Packing of Convex Polygons in a Rectangularly Bounded, Non-Homogeneous Space

Ravi Kothari and P. Klinkhachorn

Department of Electrical and Computer Engineering
West Virginia University
Morgantown, WV 26506-6101

ABSTRACT

Given an arbitrary set of C Convex polygons, compute the optimal placement of these polygons in a 2-dimensional rectangularly bounded space. An optimal solution to this problem has eluded mathematicians for many a years now [13]. The problem however, is of practical importance to many industries. The lumber industry would like to utilize the maximum possible surface area, while recovering furniture parts from lumber boards. The cargo industry would like to accommodate more cargo in a 3-dimensional space. Existent solutions to the problem restrict themselves to the realms of rectangular geometry (convex or irregular shapes are approximated as rectangles).

In this paper we present a 2-dimensional heuristical packing strategy, capable of achieving a dense packing of convex polygonal shapes. We also present a method to extend the placement strategy to non-convex n-gons. The algorithm was developed as part of a system to automate the various aspects of the hardwood manufacturing industry. The techniques developed however, are applicable to the packing problem in general.

INTRODUCTION

The availability of increased and inexpensive computing power has triggered the introduction of computers into several industries, the lumber industry being no exception. The possibility of involving computers in the decision making process was investigated giving rise to an array of sophisticated systems to increase both the throughput and efficiency in the remanufacture of lumber.

One such system was proposed by the McMillin et al. and was acronymed ALPS - Automated Lumber Processing System [1], [2]. The ALPS system in its completed state is expected to have the following skeleton.

1) A non destructive computer vision system to establish defects on lumber boards [3].
2) A yield optimization program that would use the above information, and would determine an efficient placement of cuttings, to conform to the manufacturers cutting bill [4].
3) A laser cutting system to recover the cuttings placed by the yield optimization program [5], [6].
4) A computer program to grade hardwood lumber according to the rules of the NHLA [7].

The ALPS system besides other things relies on two factors - Firstly, development of well behaved computer programs to determine the placement of cuttings on a board. Once the placement of cuttings has been determined, laser beams could be used to “punch-out” the cuttings placed. This paper addresses the second of the four sub-systems of ALPS.

The packing problem is of immense practical importance to various industries, though with a different set of constraints. At the outset therefore we would like to present the problem as applicable to the lumber industry.

A typical lumber board is illustrated in Figure 1. The board specifications along with a list of the quantity and specification of the polygonal shapes (hereafter used interchangeably with cuttings) to be remanufactured (cutting bill) from the board are provided. There are of course a number of ways in which the cutting bill, or a part thereof, could be recovered from the board. However, the cuttings should be placed on the board so as to:

i) yield pieces in accordance with the cutting bill
ii) be free of defects
iii) be aligned with the direction of the grain for strength and aesthetic reasons
iv) utilize the maximum percentage of the defect-free surface area of the board

Figure 1: A typical board

The first three of these four restrictions are self explanatory. The fourth one deserves mention. The cutting bill provided by the furniture manufacturer, would have cuttings of various shapes and sizes. It is easy to achieve a

CH2714-4/89/0000/0200$01.00 © 1989 IEEE
dense packing by placing the smaller cuttings specified in the cutting bill. Such a solution however, has far reaching consequences. The reader is again referred to Figure 1. It can be observed that a large clear area may often be difficult to find on this or subsequent boards. Thus, the larger cuttings should be given priority in the placement procedure. These large cuttings have often been referred to in the literature as cuttings with a larger "value".

An earlier attempt was made by Klinkhachorn et al. [4] to increase the yield when the cuttings to be recovered were rectangular in shape. Lumber is however, processed into peculiarly shaped secondary products. Approximation of these complicated shapes by rectangles inevitably diluted the optimization results obtained thereafter. This paper extends the solution to the general case of a convex polygon, thus allowing for a realistic and a more efficient modelling of the actual cutting shapes. We shall also show that the cutting placement strategy can be extended to more complicated shapes by performing suitable 'enveloping' operations.

**PRELIMINARIES**

The packing problem as it applies to the lumber industry does not lend itself to exact mathematical optimization techniques. An exact mathematical formulation would require a well defined objective function that could be maximized or minimized. Such an objective function is not readily available for the cutting placement problem. This can be attributed to the conflicting solutions arising out of increasing the area of the board utilized and increasing the value recovered from the board. The solutions arising from the two approaches can be shown to be not necessarily the same. The randomness of the board sizes, the defects and the shapes of cuttings further complicate matters. An overall optimization would require a non causal data structure and tremendous computer power making it unsuitable for on-line operation. For all of these reasons the algorithm developed uses heuristic considerations and local optimization strategies.

The concept of giving priority to larger cuttings (more value recovery) is realistic. Many authors have thus advocated the use of Length x Width as a measure of value. Such a measure suffers from two distinct disadvantages:

(i) It presupposes a larger cutting is always longer than it is wider.
(ii) The concept of Length or Width is hard to define for non-rectangular shapes.

We thus adopt area as a measure of value. Conversely, the packing strategy sequences the cuttings in descending orders of area thereby, allowing larger cuttings to be placed first. The decision of which cutting to place first on a board is one critical importance to the global optimization of the placement of cuttings. In the absence of the 'best' sequence to place the cuttings no strategy can be called truly optimal. What can be expected is that the cuttings are placed in an optimal fashion locally, and expect the overall packing to be near optimal. Such an approach has given encouraging results as shown later.

**ALGORITHM DEVELOPMENT**

The algorithm proceeds by placing the cuttings in a locally optimal fashion from left to right on the board. The optimality resulting from the resolve by this algorithm is thus in direct proportion to the development of locally optimal modules. Such a module would consist of identical cuttings and each of these cutting shapes within the module would complement the shape of every other cutting within the module. Each of the individual components of such a module must also be aligned with the direction of the grain. Such a restriction allows for only three operations to form locally optimal modules. They are:

(i) A flip of $180^0$ about the maximum X co-ordinate.
(ii) A flip of $180^0$ about the maximum Y co-ordinate.
(iii) A displacement in the X or Y direction.

Figure 2 shows the sequence of steps involved in finding such locally optimal modules. The operations described above are identified in the Figure as OP1, OP2, OP3 respectively. Notice that for cases where the cutting has more than four vertices ($N \geq 4$) the second component in the module is displaced in the Y direction to allow for the third cutting to fit in.

![Figure 2: Generating Locally Optimal modules (a) for $N \leq 4$ (b) for $N > 4$.](image-url)
The algorithm thus selects the 'largest' cutting first and goes on placing its locally optimal modules in succession. There comes a point when no more cuttings can be placed without violating a constraint (overlapping with a defect, going past the edge of the board etc.). A 'smaller' cutting is then tried. The process is then iterated till no further cutting can be placed in this, the first row of placements.

The reader has perhaps realized that this strategy results in a row of cuttings with sizes deteriorating from left to right and thus resulting in a row with an uneven top edge (Figure 3 (a)).

![Figure 3: Snapshots during a board resolve](image)

Initiating a second row of placements would imply that the cuttings of the second row would have to be shifted down thereby breaking up a locally optimal module. The situation is analogous to a blindfolded man walking on the top edge of the first row of cuttings. The fellow is tiny and takes small steps. He would have a number of 'free falls' (cuttings of the second row) given the situation of Figure 3 (a). The top edge of the first row must then be first smoothed to avoid the possibility of such free falls. This 'smoothing' operation leads to Figure 3 (b). Notice that our mythical man would now have less 'free falls' analogous to the situation where the cuttings of the second row will not have to shifted down, thereby preserving a locally optimal module.

An iterative application of the above steps leads us to the situation of Figure 3 (d). Before concluding this section another points deserves mention. Our strategy mentioned above leaves the possibility of a cutting overlapping another. We use a unique way to detect the occurrence of such an overlap. The process is made somewhat involved since the cuttings can be of arbitrary shapes. All edges of cuttings already placed are viewed as a sequence of closely connected points as shown in Figure 4. To detect the possibility of an overlap if a cutting is placed as shown if Figure 4, it is sufficient to detect if any of the points (dark circles in Figure 4) fall into the region defined by the dotted lines (the cutting to be placed).

![Figure 4: Check for overlap of a cutting with those already placed](image)

The detection of a point within a polygonal region defined by N vertices is illustrated in Figure 5. This is a two step process:

1) Calculate the area of the polygon. For this view the polygon as made up of (N-2) triangles as shown in Figure 5 (a), N being the number of vertices in the polygon. Thus the area of the hexagon in Figure 5 (a) is:

\[
\text{AREA1} = \text{area ABC} + \text{area ACD} + \text{area AED} + \text{area AEF} = 4 + 4 + 6 + 2 = 16 \text{ sq. units.}
\]

2) We now find if the point v' in Figure 5 (b) is within the region defined by the hexagon. For this connect v to every vertex of the polygon and calculate the sum of the N triangle areas resulting. Thus,

\[
\text{AREA2} = \text{area AVF} + \text{area FVE} + \text{area EVD} + \text{area DVC} + \text{area CVD} + \text{area BVA} = 2 + 3 + 4 + 2 + 3 + 3 = 16 \text{ sq. units.}
\]

Since \(|\text{ABS(AREA2-AREA1)}| < \mu, \text{ say } 0.01, \text{ and since none of the sub triangles had an area 0, the point v' is conclusively within the region defined by the hexagon.}

The strategy presented thus far applied specifically to convex polygons. Lumber is however re-manufactured into shapes which are often, but not necessarily convex polygons. We thus model all non-convex n-gons as convex n-gons by finding the convex hull [8], [9]. Such a convex hull is the smallest convex polygon that encloses the non-convex n-gon. The 'enveloped' non-convex n-gon can then be processed as has been described above.
Figure 5: Testing the presence of a point within a polygonal region.

(For the purposes of area calculation let the coordinates of the points be: a(2,4), b(0,2), c(2,0), d(4,0), e(6,2), f(4,4).

USING THE PROGRAM

The algorithm defined above has been implemented into a Fortran program. In its present state, the program can be used as a valuable research tool for further investigations. The program has been structured as menu driven and user friendly. The program allows storage of multiple boards within a single file allowing for batch processing. Facilities also exist to correct any mistakes in the board file or in the cutting file that were discovered later. The program produces a graphic, color coded display showing the result. The display can of course, be bypassed if batch processing is desired.

The program has been written in standard Fortran code and runs on IBM PC and compatibles. An on board RAM of at least 640KBytes and an Enhanced Graphics Adapter (EGA) card is required.

CONCLUSIONS

The results obtained by comparing this algorithm with the one written by Klinkhachorn et. al. are presented in Figure 5. The algorithm by Klinkhachorn et al. [4] has been shown to recover an average of 18% higher yield over the standard yield recovery tables [12]. As can be observed, both the algorithms performed almost equivalently. It is an advantage of this work, in that the extension to the realms of a generalized shape of cuttings was achieved without a reduction in yield. During the course of this work it was observed that the cutting placement strategy is computationally intensive. Efforts are currently underway to reduce the computation time so as to make the algorithm suitable for real time application.

BIBLIOGRAPHY


