University of Jordan

Faculty of Graduate Studies

Application of Artificial Intelligence Techniques to Two Dimensional Material Cutting

By

Rami Mustafa El-Naqa

Supervisor

Dr. Yousef Al-Assaf

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Industrial Engineering/ Design & Manufacturing, Faculty of Graduate Studies, University of Jordan.

July, 1996

This thesis was defended successfully on the 31st of July, 1996

Committee Members	Signature

1. Dr. Yousef Al-Assaf
   Industrial Engineering Department
   University of Jordan

2. Dr. Khaled Toukan
   Industrial Engineering Department
   University of Jordan

   Khaled Toukan

3. Prof. Dr. Mohamed Khader
   Electrical Engineering Department
   University of Jordan

4. Dr. Munib Qutaishat
   Computer Science Department
   University of Jordan

5. Dr. Rawshan Basha
   Computer Science Department
   Al Al-Bayt University

Dedication

Dedicated to my loving parents

and

to my dear brothers

Acknowledgment

I would like to express my deepest gratitude to my supervisor Dr. Yousef Al-Assaf for his encouragement, guidance and support during this research. It has been a pleasure working with him.

I thank Beta Industries for Rotary Screw Compressors for their financial & technical support for the practical part of the project.

Many thanks are due to El-Zay Ready Wear Manufacturing Co. They were of great help, and I do acknowledge them for the useful information they provided, and for the fruitful discussions on the practical aspects of the Layout Problem in textile industry.

Table of Contents

List of Tables

List of Figures

Abbreviations

CNC	Computer Numerical Control.
CSP	Cutting Stock Problem.
BOM	Bill Of Material.
NP-Complete	Non-deterministic Polynomial in time.
FFDPL	First Fit Decreasing Length Perpendicular.
FFD	First Fit Decreasing.
BF	Best Fit fill.
BE	Best End fill.
S	Shift pieces.
M	Merge gaps.
L	Largest Rectangle.
P.D	Packing Density.
BLU	Basic Length Unit.
MER	Minimum Enclosing Rectangle.
CH	Convex Hull.
MCH	Minimum Convex Hull.
CCW	Counter Clock Wise.
PC	Personal Computer.
RAM	Random Access Memory.
I/O	Input/Output.
DAC	Digital to Analog Conversion.
VDC	Direct Current Voltage.
DC	Direct Current.
DXF	Data eXchange File.
GPC	Generalized Predictive Control.

Abstract

Application of Artificial Intelligence Techniques to Two Dimensional Material Cutting

By

Rami Mustafa El-Naqa

Supervisor

Dr. Yousef Al-Assaf

Cutting variety of two dimensional shapes from plane sheets or plates of raw material is usually preceded by a *nesting* or *allocation* process, where a decision should be made on how to distribute a batch of pieces to be cut on the raw material, such that maximum utilization of the resource is obtained. This problem exists in various industries. The problem has been traditionally solved by humans in a manual way. Automatic approaches have been suggested to solve such a problem; however, humans are still believed to be better in solving it.

In this research, two cases of this problem have been tackled: the rectangular shapes cutting stock problem, and the irregular shapes cutting stock problem. Automatic solution approaches have been proposed to solve them. For the first problem, a *heuristic* approach based on human intuitive thoughts has been suggested. The experimental results obtained show its success in obtaining adequate layouts without any manual human intervention. For the second problem, a *state space representation* is suggested, then a fast solution approach based on a *Hill Climbing* search technique was investigated. Initial results obtained demonstrate that this method is promising; however, more research has to be performed to enhance its capabilities.

The suggested automatic solutions for these problems are intended to be a part of a decision supporting system, for a *flame cutting machine*, for sheet metal industry. A design for that machine is also presented in this research.

Chapter 1

Introduction

Flame cutting is one of the manufacturing processes that is being utilized extensively in Jordan in cutting metal sheets and plates. Unfortunately, in many industries the process is being conducted in a primitive manual way. A few number of semiautomatic machines are found at some local companies and institutions; nevertheless, they show low productivity and suffer many problems e.g. they don't work on batches of shapes provided as complete layouts, but rather they work on single piece layout then cutting it in a sequential manner. Also they are preset at constant motion speed, which causes bad curved shape cuts. Although automated flame cutting machines based on computer numerical control (CNC) technology [1] have long existed in the industrialized world, their local existence is scarce, the main reason for that being the relatively high initial and running costs for such a technology.

Considerable changes have been taking place in manufacturing systems, due to the application of new computerized technologies and techniques. The computer has had and continues to have a dramatic impact on the improvement of industrial productivity and quality. Therefore, this project is aimed at building up more practical local experience, and gaining the know-how in such new technologies, through the construction of a prototype automatic flame cutting machine -in cooperation with other colleagues- based on a microcomputer system, and supporting this machine with a computer aided *nesting* software package, which is the main focus of this thesis.

Cutting a variety of two dimensional shapes from plane sheets or plates of raw material is usually preceded by a *nesting* or *allocation* process, where a decision should be made on how to distribute the batch of pieces to be cut on the raw material, such that maximum utilization of the resource is obtained, or equivalently, minimum waste is produced. Figure 1.1 illustrates this problem in the context of the manufacturing system. In literature, this problem is also known as the Cutting Stock Problem (CSP). Figure 1.2 shows typical two-dimensional shapes nesting.

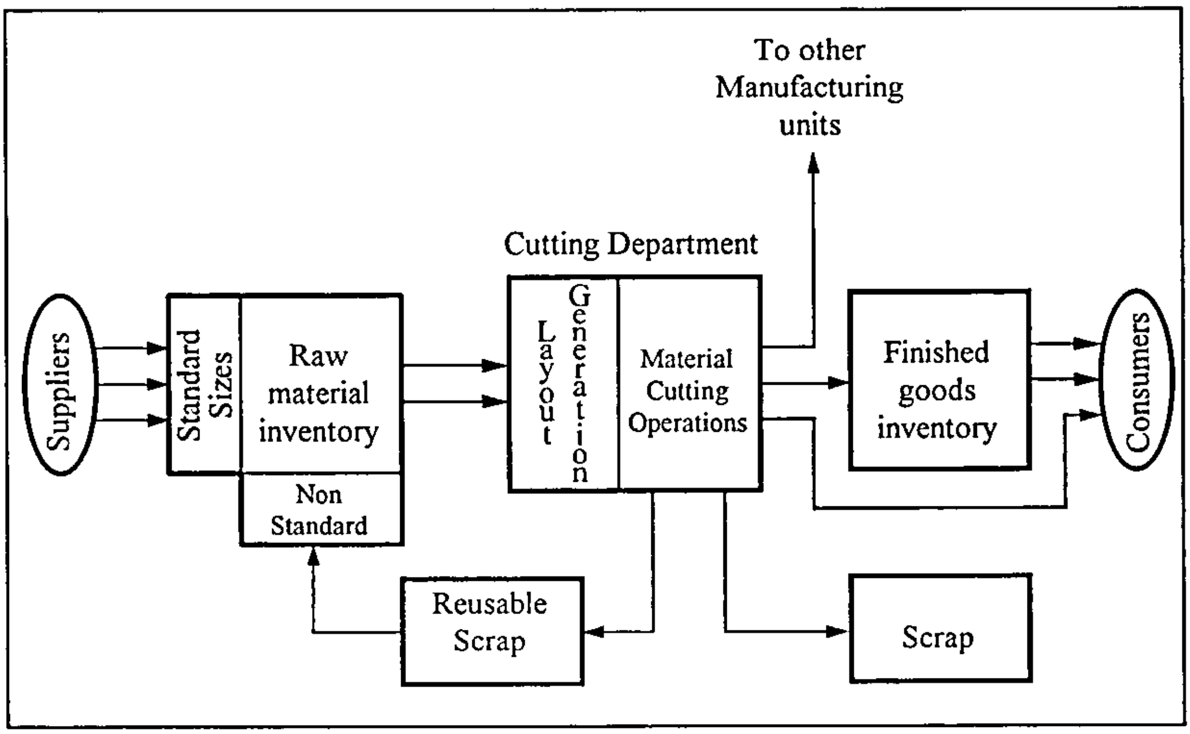


Figure 1.1 Layout problem placed in manufacturing system context.

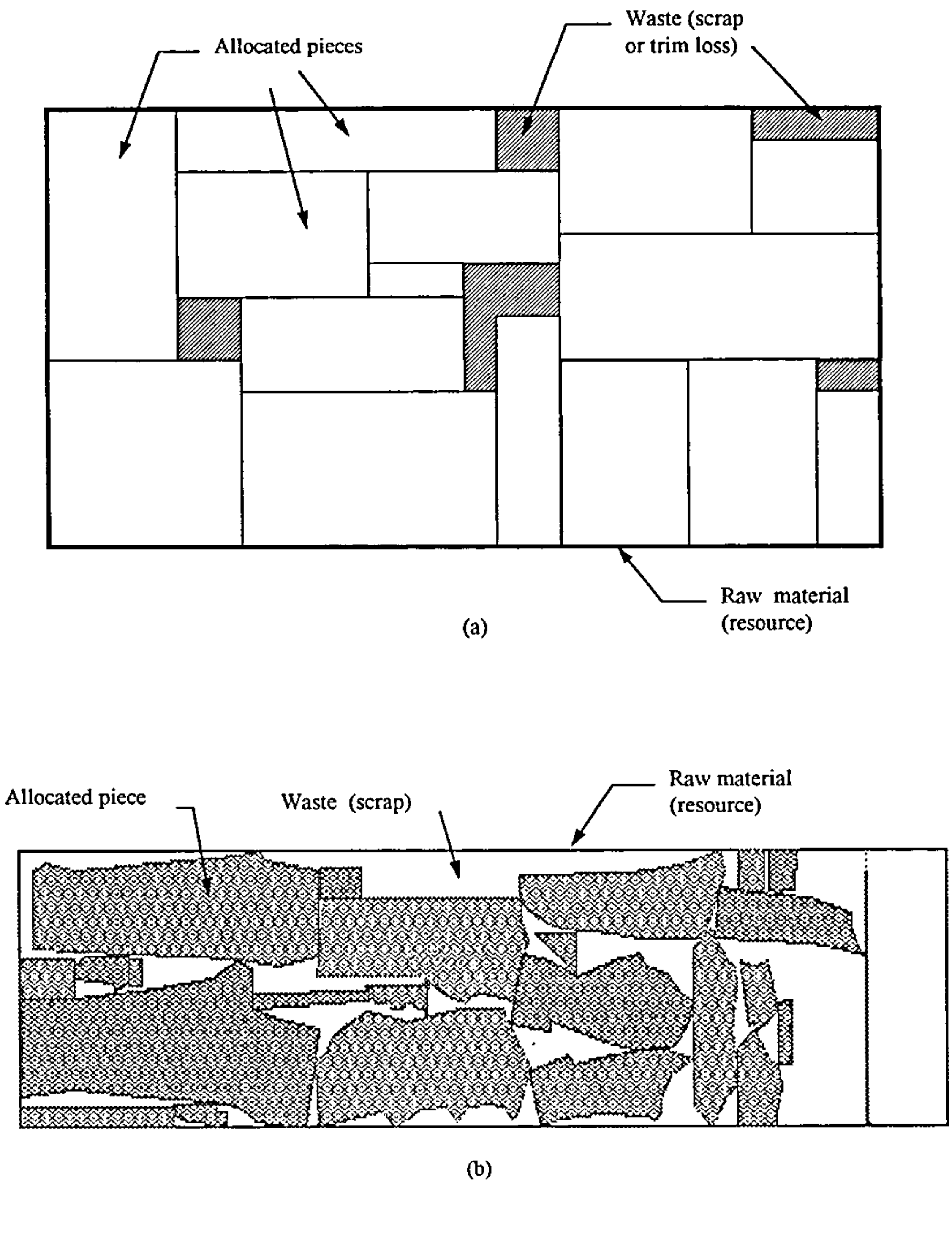


Figure 1.2 Typical two dimensional shapes nesting.
a) Rectangular shapes layout. b) Irregular shapes layout.

1.1 Cutting stock problem

Proper utilization of raw materials by minimizing trim losses (cutting waste) is one of the main objectives in various industries. It is a major problem in sheet metal, textile, furniture, leather, paper, and flat glass industries, just to mention a few areas of application. To achieve "optimal" resource utilization effective allocation (layout) of a set of demanded two dimensional shapes (rectangular and /or irregular) onto a relatively larger stock sheet should be carried out. This problem is one of the problems that are easy to state, but very difficult to solve. Figure 1.3 shows a general classification for the CSP and its variants.

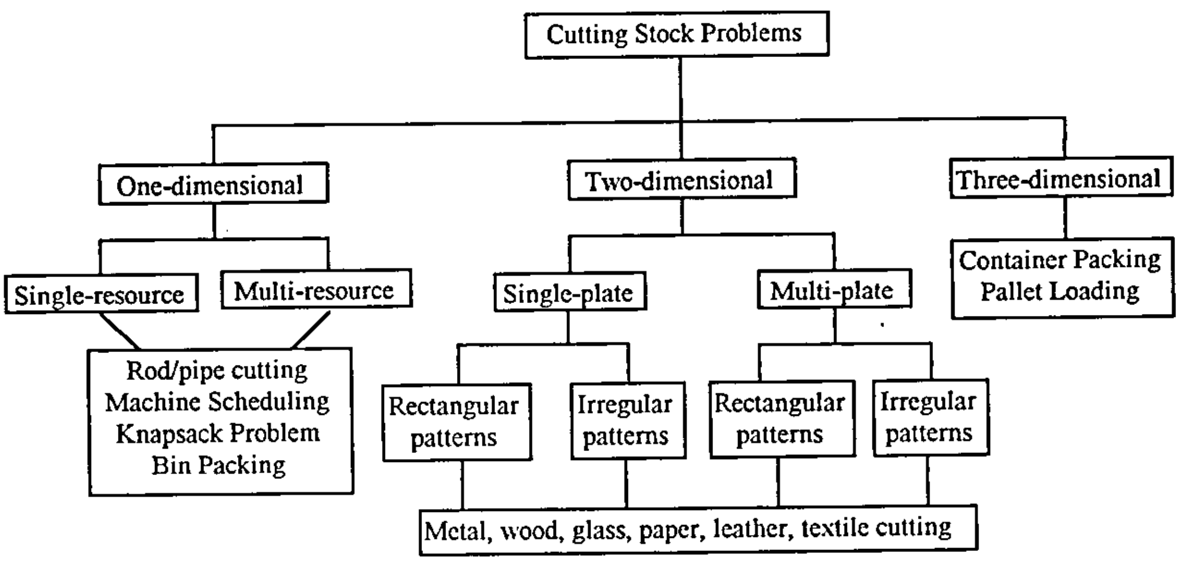


Figure 1.3 Classification of cutting stock problems [2] (with modifications).

Traditionally, such allocation of shapes is performed by humans. In recent years, several layout algorithms have been developed to automate the solution of the problem. However, a comprehensive allocation algorithm which is suitable for various applications is not easily achievable, since each industry has its own special allocation restrictions and patterns [3]. Even for a single application, the problem is often considered to be difficult (intractable) [4]. The major source of difficulty in this type of problems is due to the very large number of feasible cutting patterns that could be generated in the allocation process [3,4,5].

The problem has been classified as a large scale combinatorial optimization problem [5], and has been proven to be NP-Complete [6] (see Figure 1.4) i.e. it has no polynomial time exact solution algorithm (one which guarantees an optimum solution, and requires a number of steps that grows as a polynomial in the size of the input), and it is unlikely, if not impossible to solve its general case by an exact solution approach. Therefore, one should settle for goals that are less ambitious than developing an algorithm that always finds an exact solution, and has time requirements that never exceed a given polynomial growth [7].

Heuristic methods, which are typical in artificial intelligence and operations research are well suited to efficiently solve such problems, since they are based on rules of thumb, and human intuitive thoughts that have proven to be successful in solving combinatorial problems in practice. They tend to simplify, reduce, or limit the search in large problem spaces; in particular, they search for solutions in domains that are difficult and poorly understood. For these reasons interest in the cutting stock problem has been shifting from a purely mathematical and theoretical basis to more practical and heuristically based solution approaches [8].

It is worth mentioning also that even though a number of nesting algorithms are currently in use in industry, virtually all of these systems are considered to be highly proprietary and specific [8,9].

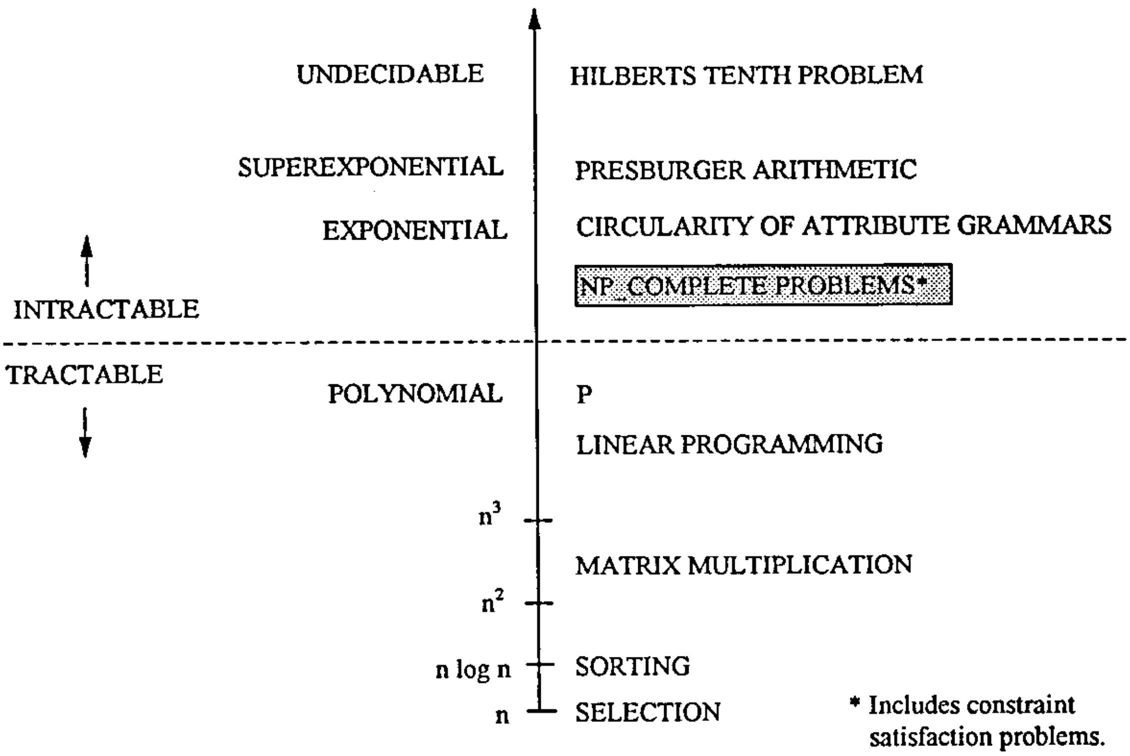


Figure 1.4 The spectrum of computational complexity [10].
Where n is the size of entities in the problem.

1.2 Problems addressed by this thesis

In addition to the participation in designing and constructing a prototype automatic flame cutting machine, two cases of the cutting stock problem are tackled by this research. They are related to local sheet metal and textile industries primarily, but could be improved upon to suit other applications. The statements of the two instances to be addressed by this thesis are as follows :

1) Given a batch of different two dimensional rectangular shapes, it is required to allocate them on a number of standard sized rectangular raw material sheets of

the same size; such that the waste is minimized, or equivalently, the number of raw material sheets is reduced.

2) Given a batch of two dimensional irregular shapes, it is required to allocate them on a single finite width, infinite length rectangular raw material sheet, such that the waste is minimized (The assumption of infinite length comes from the fact that the resource considered is usually a roll, a strip, or a sheet with length that is much more greater than the lengths of pieces to be allocated).

1.3 Overview of the thesis

The thesis is organized as follows: In chapter two, the rectangular shapes layout problem is reviewed. Then a heuristic solution approach based on capturing human intuitive thoughts in nesting problems is proposed. After that, randomly generated batches of rectangular shapes with certain characteristics -to be discussed in chapter two- are used to test the approach. A comparison with already published results in literature is then performed. Finally, the results are presented and discussed with recommendations for future improvements.

Chapter three is devoted to the irregular shapes nesting problem, where it is also reviewed, and a classification of already developed approaches is presented and commented upon. Then a state space representation of the problem solution is suggested, with an attempt to asses its complexity, along with a proposed heuristic search scheme. A number of sample batches of irregular shapes obtained from local industries is then used to test the solution approach. Finally, results are presented and discussed.

In Chapter four, the intended prototype flame cutting machine design is presented with discussion of different considerations taken into account during its design and construction.

Chapter five is devoted for general conclusions and suggestions for possible future improvements and extensions.

Chapter 2

Two Dimensional Rectangular Shapes Allocation

456516

2.1 Introduction

The two dimensional cutting stock problem requires cutting a plane rectangular resource into smaller rectangular pieces of given sizes and quantities, as specified by a given bill of material (BOM). A value could be associated with each piece; usually taken to be proportional to its area, thereby the problem of waste minimization could be formulated as maximizing the value of pieces cut or allocated [11]. Thus a mathematical formulation of the problem could be stated as follows [8]: given a rectangular stock sheet *R* of dimensions *L x W*; a set of *p* distinct types of smaller rectangular pieces is to be cut from (or allocated to) the stock sheet. This set is denoted by $S=\{(l_1,w_1),(l_2,w_2),\ldots,(l_p,w_p)\}$, where l_i and w_i represent the length and width respectively of a given rectangular type . Any number (or non at all) of one type of piece may be cut from *R*. The general formulation is given as follows:

The objective is to maximize *z*: $z = \sum_{i=1}^{p} x_i v_i$ (1.1)

Such that: 1) A number of geometrical and topological constraints is satisfied. (1.2)

2) A number of demand constraints is satisfied. (1.3)

3) $x_i \geq 0$ and x_i is integer. (1.4)

Where v_i is the "value" (often, the area) associated with each of the rectangles i to be cut, and x_i is the decision variable representing the number of pieces of each type i.

The main sources of difficulty in solving such a mathematical program arises from the fact that a very large number of feasible patterns could be obtained, which is reflected in a very large number of columns in the mathematical model. The other source of difficulty is in expressing the geometrical and topological constraints, where nonlinearities are introduced into the problem, which are not easily or satisfactorily overcome by classical optimization techniques[3,5,8, 12,13,14].

2.2 Literature review

Surveying the literature, various approaches were proposed to define, model and solve the problem with variations on the basic mathematical model presented by the formulation of Equation (1.1) to Equation (1.4) [15]. Such techniques include restricted mathematical optimization [4], tree search [11,16], heuristics [8, 15], neural networks [3], and recently simulated annealing [17].

Due to the fact that this problem is a large scale combinatorial problem, as well as being NP-Complete [6], simplifying assumptions and restrictions were imposed in order to make it possible to formulate the problem, and thereby make it amenable to mathematical solution within practically acceptable computational time and computer storage requirements.

Some of the commonly used assumptions are:

-the *guillotine** patterns cut (see Figure 2.1), for which, most of the effort has been directed in literature [4, 11, 18, 19, 20],

-limited number of patterns variety [20],

-and constraint on the maximum number of each type in a pattern [11, 14], where a pattern in this context is considered to be a complete layout on a single resource; leading to a feasible complete solution or nest.

Some of the above assumptions could be accepted in practice, while others impose real limitations to the applicability of the used optimization technique, and the quality of the solution obtained. In addition, such approaches require excessive, and some times prohibitive, amount of computing [18], especially for medium to large scale problems (problems with roughly more than 10 different sizes of rectangles).

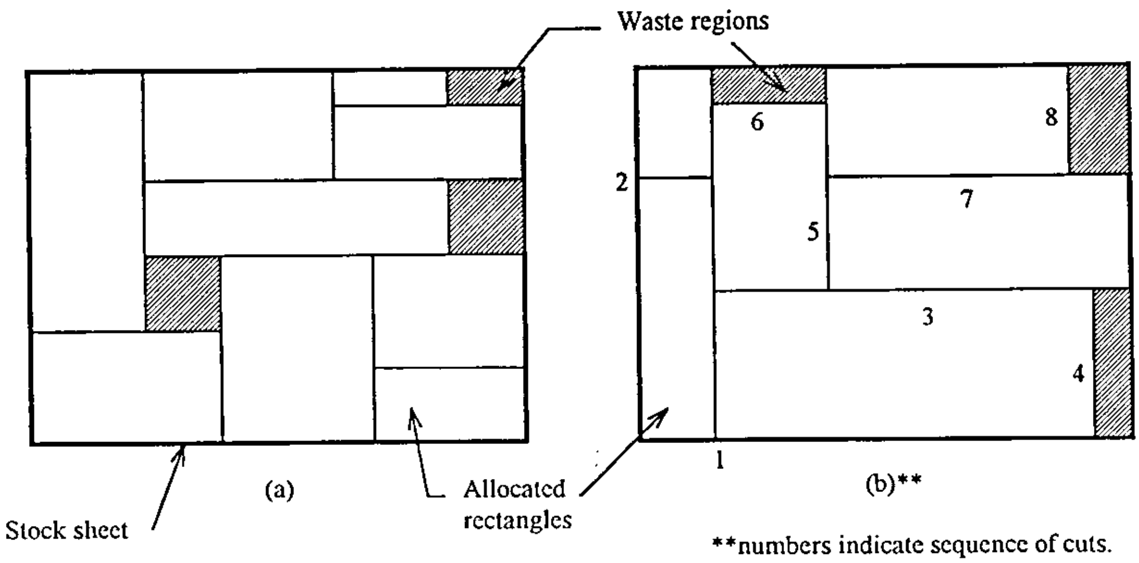


Figure 2.1 Allocation/Cutting patterns: a) Non-Guillotine pattern b) Guillotine pattern.

* A guillotine cutting pattern means that only edge-to-edge cuts of the raw material are allowed.

Tree-search and related algorithms were alternative approaches, that have been used to tackle the allocation problem. However, to make the two-dimensional problem amenable to efficient solution by these algorithms the size of tree search is limited by deriving and imposing necessary conditions for the cutting pattern to be optimal, as well as including designed node evaluation functions to give bounds for driving the search [11, 21]. Such imposed conditions and bounds reflect negatively on the optimality of the solution.

Achieving close to optimal solutions with reasonable computing time is the objective of numerous heuristic approaches reported [8, 14,17, 18, 22, 23]. Various heuristics were adopted to tackle the two-dimensional allocation problem, such as: Bottom-Left heuristic[24], Decreasing Length Perpendicular Strip Packing [8], the Bin Packing heuristics [5,8] just to mention a few. As with all allocation algorithms heuristics schemes are very dependent on the particular problem being solved, hence the allocation mechanism used must be derived from the problem environment. Even though heuristics do not guarantee the optimality of the solution obtained, they have proved to be a widely acceptable alternative to mathematical optimization. Furthermore, a trend has been taking place, since the seventies, in trading off optimality for faster algorithms providing near optimal solutions[15]. Also, it is worth mentioning that multiple-objectives can be incorporated within such heuristic techniques[14].

Human intervention, even if limited, has demonstrated that it had improved the quality of some of the above mentioned solution approaches substantially [15]. However, although human intervention has demonstrated improvements, the efficiency and time required for such intervention depends on the skill of the operator, the size of the problem (and thereby the number of sheets to be cut), as well as the characteristics (features & variety) of BOM [25].

Kopardekar et.al.[26] focused on identifying human intuitive thought process in laying out of irregular parts. In that work, instead of adopting the human intervention to improve the outcome of a basic heuristic scheme, the human strategies of allocating the whole sheet were suggested to constitute a heuristic approach. However, human strategies in allocating the whole sheet may not be effective or easy to comprehend if BOM characteristics are wide. Hence to overcome such a hurdle, an underlying allocation algorithm could be used, then human strategies could be applied to continue the allocation process.

2.3 Proposed solution approach

Human operators are -until this time- believed to perform -in most of the cases- better than pure automatic nesting algorithms [8, 26, 27]; although, their performance degrades as the size and variety of the problem increases. Therefore, the identification and computer implementation of various human strategies in filling spaces, generated from an underlying automatic allocation algorithm is the proposed heuristic approach to enhance the solution of the two dimensional rectangular layout problem in this thesis, whereby this approach direct human intervention is eliminated but not his/her intuitive thoughts. Heuristic and non-heuristic algorithms could be adopted as an underlying allocation technique; however, the First Fit Decreasing Length Perpendicular (FFDLP) heuristic [15] is used here, because of its simplicity, and for comparison with already published results.

2.4 Problem constraints and requirements

The pattern allocation process considered here is related to sheet metal industry, in which a standard-size rectangular stock sheet is available with unlimited number; is to be cut into demanded rectangles of smaller sizes as specified by a given BOM.

The following constraints and specifications are to be taken into account:

1- The cuts to be performed are not restricted by the guillotine constraint (since the flame cutting machine is a flexible cutting facility), and they should be *orthogonal* (better utilization is expected from orthogonal cuts as Figure 2.2 shows [28]).

2- The orientation of a piece is not fixed, it could be rotated 90° , which preserves orthogonality constraint.

3- The pieces to be allocated should not overlap.

4- The pieces should be entirely packed within the stock sheet boundaries.

5- The demand of each type of piece should be met exactly. No over runs or under runs are allowed.

6- Since the cutting flame has a significant diameter, there must exist a specified cutting tolerance (offset) between pieces, usually known as the cutter diameter compensation. To meet this requirement, pieces to be allocated are enlarged in size in all directions by an amount equal to half the diameter of the flame nozzle (to be used in cutting).

7- The pieces to be cut from the stock sheets, should be allocated in such a way that would produce minimum waste, or equivalently reduce the number of needed stock sheets. The work is directed towards relatively large scaled problems with relatively large pieces variety, since its effectiveness becomes more feasible.

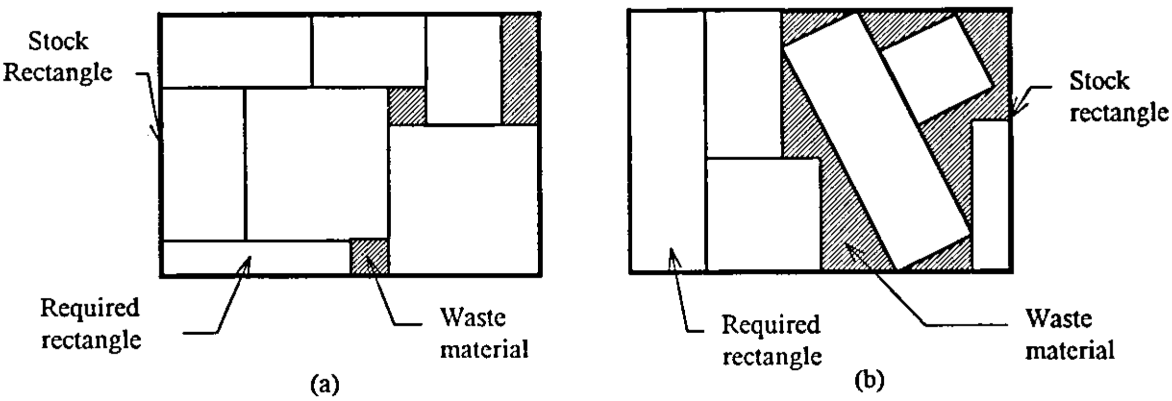


Figure 2.2 (a) Orthogonal cutting pattern. (b) Non-Orthogonal cutting pattern [28].

2.5 Human reasoning in space recognition and filling

In this section various techniques, that are representative among others to the way a human operator would recognize and fill confined spaces, which are generated after applying an automatic allocation algorithm (like the FFDLP) are presented. These techniques are gathered from literature and from interviews made with human operators. It is the intention to combine the computer's accuracy and speed of calculation with such heuristic rules to get a better solution approach that takes advantage of both the human thoughts and the computer capabilities.

1. *First-fit decreasing(FFD)* [5]: It is the very simple bin-packing heuristic that deals with the available gaps as separate bins or isolated subspaces (a gap or subspace (see Figure 2.3) is defined as the part of the sheet that is unallocated yet, and is adjacent to an allocated piece, such that the dimensions of this gap are confined to the adjacent allocated piece and within the allocation cycle region -to be explained later-), and works on filling them with the first piece in the BOM that fits in the space as illustrated in Figure 2.3. Obviously for any skilled or 'intelligent" operator this approach is hardly used since the first piece which can be fitted may not be the best (optimal) piece in filling the considered space. However, it represents a simple and fast, but short sighted rule to fill the gaps. It works well in one dimensional large scaled problems. It is included here to show the gradual increase in depth of thought about the problem.

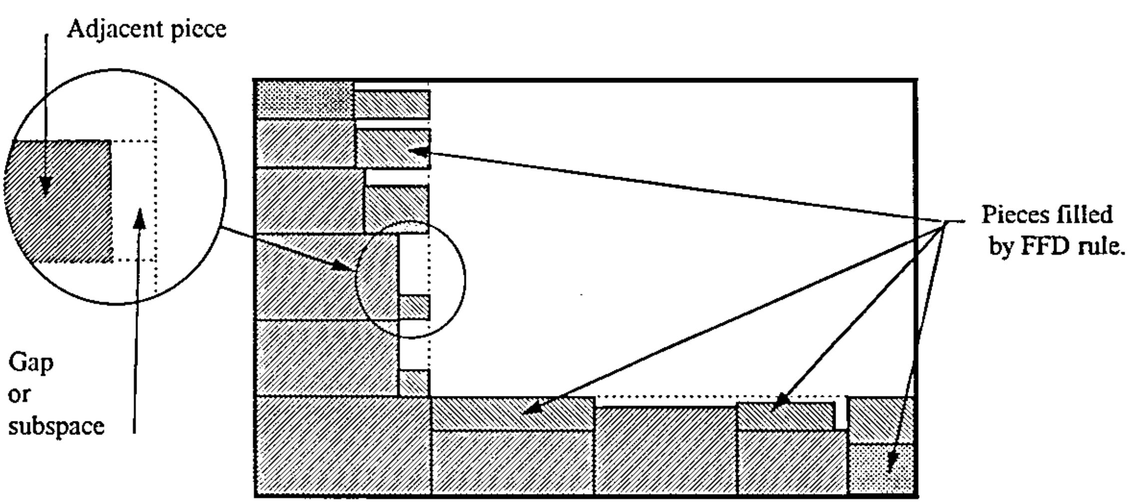


Figure 2.3 First-Fit Filling of end-of-cycle subregions and gaps.

2. *Best-fit decreasing(BF)*: A human thought would naturally suggest such strategy in filling of gaps, where for a given space a human operator would look for a piece with the largest area that would cover as much as possible of a gap, such that minimum waste is produced. Such a rule is considered to be one of the greedy approaches [29]. In implementing such heuristic rule, a 90° rotation of a piece is also permitted to test for best filling, and thus provide the chance for better results as shown in Figure 2.4.

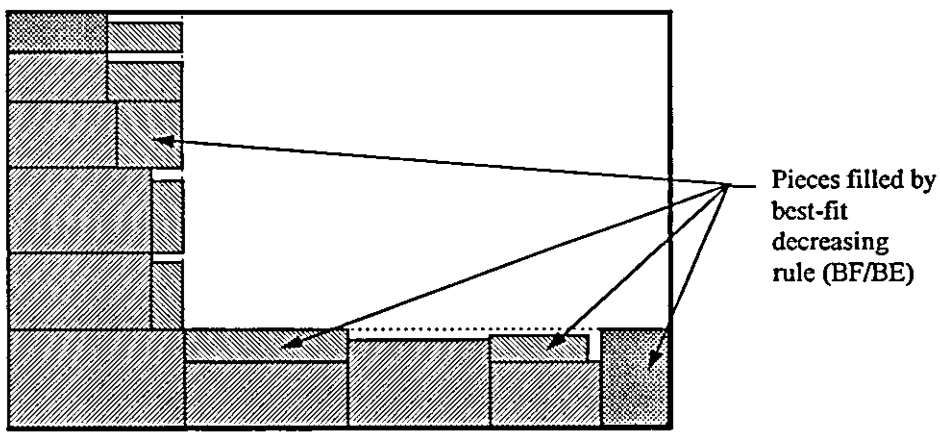
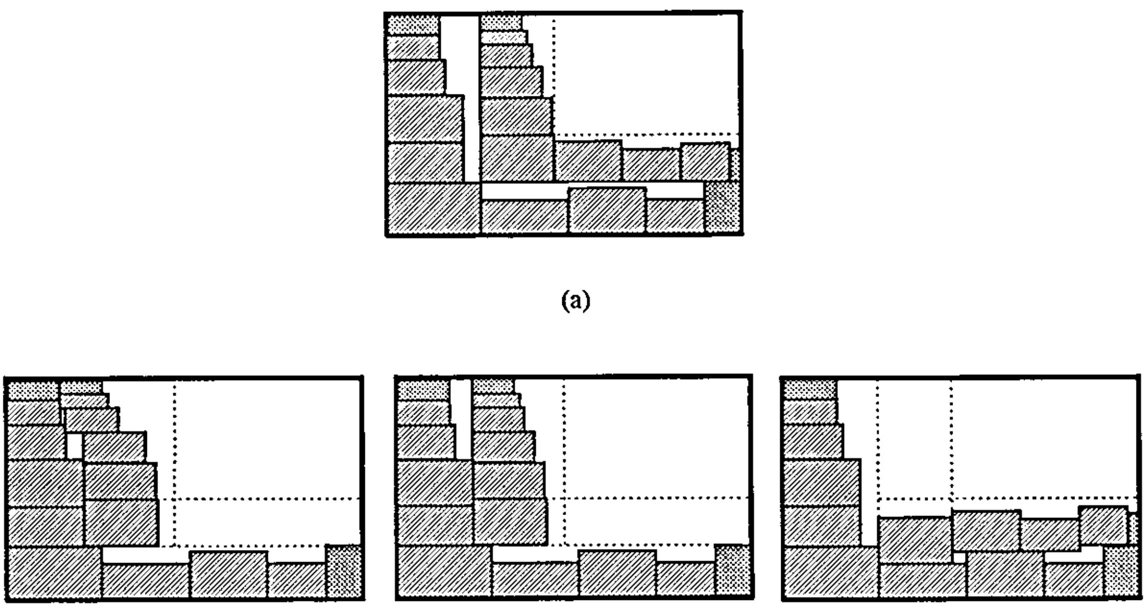


Figure 2.4 Best-Fit filling of end-of-cycle subregion and gaps.

3. *Shift(S)*: Different types of shifts could be considered by a human operator (see Figure 2.5); however, the one shown in Figure 2.6 is the implemented type, whereby in this approach the gap spaces generated from previous allocations of pieces are reduced, and the recursive nature of the whole algorithm -as will be presented later- is preserved. This restricted type of shift is applied when the set of pieces on one edge of the sheet -excluding the corner piece- are less in their dimension than the corner piece.

4. *Merging of gaps into larger rectangles(M)*: By this approach the human ability to recognize contiguous space is realized and implemented in a simplified manner, where spaces sharing at least one common dimension are joined together, then the merged spaces are reconsidered for filling by rule (1) or (2) as Figure 2.7 shows.



(a)

(b)

Figure 2.5 Different possible types of shift.
a) Before applying shift. b) After applying some of the different possible types of shift.

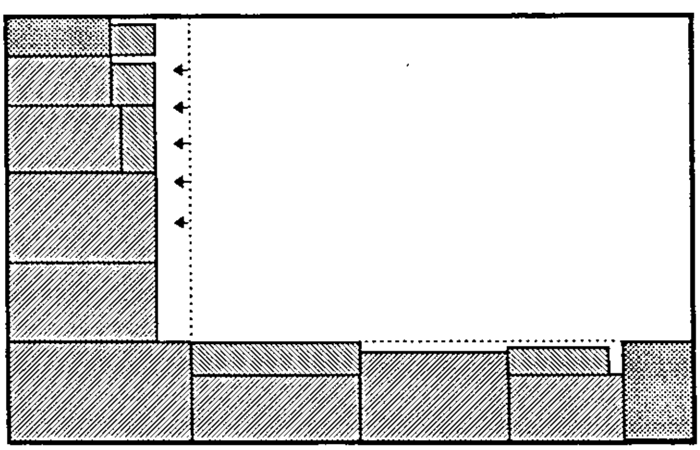


Figure 2.6 Shift, then Best-Fit filling of end-of-cycle subregions and gaps.

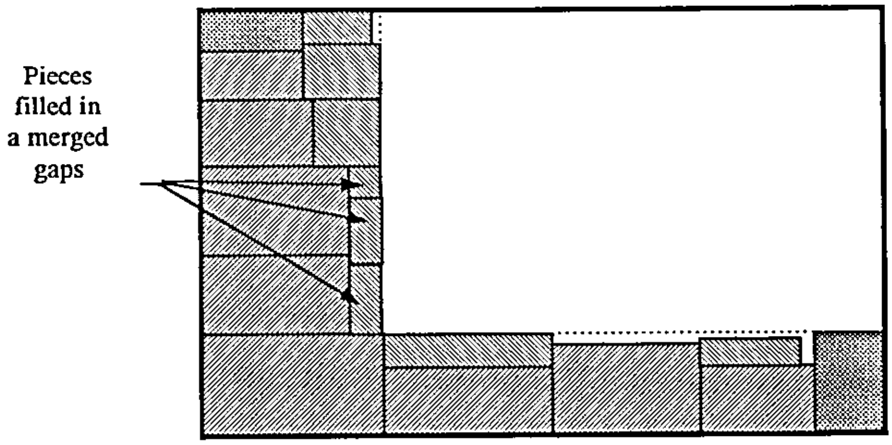


Figure 2.7 Merge-gaps, then Best-Fit filling of end-of-cycle subregions and gaps.

5. *Largest rectangle(L)*: The largest rectangular space that could be realized in an irregularly bounded unfilled space is searched for by this heuristic, and considered for allocation with largest piece, then the remaining spaces around it are reconsidered for filling by other heuristics.

6. *Strips*: When a human operator recognizes at least one common dimension between pieces he/she tries to join them together and then allocate them as a single unit (a strip) of two or more pieces as shown in Figure 2.8. This approach has the benefit of keeping edges of pieces and spaces aligned together which introduces some homogeneity to edges and thereby has potential for reduced waste.

7. *Trial and error:* It is the most natural and probably least efficient approach; especially for medium to large sized problems. Pieces are seemingly randomly selected and checked for legal placement, or combinations of pieces could be tried. Actually, the human operator makes some fast assessments (recognition)

of size and shape before such trials are made, which in most cases reduces the number of possibilities, and yield "good" solutions.

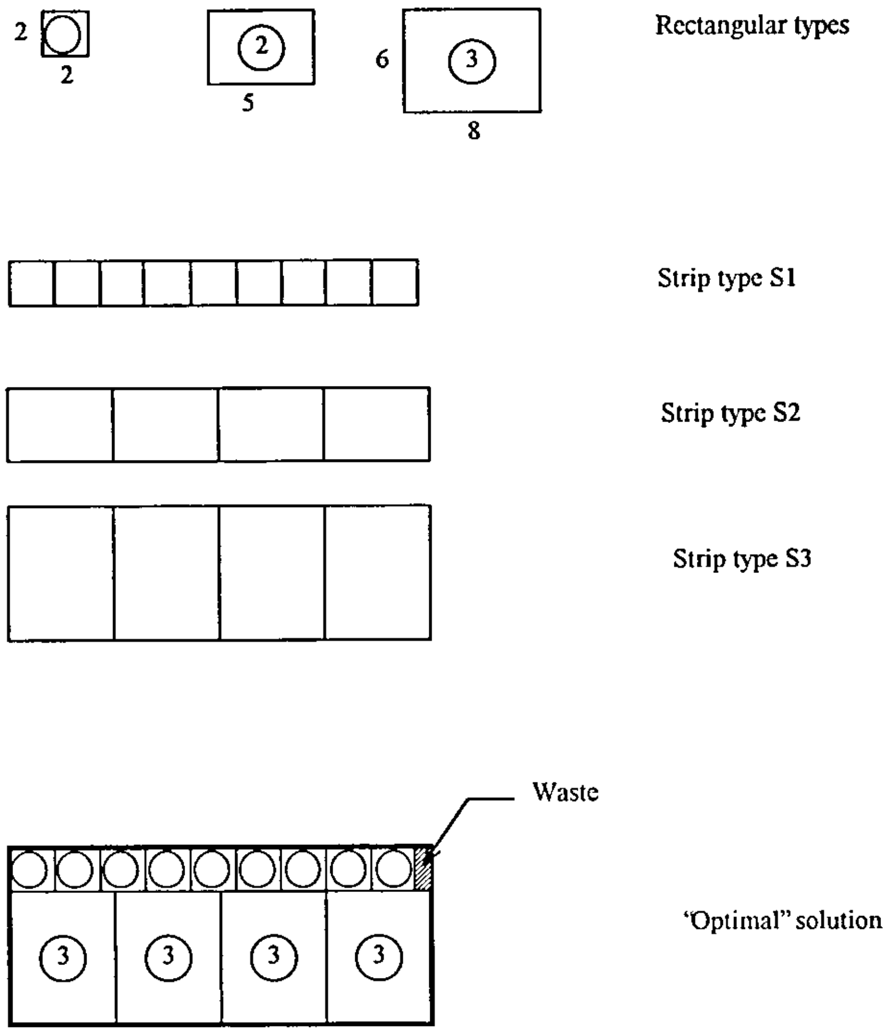


Figure 2.8 Strip layout example [30].

These techniques are probably not the only ones used by human operators in various industries; furthermore, not all of them are easy or suitable for computer implementation. Therefore, some of them have been programmed and tried in this work with the following considerations:

-Combinations of the previous heuristics could also be used to take advantage of different characteristics of each, such as applying the shift strategy at some position then trying to fill the produced gap by rule 2, or the shift could be applied, then merging of spaces is considered, and after that the gaps are filled with best-fit decreasing as shown in Figure 2.9.

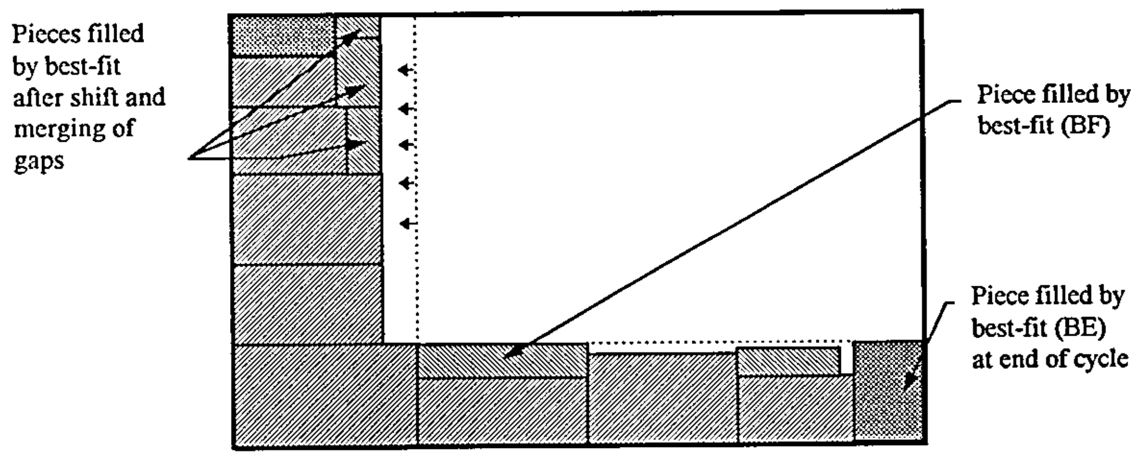


Figure 2.9 Shift, Merge-gaps, then Best-Fit filling of end-of-cycle subregions & gaps.

-Sorting the BOM according to non-increasing length or width and starting with larger pieces in terms of area, length or width is also considered as a pre-allocation process on the BOM, with the aim of consuming larger pieces (relative to the stock sheet) at earlier stages of allocation, and preserving smaller ones for filling of possibly generated gaps and reduced sheet area at later stages.

-Deciding upon the rule (or combination of rules) to be applied at specific points of allocation, based on different characteristics of the original BOM or the remaining pieces to be allocated in conjunction with the space to be filled, could also be implemented. This decision making process which could represent an inference engine of the approach is not considered in this work; however, heuristics or their combinations to be used in a specific allocation instance are left to the operator to be pre-selected before the allocation starts.

2.6 Sheet Conceptual divisions

In the presented algorithm the stock sheet to be filled is divided conceptually into four types of regions, on which different heuristic rules will be applicable. As can be seen from Figure 2.10, the four subregions and the currently tested heuristics related to them are as follows:

1) *Main-space*: The half perimeter of a sheet as suggested by Israni & Sanders[8] to be filled with the First-Fit Decreasing heuristic, where pieces are alternately allocated on the length and width to fill half the perimeter of the sheet.

2) *End-of-cycle space*: To be filled by the First-Fit or Best-Fit heuristics.

3) *Gaps between cycles*: To be filled by a selected heuristic or combination of heuristics from the following set: 1) First-fit-decreasing, 2) Best-fit-decreasing, 3) Shift, 4) Merge spaces, 5) Largest rectangle, 6) Strips.

4) *Reduced sheet*: Where the divisions above are reconsidered in a recursive manner.

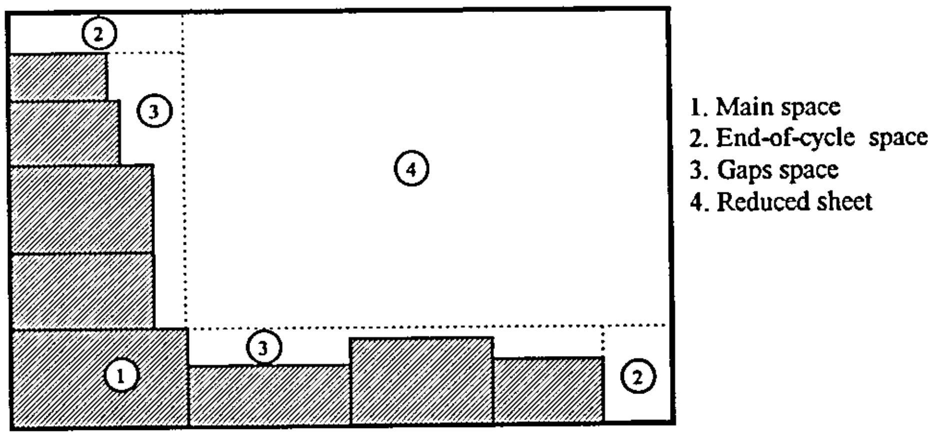


Figure 2.10 Sheet space division into four subregions.

2.7 The proposed solution algorithm

The flowchart of Figure 2.11 represents the major steps of the proposed algorithm which are stated as follows:

1- The FFDLP heuristic is applied to the main space, which is detailed as follows:

a. The original BOM is sorted by decreasing length of shapes.

b. The ordered BOM is then allocated onto the stock rectangle R along its half-perimeter, starting at the bottom left corner.

c. The pieces are allocated alternately along the length and width of R. If the space left along either dimension is too small for the next piece in the sorted BOM, an end-of-cycle is reached.

2. At the end of cycle a pre-selected heuristic, which in this case is either the first-fit decreasing, or the best fit decreasing is used to fill the end-of-cycle spaces.

3. Gaps generated between cycles are then considered for filling by one or more of the previously suggested heuristics.

4. At the end of the previous step, the sheet is reduced in size and the above steps from 1 to 3 are repeated in a recursive manner until the BOM becomes empty or the sheet is completely packed, where at that time a new sheet is considered for allocation.

5. Any left gaps or spaces are then considered to be waste.

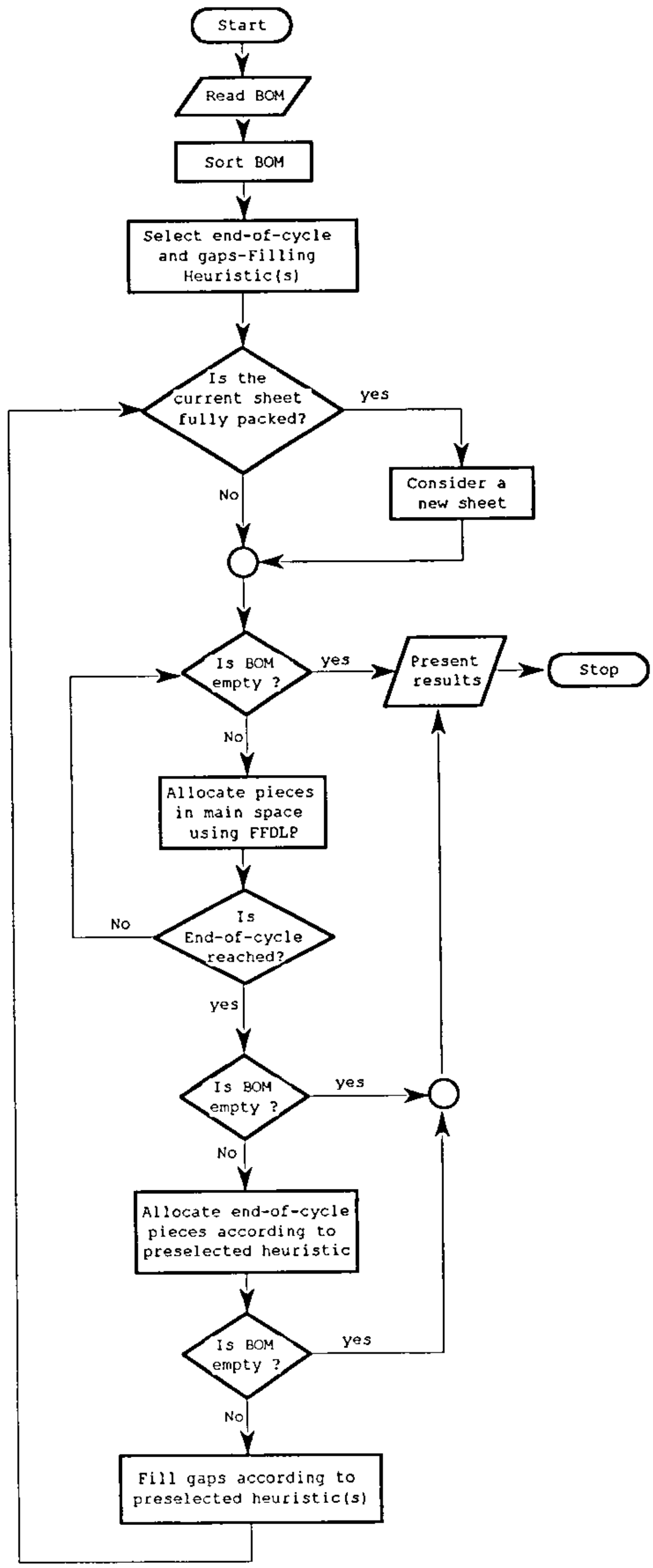


Figure 2.11 Two-dimensional shapes allocation algorithm flow chart.

2.8 Software Development

2.8.1 Design Objectives

The designed software system takes into account that shop-floor level operators are the intended users of such a system, that need to produce layouts for large scale problems in short time. This implies the need for an easy to use system that trades-off the optimality of the solution by its practicality. Therefore, a user friendly interface is at its front end, a heuristic-based inference engine is in its kernel, and since eventually, a layout of a set of pieces is to be produced by this system, graphical visualization of the obtained layout is at its back end.

Since this system is intended to be a part of a decision supporting system for a flame cutting machine, it has been made integrable to other modules like the tool path generator. Also the output of this system is made transferable to the industrially popular computer-aided-drafting package; AutoCAD®, so that further manipulations of the produced layouts could be made, in addition to the utilization of its plotting capabilities.

2.8.2 Achievement Approach

A highly modular software package; that is flexible for future enhancement has been designed. Such modularity permits the addition of more human intuitive thoughts in laying out two-dimensional shapes, and other practical layout constraints, into the already suggested algorithm easily. Further more, the automation of heuristic selection could be easily incorporated into the package.

A computer program written in Microsoft Quick Basic® V4.5 was developed to implement the proposed approach. The Quick Basic was chosen due to its simplicity, high modularity, and sufficiency to handle the problem under consideration.

2.8.3 System Architecture

To achieve the intended objectives the following modules have been developed:

- User interface: It is a menu driven system that enables the user to interact with the system; specifically, selecting the different functions that could be performed by the system, entering problem specific data, like: raw material specifications, BOM details, and selection of heuristics to be used for a given problem instance.
- Data editor: An interface to an already existing text editor has been incorporated into the system to facilitate numerical data entry, and provide the capability for future modification of such data. The text editor used is the popular DOS Edit command available on any PC-based computer. Also interfacing to the AutoCAD is considered for graphical data entry.
- Statistical module: This module performs some of the basic descriptive statistics on the BOM to be allocated, to provide insight on the characteristics of the shapes to be nested. It is also intended to use this module in the automation of heuristic selection process, making use of the statistical characteristics of the BOM to be allocated.

- Heuristics module: The different heuristics to be used are entered by this module, the user then has the capability to select from the installed heuristics at later stages the ones to be used in a given allocation process.

-Allocation module: This is the inference engine of the system, where the actual processing of the data is made through the application of the preselected heuristics on the provided BOM. Waste handling is also provided through this module. IF-THEN rules which are typical in a production system are used to detect the applicability of any given heuristic at its relevant condition(s) of application.

-Graphical visualization: The results of allocation are processed and displayed graphically on the screen of the computer to provide the user with the obtained layout. Also other useful information are displayed by this module regarding the obtained allocation like: utilization rate, percentage allocation made, and the processing time.

2.8.4 System Operation

The flowchart of Figure 2.11 illustrates the major steps in using the approach. A human operator enters the data of the problem, then selects the heuristics to be used, after that a solution is obtained automatically and inspected by the operator, who accepts the solution or makes other trials by reselecting other heuristics and/or other problem parameters.

2.9 Test of the approach

Two tests are conducted to examine the performance of the suggested algorithm, such that the results could be compared with previously published results of rectangular allocation problems with nearly similar requirements and constraints:

1) A random BOM generator (see Appendix A) based on a uniform distribution has been developed and used to generate test samples with specified maximum piece area to sheet area ratio, maximum aspect ratio of pieces, and maximum permissible demand of a piece type. The generated samples were categorized into seven classes based on maximum piece area to stock sheet area , where the aspect ratio was kept at a constant value, as well as the sheet dimensions. Twenty samples of each category were generated. Then different individual heuristics and combination of heuristics were then applied to these BOM's. The major characteristics of the BOM's used along with the results are given in the next section. Figure 2.19 to Figure 2.25 represent typical layouts obtained by the approach, where each one represents the first sheet allocation obtained for one sample BOM from each category of the tested samples.

2) Another set of BOM's was generated by sampling from a Beta distribution, which assumes different shapes with different parameter values (see Appendix B) to control the area distribution of pieces in a sample and to also control the aspect ratio distribution in the same sample. The samples were categorized into nine classes that represent the different possibilities of cases for area and aspect ratio.

Also, twenty samples of each category were generated. The categories and the corresponding results are given in the next section. Figure 2.26 to Figure 2.34 also represent typical layouts obtained, where each figure represents the first sheet allocation of a sample BOM from each category of tested samples.

2.10 Experimental results

To assess the performance of the approach several indices for measuring solution quality could be used. The most obvious one is the total waste produced after allocating all the pieces in a BOM, calculated over fully packed stock sheets. The other index that could be used also is the waste on the first sheet allocated for a given BOM, rather than the total waste over all sheets. Since this shows the ultimate potential of the approach, where it is focused on filling waste regions, and when allocating the first sheet a relatively -to other subsequent sheets- larger number of remaining unallocated pieces could be considered for filling purposes. Also taking into account the practical concept of reusable scrap (illustrated in Figure 1.1), packing density; which is the ratio of total area of pieces in a BOM to stock sheet area used, could also be utilized. The sheet area used is defined as follows:

$$\text{Sheet area used} = N{\cdot}A - S \qquad (2.1)$$

Where N is the number of stock sheets used, A is the area of a stock sheet, and S is the total reusable scrap generated.

Where the reusable scrap area is accumulated based on a reusable scrap policy [15], specified as follows: a waste region is considered reusable if it could accommodate the average piece in the BOM to be allocated.

Thus the packing density (*P.D.*)becomes:

$$P.D = \frac{\sum_{i=1}^{n} a_i}{N \cdot A - S} \tag{2.2}$$

Where a_i is the area of a rectangular shape in the BOM to be allocated.

Also, computational time of allocation could be used to assess the performance of the algorithm, especially when comparing it with approaches based on human manual intervention in automatic layout solution approaches, or with pure manual layouts production.

It was clear from the obtained results that combinations of heuristics showed to out perform individual heuristics over the range of tested samples of BOM's. The effects of applying different heuristic rules, and their different possible combinations are demonstrated on a sample BOM of 1/50 maximum area ratio and given in Table 2.1. Figure 2.12 to Figure 2.17 illustrate some of these results on the first sheet of each case.

Table 2.1 Effects of using different heuristic rules on layout as applied to a sample BOM with maximum area ratio =1/50, maximum aspect ratio =3, and BOM size =214. Stock sheet dimensions: 70 x 40 units.

No.	Heuristic rule(s)	Scrap % on 1st sheet	Packing Density % (P.D.)	Total scrap %	Total Allocation time (sec.)	Figure No.
1	FFD	7.07	89.7	9.75	0.219	2.12
2	FFD+BE	3.50	91.32	5.59	0.281	2.13
3	FFD+BF	2.79	95.23	4.34	0.492	-
4	FFD+S	8.32	90.23	9.46	0.336	-
5	FFD+M	1.64	96.43	3.16	0.664	-
6	FFD+L	1.79	96.36	3.34	0.820	-
7	FFD+BE+BF	1.43	95.90	3.07	0.609	-
8	FFD+BE+S	3.14	91.82	5.27	0.438	-
9	FFD+BE+M	1.29	97.02	2.96	0.727	-
10	FFD+BE+L	1.43	97.48	3.14	0.750	-
11	FFD+S+BF	3.11	94.95	4.48	0.609	-
12	FFD+S+M	0.57	96.96	3.41	0.813	2.14
13	FFD+S+L	1.21	97.24	3.20	0.938	2.15
14	FFD+BE+S+BF	1.61	96.25	2.89	0.539	-
15	FFD+BE+S+M	1.61	96.99	2.68	0.766	2.16
16	FFD+BE+S+L	1.61	97.02	2.79	0.883	2.17

From Table 2.1, the best results as represented by the minimum total scrap percentage are for the combination of heuristics that include the best fit of end of cycles (BE), with shift(S) and merging regions(M) for gaps between cycles. Also, very similar results are obtained when replacing merge regions(M) with the largest rectangle region (L) find and fill rule.

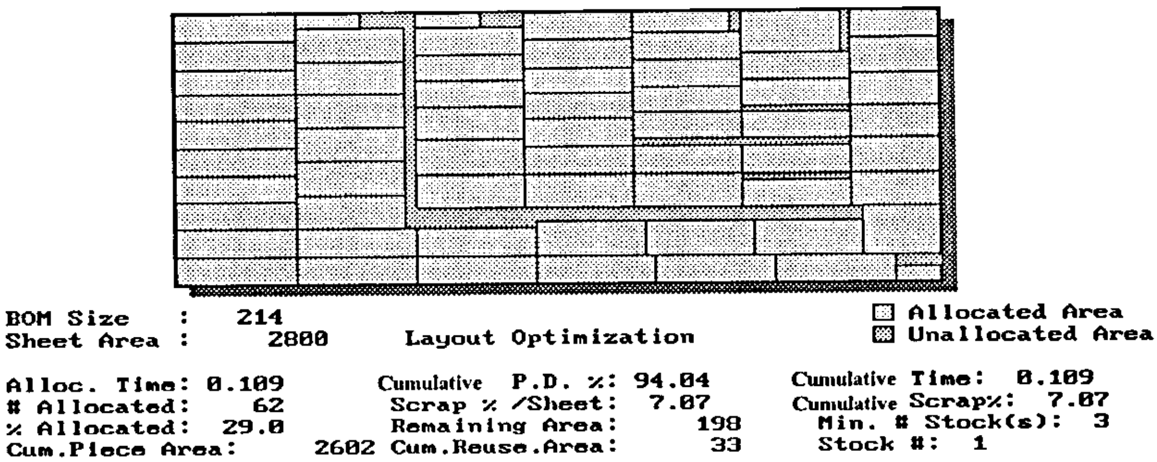


Figure 2.12 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD heuristic rule for main space allocations and end of cycle regions.

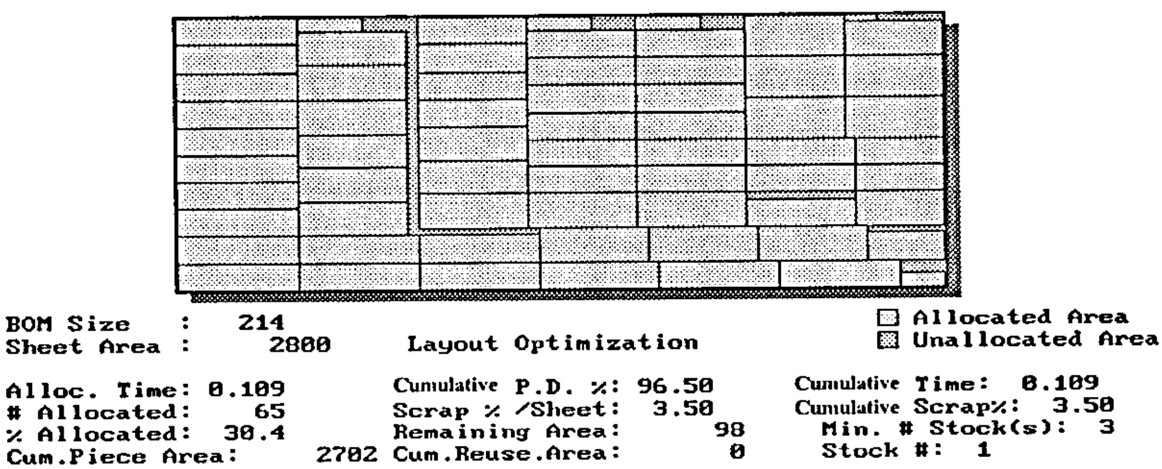


Figure 2.13 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD+ best fit filling of end of cycle regions(BE).

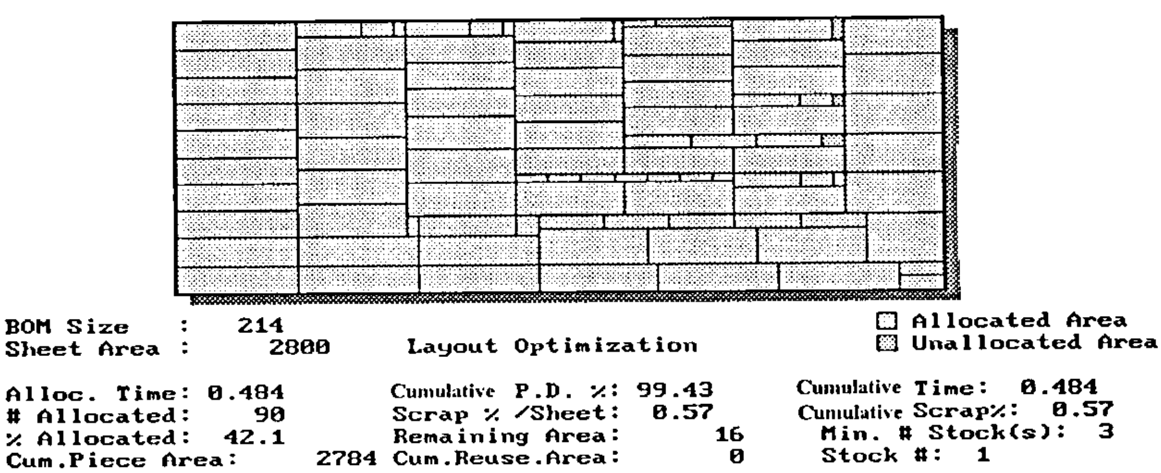


Figure 2.14 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD+ Shift(S)+ merge waste gaps into larger regions then filling them(M).

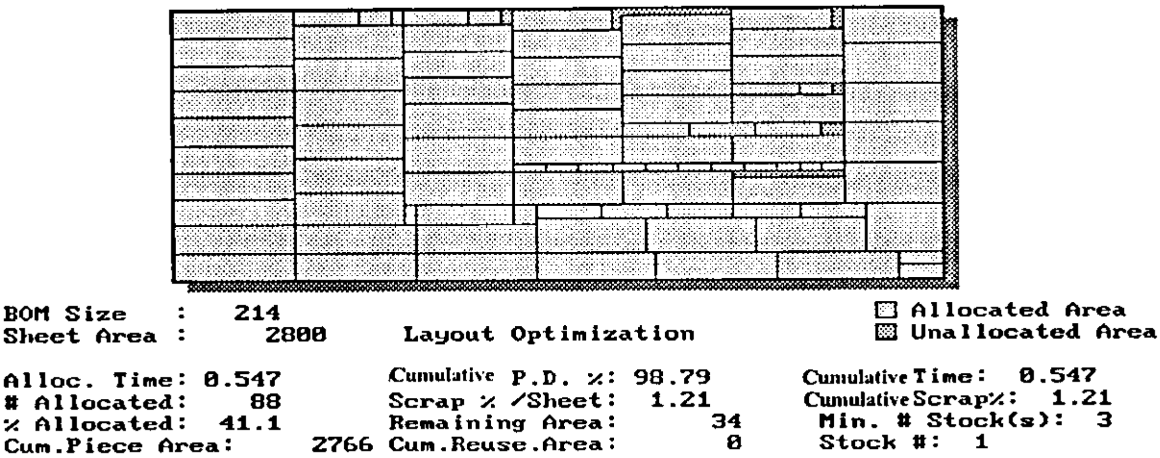


Figure 2.15 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD + shift(S)+ get the largest rectangle waste then fill it(L).

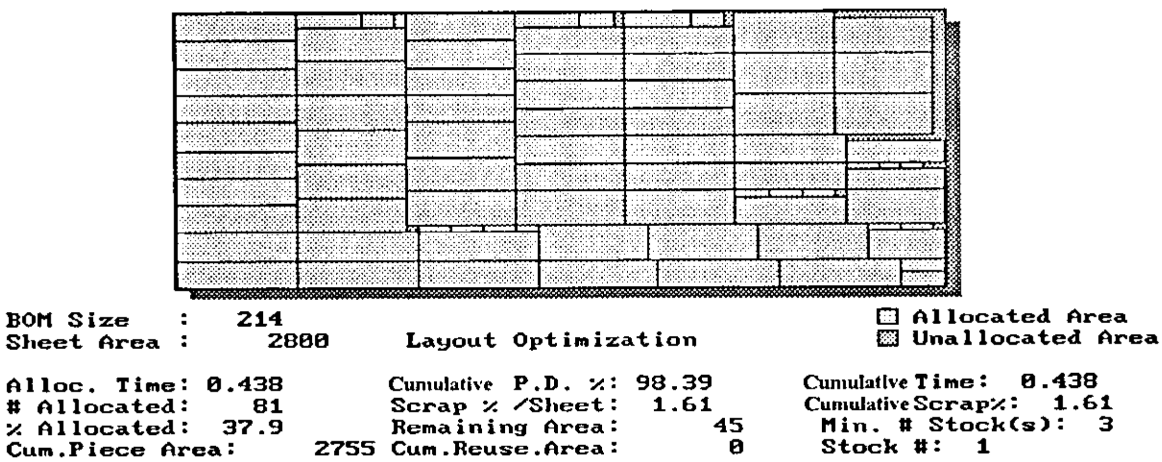


Figure 2.16 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD+ shift(s)+ merge gaps & fill(M)+ best fit fill of end of cycle regions(BE).

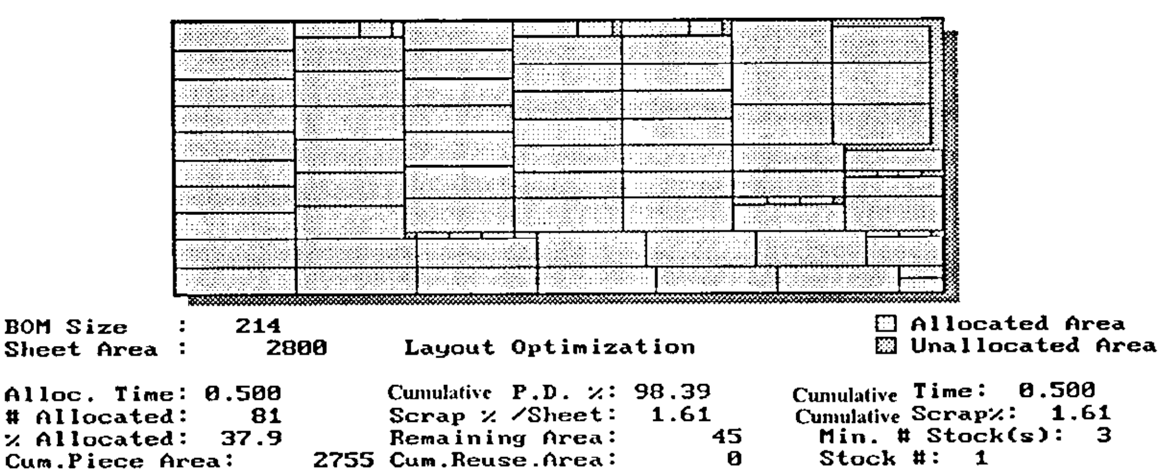


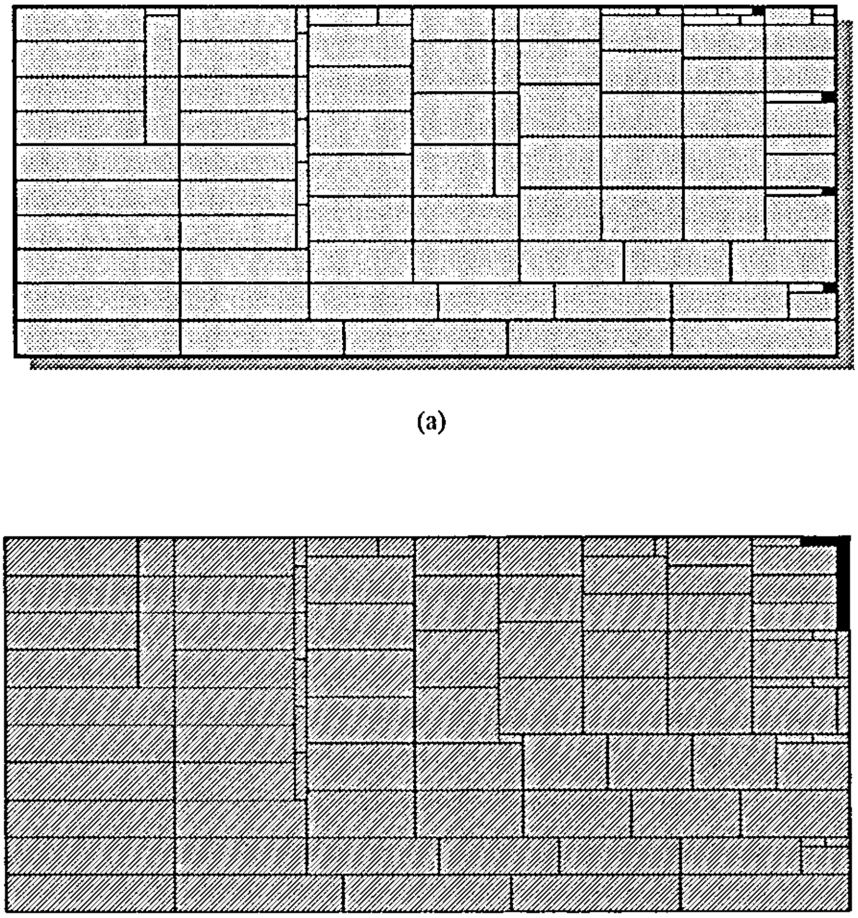
Figure 2.17 Rectangular shapes layout of a BOM of 1/50 maximum area ratio using FFD + shift(S)+ largest rectangle find & fill (L)+ best fit fill of end of cycle regions(BE).

Figure 2.12 illustrates the layout without any gaps filling heuristics except for end of cycle regions were the first-fit-decreasing was used. The total waste obtained is relatively large as one would expect. Figure 2.13 shows the effect of filling the end-of-cycle regions with best-fit-filling rule, where an improvement on the waste is introduced. Figure 2.14 and Figure 2.15 show the effect of combining two heuristic rules for filling gaps between cycles. Their results also show improvement especially on the first sheet as given in Table 2.1. However, the total waste could still be improved upon as Figure 2.16 and Figure 2.17 show, where best-fit-fill of end-of-cycles is combined with the two previous rules to get better total waste, even though the waste on the first sheet increased.

It was noted that over the range of tested samples the following combination dominated the others in the quality of the layout obtained as measured by the total waste: Best-fit-decreasing for filling end of cycle regions, and Shift combined with Merging of spaces, and Shift combined with Largest rectangle being nearly equivalent for filling the Gaps between cycles. Table 2.2 and Table 2.3 represent the results obtained for the two tests performed using the latter combination of heuristics. The last column in both tables show the results obtained by Israni and Sanders [8] for their FFDLP algorithm with human intervention to fill the gaps generated during allocation, and the results for Israni and Sanders[15] respectively.

It was difficult to directly compare the results of this work with previous research results since no faithful common bases for comparison purposes could be found, due to differences in constrains and requirements of layout problems published. Nevertheless, a number of examples (shown in Table 2.2) with characteristics that match those of the examples presented by Israni & Sanders [8] was generated. However, the results of [8] were reported for three trials of each category as shown in the last column of Table 2.2. Whereas results of this work are presented for twenty trials of each category since this is -relatively- more faithful in characterizing the performance of the algorithm, since an analytical evaluation of performance of such a heuristic approach could not be easily derived [15]. Furthermore, if the obtained results were reported for three trials only, we could have seen better -but not reliable- results than those reported in [8]. To demonstrate this, the detailed runs given in Table 2.4 for maximum area ratio of 1/100 are presented, where it can be seen from the first three trials that the scrap rate is zero. Then adopting these results to represent the performance of the algorithm is totally incorrect and misleading.

Figure 2.18 shows the allocation obtained by the presented approach using a BOM that was used by [8], were they got a 0.46% scrap on the first sheet (the fully packed sheet in this case), and using the suggested algorithm a 0.14% scrap was obtained. The details of the BOM (length, width, demand) is as follows: {(14,4,10), (11,4,7), (10,4,10), (9,5,11), (8,3,1), (7,6,10), (7,5,4), (7,4,6), (7,3,5), (6,5,1), (6,4,7), (6,3,8), (6,2,5), (5,4,5), (5,1,9), (4,3,7), (4,2,4), (3,214), (3,1,5), (2,1,6)}. The stock sheet dimensions are: 70x40 units.



(a)

(b)

Figure 2.18 A fully packed sheet using a sample BOM from [8].

(a)Allocation as obtained using the suggested algorithm. Scrap rate is 0.14%.

(b)Allocation using the algorithm of [8]. Scrap rate is 0.46%.

The same note for the number of trials given for the first set of results holds also when comparing results of the proposed approach with the results of [15]. Where outcomes for twenty trials on each category are reported in Table 2.3; however, the results of [15] were reported for four trials only.

Table 2.2: Randomly generated BOM's from a uniform distribution and their allocation results using the proposed filling heuristics*.

Max. area ratio	Number of trials (BOM's)	Average BOM size	Average number of pieces types	Average number of sheets	Average allocation time (sec)**	Average packing density%	Average scrap% ***	Average Scrap% by:[8]♣
1/200	20	487	10	2	1.61	99.90	0.14	-
1/100	20	311	19	2	1.20	99.03	0.84	1.36
1/50	20	265	20	3	1.20	97.08	2.84	0.62
1/20	20	132	20	4	0.49	94.51	4.97	-
1/10	20	84	14	5	0.39	92.14	7.06	5.74
1/6	20	57	10	6	0.28	87.44	13.13	9.04
1/4	20	56	11	9	0.28	85.08	16.20	11.96

* All BOM's have a max. aspect ratio of 3.
** On a PC 486 DX2
*** Scrap % is calculated on fully packed sheets over 20 trials.
♣ Scrap % is calculated on fully packed sheets over 3 trials.

Table 2.3: Randomly generated BOM's from Beta-distribution and their allocation results using the proposed filling heuristics*.

#	Number of trials (BOM's)	Area ratio status	Aspect ratio status	Average number of pieces types	Average number of sheets	Average allocation time (sec) **	Average packing density %	Average packing density by (LI) *** [15]
1	20	S	S	61	3	0.40	93.65	94.30
2	20	S	M	73	3	0.40	95.71	97.59
3	20	S	L	59	4	0.46	96.00	97.28
4	20	M	S	83	15	0.63	89.05	89.82
5	20	M	M	90	14	0.55	91.81	91.99
6	20	M	L	53	15	0.71	92.76	92.43
7	20	L	S	60	37	0.68	72.39	70.13
8	20	L	M	73	29	0.71	80.47	76.54
9	20	L	L	29	26	0.63	88.92	85.35

S: Small M: Medium L: Large
* All BOM's have a size of 100 pieces.
** On a PC 486 DX2
*** (LI) Decreasing length perpendicular strip packing with human intervention. Packing Density calculated over 4 trials/ category.

Table 2.4: Detailed results for Allocating 1/00 area-ratio pieces:

#	BOM Size	Variety	Total Scrap % **	Packing Density %	No. of Sheets	Time (sec) •
1	357	19	00.00	99.37	2	1.258
2	407	19	00.00	99.33	2	1.984
3	394	19	00.00	99.86	2	1.641
4	279	20	00.00	99.25	2	0.930
5	279	19	00.00	99.30	2	0.980
6	236	19	00.18	99.45	2	0.719
7	222	19	00.18	99.74	2	0.602
8	324	19	00.46	99.36	2	1.539
9	309	19	00.50	98.62	2	1.211
10	329	19	00.64	99.16	2	1.102
11	224	19	00.68	98.96	2	0.652
12	454	19	01.00	99.49	3	2.301
13	381	19	01.04	98.88	2	1.375
14	232	18	01.32	98.57	2	0.820
15	194	19	01.46	98.14	2	0.488
16	292	19	01.75	98.84	2	1.219
17	331	19	01.82	98.45	2	1.313
18	327	19	01.89	97.97	2	1.383
19	253	19	01.96	98.36	2	0.934
20	400	19	01.98	99.50	3	1.594
Averages	311	19	00.84	99.03	2	1.20

• On a PC 486 DX2 ** Scrap % is calculated on fully packed sheets.

As can be deduced from Table 2.2, generally it is concluded that the approach is practically efficient, especially for large scale problems. And its performance degrades as the piece-to-sheet area-ratio increases, which is natural, since the ability to maneuver larger area pieces is more restricted than smaller pieces, furthermore size mismatch among larger pieces produces larger scrap regions that could not be filled by other large pieces. But still the obtained figures for scrap rate are well below those practically accepted in industry[15]. It is apparent that the presented approach gives remarkably better results in cases of large number of sheets, which is an expected result, since a human operator could not comprehend such large scale problems effectively. It is also worth mentioning that the performance of the algorithm becomes superior when the distribution of piece areas is smooth, i.e. with no abrupt changes in areas.

In both sets of tests the algorithm proves to be efficient in terms of computational time, especially for large scale problems as the reported results for the time of allocation in both Tables 2.2 and 2.3 show. This is contrasted to the fact that the time required for laying out pieces when human intervention is permitted, is highly dependent on the skill of the operator, and ranges from ten seconds to few minutes per stock sheet, and may extend to hours if large number of sheets with different sizes are to be tried [27]. Furthermore the trim losses in such cases could be significantly large. Figure 2.19 to Figure 2.34 are representative layouts for the first sheet for each category in Table 2.2 and Table 2.3.

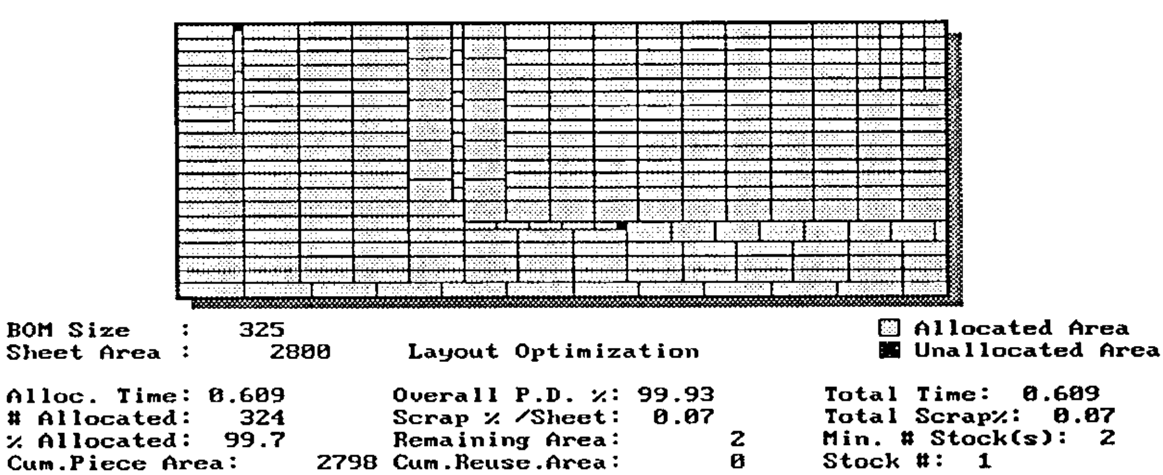
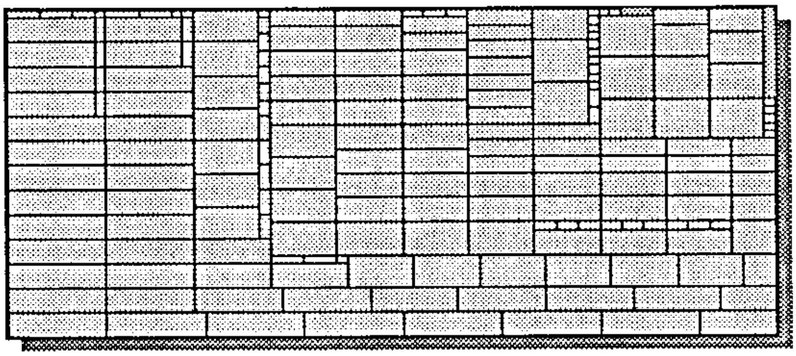


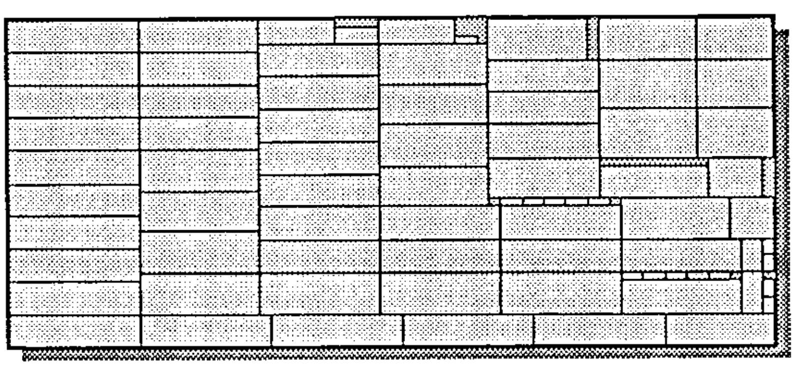
Figure 2.19 Rectangular shapes layout of a BOM with 1/200 maximum area ratio.



BOM Size : 309
Sheet Area : 2800
Layout Optimization
Allocated Area
Unallocated Area

Alloc. Time: 1.039
Allocated: 193
% Allocated: 62.5
Cum.Piece Area: 2786
Overall P.D. %: 99.50
Scrap % /Sheet: 0.50
Remaining Area: 14
Cum.Reuse.Area: 0
Total Time: 1.039
Total Scrap%: 0.50
Min. # Stock(s): 2
Stock #: 1

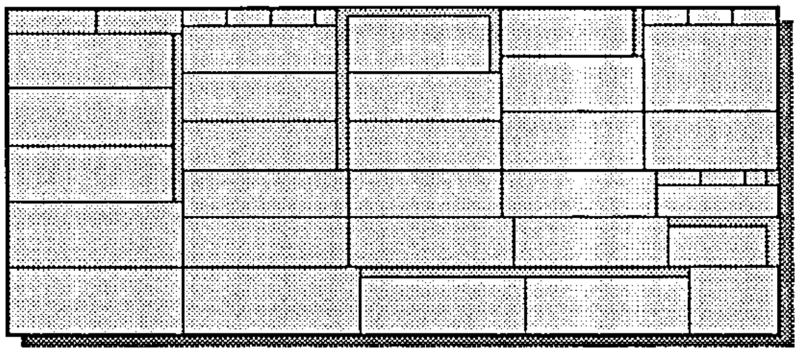
Figure 2.20 Rectangular shapes layout of a BOM with 1/100 maximum area ratio.



BOM Size : 220
Sheet Area : 2800
Layout Optimization
Allocated Area
Unallocated Area

Alloc. Time: 0.438
Allocated: 78
% Allocated: 35.5
Cum.Piece Area: 2765
Overall P.D. %: 98.75
Scrap % /Sheet: 1.25
Remaining Area: 35
Cum.Reuse.Area: 0
Total Time: 0.438
Total Scrap%: 1.25
Min. # Stock(s): 3
Stock #: 1

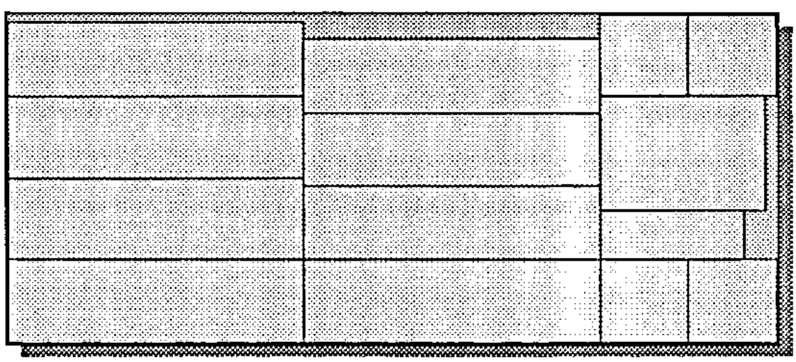
Figure 2.21 Rectangular shapes layout of a BOM with 1/50 maximum area ratio.



BOM Size : 110		☐ Allocated Area
Sheet Area : 2800	Layout Optimization	▩ Unallocated Area

Alloc. Time: 0.219	Overall P.D. %: 95.89	Total Time: 0.219
# Allocated: 40	Scrap % /Sheet: 4.11	Total Scrap%: 4.11
% Allocated: 36.4	Remaining Area: 115	Min. # Stock(s): 3
Cum.Piece Area: 2685	Cum.Reuse.Area: 0	Stock #: 1

Figure 2.22 Rectangular shapes layout of a BOM with 1/20 maximum area ratio.



BOM Size : 51		☐ Allocated Area
Sheet Area : 2800	Layout Optimization	▩ Unallocated Area

Alloc. Time: 0.047	Overall P.D. %: 95.00	Total Time: 0.047
# Allocated: 14	Scrap % /Sheet: 5.00	Total Scrap%: 5.00
% Allocated: 27.5	Remaining Area: 140	Min. # Stock(s): 3
Cum.Piece Area: 2660	Cum.Reuse.Area: 0	Stock #: 1

Figure 2.23 Rectangular shapes layout of a BOM with 1/10 maximum area ratio.

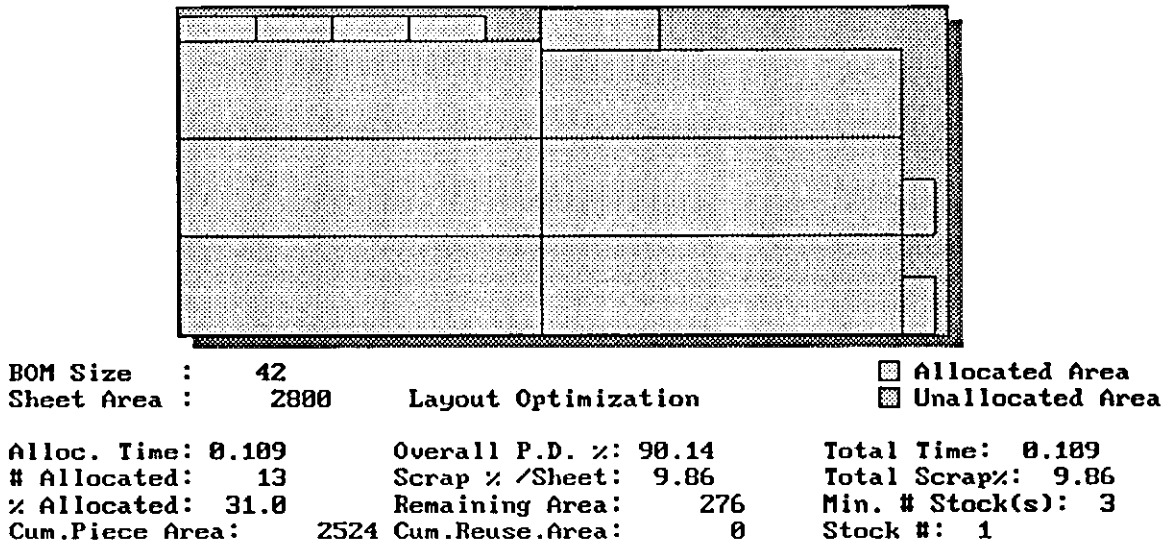


Figure 2.24 Rectangular shapes layout of a BOM with 1/6 maximum area ratio.

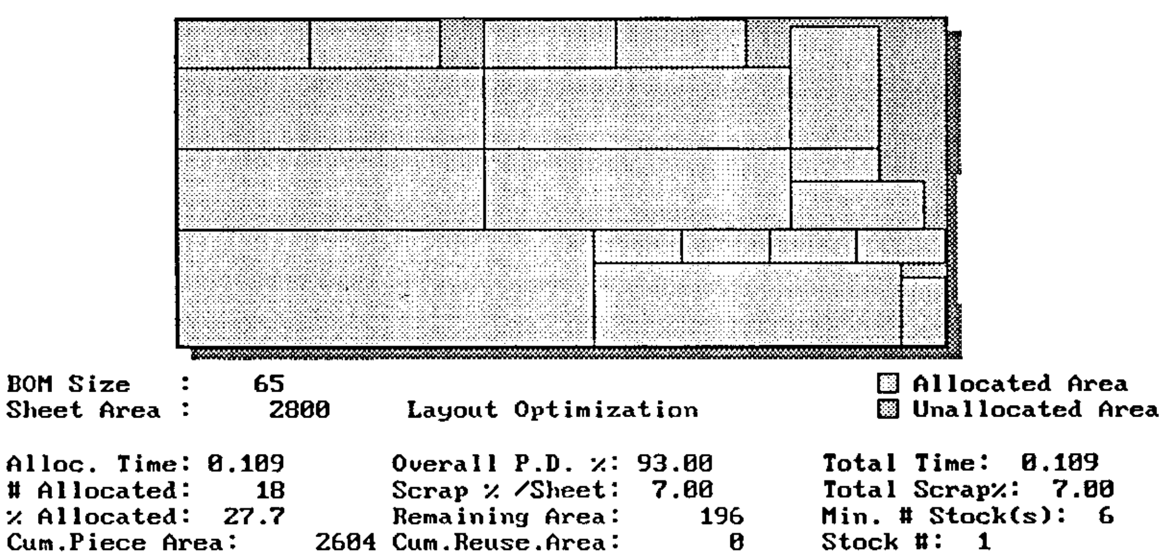


Figure 2.25 Rectangular shapes layout of a BOM with 1/4 maximum area ratio.

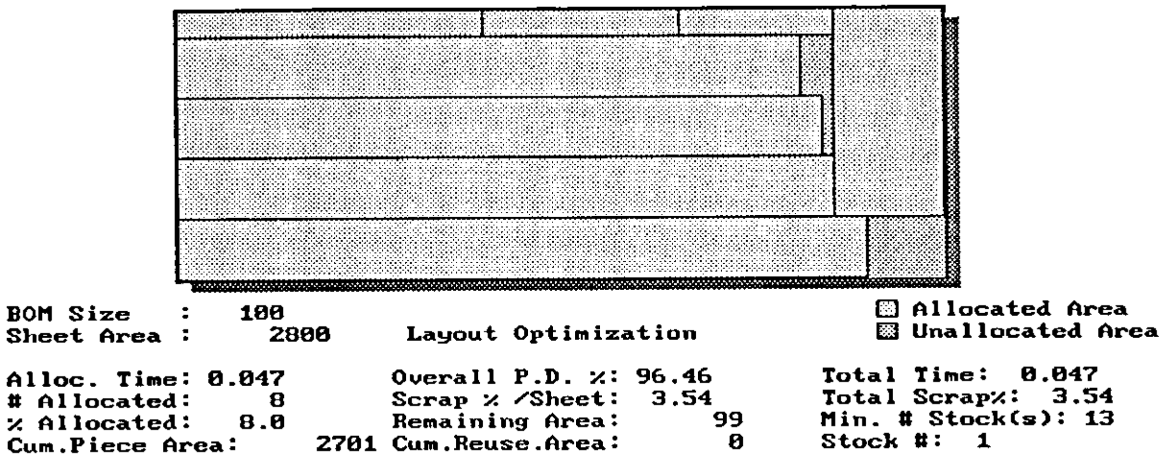


Figure 2.26 Rectangular layout of a BOM with medium area ratio & medium aspect ratio

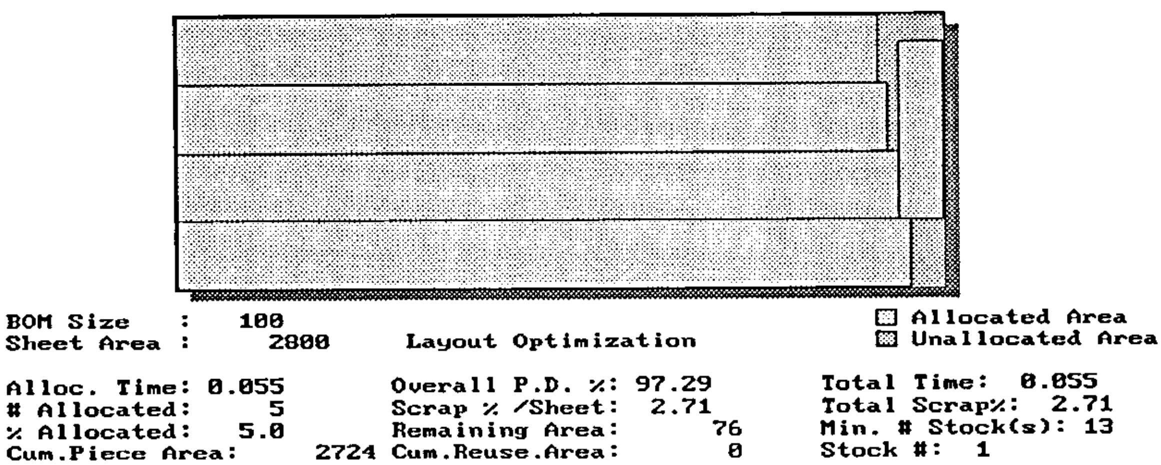


Figure 2.27 Rectangular layout of a BOM with medium area ratio & large aspect ratio.

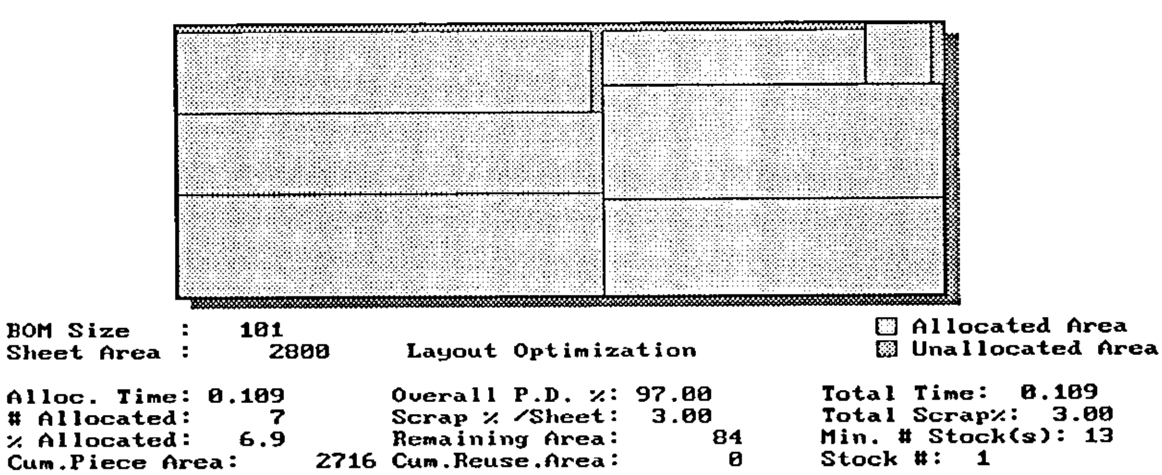


Figure 2.28 Rectangular layout of a BOM with medium area ratio & small aspect ratio.

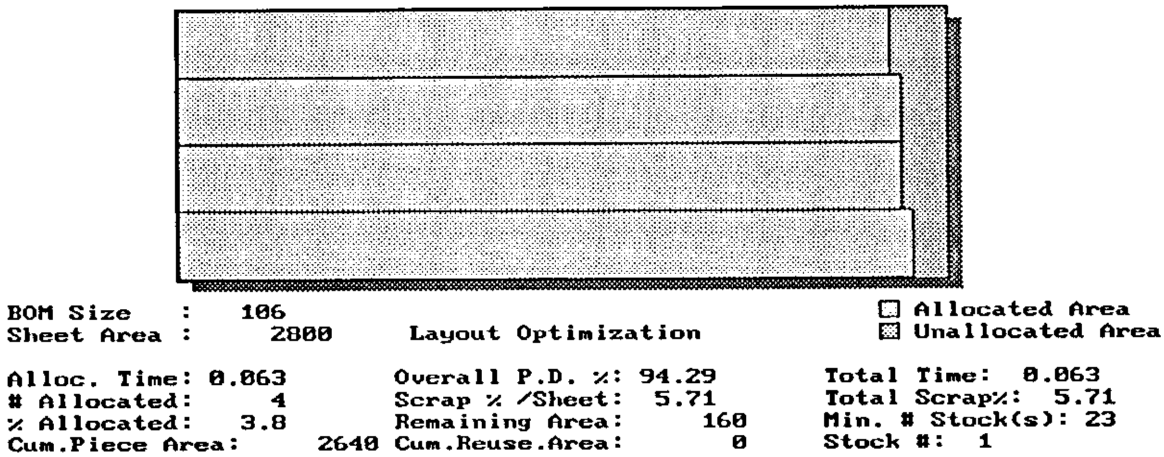


Figure 2.29 Rectangular layout of a BOM with large area ratio & medium aspect ratio.

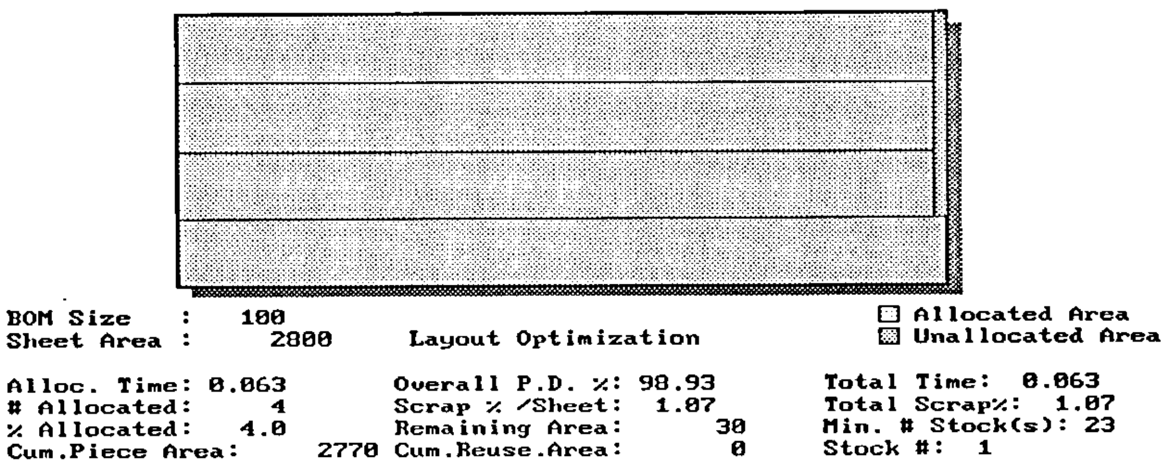


Figure 2.30 Rectangular layout of a BOM with large area ratio & large aspect ratio.

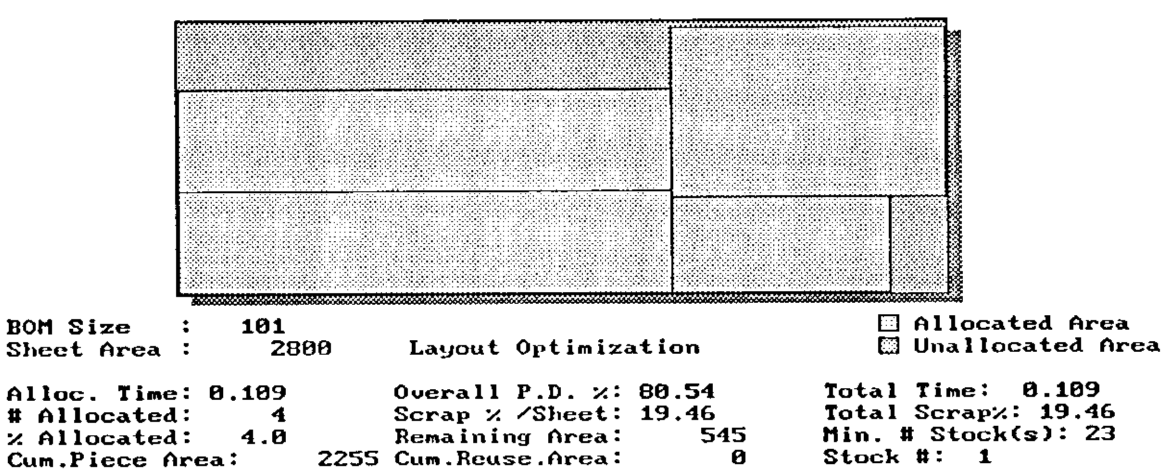


Figure 2.31 Rectangular layout of a BOM with large area ratio & small aspect ratio.

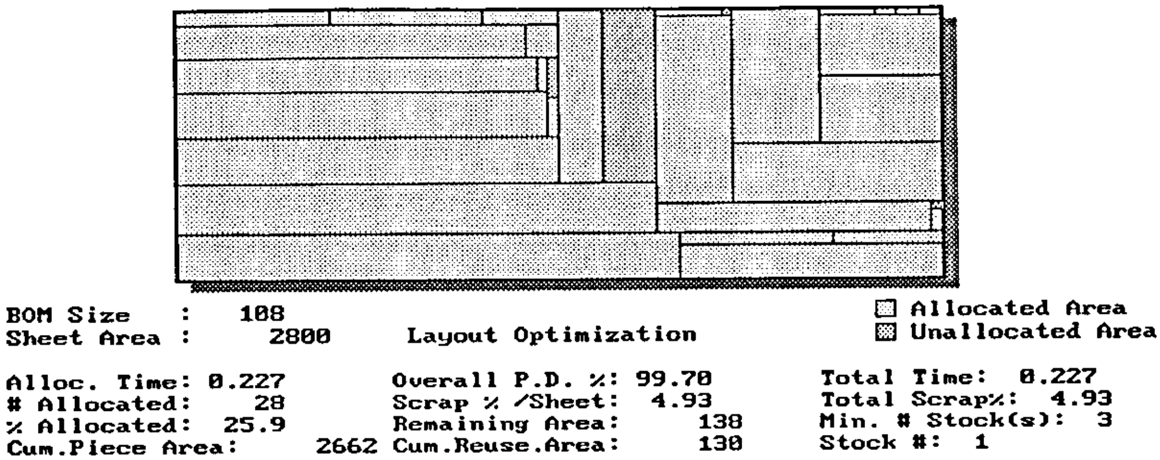


Figure 2.32 Rectangular layout of a BOM with small area ratio & medium aspect ratio.

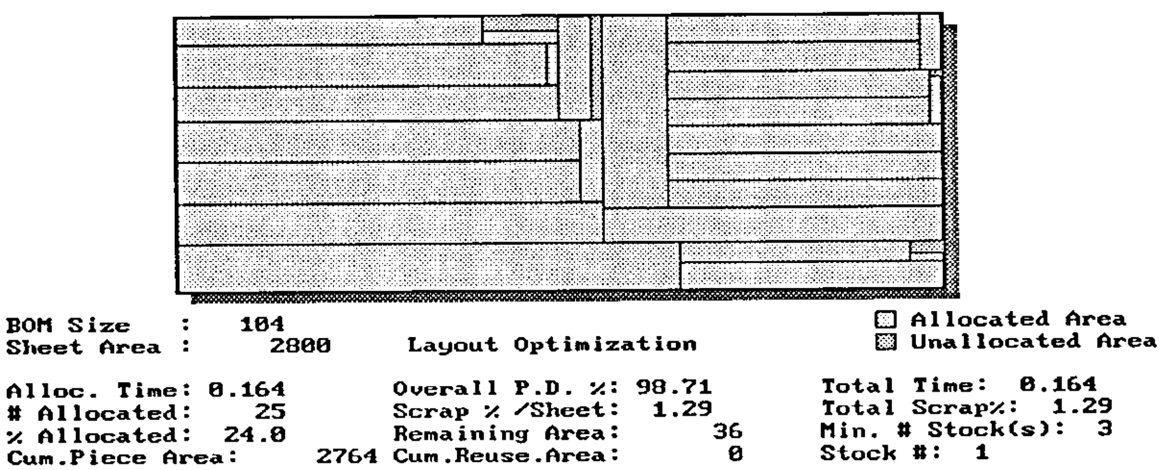


Figure 2.33 Rectangular layout of a BOM with small area ratio & large aspect ratio.

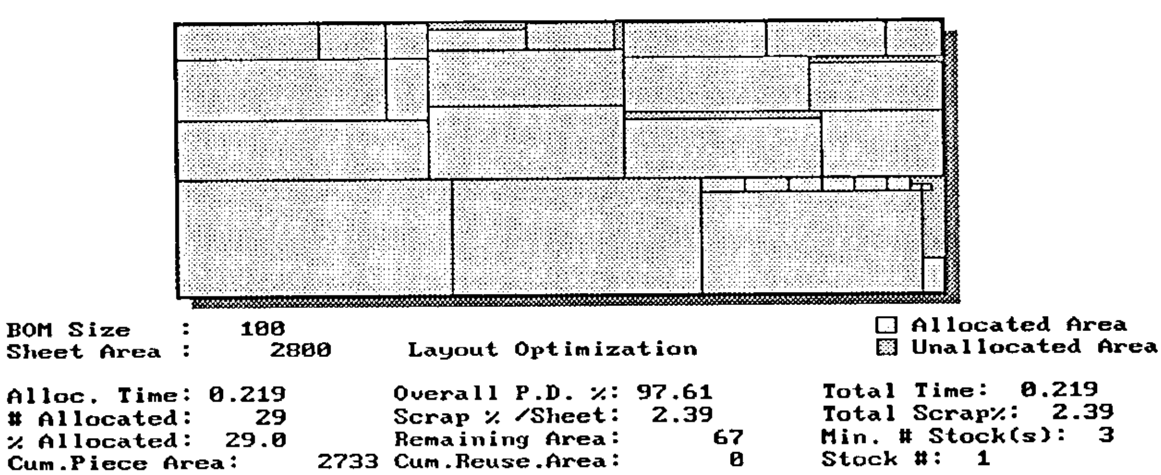


Figure 2.34 Rectangular layout of a BOM with small area ratio & small aspect ratio.

2.11 Conclusions

From the results presented in Table 2.2 and Table 2.3, and as illustrated by Figure 2.19 to Figure 2.34 it can be seen that the approach presented previously has succeeded in eliminating the manual human intervention, but not his intuitive thoughts. Simultaneously the scrap, and utilization figures are nearly kept at the same levels of previous approaches if not enhanced. The approach is much more promising for large scale problems and similar ideas could be applied for irregular shapes allocation. The different characteristics of the BOM affecting the performance of the approach requires more research, and is suggested for future work.

Only a few of used human strategies were tested in this work to show its potential, and to compare it with already published results.

Automating the selection of filling heuristics at different strategic points of allocations based on BOM characteristics could be suggested as an improvement on the approach (Fuzzy Logic could be considered for such reasoning process), in addition to identifying and including more human intuitive thoughts.

Treating the rectangular shapes layout separately from the irregular shapes is deliberately made to take advantage of the simplicity in rectangular shapes representation alone, also due to, relatively easier functions needed to position and rotate such shapes. It has also been noted that throughout literature there has been a rare mixing of the two problems[2,31,32]. Also the assumption of infinite length sheet in the case of irregular shapes allocation -to be treated in the next chapter- obligates such distinction in this research.

Chapter 3

Two dimensional irregular shapes allocation

3.1 Introduction

The problem of allocating (nesting) two-dimensional irregular shapes on a single sheet or plate of raw material is of theoretical and practical importance. However, it hasn't been given so much attention in literature, as the rectangular shapes version of the problem. On the other hand, although algorithms have been developed for irregular parts nesting, very few of them have been published due to commercial confidentiality [9].

In addition to the wide existence of this problem in sheet metal industries, it is a major problem in textile, leather and wood industries among others. The mass production in sheet metal industries is the immediate justification for the quest of waste reduction, through better shapes nesting, since the raw material cost is relatively cheap. However, in the textile, leather and wood industries the relatively higher cost of raw materials is the prime factor, in addition to the mass production effect.

These facts combined with our objective of constructing a flame cutting machine capable of automatically cutting, in an efficient way, different two-dimensional geometrical shapes, motivates our work on developing an automatic nesting algorithm for irregular shapes, in addition to the rectangular nesting algorithm presented previously. Thus the machine becomes served by automatic nesting modules as a decision supporting system.

3.2 Literature review & currently available solution approaches

The literature review for this problem -as will be presented- and the field visits made to a number of local industrial companies, show that the approaches developed and practiced so far for solving this problem, could be grouped in three categories:

1. Manual approaches:

a) A human operator makes use of templates (patterns), made of cardboard or metal, that are 1:1 scaled to actual sizes of pieces to be cut. The operator manipulates these patterns over a working table within the boundaries of raw material, making several trials to reach an arrangement of the pieces that seems to have an "optimal" nesting, i.e. minimum possible waste[33].

b) In another approach, although a computer is used to improve accuracy and efficiency, descriptions of pieces are filed and processed by the computer. A plotter produces copies of the pieces (scale is not necessarily 1:1). The pieces are cut out and arranged manually into frame representing the stock material. The locations of the arranged pieces are then fed back into the computer by terminal or digitizing table. The final nesting drawing is then generated by the computer on a graphic device [23].

c) In the third approach, nesting is carried out using an interactive computer graphics system. The pieces and raw material sheet are displayed on the screen, the operator manipulates (translate, rotate, flip...etc.) pieces using an input device such as light pen, joystick, mouse...etc. During the nesting process the computer continuously checks overlaps and other possible constraints, and calculates the waste or utilization achieved. Like pure manual nesting several trials and manipulations need to be performed by the operator to reach a seemingly satisfactory allocation [33].

2. Automatic approaches:

a) Random generation of feasible nesting cited in [23] with no further details.

b) A two-stage computer algorithm is used, where in such approach the problem is converted from one of placing irregular shapes into allocation of rectangular modules. These modules are obtained, in general, from the rectangular enclosure for various clusters of irregular and rectangular pieces in a first stage. Then in the second stage, one of the approaches for solving the rectangular nesting problem is utilized to get a final allocation of these rectangular modules [31, 34, 35]. Although these approaches have been successful in handling specific cases of the general problem their approach; however, has resulted in procedures that are marginally useful for applications with small number of pieces when an exact ordering of the pieces is required, cited in[33].

c) A third class of the automatic approach, where a procedure is defined, which searches for an optimal arrangement by operating directly on the irregular shapes. They are based on a set of heuristic rules used by deterministic sequential placement techniques; once a piece has been placed in a certain position, it will not be reconsidered any more in respect to any consideration that may result later, cited in [33].

d) A search-space approach for solving the problem was suggested in [33]. Where a prefixed placement policy; namely the left-most-lowest allocation was used as a placement heuristic, in addition to a number of simplifying assumptions to reduce the size of the solution space. An evaluation function based on the minimum added waste at each stage was used to select the next piece to be placed. The work was applied directly on the irregular shapes without any rectangular enclosures or approximations.

e) A heuristic approach based on representing the irregular shapes by few non overlapping rectangles was presented in [25]. Allocation of pieces is then performed on subregions of the stock sheet, that are either horizontal or vertical strips. The piece to be allocated at each step is chosen by an evaluation function that selects the piece that covers most of the waste area in the minimum enclosing rectangle of the previous piece allocated. The process continues in strip-filling approach until the sheet is fully packed or the BOM is empty.

f) Also another heuristic approach was presented in [2]. The approach is claimed to have the capability for processing both rectangular and irregular shapes. The algorithm is a two-stage hierarchical approach that deals with multi-plate allocations. In the first stage, initial allocation of shapes to different raw material plates is made through mathematical programming, then based on the results of this stage, detailed allocation is made through heuristic priority rules based on certain shape features.

g) Recently, approaches based on simulated-annealing schemes are being suggested, and they are still under investigation. There hasn't yet enough practical experience gained with such methods[17, 36].

h) Kopardekar et.al.[24] claimed that utilization obtained by automatic nesting approaches is lesser or no better than layouts generated by humans. Therefore, they suggest that an approach based on identifying human intuitive thoughts involved in laying out irregular parts should be investigated. Based on extracting such thoughts one could devise a new heuristic approach that would yield better layouts. These notes by Kopardekar et. al. were actually the basis on which the proposed approach presented in chapter two -for rectangular shapes nesting- is developed. And they will still affect the suggested approach to solve the irregular shapes nesting problem, as will be presented later in this chapter.

3. Semiautomatic approaches:

To alleviate shortcomings of previous approaches a hybrid approach is proposed, where a tentative automatic solution is generated, then improved upon by human interactive intervention. It is the belief that the human flexibility in recognizing matched patterns in very short time, and the capability to handle special cases, is the motivation for these approaches [21].

3.3 Discussion of previous approaches & motivations for a heuristic based solution approach

Manual approaches depend greatly on the skills and patience of the human operators. Their results degrade greatly for medium to large scale problems. Even though good quality results could be achieved in such approaches, they are time factorially with the depth of the search, as will be shown later. Finally more research on problem representation and procedures for solving it is needed[15].

3.4 Representing spatial knowledge

Reasoning about space is considered to be challenging from computational and representational perspectives [38]. The physical space of this problem has two dimensions, and the variables that govern the allocation of one piece in such a space are continuous; namely its position specified by the x and y coordinates of a shape reference point, and the orientation (θ). Thus a piece has -theoretically- an infinite number of possible positions and orientations for placement even though they should be placed within a finite sized resource of raw material, and they should not overlap. The shapes themselves are irregular, this complicates any positioning process since complicated algorithms are needed to make sure a piece is placed in a legal position (i.e. within the boundaries of the raw material and without overlapping or intersecting any other piece or forbidden region in the raw material). Thus spatial inference often requires quantitative reasoning involving geometric relations to represent the real object boundaries, to calculate distances, areas and angles. Also qualitative reasoning is required to govern topological relations such as touching, overlapping and containment[38].

3.5 Problem representation

A *state-space* representation is a general representational scheme, that not only aids in problem solution, but also is useful in visualizing and analyzing problem complexity. Furthermore, such representation could be used to predict the behaviour of proposed search algorithms [37].

In this representation a *graph* structure, shown in Figure 3.1, is used, which consists of *nodes* and *links* identifying *states* and *actions* respectively in problem solving process.

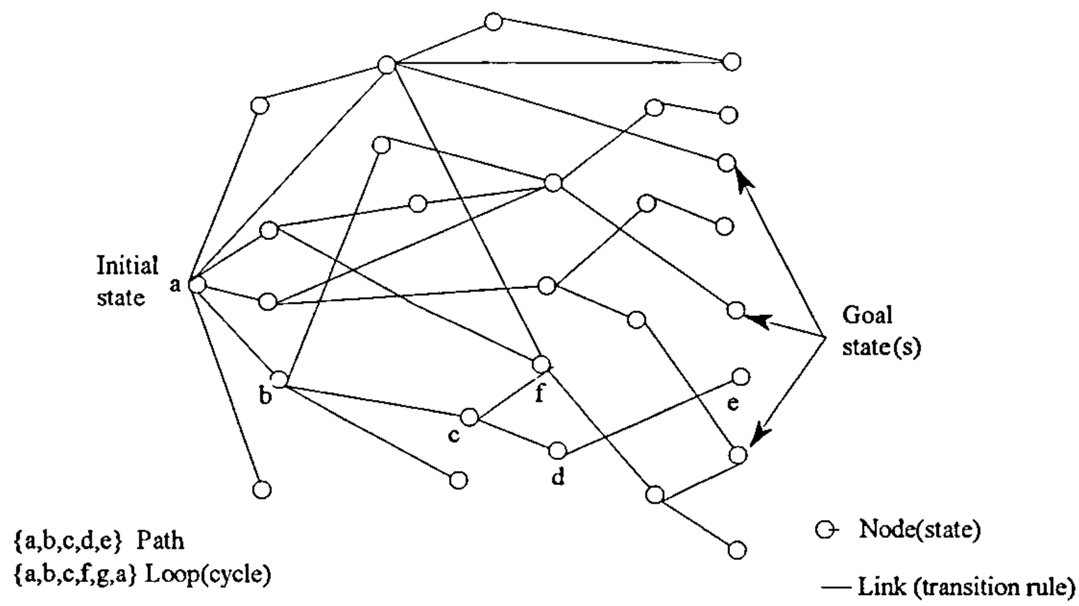


Figure 3.1 A typical problem state space graph, with basic terminology.

3.5.1 State space representation of the two-dimentional irregular shapes nesting

The two dimensional irregular cutting-stock problem could be cast in a state-space-search representation as follows:

- a *node* in the graph -a state in problem solution- is a partial allocation of pieces on the raw material, Figure 3.2 illustrates the idea.
- a *link* in the graph - an action in problem solution- which is also known as a transition rule from state to state in problem solution, is the possible subsequent legal allocation, governed by the application of:

- P*ositioning rule*: which determines at what (x, y) coordinates of a shape's reference point a piece could be placed without violating containment and overlap constraints.

- O*rientation rule*: which determines at what angle (θ) a shape could be placed.

- *Flip rule*: if the piece could be flipped about a horizontal or a vertical axis then this rule determines its possible orientation.

- N*ext piece selection rule*: that determines which of the not yet allocated pieces should be considered for placement on the raw material.

- *Other application specific rules*: in different industries specific constraints may be introduced to the set of transition rules; in natural leather industry and in textile industry for example, some pieces are restricted to be placed at certain parts of the raw material. In other cases, some regions of the raw material could be defective, and therefore pieces should not be placed on them. These application specific restrictions are not dealt with in this work, but are mentioned to show the flexibility in adding constraints in such representation.

Moving from one state to any other subsequent state is equivalent to allocating a new piece from the set of unallocated pieces at a specific legal position and in a legal orientation. Legal position means in this context that the piece if fully contained within the boundaries of the raw material (containment constraint), and it does not overlap any of the previously allocated pieces (overlap constraint). Legal orientation comes from the fact that in some applications piece placement is restricted to a limited number of orientations; like in textile industry, where the fabrics have directional properties.

To complete the representation of the problem solution as a search in a graph, the *initial node* and the final node (*goal node*) should also be defined. The *initial node* could be considered the empty raw material with no piece allocated yet. The *goal node* is then defined as the state with all the pieces allocated on the raw material and that has the minimum total waste or equivalently the maximum raw material utilization. Following these definitions, a *leaf node* - which may or may not be a goal node- is a state with all pieces allocated, but does not necessarily have the property of minimum total waste. Figure 3.2 illustrates the proposed problem representation.

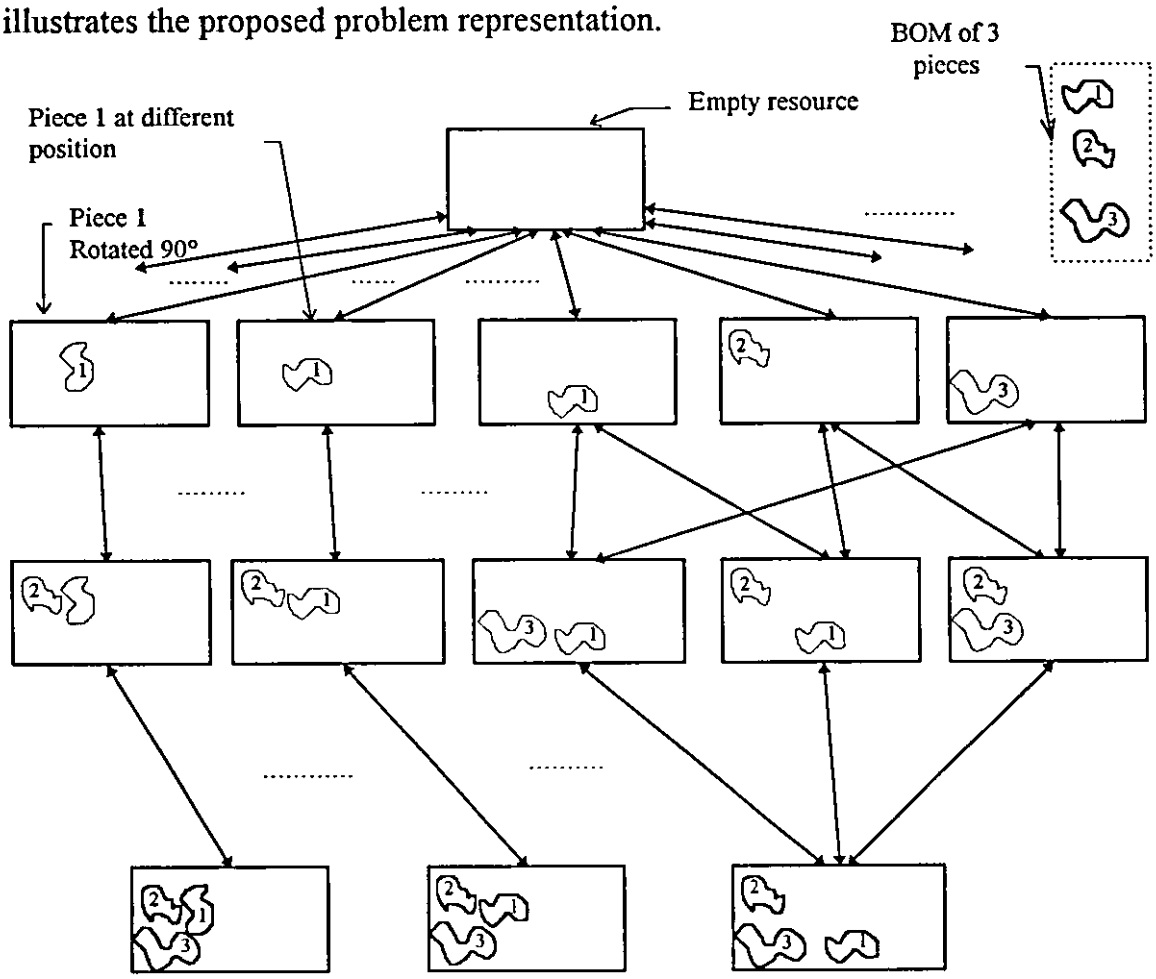


Figure 3.2 Representation of irregular shapes nesting as a state space graph. A very small portion of the whole graph of three pieces allocation is shown. One piece is tried at two different legal positions, and at two different possible orientations

This representation introduces an infinite state space. Furthermore, it is apparent from Figure 3.2 that such a representation has cycles where it is possible for a path to loop endlessly through a sequence of nodes.

3.5.2 Simplifying assumptions

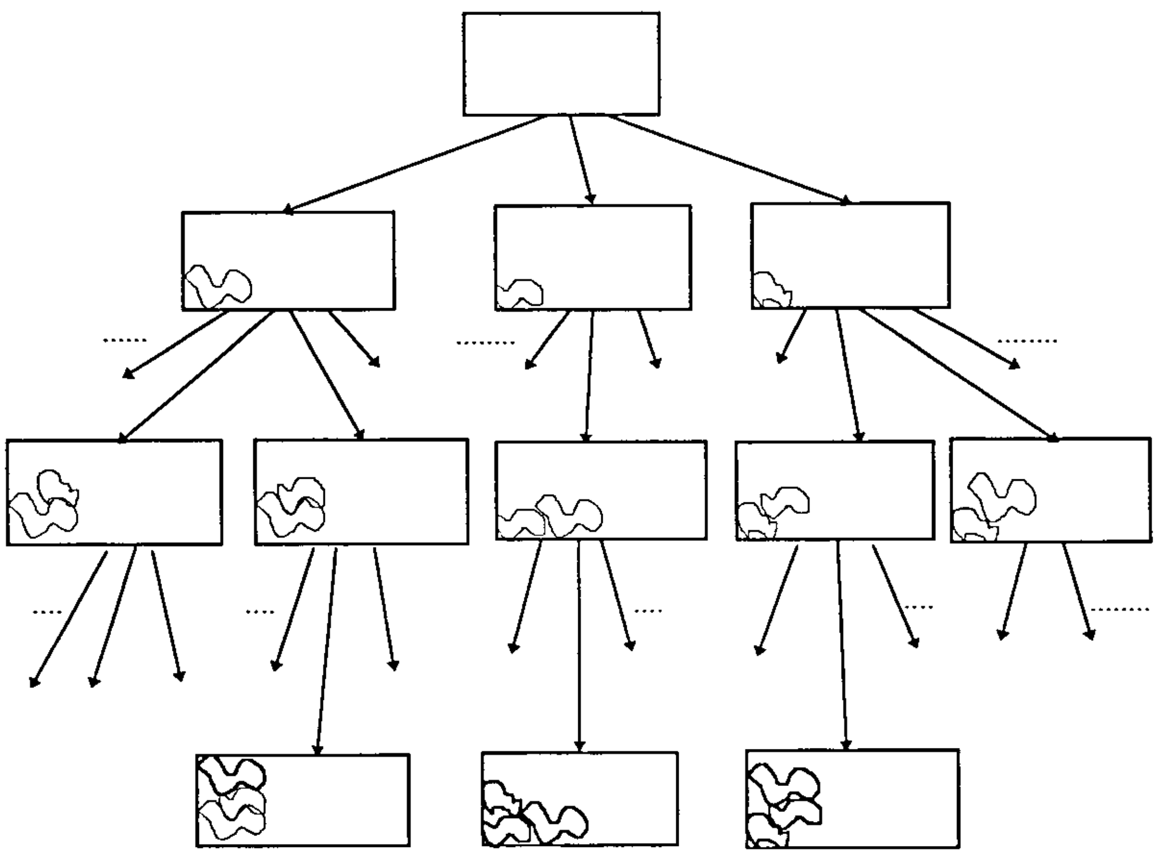
To overcome the deficiencies in the proposed representation, the following simplifications and restrictions are introduced:

1-When trying to position or orient any given piece at any stage of the problem solution a pre-specified positioning and orienting increment step is used. This converts the positioning and orienting processes from continuous to discrete space. Thus the state space is drastically reduced and becomes finite, but still explosively growing as will be discussed later. This assumption is with practical compliance, since eventually for any automated cutting facility -as the intended flame cutting machine- there is a limited positioning precision expressed usually by its basic length unit(BLU).

2-The placement of pieces should start at the lower left corner of the raw material. This restriction introduces directional properties on the whole allocation process; where the general flow of allocation becomes from left to right, and from bottom to up, thus systematic allocation could be realized. In most of the previous researches such an assumption has been introduced, even in the rectangular shapes problems - as previously presented in chapter two-.

3-Then it progresses by adding a subsequent piece that should touch the boundary of a previously allocated one. This restriction tends to pack the pieces in such a way that tries to reduce the introduced waste at any allocation stage.

The last two restrictions eliminate loops in the representation, and the graph is converted into a *tree* as shown in Figure 3.3, with some expense, where redundancy is introduced (i.e. some nodes could be repeated). Thus the problem representation becomes a finite state-space tree as Figure 3.3 shows, and the solution space becomes reduced drastically, with no more guarantees of optimality on the solution to be obtained.



Figure(3.3): The irregular shapes nesting as a state space tree.
Part of the tree is shown only.

The problem solution becomes the process of finding the sequence of allocations to be performed on pieces until all the pieces are placed, and at the same time waste is kept at minimum possible levels. This process is equivalent to finding the *path* in the state-space tree that will lead to the goal state, which is expected to be one of the leaf nodes.

3.6 Insight on problem complexity

From the previous problem representation an attempt to give insight on its complexity is presented as follows:

Assuming that the allocation process consists of three decisions to be made: the first one is to answer the question "Which piece to select and allocate at each node?", the second is to answer the question "Where to position that piece?", the third is to answer the question "In what orientation to place it?", then denoting the size of the bill of material (BOM) to be n different shapes. At initial node (zero level of the tree) no pieces are allocated, and any of the n unallocated pieces has an equivalent opportunity to be selected. Assuming that by using a certain oracle a piece has been chosen, and placed somewhere in some orientation satisfying all problem constraints. At the next tree level there will be (n-1) pieces left to select from. At the subsequent level (n-2) pieces will remain to select from, and so on. Thus at each level one piece of the remaining unallocated pieces is selected. Consequently this will result in an ($n!$) process complexity, for just the selection of piece process. As for the positioning rule, a piece could be placed any where within the boundaries of the raw material on a continuous scale for positioning and orientation. But making use of our previously presented simplifying assumptions, the number of possible positions for a given piece becomes dependent on:

- the size of the piece to be placed,

In the suggested representation the depth of the tree is $d = n$ where all pieces become allocated. And since the estimated number of nodes at any tree depth(d) is given by: number of nodes at level (d) = $\bar{b}^d$ (3.2)

Then the estimated number of paths in the tree = number of nodes at $d= n$ and is:

$$\bar{b}^n = (n+1)^n \qquad (3.3)$$

2)Another estimation of the number of paths that could be generated in this representation could be obtained as follows:

At $d=0$ we have $2n$ possible choices; for each we get $2(n-1)$ at $d=1$; then for each of them we have $2(n-2)$ at $d=2$...continuing until $d=n-1$ then the number of paths developed becomes:

$$2n*2(n-1)*2(n-2)*...*2(n-(n-1)) = 2^n.n! \qquad ...(3.4)$$

Table(3.1) illustrates the growth rates of such space as a function of the size of the problem.

Realizing that in our case where the number of possible positions and orientations are much larger, then it is apparent that the solution space of this problem is extremely large, which prohibits the consideration of any exhaustive search approach from being used to solve the problem, even with recent computational technology. Thus a heuristic search method is justifiable, and it would prune as much as possible of the solution space and direct the search towards most promising paths.

Table(3.1): Estimated number of paths in the state space as a function of the number of different shapes in the BOM.

BOM size (n)	$(n+1)^n$	$2^n n!$
1	2	2
5	7776	3840
10	$2.6\text{x}10^{10}$	$3.7\text{x}10^{9}$
15	$1.2\text{x}10^{18}$	$4.2\text{x}10^{16}$
20	$2.8\text{x}10^{26}$	$2.6\text{x}10^{24}$

3.7 A heuristic solution approach

A heuristically informed search approach makes use of domain specific knowledge to reduce the search space. Knowing more about a problem, and utilizing such knowledge, can significantly improve the chances of reaching a goal state more quickly, rather than considering a large number of states and rejecting many of them before reaching that goal.

To devise a heuristic search we need to define a measure or an evaluation function, that would estimate how close a given node or state to the goal. Such a measure would direct the search such that most promising paths are visited first, which hopefully will lead faster to the goal. Such heuristic approach is supposed to perform well in practice, but provides no guarantees on the optimality. It is generally judged to perform well more than not, since it is based on human intuitive thoughts and rules of thumb that practically have proven successful in most cases.

3.7.1 The proposed heuristic approach

The suggested heuristic approach is based on a "*Hill Climbing*" technique [38], in the sense that at any visited node, only the child with best value of the evaluation function is kept and further explored. By no means such an approach guarantees a global optimal solution, furthermore, by using this approach a single path is elected in the solution process that is hopefully of good quality i.e. at complete allocation it gives minimum waste. Such a search is basically of the *breadth-first* search approach. It is a data driven (forward chaining) search.

The steps of the proposed approach are given as follows:

1. Allocation starts at the lower left corner of the raw material.

2. The inference rules are applied on all unplaced pieces, and thus all possible succeeding allocations are generated as follows:

a. For each unplaced piece all possible orientations (including flips) are determined.

b. For each orientation of each unplaced piece all possible positions are searched for by sweeping the piece along the boundary of the exposed-part-for-allocation of the previously allocated pieces except for the corner piece. The sweeping is made in increment steps with the increment prefixed at the beginning of the algorithm. At each position checking are made to make sure a piece is being placed in a legal position.

3. The evaluation function is applied on all the generated nodes of step 2.

4. The one with the best evaluation function is selected. Any ties are broken according to a number of conflict resolution rules to be presented later.

5. The selected piece is removed from the list of unallocated pieces and placed in the list of allocation.

6. The added shape is combined with previously placed shapes into a temporary hypothetical shape (the reason for this combination is explained later).

7. Any generated waste region is collected and considered for allocation by the largest unallocated piece that may fit within this region. Where a waste region is defined as the closed region surrounded by allocated pieces and/ or sheet boundaries.

8. Steps 2 to 7 are repeated until all pieces are allocated.

It is worth mentioning that backtracking is not considered in this algorithm; even though, it's believed that it would enhance the performance of the approach. The reason for that is mainly due to the large amount of information, that are needed to be saved for a visited node so that backtracking could be made for it from any other descendant node in the tree. Therefore, optimality is not guaranteed, but hopefully a good quality solution could be obtained in a fast manner. Thus the solution is greatly affected by the quality of the evaluation function to be used.

3.7.2 The evaluation function

Since it is required to minimize the overall waste, the optimal path to be searched for, is the one with the total minimum waste. To preserve the admissibility of a heuristic search (ability to reach an optimal solution if it exists) an evaluation function of the form $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$ should be used[38]. Where n *is* a node in the state space, $\hat{g}(n)$ an estimate of $g(n)$ the least cost from root node to node n, $\hat{h}(n)$ an estimate of $h(n)$, the least cost from node n to the goal node. And if $\hat{h}(n)$ $h(n)$ then the heuristic is admissible, with $\hat{f}(n)$ the total estimated waste at node n. If at each node visited $\hat{f}(n)$ could be calculated, then the node with the least cost would be chosen for further exploration. A good choice of $\hat{g}(n)$ is suggested by [33] to be the waste accumulated from the initial node until the reached node n. Unfortunately, a good estimate of $h(n)$ could not be easily derived, especially with the requirement of being a lower bound on $h(n)$. Therefore, $\hat{h}(n)$ is kept at zero on the expense of obtaining generally non optimal solution. However, a solution which is hopefully described as being "good" and efficient in terms of computational time and storage requirement still could be achieved.

Rules of thumb have been suggested through out literature for making the selection decision at any stage in the solution process, such as: maximum area, maximum perimeter, maximum irregularity, and their minima also were tested in [2]. Maximum covering of waste in the minimum enclosing rectangle of a previous piece was suggested by [25]. Maximum length was used by a commercial package at a local industry [39].

Since a heuristic rule is only an informed guess of the next step to be taken in solving a problem [37], which is often developed by trial and error; based on domain specific knowledge, in conjunction with a number of reasonable approximations, and simplifications [10]. Since a heuristic algorithm could be thought of as consisting of two parts: the heuristic measure (evaluation function), and an algorithm that uses this measure to direct the search in the state space, a number of possible evaluation functions that tend to minimize the waste , or equivalently, maximize the utilization at each selected node could be suggested as follows:

- *Minimum enclosing rectangle* (MER) area of a collection of pieces,
- Minimum waste in the MER of a collection of pieces,
- The maximum ratio of added-piece-area to the waste-added-in-the-MER,
- Functions as previous ones, but rather than using the MER, the minimum *convex hull** (MCH) could be suggested.

All of these functions are based on enclosing the irregular collection of shapes into a more regular shape, that would reduce the computational burden; however, still work on satisfying the objective of waste minimization, by efficiently packing the pieces within the suggested enclosure.

An evaluation function similar to the one that has proven to be successful in solving the Knapsack problem[29] is adopted here, but with added biasing terms towards desired properties in the solution scheme, which are captured from the human approach in solving the problem. The evaluation function suggested is as follows:

$$\bar{f}(n) = \alpha \cdot \frac{A_n}{W_n} + \beta \cdot g(a_n) \tag{3.5}$$

* The convex hull of a two dimentional shape is the minimum area enclosing convex polygon.

Where n is as previous a given node or state in the search space.

α and β are positive real numbers that indicate the contribution of each of the terms in the evaluation function.

$A_n = \sum a_i$, where A_n is the sum of areas of the so far allocated pieces along the traversed path till node n, and a_i is the area of a single shape i.

$W_n = \sum w_i$, where W_n is the accumulated waste along the path till reaching node n, and w_i is the added waste at each previous node along the same path till node n. This waste is calculated in the MER of the collection of so far allocated pieces.

$g(a_n)=$ is a biasing function of piece area; where it is a general practice to try to allocate larger pieces before smaller ones during the allocation process. Also since any waste regions that are generated during the allocation process are going to be collected, which are expected to be of relatively small area, it would be more effective to preserve smaller pieces for such regions.

The first term in the evaluation function tends to maximize the useful local area and at the same time tends to reduce the generated local waste (added waste) at each allocation step.

3.7.3 Conflict resolution rules

During the allocation process the following tie-breaking rules are proposed:

1. For the same piece if the evaluation function is equal at different possible positions, the left most lower position is selected. Since this would tend to pack the piece more efficiently by preventing waste regions from being created to the left of or below added pieces, thus preserves the general allocation flow.

2. For different pieces if the evaluation function is equal, then the one with the largest area is selected. This rule is with accordance with the general practical practice.

3.8 Computer implementation aspects and considerations

A computer software package has been developed for implementing and testing the proposed approach. The language used for implementation is Borland Turbo C® v3.1. The selection of this programming language was dictated by the requirement of relatively complicated data structures to represent the different aspects of the problem solution, the need for high speed processing, the need for efficient graphical visualization tools. PROLOG® could have also been used, but the need for efficient numerical and graphical manipulations, in addition to high processing speed, and large memory models, suggested the use of C language.

It is a very important practical point to clarify that the full tree for even a moderate sized problem is surprisingly very large -as discussed previously- so that it could never be fully represented at one time in the computer's memory. Therefore, at any node or state in the solution process the subsequent nodes(*children*) are generated as required by applying the transition rules each time a node is visited.

3.8.1 Shape representation

The interaction among the shapes to be allocated and between any shape and the raw material on the other hand, is only governed by the shape's outline or boundary. Therefore, a suitable representation [31]for shapes that is efficient in terms of computer storage requirements, and that is effective in describing the boundaries of shapes is based on vector graphics as follows: each shape is represented by a *polygonal* approximation. A polygon is described by a structure consisting of the following data:

-Number of edges/vertices in the polygon.

-A list of x and y-coordinates of each vertex. Given in a counter clockwise (CCW) direction. And are relative to a local polygon reference point that is selected arbitrarily.

- The x, y coordinates of an arbitrary reference point that could be one of the polygon vertices or may not.

- A reserved field for the area of the polygon to be calculated by the program.

All the shapes that are dealt with are assumed to be simply connected (i.e. they don't contain any internal features), thus they may be *convex* or *concave* polygons.

3.8.2 Two dimensional transformations

The positioning of a shape is done by changing the values of the reference point coordinates, and accordingly determining the actual positions of the polygon vertices.

-Translating a shape is thus simply achieved by adding/subtracting the increment step to/from the reference point coordinates.

-Rotating a shape is defined about its reference point by the two dimensional transformation equations (see Figure 3.6):

$$\left.\begin{aligned} \bar{x}_i &= x_i\cos(\theta) - y_i\sin(\theta) \\ \bar{y}_i &= x_i\sin(\theta) + y_i\cos(\theta) \end{aligned}\right\} \quad (3.6)$$

Where θ is the angle of rotation.

x_i, y_i : local coordinates of the ith vertex before rotation.

$\bar{x}_i, \bar{y}_i$: local coordinates of the ith vertex after rotation.

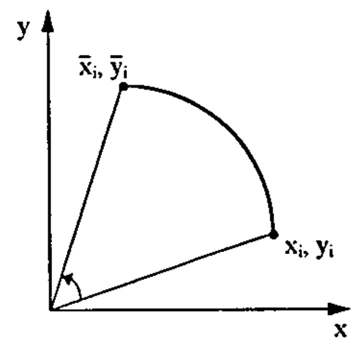


Figure 3.6 Plane rotation of a point.

-Flipping a shape about the horizontal axis is done by changing the sign of the y-coordinate of each vertex in the shape, and flipping about a vertical axis is done by changing the sign of the x-coordinates.

3.8.3 Combining shapes

A useful simplifying tool for positioning of pieces is the combination of successively allocated pieces onto a one temporary hypothetical shape; in that shape the parts of its boundary exposed to the part of the raw material not yet used is called the *profile*[33]. Figure 3.7 illustrates the idea. This combined shape is updated each time a new piece is tried for allocation or selected as a new allocated piece. The use of such a tool simplifies the calculations of added waste, by simply calculating the area of the combined shape, then subtracting from it the sum of areas of allocated pieces. Furthermore, the positioning of any new piece is done by pairwise comparisons between the combined shape and the shape to be positioned. The combined shape is thus a dynamically changing structure, depending on the boundaries of the shapes, therefore, it is implemented as a *doubly linked list* with dynamic memory allocation/deallocation during the process. Each node of this structure contains the following information:

-Vertex number, (x, y) coordinates of the vertex, pointer to next vertex, pointer to previous vertex.

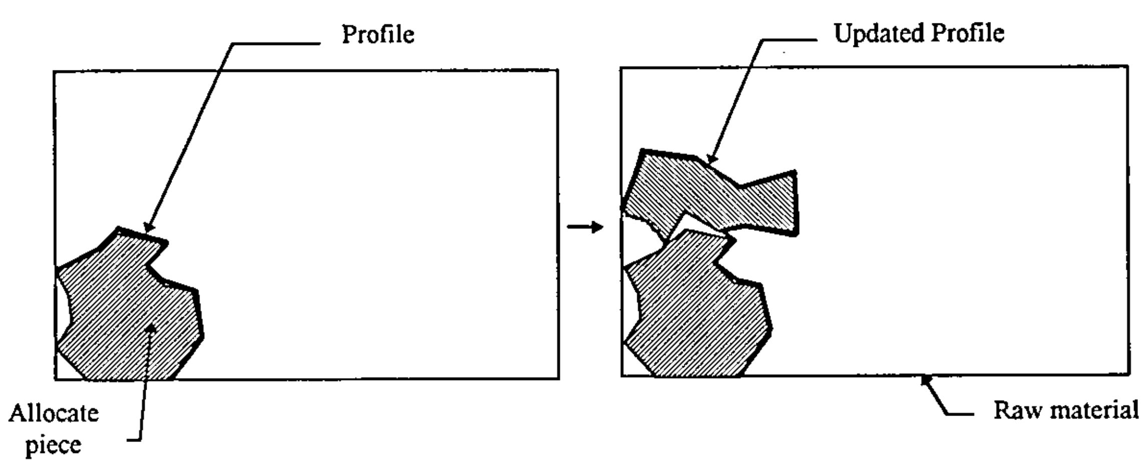


Figure 3.7 Profile definition and updating

3.8.4 Other implementation aspects

The unallocated pieces are kept in a list that is updated each time a piece is allocated. This list is implemented as a doubly linked list to speed up addition/removal of nodes from it. Also allocation list is used to keep data relevant to each allocation step; recording the added piece number, its x, y coordinates of the reference point, its orientation, the total waste so far achieved, total utilized area and the length of allocation.

Following is a brief description of the more significant functions developed to underlay the allocation process:

-`X_position(polygon[i],y)`:given the y-coordinate for the reference point of polygon[i], this function finds the x_coordinate for its reference point such that the polygon is legally placed from right within the resource.

-`Y_position(polygon[i],x)`: given the x-coordinate for the reference point of polygon[i], this function finds the y_coordinate for its reference point such that the polygon is legally placed from top within the resource.

-`Rotate(polygon[i],theta)`:rotates a polygon[i] theta degrees about its origin.

-`Flip_horizontal(polygon[i])`: Flips polygon[i] about a horizontal axis through its minimum y-coordinate vertex.

-`Area_poly(polygon[i])`:calculates the area of polygon[i]. The sum of signed areas of the trapezoids under the edges of the polygon gives its area.

-`Update_profile(polygon[i])`: updates the combined shape by including the segments of the boundary of the added polygon[i], and removing any segments that could never be used in further allocation, also it closes the profile with the edges of the raw material appropriately to form the combined shape.

-`Plot_allocation()`: provides a graphical visualization of the so far achieved allocations, through reading the allocation list at any desired node of the state space.

-`Evaluate(polygon[i])`:calculates the evaluation function value for the node in which polygon[i] is being added to allocation.

`-Waste()`:Collects the waste area produced at a given stage of solution and redefines it as separate raw material piece. The waste region is collected by searching for intersections (touching points) between the newly added shape and the previously combined shapes at each allocation stage.

`-Fill_waste(waste_region)`:Attempts to fill a given waste region by pieces not yet allocated. This is done by performing two dimensional sweeping - on shapes to be filled- over the waste region, trying to fit that piece legally, through moving it by a prefixed increment step in the vertical and horizontal directions, while continuously monitoring overlap, containment, and intersection conditions.

The program includes many other functions and displays information on memory usage and time during the solution process. In addition it shows intermediate trials of positioning of pieces graphically.

3.9 Testing the solution approach

In the following section testing the proposed solution approach is conducted. First the different effects of using various evaluation functions are demonstrated. Then the algorithm is applied on actual data obtained from local industries, and its results are compared to manual and automatic layouts achieved by these industries.

3.9.1 Showing the effects of using different evaluation functions

The effect of using some of the different evaluation functions mentioned previously is illustrated on a sample of six different shapes (n=6) as shown in Figure 3.8 to Figure 3.11.

In Figure 3.8 the MER is used as a selection criterion, where pieces to be placed are enclosed within their MER, then the one with minimum area is placed first, at the next step the combination of already placed piece and the piece to be added are enclosed within their MER, and the combination that gives the minimum area rectangle will dictate the piece to be fixed next. The procedure of combining pieces and enclosing the combination at each stage within a MER is repeated until all pieces are allocated. This criterion tends to pack shapes at positions and in orientations that would reduce the total area at each allocation step (*local minimum*), but not the overall area over the whole allocation process. It does not work directly on minimizing the waste added at each stage; however, the consequence of minimum area rectangle packing is expected to reduce the added waste at each stage. This function is, generally noted, to be biased towards smaller pieces.

To alleviate the drawbacks of the previous function the waste within the MER of combination of pieces could be suggested as a selection criterion. Figure 3.9 shows the allocation produced by this function. The waste is obtained by

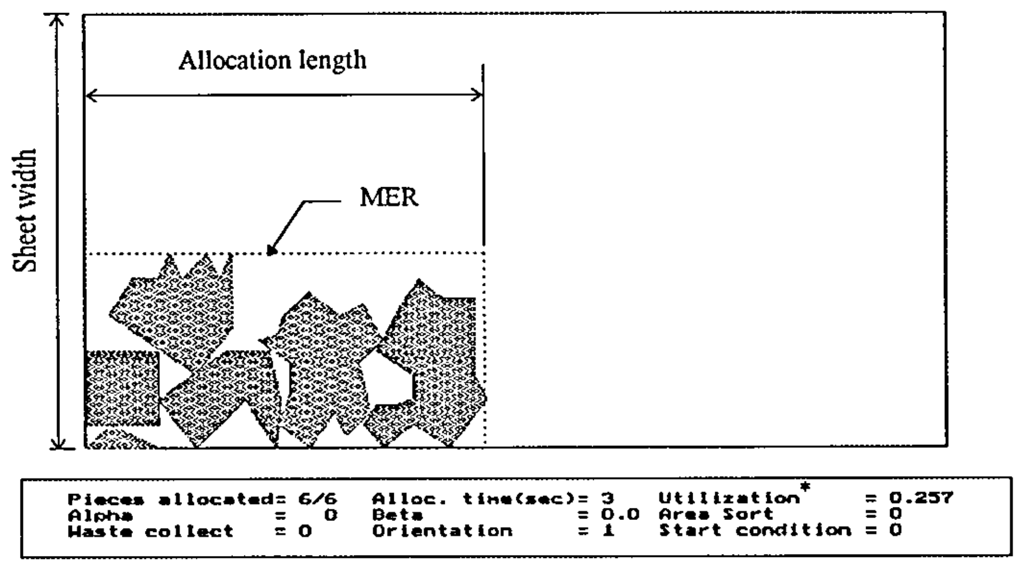


Figure 3.8 Effect of using MER area as an evaluation function on layout.

Utilization within the MER =57.3%.

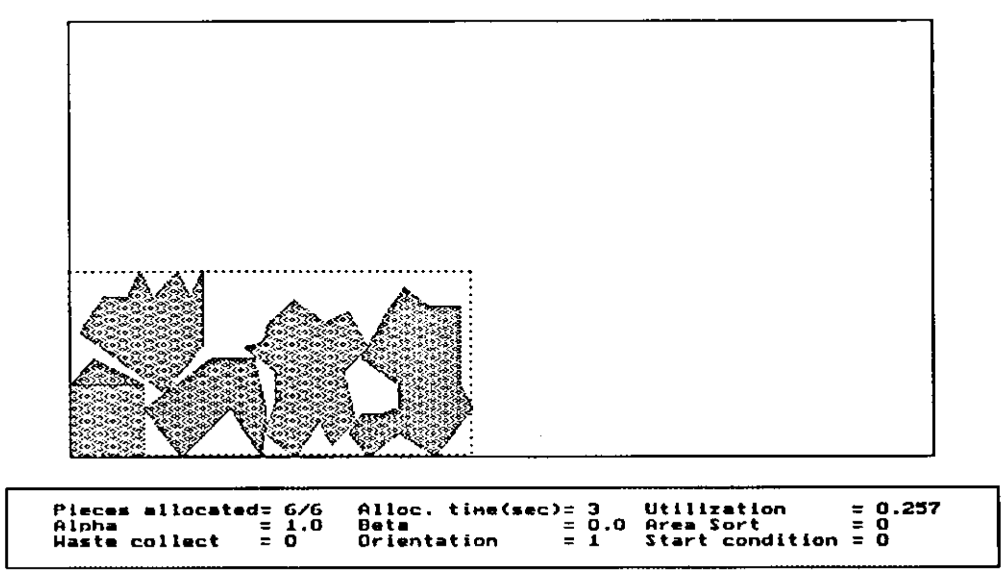


Figure 3.9 Effect of using MER waste as an evaluation function on layout.

Utilization within the MER =60.6%.

---

* Utilization values beneath each figure are calculated with sheet width and allocation length as the sides of the MER rather than using the actual sides of the MER.

subtracting the sum of areas of pieces within the MER from that rectangle area. The piece that when tested for allocation at certain position and in certain orientation, and produces minimum waste is the one that is selected for allocation at each stage (i.e. the function will force the position and orientation of a piece that will yield minimum waste in the MER). Although, good packing is expected to be achieved, the waste used in this evaluation function is an overestimate of the actual waste being added at each stage. It is also a local minimization function, since it works on added waste at each stage. The function is biased towards larger area pieces when the area of MER for two different pieces is equal.

Since the general practice in solving this problem is to try to pack the larger pieces earlier in the allocation process, and at the same time trying to minimize the waste, the ratio between the useful area (some of allocated pieces areas) and the waste area within the MER could also promise good results. Figure 3.10 shows the allocation obtained by this criterion, with the added modification of considering the MER width to be the raw material width. This modification forces the allocation to fill across the width of the sheet first, then continue along its length. The consequence of this is to try to minimize the length of the MER at later stages rather than minimizing the waste; which biases the allocation towards smaller pieces in general, or more precisely, pieces that have less overall span along the sheet length than along its width. This could be resolved by adding the area biasing term suggested previously in this chapter and will be shown in the examples to be presented later.

Figure 3.11 Effect of using CH area as an evaluation function on layout.

Utilization within the CH =53.0%.

3.9.2 Testing the allocation approach using actual data

To test what has been achieved so far, a number of typical layouts from two different local manufacturers working in sheet metal and textile industries are obtained.

(a) The example from the sheet metal industry is a handmade nesting of twelve identical shapes shown in Figure 3.12. The utilization achieved by manual allocation is 63.9%. While the different allocations for the same example as obtained by the proposed automatic algorithm in this project, using different parameters values for the suggested evaluation function (Equation 3.5), are shown in Figures 3.13-3.15.

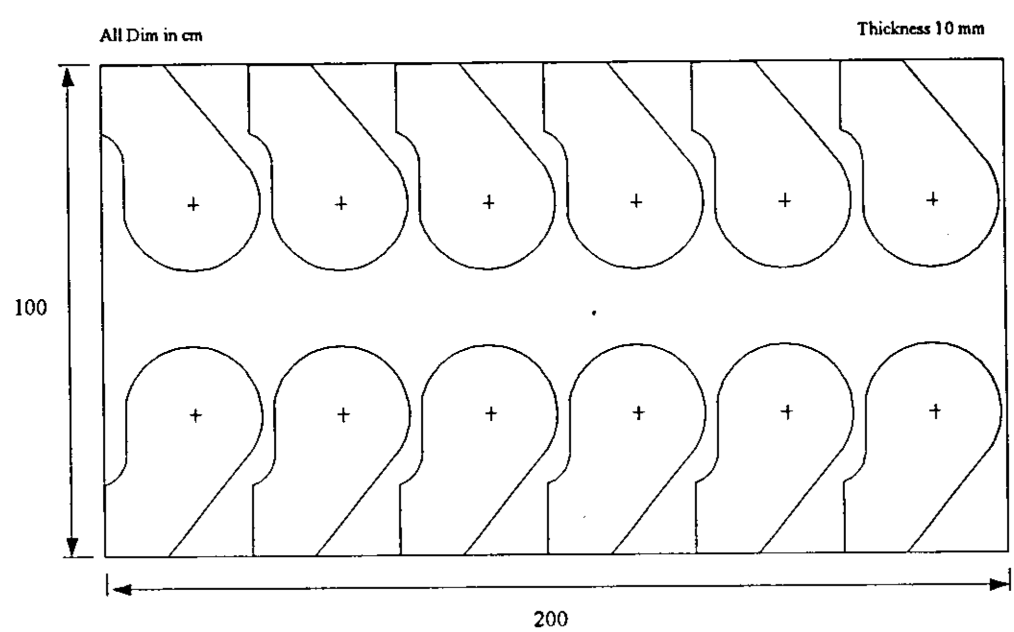
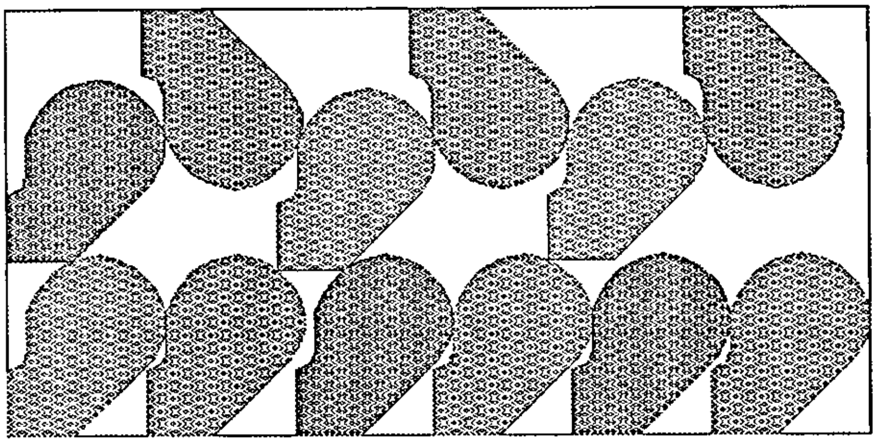


Figure 3.12 Handmade layout (nesting) of 12 identical irregular shapes

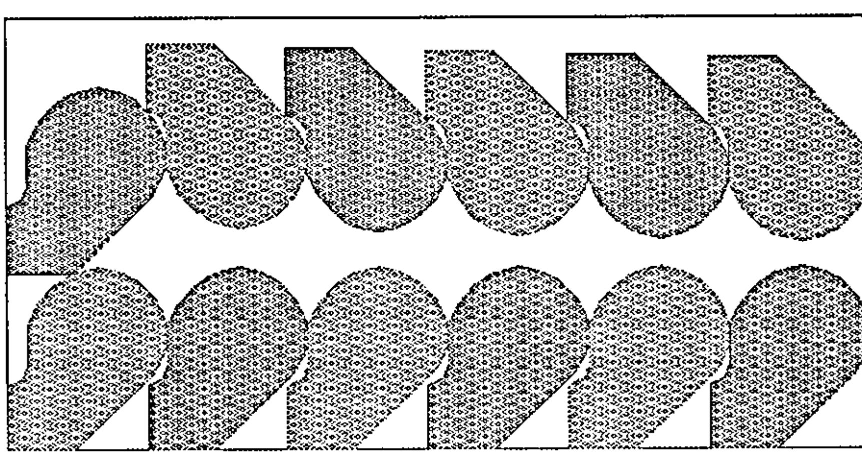
Utilization is 63.9%.



```
Pieces allocated= 12/12 Alloc. time(sec)= 82   Utilization      = 0.651
Alpha            = 1.0  Beta             = 0.0  Area Sort        = 0
Waste collect    = 0    Orientation      = 0    Start condition  = 0
```

Sheet width= 100 cm. Allocation length=196.3 cm.

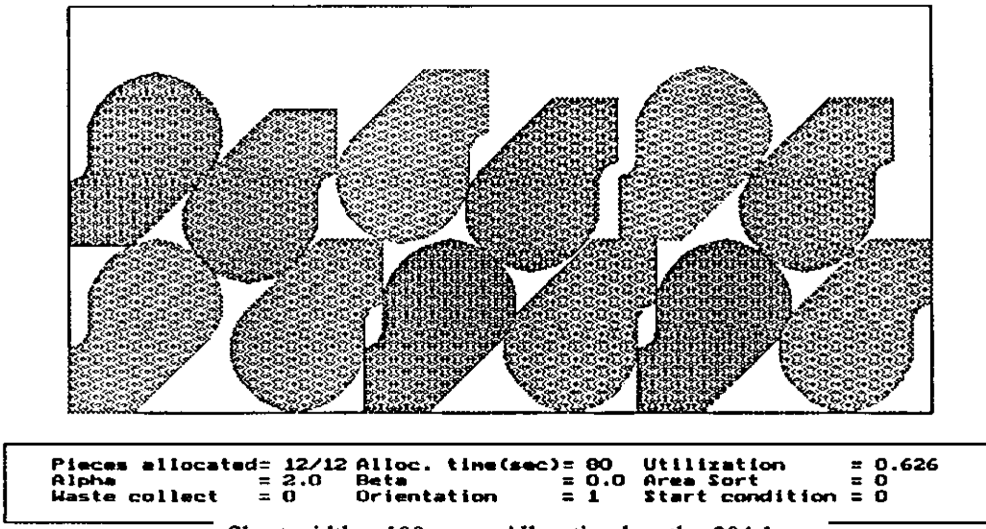
Figure 3.13 Layout produced by the proposed algorithm for the same set of shapes of Figure 3.12. Utilization is 65.1%. α=1, β=0. Horizontal flip permitted



```
Pieces allocated= 12/12 Alloc. time(sec)= 96   Utilization      = 0.659
Alpha            = 2.0  Beta             = 0.0  Area Sort        = 0
Waste collect    = 0    Orientation      = 0    Start condition  = 0
```

Sheet width= 100 cm. Allocation length=193.9 cm.

Figure 3.14 Layout produced by the proposed algorithm for the same set of shapes of Figure 3.12. Utilization is 65.9%. α=2, β=0. Horizontal flip permitted



Sheet width = 100 cm. Allocation length =204.1 cm.

Figure 3.15 Layout produced by the proposed algorithm for the same set of shapes of (Figure 3.12). Utilization is 62.6%. α=2, β=0. Orientations permitted θ=0°, θ=180°.

Since the used shapes are identical, the second term of the evaluation function (Equation 3.5) is not expected to affect the allocation process. Therefore, β is kept zero in previous trials. Also changing α will not affect the results for the same reason. In Figure 3.14 instead of taking the width of the sheet as one of the sides of the MER, to force filling the width of the raw material first, the actual MER of allocated shapes was used. In Figure 3.15 instead of permitting the horizontal flip of shapes, two orientations are permitted; at θ=0°, and θ=180°. The cutting tool diameter offset mentioned previously was taken into account in these examples (the effective radius of the cutting flame is added as an offset to the dimensions of the shape) . Waste collection and filling has no effect

effect here, since the areas of waste regions produced are much less than the areas of the shapes.

(b) Two nesting examples from the textile industry are obtained. The first has a BOM of 12 different irregular shapes (n=12), while the other BOM consists of 24 different shapes. The first example solution by a manual approach is shown in Figure 3.16. A commercial computer interactive software package was used in performing the layout[39]. While an automatic layout of the same problem, produced by an automatic module of the same software package, is shown in Figure 3.17. The allocation heuristic used is based on nesting pieces in decreasing shape area order, combined with other priority rules. The results of laying out this sample BOM by the proposed algorithm in this research are presented in Table 3.2 for different values of α and β.

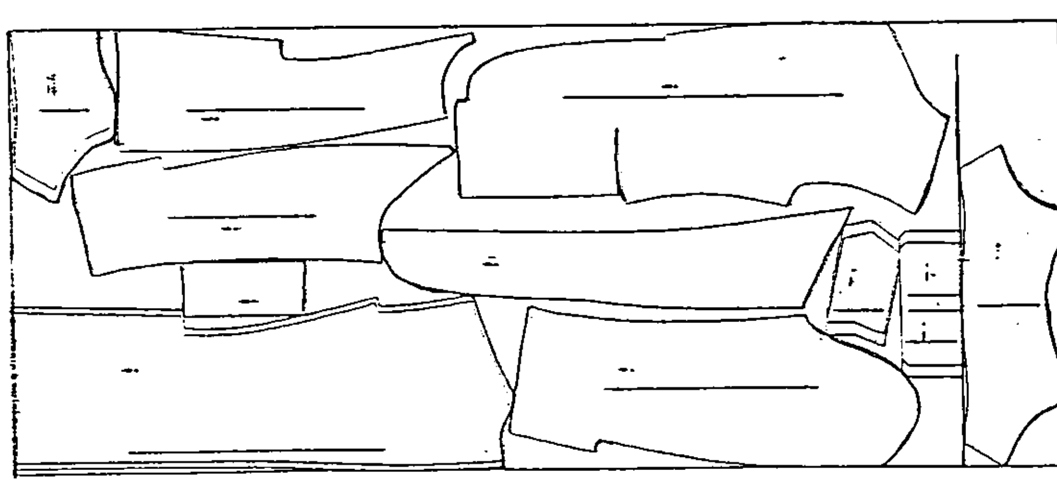


Figure 3.16 Manual solution of textile example 1 by interactive computer graphics package. Utilization =77.6%. Allocation time =5~10 min. for experienced operator.

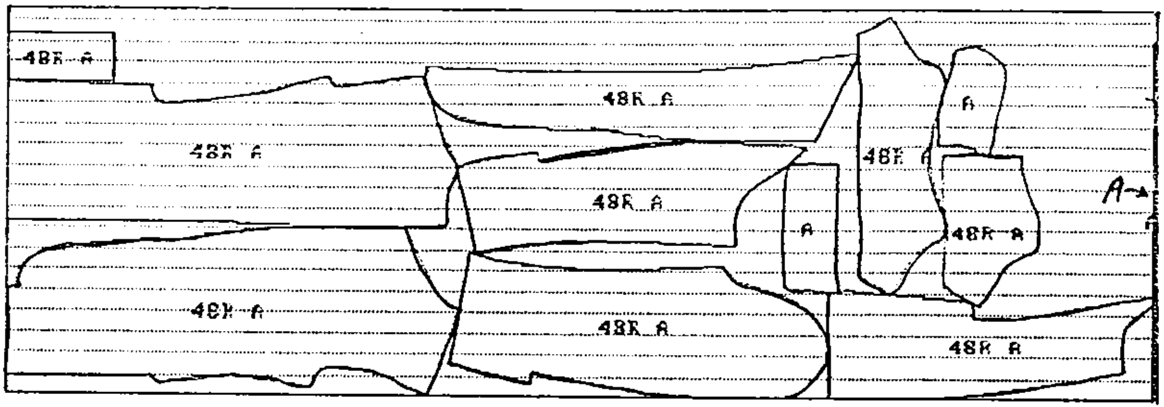


Figure 3.17 Automatic solution of textile example 1 by a commercial nesting package.

Utilization = 64.52%. Allocation time = 6 sec.

Table 3.2 Effect of changing α and β on the utilization rates for textile example 1. (BOM size = 12).

Trial NO.	α	β	Utilization	Figure No.
1	1	0	56.7%	-
2	0	1	58.2%	-
3	1	1	56.1%	-
4	1	2	72.5%	Figure 3.18
5	1	3	59.7%	-
6	1	5	59.7%	Figure 3.19
7	1	7	66.1%	-
8	1	10	56.7%	-
9	1	20	60.8%	-
10	1	50	54.5%	Figure 3.20
11	5	1	68.7%	Figure 3.21
12	10	1	68.7%	-
13	20	1	68.7%	-
14	50	1	68.7%	-

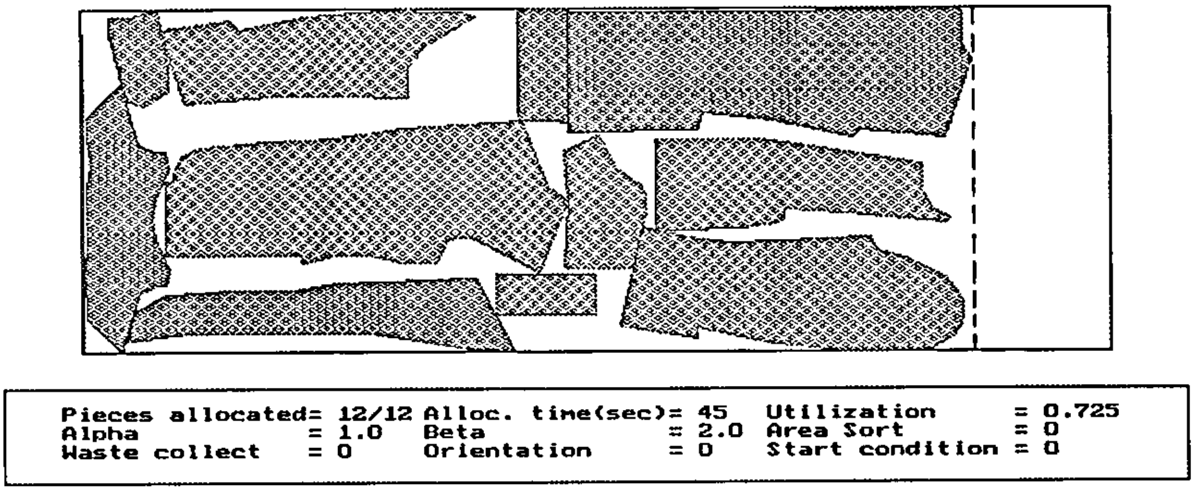


Figure 3.18 Layout obtained for textile example 1 with α=1, β=2.

Utilization =72.5%.

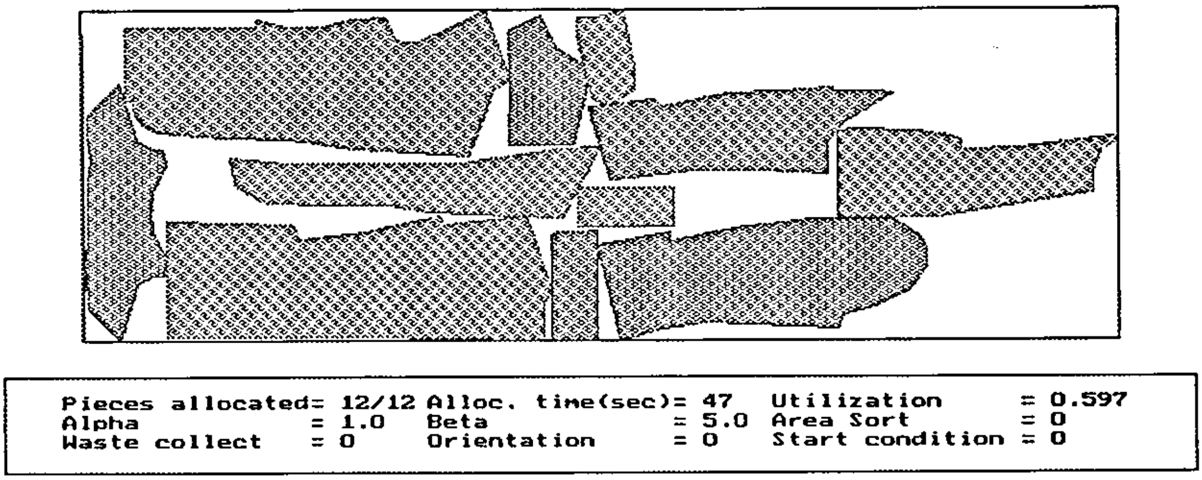
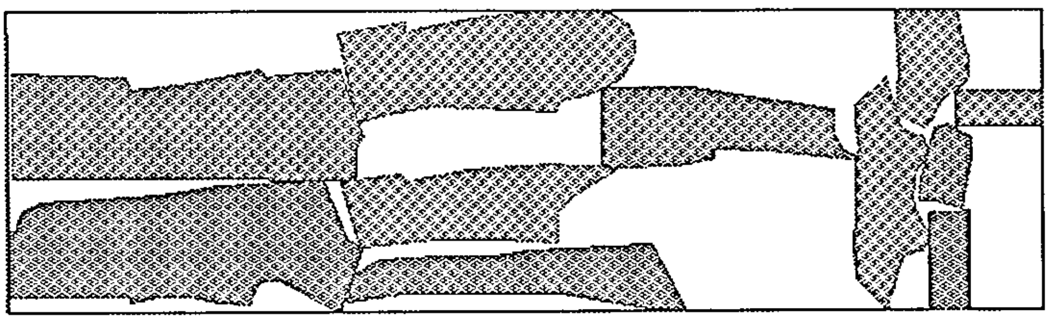


Figure 3.19 Layout obtained for textile example 1 with α=1, β=5.

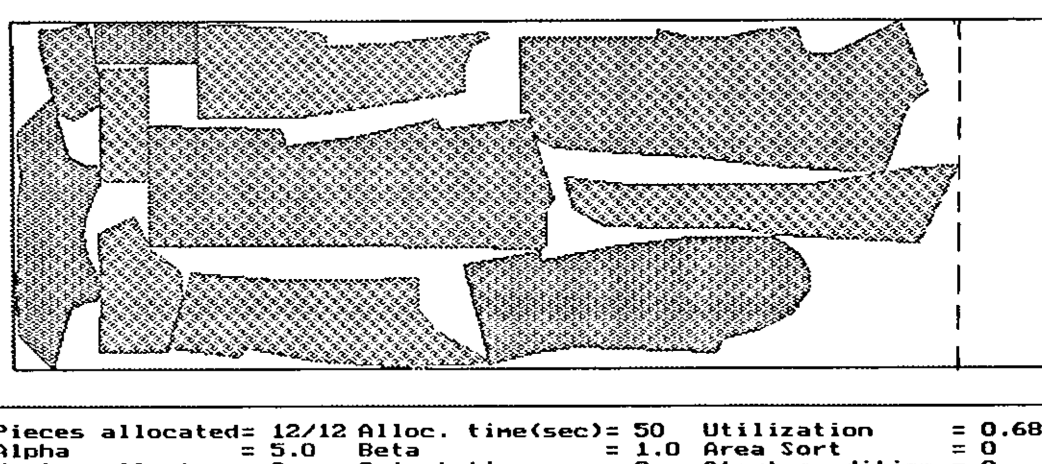
Utilization =59.7%.



Pieces allocated= 12/12 Alloc. time(sec)= 43 Utilization = 0.545
Alpha = 1.0 Beta = 50.0 Area Sort = 0
Waste collect = 0 Orientation = 0 Start condition = 0

Figure 3.20 Layout obtained for textile example 1 with α=1, β=50.

Utilization =54.5%.



Pieces allocated= 12/12 Alloc. time(sec)= 50 Utilization = 0.687
Alpha = 5.0 Beta = 1.0 Area Sort = 0
Waste collect = 0 Orientation = 0 Start condition = 0

Figure 3.21 Layout obtained for textile example 1 with α=5, β=1.

Utilization =68.7%.

In these trials no waste regions collection has been performed, since size differences among pieces are not that significant (very small waste regions are produced that don't permit their filling); as will be demonstrated by the following results. An increment step of 0.5 units was employed for positioning pieces. Two orientations were permitted; these that are obtained by horizontal flipping of pieces, since this is a requirement in textile industry.

The effect of increasing β is obvious from the previous figures; it tends to delay the allocation of small area pieces until later stages of allocation, while trying to attract larger area pieces at earlier stages. Utilization of such effect could prove useful when waste collection and filling is employed. Since one would expect relatively larger waste regions among large pieces than their would be among small pieces or a mix of small and large pieces. Furthermore, the delay of allocation of small pieces would reserve these pieces for waste regions filling if any could be filled. It is also interesting to note that in Figure 3.19 most of the waste has been added at the last allocation piece which is, relatively, a large one.

The effect of increasing α is to balance the delay of small area pieces with added waste at each allocation step; i.e. if the waste to be produced by allocating a large piece is significant, then a smaller piece could be selected in favoring of less waste.

The effects of waste collection and filling at each allocation step could be seen by comparing the allocations of Figure 3.20 and Figure 3.22; where enhancement on the utilization is obtained as would be expected, but not that significant in this case due to small area differences as mentioned previously.

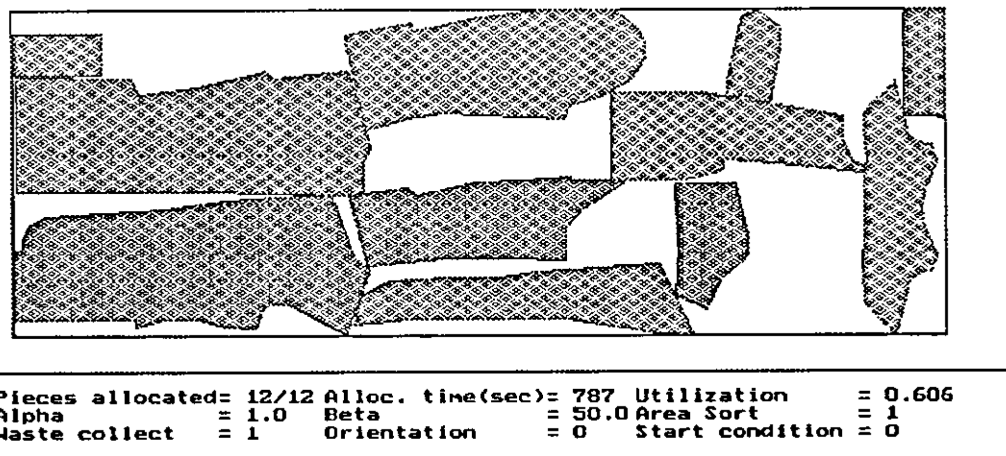


Figure 3.22 Layout obtained for textile example 1 with α=1, β=50. With waste collection and filling by a single piece. Utilization =60.6%.

The effect of permitting different orientations of pieces is illustrated by comparing the layouts of Figure 3.23 and Figure 3.24; where the orientation could be at $\theta=0°$ or $\theta=180°$. In textile industry these rotations are not permitted since textile fabrics have directional properties to be preserved, but in sheet metal industry such a restriction is not essential, and these examples are used to show the effect of permitting piece rotation on the algorithm performance irrespective of the intended application. It is noted that permitting such orientations would lead to generating new nodes in the solution space, And thereby enhancement on the utilization could be introduced as Figure 3.23 illustrates.

Some of the "good" allocation positions are noted to be missed (see Figure 2.23). This is probably due to the increment step used in the allocation, which if reduced to smaller values would not miss such positions. But this would be on the expense of the number of nodes to be generated in the solution space, and thereby on the allocation time, which would increase drastically.

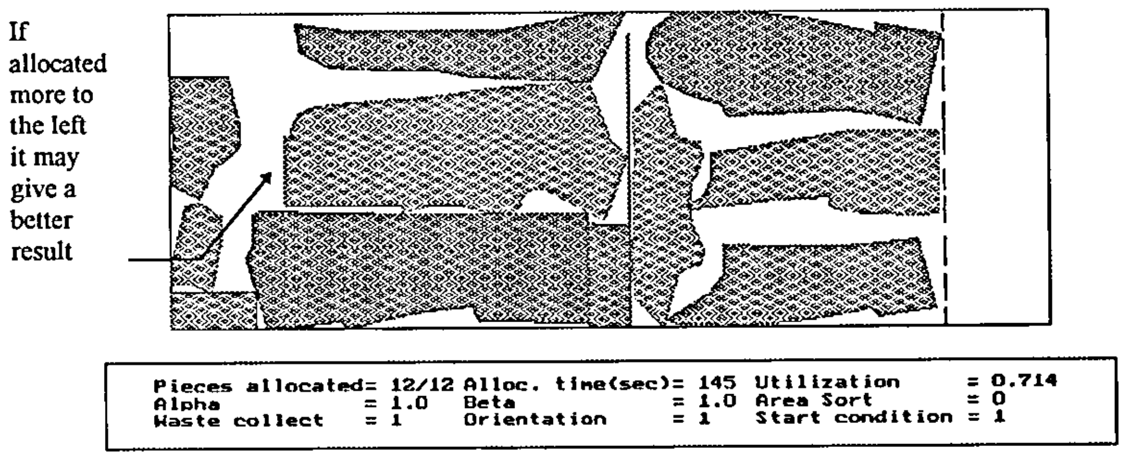


Figure 3.23 Layout obtained for textile example 1 with α=1, β=1. With waste collection and filling by a single piece. And orientations of θ=0° or θ=180° permitted. Utilization =71.4%.

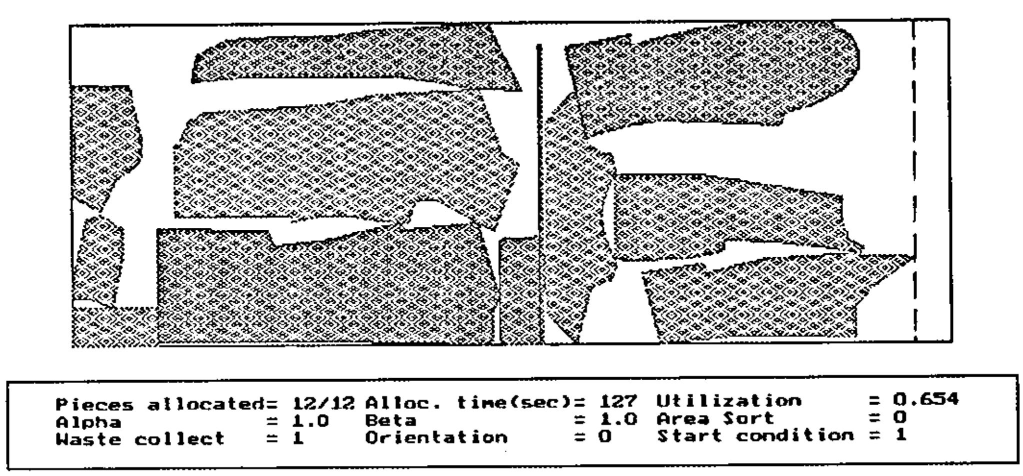


Figure 3.24 Layout obtained for textile example 1 with α=1, β=1. With waste collection and filling by a single piece. And orientations of horizontal flip permitted. Utilization =65.4%.

(c) As for the second textile BOM example (n=24), two automatic solutions by the commercial software package are shown in Figure 3.25 and Figure 3.26. Both automatic approaches are based on heuristics combined with priority rules. The first one is based on allocating pieces by decreasing area of shapes, while the other one is based on decreasing length of shapes. The manual layout for the same example was obtained with the exclusion of two shapes (shown in Figure 2.27); thus a BOM of 22 pieces. This is due to special requirements in textile industry, which obligate these exclusions. However, for getting appropriate comparisons with the results of the proposed approach in this research, they are going to be dealt with as two separate examples one of n=24, and the other with n=22.

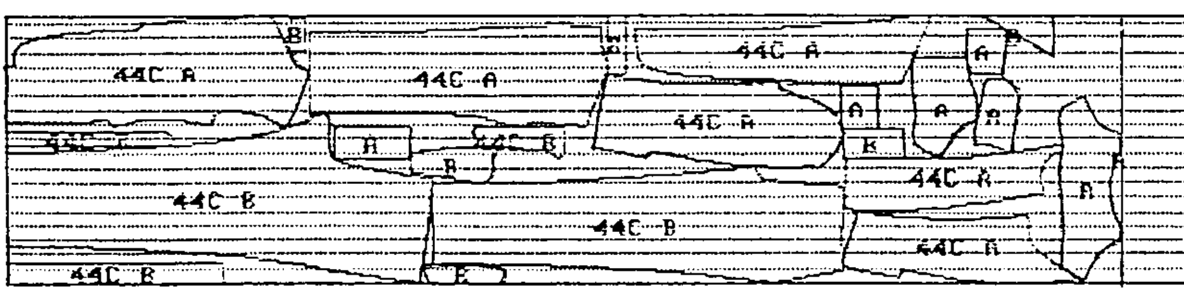


Figure 3.25 Automatic layout of textile example 2.a (n=24). The major heuristic rule is the decreasing area of shapes. Utilization 76.51%. Allocation time 42 seconds.

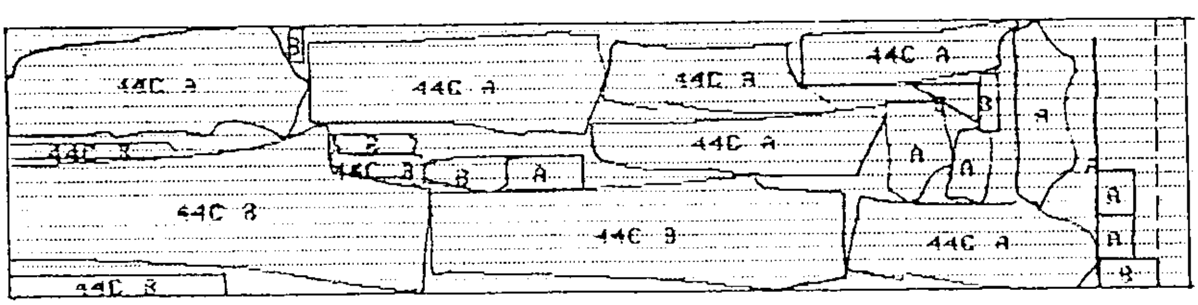


Figure 3.26 Automatic layout of textile example 2.a (n=24). The major heuristic rule is the decreasing length of shapes. Utilization 74.41%. Allocation time 42 seconds.

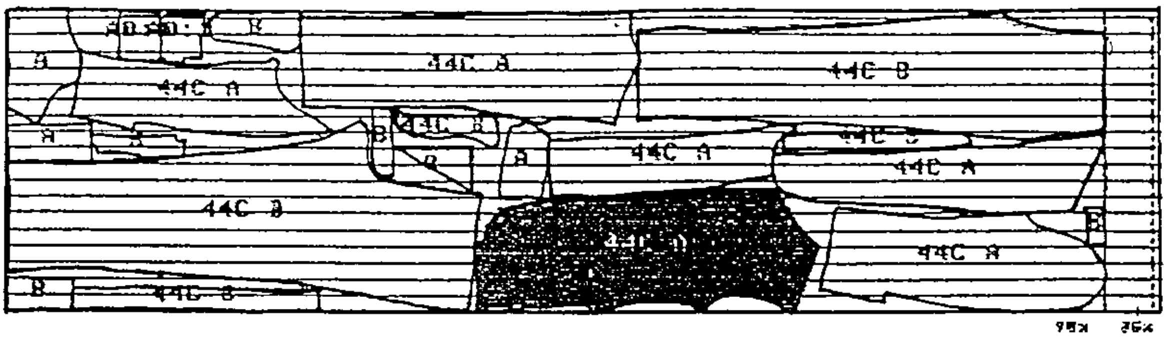


Figure 3.27 Manual layout of textile example 2.b (n=22). Utilization 84.25%. Allocation time 10~15 minutes.

Figure 3.28 to Figure 3.32 show the obtained layouts by using the suggested approach in this research for different values of α and β on the first example (n=24). Again the effect of increasing β is to delay the allocation of small area pieces. The effect of waste collection and filling is also shown by comparing Figure 3.29 with Figure 3.32.

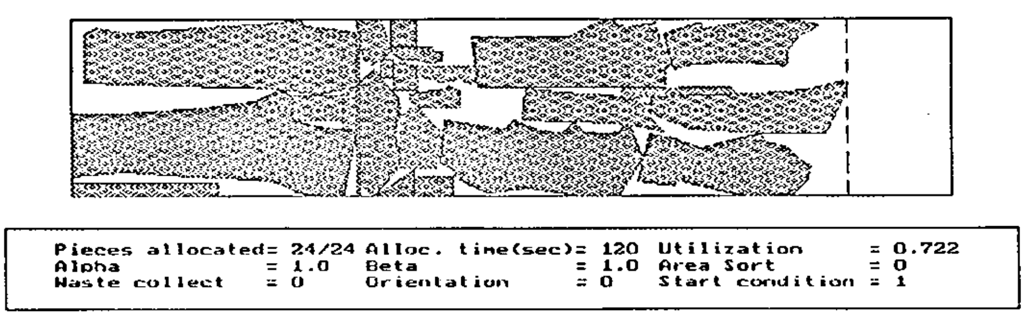


Figure 3.28 Obtained layout for textile example 2.a, α=1, β=1. Utilization 72.2%

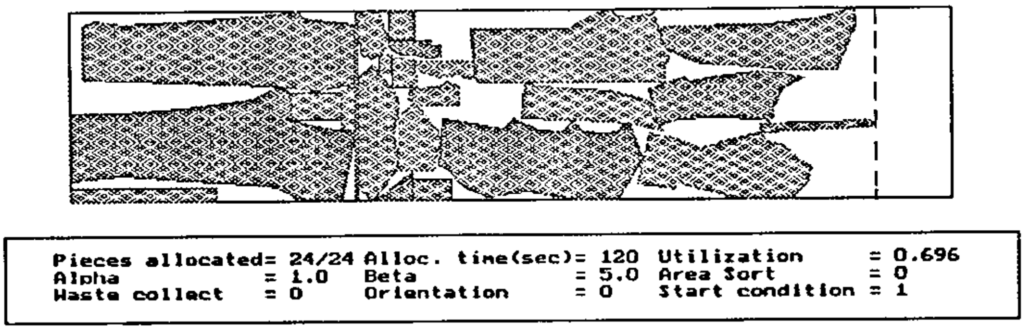


Figure 3.29 Obtained layout for textile example 2.a, α=1, β=5. Utilization 69.6%

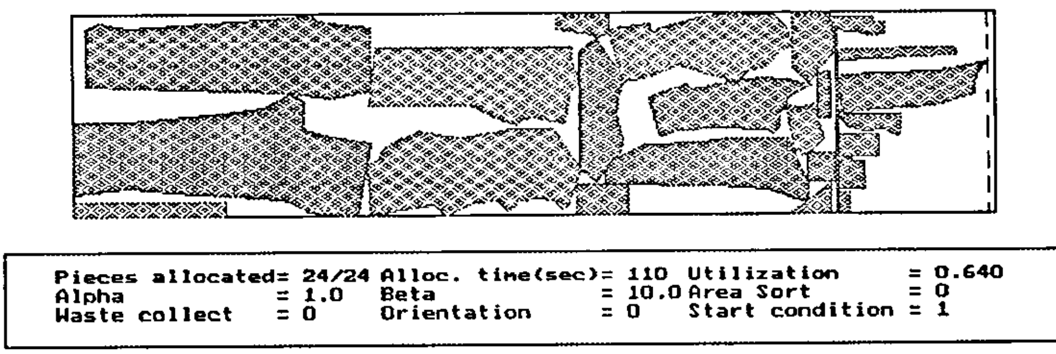


Figure 3.30 Obtained layout for textile example 2.a, α=1, β=10.
Utilization 64.0%

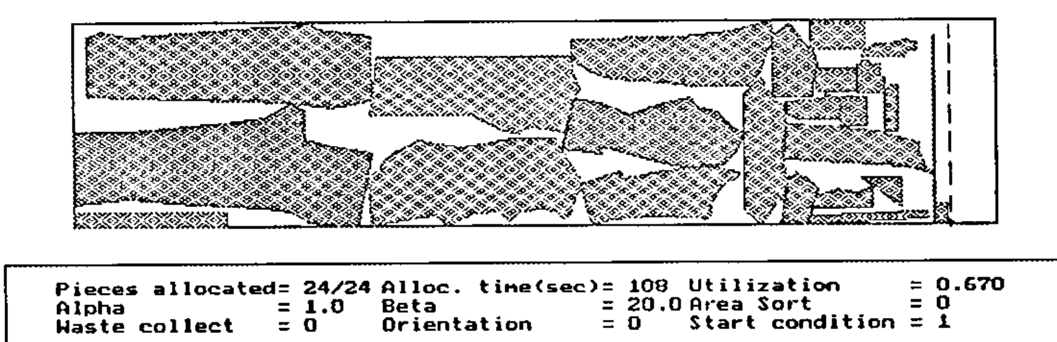


Figure 3.31 Obtained layout for textile example 2.a, α=1, β=20.
Utilization 67.0%

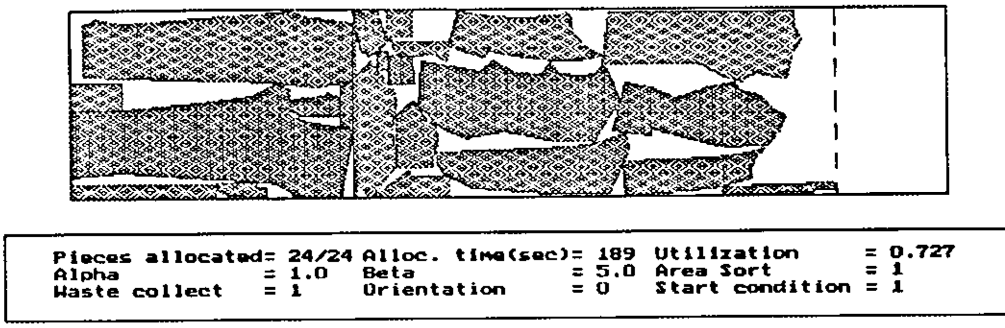


Figure 3.32 Obtained layout for textile example 2.a, α=1, β=5.
With waste filling. Utilization 72.7%

It is worth mentioning that the reported allocation times for all of the examples solved by the proposed approach, include the time for graphical display of all intermediate trials for positioning and orienting pieces during the allocation processes, which contributes significantly to these times. Therefore , a direct time comparison is not appropriate with commercial automatic algorithms. Nevertheless, all allocation times are much less than those of manual layouts.

Figures 3.33 and Figure 3.35 show the obtained layouts for the second example (n=22).

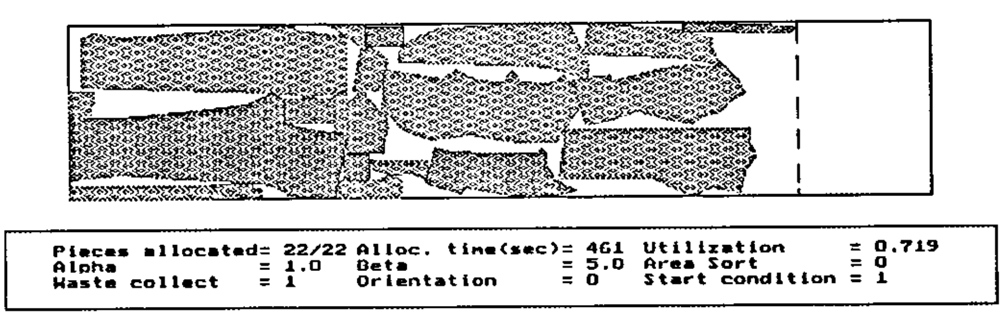


Figure 3.33 Obtained layout for textile example 2.b, α=1, β=5.
With waste filling. Orientations of horizontal flip permitted.

Utilization 71.9%.

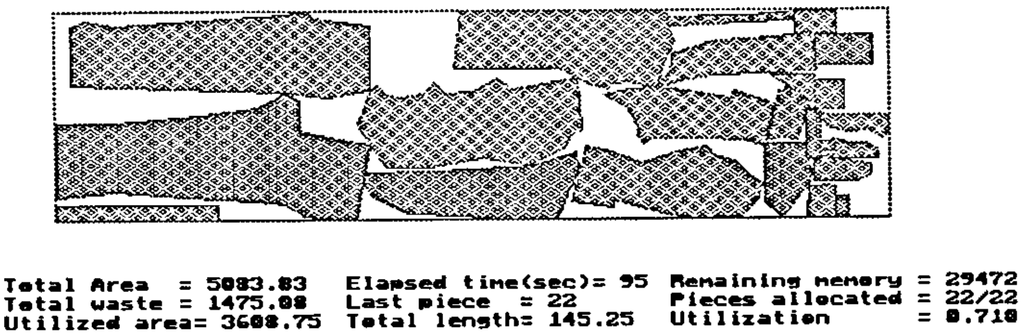


Figure 3.34 Obtained layout for textile example 2.b, α=1, β=20.

Without waste filling. Orientations of θ=0° or θ=180° permitted.

Utilization 71.0%.

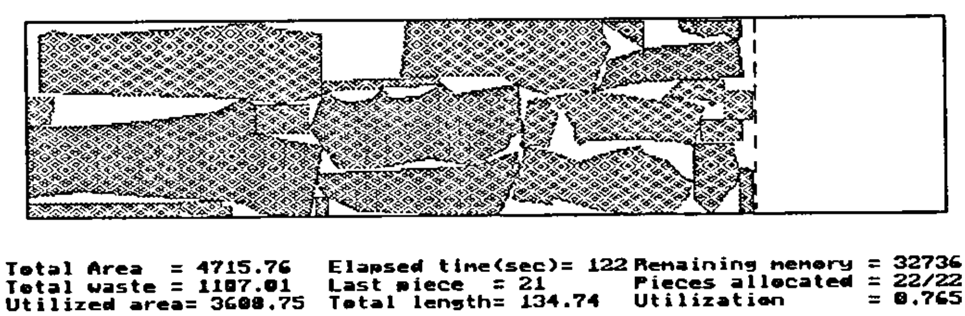


Figure 3.35 Obtained layout for textile example 2.b, α=1, β=20.
With waste filling. Orientations of θ=0° or θ=180° permitted.

Utilization 76.5%.

In all previous examples the waste filling process was not implemented in a recursive manner (i.e. after filling a waste region by a piece, that region is updated, and worked upon as if it is a new smaller waste region to be filled again, this recursive process would continue until no remaining unallocated shapes could fit within it), but rather a single piece filling was attempted. This is due to , unfortunately, incomplete module for such recursive approach. Which has the promises of improving the utilization at higher values of β, for BOM's with significant differences in shapes areas (large waste regions).

3.10 Conclusions

The problem of allocating two dimensional irregular shapes on a rectangular raw material was reviewed, the different approaches suggested throughout literature and the practiced approaches in industry were presented.

A solution approach for the problem based on state-space representation and search was suggested. The proposed search approach was just an attempt to get a fast but hopefully "good" solution. A more elaborate search approach, which would test a number of paths in the state-space tree (as opposed to the "Hill Climbing" approach used) is believed to give a better solution on the expense of time and storage requirements. Even though the obtained layouts do not beat (in most cases) the manually obtained solutions by human operators from a utilization point of view, it is effective from a time perspective. Also its effectiveness is expected to be more appropriate for large sized problems. In addition it approaches the quality of the already available automatic approaches against which it was compared, as the produced layouts illustrate.

Different evaluation functions could be suggested to govern the selection of a piece to be allocated at any solution stage. A number of these functions were presented and their effects on the solution was illustrated on a hypothetical BOM.

An initial test of the approach was presented on typical real life allocation examples obtained from local industries. The results are considered to be an initial assessment of the solution approach, and are aimed at manifesting the potentials of the approach rather than fully evaluating its performance capabilities. More tests and evaluations of the different factors affecting the solution quality are needed to be performed on this approach.

Chapter Four

Flame Cutting Machine Design

4.1 Introduction

Part of this research is to build a microcomputer based flame cutting machine and apply modern techniques and algorithms for the control of the machine, and for minimizing the waste produced by the cutting process.

An important point of view to be mentioned here is that the machine will be completely designed and built locally. That means gaining important experience in machine design and building, the availability of spare parts when needed, as well as the possibility of making changes to the machine in the future to suit any coming needs of the end users or any other potential users that might show interest in this machine. The design and build-up of the machine will be conducted in cooperation with two other colleagues

The general specifications and requirements of this machine are:

- Material to be cut: Plain carbon steel.
- Stock type: Standard 2D-sheets and plates ($1x2m^2$, $1.25x2.5m^2$ and $1.5x3m^2$).
- Shapes of products: Mainly 2D basic regular shapes and some irregular shapes.
- Sizes of shapes to be produced: For circles and arcs Φ=20-300mm, for rectangular and other shapes $50x60mm^2$ up to $2m^2$.

- Accuracy of cut : ±0.5mm.

In the phase of the design of the flame cutting machine the modularity concept was the guiding line for constructing the different subsystems of the machine. The machine is divided into the following systems:

1. The mechanical frame and motion axes.
2. The control system.
3. The flame torch assembly and gas supply.

The following sections will discuss these three systems in more details, presenting their components and subsystems, as well as their functionality.

4.2 The Mechanical Structure

The machine configuration adopted is the *gantry* type, where two orthogonal actuators move the two machine axes relative to each other to position the cutting tool in a plane. This configuration resembles a bridge crane that spans a confined work area, and is typical in industry. A general overview of the designed machine is provided in Figure 4.1.

The frame of the machine has been constructed from standard structural steel sections and parts, which provide the required rigidity and support for other machine members and parts. In addition to their local availability and their relatively low costs, the manufacturing costs are reduced by using these standard parts. U-channel beams are used to carry the long axis guiding rods and other machine members, since they are ideal to support loads and provide sufficient rigidity.

I-section beams are used to carry the short axes assembly, including the carriage on which the cutting head is to be mounted. Twin I-sections are used to provide the required deflection resistance, and carry a maximum tool weight of 50 Kg. This structure would also support thermal stresses that are typical to be induced in the working environment of this machine.

The linear movement of both machine axes is provided through the use of high precision, low backlash, and low friction ball screw-nut assembly. These ball screws raise the efficiency of power transmission to 90% rather than 40% of conventional power screws through their ball-nut sub assembly as shown in Figure 4.2. These assemblies convert the rotational motion provided by the actuators into smooth linear motion. Technical specifications of the selected power screws are available in Appendix C. Rotary bearings are used to mount the ball screws at each end. Self-alighting type is used on the drive side, while a standard ball type is used on the other one.

Both moving axes are carried over and guided by high precision supporting and guiding rods (see Figure 4.3), through linear bearings. They intend to relieve the actuators from directly carrying loads, in addition to the guarantee of smooth running axes with optimum guiding accuracy. The linear bearings used to carry the moving parts of the axes over the guiding rod provide frictionless relative motion. More details about these guiding rods are provided in Appendix C.

The torch carriage is equipped with a “rack and pinion” mechanism to provide the manual height adjustment of the torch. An improvement of this assembly could be easily made in the future by providing a motorized height adjustment, and swiveling mechanism to provide the capability of rotating the torch so that beveled cuts could also be made by the machine.

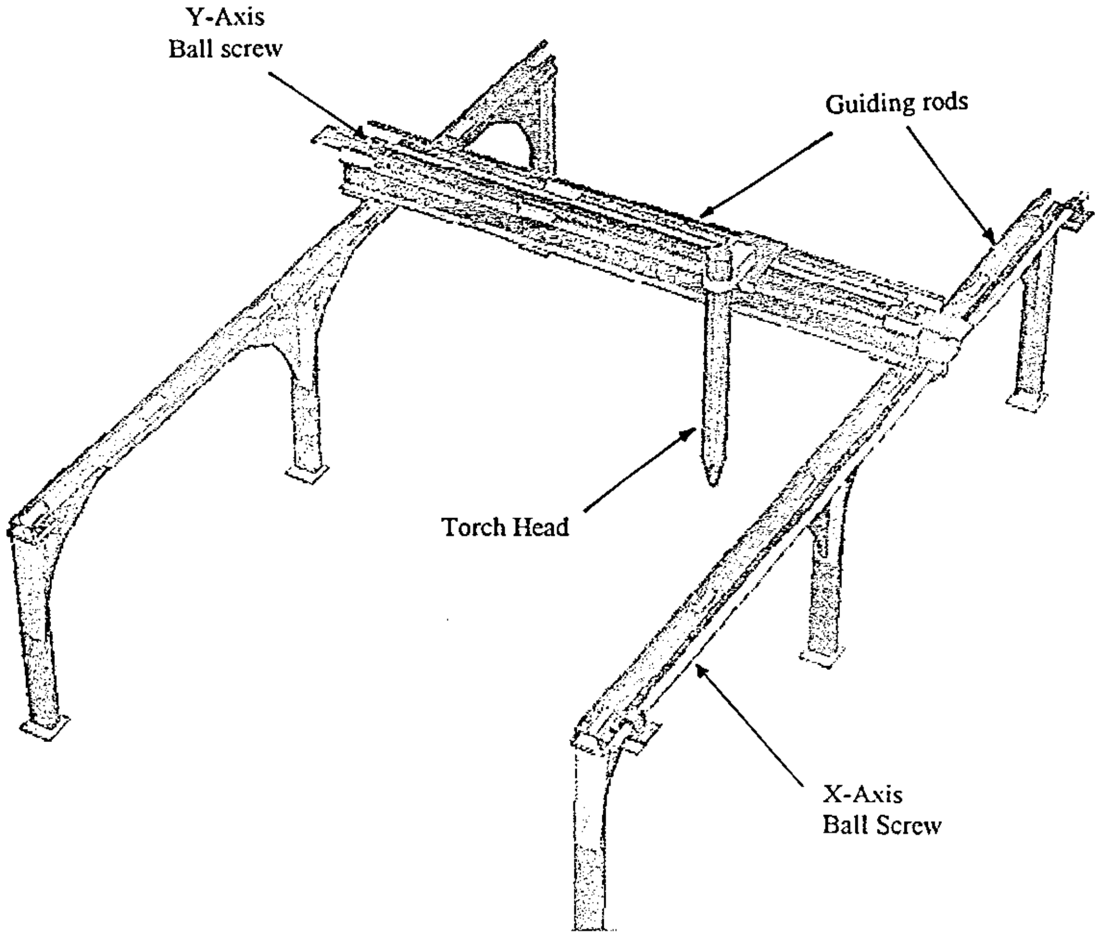


Figure 4.1a General View of the Flame Cutting Machine

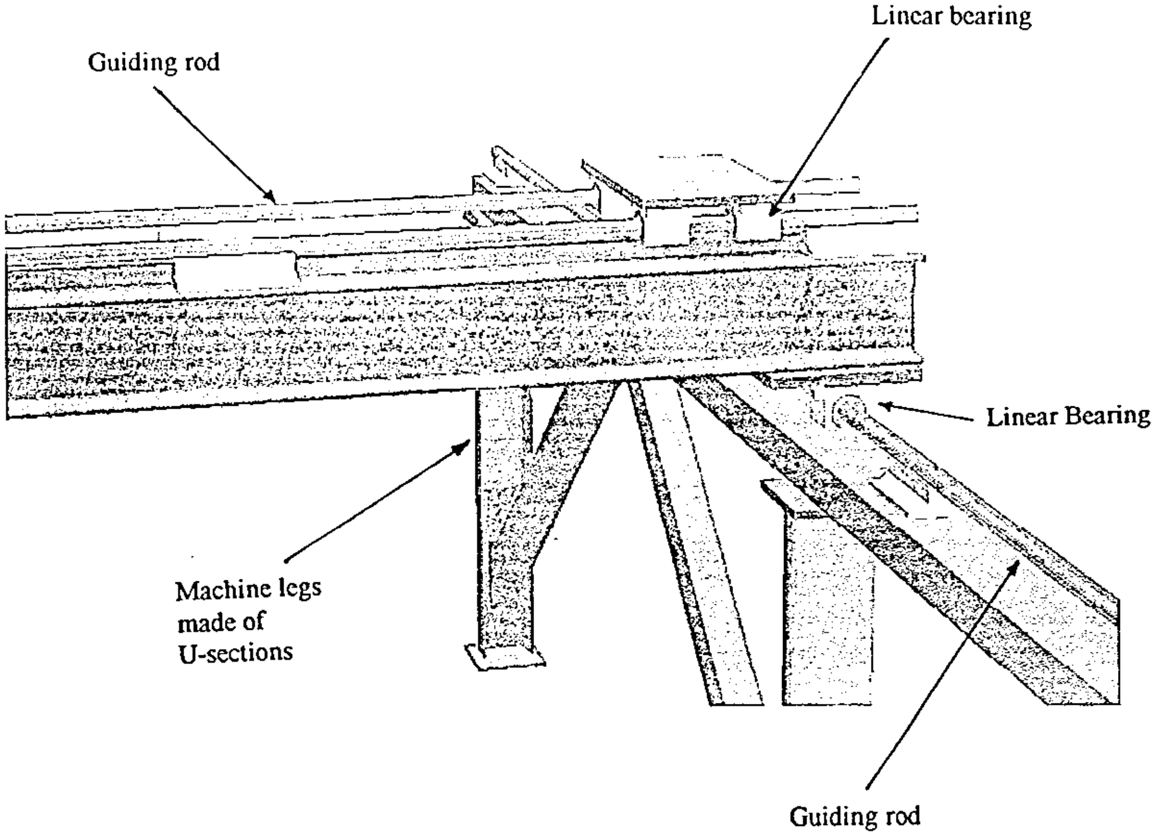


Figure 4.1b Guiding rods and linear bearings of the machine.

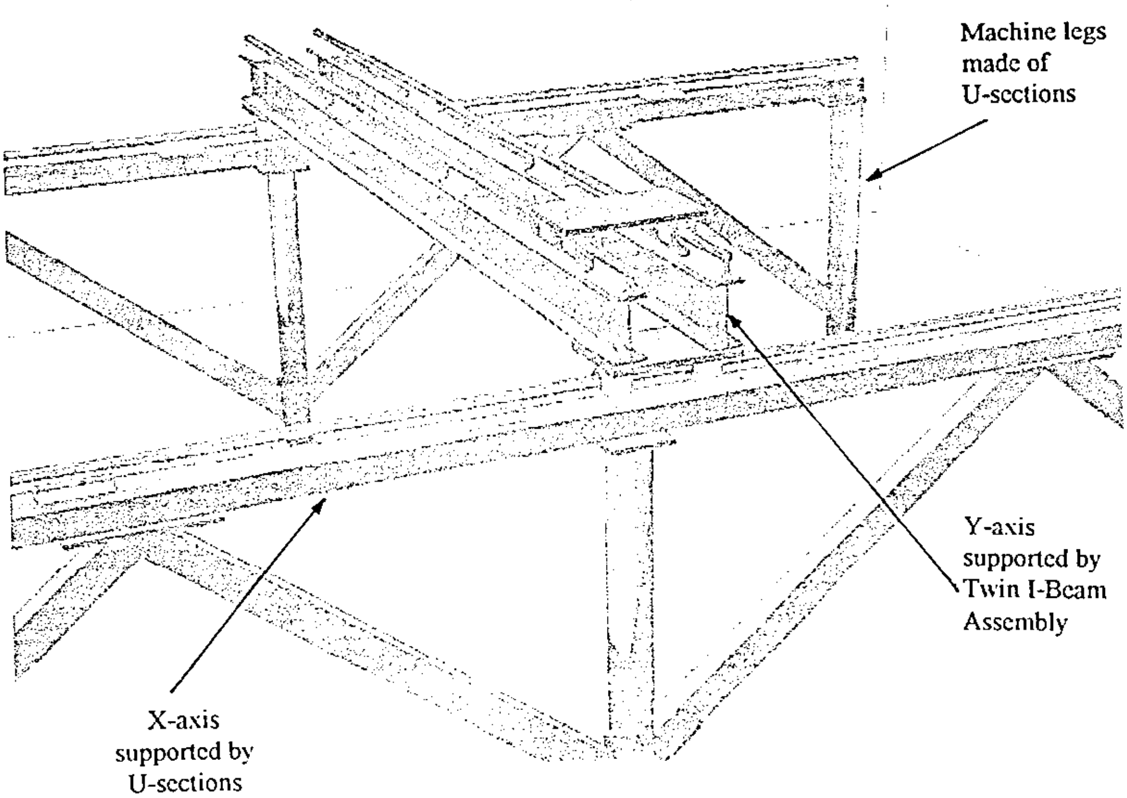


Figure 4.1c Structural Members of the Machine

Figure 4.1 Mechanical System of the Flame Cutting Machine

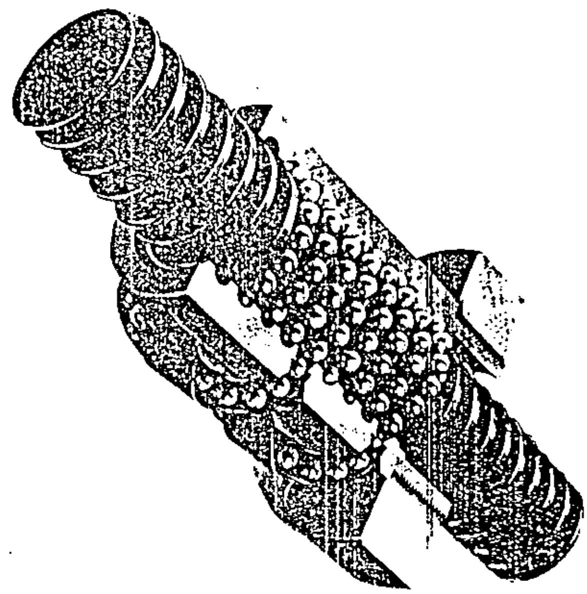


Figure 4.2 Ball Screw- Nut Assembly

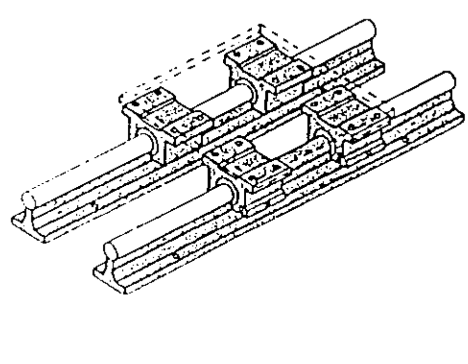


Figure 4.3 The guiding and supporting rods.

4.3 The Control System

The control system for the above described flame cutting machine has two main subsystems. The first one is the hardware and the second is the software. Following is a brief description of these two subsystems, their components and their functions.

4.3.1 Hardware Components :

4.3.1.1 The Personal Computer

The PC that is going to be used is a 80486 based machine with 33 MHz clock speed and 4 mega byte of Random Access Memory (RAM). It is equipped with a Fixed Disk Drive (Hard Disk) and a Floppy Disk Drive. The PC will run the software that will be described later on in this chapter.

4.3.1.2 The Interface Card

This is the card that will handle all the data I/O between the PC and the machine. Full specifications of the card are available in Appendix C. The main features of interest in this card are :

1. Two 12-Bit Digital-To-Analog Converters (DAC) which will convert the digital control signal issued by the software to an analog signal in the range ± 5 VDC. Simple calculations can show that the resolution of these DACs is sufficient for this application. The full speed range of the machine is 75 to 3000 mm/min or 1.95 to 50 mm/sec. If the

variation of the speed can be assumed to have 100 steps to cover the whole speed range, then a resolution of 8 bit would be sufficient.

2. Two 24-Bit Up-Down Counters that will receive the signals from the feedback devices of the flame cutting machine (Optical Encoders). The x-axis span is 3000 mm, which results in 6000 Basic Length Units (BLU), taking the required accuracy of ±0.5 mm into account, i.e. a 16-bit counter should satisfy the needs of the feedback system.
3. Thirty two Digital I/O Lines. Some of these lines will be used as inputs, where some others will be used as outputs. In Table 4.1, Section 4.3.1.5, more details about these interlocks are provided.

4.3.1.3 Power Amplifier Unit

The input of this unit will be a PC signal (±5VDC, 4 to 20 mA). This signal will be transmitted to the output using the above described interface card. The output will be the drive signal for the motors (±50VDC, 4A). This unit was designed and manufactured by a local firm.

4.3.1.4 Optical Encoders

The encoders will provide feedback signals for the PC. Each encoder supplies two pulse train signals, between which there is a 90° phase shift. This will enable the controller to detect both position as well as direction of movement of the carriage. The following figure illustrates this connection and the phase shift between the two signals. The counter counts the 0-to-1 transitions on the clock input, where the state of the gate input selects between up and down counting. If the shift is +90° then signal B is always High when the 0-to-1 transition takes place on the clock input. If the phase is -90° ,i.e., when the direction of movement is reversed then signal B is Low when the transition takes place.

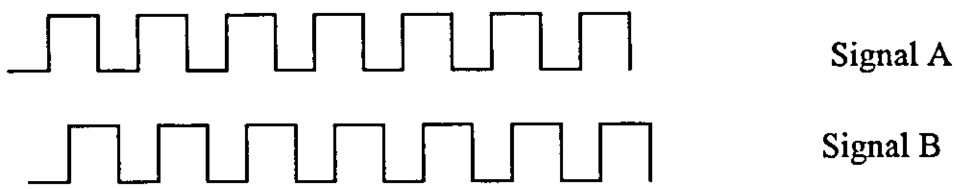


Figure 4.4 The 90° Phase Shift between the Encoder Signals

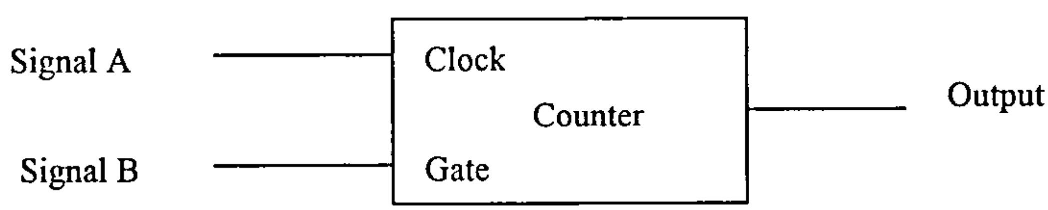


Figure 4.5 Connection of the Encoder Signals to the Counter

4.3.1.5 Digital Interlocks

The following table shows the number of digital interlocks that are built in the machine, their type and their functions. As noticed all of them serve the protection of the machine against damages resulting from improper operation, or driving the machine beyond its operational limits.

Table 4.1 Digital Interlocks of the Flame Cutting Machine

Description of the Signal	**Type (Input/Output)**	**Number of Signals**	**Source/Destination**
Oxygen supply valve	Output	1	Solenoid valve on the oxygen supply hose.
Acetylene supply valve	Output	1	Solenoid valve on the Acetylene supply hose.
Cutting gas supply valve	Output	1	Solenoid valve on the cutting gas supply hose.
Ignition signal	Output	1	Spark plug
Ready/Operating signal	Output	1	Green light on operator desk.
Alarm signal	Output	1	Visual and audio alarm equipment
Emergency stop	Input	1	Emergency stop buttons
X-Axis end-of-travel limits	Input	2	Limit switches at the two ends of the X-Axis
Y-Axis end-of-travel limits	Input	2	Limit switches at the two ends of the Y-Axis

4.3.1.6 Actuators

DC-Servo motors were selected as actuators for this machine type. They are being widely used in similar applications in the industry. In light of the power requirements for this machine, that are detailed later on in this section, permanent magnet DC-motors were selected for this application. Their control using a PC is relatively easy and they can meet the requirements of the control system The following specifications were determined in the design phase to meet the requirements and specifications of the cutting process:

Speed : The maximum linear speed that the machine should travel at is 50 mm/s. Using a power screw with a 6 mm pitch as described in section 4.2 the maximum output speed of the motors should be 500 rpm.

Torque : According to a maximum carriage weight of 50 kgs, and taking into account other structural elements that should be moved by the motors and their inertia, the required torque to be generated by the motors was found to be 1.3 Nm for the X-Axis and 1.1 Nm for the Y-Axis.

Taking the above requirements into consideration, the nearest standard motors that could be found were of the following specifications:

For the X-Axis motor : Speed 480 rpm, torque 2.0 Nm.

For the Y-Axis motor : Speed 480 rpm, torque 1.2 Nm.

4.3.2 Software Components

The main functions that the software should perform for the operation of the machine can be seen in the following list:

1. Provide a user-friendly, graphical user interface for the operator.
2. Optimize the layout of the shapes on the sheet.
3. Generate the tool path for the machine.
4. Control the movement of the machine.

4.3.2.1 The User Interface

The main function of this module is to enable the operator to draw or select the shapes that are going to be cut, and to input the thickness of the stock sheet. To simplify the process of drawing a shape or selecting a pre-drawn shape from a library, the general purpose and commonly used drafting package AutoCAD® release 12 is going to be used as the user interface. The operator will draw the shapes (or select them) and generate a file of the Data Exchange File (DXF) format. This file will be used as the input for the layout optimization module.

4.3.2.2 The Layout Optimization Module

This module is used to minimize the waste produced by the cutting process by optimizing the layout of the shapes on the stock sheet. This module consists mainly of the previously presented Automatic layout algorithms in Chapters 2 and 3. The input of this module is, as mentioned above, is the DXF file of the user drawn shapes generated by AutoCAD®. The output of this module will be also a DXF file containing the same shapes after relocating them to minimize the scrap percentage. The user has an option to accept this new layout as is or to make modifications on it using AutoCAD® , which implies a certain amount of recursion between user interface module and the optimization module. The layout optimization module can be called within the AutoCAD® environment.

4.3.2.3 The Tool Path Generator

This module will have the entities of the DXF file generated after points 1 and 2 above, and the thickness of the plate to be cut as inputs. The thickness will serve to decide on the minimum and maximum speed of cut to be applied to cut sheets of this thickness using a look-up table. Using the speed constraints and the entities of the DXF file this module will generate a set of point coordinates suitable for the controller module. It will also have an indication for the control module whether the flame should be on or off during a travel distance, i.e., will distinguish between a cutting line or a rapid traverse line between two shapes. This module is to be presented by another Master thesis research by another collegue.

4.3.2.4 The Control Module

This module is responsible for the accurate tracking of the shapes to be cut by an accurate control of the position and speed of the cutting torch. The control algorithm applied here is a Self-Tuning Generalized Predictive Controller (Self-Tuning GPC). This module will also monitor the state of the digital interlocks in order to take certain actions when necessary, such as generating alarms or stopping the machine in case of emergency. The algorithms used in this module are presented in another Master thesis research[40].

4.4 The Flame Torch Assembly and Gas Supply

The subsystems described in sections 4.2 and 4.3 constitute an X-Y-motion system. From modularity point of view, different cutting heads representing different cutting technologies can be installed on that motion system. Examples for that are water jet cutting machines or laser cutting machines. The Oxy-Acetylene cutting technology, suitable for cutting plane carbon steel sheets and plates, is used in this project. Therefore, a suitable head assembly has been selected. This head assembly consists of a straight-headed torch with the tip in line with the torch axis. A kit of different cutting tips (nozzles) is provided to suit a variety of sheet metal thicknesses. More details on this head assembly can be found in Appendix C.

The gases needed for this process, namely cutting oxygen, heating oxygen and acetylene, are supplied from three gas cylinders through a set of regulators and their flow is controlled using solenoid valves, which are controlled digitally from the controller module. Safety valves are also provided on the gas supply lines to prevent back fire.

The flame cutting machine, whose design was presented in this chapter, is currently in the phase of assembly. Some of the standard parts described above had to be imported from abroad, and therefore, considerable delays had to be taken into account.

Chapter 5

Conclusions and Recommendations

Since the thesis has been organized on self contained chapters, conclusions are presented at the ends of each chapter. However, A general conclusion on the work is presented in this chapter.

In this thesis two cases of the two dimensional cutting stock problem have been tackled. The first one is the rectangular allocation of a BOM on a number of standard sized stock sheets of finite dimensions. For this case, a heuristic approach based on capturing and programming human intuitive thoughts in laying out pieces, was suggested to fill the gaps generated after the application of an underlying algorithm. A number of such intuitive thoughts have been tested in this work. The results obtained show the effectiveness of the approach; from utilization point of view, and from computational time requirements. Such approach is believed to be adequate for solving problems of medium to large scaled size, which are typical in many industries, that need "good" and fast solutions usually, and could be obtained by using microcomputer systems. More heuristic rules could be added to enhance the performance of the solution approach suggested. An expert system solution approach could also be suggested for solving such type of problems. The second problem investigated in this research was the irregular shapes allocation on an infinite length raw material. It was noted that this problem is much more difficult to handle. Since in addition to the difficulty raised by the very large number of possible feasible layouts that may be produced for a given problem instance, the irregularity of the shapes makes it much more difficult to program and control the inference rules, such that legal transitions in the solution space could be performed. A simplified state space representation was suggested to represent the problem solution in this research. Initial results showed its promises; however, more elaborate search approach is needed to enhance its capabilities. Also a recursive filling of collected waste regions could prove to be effective in reducing the amount of waste generated. Heuristic search approaches relies greatly on the evaluation function used during the search process, therefore, more investigation concerning the selection of adequate evaluation function is suggested for future work.

A practical constraint that may increase the difficulty of the problem, which could also be investigated is the ability to handle defective parts of the raw material; where shapes to be allocated should not overlap with such regions.

Human operators are until now believed to produce better results in the layout problems than automatic approaches, especially in the irregular case. This is mainly due to the flexibility in handling, in a fast manner, similarities and mismatches among the different parts of the shapes and the space in which these shapes are to be placed. A human operator could easily recognize such similarities, in addition to the capability to evaluate, a relatively, large number of shapes placement combinations, that have good potential for waste reduction.

This may suggest that a pattern recognition approach for problem solution could promise better results.

The design of the intended prototype flame cutting machine has been a great experience, where modularity and system integration were the leading concepts in its design. Using standardized parts was also considered, were manufacturing costs could be reduced, and future maintenance requirements could be easily satisfied. Even though the machine is intended to serve the flame cutting process, it could be easily adapted for other two dimensional cutting technologies, since its basically a two dimensional positioning system.

References:

1. Smith, D.:" Welding: Skills and Technology", McGRAW-Hill Book Co., Singapore, 1989, pp. 99-136.
2. Dagli, C., and Tatoglu, M.:"An approach to two-dimensional cutting stock problems", International Journal Of Production Research, Vol. 25, No. 2, 1987, pp. 175-190.
3. Poshyanonda, P., Bahrami, A., and Dagli, C. H.:"Artificial neural networks in stock cutting problem", Neural Networks in Manufacturing and Robotics , PED, Vol. 57, Publ. By ASME, New York, USA, 1992, pp143-153.
4. Gilmor, P., and Gomory, R., "Multistage cutting stock problems of two and more dimensions", Operation Research, Vol. 13, 1965, pp 94-120.
5. Golden, B.,"Approaches to the cutting stock problem", AIIE Trans., Vol.8, No.2, 1976, pp 265-274.
6. Gary, M., and Johnson, D.: "Computers and intractability: A guide to the theory of NP-Completeness", San Francisco: Freeman, 1979.
7. Papadimitriou, C., and Steiglitz, K.:"Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, Englewood Cliffs, New Jersey, 1982, pp. 342-343.
8. Israni, S., and Sanders, J.:"Two-dimensional Cutting Stock problem research: A Review and a New rectangular Layout Algorithm", Journal of Manufacturing Systems, Vol.1, No. 2, 1982, pp. 169-182.

9. Qu, W., and Sandres, J.: "A nesting algorithm for irregular parts and factors affecting trim losses", International Journal Of Production Research, 1987, Vol. 25, No. 3, 1987, pp. 381-397.

10. Schalkoff, R.: "Artificial Intelligence: An engineering approach", McGraw-Hill Pub. Co., Singapore, 1990, pp. 493-512.

11. Christofides, N. and Whitlock, C. :"An algorithm for two-dimensional cutting problems," Operation Research, Vol. 14, 1977, pp 30-40.

12. Coverdale, I., and Wharton, F. :"An improved heuristic procedure for a nonlinear cutting stock problem", Management Science, Vol. 23, No. 1, 1976, pp. 78-86.

13. Chen, C.:"A mixed-integer programming model for a class of the assortment problems", European Journal of Operational Research, Vol. 63, 1993, pp. 362-367.

14. Haessler, R.W., and Sweeney P.E.,"Cutting Stock Problems and Solution Procedures", European Journal of Operational Research, 54, 1991, pp. 141-150.

15. Israni, S., and Sandres, J.: "Performance testing of rectangular parts-nesting heuristics", International Journal Of Production Research, 1985, Vol. 23, pp. 437-456.

16. Beasely, J. E. : "An Exact Two-Dimensional Non-Guillotine Cutting tree Search Procedure", Operations Research, Vol. 19, part 21, 1982, pp. 49-64.

17. Dagli, C. H.,and Hajakbari A.,"Two Dimensional allocation of irregular patterns: a simulated annealing approach", Robotics and manufacturing: Recent Trends in Research, Education and Applications, 3, ASME press, NewYork, 1990a, pp. 975-980.

18.Ghandforoush, P. and Daniels, J.:"Heuristic algorithm for the guillotine constrained cutting stock problem", ORSA Journal on Computing, Vol. 4, No. 3, Sept. 1992, pp 351-356.

19.MacLeod, B., Moll, R., Girkar, M., and Hanifi, N.:"Algorithm for the 2D guillotine cutting stock problem", European Journal of Operational Research, v. 68, n. 3, Aug. 1993, pp 400-412.

20.Wang, P., "Two algorithms for constrained two-dimensional cutting stock problem", Operation Research, Vol. 31, 1983, pp 573-586.

21.Morabito, R. N., Arenales, M. N. and Arcaro, V. F.:"And-or-graph approach for two-dimensional cutting problems", European Journal of Operational Research, v. 58, n. 2, Apr. 1992, pp 263-271.

22.Albano, A., and Orsini, R.," A heuristic solution of the rectangular cutting stock problem", The Computer Journal, Vol. 23, No.4, 1979, pp. 338-343.

23.Bengtsson, B.," Packing rectangular pieces-A heuristic approach", The Computer Journal, Vol. 25, 1982, pp. 353-357.

24.Chazelle, B., "The bottom-left Bin-Packing heuristic: An efficient implementation", IEEE Trans. on Computers, Vol. C-32, No. 8, 1983, pp 697-707.

25.Qu, W., and Sandres, J., "A Nesting algorithm for irregular parts & factors affecting trim losses", International Journal Of Production Research, 1987, Vol. 25, pp. 381-397.

26.Kopardekar, P., Mital, A., Allada, V. and Anand, S.:"Human strategies in cutting stock problem", Proceedings of the Industrial Engineering Research Conference 1993, Publ. by IIE, Norcross, GA, USA, 1993, pp183-186.

27. Qu, W., and Sandres, J., "Sequence selection of stock sheets in two-dimensional layout problems", International Journal Of Production Research, 1989, Vol. 27, pp. 1553-1571.

28.De Cani, P.:"A note on the two-dimensional rectangular cutting-stock problem", Journal of Operational Research Society, Vol. 29, No. 7, 1978, pp. 703-706.

29.Horwitz, E. and Sahni, S.:"Fundementals of Computer Algorithms", Pitman Pub., London, 1978, pp. 152-197.

30. Adamowicz, M., and Albano, A.:"A solution of the rectangular cutting-sotck problem", IEEE Trans. on SYS.MAN.CYB., Vol. SMC-6, No. 4, 1976, pp. 302-310.

31.Adamowics, M., and Albano, A.:"Nesting two-dimensional shapes in rectangular modules", Computer Aided Design, Vol. 8, No. 1, 1976, pp. 27-33.

32.Garcia, G., Ghosh, S., and Douglas S.:"A CAD-Based Mathematical Model for the Optimization of Plane Cutting Processes Applied to a Flexible Cutting Cell", Conference on Advanced Technology in Design & Manufacturing, Vol. 1, 1988, pp. 340-352.

33.Albano, A., and Supuoppo, G.:"Optimal allocation of two-dimensional irregular shapes using heuristic search methods", IEEE Trans. on SYS, MAN, CYB, Vol. SMC-10, No. 5, 1980, pp. 242-248.

34.Haims, M., and Freeman, H.:"A multistage solution of the template-layout problem", IEEE Trans. on SYS,SCI,CYB,Vol. SSC-6, No. 2, 1970, pp. 145-151.

35.Adamowicz, M., and Albano, A.:"A two-stage solution of the cutting-stock problem", Information Processing 71, (Proc. IFIP Congress 71), Amsterdam: North-Holland, 1972, pp. 1086-1091.

36.Lutfiyya, H., McMillin, B., Poshynonda, P., and Dagli, C.:"Composite stock cutting through simulated annealing", Mathematical And Computer Modelling, Vol. 16, No. 1, 1992, pp. 57-74.

37.Luger, G., and Stubblefield:"Artificial intelligence and the design of expert systems", Benjamine/Cummings Publ. Co., Redwood City, California, 1989, pp.77-186.

38. Dean,T., Allen, J., and Aloimonoos, Y.:"Artificial Intelligence, Theory and Practice", Benjamin/Cummings Pub. Co.., Redwood City, California, 1995, pp.131-289.

39. Gerber Garment Technology Inc.(GGT®):"Marker Maker", Tolland, Connecticut, 1996.

40. Bajjali, F.:" Self-tuning control of a flame cutting machine ", Internal report, University of Jordan, 1996.

Appendix A:
Generation of sample BOM's from a Uniform distribution

A computer program written in Microsoft Quick Basic® V4.5 was developed to generate sample BOM's with desired overall characteristics.

The program uses an algorithm that is adapted from [11]. In generating the BOM's the following information was required for each set of data:

-stock sheet dimensions,

-maximum area ratio that could be assumed by a piece in the BOM(the ratio between the maximum piece area in the BOM and the area of the stock sheet used),

-maximum aspect ratio (the ratio between the length and width of a piece),

-the maximum number of different shapes to be generated (if that number could be generated; since this is dependent on the previous ratios mentioned).

-and the maximum permissible demand of any shape type.

The out put of the random generator is a BOM that includes:

-the stock sheet dimensions(as entered to the generator).

-a list of different shapes specified as: shape number, length, width, and demand.

The area ratio and the aspect ratio were selected as the major attributes for classifying the BOM's so that they could be independent of the specific dimensions of the shapes or the raw material (stock sheet) used Also to facilitate comparisons with already published results in literature. The concentration was on the area ratio throughout the tests, where the aspect ratio was fixed at a moderate value of 3.

Appendix B:

Generation of sample BOM's from a Beta distribution

A computer program written in Lahy FORTRAN77 ®V4.1 was developed to generate sample BOM's with desired overall characteristics. The program uses the MATLAB® V4.0 for the graphical visualization of area and aspect ratio distributions. The generated BOM's were classified into nine categories based on area ratio and aspect ratio. The tree levels of difference in area ratio and aspect ratio used in the classification are: mostly small(S), average or medium (M), and mostly large(L) pieces within a BOM.

The Beta distribution (given in Equation A2.1) was selected for generating these classified BOM's because it assumes different shapes for the distribution, by controlling the values of two shape parameters: α, β, which are positive real numbers. The different shapes that could be obtained from this distribution are as follows:

-"Bridge"-shape when $\alpha>1$ and $\beta>1$.(see Figure A2.a)

-"U"-shape when $\alpha<1$ and $\beta<1$. (see Figure A2.b)

-"J"-shape when ($\alpha<1$ and $\beta>1$) or ($\alpha>1$ and $\beta<1$). (see Figure A2.c and Figure A2.d).

$$\underset{a,b}{f(x)} = \frac{x^{(a-1)} \cdot (1-x)^{(b-1)}}{B(a,b)} \qquad ;0 \le x \le 1 \qquad \text{(A2.1)}$$

Where x is a continuous random variable, α and β are real positive shaping prameters, B is the Beta function.

The distribution has the following attractive characteristics: it is defined over finite range, it is a continuous random variable distribution, and its shape is controllable through the two shape parameters mentioned previously.

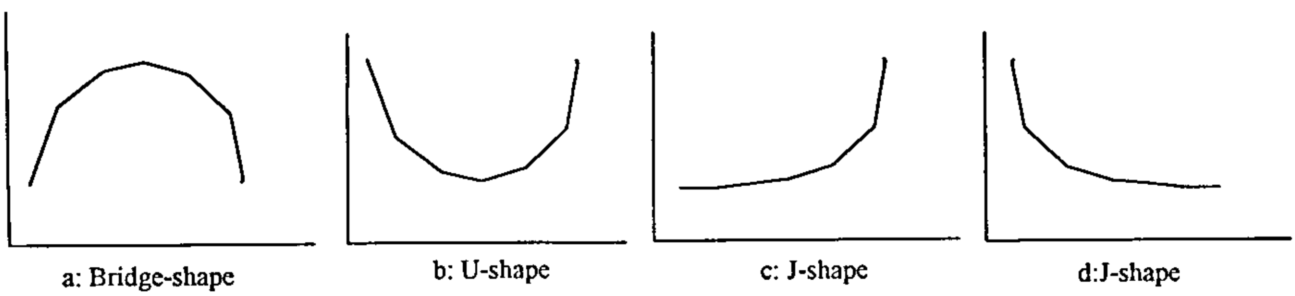


Figure A2 Different shapes that could be assumed by a Beta distribution

Appendix C:

Data Sheets of Major Machine Components

STANDARD BALL NUT ASSEMBLIES

SEE PAGES 5 & 6 FOR MATCHING SCREWS

(SEE PAGE 8 FOR LARGER SIZES)

For use with the Standard and Precision Rolled-Hardened Ball Screws shown on pages 5-6.
Nuts are hardened to Rc 56-62.
Stainless Steel Nuts are hardened to Rc 40-45.

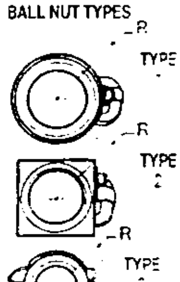
SCREW SIZE (See Pages 5-6 For Matching Screws) Nom. Dia.	Lead	STANDARD BALL NUTS: Ball Nut Part No.	Type	CAPACITY: Dynamic Load for 10^6 Inches Life (Lbs.)	Max Static Load (Lbs.)	DIMENSIONS: Outside Dia. D	Radius Over Tube R	Overall Length L	Mounting Thread Size T	Thread Length TL	BALL SIZE & QUAN.: Nominal Diameter **	Quantity (±5%)	TORQUE: To Raise 1 Pound	FLANGE: Fits Flange Part No. Page 28
3/8	.125	RS-10-2	1	130	1,300	.750	.460	1.00	664-32	.250	1/16	62	.001 Lbs. in.	R10-3
		R-11-2		260	2,600			1.875				124		R10-3
		•RS-15-2		35	230			1.00				62		•R15-3
		•R-16-2		57	460			1.875				124		•R15-3
1/2	.500	RS-20-2	3	700	3,900	1.062	.670	1.75	.837-16	.375	1/8	70	.097 Lbs. in.	R30-3
		•RS-21-2		135	725		.670	1.75						•SR30-3
5/8	.200	R-30-2	2	725	5,600	1.0 sq.	.780	1.71	.937-16	.500	1/8	67	.035 Lbs. in	R30-3
		RS-30-2												
		•SRB30-2	1	140	1,150	1.375	.80							•SR30-3
	.200 LH	RS-31-2		725	5,600									R30-3
		R-31-2	2			1.0 sq.	.780							
		•SR-31-2	1	140	1,150	1.375	.80							•SR30-3
	.200	RB30A-2	1	1,300	10,000	1.375	.80	2.75				134		R30-3
		•SRB30A-2	1	250	2,000	1.375	.80	2.75						•SR30-3
3/4	.200	R-34-2	1	1,900	18,800	1.250	.875	2.875	1.125-16	.500	1/8	165	.035 Lbs. in	R35-3
		R-35-2		950	9,400			1.875				64		
		RS-35-2		950	9,400	1.375								
		•RS-36-2		160	1,350									•R36-3
	.500	RS-37-2	3	3,150	18,500	1.312	1.00	2.937	1.250-16		5/32	150	.088 Lbs. in	R37-3
		•RS-38-2		570	3,950									•R38-3
1 (More on Page 10)	.250	R-40-2		1,500	13,000	1.5 sq.	1.150	2.347	1.563-18	.500	5/32	85	.043 Lbs. in	R40-3
		†RC40-2	2	290	2,500									•SR40-3
		R-40A-2		3,000	26,000			3.00				170		R40-3
		R-40C2-2	1			1.625	1.107		Integral Flange					–
	.250 LH	R-41-2	2	1,500	13,000	1.5 sq	1.150	2.347	1.563-18			85		R40-3
		R-41C2-2	1	3,000	26,000	1.625	1.107	2.00	Integral Flange			170		–

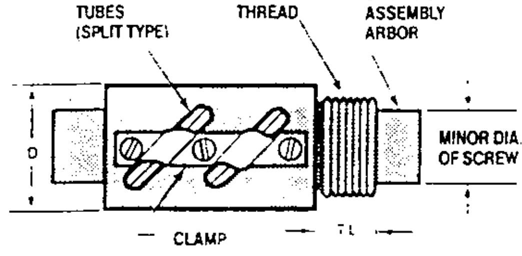
• Denotes Stainless Steel ** Note: Ball sizes may vary from nut to nut, but not in any one nut. *Never* mix balls from one nut to another. Use only factory-approved balls.
† Denotes Hard Chrome Plate with Stainless Steel Balls.

Dynamic load ratings are based on a lubricated assembly with a suitable lubricant. Normal ballnut backlash is .004 to .015", may be reduced to .002" on special order. Ball nuts are shipped on a shipping-assembly arbor unless assembly to the ball screw is requested. Specify direction of ball nut if screw ends are machined.

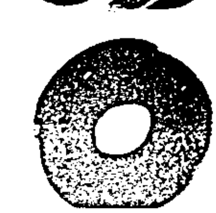
NSK NO. 1 GREASE IS RECOMMENDED FOR THESE ASSEMBLIES (STOCK ITEM).

BALL NUT TYPES





OPTIONAL NYLON BRU WIPER KITS PAGE 29



OPTIONAL MOUNTING FLANGES PAGE 28

STANDARD BALL NUT ASSEMBLIES

SEE PAGE 6 FOR MATCHING SCREWS

FOR BALL SCREWS 1" DIA. TO 4" DIA.

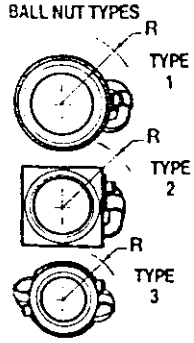
For use with the Standard and Precision Rolled-Hardened Ball Screws shown on page 6.

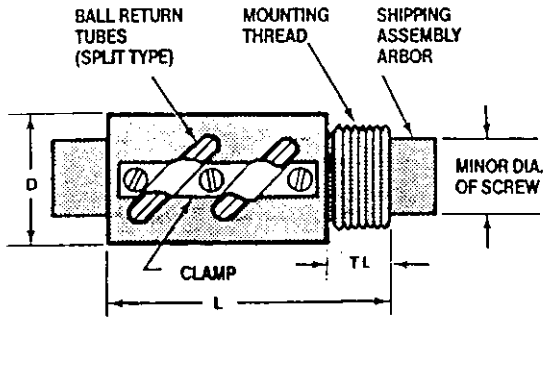
Nuts are hardened to Rc 56-62.
Stainless Steel Nuts are hardened to Rc 40-45.

SCREW SIZE		STANDARD BALL NUTS												FLANGE
See Page 6 For Matching Screws				CAPACITY		DIMENSIONS					BALL SIZE & QUAN.		TORQUE	
Nom. Dia.	Lead	Ball Nut Part No.	Type	Dynamic Load for 10^6 in. Life	Max Static Load (Lbs.)	D Nut Dia.	R Radius Over Tubes	L Overall Length	T Mounting Thread Size	TL Thread Length	Nominal Diameter ††	Quantity (±5%)	To Raise 1 Pound	Fits Flange Part No. Page 28
1	.250	R-42-2	1	3,400	30,000	1.687	1.150	3.10	1.563-18	.600	5/32	178	.043 Lbs. in.	R40-3
	.500	R-43-2	3	4,300								186	.087 Lbs. in.	
	1.00	R-44-2	2	2,050	11,150	1.5sq.		3.00				98	.175 Lbs. in.	
		RC44-2		300	2,800									•SR40-3
1⅛	.200	R-45-2	1	2,450	24,800	1.687	1.150	2.50	1.625-20	.485	⅛	244	.035 Lbs. in.	R45-3
		•R-46-2		440	4,450									•SR46-3
	.200 LH	R-47-2		2,450	24,800									R45-3
1½	.500	R-50-2	1	9,050	54,100	2.625	1.812	4.687	2.548-18	.750	5/16	102	.087 Lbs. in.	R50-3
	.250 LH	R-53-2	1	4,525	44,800	2.093	1.340	3.00	1.967-18	.500	5/32	260	.035 Lbs. in.	R54-3
	.250	R-54-2												
	1.00	R-55-2	3	6,900	32,400	2.25 sq.	1.70	3.625	2.250-20	1.00	11/32	64	.175 Lbs. in.	R55-3
	1.00 LH	R-56-2												
	1.875	R-58-2		7,240	29,800			5.00			9/32	84	.208 Lbs. in.	
2	1.00	R-61-2	1	23,000	130,000	3.250	2.30	6.375	3.00-12	1.50	⅜	160	.175 Lbs. in.	R61-3
	.500	R-62-2		18,000								150	.087 Lbs. in.	
2¼	.500	R-60-2	1	19,800	132,000	3.375	2.275	6.75	3.137-12	1.56	⅜	154	.087 Lbs. in.	R60-3
2½	.500	R-70-2	1	22,500	138,500	4.00	2.75	6.75	3.62-12	1.75	⅜	186	.087 Lbs. in.	R70-3
	1.00	R-71-2		26,500								194	.175 Lbs. in.	
	.250	R-74-2		6,300	81,000	3.375	2.01	3.75	3.34-12	.750	5/32	468	.035 Lbs. in.	R74-3
3	.660	R-80-2	1	42,400	254,000	4.750	3.125	9.312	4.325-12	2.00	½	177	.115 Lbs. in.	R80-3
4	1.00	R-90-2	1	85,700	476,950	5.875	3.756	12.593	5.497-12	2.00	⅝	190	.175 Lbs. in.	R90-3

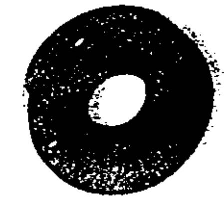
•Denotes Stainless Steel. Part No. RC44-2 is hard chrome plated. Dynamic load ratings are based on a lubricated assembly with a suitable lubricant. Normal Ball Nut Backlash is .004 to .015", may be reduced to .002 on special order. Ball Nuts are shipped on a shipping-assembly arbor unless assembly to the Ball Screw is requested. Specify direction of Ball Nut if Screw Ends are machined. ††Note: Ball sizes may vary from nut to nut but not in any one nut. Never mix balls from one nut to another. Use only factory-approved balls.

NSK NO. 1 GREASE IS RECOMMENDED FOR THESE ASSEMBLIES (STOCK ITEM).





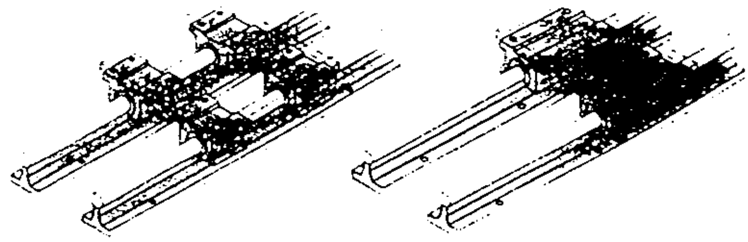
OPTIONAL NYLON BRUSH WIPER KITS PAGE 29

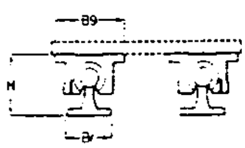


OPTIONAL MOUNTING FLANGES PAGE 28

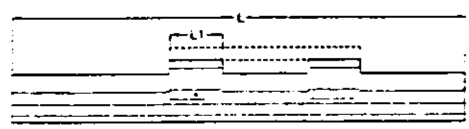
1CB

Double Shaft Fully Supported System

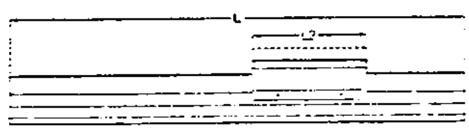




Double Shaft Fully Supported System with 4 Pillow Blocks



Double Shaft Fully Supported System with 2 Twin Pillow Blocks



Maximum Stroke Length is determined by subtracting pillow block length **(L2)** from total system length **(L)**.

Part No	Nom Shaft Dia	Load (lbf)*		Dimension (in.)				Pillow Block	Rail Assembly
		Max. On System	Max. On Any Bearing	L1	H ±0.001	B1	B9		
1CB-08-FAO	1/2	720	180	1.50	1.812	1.50	2.00	SPB-8-OPN-XS	SRA-8-XS
1CB-12-FAO	3/4	1880	470	1.88	2.437	1.75	2.75	SPB-12-OPN-XS	SRA-12-XS
1CB-16-FAO	[illegible]	3120	780	2.63	2.937	2.13	3.25	SPB-16-OPN-XS	SRA-16-XS
1CB-24-FAO	1 1/2	6240	1560	3.75	4.250	3.00	4.75	SPB-24-OPN-XS	SRA-24-XS

*Based on a travel life of 2 million inches

Part No	Nom Shaft Dia	Load (lbf)*		Dimension (in.)				Maximum Stroke Length (in.)	Pillow Block	Rail Assembly
		Max. On System	Max. On Any Bearing	L2	H ±0.001	B	B9			
1CB-08-HAO	1/2	720	180	3.50	1.812	1.50	2.00	L − (3.50)	TWN-8-OPN-XS	SRA-8-XS
1CB-12-HAO	3/4	1880	470	4.50	2.437	1.75	2.75	L − (4.50)	TWN-12-OPN-XS	SRA-12-XS
1CB-16-HAO	[illegible]	3120	780	6.00	2.937	2.13	[illegible]	L − (6.00)	TWN-16-OPN-XS	SRA-16-XS
1CB-24-HAO	1 1/2	6240	1560	9.00	4.250	3.00	4.75	L − (9.00)	TWN-24-OPN-XS	SRA-24-XS

*Based on a travel life of 2 million inches

System	8"	12"	16"	18"	20"	24"	28"	30"	32"	36"	40"	42"	44"	48"
1CB-08	■	■	■		■	■	■		■	■	■		■	■
1CB-12		■		■		■		■		■		■		■
1CB-16		■		■		■		■		■		■		■
1CB-24			■			■			■		■			■

Systems ordered in standard lengths are typically shipped in one week. Custom length systems are available and require two to three weeks for delivery. Lengths exceeding 156.00 ins. require butt joints and will need four to six weeks for delivery. For special requirements, please contact the Thomson **Systems** application engineering department

Enhanced Multifunction I/O Boards for ISA

Specifications

Typical for 25° C unless otherwise noted

Analog Input

Input Characteristics

Number of channels .... 16 single-ended or 8 differential, software selectable
Type of ADC .... Successive approximation
Resolution .... 12 bits, 1 in 4,096
Maximum sampling rate .... 100 kS/s
Input signal ranges

Board Gain (Software Selectable)	Board Range (Software Selectable)	
	±5 V	0 - 10 V
0.5	±10 V	–
1	±5 V	0 to 10 V
2	±2.5 V	0 to 5 V
5	±1 V	0 to 2 V
10	±500 mV	0 to 1 V
20	±250 mV	0 to 500 mV
50	±100 mV	0 to 200 mV
100	±50 mV	0 to 100 mV

Input coupling .... DC
Max working voltage
(signal + common mode) .... Each input should remain within ±11 V of ground
Overvoltage protection .... ±35 V powered on, ±25 V powered off
Inputs protected .... ACH<0..15>, AISENSE
FIFO buffer size .... 512 samples
Data transfers .... DMA, interrupts, programmed I/O
DMA modes .... Single transfer, demand transfer
Configuration memory size .... 512 words

Transfer Characteristics

Relative accuracy .... ±0.2 LSB typical dithered, ±1.0 LSB maximum undithered
DNL .... ±0.2 LSB typical, ±0.5 LSB maximum
No missing codes .... 12 bits, guaranteed
Offset error
Pregain error after calibration .... ±2 µV maximum
Pregain error before calibration .... ±24 mV maximum
Postgain error after calibration .... ±0.5 mV maximum
Postgain error before calibration .... ±100 mV maximum
Gain error (relative to calibration reference)
After calibration (Gain = 1) .... ±0.01% of reading maximum
Before calibration .... ±2.0% of reading maximum
Gain ≠ 1 with gain error adjusted to 0 at gain = 1 .... ±0.05% of reading maximum

Amplifier Characteristics

Input impedance
Normal powered on .... 100 GΩ in parallel with 50 pF
Powered off .... 3 kΩ minimum
Overload .... 3 kΩ minimum
Input bias current .... ±200 pA
Input offset current .... ±100 pA
CMRR (all input ranges) .... 90 dB, DC to 60 Hz

Dynamic Characteristics

Bandwidth
Small signal (-3 dB) .... 200 kHz
Large signal (1% THD) .... 300 kHz
Settling time for full-scale step .... 10 µs maximum to ±0.5 LSB accuracy
System noise (not including quantization)

Gain	System Noise
0.5 to 10	0.07 LSBrms
20	0.12 LSBrms
50	0.25 LSBrms
100	0.5 LSBrms
dither on, any gain	0.5 LSB rms

Crosstalk .... -80 dB, DC to 100 kHz

Stability

Recommended warm-up time .... 15 minutes
Offset temperature coefficient
Pregain .... ±15 µV/°C
Postgain .... ±240 µV/°C
Gain temperature coefficient .... ±20 ppm/°C
Onboard calibration reference
Level .... 5.000 V (±2.5 mV) (actual value stored in EEPROM)
Temperature coefficient .... ±5 ppm/°C maximum
Long-term stability .... ±15 ppm/ $\sqrt{1000\ h}$

Analog Output

Output Characteristics

Number of channels .... 2 voltage
Resolution .... 12 bits, 1 in 4,096
Maximum update rate .... 100 kS/s
Type of DAC .... Double buffered, multiplying
FIFO buffer size .... None
Data transfers .... DMA, interrupts, programmed I/O
DMA modes .... Single transfer

Transfer Characteristics

Relative accuracy (INL)
After calibration .... ±0.3 LSB typical, ±0.5 LSB maximum
Before calibration .... ±4 LSB maximum
DNL
After calibration .... ±0.3 LSB typical, ±1.0 LSB maximum
Before calibration .... ±3 LSB maximum
Monotonicity .... 12 bits, guaranteed after calibration
Offset error
After calibration .... ±1.0 mV maximum
Before calibration .... ±200 mV maximum
Gain error (relative to internal reference)
After calibration .... ±0.01% of output maximum
Before calibration .... ±0.5% of output maximum
Gain error
(relative to external reference) .... 0% to +0.5% of output maximum, not adjustable

Voltage Output

Ranges .... ±10 V, 0 to 10 V, ±EXTREF, 0 to EXTREF (software selectable)
Output coupling .... DC
Output impedance .... 0.1 Ω maximum
Current drive .... ±5 mA maximum
Protection .... Short-circuit to ground
Power-on state .... 0 V
External reference input
Range .... ±11 V
Overvoltage protection .... ±35 V powered on, ±25 V powered off
Input impedance .... 10 kΩ
Bandwidth (-3 dB) .... 300 kHz

Dynamic Characteristics

Settling time for full-scale step .... 10 µs to ±0.5 LSB accuracy
Slew rate .... 15 V/µs
Noise .... 200 µVrms, DC to 1 MHz
Glitch energy (at midscale transition)
Magnitude .... ±100 mV
Duration .... 3 µs

Stability

Offset temperature coefficient .... ±50 µV/°C
Gain temperature coefficient
Internal reference .... ±25 ppm/°C
External reference .... ±25 ppm/°C
Onboard calibration reference
Level .... 5.000 V (±2.5 mV) (actual value stored in EEPROM)
Temperature coefficient .... ±5 ppm/°C maximum
Long-term stability .... ±15 ppm/ $\sqrt{1000\ h}$

Enhanced Multifunction I/O Boards ISA

Digital I/O

Number of channels
- AT-MIO-16E-10 .... 8 input/output
- AT-MIO-16DE-10 .... 32 input/output

Compatibility .... TTL/CMOS

Digital logic levels

DIO<0..7>

Level	Minimum	Maximum
Input low voltage	0 V	0.8 V
Input high voltage	2 V	5 V
Input low current (V_{in} = 0 V)	–	-320 μV
Input high current (V_{in} = 5 V)	–	10 μA
Output low voltage (I_{OL}=24 mA)	–	0.4 V
Output high voltage (I_{OH}=13 mA)	4.35 V	–

PA<0..7>, PB<0..7>, PC<0..7> (Remaining 24 lines of AT-MIO-16DE-10)

Level	Minimum	Maximum
Input low voltage	0 V	0.8 V
Input high voltage	2 V	5 V
Input low current (V_{in} = 0 V)	–	-60 μA
Input high current (V_{in} = 5 V)	–	10 μA
Output low voltage (I_{OL}= 2.5 mA)	–	0.4 V
Output high voltage (I_{OH}= 2.5 mA)	3.9 V	–

Handshaking
- (AT-MIO-16DE-10 only) 3-wire

Power-on state .... Tri-state

Data transfers
- AT-MIO-16E-10 .... Programmed I/O
- AT-MIO-16DE-10 .... Interrupts, programmed I/O

Timing I/O

Number of channels .... 2 up/down counter/timers, 1 frequency scaler

Resolution
- Counter/timers .... 24 bits
- Frequency scalers .... 4 bits

Compatibility .... TTL/CMOS

Base clocks available
- Counter/timers .... 20 MHz, 100 kHz
- Frequency scaler .... 10 MHz, 100 kHz

Base clock accuracy .... ±0.01%

Maximum source frequency .... 20 MHz

Minimum source pulse duration .... 10 ns in edge-detect mode

Minimum gate pulse duration .... 10 ns in edge-detect mode

Data transfers .... DMA, interrupts, programmed I/O

DMA modes .... Single transfer

Triggers

Digital Trigger

Compatibility .... TTL

Response .... Rising or falling edge

Pulse width .... 10 ns minimum

RTSI

Trigger lines 7

Bus interface .... Slave

Power Requirement

+5 VDC (±5%) .... 0.7 A

Power available at I/O connector .... +4.65 VDC to +5.25 VDC at 1 A

Physical

Dimensions
(not including connectors) .... 33.8 by 9.9 cm (13.3 by 3.9 in.)

I/O connector
- AT-MIO-16E-10 .... 68-pin male SCSI-II type
- AT-MIO-16DE-10 .... 100-pin female 0.050 D-type

Environment

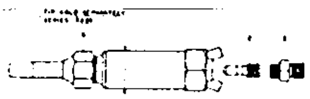
Operating temperature .... 0° to 55°C

Storage temperature .... -55° to 150°C

Relative humidity .... 5% to 90% noncondensing

AT-MIO-16E-10, AT-MIO-16DE-10

MODEL	GAS
33-2	ACETYLENE
33-2/28	
33-2F	PROPANE
33-2F/28	NAT GAS



2	C-62	Injector (ACETYLENE)
	C-62-F	Injector (PROPANE)
3	13319-A	Inlet connections (ACETYLENE)
	13319-F	Inlet connections (PROPANE)
4	13357-2	Inlet connections (GAS)
5	13357-2	Inlet connections (OX)
6	[illegible]	[illegible]

ملخـص

تطبيق تقنيات الذكاء الاصطناعي في قطع المواد الثّنائية البعد

إعداد

رامـي مصطفى علـي الناقــــه

المشرف

الدكتور يوســـف العســـــاف

الاستغلال الأمثل للمواد الخام في عمليات قطع المـواد ثنائيـة البعـد (الصفـائح و الألـواح) يعتـبر مـن أهـم الأهداف في كثير من الصناعات. حيث يجب توزيع مجموعة من الأشكال المراد قطعها بشكل مناسب يحقق أقل نسبة ممكنة من الفاقد. لقد جرت العادة على أن تتم عمليـة التوزيـع هـذه مـن قبـل عـامل متخصـص، وبشكل يدوي. وبالرغم من تطوير عدد من الطرق الآلية لحل هذه المشكلة، إلا أن العامل البشـري مـازال أقدر على إعطاء نتائج أفضل من الطرق المحو سبة.

في هذا البحث تم التطرق لحالتين من هذه المسألة، الأولى هي عملية توزيع الأشكال المنتظمة (المسـتطيلة)، والثانية هي عملية توزيع الأشكال غير المنتظمة، حيث تم اقتراح حلول آلية لهاتين المسألتين. بالنسبة للحالـة الأولى فقد تم اقتراح حل آلي مبني على اقتباس وبرمجة أفكـار الإنسـان في عمليـة توزيـع الأشـكال. النتـائج المستحصل عليها تظهر نجاح الطريقة في الحصول على توزيعات جيدة، بدون أي تدخل للعامل في التوزيع. أما بالنسبة للمسألة الثانية فقد تم اقتراح تمثيل للمسألة على شكل "فضاء الحالات"، ومن ثم تم اقـتراح حـل سريع مبني على طريقة "صعود التل". تظهر النتائج الأولية أن هذه الطريقة جيدة، ولكن يعوزها مزيـد مـن البحث و التطوير.

لقد تم البحث في هاتين المسألتين من أجل استخدامهما في آلية دعم القرار لمنظومة آلة قطع باللهب، حيـث تم تقديم تصميم لهذه الآلة أيضا في نهاية البحث.