OPTIMAL SURFACE CUTTING

Surface cutting problems in two dimensions are considered for nonrectangular items. An exact solution method is discussed. Outlines of several possible heuristic algorithms are also presented. For the heuristic methods a first approximation to the optimal solution is obtained by encompassing each item by a rectangle and then using some available strategy for this standard problem. Different approaches are then suggested for more accurate methods.

1. Introduction

ProActive Technology Pty Ltd is a small computer software company situated in a southern suburb of Sydney. One of the application programs it has developed involves determining the cutting strategy of one dimensional sections to satisfy given orders and so that the associated trim loss is kept to a minimum. The company's request of the Study Group was to evaluate methods for the corresponding two dimensional problem.

There are many applications of these types of problems to industry - paper trimming, bin packing, container loading, capital budgeting - and several classifications according to the specifics of the problem. For a general overview of cutting and packing problems and the relevant literature the reader should refer to Dyckhoff (1990).

The one dimensional problem for cutting sections can be described as follows. A company has in stock several standard length pipe sizes. For example it may have 100 lengths of 4 metre pipe, 80 lengths of 6 metre pipe and 20 lengths of 10 metre pipe. Orders are placed with the company for 15 lengths of 0.3 metre pipe, 27 lengths of 0.7 metre pipe, 32 lengths of 0.8 metre pipe and 16 lengths of 1.2 metre pipe. How should the company cut the pipes from the stock lengths so as to minimise the wastage? The wastage is defined to be the end pieces of pipe that remain after several ordered lengths have been cut from that pipe. If the endpieces are not too small they can be added to stock, but if they are below a certain size they must be discarded.

The two dimensional problem is similar except instead of dealing with lengths we allow two dimensional objects. Just about all applications consider rectangular stock items, and many only have rectangular ordered items as well. For instance, the glass cutting industry cuts rectangular pieces of glass from rectangular stock sheets. The cutting process for that application is special since the glass is cut by guillotines. This type of cut is used in the paper industry as well. Figure 1 illustrates various cuts.

In this discussion paper we will consider rectangular sheets, general ordered items, and general cuts.

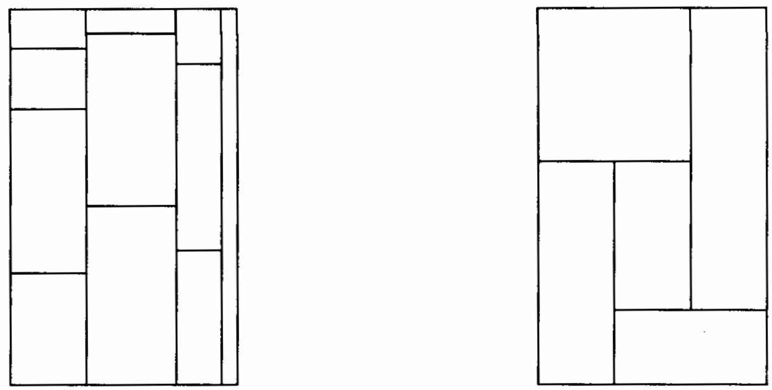

Figure 1: A 2-stage guillotine pattern and a non-guillotine pattern

The variety of applications and the size of the possible problems can be extremely large, especially when we are dealing with nonrectangular shapes. Let us therefore state the range of problems we will consider. In line with the applications that ProActive Technology meets in the metal plate cutting area, we assume the following

1. The problems are small to medium in size; approximately 100 items requiring 10 to 20 sheets.

2. There are no restrictions on the cutting patterns, so we are not limited to guillotine cuts. In the metal plate industry the items are cut from the plates using oxy-acetylene torches.

3. The ordered items can be any shape. These may include rectangular shapes as well as several types of nonrectangular ones.

4. There will only be one size of stock sheet. This is in general not true. However if we pack the items appropriately with a preference for packing into rectangular regions on the plate, then, at the end, instead of ordering that complete sheet, the smallest plate that encompasses the used area is ordered. So after a packing has been determined, a variety of plate sizes are used making this a less restrictive assumption.

Even with these assumptions we still face a daunting task. This is based on the knowledge that even the rectangular problem is NP-complete. NP-complexity refers to the computational complexity of algorithms. Most efficient algorithms in combinatorial optimisation are those where the number of computations required to determine the optimal solution is polynomially related to the size of the problem. Such algorithms are called polynomial algorithms and they have the property that, as larger problems are encountered, the time taken to compute a solution grows at a reasonable rate.

The worst problems as far as computational complexity are those that require exponential algorithms, that is the number of computations required to solve the problem for a worst case scenario is an exponential of the problem size. Since exponentials grow exceedingly quickly, a small increase in problem size may make the problem computationally intractable.

NP-complete problems fall in between these two classes except that there are no known polynomial algorithms that will solve the problems. These may exist but at present we must rely on exponential algorithms or use heuristic algorithms where the optimal solution might not be found, but perhaps a good solution will be determined in a reasonable amount of time.

With this knowledge, we see that a search for an optimal solution will in general fail. The hope is that the assumptions we have described above will make the calculation of a good solution computationally reasonable.

2. Solution methods

One method for these problems encountered in the literature generates patterns and then determines the best collection of patterns to use. This is based on the succesful approach for one dimensional cutting problems proposed by Gilmore & Gomory (1961). A pattern contains a specific number of ordered items that will be cut from a sheet in a prescribed manner. For example a pattern might contain 3 pieces of item 1, 2 pieces of item 3, and 8 pieces of item 6. The way these are cut from the sheet is also given. The cost of the pattern equals the amount of wastage.

Let us describe this formally. Define

c_j = the cost of pattern j
x_j = the number of times pattern j is used to fill an order
a_{ij} = the number of times item i is used in pattern j
d_i = numbers of item i ordered

The optimisation problem describing the cutting stock problem is then

$$\begin{aligned} \text{minimise} \quad & \sum_j c_j x_j \\ \text{subject to} \quad & \sum a_{ij} x_j \geq d_i \\ & x_j \geq 0 \\ & x_j \text{ integer} \end{aligned}$$

One of the first difficulties encountered with this method is determining the patterns. Some simple patterns may be generated but a good pattern may be difficult to find. If the method is to work well it requires a large number of patterns to choose from. Most of these patterns however will be discarded from the final solution.

For industries that require many pieces of each item and where the number of items is large, this is a reasonable approach. For the problem types we investigate however, the great deal of initial work in generating the patterns cannot be justified. Hence we did not consider this to be a fruitful technique.

3. An exact method

One approach that was discussed was to convert the placement of items on sheets to an optimisation problem. The model can be defined precisely but to give the basic idea define the following: for each item $A^{(i)}$ let $y^{(i)}$ represent a measure of the placement of the item on the plate. This may include several components such as height, unused space generated, *etc.* Then the problem can be stated as

$$\begin{aligned} &\text{minimise} \quad && \sum_i y^{(i)} \\ &\text{subject to} \quad && \text{dist}(A^{(i)}, A^{(j)}) \geq 0 \qquad \forall i,j \end{aligned} \tag{1}$$

We can specify the objects by their corner points, approximating any rounded edges by piecewise linear ones. The constraint (1) would be implemented via the separation of the corner points of each object. Except for the approximation of items with rounded edges by ones with piecewise linear edges, this method: in combination with a general optimisation algorithm: would be exact. However it would entail large computing needs.

Some of the properties of the problem that would contribute to the lengthy computing time are

Rotation: each item must be considered in all orientations.

Constraint feasibility: no two items can overlap so there is considerable constraint checking involved.

Local optima: because of the constraints there will be many local optima which will repeatedly stop the search for the true minimum.

Nevertheless this approach is discussed for rectangular items by Beasley (1985). For small problems it may very well be the best approach and even for large problems may contribute by its application to solve small subproblems.

4. Heuristic methods

Rather than trying to solve the problem exactly, we can approach our task with the more modest aim of trying to find an acceptable solution in a reasonable amount of time. The methods discussed below are heuristic and have a similar structure to each other but with different implementations. The steps for these methods follow the template given below

Step 0 specify the items.

Step 1 presort the sequence of items in some way such as

a) into groups with area approximately the same as the area of the sheets, or

b) by area, or diameter, or ...

Step 2 place the items onto the sheets using some method.

Step 3 evaluate the placement according to some objective.

Step 4 perturb the sequence and return to Step 2.

This strategy can be implemented in two ways. In both of these however, we try to fit each item by either a rectangle or a union of rectangles. The preference for rectangular items stems from three reasons. First there are many heuristic methods that are available in the literature for rectangular regions while those for nonrectangular regions are few. Second, it is quite easy to determine good placements of rectangular items next to each other; simply place an edge of one item next to the edge of the other item. On the other hand this is not as straightforward for nonrectangular shapes, and since a great deal of the method relies on close packing of items this causes some difficulty. Third, rectangular regions have only two orientations - on their side or standing up - while nonrectangular items can have an infinite number of orientations. Fitting an item according to a particular orientation may not be as good if it were oriented in a different way. When all different orientations for all the items are considered the computation becomes large.

Our guide as to "good" placement of items onto a sheet is the minimisation of waste. However not all unused area is waste. We started with the assumption that the stock sheets are all one size; in the end however, we order the smallest size sheet available that covers the items placed on the standard one. Hence we do not want to count as wasted space any part of the sheet that can be omitted when the sheet is purchased. Therefore when we are considering placing items on a sheet, we try to put them into a rectangular subregion of the plate. To achieve this, we put them as far to the left as

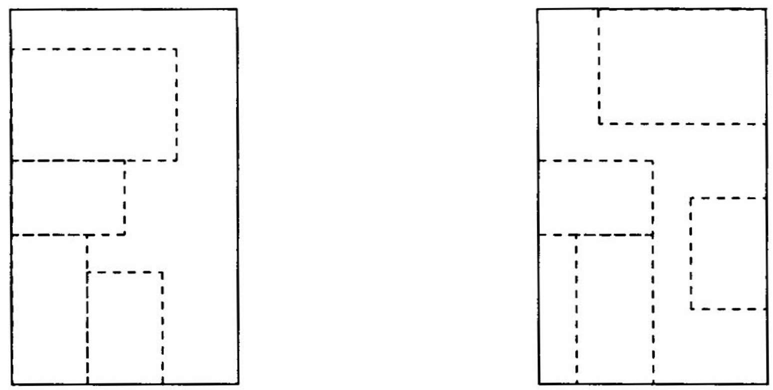

Figure 2: Good and bad placements on a sheet

Figure 3: Converting a triangle to a union of squares

possible and as far to the bottom as possible. Combining this approach with some other appropriate strategies helps achieve our aim.

Implementation 1

The first approach we consider is where we immediately try to specify the items in a rectangularised fashion as closely as possible. For this we choose a grid on which to lay each nonrectangular item. Those squares of the grid containing any part of the item are then used for the description of the item. The new description then consists of a union of squares. For example if the item were a right angle triangle with the two sides forming the right angle running parallel to the lines of the grid, then the hypotenuse would be replaced by a sequence of vertical and horizontal segments - it would now look like a set of stairs.

Now that the items have been rectangularised, we must determine their placement on the sheets. If we are using method (a) in Step 1, then the items within a group can be placed on the sheets in any order. If we are using method (b), then we place them on the sheets according to whichever is next in line, if at all possible.

The placement strategy can be any of the heuristics found in the literature. For example we could use a bottom left heuristic where each item is placed to the bottom of the sheet and as far left as possible. Or we could use a shelf heuristic where the sheet is divided up into shelves of various height and the items are placed on the lowest shelf where they will fit and as far to the left as possible.

Since we are here allowing the items not to be rectangles but more general shapes then we should decide how to place the items next to each other. For the simplest case when we have two items only on the sheet with the first placed at the bottom left corner and the second on the bottom of the sheet as well, we can put them together so as to minimise the intervening space, or maximise it, or according to some other strategy. Trying to minimise the space seems an obvious approach because we then have as much space remaining in one connected section as we can obtain. However the other approach of maximising the intervening space between two items that are placed next to each other also has its advantages. At the end of the entire placement we will probably have some small items that could fit into these large gaps. Making these gaps as usable as possible makes the task easier.

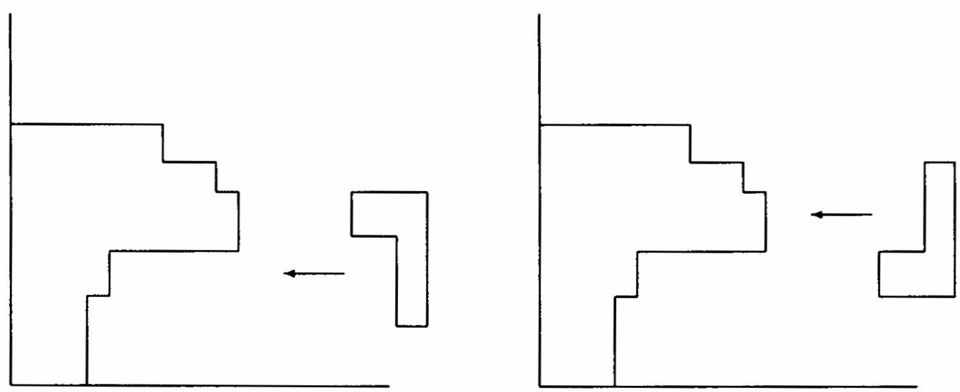

Figure 4: Maximising and minimising spaces

Implementation 2

Another approach that can be tried is to only use rectangles. As an initial approximation to the optimal solution, this is not a bad start. Let us consider the process a metal plate cutting company follows to bid for a contract for supplying the components for say a bridge. First they are given rough plans of the bridge which an engineer studies to determine what components are needed. These are not necessarily included in the plans. Given the engineer's assessment, an estimate is made of the amount of metal plate and sections needed. This is not accurate because a detailed analysis would require a significant investment in time by the engineer which would be wasted if the company's tender for the job failed. So it is an educated guess based on experience and with a margin for

Figure 5: Enclosing a triangle in a rectangle

error. Using rectangles for the items at this initial stage is probably as accurate or more so than what is done in practice.

If the company wins the tender they use detailed plans to get an accurate specification of the shape and size of each component. They then estimate as precisely as they can the amounts of plates and sections they will need. This also builds in an estimate of the waste that will be generated by their cutters. The profit on the contract will be the difference between the amount they submitted in the tender and the cost of the metal with labour that is actually used.

At this last stage, if they can reduce their estimate of the plate and section required then they will increase their profit - so it is important to have a good surface cutting scheme available. For this stage, we therefore want to improve on the initial estimate as much as possible. In terms of the scheme proposed here, we try to do this by considering groups of items enclosed in single rectangles, rather than by simply encompassing each nonrectangular item in a rectangle.

For example, suppose two of the components are a triangle and a circle. The initial process places each in a rectangle and solves that problem via some heuristic. The second process may combine these two items into one rectangle, and along with other combinations, uses an heuristic to obtain a solution. Since the space used for this one rectangle will be less than that used by one rectangle for each item, there should be some improvement.

Taking several attempts at this combination process and resolving will in general find a better solution than the original one.

So the strategy is as follows:

Step 1 place rectangles around each item.

Step 2 use a plate cutting method for rectangles to find a solution. If an appropriate

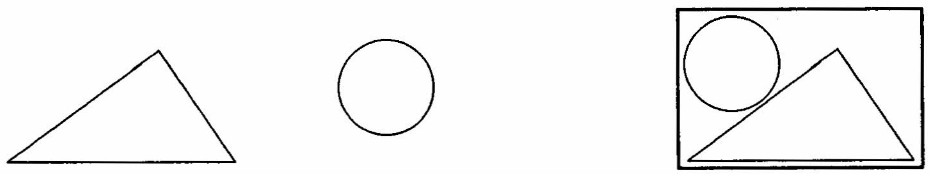

Figure 6: Enclosing a triangle and a circle in a rectangle

convergence condition is satisfied then Stop. Otherwise go to Step 3.

Step 3 group items together into single rectangles. Go to Step 2.

Steps 1 and 2 give a suitable answer for the initial tendering stage. Carrying on with several passes through Step 3 will in general improve the solution and give the best cutting strategy.

Orientation of objects is a smaller concern here than in previous strategies because the only time it occurs is when we place individual items or groups of items in rectangles. This is a smaller scale problem than for the other approaches.

5. Discussion

This is an easily stated problem but – as so often occurs in combinatorial optimisation problems – it hides a difficult computational problem. We have listed several strategies that could be implemented; until these methods have been programmed and tested on representative examples we can make few assertions about which will perform best. In fact it may very well be that several of these strategies should be combined to take advantage of each of their strengths and cover more possibilities.

The Study Group has therefore provided suggestions that ProActive Technology might follow, even though most of the work remains to be done.

Acknowledgements

Many people contributed to the discussion and made suggestions of strategies for this problem. Our guides in this were Paul Saunders and Tim Young from ProActive Technology and also Wal Stinson from the metal plate cutting industry. The other substantial contributors were Roderick van Beelen, Natashia Boland, Andreas Ernst, John

Hewitt, Mohan Krishnamoorthy, Nian Li, John Lions, Fawang Liu, Rhys Newman, Duncan Roper, Tony Watts, Mark Westcott, Kevin Wilkins, Matt Williams and Simone Zmood.

References

J.E. Beasley, "An exact two-dimensional non-guillotine cutting tree search procedure", *Op. Res.* 33 (1985), 49-64.

H. Dyckhoff, "A typology of cutting and packing problems", *Eur. J. Op. Res.* 44 (1990), 145-159.

P.C. Gilmore & R.E. Gomory, "A linear programming approach to the cutting-stock problem", *Op. Res.* 9 (1961), 849-859.