INPE-5383-PRE/1749

TWO-DIMENSIONAL CUTTING STOCK: BOARD LENGHT DETERMINATION

Horacio Hideki Yanasse
Alan S. I. Zinober
Rrginald G. Harris

INPE
São José dos Campos
Abril 1992

SECRETARIA DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-5383-PRE/1749

TWO-DIMENSIONAL CUTTING STOCK: BOARD LENGHT DETERMINATION

Horacio Hideki Yanasse
Alan S.I. Zinober
Reginald G. Harris

This paper was presented at EURO XI - European Congress on Operational Research, RWTH Aachen, July 16-19, 1991

INPE
São José dos Campos
April 1992

CDU: 519.8

Key Words: Cutting stock problem; Assortment problem; heuristics

Two-dimensional cutting stock: board length determination

Horacio Hideki Yanasse *
Alan S.I. Zinober **
Reginald G. Harris **

* INPE/LAC - São José dos Campos-SP - BRAZIL
** University of Sheffield - Sheffield - Inglaterra

Abstract

The problem of determining the optimal board length from which panels have to be cut to satisfy some specified demand with minimum waste is analysed. Two heuristic procedures are suggested. The first procedure uses bin-packing heuristics successively; the second method uses a branching scheme where potentially good board lengths are analysed.

Keywords: cutting stock problem; assortment problem; heuristics

Previously entitled "Cutting stock: board length determination", this paper was presented at EURO XI - European Congress on Operational Research, RWTH Aachen, July 16-19, 1991.

INTRODUCTION

We consider a cutting stock problem in a furniture manufacturer setting, where rectangular panels of different sizes and quantities have to be cut from rectangular boards. Manufacturers in the furniture industry, for instance, are often able to specify to suppliers of the boards the precise length of board they require, rather than accept standard board lengths. This may be on a job to job basis, or the length may stay constant for some period of time.

Many manufacturers often have available for their use some kind of optimising program that generates "good" cutting patterns and corresponding quantities, given a set of jobs[1]. Most of these programs assume the board lengths are given. Obviously the manufacturer could run such an optimiser program many times with various board lengths and take the one which produces the least waste. However this is time consuming and prone to error.

Little has appeared in the literature on the assortment problem which is the problem of determining what should be the stock sizes. We have studied a few papers, two in the glass industry context[2-3], and another in the paper industry setting[4]. A fourth paper in the furniture manufacturing context[1] analyses a closely related problem when different board types from which the panels can be cut, are available and the best mix of boards, which minimises the cost, has to be determined.

We propose here two heuristic procedures to determine the best board length to order. The first heuristic is straightforward and uses a sequence of bin-packing[5-7] and job scheduling[8] heuristics. Its application is most suitable for cutting stock problems with special restrictions, for instance, those that require orthogonal guillotine cuts[1].

The second heuristic procedure is more general in nature and can be used in any cutting stock problem. However, it requires an optimiser program that provides a solution to the cutting stock problem given the board length. The procedure repeatedly generates potentially good board lengths which are analysed by the optimiser program until some stopping condition is achieved. The approach is quite similar to a branch-and-bound method. It was developed primarily to be used for solving small sized problems.

Before starting the description of the methods we make the observation that the minimum and maximum lengths required are generally known and they are going to be denoted L_{min} and L_{max}, respectively.

Method 1

Consider an infinitely long, fixed width rotated board (bin) (Figure 1).

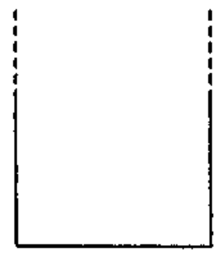


Figure 1
Rotated board of infinite length

Insert a panel in the bottom left hand corner according to some rule, e.g. largest, widest, etc, and form a strip or shelf. Fill the strip using an appropriate rule, for instance, following some priority list. Insert the next panel in the list that fits in the available empty space(s). Continue in this manner up the board (or bin) until all panels are fitted (see Figure 2). Any shelf bin-packing heuristic[9-10] would be applicable.

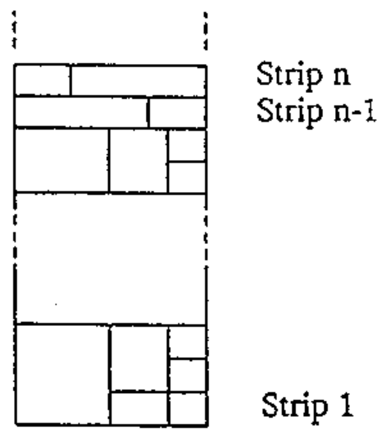


Figure 2
Cutting pattern for a board of infinite length

We now divide the board into its strips (see Figure 3). The aim is to produce a grouping of these strips into equal length boards with minimum loss.

Strip 1 Strip n-1 Strip n

Figure 3

Given the maximum possible board length of L_{max} and the fixed board width, it is possible to estimate the minimum number of boards (N_{min}) into which these strips could be fitted. Each strip traps some waste so this calculation does not use the original panel area but the total area of the strips. Try to fill N_{min} boards with these strips: this is similar to a machine shop scheduling problem[8] where the strips represent jobs and the boards represent machines. A possible strategy to follow is to fill the board having the greatest waste remaining with the largest strip still in the list. If the strips do not fit into N_{min} boards, start again with $N_{min}+1$, then $N_{min}+2$, and so on until a feasible solution is found. Call this N_{smin}.

Now fit the strips into $N_{smin}+1$ boards (use, for instance a heuristic for minimising the total makespan in a job scheduling problem with $N_{smin}+1$ machines) and observe the greatest total strip 'height' in any board: this would be the final board length for that number of boards because clearly every other total strip height would fit into it. In Figure 4 we illustrate a possible output obtained at this stage.

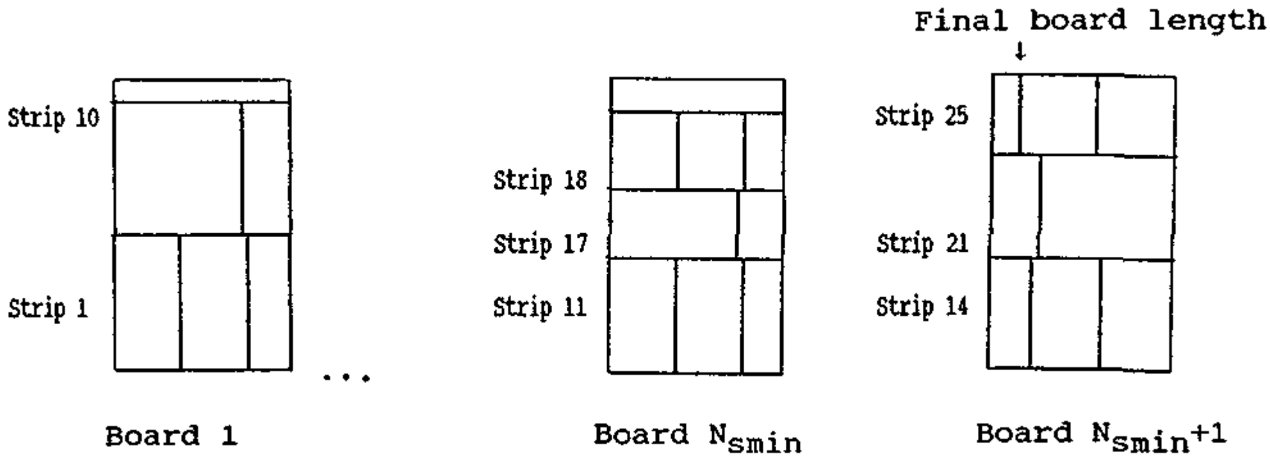


Figure 4

Strip groupings with $N_{smin}+1$ boards

Repeat the previous process with $N_{smin}+2$ boards, then $N_{smin}+3$ boards, etc. until the calculated final board length falls below the minimum final board length L_{min}. The solutions are summarised in Table 1.

Table 1
Solution of the bin-packing problem

Boards	Final board length	Board area used
N_{min}	<no solution>	
$N_{min}+1$	<no solution>	
$N_{min}+2$	<no solution>	
.	.	
.	.	
N_{smin}	L(0)	A(0)
$N_{smin}+1$	L(1)	A(1)
.	.	.
.	.	.
$N_{smin}+k$	L(k)	A(k)
$N_{smin}+k+1$	<less than L_{min}>	

The chosen board length is that which is associated with the min [A(0),A(1),...,A(k)].

The drawbacks of this approach are:

a. it assumes that boards can be cut in their rotated orientation. Alternatively if the patterns are returned to their normal orientation before cutting, it assumes that the resulting patterns are both feasible and economic for a particular manufacturer's saw operation;

b. patterns are built up from pre-determined strips. The limitations on the patterns have a direct impact on the quality of the solutions because, for the same number of boards, other acceptable patterns may produce panel arrangements with overall smaller wastage.

The major advantage of this approach is that it is very simple and yields a solution very rapidly.

Method 2

Method 2 relies on running an optimising program for a set of board lengths. We assume, therefore, that we have a program that finds a solution to the cutting stock problem given the board length.

The search path is best described graphically as depicted in Figure 5. The axes are total board area (vertical) against board length (horizontal). The objective is to find the lowest horizontal line, corresponding to a particular board area, on which there is a solution. The point on the line where this occurs determines the optimal board length.

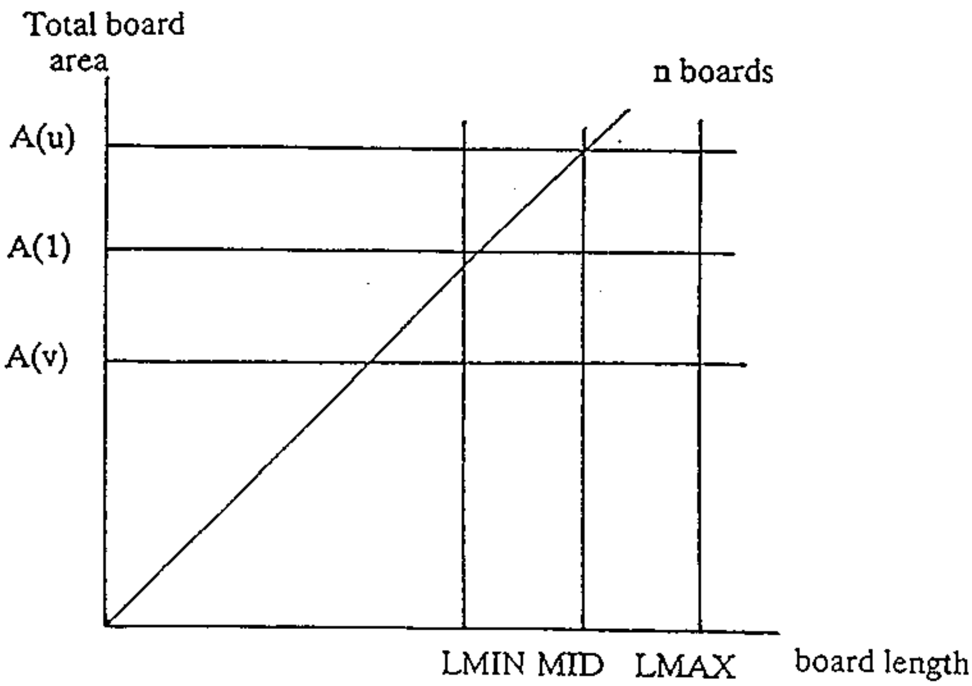


Figure 5

Run the optimiser using the midpoint MID of the board length range [LMIN,LMAX] and note the board area used. Call this the upper area limit A(u) and call the total required panel area the lower area limit A(v).

[1] Find an area in between (e.g. half way, one third, golden section) say, half way; call it A(1). We have found a solution at A(u); we now want to find if there is a solution at A(1) (see Figure 5).

Draw lines across the diagram (see Figure 6), each corresponding to the number of boards in the solution. The first line is that shown in Figure 5 and goes through the origin and the point (MID,A(u)). Other lines are drawn either side through the origin and the points (A(u)/i, A(u)), where i=...n+2,n+1,n,n-1,n-2... is the number of boards in the solution. Clearly these points do not relate to alternative solutions: they exist to motivate the rest of the method.

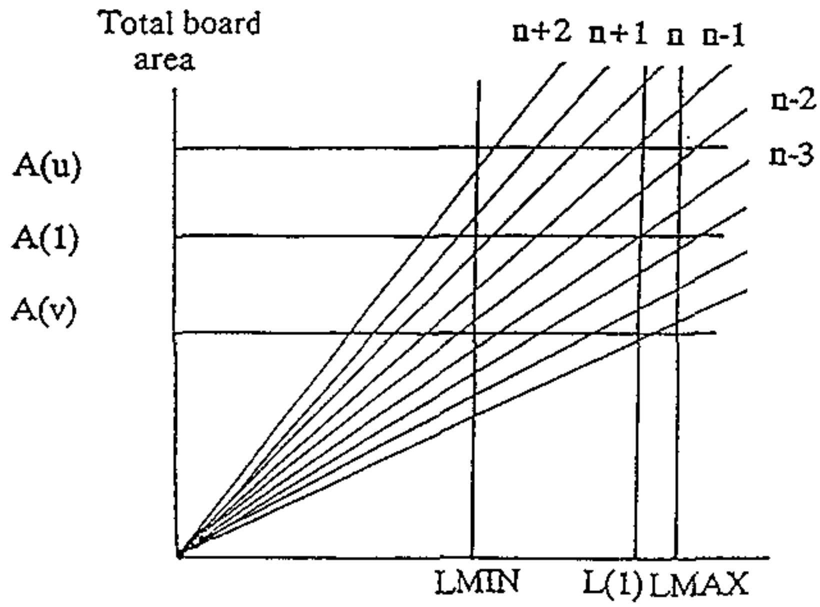


Figure 6

[2] Find the largest board length less than LMAX (L(1), say) where there is an intersection on the line A(1). Run the optimiser for this board length and obtain the number of boards in the solution. The solution will be represented by one of the intersections above L(1) (see Figure 6).

If the solution relates to an area greater than A(u), repeat from [2] with the board length corresponding to the next lower intersection on the horizontal A(1) line. If there is no solution, A(v) is updated to A(1); stop if abs(A(u)-A(v))< ϵ (a pre-specified error).

[3] If the solution relates to an area less than A(u), that area becomes the new upper limit. Tune the result locally - see details below. Repeat the process from [1].

Keeping 'solution intervals'

Every time the optimising program is run with a particular fixed board length, the corresponding number of boards is obtained. It is possible to construct 'solution intervals' for each number of boards as the search process proceeds.

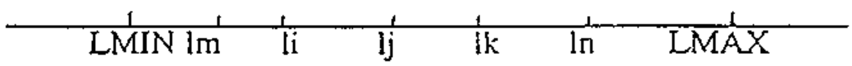


Figure 7

Suppose we have already run the optimiser for l_i and l_k (see Figure 7) and for both lengths we obtained n^* boards. It seeems reasonable to assume that within the interval $[l_i,l_k]$ the solution from the optimising program will also be n^* (it would be true if the program used an exact method).

Hence, if during the search process, we obtain a board length l_j, where $l_i<l_j<l_k$, we do not need to run the optimising program again since it is most likely that the number of boards for l_j will be n^*. Suppose we run the program for l_m (or l_n) and we also get n^*. The new solution interval for $n*$ can then be updated to $[l_m,l_k]$ (or $[l_i,l_n]$).

By storing these intervals we can reduce the computational time considerably.

Tuning

If a better solution is obtained in [3], local improvements might be achieved by adjusting the board length slightly, as follows. Figure 8 illustrates the true, jagged line of possible solutions if we were using an exact method.

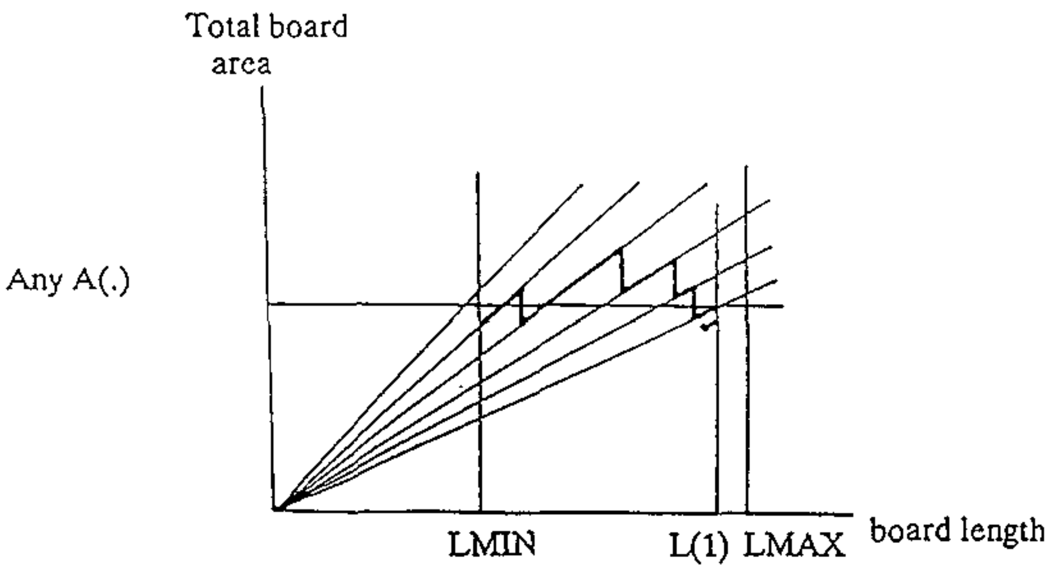


Figure 8

[4] We can reduce the board length by delta (a value chosen conveniently, for instance 1% of the board length) and run the optimising program (if this new board length value does not fall within any of the known solution intervals).

If the number of boards is unchanged, we have a new (better) solution, and we can repeat from [4].

If the number of boards increases (also by checking the solution intervals), try half way between the previous board length and the closest smaller known upper bound length limit (upper limit of the solution intervals) for the higher number of boards. (The solution intervals should be updated every time the optimising program is run.) Keep bisecting the appropriate distance (until it reaches a predetermined value), running the optimising program and updating the solution intervals.

As a final adjustment, it is often possible to inspect the final patterns and find that the board length can be rounded down or even reduced by a small quantity.

Examples

We present some limited computational results to illustrate the algorithms. The relevant data of the test problem is given in Table 2. The optimising program used to generate a solution to the cutting stock problem given the board length, is described in

Yanasse, Zinober and Harris[1]. Those unfamiliar with the cutting terminology used can refer to ref 1. The cutting patterns are constrained to contain guillotine cuts.

Table 2

SAWKERF:	5
MAXIMUM NUMBER HEAD CUTS:	2
TRIMS:	
TOP:	5
BOTTOM:	0
RIGHT:	5
LEFT:	0
HEADCUT:	5
BOARD ROTATABLE FOR CUTTING:	NO
BOARD WIDTH:	1220
BOARD LENGTH RANGE:	
	[2440,3660]

PANEL NUMBER	LENGTH	WIDTH	DEMAND
1	850	550	22
2	850	700	16
3	900	550	10
4	900	700	10
5	1100	550	2
6	1100	700	2
7	1300	550	2
8	1300	700	2

Total panel area = 37,055,000

For this particular data set, the board cannot be rotated for cutting. Method 1 is not strictly applicable as the patterns in their normal orientation might not be feasible for particular saw constraints. For the sake of illustration only we apply method 1 to this data set.

Method 1

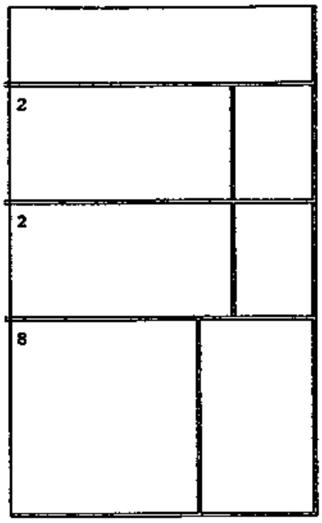
Method 1 was implemented with several passes using different priority lists, by length, width, area, weighted area (demand*area), and the best solution was chosen. Initially we assume a board of width 1220 and infinite length. The necessary total board length to cut all panels is found to be 30660.

Hence, a lower bound on the number of boards is 9 (30660/3660 rounded up). The optimal board length obtained using method 1 is 3170. With this board length 13 boards are used to cut the panels. The total board area for this solution is 50,276,200.

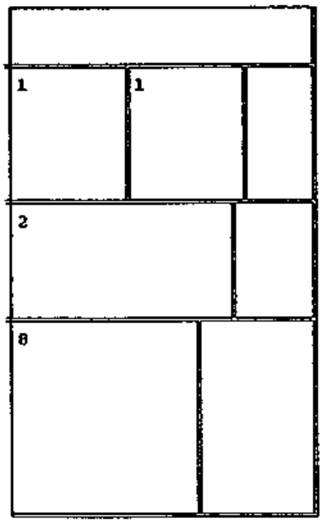
The computer running time to obtain this solution was 89 seconds on a 286/AT PC (Computing index equal to 13.7 compared to an IBM PC/XT).

Figure 9 presents the complete solution of the problem given by Method 1. The patterns are shown in the orientation and order in which the method created them.

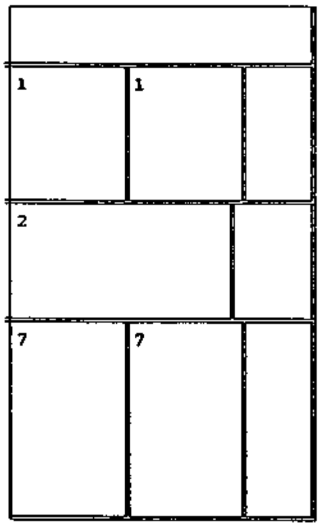
Board 1



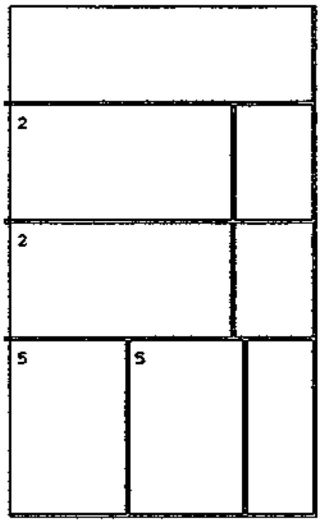
Board 2



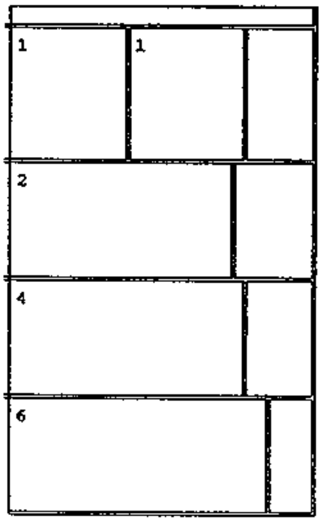
Board 3



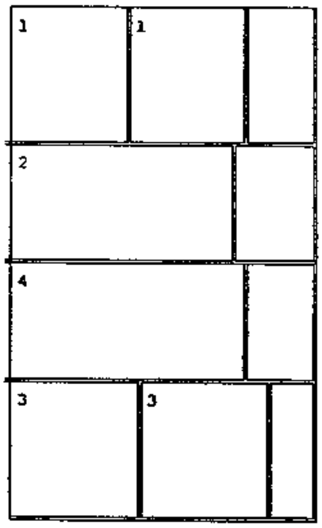
Board 4



Boards 5 and 6



Boards 7,8,9 and 10



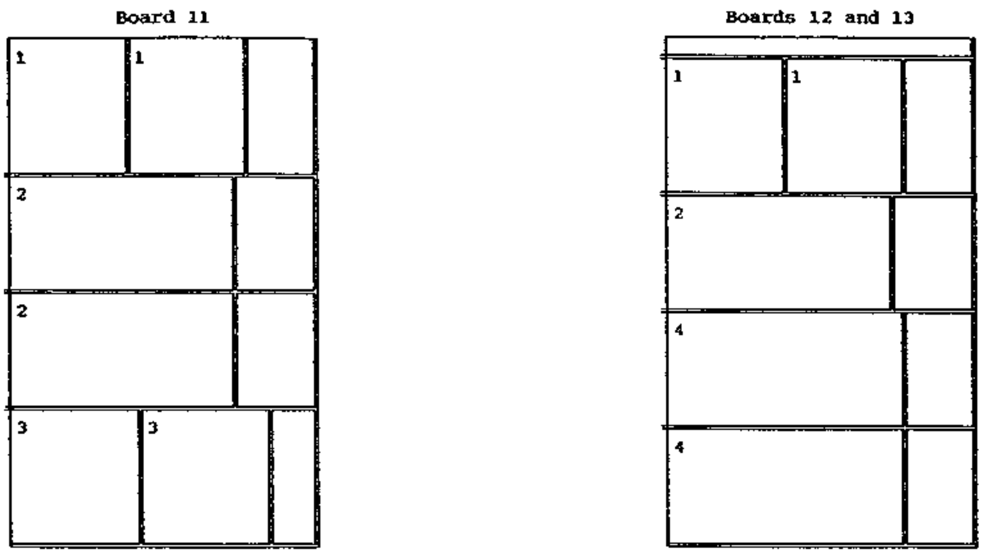


Figure 9
Solution of the problem given by Method 1

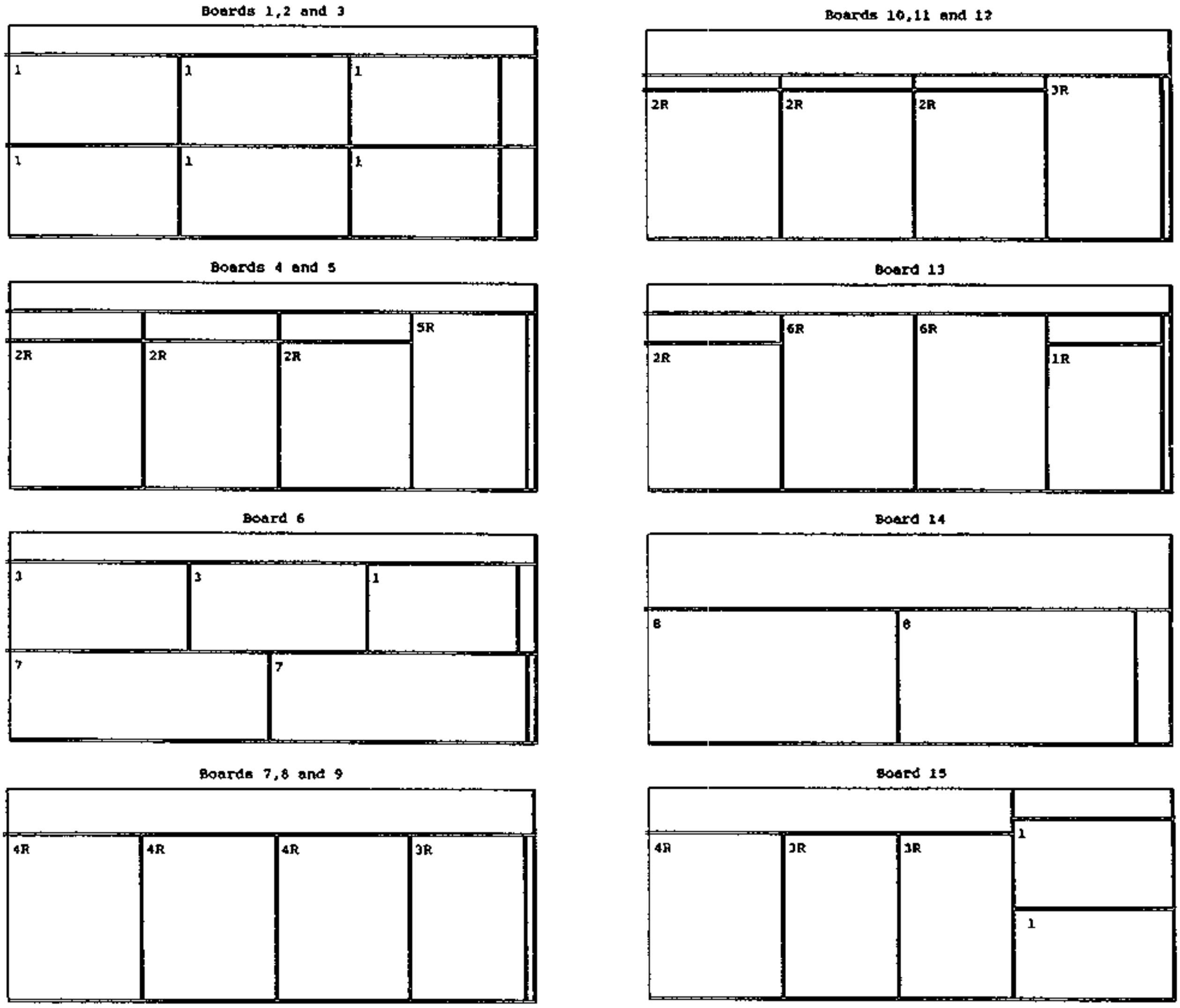
Method 2

Table 3 presents the board lengths obtained successively when we apply Method 2 to solve the same problem. In column 2 of the table, the number of boards necessary to cut all the panels is presented. These values are given by running the optimising program given the board length. Column 3 of the table gives the solution for the cutting stock problem given by the 'solution intervals', that is, we previously found that for some board length xlow, the number of boards necessary to cut the panels was n and for another board length xhigh, xhigh > xlow, the same number of boards was required. Therefore, we assume that any board length in between these points will also require n boards to cut the panels. There is no need to run the optimising program for these points. Tuning was included and the delta value used was 25. The bisecting process was aborted since an error of approximately 1% was considered good enough. Also, the pre-specified error e was set to 1%.

Table 3
Potential board lengths tried by Method 2

board length	solution by optimising program	solution by interval
(STARTING POINT)		
3050	14	
3333	14	
3055		14
2819	15	
2794	15	
2769	15	
2744	15	
2719	15	
2694	15	
2669	17	
3552	12	
3229		14
2960	15	
2732		15
2536	18	
3452	12	
3164		14
2920		15
2712		15
2530	18	
3564	12	
3267		14
3015	15	
2799		15
2612	17	
2449	18	
3620	12	
3318		14
3062		14
2843		15
2653		17
2487		18
3648	12	
3343	14	
3086		14
2865		15
2674	16	
2506		18

The running time taken to obtain the final solution was 209 seconds. In Figure 10 we present the patterns obtained with the best board length selected, which was 2694.



Total number of boards used to cut the panels was 15.
Total area of the boards is 49,303,860.

Figure 10
Solution given by method 2

This solution represents an improvement of approximately 1.97% over the solution given by Method 1. In fact, by just checking the patterns obtained, we can adjust the board length to the corrected value of 2675 (the patterns shown in Figure 10 are already adjusted to this value). The total area of the boards becomes 48,952,500, an improvement of about 2.7% over the solution given by Method 1.

As can be seen, the running time to obtain a solution for the second method is longer than that for the first method. This is often the case. In larger problems the first method is able to produce a solution in a few minutes but the second method might take a long time, since it relies on the optimising program to produce a solution for every board length tried.

Concluding Remarks

We have presented two heuristic procedures to determine the best board length from which panels are to be cut in a cutting stock problem. Method 1 is of limited use, due to the special nature of the patterns it generates. Method 2 has more general application because the optimising program used to generate patterns can be replaced by any other which the user might feel appropriate for the problem.

In theory, Method 2 can be used to solve cutting problems of any size. From the practical point of view, however, we have to limit its use to small sized problems since the running time depends on how long the optimising program takes to generate a solution. The procedure uses the optimising program every time it finds a potentially good board length. If the problem is large, each optimisation will take a long time. In addition, the number of boards in the final solution might be large implying that a potentially larger number of board lengths may have to be analysed. In this case, there would be no advantages in using Method 2 over a trial and error method which would run the optimising program for some selected values of board lengths chosen conveniently within the allowed range.

We have noticed that the 'optimal' board length seems to be strongly influenced by the odd sized panels which are to be cut. Smaller panels have a reduced impact on the determination of the board length as they can, in general, be accommodated in any smaller spaces left on the boards. This suggests the application of Method 2 only to a subset of the original set of panels to be cut. This subset should include panels with relatively large requirements, large area, length and/or width. The smaller problem so defined would run faster and the first k best board lengths could be checked with the whole set of data and the best board length chosen. Our limited computational test experiments using this approach were quite promising.

Acknowledgements: This work was partially funded by SERC ACME Directorate grant number GR/F/68942.

References

1. H.H.YANASSE, A.S.I.ZINOBER, R.HARRIS (1991) Two-dimensional cutting stock with multiple stock sizes. *J.Opl.Res.Soc.* **42**, 673-683.

2. M.L.CHAMBERS, R.G.DYSON (1976) The cutting stock problem in the flat glass industry: selection of stock sizes. *Op. Res.Quarterly* **27**, 949-957.

3. A.DIEGEL, H.J.BOCKER (1984) Optimal dimensions of virgin stock in cutting glass to order. *Decision Sciences* **15**, 260-274.

4. J.E.BEASLEY (1985) An algorithm for the two dimensional assortment problem. *Eur.J.Opl.Res.* **19**, 253-261.

5. W.T.RHEE (1988) Optimal bin packing with items of random sizes. *Math. of Op. Res.* **13**, 140-151.

6. P.Y.WANG, J.O.BERKEY (1987) Two dimension finite bin packing algorithms. *J.Opl.Res.Soc.* **38**, 423-429.

7. F.R.K.CHUNG, M.R.GAREY, D.J.JOHNSON (1982) On packing two dimensional bins. *SIAM J. Alg. Disc. Meth.* **3**, 66-76.

8. J.K.LENSTRA, A.H.G.RINNOOY KAN, P.BRUCKER (1977) Complexity of machine scheduling problems. *Ann. Discrete Math.* **1**, 343-362.

9. B.S.BAKER, D.S.BROWN, H.P.KATSEFFA (1981) 5/4 algorithm for two-dimensional packing. *Journal of Algorithms* 2, 348-368.

10. B.S.BAKER, J.S.SCHWARZ (1979) Shelf algorithms for two dimensional packing problems. Proc. 1979 Conf on Information Sci. and Systems, 273-276, John Hopkins Univ., Baltimore, Md., Dept. of Electrical Engineering.

PRESIDÊNCIA DA REPÚBLICA
SECRETARIA DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

AUTORIZAÇÃO PARA PUBLICAÇÃO

MFN 6082

TÍTULO

TWO-DIMENSIONAL CUTTING STOCK: BOARD LENGHT DETERMINATION

AUTOR

Horacio H. Yanasse; Alan S.I. Zinober; Reginald G. Harris

TRADUTOR

EDITOR

ORIGEM	PROJETO	SÉRIE	Nº DE PÁGINAS	Nº DE FOTOS	Nº DE MAPAS
LAC			~~22~~ 25		

TIPO

☐ RPQ ☒ PRE ☐ NTC ☐ PRP ☐ MAN ☐ PUD ☐ TAE ☐ ___

DIVULGAÇÃO

☒ EXTERNA ☐ INTERNA ☐ RESERVADA ☐ LISTA DE DISTRIBUIÇÃO ANEXA

PERIÓDICO/EVENTO

This paper was presented at EURO XI - European Congress on Operational Research, RWTH Aachen, July 16-19, 1991.

CONVÊNIO

AUTORIZAÇÃO PRELIMINAR

___/___/___ ASSINATURA

REVISÃO TÉCNICA

☐ SOLICITADA ☐ DISPENSADA ASSINATURA

RECEBIDA ___/___/___ DEVOLVIDA ___/___/___ ASSINATURA DO REVISOR

REVISÃO DE LINGUAGEM

☐ SOLICITADA ☐ DISPENSADA ASSINATURA

Nº ___

RECEBIDA ___/___/___ DEVOLVIDA ___/___/___ ASSINATURA DO REVISOR

PROCESSAMENTO/DATILOGRAFIA

RECEBIDA ___/___/___ DEVOLVIDA ___/___/___ ASSINATURA

REVISÃO TIPOGRÁFICA

RECEBIDA ___/___/___ DEVOLVIDA ___/___/___ ASSINATURA

AUTORIZAÇÃO FINAL

12/03/92

ASSINATURA Paulo Renato de Morais
Chefe
Centro de Tecnologias Associadas - CTE
Substituto

PALAVRAS-CHAVE

cutting stock problem; assortment problem; heuristics.

519.8