A HEURISTIC ALGORITHM FOR NESTING PROBLEMS WITH IRREGULAR SHAPES

By

Muhammad Asif Bashir

A thesis submitted to the faculty of Electrical Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfilment of the requirements for the degree of MS in Electrical (Telecommunication) Engineering

MAY 2019

ABSTRACT

The irregular nesting problem, also known as stock placement problem has been studied and investigated over many decades where convex/non-convex pieces have to be placed inside the main sheet such that no piece overlap and cross the boundary of sheet. Nesting problem is widely used in raw material industries, like clothing, wood, metal, leather and paper making industries. The objective is to maximize the utilization of sheet and minimize the waste of sheet keeping the sheet width constant. There are many heuristic techniques in the literature used to solve the nesting problems. In nesting problem, there are two basic categories that are used by the researchers; one is placement strategy and second one is sorting strategy. Geometric constraints are the fundamental problems in nesting problems.

In this thesis, a heuristic algorithm is used to solve nesting problems which uses the concept of optimal groups of unique shapes based on placement routine which is a combination of three optimization functions, boundary overlap, convex hull and wastage. Overlap detection is performed to check the polygons overlap. Iterative method is used to generate a list of polygons placement and an objective function is measured against each placement which is again a combination of three optimization functions: boundary overlap, compaction and wastage. The polygon is placed at the placement which has maximum value of boundary overlap, maximum compaction and minimum wastage.

The proposed technique is tested with few examples of convex and non convex shapes. Our proposed algorithm's results are compared with previous benchmark algorithms already available in the literature. Our proposed algorithm's results are superior to previous works in the literature and is a strong candidate for real industries.

DEDICATION

This thesis is dedicated to

MY BELOVED FAMILY MEMBERS,

RESPECTABLE TEACHERS AND FRIENDS

for their support, love and encouragement

ACKNOWLEDGEMENTS

I am very thankful to Allah Almighty who has blessed me with the strength and the spirit to complete my thesis and giving me patience to successfully conclude my Master research thesis.

With fondness and intense appreciation I acknowledge my thesis supervisor Dr. Hasnat khurshid, PhD who not only supervised and guided me during my thesis but also encouraged my spirits to successfully and effectively complete my thesis. I am also very thankful to all my committee members including Dr. Adil Masood Siddiqui, PhD and Dr. Attiq Ahmad, PhD for their kind and moral support.

I would also like to thank to all my friends specially Mr. Muhammad Shahzad and Mr. Naveed Ahmad Chughtai for their help and cooperation to complete this thesis.

Finally, I would like to thank to my wife to encourage my spirits to successfully complete this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Nesting problem is faced in many industries where a list of polygons are placed within a sheet of fixed width and varying length without overlapping and no polygon crosses the boundary of sheet. The objective may change based on the application, however the purpose of this research is to minimize the waste and maximize the profits. Nesting problem is a type of irregular stock placement problems which are not only significant for raw material industries, such as clothing, wood, metal, leather and so on but also economically and environmentally important and reduce the use of raw materials. Economically, the solution reduces the amount of material that is required for polygon pieces and reduces the production cost. Nesting problem is a NP hard problem where we do not have an exact form of solution. We would have multiple solutions and we have to choose most appropriate solution for our problem. In most of the literatures, heuristic techniques are used to solve these kinds of problems. Section 1.1 describes the overview of Nesting Problem.

1.1 Overview

Irregular Nesting problem, also known as stock placement problem widely exists in raw material industries where large pieces are divided into small pieces and these pieces are placed inside the main sheet such that no piece overlaps and crosses the boundary of main sheet. The main goal is to maximize the sheet utilization [2]. Fig. 1.1. illustrates the basic solution to the problem [1].

Nesting problem is NP hard problem where no exact method has been proposed in literature and we do not have a close form solution and only heuristic approaches are used to solve these problems [3]. Bin packing problem is a type of nesting problem where we have only one bin and the items that have to be placed inside the bin are characterized by area or some other value and the objective is to fit maximum items

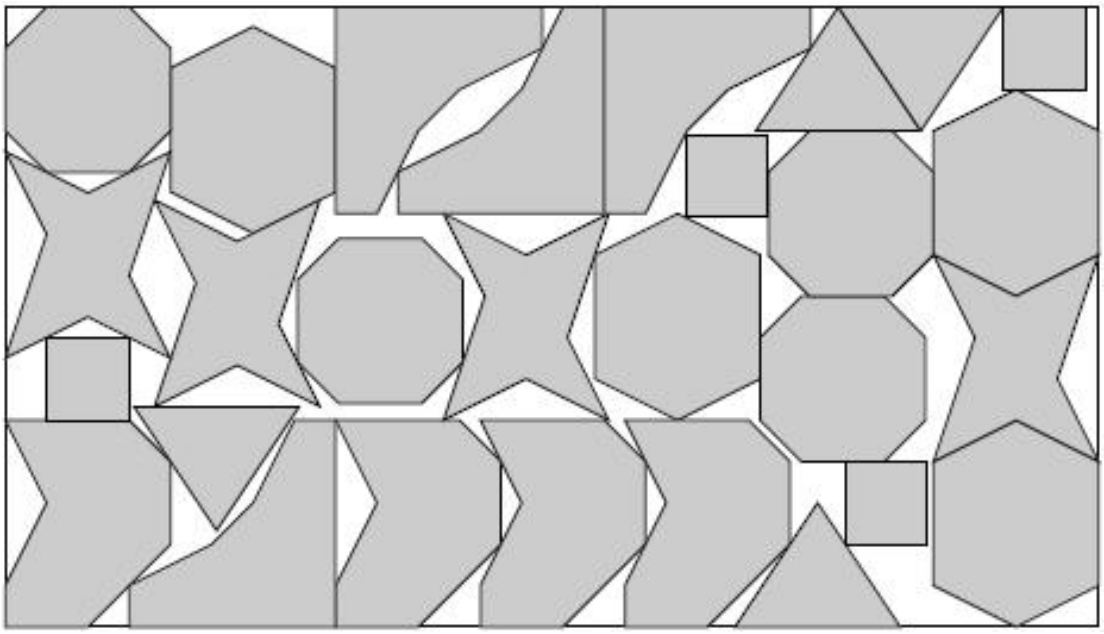


Figure 1.1: General solution to nesting problem

inside the bin that relates to our work.

Two must important strategies that are followed by researchers are placement strategy and the second one is sorting strategy. In placement strategy, researcher have main focus on minimizing the area, maximizing the length of sheet, minimizing the overlap and evaluation criteria such as measuring waste overlap. Sorting strategy deals with placement order and selection of piece to be placed next [4]. One of the main challenges in nesting problem is overlap detection method. From human prospective, overlap detection is a simple task but computationally it is really hard task and gets more complexed as the complexity of polygon increases. In literature, many methods were used for overlap detection. In most of the literatures, a method of No Fit polygon was used to detect the overlap between polygon and sheet. No Fit polygon method has advantages on traditional overlap detection methods due to its low computational time which has significant effect on the overall placement computational time.

Another heuristic technique was developed that was based on simple search where the neighborhood was horizontal or vertical translation of previous placed polygon [5]. Pairwise clustering and Guided cuckoo search techniques have also significant role in nesting problem in which polygons are used in pair based on the features that are

mostly matched in multiple polygons [6]. Fig. 1.2. illustrates the overview of nesting problem [7].

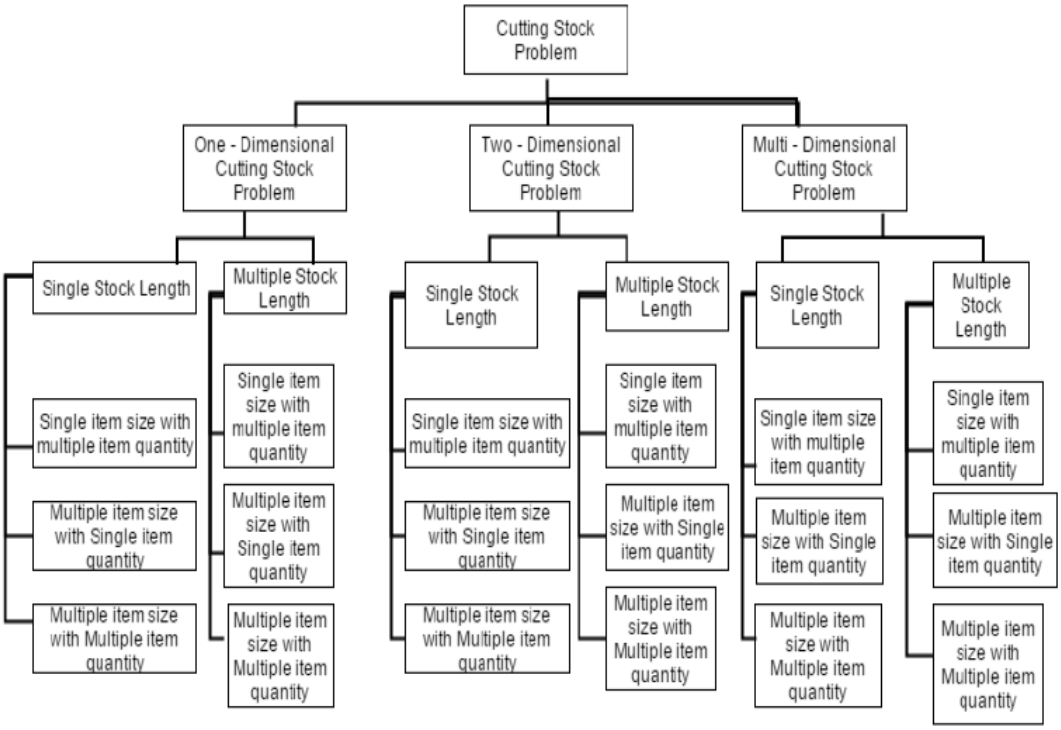


Figure 1.2: Overview of nesting/cutting problem

1.2 Problem Statement

Due to very importance of nesting problem in many industrial applications such as leather, wood, clothing, metal and papers, Heuristic algorithms are promising solutions to solve these problems. Nesting problem is a basic part of cutting and packing problems and also known as combinatorial optimization problem and deals with different type of polygons geometry i.e. regular, irregular, convex and rectangular. The problem deals with the placement of large number of polygons inside a rectangular sheet in such a way that no polygon overlaps and crosses the boundary of rectangular sheet.

1.3 Motivation and Objectives

Computational optimization problem also known as stock placement problem is faced in many industries and solution to this problem is quite important for industries which require development of good algorithms. Development of such algorithms is not an easy task. Researchers had started working on these algorithms a long time ago and

last few years algorithms deserved real investment of researcher in this field. Nesting problem has a wide range of applications and many problems are being solved with these techniques in industries. Problem has many practical variants and is very important for researchers where they do not have a good attempt to solve these problems. These problems are open for researchers and need more techniques to be developed in future. Our aim is to study and investigate all these problems and develop state of the art algorithm that could produce good results for industries in less computational time. Fig. 1.3. illustrates the structure of different problem fields (adapted from Dyckhoff 1990).

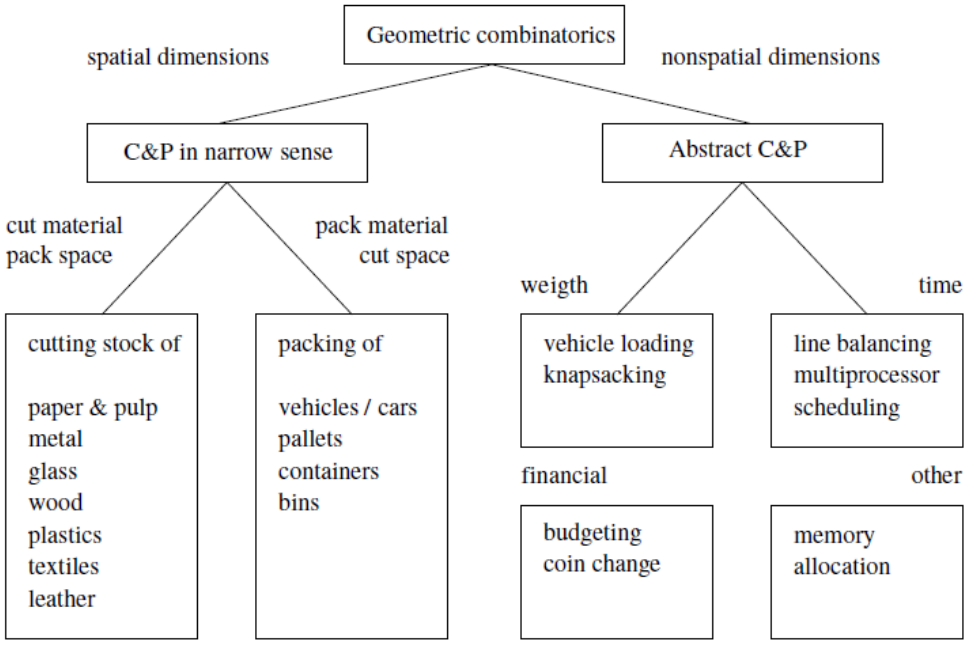


Figure 1.3: Structure of different problem fields

The objectives of this study are listed below:

- Primary objective of the research is to find most prominent solution for 2 dimensional nesting problem.

- Secondary objective of the study is to maximize the sheet utilization or minimize the wastage of sheet while placing all the polygons inside the sheet. Maximum utilization is requested in many industries to reduce the waste material and this

reduction in waste can save the production cost since material cost is major contribution to production cost.

- Tertiary objective is to improve the efficiency of algorithm in term of computational time of algorithm.

So nesting problem is quite important for many industries and in industrial environment this problem is dealt by workers who have built the nesting layout and techniques. The principle motivation behind nesting problem is to minimize the stock wastage which reduces the production cost and quite beneficial for industries.

1.4 Contributions of Thesis

This thesis makes several contributions to stock placement problem. We propose a heuristic algorithm for nesting problem based on: optimal groups of unique polygons follow by placement routine for optimal placement. Overlap detection is used to eliminate the overlap of polygons. A short description of each contribution is illustrated in this section (each contribution is discussed in detail in chapter 3).

Optimal groups of unique polygons are used to produced optimal placement results. The unique polygons are first grouped into two, three and maximum four polygons based on three functions: boundary overlap, compaction and wastage. These grouped polygons are used to place inside the sheet.

Placement routine function (max boundary overlap, min convex hull and min wastage) is used to place the polygon at optimal placement. This function extracts the most suitable placement for next piece to be placed inside the sheet.

1.5 Thesis Outline

Chapter 1 is the introduction of nesting problem which consists of overview of nesting problems, problem statement, motivation and objectives of study and contributions of this thesis to nesting problem. Chapter 2 is literature review of nesting problems. Chapter 3 is about experimental setup. Chapter 4 deals with the methodology and proposed technique used in this thesis. Chapter 5 deals with the results and discussion

which comprises of model parameters, experimental results and benchmarks problems. Chapter 6 provides the conclusion and future works.

Chapter 2

LITERATURE REVIEW

2.1 Review of Stock Placement Problems

Stock Placement problems have different forms and versions. These versions were first arranged in categories by $Dyckho$ and $Finke$. These categories were based on four criteria: dimensionality, kind of assignment, assortment of large objects, and assortment of small item. These categories are further divided into sub categories [8]. $Dyckho$ and $Finke$ scheme was further extended by $W\ddot{a}scher$ et al. (2007) and fifth criteria "shape of small items" was added [9].

The dimensionality deals with the number of dimensions of the problems which can have one dimension, two dimensions, three dimensions and N dimensions. Kind of assignment consists of output value maximization and input value minimization. Small pieces can be assorted in three categories: strongly heterogeneous (many pieces of different types), weakly heterogeneous (many pieces of few shapes) and identical (single type pieces). Large pieces can be assorted in two main categories: single large piece with fixed or variable dimensions and several large pieces with fixed dimensions. In case of more than one dimensions, the items can be divided into categories of regular (rectangles, boxes, circles) and irregular.

According to the typology proposed by $W\ddot{a}scher$ et al. (2007), two dimensional irregular nesting problem can be divided into six main categories: (i) Identical item packing problem, (ii) Placement problem, (iii) Knapsack problem, (iv) Cutting stock problem, (v) Bin packing problem, (vi) Open dimension problem. Fig. 2.1. illustrates the nesting problem types which was adapted from $W\ddot{a}scher$ et al.(2007 [9].

This chapter discussed the basics of nesting problem. Section 2.1 describes the geometric representation of pieces and how this geometry can be used to solve the geometric limitations of the problem.

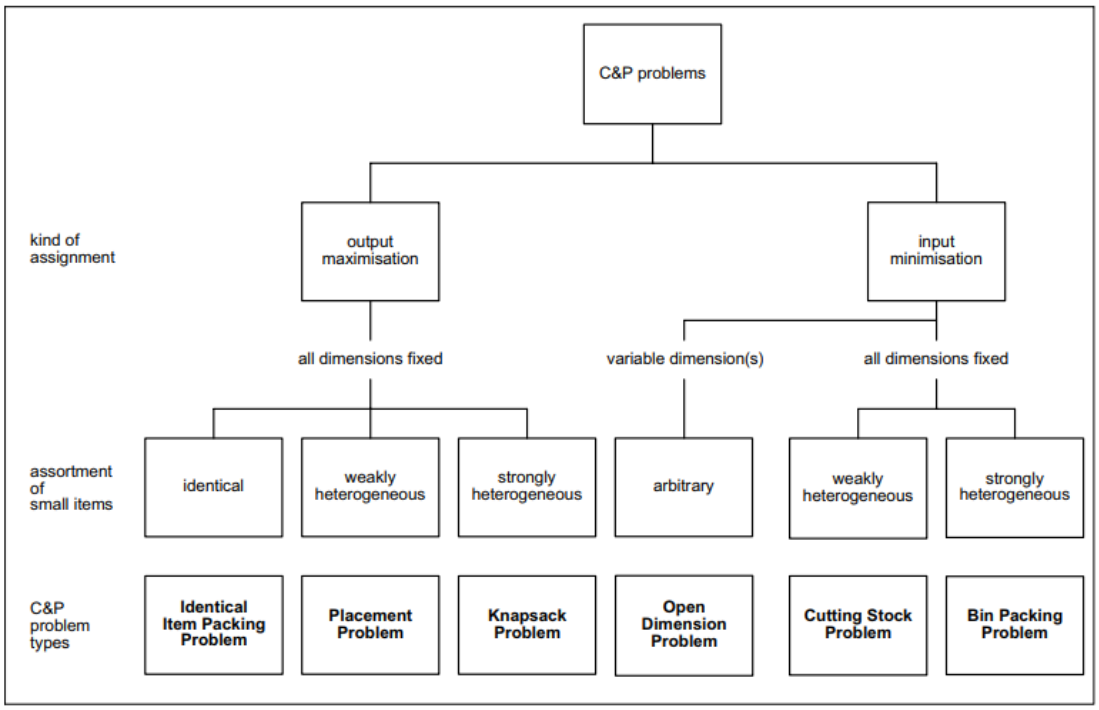


Figure 2.1: Nesting problem types

2.2 Nesting Problem Types

The cutting and packing problem has a wide range of problem types which are listed below:

2.2.1 Bin Packing

Bin packing problem deals with maximum number of items that could be placed inside the bins. The goal is to use less number of bins that contain maximum number of items. Bin packing problem has different variants and items and bins could be single dimension or multidimensional. Problem has a wide range of industrial applications such as manufacturing, vehicle scheduling, loading and vehicle routing. Garey, Coffman and Johnson had done a detailed survey of bin packing problem [10].

2.2.2 Knapsack Packing

Knapsack is an important type of stock placement problems and very similar to bin packing problem. The problem consists of fixed capacity knapsack and different types of objects. The main goal is to put maximum number of objects inside the knapsack by keeping in mind the maximum capacity of knapsack [11].

2.2.3 Capacity Allocation

Capacity allocation problem is also called Space allocation problem and closely related with bin packing and knapsack problems. The objective is to allocate space to all objects by keeping in mind all constraints. The problem can be explained with a real life example. Let say, we have a small office with fixed capacity/space and 10 office people. Now the objective is to assign space to all people of this office. More detail of this problem can be viewed in [12, 13].

2.2.4 Strip Packing

The strip packing problem deals with the packing of rectangles where only 90^0 rotation is allowed. The sheet should be of limited height. Unlimited height sheets consist of roll of material.

2.2.5 Trim Loss Problem

Trim packing problem is also called sheet wastage problem which occurred as result of cutting irregular parts from sheet. The problem is quite important and complicated for industries when these irregular parts are planned to be used in future packing problems.

2.2.6 Loading Problem

Loading problem is a two or three dimensional regular packing problem where regular boxes are placed within a container. It contains many industrial objectives and constraints. The problem can be explained more with an example where a delivery lorry is loaded to deliver customer requested objects.

2.2.7 Nesting Problem

Nesting problem is very important type of stock placement problems and gains interest of researchers. Due to its wide range of industrial applications, research area is open for researchers and it needs more work has to be done in this area. Nesting problem mostly represent two dimensional problems where irregular shapes has to be placed inside rectangular or non-rectangular sheets. The main focus of our research is on nesting problem.

2.3 Stock Placement Problem Surveys and Bibliographies

Facts of stock placement problem can be used to in different areas. As mentioned by Dyckhoff, the research area is opened for different applications and research is conducted in following areas: manufacturing, mathematics, optimization, industries, engineering science, operation research etc. Huge library of surveys and bibliographies has been conducted so far on stock placement problems. Table 2.1 illustrates the list of surveys and bibliographies derived from [14, 15, 16].

Author(s)	**Year**		**Author(s)**	**Year**
Brown [17]	1971		Rode and Rosenberg [32]	1987
Salkin and de Kluyver [18]	1975		Dyckhoff, Finke and Kruse [33]	1988
Golden [19]	1976		Coffman and Shore [34]	1990
Hinxman [20]	1980		Dyckhoff and Wacher [35]	1990
Garey and Johnson [21]	1981		Dowsland [36]	1991
Israni and Sanders [22]	1982		Haessler and Sweeney [37]	1991
Sarin [23]	1983		Dowsland [38]	1992
Rayward-Smith and Shing [24]	1983		Dyckhoff and Finke [39]	1992
Coffman, Garey and Johnson [25]	1984		Haessler [40]	1992
Berkey and Wang [26]	1985		Lirov [41]	1992
Dowsland [27]	1985		Ram [42]	1992
Dyskhoff, Kruse, Abel and Gal [28]	1985		Lodi, Martello and Vigo [43]	2002
Israni and Sanders [29]	1985		Cagan, Shimada and Yin [44]	2002
Dudzinski and Walukiewicz [30]	1987		Lodi, Martello and Monaci [45]	2003
Martello and Toth [31]	1987		Oliveria [46]	2003

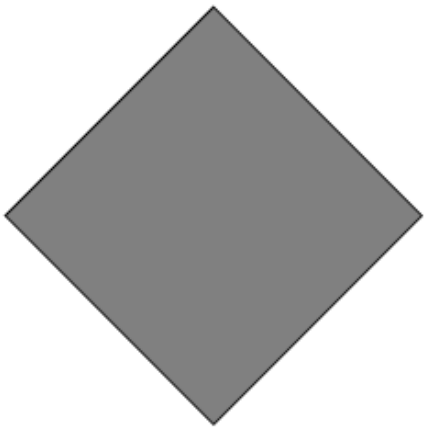
Table 2.1: Stock placement problem surveys

2.4 Geometric Representation

Nesting problem has two important constraints: (i) polygon pieces are placed inside the sheet with no overlap (ii) no polygon cross the boundary of sheet. Geometry of pieces has a direct relation with the solution of nesting problem and to deal with these constraints. Bennell and Oliveira (2009) had done a detailed survey of geometric problems. In literature, geometric problems are represented by different techniques and approaches. The most common geometric methods are pixel/ raster method, D function, No fil polygon and phi- function [47].

2.4.1 Pixel/Raster Method

Polygon pieces and sheet can be represented by pixel matrices. The continuous data sheet and polygons are divided into discrete values and polygons and sheet grids are represented by a set of pixels. Different authors used different techniques to represent the geometry of pieces and sheet. Oliveira and Ferreira (1993) proposed a very simple scheme in which pixel value 1 denoted the existence of piece and pixel value 0 denoted empty space [48]. Fig. 2.2. illustrates polygons matric representation based on the Raster method.



0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0

Figure 2.2: Polygons matric representation based on Raster method

Maximum utilization can be achieved by placing the polygons close to each other and no polygon cross the boundary of sheet. Polygons can also be represented by number of vertices and a reference point is used to control the placement of piece on sheet.

Inner no fit polygon method is used to make sure that no polygon cross the boundary of rectangular sheet. Let say we have a polygon A which has to be placed inside the sheet. A reference point is selected and the reference point could be any vertices of the polygon. The $INFP_A$ can be defined as one or several closed polygons formed by moving reference point of polygon A along the internal side of polygon, by keeping sheet stationary and A reference point always touches the sheet without overlap. The inner no fit polygon of rectangular sheet is also a rectangle. In most of the literature, INFP is used with NFP to extract the feasible position points. Fig. 2.3. illustrates the INFP of P relative to the rectangular sheet [49].

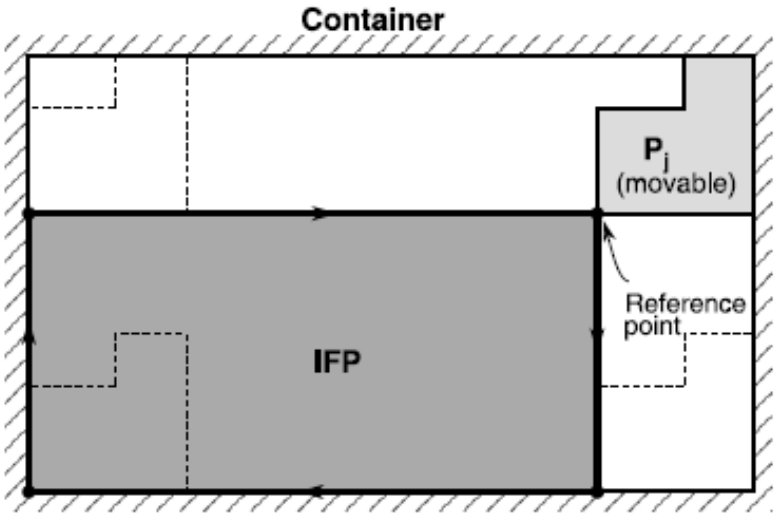


Figure 2.3: INFP of P relative to the rectangular sheet

2.4.2 D-Function

Raster method has more accurate results with regular shapes. Instead of using discrete values, direct polygon can be used for irregular shapes. A polygon can be represented by number of vertexes and edges. Fig. 2.4.illustrates the vertex and edge of a polygon [50].

In computer language, measuring the overlap between polygon pieces is very complex task and takes much computational time. One of the techniques used in the literature to detect overlap between polygons is D-Function. A line is oriented in any

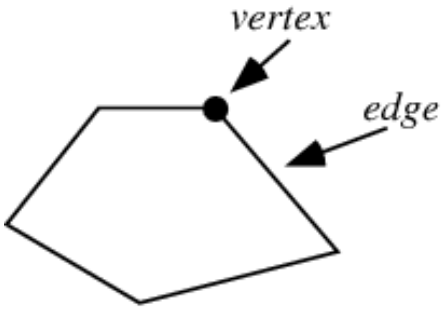


Figure 2.4: Polygon vertex and edge

direction and D-function determines whether a point is on left side or right side of this line. The overlap can be detected by using the information of vertices and edges that make a polygon. Fig. 2.5. illustrates the basic concept of D-function.

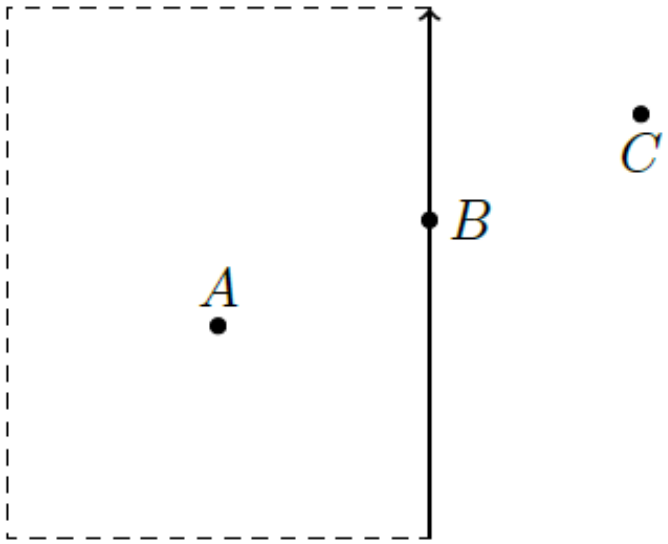


Figure 2.5: D-Function method

2.4.3 No-Fit Polygon (NFP)

The most important technique to detect the overlap between pieces is no fit polygon. NFP polygon technique has importance due to its less computational time. The first use of no fit polygon technique was done by Art (1966) and in his research Art used the term shape envelop instead of no fit polygon [51]. Later in 1976 Adamowicz and Albano replaced the term shape envelop with no fit polygon and the technique was used in stock placement problem to detect the overlap between pieces using minimum

rectangular enclosure [52]. The main function of NFP is to measure the places where two polygons intersect with each other.

NFP can be made by moving a polygon over the perimeter of stationary polygon. Given two polygons A and B where A is stationary and B is moving polygon. The first step in NFP is to mark a reference point on moving polygon B which will be identified when B moves over the outer boundary of polygon A. The reference point could be any vertex of moving polygon B. Start moving polygon B around the outer boundary of stationary polygon such that polygon always touch and do not overlap. As polygon B traced over the stationary polygon A the output polygon is called NFP_{AB}. The resulting NFP_{AB} and reference point of B is used to test whether Polygon A and B overlap or not. In case reference point of polygon B is inside the NFP_{AB} then polygon A and B overlaps. In case the reference point of polygon B is on the boundary of NFP_{AB} then polygon A and B touches. And finally if the reference point of polygon B is outside the NFP_{AB} then polygon A and B do not overlaps. Fig. 2.6. illustrates the function of NFP. Fig. 2.7. illustrates the intersection test of polygon B with polygon A using NFP [53].

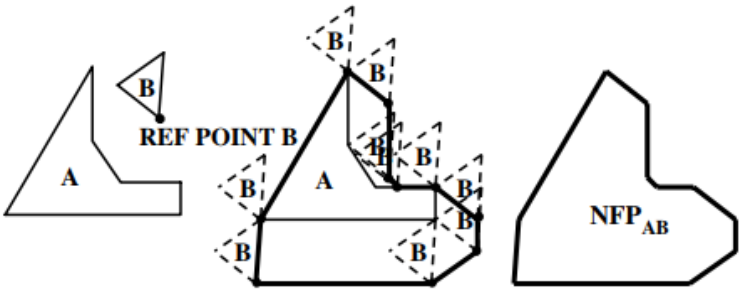


Figure 2.6: Working principle of NFP

2.4.4 Phi-Function

Phi function is a latest technique that deals the geometric issues in nesting problem. Phi function is a linear or non-linear function which is used in stock placement problem to represent the mutual position between pieces. The Phi function was first invented in cutting and packing problem by Stoyan et al (2001) in which he purposed that the value of Phi function should be less than zero when two pieces overlap, equal to zero

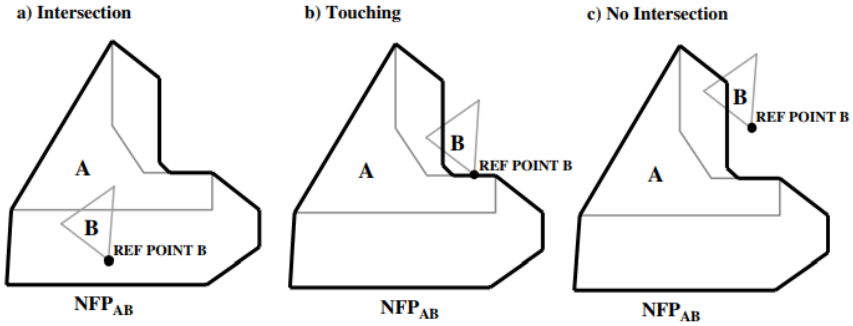


Figure 2.7: Intersection test between polygon A and B using NFP

when their boundaries touch and greater than zero when two pieces do not overlap [54] . Fig. 2.8. illustrates the polygon and its Phi function representation. In figure left side is polygon and right side is phi function representation of polygon.

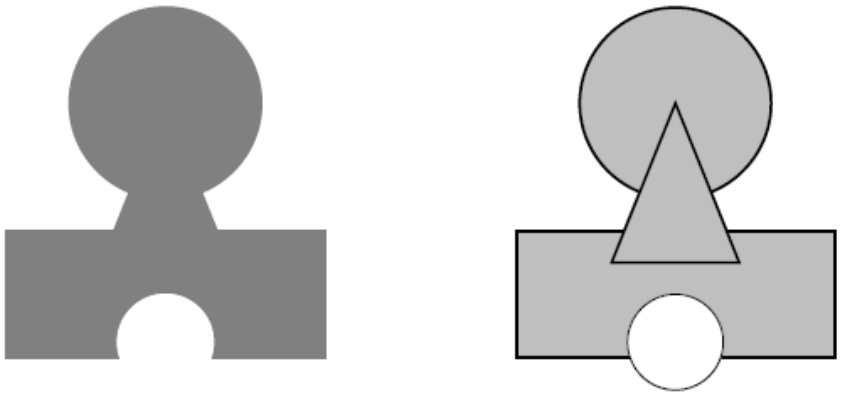


Figure 2.8: Polygon and Phi-function representation of polygon

2.5 Search Methods

This section discusses some search methods that are used in stock placement problems. A brief description of searching techniques and their examples is given in this section.

2.5.1 Iterative Improvement Search

The first search technique is iterative improvement search where the next iteration to be evaluated belongs to the neighborhood of current iteration. This search requires most suitable neighborhood which can fit in next iteration. In rectangular packing problems, the rectangles can be placed on the sheet in a given sequence. The number of these sequences can be used to generate a suitable neighborhood solution and the sequence

number can be swapped. Most of the search methods need an initial solution. This initial solution is based on the quick heuristics. The sequences are first sorted based on height or area of the piece. In our case, the pieces are sorted based on descending order of polygon areas. The large size polygons can be placed first and small size polygons can be placed inside the holes and at small places and it produces a good quality solution.

2.5.2 Hill Climbing

Hill climbing is a very simple optimization research, where a neighborhood of current solution is generated and then this neighbor is evaluated. If the neighborhood solution has a good quality than the current solution then current solution is replaced with neighborhood solution and process continues. If the neighborhood solution has bad quality than the current solution then neighborhood is discarded and original solution is adapted. At the end of the search, best solution is finalized and this should be the current solution of hill climbing.

2.5.3 Tabu Search

Tabu search is a next extension of Hill climbing search technique and was first proposed by Glover [55]. Tabu search is slightly different from Hill climbing and unlike hill climbing, Tabu search evaluates a group of neighborhoods at each iteration and choose the solution which has best evaluation value. Current solution is always replaced with new solution which has better evaluation value. The history of last solution is also maintained in a list. This list is used to avoid the revisiting of last areas which has already been covered in previous searches.

2.5.4 Random Search Algorithm

Random search algorithm is a very common algorithm that is used in literature. In random search algorithm, the polygons/pieces are saved in a list and each polygon is represented by an index number. Random permutation of index is created in each iteration and this permutation represents the order of pieces. First piece is selected from the list of random permutation. Then second piece is selected from the remaining

list of random permutation. The NFP of these two pieces is calculated which gives a list of points where second polygon can be placed. Second polygon is placed at each point of NFP and bounding box area is calculated. Bounding box is the minimum area which can cover the whole piece. Second polygon is placed at point which has minimum bounding box area. Now the convex hull of first and second polygon is calculated and this convex hull should be used as static polygon for next iteration [56]. Details about convex hull can be found in next section.

2.6 Heuristic and Metaheuristic Methods

In literature, many heuristic and meta heuristic methods are used for solving irregular stock placement problems. In heuristic methods, we dont have an exact solution to the problem, instead we have multiple solutions available in list and we choose one more suitable solution to the problem. Heuristic methods are optimization methods which have less computational time than exact approaches. Bennell and Oliveira presented the first review of heuristic methods in 2009. There are different heuristic methods used in the literature. Some methods are constructive heuristics and some are improvement heuristics.

Next section describes some heuristic methods which are used in the literature.

2.6.1 Bottom Left Algorithm

Bottom left heuristic algorithm was first presented by Art in 1966. In bottom left algorithm, heuristic is applied in each step and one piece is placed inside the sheet. To place a piece inside the sheet, piece is first placed at top right side of the sheet (Figure 2.9 (a)) then it is moved horizontally to the left side until it cant be moved left further (Figure 2.9 (b)). Then, the piece is moved vertically down until it can be moved to left side again (Figure 2.9 (c)). This process continues until the piece can be moved further left or down (Figure 2.9 (d)). (Figure 2.9 (e)) shows the place where piece 5 is placed inside the sheet using bottom left heuristic [57].

The bottom left heuristic depends upon the order and sequence of pieces which needs more investigation. For sequences of pieces, different investigations by authors can be

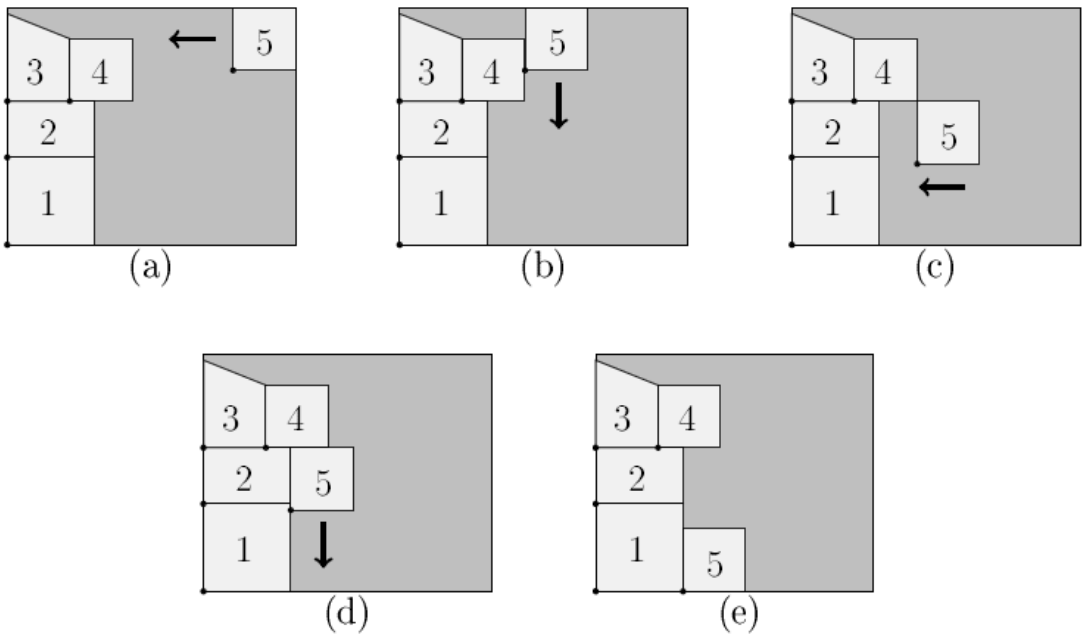


Figure 2.9: Bottom-left process

found in literature. In [58], Oliveira describes some rules to arrange the sequence of pieces. More than 120 variants of bottom left heuristic were derived from these rules. In [59], Dowsland proposed nine rules to arrange the pieces in sequence and then these rules are used in efficient implementation of bottom left heuristic. In [60], Gomas and Oliveira introduced the search over the sequence of pieces. In each iteration, the position of two shapes in the sequence was alternated and then bottom left heuristic was performed.

2.6.2 Simulated Annealing

Simulated annealing is the most efficient algorithm for search over the sequence. The concept of SA was first developed by Metropolis et al. [61] in which he presented an algorithm to measure the cooling of heating bath. Kirkpatric et al. [62] first time proposed the simulated annealing to solve the combinatorial optimization problems. After that many researcher had started working on simulated annealing and there has been many applications where simulated annealing has been applied to different optimization problems. Simulated annealing is an advance version of local optimization where initial solution is evaluated and then some iterations are performed to improve this

initial solution until no such iteration produced better results. Simulated annealing is random search procedure in which the probability of getting stuck at poor iteration is very low. Because it avoids to get stuck at poor local optima, it produces better results.

2.6.2.1 SA Algorithm Example

To understand the simulated annealing algorithm, consider a simple example derived from [63].Let $P = \{c_1, c_2, c_3, ...c_n\}$ is a tour among n countries, where $c_i \in \{1, 2, 3, ...n\}$. Let F(P) denotes the function of tour P, then function can be written as:

$$F(P) = \sum_{i=1}^{n-1} d_{c_i, c_{i+1}} + d_{c_n, c_1} \tag{2.1}$$

where d is distance between two countries. The objective of simulated annealing is to minimize the equation (2.1) by choosing random order of closed tour.

The detail procedure of simulated annealing algorithm is given as follows:

Step 1: Initialization : Setup initial temperature T_{ini-1} and end temperature T_{end-1} and cooling coefficient α

Step 2: Initial solution : Randomly select the a country x from the closed tour and set it as the first country $c_1^o (c_1^o = x)$ of the tour. Now mark the most closed city of c_1^o as the second city c_2^o of the tour and do the same process until the last city c_n^0 of tour is selected. Finally, the initial solution of tour is created $P = \{c_1^o, c_2^o, c_3^o, ...c_n^o\}$. Now, measure the objective value $V_0 = F(P_0)$ according to the equation (2.1).

Step 3 : Create a new solution : Randomly select three solution $x_1, x_2 and x_3$ from the closed tour where $x_1, x_2, x_3 \in= \{1, 2, 3, ..., n\}$.Let

$$d_{c_{coun1}, c_{coun1 mod(n)+1}} = Max\{d_{c_{x1}, c_{x1 mod(n)+1}}, d_{c_{x2}, c_{x3 mod(n)+1}}, d_{c_{31}, c_{31 mod(n)+1}}\} \tag{2.2}$$

Now randomly select three positive integers $y_1, y_2 and y_3$ from the closed tour where $y_1, y_2, y_3 \in= \{1, 2, 3, ..., n\}$.Let

$$d_{c_{coun1}, c_{coun2}} = Min\{d_{c_{coun1}, c_{y_1}}, d_{c_{coun2}, c_{y_2}}, d_{c_{coun2}, c_{y_2}}\} \tag{2.3}$$

Where coun2$\in \{1, 2, 3, ..., n\}$, coun2 $\neq$coun1, coun2 $\neq$ (coun1-2+n) mod (n)+1,

coun2 $\neq$ coun1 mod (n)+1. If coun 1 $<$ coun2, then

$$\acute{c}_i = \begin{cases} c_i & 1 \leq i \leq coun1 \\ count2 & i = coun1 + 1 \\ c_{i-1} & coun1 + 1 < i \leq coun2 \\ c_i & coun2 < i \leq n \end{cases}$$

otherwise

$$\acute{c}_i = \begin{cases} c_i & 1 \leq i \leq coun2 \\ c_{i+1} & coun2 \leq i < coun1 \\ c_{coun2} & i = coun1 \\ c_i & coun1 < i \leq n \end{cases}$$

Step 4: Get new objective value: Calculate the new objective value $F(\acute{P})$. If $F(\acute{P}) < V_0$ then $V_0 = F(\acute{P}) and P_0 = \acute{P}$

Step 5: Selection of new solution: Let ΔF is the relative difference of two objective values, then $\Delta F = F(\acute{P}) - F(P_0)$, if $\Delta F \leq 0, P_0 = \acute{P}$ then $\acute{P}$ is accepted. Otherwise if $\Delta F > 0$, then calculate the probability to accept the new solution $\acute{P}$ by using this equation:

$$\rho = e^{-\Delta F/t} \tag{2.4}$$

And $\acute{P}$ is only accepted this probability ρ.

Step 6: Continuous or end of SA: Check out the current temperature T, if T $< T_{e-1}$ then SA will end otherwise go back to step 3.

2.6.3 Genetic Algorithms

Genetic algorithms are based on the concept of evolution of population within the world. First time genetic algorithms were proposed by Fraser in 1957 [64]and Bremermann in 1958 [65]. Genetic algorithms consist of several solutions known as chromosomes/individuals. Each chromosome consists of several parts known as genes. A value is assigned to each gene and that value is called alleles. It's important to measure the quality of each chromosome called fitness as these chromosomes represent a solution. The flow chart diagram of genetic algorithm is illustrated in Fig.2.10. [66].

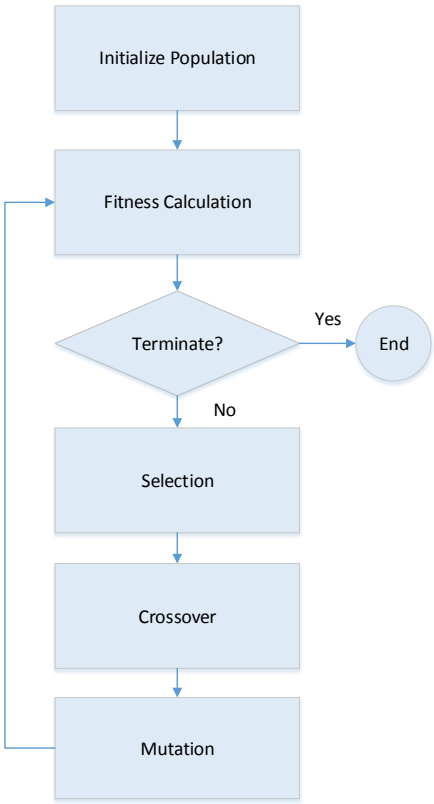


Figure 2.10: Flow chart diagram of Genetic Algorithm

2.6.3.1 Initialize Population

Genetic algorithms initialize with population; a set of individuals. Each individual is actually a solution of the problem which we want to solve. These individuals are characterized by set of variables known as genes. A joint combination of genes is called a chromosome. In GA, the genes are represented by a set of binary digits (a string of 1s and 0s) then we encode and decode these genes in chromosome as shown in Fig.2.11.

2.6.3.2 Fitness Function

Fitness function is a function that compares all the individuals and determines how fit an individual is among all individuals. It assigns a fitness score to individuals which calculates the probability for each individual that can be used for the reproduction.

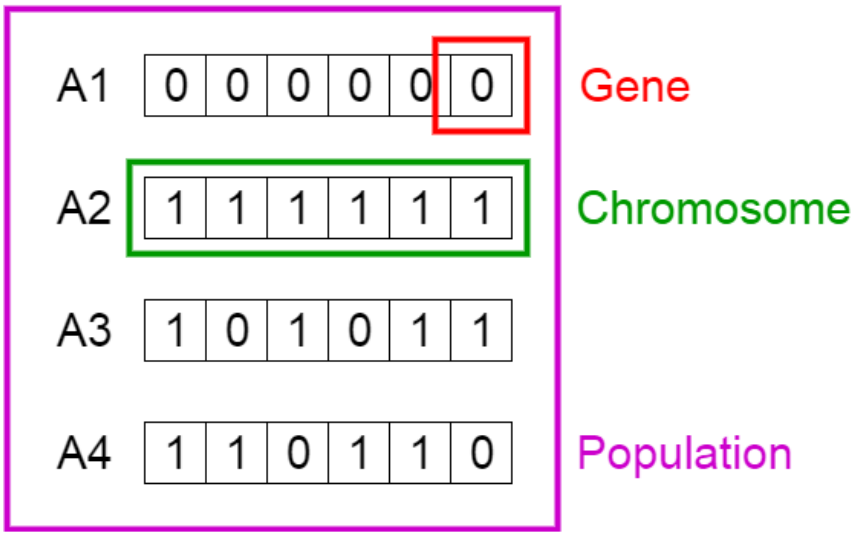


Figure 2.11: Chromosomes, Genes and Population of GA

2.6.3.3 Selection

In the selection phase, those individuals are selected which has maximum value of fitness score and these individuals are used for next generation. Two pairs of individuals are selected based on their fitness function, these pairs are known as parents.

2.6.3.4 Crossover

Crossover is the most fundamental part of generic algorithm. Crossover point is selected randomly from the genes for each parent's pair. Consider an example in which crossover point is selected 3 as shown in Fig.2.12.

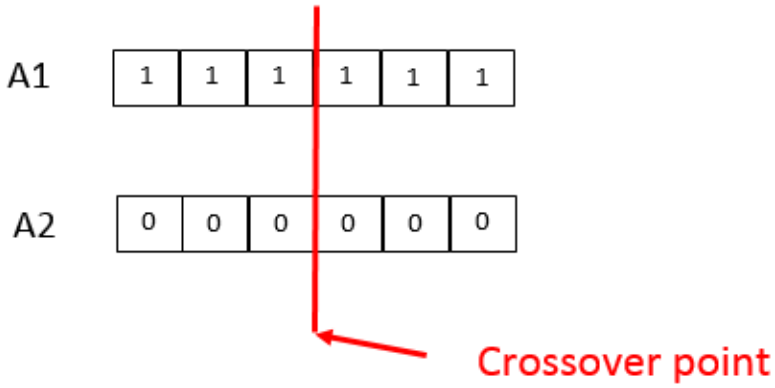


Figure 2.12: Crossover in GA

Now offspring are created by exchanging all the binary values of genes of parents

until crossover point is reached as shown in Fig.2.13 and Fig.2.14. And then these offspring are added to the population.

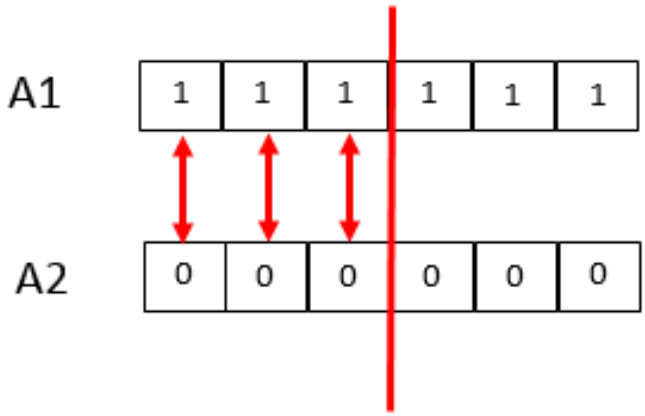


Figure 2.13: Exchanging genes of parents

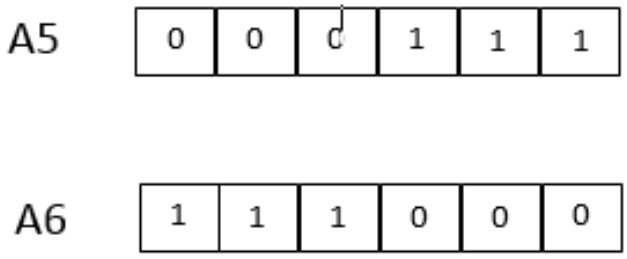


Figure 2.14: New offspring

2.6.3.5 Mutation

Once the offspring formed, some of the genes of parents can be selected for mutation with low random probability. It means that some of the bites can be flipped as shown in Fig.2.15. Mutation prevents the premature concurrence.

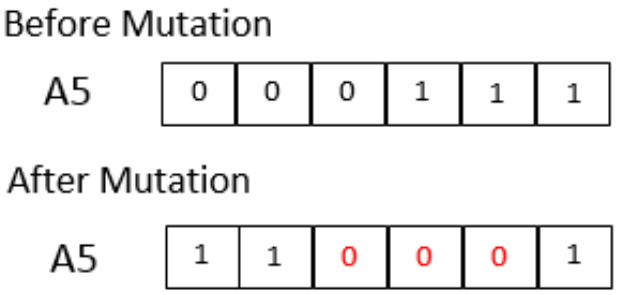


Figure 2.15: Mutation

2.6.3.6 Termination

If the offspring of new generation is not different from the previous one then population has converged and algorithm terminates. Finally, these are the solutions to our problem.

2.6.4 Linear Programming

Linear programming is a latest approach to solve stock cutting problems. In simplex Linear programming method, we look for a new column or activity to improve our solution. So instead of looking at all columns to pick out best one, we insert a new useful column to solve the problem. For stock placement problems, consider a simple example in which an order is placed for N_i number of pieces of length l_i of the material, for $i = 1, 2, 3, ..., m$. We have a stock of standard lengths $L_1, L_2, L_3, ..., L_k$ from which lengths have to be cut off to fill the customer order. Consider large number of pieces are available in stock for each length $L_1, L_2, L_3, ..., L_k$. For some value j and i, as long as $L_j \geq l_i$, an order can be filled. Each length in stock has a predefined cost. The cost to fill an order is the total cost of a material to be cut off the stock. Here the problem is fill the stock order at least cost.

Here the activity means cutting a specific stock length from available stock. For example, the customer placed an order of cutting three pieces one of length 5 and two of length 4 from a stock of length 17, this is an activity. Now assign a value to each activity that ordered the cutting of length $l_1, l_2, ..., l_m$ from a stock of length $L_1, L_2, L_3, ..., L_k$, so this cutting problem is a linear programming problem, where the

each value of a variable indicated the number of times the activity is performed. The variables assigned to activities are $v_1, v_2, v_3, ..., v_n$ and these variable should fulfill the m inequalities. Thus,

$$a_1x_1 + a_2x_2 + a_3x_3 + ... + a_nx_n \geq N_i, (i = 1, 2, 3, ..., m) \tag{2.5}$$

Let say, an order of N_i pieces of length l_i to be satisfied, then the cost function to be minimized is given as [67];

$$C_1x_1 + C_2x_2 + C_3x_3 + ... + C_nx_n \tag{2.6}$$

2.6.5 Cuckoo Search Algorithm

Cuckoo search was first time proposed by Yang and Deb in 2009 [68]. The algorithm was sparked by the reproductive concept of cuckoos in which the female lays her fertilized eggs in the nests of other species so that surrogate parents unintentionally raise her brood. Three basic rules of cuckoo search are shown in Fig.2.16.

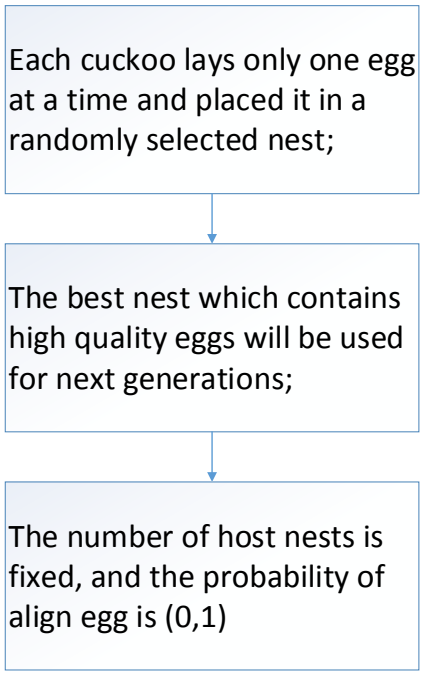


Figure 2.16: Basic rules of Cuckoo search algorithm

Following steps describes the cuckoo search algorithm:

- Start with initial population of n host nests.

- Lay the egg $(a\acute{k}, b\acute{k}$ in the k nest where k is a randomly selected nest. Cuckoo egg is quite similar to host egg.

- Compare the fitness score of cuckoo egg with the fitness value of host egg and calculate the root main square error.

- If the fitness score of cuckoo egg is better than the fitness score of host egg then replace the kth nest egg by cuckoo's egg.

- If the host nest notice it, the nest is discarded and a new nest is built.

- Repeat 2nd step to step 5 until termination criteria is satisfied.

  Flow chart diagram of cuckoo search algorithm is depicted in Fig.2.17 [69].

2.6.6 Pairwise Clustering

Piarwise clustering is also an important technique used in nesting problems. Pairwise clustering is based on the concept of matched features. Making the pair of these matched features in a case where large number of pieces have to be cut out of a stock can yield good results [70]. The polygons first segregated into pairs of convex and concave polygons. Then only pairs of concave polygons are used to produce better results. Fig.2.18. depicts match features of polygons.

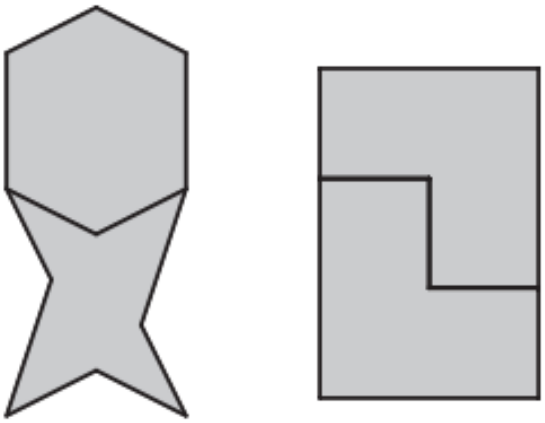


Figure 2.18: Matched features of polygons

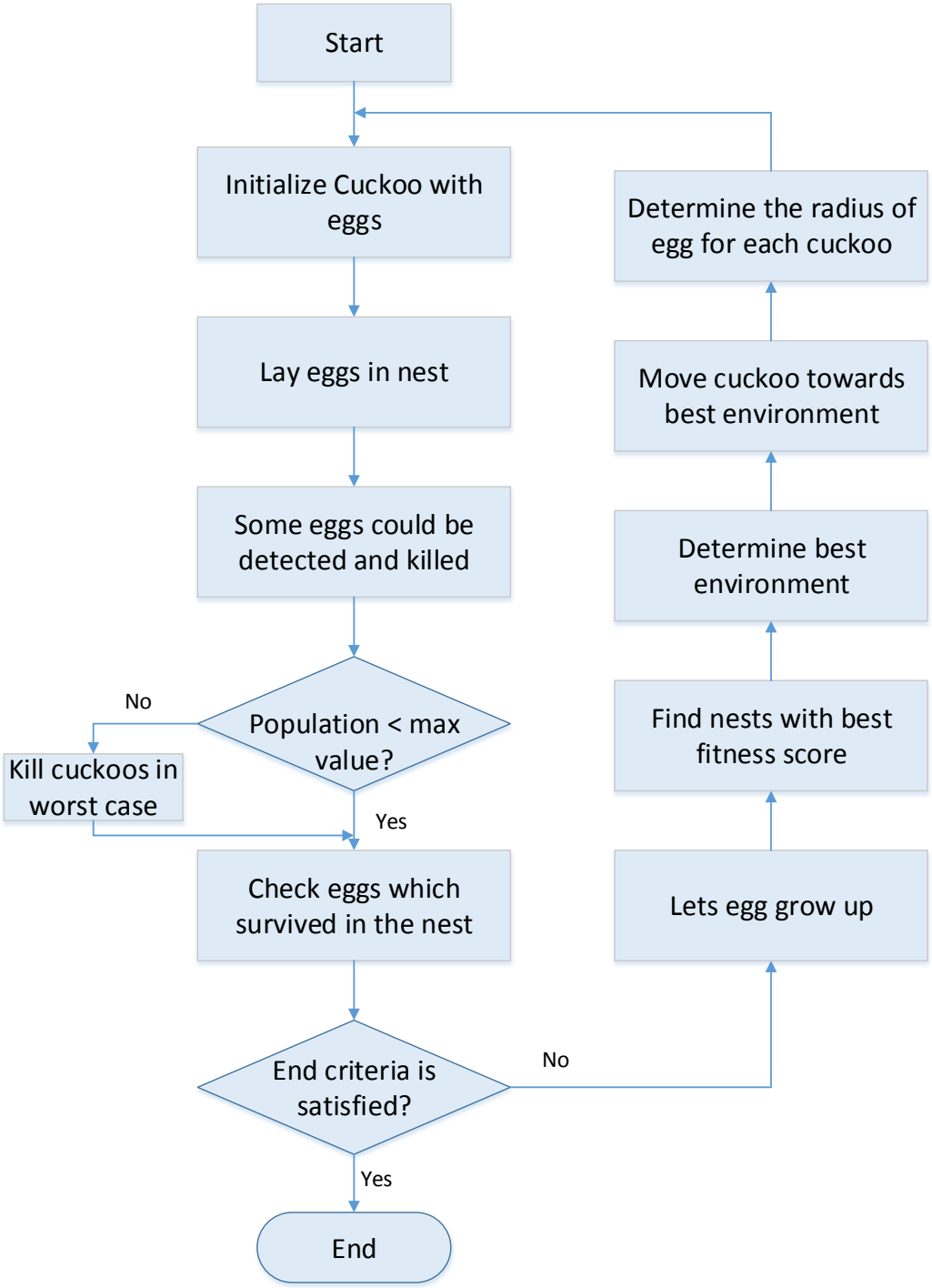


Figure 2.17: Flow chart diagram of Cuckoo search algorithm

Chapter 3

EXPERIMENTAL SETUP

3.1 Overview

The proposed algorithm is simply implemented by using MATLAB R2018a and Microsoft Visio 2016. Different data sets are created and used to validate our proposed algorithm. Some data sets are already available on ESICUP website [71] and these data sets are used by the researchers to check the effectiveness of their solutions to the stock placement problem. Next section describes the coordinates and physical shapes of all these data sets.

3.2 Data Sets

This section describes the coordinates and shapes of all data sets that are used in our proposed algorithm.

3.2.1 Data Set 1

This data set is created by us and used in our proposed algorithm. Fig.3.1. shows the basic shapes of this data set. The data set contains 4 shapes with area of 8, 3, 4 and 4 and each step is 1 unit. The average number of vertices by piece are 6.5. Table 3.1. describes the basic coordinates of this data set.

Figure 3.1: Shapes of data set 1

Shape 1	X	0	2	2	10	10	12	12	0
	Y	0	0	2	2	0	0	4	4
Shape 2	X	0	-4	-4	-2	-2	2		
	Y	0	0	4	4	2	2		
Shape 3	X	0	2	2	4	4	6	6	0
	Y	0	0	-2	-2	0	0	2	2
Shape 4	X	0	4	4	0				
	Y	0	0	4	4				

Table 3.1: Coordinates of data set 1

3.2.2 Data Set 2

This data set contains 7 shapes with area of 4, 7, 5, 6, 4 , 7 and 6. Each step is 1 unit in length. Average number of vertices by piece are 7.2. Fig.3.2. shows the basic shapes of this data set. Coordinates of this data set are given Table 3.2.

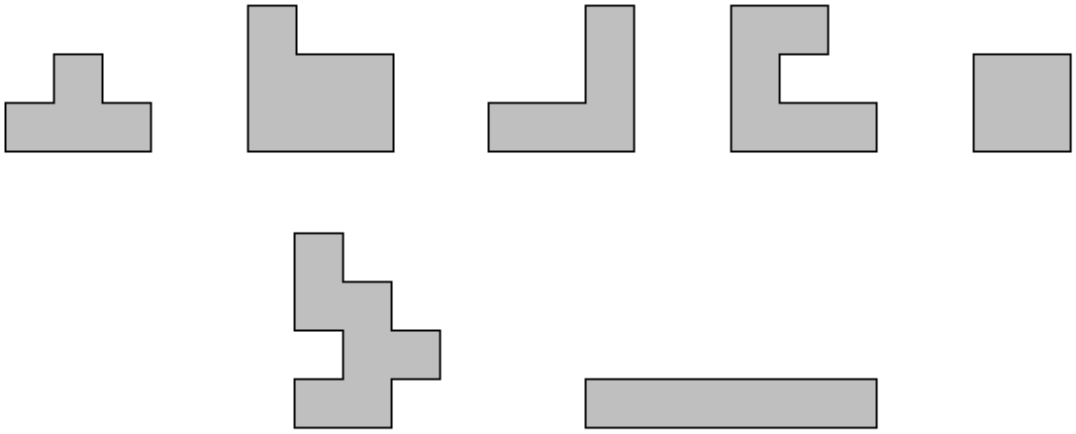


Figure 3.2: Shapes of data set 2

3.2.3 Data Set 3

The data set contains 6 shapes with area of 4, 2, 3, 3.75, 4 and 7 and each step is 1 unit. The average number of vertices by piece are 6.6. Fig.3.3. shows the basic shapes of this data set. Coordinates of this data set are given Table 3.3.

3.2.4 Data Set 4

A data set hexagon is used to experiment the results of proposed approach. Each length is 2 units. Number of vertices are 6. The area of hexagon is 10.39. Table 3.4. shows the coordinates of hexagon.

Shape 1	X	0	2	2	4	4	6	6	0						
	Y	0	0	-2	-2	0	0	2	2						
Shape 2	X	0	2	2	6	6	0								
	Y	0	0	2	2	6	6								
Shape 3	X	0	4	4	6	6	0								
	Y	0	0	-4	-4	2	2								
Shape 4	X	0	4	4	2	2	6	6	0						
	Y	0	0	2	2	4	4	6	6						
Shape 5	X	0	4	4	0										
	Y	0	0	4	4										
Shape 6	X	0	2	2	4	4	6	6	4	4	0	0	2	2	0
	Y	0	0	2	2	4	4	6	6	8	8	6	6	4	4
Shape 7	X	0	12	12	0										
	Y	0	0	2	2										

Table 3.2: Coordinates of data set 2

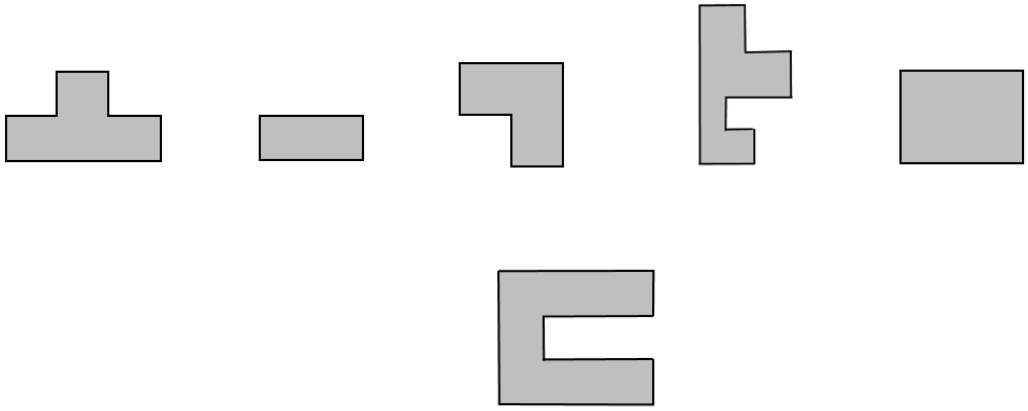


Figure 3.3: Shapes of data set 3

3.2.5 Data Set 5: SHAPES0

Data set SHAPES0 is presented on the ESICUP website [71] and we used this data set to experiment and benchmark our proposed algorithm. The data set contains 4 different pieces and total number of pieces are 43. Average number of vertices by piece are 8.75. Sheet width used is 40. Fig.3.4 shows the basic shapes of this data set. The coordinates of these shapes are extracted from the ESICUP website [71].

3.2.6 Data Set 6: Dighe1

This data set is also presented on the ESICUP website [71] and we used this data set to experiment and benchmark our proposed algorithm. The data set contains total 16 unique pieces. Sheet width used is 100. Fig.3.5 shows the basic shapes of this data set.

Shape 1	X	0	2	2	4	4	6	6	0		
	Y	0	0	-2	-2	0	0	2	2		
Shape 2	X	0	4	4							
	Y	0	0	2	2						
Shape 3	X	0	4	4	2	2	0				
	Y	0	0	4	4	2	2				
Shape 4	X	0	2	2	4	4	1	1	2	2	0
	Y	0	0	2	2	4	4	5	5	6	6
Shape 5	X	0	4	4	0						
	Y	0	0	4	4						
Shape 6	X	0	6	6	2	2	6	6	0		
	Y	0	0	2	2	4	4	6	6		

Table 3.3: Coordinates of data set 3

Shape 1	X	0	2	4	4	2	0
	Y	0	-1	0	2	1	2

Table 3.4: Coordinates of hexagon

The coordinates of these shapes are extracted from the ESICUP website [71].

3.2.7 Data Set 7: Dighe2

This data set is also presented on the ESICUP website [71] and we used this data set to experiment and benchmark our proposed algorithm. The data set contains total 10 unique pieces. Sheet width used is 100. Fig.3.6 shows the basic shapes of this data set. The coordinates of these shapes are extracted from the ESICUP website [71].

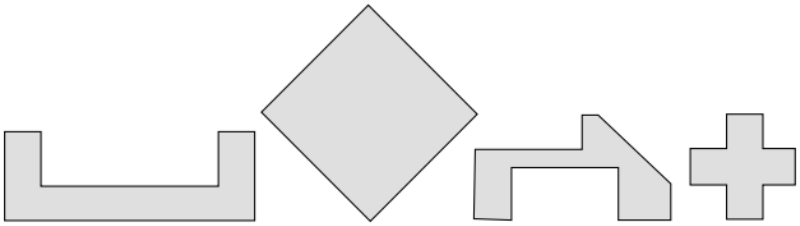


Figure 3.4: Data set : SHAPES0

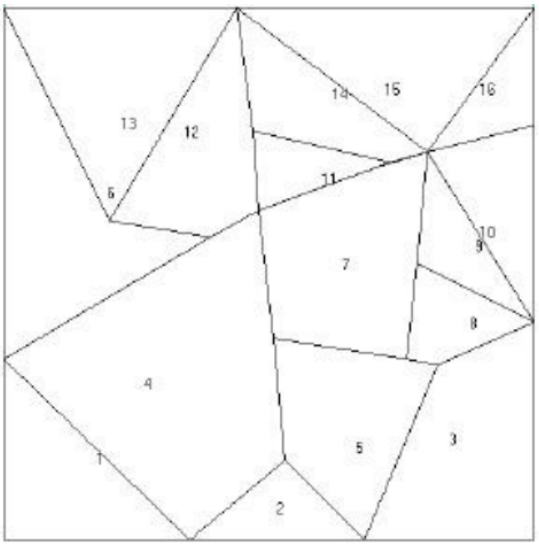


Figure 3.5: Data set : Dighe1

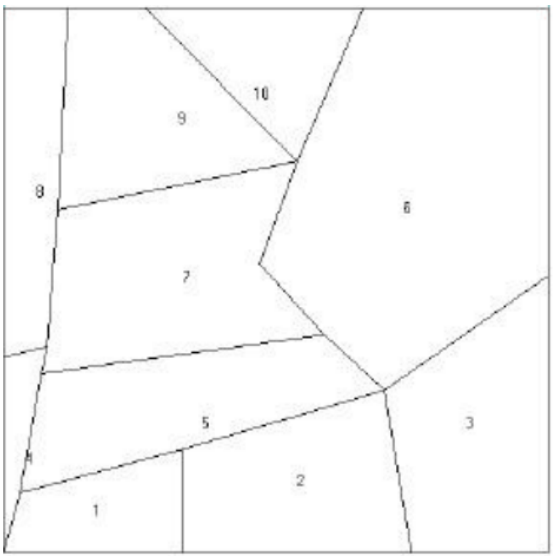


Figure 3.6: Data set : Dighe2

Chapter 4

METHODOLOGY AND PROPOSED TECHNIQUE

4.1 Problem Formulation

This section explains the mathematical formulation of nesting problem. Consider a set of polygons denoted by a set P = $\{p_1, p_2, p_3, p_4, ..., p_n\}$, the possible permitted angles A = $\{a_1, a_2, a_3, a_4, ..., a_n\}$ and a rectangular sheet S(W_{fxd},L_{used}) with $W_{fxd} \geq 0$ and$L_{used} \geq 0$ where W_{fxd} denotes the fixed width of polygon and L_{used} denotes the length that is actually used after the placement of all polygons. For simplicity, the sheet edges are set parallel to x and y axis as shown in Fig.4.1.

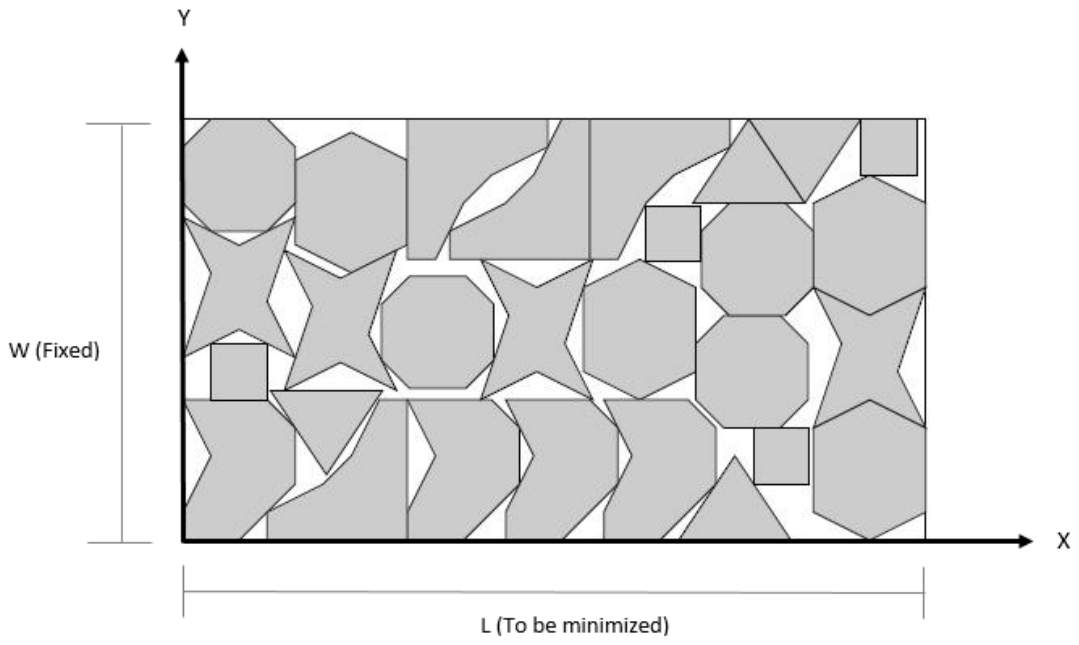


Figure 4.1: Example of sheet layout

A polygon $p_i|p_i \in P$ with angle $a_i|a_i \in A$ is denoted by $p_i(a_i)$ which may be written as p_i for convenience. The position of p_i is described by vector $\mathcal{V}_i = (x_i, y_i)$. The translation of a polygon by position vector $\mathcal{V}_i$ can be defined by Minkowski sum $p_i \oplus \mathcal{V}_i$. The main objective of nesting problem is to maximize the sheet utilization

or minimized the sheet length L such that no pieces overlap and cross the boundary of sheet. The sheet nesting problem is mathematically described as follows:

$$minL_{used} = max\{x|(x,y) \in (p_i \oplus \mathcal{V}_i)\} - min\{x|(x,y) \in (p_i \oplus \mathcal{V}_i)\} \tag{4.1}$$

$$\begin{aligned} maxU &= \sum_{i=1}^{n} Area_{pieces}/W_{fxd} * L_{used} \\ s.t. &(p_i(a_i) \oplus \mathcal{V}_i) \cap (p_j(a_j) \oplus \mathcal{V}_j) = \emptyset, \\ &(p_i(a_i) \oplus \mathcal{V}_i) \subseteq S(W_{fxd}, L_{used}) \end{aligned} \tag{4.2}$$

Where maxU describes the maximum utilization of sheet.

4.2 Proposed Technique

A heuristic algorithm is used for proposed optimization problem. This is an iterative algorithm which performs iterations until all the polygons are placed inside the main sheet. The objective of each iteration is to find an optimal placement where maximum boundary of floating polygon overlap with sheet boundary and placed polygons followed by minimum convex hull value and minimum sheet wastage. The algorithm starts with fixing the first polygon at any placement inside the main sheet however in our case mostly the first polygon is fixed at the top left corner of the stock. In the next step, the area of each polygon is calculated and these polygons are saved in a variable X_{flt} in the descending order based on their areas. Multiple polygons from variable X_{flt} are used at a time to place at the most feasible placement inside the stock. The multiple polygons that are used at a time to place inside the sheet can yield better results because the are much chances that one of these polygons might fit properly at the feasible placement. The flow chart diagram explaining the complete algorithm is shown in Fig.4.2. The most important steps of algorithm are Group unique polygons, Boundary extraction, overlap detection and Objective function. All the steps of algorithm are explained in detail in next section.

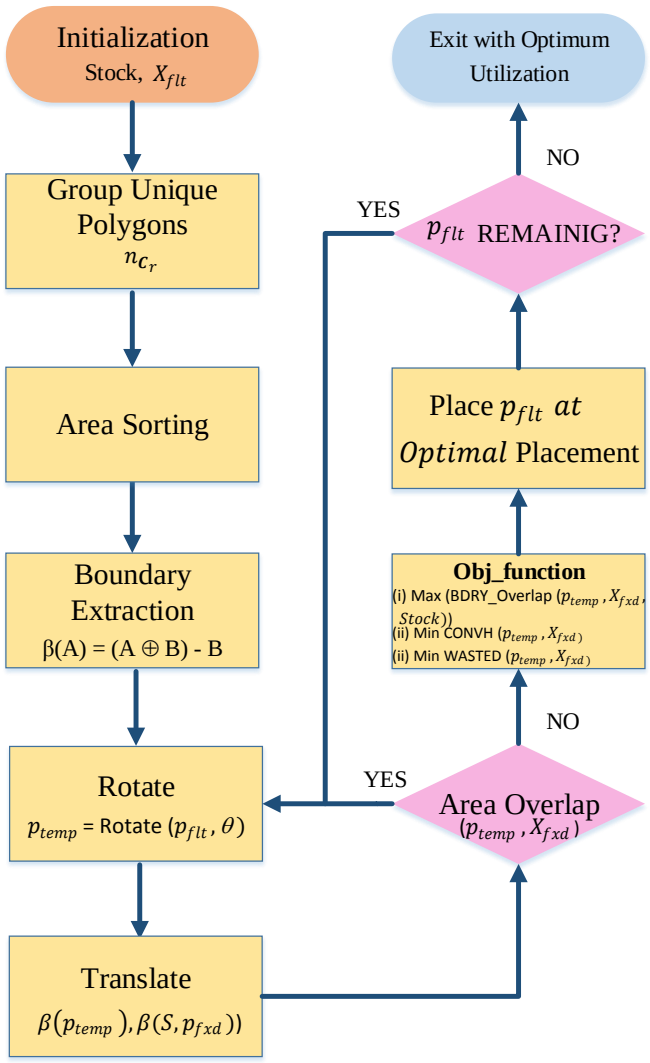


Figure 4.2: Algorithm flow chart diagram

4.2.1 Pseudo Code of Proposed Algorithm

This section explains the pseudo code of our proposed algorithm that is built using MATLAB R2018a. The coordinates of stock and polygons are saved in Excel sheet. Basic variables and functions of variables used in the algorithm are explained as:

$Stock$:	The main stock sheet.
X_{flt} :	Set of all floating polygons.
X_{fxd} :	Set of all fixed polygons.
p_{flt} :	A floating polygon where $p_{flt} \in X_{flt}$.
$Grouped(p\dot{}_{flt}, p\ddot{}_{flt})$:	A function that grouped the unique polygons.
p_g :	A grouped polygon.
$Area(\dot{p_g}, desc)$:	A function that calculates the area of grouped polygons and arrange them in descending order.
$Bound\_Ext(Stock, X_{fxd})$:	A function that extracts the boundary of $Stock$ and fixed polygons.
$Rotate(p_{flt}, \theta)$:	A function that rotates the p_{flt} by angle θ .
p_{temp} :	A temporary polygon achieved with the rotation of p_{flt} by angle θ .
$\beta(A)$:	Boundary of polygon A.
$A \oplus B$:	Dilation of polygon A with B.
$Translate(p_{flt}, p_{temp}, Stock)$:	A function that moves the p_{flt} on the extracted boundary of $Stock$ and p_{fxd}.
p_{fxd} :	A fixed polygon where $p_{fxd} \in X_{fxd}$.
$OverlapDetection(p_{temp}, X_{fxd}, Stock)$:	A function that detects the overlap of p_{temp} polygon with all fixed polygon as well as stock.
$BDRYOverlap(p_{temp}, X_{fxd}, Stock)$:	A function that calculates the boundary overlap of p_{temp} polygon with all fixed polygon as well as stock.
$CONVH(p_{temp}, X_{fxd})$:	A function which calculated the convex hull for the polygons p_{temp} and X.
$WASTED(p_{temp}, X_{fxd})$:	A function that returns the wastage after the placement of p_{temp}.

Pseudo code is given as:

\\ Initialization:

Input the polygon's coordinates and save in variable X_{flt}.

Save stock coordinates in variable $Stock$.

Choose number of polygons that should be tried at

a time to place inside the stock.

Choose the angles θ.

Choose the first piece to be placed inside the stock

and place it at top left side of the stock.

\\ Placement Process:

Do {

For $\forall\ p_{flt} : p_{flt} \in X_{flt}$

{

$\dot{p_g} = Grouped(\dot{p_{flt}}, \ddot{p_{flt}})$

For $\forall\ \dot{p_g} : \dot{p_g} \in \dot{X_{flt}}$

{

$\ddot{p_g} = Area(\dot{p_g}, desc)$

$\beta = Bound\_Ext(Stock, X_{fxd})$

For $\forall$ values of θ

{

$p_{temp} = Rotate(\ddot{p_g}, \theta)$

For $\forall\ \beta_i(p_{temp}) : \beta_i(p_{temp}) \in \beta(p_{temp})$

{

For $\forall\ \beta(Stock)$

{

For $\forall\ \beta_i(p_{fxd}) : \beta_i(p_{fxd}) \in \beta(p_{fxd})$

{

$p_t = Translate(p_{temp}, Stock, p_{fxd})$

$Area\ \ overlap(p_t, X_{fxd}) = ?No$

{

$Max\ \ BDRY\_Overlap(p_t, X_{fxd}, Stock)$

$Min \;\; CONVH(p_t, X_{fxd})$

$Min \;\; WASTED(p_t, X_{fxd})$

}

}

}

}

}

}

}

Best Placement = p_{flt} \\ Save best placement.

$p_{flt} \Leftarrow X_{flt}$ \\ Remove the p_{flt} from X_{flt}

Update X_{flt}

}

While p_{flt} remaining in X_{flt}

4.2.2 Initialization

In the initialization step, input the main sheet coordinates and save in sheet. The polygons coordinates are saved in variable X_{flt}. In our case, the polygon coordinates are read from a file. The algorithm assumes that one polygon must be placed at known placement with orientation that yields maximum overlap with sheet boundary, so that the remaining polygons should be placed on maximum boundary overlap of fixed polygon.

4.2.3 Group Unique Polygons

In any 2-D nesting problem, the goal is to place as many as pieces inside the sheet such that no piece overlap with each other and cross the boundary of main sheet. In our approach, the concept of minimizing the the waste depends upon the concept of placing the optimal grouped polygons inside the sheet. The grouped polygons would be a combination of two, three, four and a maximum of five unique polygons depending

upon the placement routine. The polygons are grouped using the following formula:

$$\mathcal{C}(n,r) = \frac{n!}{(n-r)!r!} \tag{4.3}$$

Placement routine is a function that returns maximum value of boundary overlap, minimum value of convex hull and minimum value of wastage. Placement routine function is described in detail below:

4.2.3.1 Boundary Overlap

The first and the most important component of placement routine is boundary overlap. Boundary overlap is maximum pixel/perimeter overlap of floating polygon with boundary of sheet and boundary of placed polygons inside the sheet. Fig. 4.3. depicts the maximum boundary overlap of polygon A with sheet.

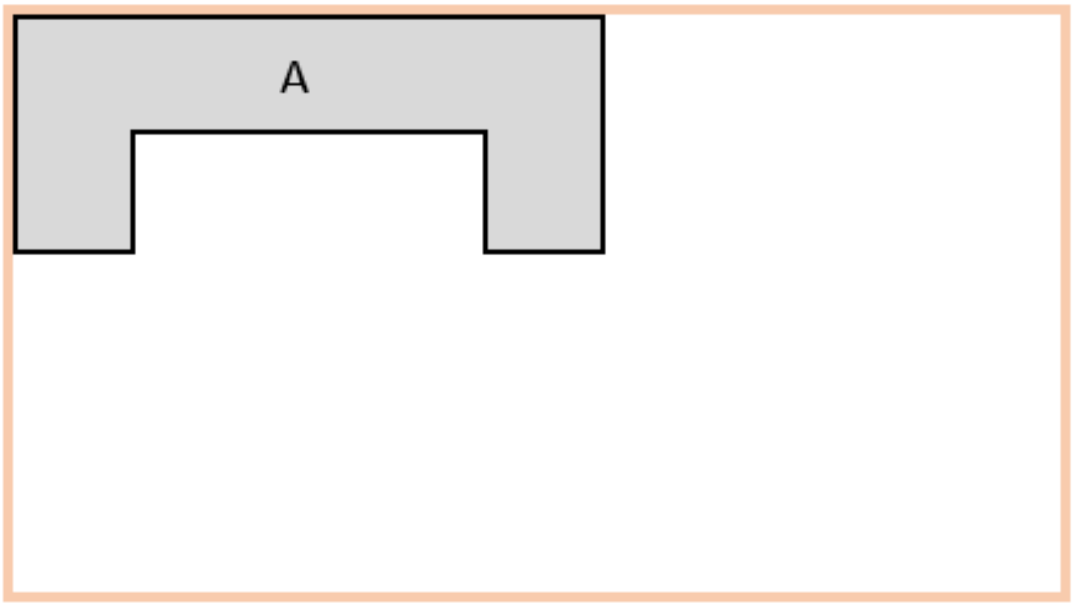


Figure 4.3: Boundary Overlap of polygon A with sheet

Consider a polygon placement example in which polygon A has an area of 7 and perimeter of polygon A is 16. The perimeter of the polygon is the sum of lengths of the side of the closed polygon. It can be seen in Fig.4.3. that maximum perimeter of 7 overlaps with the boundary of sheet at given orientation. Consider an other placement example where a rectangular polygon B is placed at the boundary of polygon A such that polygon A and B do not overlap, Fig.4.4. Whenever a polygon B is placed at the

boundary of polygon A, new configuration is made. The configuration with maximum boundary overlap will be considered as the final placement. Fig.4.4 depicts some configuration and clearly, when the polygon B is placed at hole (right fig) it will result in maximum boundary overlap of polygon B with polygon A.

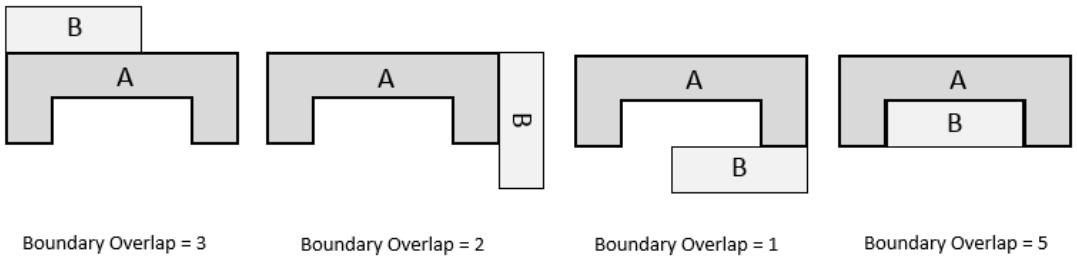


Figure 4.4: Some boundary overlap orientations of two polygons

4.2.3.2 Convex Hull

Second component of placement routine is convex hull. Convex hull of a set P of points is the smallest set which consists of all the points of P. Convex hull is described by an example shown in Fig.4.5. In this example, area of polygon A is 7 and it's convex hull area is 10.

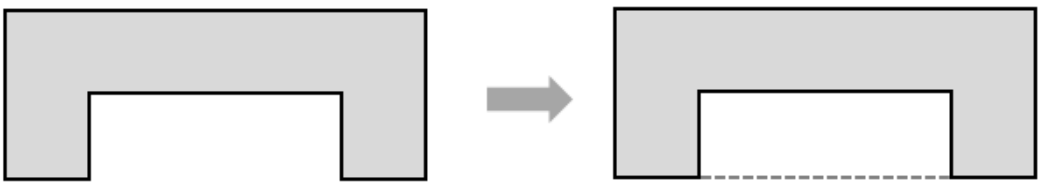


Figure 4.5: Convex hull of a polygon

Consider an other example where polygon B has to place on the boundary of polygon A with minimum convex hull value Fig.4.6. Only those configurations are shown in Fig.4.6. which has minimum convex hull area. Clearly, configuration 1 has minimum convex hull area so this will be considered as final placement.

From the placement routine, it's more convenient to group more than two polygons for better compaction, Fig.4.7. It can been seen that of all possible positions, placing

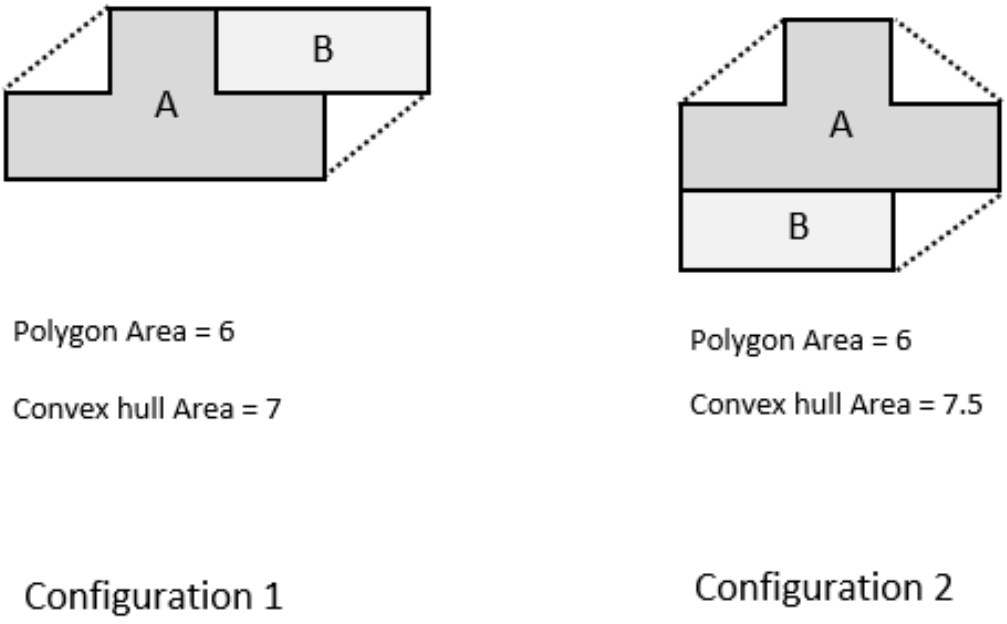


Figure 4.6: Polygon's placement with minimum convex hull area

the polygons B and C as shown in Fig.4.7. is more suitable as it yields minimum convex hull area. So in our proposed methodology, our objective is to group two or more than two polygons for better compaction.

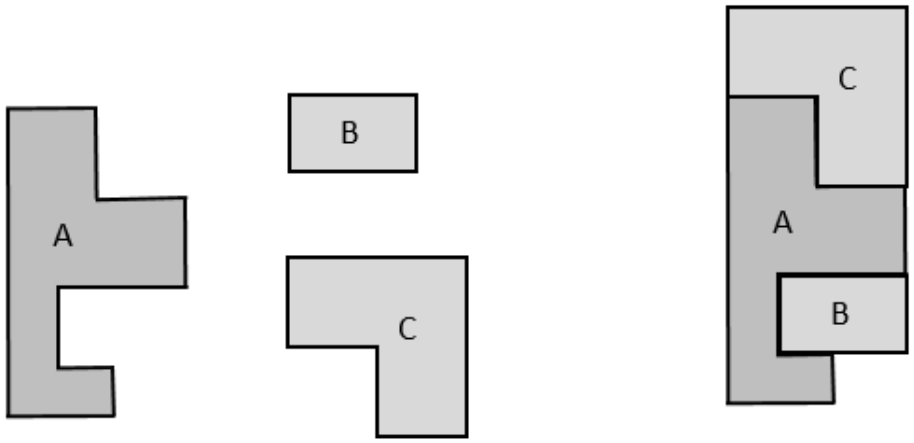


Figure 4.7: Placement routine optimization with convex hull

4.2.3.3 Wasted

The third component of placement routine is minimizing the sheet wastage. Wastage is minimized by finding the holes in sheet after placement of polygons. It can be seen in Fig.4.4. (right side) when the polygon B is placed at hole it will cover the hole and wastage is zero in this case. In placement routine optimization Fig.4.7., the wastage is also zero.

4.2.4 Geometric Transformations

Translation and rotation are required to place the polygon on each other. Suppose we want to translate (x,y) coordinates of a Polygon in the Cartesian coordinate system to another place with coordinates (X,Y), where X= x+r and Y=y+s. For translation and rotation, we use homogeneous solution in which third component of the column is 1. The relationship between (X,Y) and (x,y) can be written in matrix as:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & r \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.4}$$

Rotational transformation can be used to rotate (x,y) coordinates by some angle θ and it yields a new point (X,Y), the matrix relationship can be expressed as:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.5}$$

where

$$X = x\cos\theta - y\sin\theta \tag{4.6}$$

$$Y = x\sin\theta + y\cos\theta \tag{4.7}$$

In some cases, it is required to translate and rotate polygon's coordinates (x,y) simultaneously. The combined function of translation and rotation can be written as follows:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & r \\ \sin\theta & \cos\theta & s \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.8}$$

4.2.5 Boundary Extraction

In our proposed algorithm, boundary of sheet and placed polygon is extracted for optimal placement. Boundary is extracted by using dilation and erosion equations. Equation 4.9 and 4.10 illustrates the mathematical forms of dilation and erosion respectively.

$$\begin{aligned} A \oplus B &= \{z|(\hat{B})_z \cap A \neq \emptyset\} \\ A \oplus B &= \{z|[(\hat{B})_z \cap A] \subseteq A\} \end{aligned} \tag{4.9}$$

In this equation,B is reflected around it's origin then shifted by z. The dilation of A and B is simply a set of all placements z, such that $\hat{B})$ and A overlap contains at least one element.

$$\begin{aligned} A \ominus B &= \{z|(B)_z \subseteq A\} \\ A \ominus B &= \{z|(B)_z \cap A^c = \emptyset\} \end{aligned} \tag{4.10}$$

The erosion equation shows that erosion of A and B is set of all placements z, such that translation of B is a subset of A.

In dilation rule, the boundary of set A denoted by $\beta(A)$ can be extracted by first dilating A by B then measuring the difference between set A and it's dilation with B. That is,

$$\beta(A) = (A \oplus B) - A \tag{4.11}$$

In erosion rule, the boundary of set A denoted by $\beta(A)$ can be extracted by first eroding A by B then measuring the difference between set A and it's erosion with B. That is,

$$\beta(A) = A - (A \ominus B) \tag{4.12}$$

4.2.6 Overlap Detection

In our proposed algorithm, we place the new polygons on the boundary of already placed polygons. In this placement procedure, no polygon should be overlapped with already placed polygons. If any polygon overlaps with already placed polygons then a overlap detection alarm will be triggered pointing that such orientation is not allowed. In our proposed technique, overlap detection is checked by polygons area. This can be explained by a simple example. Consider a Polygon A is already placed inside the sheet and it's area is denoted by A_{area}. A new polygon B has to be placed inside the sheet such that it should not overlap with the polygon A. Polygon B area is denoted by B_{area}. The combined area of A and B after B placement is denoted by AB_{area} as shown in Fig.4.8 . If one polygon placed at some boundary of another polygon then following condition shows that two polygons are overlapping:

$$AB_{area} < A_{area} + B_{area} \tag{4.13}$$

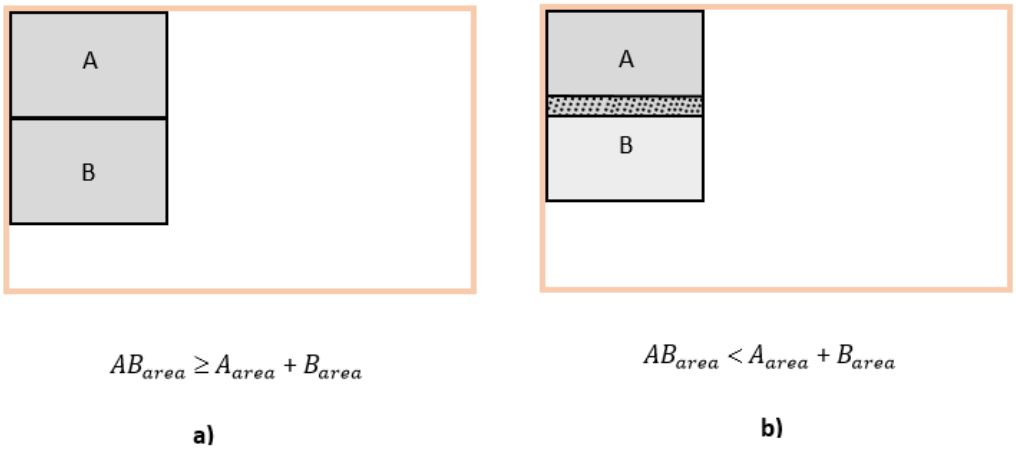


Figure 4.8: Overlap detection, (a) No area overlap (b) Area overlap

4.2.7 Objective Function

The objective function proposed in our approach is comprised of three elements. The first element is boundary overlap, which aims to maximize the pixel/ perimeter overlap of a polygon with sheet and existing polygons as illustrated in section 4.2.2.1. The

better compaction is achieved by maximizing the polygon's overlap with the sheet and existing polygons.

Second, the convex hull which tends to minimize the convex hull of a polygon with existing polygons as illustrated in section 4.2.2.2. Convex hull is the smallest set of points which contains all the points of a polygon.

Third, the wasted which tends to minimize the holes in sheet after placement of polygons as illustrated in section 4.2.2.3.

Chapter 5

RESULTS AND DISCUSSION

5.1 Result and Analysis

Experimental results of this work are obtained with MATLAB R2018a setup to optimize equation 4.1 and 4.2 . Results include some experimental examples and benchmark problems. The results of different nesting problems available in the literature are used to evaluate and compare with our results. Some experimental examples are used in the next sections to validate our approach and these examples allow us to make decisions about parameters tuning and implementation of heuristic algorithm.

5.2 Model Parameters

By using some empirical examples, it is noted that good choice of a few parameters of algorithm may yield better results. For example, the number of polygons used in grouping significantly effect the quality of solution and computational time. As we increase the number of polygons used in the group, the computational time increases a lot, but the groups of large number of polygons may improve the stock utilization. The number of grouped polygons that are tried at a time to place the most suitable polygon at most optimal placement significantly improves the results but it increases the computational time. Additionally, the some angles can yield better results in some cases however it can produce bad results in other cases. If the shapes have some similarities then a small angle may yield worse output than large angle and it effects the sheet utilization. It is also noted from the experimental results that in case of orthogonal shapes, 90° incremental angle is a good angle and can produce better results, see Fig.5.3. This is because in case of orthogonal shapes, 90° may ensure better match between different shapes of polygons. Algorithm performs very well in case of orthogonal shapes because the shapes will be grouped perfectly without any confirm wastage. Further, the large list of shapes may increase the computational time because it will take more time

at the grouping stage. In most cases, the first polygon is placed at the left upper corner of rectangular sheet. Although the position of first polygon may effect the solution and should be set carefully. This will only change no more than 5 % of the sheet utilization.

5.3 Experimental Results

The proposed algorithm is verified on nesting problem. Few experiments are done to validate the feasibility of our algorithm. Later we also compare the results with best known benchmark problem available in the literature.

5.3.1 Experiment 1

This experiment is used to prove that single objective function with only convex hull is not a good optimization criteria as it may result in wastage. Consider an example in which we have a set of orthogonal shapes as shown in Fig.5.1. Clearly, when the item 1 is placed on the same item, it results in four orientations with same value of convex hull, see Fig.5.2.

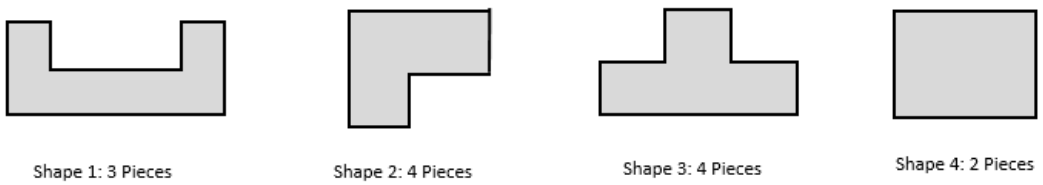


Figure 5.1: Experiment 1: Polygon shapes with area: 8, 3, 4 and 4 respectively

Piece 1 is placed initially inside the sheet and when we place piece 2, there are 4 possible orientations with minimum convex hull value of 28. Obviously, some orientations may be discarded due to overlap with polygons or piece 2 cross the boundary of main sheet. Since all the orientations have same value of convex hull so piece 2 can be placed at any orientation. But as we can see in Fig.5.2.(a), (b), (d) when piece 2 is placed inside the main sheet there is a confirm wastage. So single objective function with only convex hull is not a good optimization criteria as it results in wastage. Fig.5.3. shows the placement of shapes with Doraid Dalalah [3] and proposed algorithm. In proposed approach the unique shapes are grouped first based on the placement routine and these

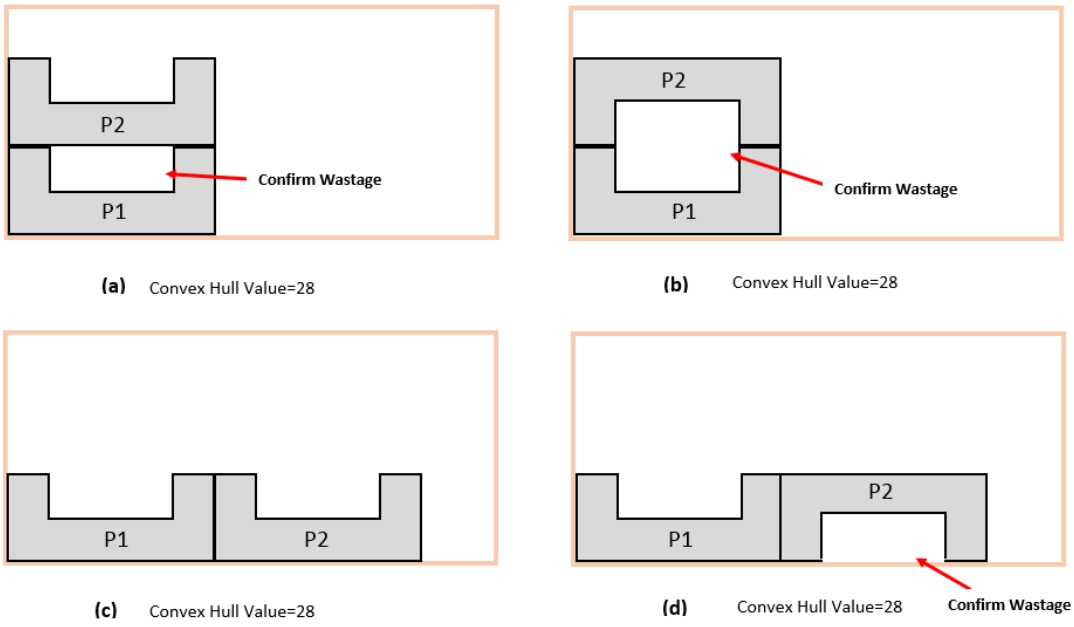


Figure 5.2: Experiment 1: Same convex hull value for all orientations

grouped shapes are represented by dotted lines. Total 12 shapes are placed inside the sheet with sheet utilization of 93.33%. Table 5.1 shows the results comparison of proposed algorithm and Doraid Dalalah [3].

Reference	Sheet area	# of shapes	# of used shapes	Unused area	Utilization (%)
Doraid Dalalah [3]	60	13	11	6	90
Proposed Algorithm	60	13	12	4	93.33

Table 5.1: Experiment 1: Results of Doraid Dalalah [3] and proposed algorithm

5.3.2 Experiment 2

Here the shapes shown in Fig.5.4. a) has to be placed inside the non convex sheet Fig.5.4. b). Each shape has 5 pieces and each mark is 1 unit long. The sheet used here is non convex with each step of 5 units. Fig.5.5. shows the placement of these shapes with Doraid Dalalah [3] and our algorithm. In our case total 30 pieces are placed inside the main sheet with maximum utilization of 93% which makes our approach superior to Doraid Dalalah [3]. Table 5.2 shows the results comparison of our algorithm and Doraid Dalalah [3] for these shapes and stock.

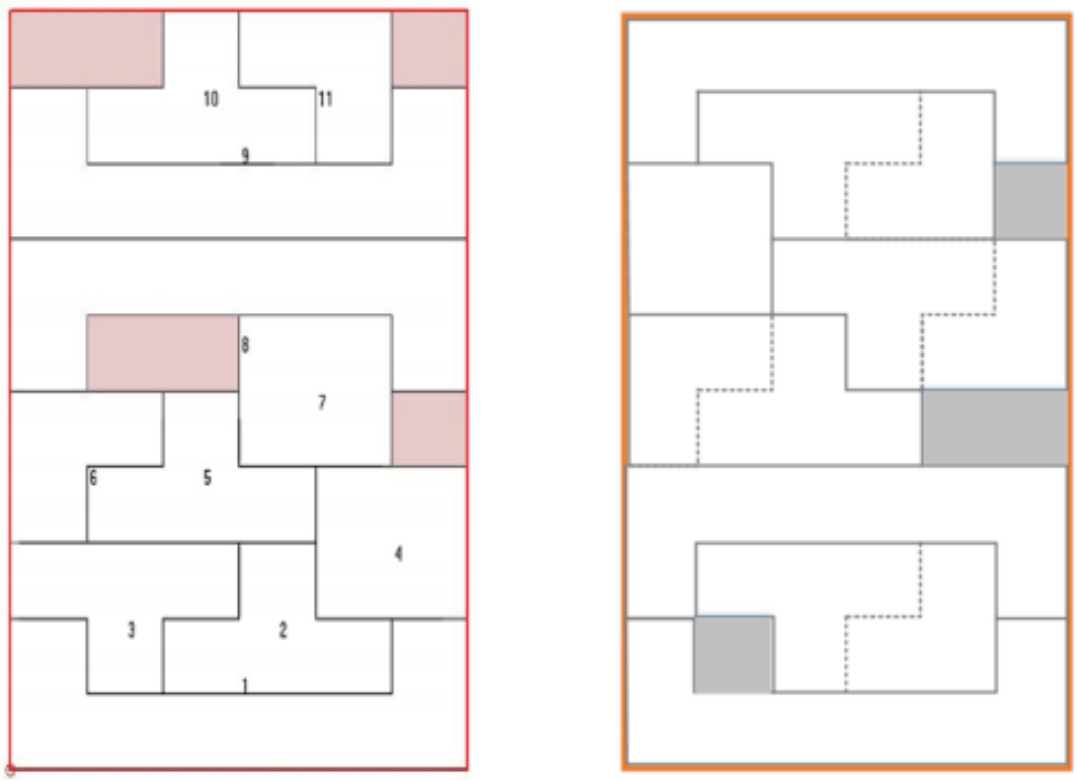


Figure 5.3: Experiment 1: Placement of shapes in 6 x 10 rectangular sheet a) Doraid Dalalah [3] with sheet utilization of 90% b) Proposed approach with sheet utilization of 93.33%

Reference	Sheet area	# of shapes	# of used shapes	Unused area	Utilization (%)
Doraid Dalalah [3]	175	35	29	18	89.7
Proposed Algorithm	175	35	30	13	93

Table 5.2: Experiment 2: Results of Doraid Dalalah [3] and proposed algorithm)

5.3.3 Experiment 3

In this experiment, the shapes shown in Fig.5.6. has to be placed inside the rectangular sheet of 8 x 12. Practical placement of some items is shown in Fig.5.7. and it can be seen that that among all the possible configurations where item 2 is placed with item 1, positioning the item 2 as shown in Fig.5.7 is more preferred as it has minimum value of convex hull. But in this situation, the convex hull value is same for both configurations, although there might be chances that one configuration yields better results than other one and have preference over other. Again, single objective function with only convex

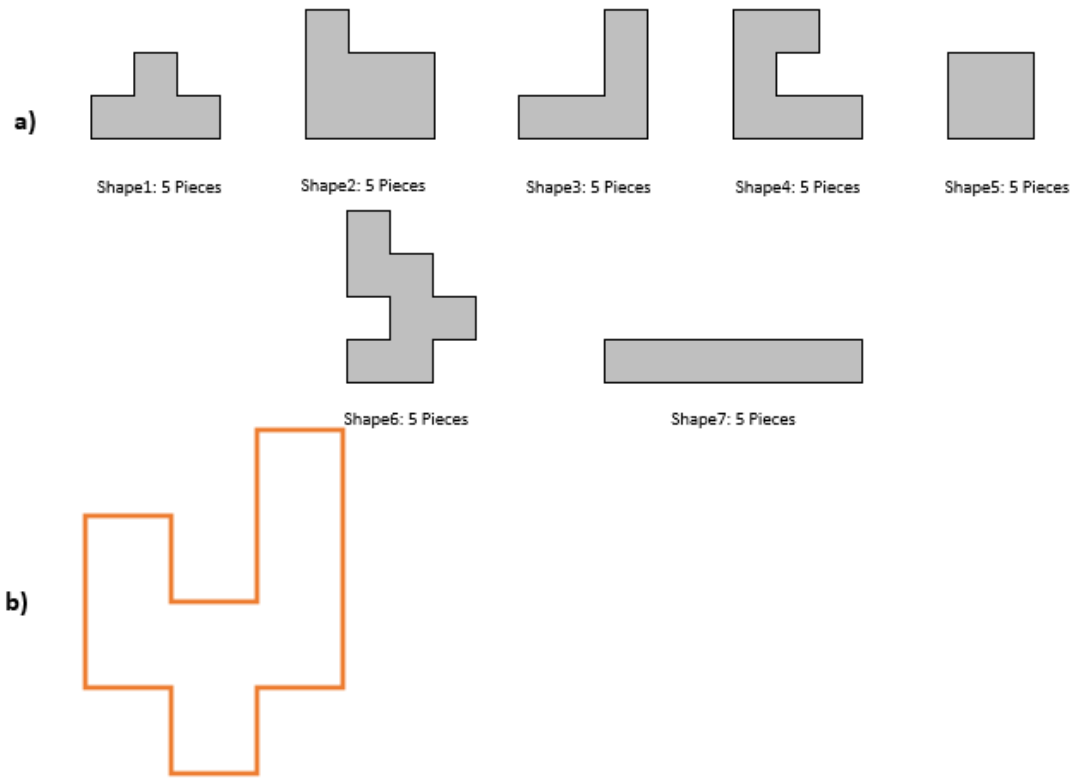


Figure 5.4: Experiment 2: a) Polygon shapes with area: 4, 7, 5, 6, 4, 7 and 6 respectively b) Non convex sheet with each step of 5 units long

hull is not a good optimization criteria as it conflicts the results. Fig.5.8. shows the results of our algorithm where unique shapes are grouped together and these grouped shapes are represented by dotted lines. These grouped shapes are then placed inside the sheet at most feasible positions where the shapes have maximum boundary overlap, maximum compaction and minimum wastage. Maximum utilization of 96% is achieved in this case due to orthogonal shapes. So our approach performs very well in case of orthogonal shapes. Here the unused area is 3.25 and shown by colored areas.

5.3.4 Experiment 4

In this experiment, hexagon shapes has to be placed inside the hexagon shaped stock. The length of one side of hexagon is 2 units and length of one side of stock is 10 units. One hexagon is fixed at the left side of the stock. Fig.5.9. shows the placement of all possible hexagons that could be placed inside the stock. Some spots are unused due to shape of stock and these unused ares are denoted by colored areas. In this case the utilization is 84% which is maximum utilization for this type of problem.

Figure 5.5: Experiment 2: a) Doraid Dalalah [3] with sheet utilization of 89.7% b) Proposed approach with sheet utilization of 93%

5.4 Benchmark Problems

This section describes the basic comparison of our algorithm's results with some literature results. Data set **shapes0** and **Dighe2** which are presented on the ESICUP website [71] are used to validate our proposed algorithm. Data set Dighe2 contains 10 unique pieces and sheet width is 100. Fig.5.10 shows the results of proposed approach for data set **shapes0**. Table.4.3. compares our proposed algorithm's results with some other literature's results. In the listed problem, the goal is to place all the items inside the stock while using minimum length of stock. It can be seen in the table.5.3 that our proposed algorithm results for data set **shapes0** are superior to all literature results except GLSHA: Gomes and Oliveira [60] which used the sheet length of 62. Fig.5.11. shows proposed algorithm results for data set **Dighe2**. The used length is 100 and utilization is 84%.

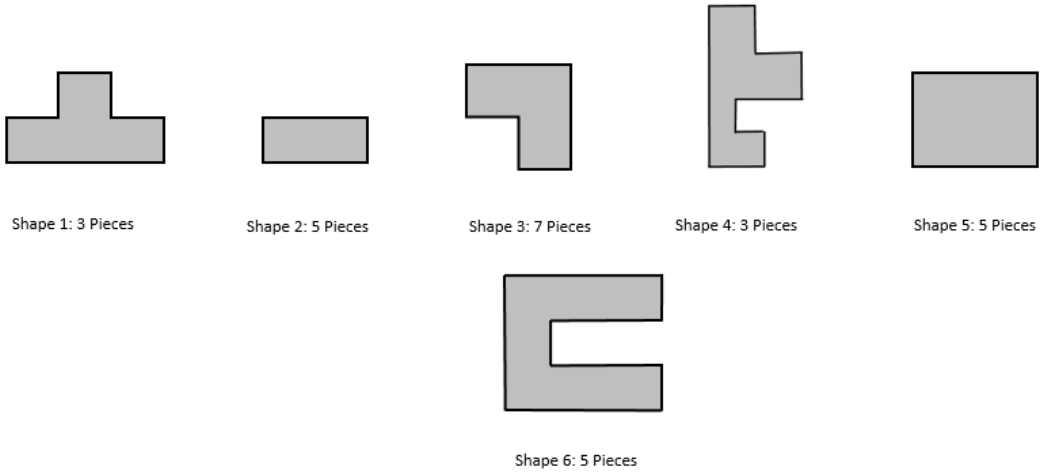


Figure 5.6: Experiment 3: Polygon shapes with area: 4, 2, 3, 3.75, 4, and 7 respectively

Reference	Sheet width	# of item types	# of items	Length	Time (s)
Hill climb: Burke et al. [72]	40	4	43	65	332.39
GLSHA: Gomes and Oliveira [60]	40	4	43	62	621
TOPOS: Oliveira et al. [58]	40	4	43	66.75	35.9
SAHA: Gomes and Oliveira [60]	40	4	43	63.15	3914
HAPE: Xiao and Jia [73]	40	4	43	67.55	79)
2DNEST: Egeblad et al. [5]	40	4	43	66	6(h)
Doraid Dalalah [3]	40	4	43	66	65
Proposed approach	40	4	43	64	2150

Table 5.3: Results of previous literature and proposed algorithm for shape0

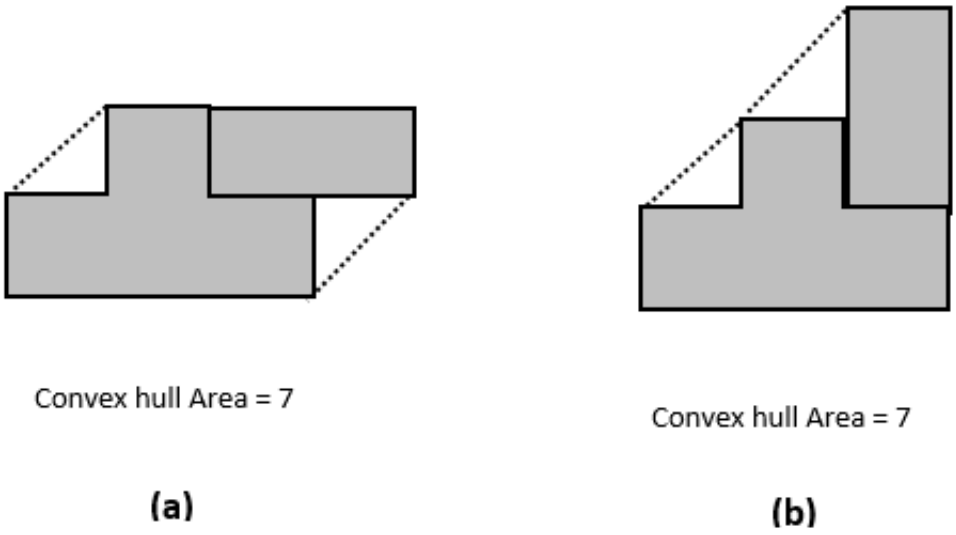


Figure 5.7: Experiment 3: Same convex hull value for two orientations

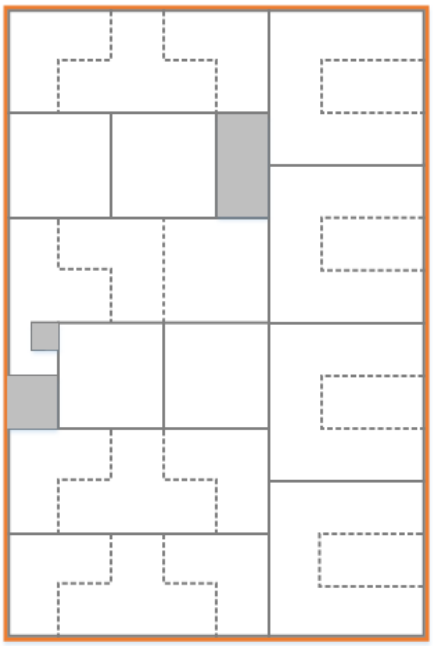


Figure 5.8: Experiment 3: Placement of shapes in 8 x 12 rectangular sheet with utilization of 96%

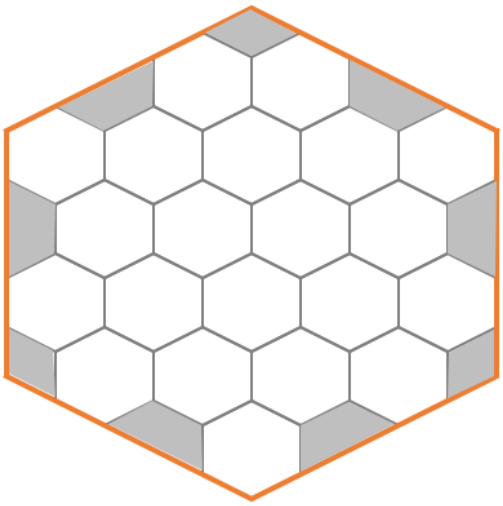


Figure 5.9: Experiment 4: Placement of hexagons in hexagon shaped sheet with utilization of 84%

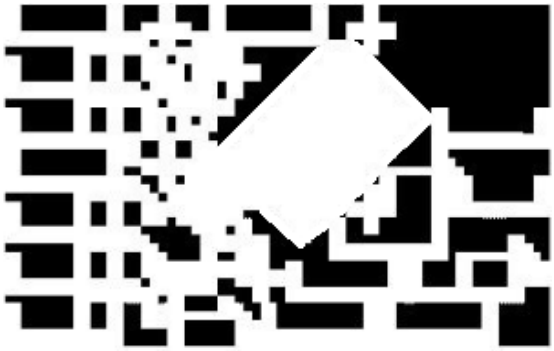


Figure 5.10: Stock placement of data set **shapes0** with proposed approach

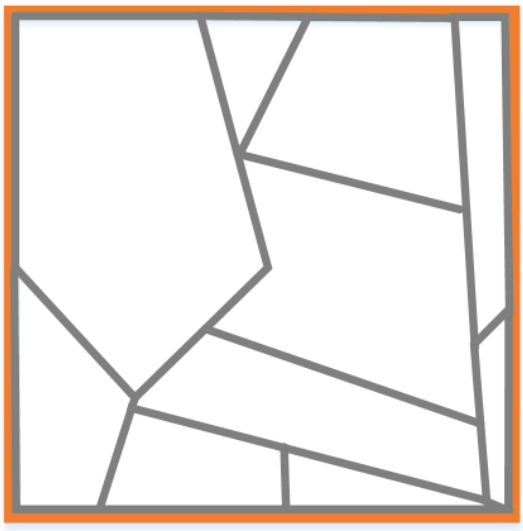


Figure 5.11: Stock placement of data set **Dighe2** with proposed approach

Chapter 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

This thesis addressed the stock placement problem. To solve this problem, a heuristic algorithm is proposed to minimize the wastage. The objective of this algorithm is to place the convex/non convex polygons inside the main stock such that no piece overlaps with each other and cross the boundary of stock. The objective of this research is to maximize the sheet utilization and minimize the stock wastage after placement of all polygons.

The heuristic algorithm used in this research is based on the concept of optimal groups of unique shapes based on placement routine which is a combination of three optimization functions: boundary overlap, compaction and wastage. Overlap detection algorithm is used to check the polygon's overlap. The algorithm starts with positioning the first polygon at random place inside the sheet, then iterative scheme is used to generate a list of placement positions and an objective function is measured against each placement which is again a combination of three optimization functions: boundary overlap, compaction and wastage. The polygon is placed at the placement which has maximum value of boundary overlap, minimum value of convex hull and minimum wastage.

The algorithm has been built in MATLAB R2018a with available data sets. Many examples with different polygons and stock shapes are tested with our algorithm, it is found that our approach yields good results. Different benchmark algorithms that are available in the literature are bench marked with our approach and found that our proposed technique is superior to the previous techniques used in the literature and makes our algorithm a very strong candidate for real industries.

6.2 Future Works

The algorithm proposed in this thesis has produced best results and successfully achieved the best known solutions on all the readily available benchmark problems. However, the author believes that several modifications in this algorithm may yield further improvements in the quality and computational time of the solution. Accordingly, to avoid the overlaps of polygon/stock, further geometry check is required for feasible polygons placement. It was beyond the scope of this thesis to provide full complexity analysis of the algorithm and it shows direction to researcher to further investigate the algorithm. In future, it is required to black list the boundary pixels on which the algorithm should not be tried to save computational time.

BIBLIOGRAPHY

[1] http://hjemmesider.diku.dk/~jegeblad/

[2] Jiang-jiang Xu, Xin-sheng Wu, Hai-ming Liu, Mei Zhang "An Optimization Algorithm based on No-fit Polygon Method and Hybrid Heuristic Strategy for Irregular Nesting Problem", *in 36th Chinese Control Conference (CCC)*, 2017.

[3] Doraid Dalalah, Samir Khrais, Khaled Bataineh,"Waste minimization in irregular stock cutting", *Journal of Manufacturing Systems*, 33, pp.27-40, 2014.

[4] Jiang-jiang Xu, Xin-sheng Wu, Hai-ming Liu, Mei Zhang "An Optimization Algorithm based on No-fit Polygon Method and Hybrid Heuristic Strategy for Irregular Nesting Problem", *in 36th Chinese Control Conference (CCC)*, 2017.

[5] Egeblad J, Nielsen BK, Odgaard A. "Fast neighborhood search for two-and three dimensional nesting problems.", *European Journal of Operational Research*, 183(3), pp.1249-1266, 2007.

[6] Ahmed Elkeran. "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering", *European Journal of Operational Research*, 231(3), pp.757-769, 2013.

[7] Asvany Tandabani, KalaiPriyan, S. Janakiraman, Sujatha Pothula "A Comparative Study of Meta Heuristic Approach for Cutting Stock Problem", *in 2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016.

[8] Dyckhoff, H "A typology of cutting and packing problems.", *European Journal of Operational Research*, 44, pp.145-159, 1990.

[9] Gerhard Wascher, Heike HauBner, Holger Schumann, "An improved typology of cutting and packing problems", *European Journal of Operational Research*, 183, pp.1109-1130, 2007.

[10] Coffman, E.G., Garey, M.R., Johnson, D.S., "Approximation Algorithms for BinPacking ", *A Survey, in Approximation Algorithms for Bin Packing for NP-Hard Problems, Hochbaum, D.S. (eds), PWS Publishing Company, Boston*, pp.46-93, 1997.

[11] Martello, S. and Toth, P., "Knapsack Problems ", *Wiley and Sons, Chichester*,1990.

[12] Landa Silva, J.D. and Burke, E.K., "Hybrid Metaheuristics Based on Cooperative Local Search for the Space Allocation Problem", *Accepted for INFORMS Journal on Computing*, 2005.

[13] Landa Silva, J.D.,"Metaheuristics and multiobjective approaches for space allocation", *Ph.D Thesis, School of Computer Science and Information Technology,The University of Nottingham, UK*, 2003.

[14] Kendall, G.,"Applying Meta-Heuristic Algorithms to the Nesting Problem Utilizing the No Fit Polygon", *Ph.D Thesis, School of Computer Science and Information Technology, The University of Nottingham*, 2000.

[15] Dyckhoff, H "A typology of cutting and packing problems.", *European Journal of Operational Research*, 44, pp.145-159, 1990.

[16] Hopper, E., "Two Dimensional Packing utilizing Evolutionary Algorithms and other Meta-heuristic Methods ", *Ph.D Thesis, School of Engineering, University of Wales, Cardiff*, 2000.

[17] Brown, A.R., "Optimum Packing and Depletion", *The Computer in Spare and Resource Usage Problems, New York, London*, 1971.

[18] Salkin, H.M. and de Kluyver, C.A., "The Knapsack Problem: A Survey",*Naval Research Logistics Quarterly*, 22 , pp.127-144 1975.

[19] Golden, B.L., "Approaches to the cutting stock problem", *IIE Transactions*, 8, pp.265-274, 1976.

[20] Hinxman, A.I., "The trim loss and assortment problem - a survey", *Operations Research*, 5,8, pp.8-18, 1980.

[21] Garey, M.R. and Johnson, D.S., "Approximation algorithms for bin packing problems: A survey ", *in Analysis and Design of Algorithms in Combinatorial Optimization, Ausiello, D. and Lucertini, M. (eds), Wien*, pp.147-172, 1981.

[22] Israni, S.S. and Sanders, J., "Two-Dimensional Cutting Stock Problem Research: A Review and a New Rectangular Layout Algorithm", *Journal of Manufacturing Systems*, 1,2, pp.169-182, 1982.

[23] Sarin, S.C., "Two Dimensional Stock Cutting Problems and Solution Methodologies", *ASME Journal of Engineering for Industry*, 104,3, pp.155-160, 1983.

[24] Rayward-Smith, V.J. and Shing, M.T., "Bin packing", *Bulletin of the IMA*, 19, pp.142-146, 1983.

[25] Coffman, E.G., Garey, M.R., Johnson, D.S., "Approximation algorithms for bin packing - An updated survey",*in Algorithm Design for Computer System Design, Ausielo, G., Lucertini, N., Serafini, P. (eds), Springer-Verlag, Vienna*, pp.49-106, 1984.

[26] Berkey, J.O. and Wang, P.Y., "Two-Dimensional Finite Bin-Packing Algorithms", *Journal of the Operational Research Society*,38, pp.423-449,1985.

[27] Dowsland, W.B., "Two and Three Dimensional Packing Problems and Solution Methods",*New Zealand Journal of Operational Research*,13, pp.1-18,1985.

[28] Dyckhoff, H., Kruse, H.J., Abel, D., Gal, T., "Trim Loss and Related Problems", *Omega*,13, pp.59-72,1985.

[29] Israni, S.S. and Sanders, J.L., "Performance testing of rectangular parts-nesting heuristics", *International Journal of Production Research*,23,3, pp.437-456,1985.

[30] Dudzinski, K. and Walukiewicz, S., "Exact methods for the knapsack problem and its generalizations", *European Journal of Operational Research*, 28, pp. 3-21, 1987.

[31] Martello, S. and Toth, P, "Knapsack Problems", *Wiley and Sons, Chichester*, 1990.

[32] Rode, M. and Rosenberg, O., "An Analysis of Heuristic Trim-Loss Algorithms", *Engineering Costs and Production Economics*,12, 1-4, pp.71-78, 1987

[33] Dyckhoff, H., Finke, V., Kruse, H.J., "Standard Software for Cutting Stock Management", *Essays on Production Theory and Planning*, pp. 209-221, 1988.

[34] Coffman, E.G. and Shor, P.W., "Average-case analysis of cutting and packing in two dimensions", *European Journal of Operational Research*, 44, pp. 134-144, 1990.

[35] Dyckhoff, H. and Wascher, G., "Special Issue on Cutting and Packing", *European Journal of Operations Research*, 44, 2, 1990.

[36] Dowsland, W.B., "Three Dimensional Packing Solution approaches and Heuristic Development" *International Journal of Production Research*, 29, 8, pp. 1673-1685, 1991.

[37] Haessler, R.W. and Sweeney, P.E., "Cutting stock problems and solution procedures", *European Journal of Operational Research*, 54, pp. 141-150, 1991.

[38] Dowsland, K.A. and Dowsland, W.B., "Packing Problems", *European Journal of Operations Research*, 56, pp. 2-14, 1992.

[39] Dyckhoff, H. and Finke, V., "Cutting and Packing in Production and Distribution", *Physica-Verlag, Heidelberg, Germany*, 1992.

[40] Haessler, R.W., "One-dimensional cutting stock problems and solution procedures", *Mathematical Computer Modelling*, 16,1, pp. 1-8, 1992.

[41] Lirov, Y., "Knowledge Based Approach to the Cutting Stock Problem", *Mathematical Computer Modelling*, 16,1, pp. 107-125, 1992.

[42] Ram, B., "The Pallet Loading Problem: A survey", *International Journal of Production Research*, 28, pp. 217-225, 1992.

[43] Lodi, A., Martello, S., Vigo, D., "Recent advances on two-dimensional bin packing problems", *Discrete Applied Mathematics*, 123, pp. 379-396, 2002.

[44] Cagan, J., Shimada, K., Yin, S., " survey of computational approaches to three-dimensional layout problems", *Computer Aided Design*, 34, pp. 597-611, 2002.

[45] Lodi, A., Martello, S., Monaci, M., "Two-dimensional packing problems: A survey", *European Journal of Operations Research*, 141, pp. 241-252, 2003.

[46] Oliveira, J.F., "Solving nesting problems with metaheuristics: a survey", *ES-ICUP Meeting: 6-10 July*,Istanbul, Turkey 2003.

[47] Bennell, J. A., and Oliveira, J. F., "A tutorial in irregular shape packing problems.", *Journal of the Operational Research Society*, 60, pp. 93-105, 2009.

[48] Oliveira, J.F., Ferreira, J.S., "Algorithms for nesting problems, Applied Simulated Annealing", *In: Vidal, R.V.V.(Ed.), Lecture Notes in Econ. and Maths Systems)*, 396. Springer Verlag, pp. 255-274, 1993.

[49] Andr Kubagawa Sato, Thiago Castro Martins, Marcos Sales Guerra Tsuzuki, "An algorithm for the strip packing problem using collision free region and exact fitting placement", *Computer-Aided Design*, 44, pp. 766-777, 2012.

[50] `https://www.mathworld.wolfram.com/polygonvertex.com/`

[51] Art, R.C., "An approach to the two dimensional irregular cutting stock problem", *IBM Cambridge Scientific Centre, Report 36-Y08*, 1966.

[52] Adamowicz, M., Albano, A., "Nesting two dimensional shapes in rectangular modules", *Computer Aided Design*, 8(1), pp. 27-33, 1976.

[53] Burke EK, Hellier RSR, Kendall G, Whitwell G, "Complete and robust no-fit polygon generation for the irregular stock cutting problem", *European Journal of Operational Research*, 179(1), pp. 27-49, 2007.

[54] Stoyan, Y.G., Terno, J., Scheithauer, G., Gil, N., Romanova, T., "Phi-functions for primary 2D objects", *Studia Informatica Universalis*, 2(1), pp. 1-32, 2001.

[55] Glover, F., "Heuristics for Integer Programming using Surrogate Constraints", *Decisions Science*, 8, pp. 156-166, 1977.

[56] D. Domovi, T. Rolich, D. Grundler and S. Bogovi, "Algorithms for 2D Nesting Problem Based on the No-Fit Polygon", *MIPRO 2014, 26-30 May 2014*, Opatija, Croatia, 2014

[57] Art, R.C., "An approach to the two dimensional irregular cutting stock problem", *Technical Report 36. Y08; IBM Cambridge Scientific Center*,Cambridge, Massachusetts, USA 1966.

[58] Oliveira, J.F., Gomes, A.M., Ferreira, J.S.. TOPOS, "A new constructive algorithm for nesting problems", *OR Spektrum*,22(2), pp. 263-284, 2000.

[59] Dowsland, K.A., Vaid, S., Dowsland, W.B, "An algorithm for polygon placement using a bottom-left strategy", *European Journal of Operational Research*,141(2), pp. 371-381, 2002.

[60] Gomes, A.M., Oliveira, J.F, "A 2-exchange heuristic for nesting problems.", *European Journal of Operational Research*,141(2), pp. 359-370, 2002.

[61] Metropolis, N., Rosenbluth, A., Rosenbluth, M.N., Teller, E., Teller, E., "Equation of state calculations by fast computing machines.", *Journal of Chemical Physics*, 21, pp. 1087-1092, 1953.

[62] Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P, "Optimization by simulated annealing", *Science*, 220, pp. 671-680, 1983.

[63] Zicheng Wang, Xiutang Geng, Zehui Shao, "An Effective Simulated Annealing Algorithm for Solving the Traveling Salesmen Problems", *Journal of Computational and Theoretical Nanoscience*, Vol 6, pp. 1680-1686, 2009.

[64] Fraser, A.S, "Simulation of genetic systems by automatic digital computers", *Australian Journal of Biological Science*, 10, pp. 492-499, 1957.

[65] Bremermann, H.J, "The Evolution of Intelligence: The Nervous System as a Model of its Environment, Technical Report, Technical Report No 1, Contract No. 477(17)", *Department of Mathematics, University of Washington Seattle*, 1958.

[66] `https://www.towardsdatascience.com/introduction-to-genetic-algorithm`

[67] Ralph Gomory, "A Linear Programming Approach to Cutting Stock Problem", *Article in Operations Research*, 1963.

[68] Yang, X. S., & Deb, S, "Cuckoo search via Levy flights", *World congress on nature & biologically inspired computing (NaBIC). IEEE Publications*, pp. 210-214, 2009.

[69] `https://www.slideshare.net/AnujaJoshi6/cuckoo-o-ptimization-ppt`

[70] Ahmed Alkeran", "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering", *European Journal of Operational Research*, 231, pp. 757-769, 2013.

[71] Euro Special Interest Group on Cutting and Packing (ESICUP)", *https://paginas.fe.up.pt/ esicup/datasets*.

[72] Burke EK, Hellier RSR, Kendall G, Whitwell G", "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem.", *Operations Research*, 54(3), pp. 587-601, 2006.

[73] Xiao L, Jia WY", "Heuristic algorithm based on the principle of minimum total potential energy (HAPE): a new algorithm for nesting problems.", *Applied and Engineering Physics*, 12(11), pp. 860-872, 2011.