Engineering Journal of Qatar University, Vol. 4, 1991, p. 25 - 33.

A COMPUTER AIDED DESIGN TOOL FOR AREA OPTIMIZATION USING BREADTH FIRST SEARCH TECHNIQUE

Ahmed Hamad L. Al-Mohanadi* and Ahmed Saleh**
*Assistant Professor, Electrical Engineering Department,
Qatar University, Doha, Qatar.
**Computer Engineer, Electrical Engineering Dept.,
Qatar University, Doha, Qatar.

ABSTRACT

Floorplanning is the process of partitioning an area on any surface such that the partitions guarantee minimum wasted area. The technique used to partition a given are depends on the requirements of the application. In an integrated circuit design the floorplan is done on one piece of silicon and the partitions depend on other information such as the electrical wiring which must be minimized. This requires a study of the functions (area and wiring) and the relations between all functions. The technique to solve the problem is not trivial since area and wiring functions are both considered. Another example where good floorplanning is required is any material cutting process. Glass sheets come in different sizes and the requirements for any application not necessarily require the same size. So it is important to cut the required sheets from the available ones. This process will require an expert to deal with the cutting however it is not necessary to guarantee the minimization in wastages.

This paper investigates a technique to minimize wastage areas and to give the best solution for multi-project with multi-sheets of different sizes. The algorithm is based on the breadth first which expand a tree based on the number of nods given. Two examples are given to demonstrate the results of the algorithm.

INTRODUCTION

In this paper we study the minimization of waste produced from material. Material sheets come in different sizes and shapes and the requirements for any application may not necessarily require the same area. So it is important to get the required material sheets from available ones by a cutting process which minimizes wasted materials. In other words the goal of placement is to determine the position and the orientation, horizontal or vertical of every required block from or on a given two dimensional place (given blocks) such as to minimize the overall wastages areas. The proposed algorithm is based first on the depth and expands a tree at certain level with all possible nodes and chooses from these nodes the most promising node, which we define as the one that produces the least waste area.

ALGORITHMS

The algorithm is developed based on the idea of finding a node that gives the solution. A node is a subset of the whole set that contains all given cells:

$N = \{ (NL1, NW1), (NL2, NW2), \ldots, (NLn, NWn) \}$

That satisfy all the following conditions:

for all $(NLi, NWi) \;\; R \;\; i = 1, 2, \ldots, n$

$NW1 = NW2 = \ldots = NWn = \text{Constant}$

where this constant $\leqslant Gwj$ for $i \leqslant j \leqslant NG$

and

$$\sum_{i=1}^{n} NLi = GLj \text{ for } 1 \leqslant j \leqslant NG$$

or

$NL1 = NL2 = \ldots = NLn = \text{constant}$

where this constant $\leqslant GLJ$ for $1 \leqslant j \leqslant NG$

and

$$\sum_{i=1}^{n} NWi = GW \quad \text{for } 1 \leqslant j \leqslant NG$$

Let the given pieces of glass sheets be in the set

$G = (GL1, GW1), (GL2, GW2), \ldots, (GLn, GWn)$

such that

$(GLi, GWi) \;\; G$ for all $1 < i < NG$

where Gli = all given lengths
GWi = all given widths
NG = total number of given glasses.

Also let the required pieces of glass sheets be in the set

R = (RL1, RW1), (RL2, RW2), ..., (RLm, RWm)

such that

(RLi, RWi) G for all $1 < i < NR$

Where NR = Total number of required glass sheets.

PROCEDURE OF THE ALGORITHMS

The algorithm given in (1) can be summarized by:

★ Expand the tree at certain level and select a nod at that level which have minimum waste area.

★ Go depth and expand tree with the remaining required sheets and also select one node with minimum waste.

★ If the process is blocked at certain level then go back to the previous level and expand tree at that level and take a node with minimum waste but larger than the previous waste.

★ If the Root node is reached go back which means that there is no solution.

★ Solution obtained if all required sheets are obtained.

CAD TOOLS

The CAD tool consists of a computer program written in quick Basic which takes inputs as given sheets sizes and also required sheets sizes. After getting the results the program plot the output on printer or plotter. The flow chart (Fig. 1) shows the behaviour of the algorithm based on given and required sheets. This can be summarized by the following steps:

1. Search for a node.

2. Compute waste area.

3. If waste area of current node < waste of selected one then current node is the selected node.

4. If there are more nodes at that level go back to step 1.

5. Check to see if you got a node from step 1,2,3 and 4.

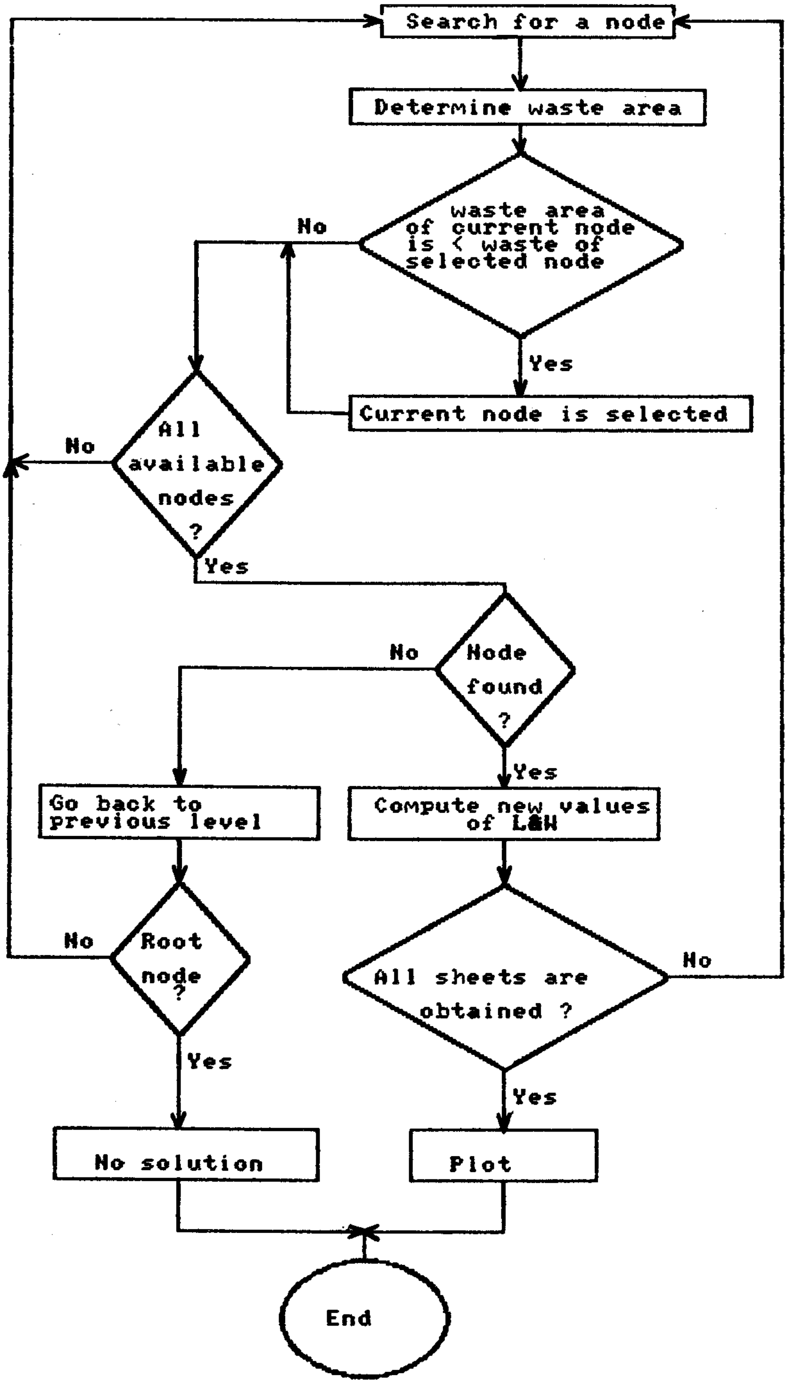
Search for a node
Determine waste area
waste area of current node is < waste of selected node
No
Yes
Current node is selected
All available nodes ?
No
Yes
Node found ?
No
Yes
Go back to previous level
Compute new values of L&W
Root node ?
No
Yes
All sheets are obtained ?
No
Yes
No solution
Plot
End

Fig. 1

6. If there is a node X then compute waste and adjust values of given sheet length and width.
7. Check to see if all required sheets are obtained. If all are obtained then plot the result and end the process or otherwise go to step 1.
8. If there is not a node found in step 5 (false) go back to the previous level.
9. Check to see if Root node is reached. If yes then there is no solution, otherwise go back to step 1.

The correctness of the tool developed is demonstrated by the following examples:

First: No waste (all given area is occupied)
Given area: 10 x 10
Required sheets:

10 x 3
6 x 3
7 x 2
6 x 2
4 x 3
3 x 3
5 x 1

1. All sheets are organized based on their areas.

Area 1 > Area 2 > Area 3 ...

2. The algorithm will try to find one edge that complete either the length or the width of the given sheet by considering the length and the width of any given sheet or sheets (Fig. 2). The program found that:
   a. block 10 x 3 satisfy one side of the given block. The remainder area of the given block is 7 x 10.
   b. Blocks 6 x 3 + 4 x 3 complete one edge which is 10 x 3 the remainder is 10 x 4.
   c. Block 7 x 12 complete one edge and the remainder is 8 x 7.
   d. Blocks 4 x 3 + 3 x 3 complete edge 7 x 3 and the remainder is 7 x 7.
3. The program found that the area of 6 x 3 + 4 x 3 = 30 is greater than 7 x 2 = 14 or 4 x 3 + 3 x 3 = 12. So the remainder is 10 x 4.
4. From the remainder area 10 x 4 it has been found to find a complete edge for

any combination of the left sheets. The program must go back to node 10 x 7 and start again by taking an area either less or equal 10 x 3.

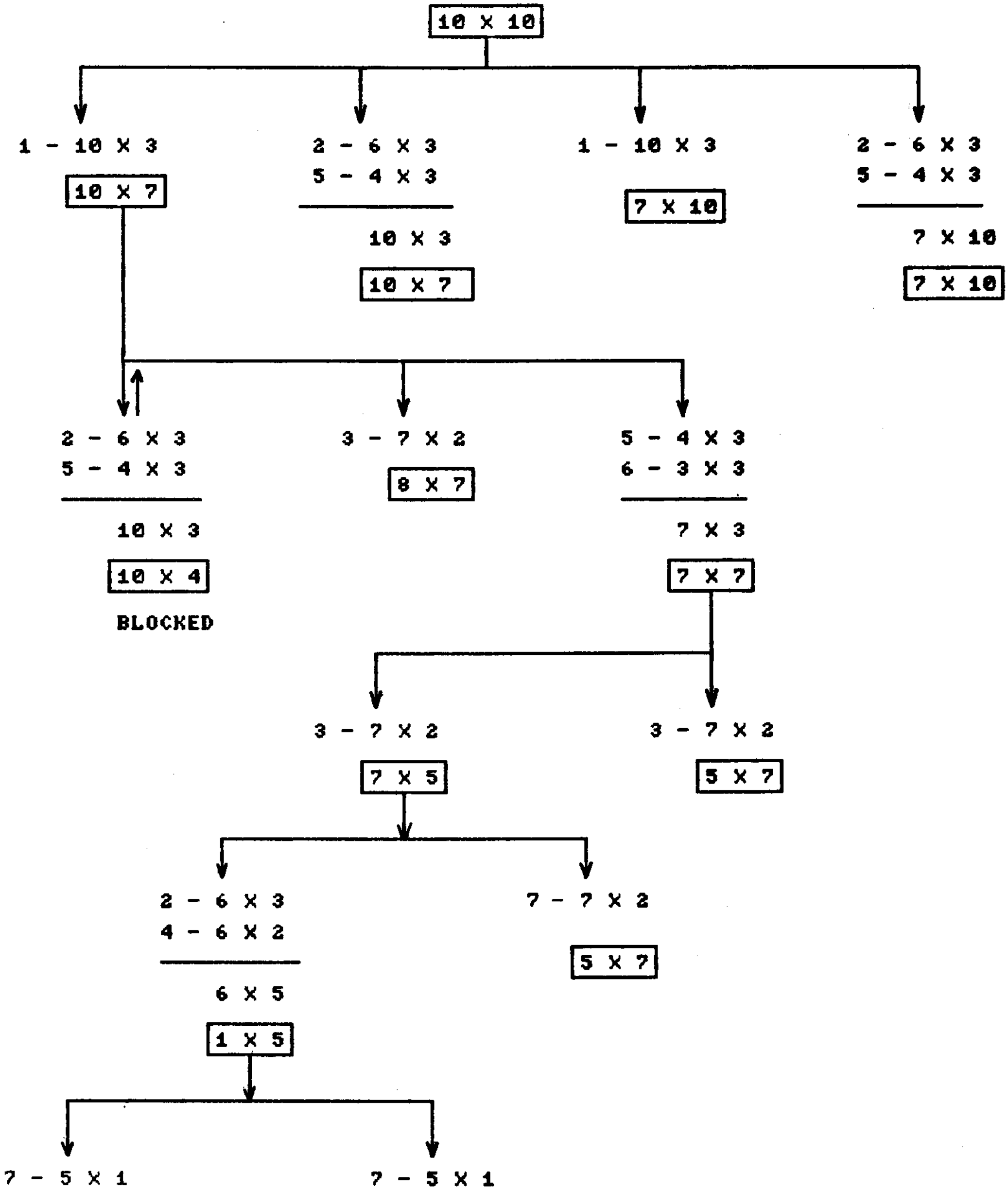

Fig. 2

5. The choice is 4 x 3 + 3 x 3 = 7 x 3 = 21 which is less than 30.

6. From step 5 the remainder is 7 x 7.

7. From the area 7 x 7 the program search for an edge such as 7 x 2 and the remainder is 7 x 5.

8. The area 7 x 5 can accept 6 x 3 + 6 x 2 = 6 x 5 which complete edge 5 and remainder is 5 x 1. The program will omit the choice of 5 x 1 since it is less than the area of 6 x 5.

9. The remainder area 5 x 1 is the size of the last sheet required to be cut from the given one.

10. The program then plot the output (Fig. 3).

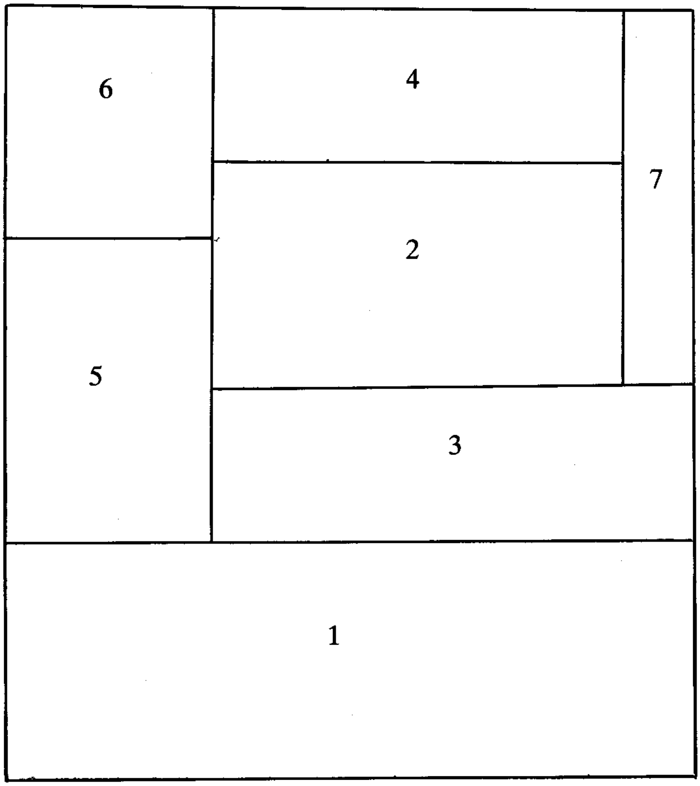

Fig. 3: Output of Example 1.

Second: With waste (some of the given area is occupied).

Given:	66 x 9
Required:	4 x 4
	4 x 2
	3 x 3
	3 x 3
	2 x 4

1. All sheets are organized based on three areas.

   Area 1 > area 2 > area 3 > .....

2. Fig. 4 shows different starts which they complete an edge. The greatest area occupied is 4 x 4 + 4 x 2 = 4 x 6 and the remainder 5 x 6.

3. Starting at A′, there are two starts either 3 x 3 = 3 x 3 = 3 x 6 or 3 x 3 + 2 x 3 = 5 x 3 as 6 x 3 is taken since it occupies greater area. The remainder is 6 x 2.

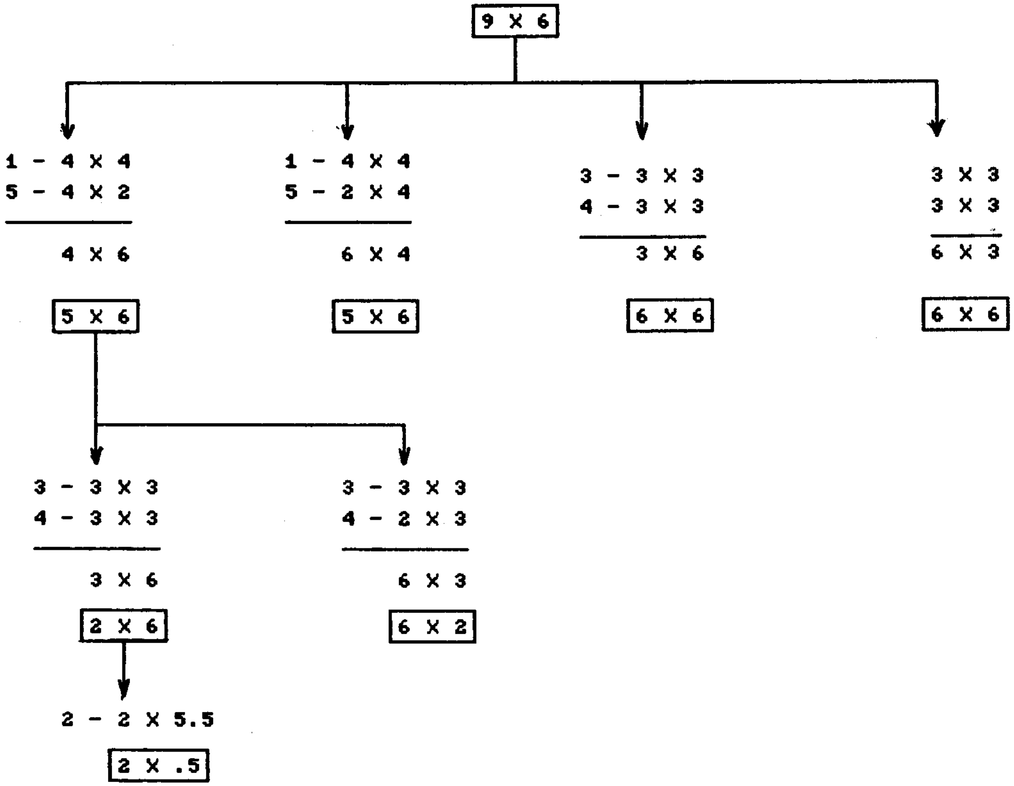

Fig. 4

4. Starting at A″ the largest sheet is 5.5 x 2 so this can be cut from 6 x 2 and the remainder is 2 x .5 and this is wasted area.

5. The program then plot the output (Fig. 5).

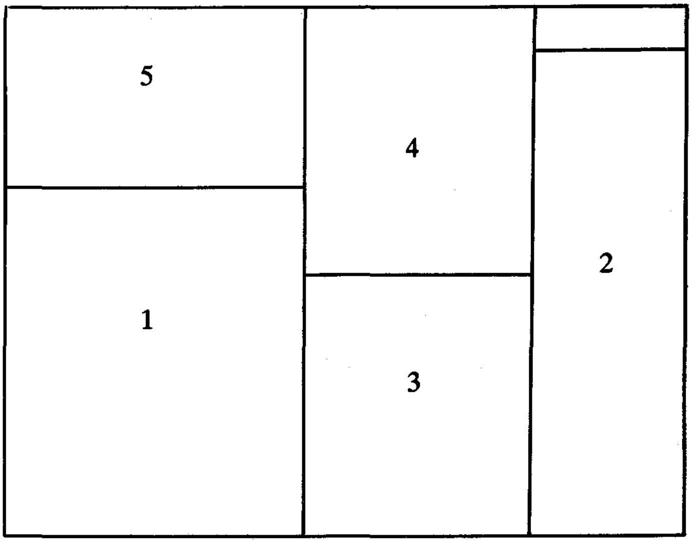

Fig. 5: Output of Example 2.

CONCLUSIONS

This topic is introduced based on ideas of Floorplanning and Electronic Integrated Circuits. However, in electronics there are relations between different blocks or functions such as wiring, speed, heat disapation, etc. but here in the process of glass cutting no relations if all required sheets are from the same type. The algorithm developed satisfies the requirements of this kind of problem since it is discrete and not continuous.

The result achieved was encouraging. The two examples described show how the algorithm is implemented and tested. We are intending in the future to interface the developed program to a robot to demonstrate the technique of cutting sheets of glass from a given sheet since this facility is readily available at Qatar University.

REFERENCE

1. **Barr, A. & Felgenbanm, E.A., 1981,** The Handbook of Artificial Intelligence. William Kaufmann Inc.