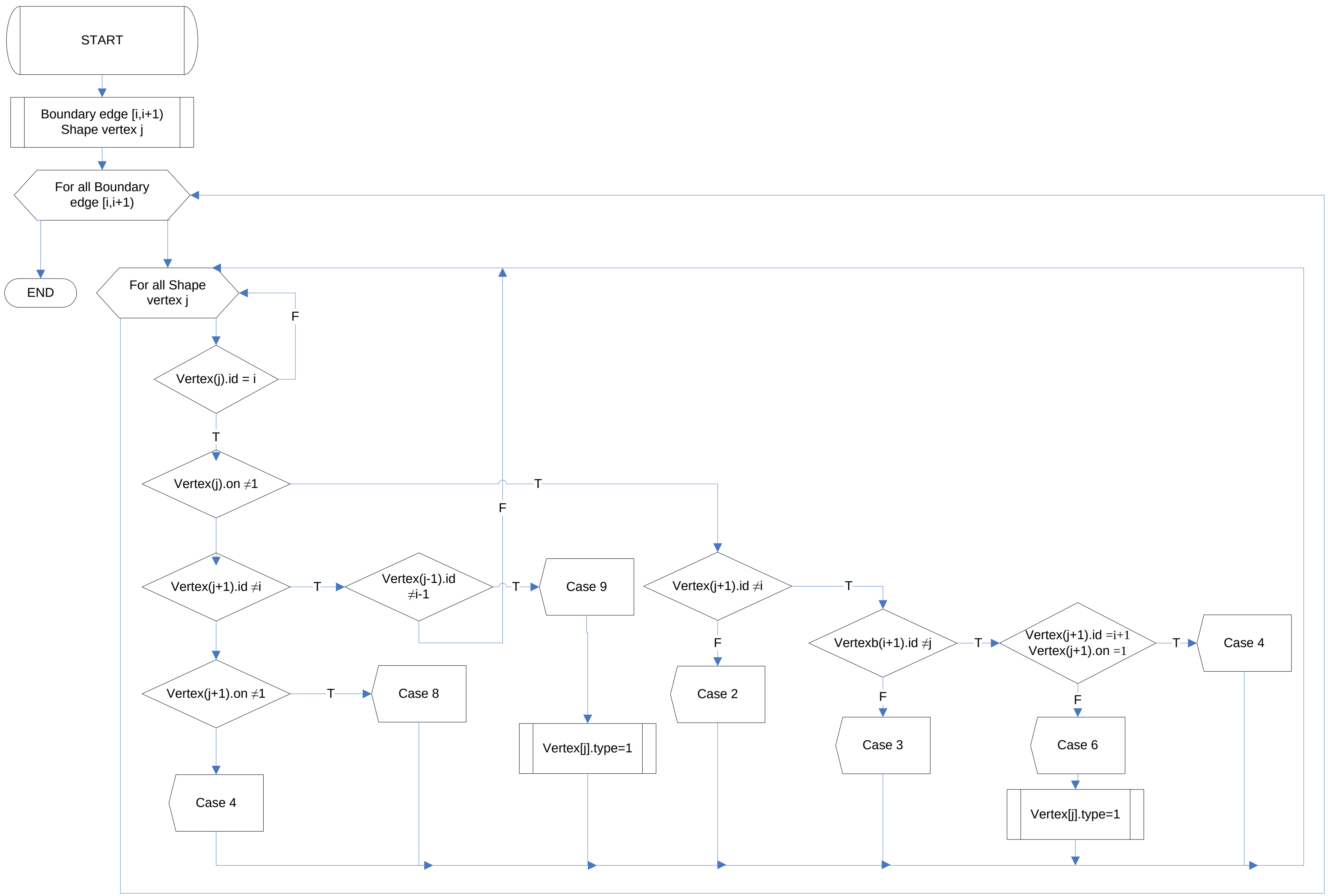


Chart 5 Flow Chart of Case analysis for Boundary

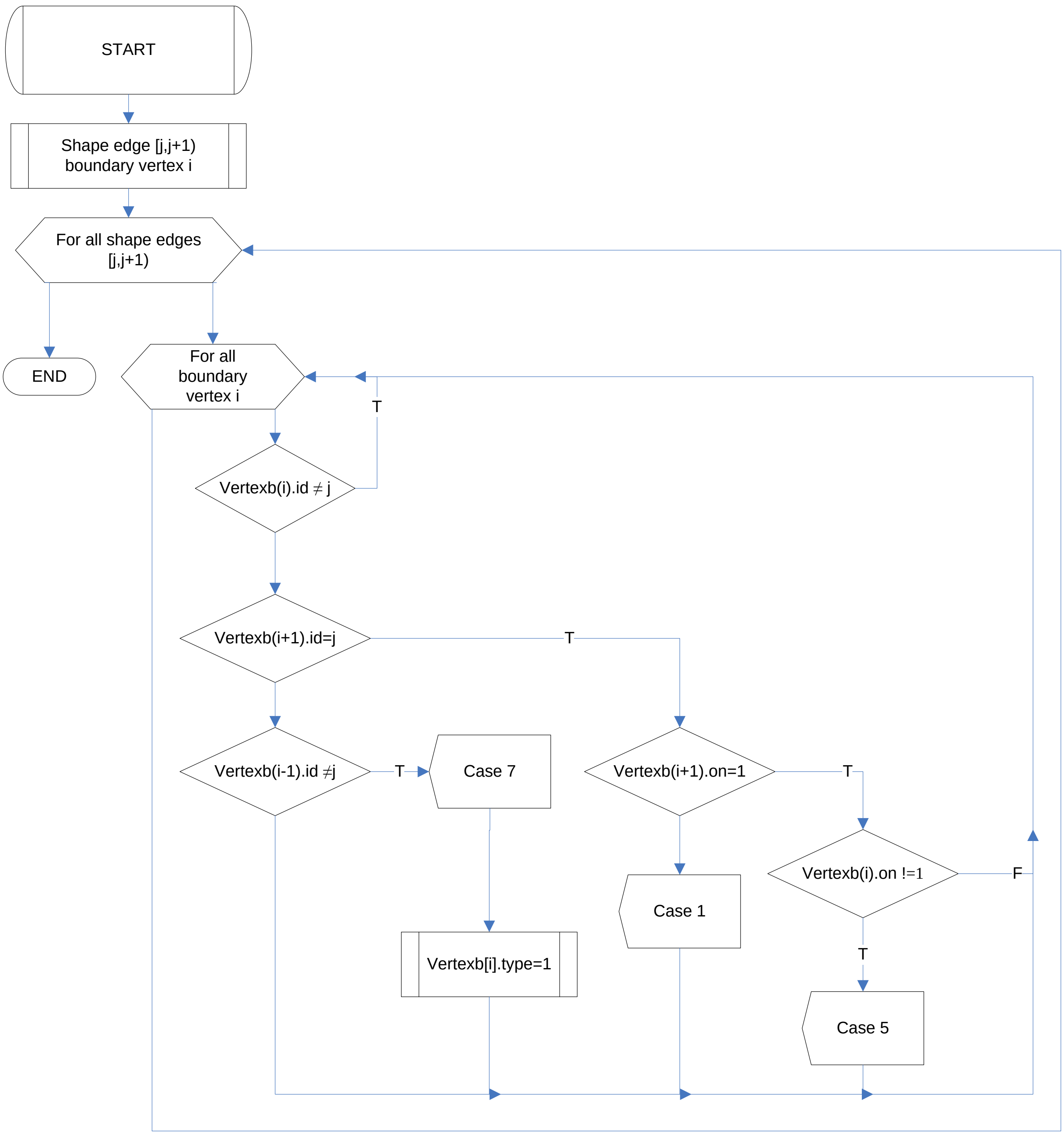
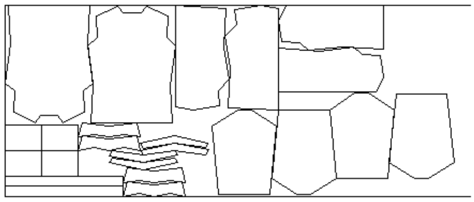
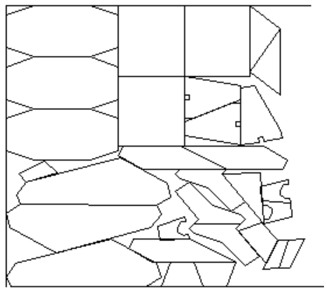
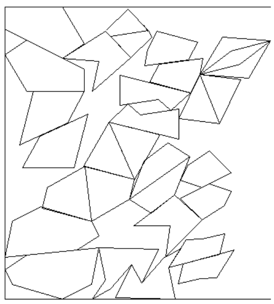
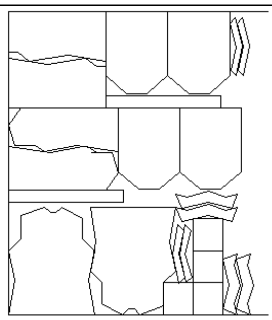
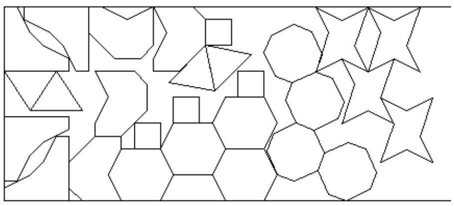
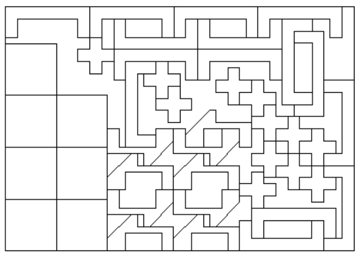


Chart 6 Flow Chart of Case analysis for a Piece

APPENDIX D – CUTTING PATTERNS FOR MERGED PIECES

 Albano, Length : 11512.98	 Dagli, Length: 63.69
 Poly 2a, Length: 36.7	 Marques, Length: 83.81
 Blaz, Length: 32.97	 Shapes 0-1, Length: 57.97

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Akbulut, Derya

Nationality: Turkish (TC)

Date and Place of Birth: 10.05.1986 / Vezirköprü

Martial Status: Single

Phone: (0312) 2812871

Email: akbulut@cankaya.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Çankaya University/ Industrial Engineering	2012
BS	Çankaya University/ Industrial Engineering	2009
High School	Samsun Anadolu High School	2004

WORK EXPERIENCE

Year	Place	Enrollment
2009-present	Çankaya University/ Dept. of Industrial Engineering	Expert
2008 July	TAI	Intern Engineering Student
2007 July	MKE Gazi Fişek Fabrikası	Intern Engineering Student

FOREIGN LANGUAGES:

Advanced English