

# Pattern Reduction in Paper Cutting

CHRIS ALDRIDGE<sup>1</sup>, JON CHAPMAN<sup>2†</sup>, REBECCA GOWER<sup>2</sup>, ROBERT LEESE<sup>2</sup>,  
COLIN MCDIARMID<sup>1</sup>, MARK SHEPHERD<sup>1</sup>, HANS TUENTER<sup>3</sup>, HELEN WILSON<sup>4</sup>, and  
ALAN ZINOBER<sup>5</sup>

<sup>1</sup> *University of Oxford*

<sup>2</sup> *Smith Institute, Oxford*

<sup>3</sup> *University of Birmingham*

<sup>4</sup> *University of Cambridge*

<sup>5</sup> *University of Sheffield*

*(Communicated to MIIR on 23 June 2021)*

**Study Group:** ESGI 29. 18–22 March, 1996, University of Oxford

**Communicated by:** Tim Myers

**Industrial Partner:** Greycon

**Presenter:** Greycon

**Team Members:** Chris Aldridge, University of Oxford; Jon Chapman, University of Oxford; Rebecca Gower, Smith Institute, Oxford; Robert Leese, Smith Institute, Oxford; Colin McDiarmid, University of Oxford; Mark Shepherd, University of Oxford; Hans Tuenter, University of Birmingham; Helen Wilson, University of Cambridge; Alan Zinober, University of Sheffield

**Industrial Sector:** Manufacturing

**Tools:** Linear programming, Optimization

**Key Words:** Paper manufacturing, Optimization, Heuristics

**MSC2020 Codes:** 90

## Summary

A large part of the paper industry involves supplying customers with reels of specified width in specified quantities. These ‘customer reels’ must be cut from a set of wider ‘jumbo reels’ as economically as possible. The first priority is to satisfy the customer demands using as few jumbo reels as possible. This is an example of the well-known one-dimensional cutting stock problem, which can be solved by creating a series of patterns, each corresponding to a different set of reels that can be cut from a single jumbo. A secondary problem is to minimize the number of patterns needed in order to reduce frequency of cutting-knife resets. Existing methods account for 90% of cases. The problem is to provide methods for the remaining 10%. This is shown to be an NP-hard problem, and so efficient design heuristics are proposed and tested for reducing further the frequency of resets.

† Corresponding Author: [chapman@maths.ox.ac.uk](mailto:chapman@maths.ox.ac.uk)

# Pattern Reduction in Paper Cutting

Greycon

## 1 Background

A large part of the paper industry involves supplying customers with reels of specified width in specified quantities. These ‘customer reels’ must be cut from a set of wider ‘jumbo reels’, in as economical a way as possible. The first priority is to minimize the waste, *i.e.* to satisfy the customer demands using as few jumbo reels as possible. This is an example of the one-dimensional cutting stock problem, which has an extensive literature [1, 2, 3, 4]. Greycon have developed cutting stock algorithms which they include in their software packages. Mathematically, the problem would be stated

### CUTTING STOCK

**Input:** positive integer  $J$  (the jumbo width), distinct positive integers  $r_1, \dots, r_s$  (the various customer reel widths), and positive integers  $d_1, \dots, d_s$  (the quantity of each customer width that must be produced).

**Task:** use as few jumbos of width  $J$  as possible to satisfy the demand for  $d_\alpha$  customer reels of width  $r_\alpha$  (for each  $\alpha = 1, \dots, s$ ).

A solution to CUTTING STOCK consists of a series of ‘patterns’, each corresponding to a different (unordered) set of customer reels that may be cut from a single jumbo. The solution also specifies how many jumbos must be cut according to each pattern to satisfy all demands. Typically, the number of patterns in a solution are in the range 5–30, with occasional large problems of up to 80 patterns. Each pattern consists of 2–10 customer reels and is used for 1–15 jumbos.

Within the class of solutions having minimum waste, a secondary problem is to minimize the number of different patterns needed. The motivation here is to reduce the number of times that the knives have to be reset on the paper-cutting machine. The specification now becomes

## PATTERN

**Input:** positive integer  $J$ , distinct positive integers  $r_1, \dots, r_s$  and positive integers  $d_1, \dots, d_s$ .

**Task:** find a minimum-waste solution to the corresponding instance of CUTTING STOCK, which further minimizes the number of patterns used.

The problem studied here is the second part of PATTERN, *i.e.* we assume that a solution of minimum waste (or very nearly so) is given and the task is to refine it by reducing the number of patterns, without increasing the waste.

It is convenient to express a collection of patterns in the form of an array  $P$ . Each row (indexed by  $i$ , taking values  $1, \dots, n$ ) represents a single pattern, which we shall denote  $P_i$ ; each column (indexed by  $\alpha$ , taking values  $1, \dots, s$ ) corresponds to a customer reel width. The entry  $P_{i\alpha}$  specifies how many times customer width  $r_\alpha$  appears in pattern  $P_i$ . This array  $P$  is supplemented with a list of ‘multiplicities’,  $c_i$ , giving the number of times each pattern is used.

There are two types of reduction that Greycon already implement in their software. First, there are  $2 \rightarrow 1$  reductions, which replace two patterns  $P_i$  and  $P_j$  with a single one  $P_k$ . This is possible whenever  $P_k$  can be found such that

$$c_i P_{i\alpha} + c_j P_{j\alpha} = (c_i + c_j) P_{k\alpha} \quad \text{for all } \alpha.$$

A necessary and sufficient condition is

$$P_{i\alpha} \equiv P_{j\alpha} \pmod{\left(\frac{c_i + c_j}{\gcd(c_i, c_j)}\right)} \quad \text{for all } \alpha.$$

Secondly, there are the so-called  $3 \rightarrow 2$  ‘staircase’ reductions, which take three patterns, of the generic form

| $r_1$ | $r_2$ | $r_3$ | $r_4$ |   |
|-------|-------|-------|-------|---|
| 1     | 1     | 0     | 0     | c |
| 0     | 1     | 1     | 0     | d |
| 0     | 0     | 1     | 1     | c |

and reduce them to

| $r_1$ | $r_2$ | $r_3$ | $r_4$ |     |
|-------|-------|-------|-------|-----|
| 0     | 1     | 1     | 0     | c+d |
| 1     | 0     | 0     | 1     | c   |

Here the customer reel widths appear in the top row, and the pattern multiplicities in the rightmost column. The condition for three patterns to be the subject of a

$3 \rightarrow 2$  staircase is that one of them should be completely ‘covered’ by the other two, and these two must have equal multiplicity, *i.e.*  $P_{j\alpha} \leq P_{i\alpha} + P_{k\alpha}$  for all  $\alpha$ , and  $c_i = c_k$ . It must also in general be checked that the patterns reached by the reduction are feasible, in the sense that  $\sum_{\alpha} r_{\alpha} P_{i\alpha} \leq J$  for all  $i$ . (Note that feasibility is automatically guaranteed in  $2 \rightarrow 1$  reductions.)

Greycon claim that  $3 \rightarrow 2$  staircases account in practice for approximately 90% of all  $3 \rightarrow 2$  reductions. The other 10% are clear targets for new approaches. For example, the following reduction, reproduced from Allwood and Goulimis [5], is not a staircase:

| 1200 | 1150 | 1100 | 1000 | 900 | 800 |   |
|------|------|------|------|-----|-----|---|
| 0    | 2    | 1    | 0    | 0   | 1   | 2 |
| 1    | 0    | 1    | 1    | 0   | 1   | 2 |
| 0    | 0    | 0    | 2    | 2   | 0   | 1 |

↓

| 1200 | 1150 | 1100 | 1000 | 900 | 800 |   |
|------|------|------|------|-----|-----|---|
| 0    | 1    | 1    | 1    | 0   | 1   | 4 |
| 2    | 0    | 0    | 0    | 2   | 0   | 1 |

The remainder of the report presents the further contributions made by participants in the Study Group. Having shown that the pattern reduction problem is NP-hard (even when the cutting stock problem is easy), we go on to put forward proposals for designing efficient heuristics, for which the computational budget imposed by Greycon was 1 minute on a Pentium-based PC. Initial application to several test problems is also described.

## 2 Computational Complexity

Before looking at the issue of pattern reduction, we first remark that the problem CUTTING STOCK is NP-hard. To see this consider the problem

### 3-PARTITION

**Input:** a positive integer  $B$  and positive integers  $a_1, \dots, a_{3m}$  such that  $B/4 < a_i < B/2$  for each  $i = 1, \dots, 3m$ , and such that  $\sum_i a_i = mB$ .

**Question:** can the input be partitioned into  $m$  disjoint sets (which must necessarily be triples) such that the sum of the  $a_i$  in each set is equal to  $B$ ?

If the  $a_i$  are the various customer reel widths (appearing with multiplicities equal to the corresponding demands) and  $B$  is the jumbo width, then 3-PARTITION asks

whether there exists a solution of zero waste, and is therefore contained in CUTTING STOCK. It is known to be strongly NP-hard (see, for example Garey and Johnson [6], pp 96–100). Hence CUTTING STOCK is also NP-hard, and remains so even if  $J$  is bounded by a polynomial in the number of customer reels demanded.

However, there are instances of CUTTING STOCK that are easy to solve. In particular, suppose that all the customer reels satisfy  $r_\alpha > J/3$ , so that no more than two customer reels can be cut from a single jumbo. A minimum-waste solution is generated by the ‘First-Fit-Decreasing’ rule. The first jumbo supplies a reel with the largest width, together with a second reel if possible, of the largest width that will fit; subsequent jumbos are cut in the same way, looking at the remaining demand.

Turning now to pattern reduction, each instance of PATTERN is clearly at least as hard as the corresponding CUTTING STOCK instance, and so is generally NP-hard. What is not so obvious is that PATTERN may remain hard even when CUTTING STOCK is easy. Let 2-PATTERN be the same as PATTERN, but with the restriction  $r_\alpha > J/3$  for all  $\alpha$ , so that the waste-minimization is easy.

### Theorem

*2-PATTERN is strongly NP-hard.*

### Proof

The proof involves constructing a reduction from 3-PARTITION to 2-PATTERN. Given an instance of 3-PARTITION, construct an instance of 2-PATTERN with  $4m$  different customer widths. Let  $J = 24m$ , with  $r_i = 8m + i$  for  $i = 1, \dots, 3m$  and  $r_{3m+i} = 12m + i$  for  $i = 1, \dots, m$ . Thus there are  $3m$  ‘small’ widths, with  $J/3 < r_i < J/2$ , and  $m$  ‘large’ widths, with  $r_i > J/2$ . Moreover, any combination of a small width and a large width fits in a single jumbo. Let the demands be  $d_i = a_i$  for  $i = 1, \dots, 3m$  (i.e. taken from the input to 3-PARTITION), and  $d_i = B$  for  $i = 3m + 1, \dots, 4m$ . The total number of customer reels required is therefore  $2mB$ .

In this instance of 2-PATTERN, the minimum number of jumbos is  $mB$ , achieved if and only if each pattern contains a large width and a small width. The number of patterns in any such solution is at least  $3m$ , since there must be at least one pattern for each small width. A minimum-waste solution with  $3m$  patterns exists if and only if the corresponding instance of 3-PARTITION has a solution, and we have already seen that this question is strongly NP-hard.  $\square$

## 3 Systematic Searching for Reductions

The 2-PATTERN problem discussed above shows that even when all the patterns that can appear in minimum-waste solutions are known, minimizing the number of patterns can be nontrivial. (Note that the theorem shows that the problem is NP-hard even when small integers are involved, in contrast to, say, the knapsack

problem.) However, it is possible to systematically search for particular types of reduction, of which the  $2 \rightarrow 1$  and  $3 \rightarrow 2$  staircase are examples.

With a little more effort, one can pick out all  $3 \rightarrow 2$  reductions. Suppose that there is a set of  $p$  ‘available’ patterns. (Ideally, this set will include all patterns that appear in some minimum-waste solution, but in any case should represent the most economical ways of cutting a single jumbo.) Provided the customer reel widths  $r_\alpha$  are all bounded below by a fixed fraction of the jumbo width  $J$ ,  $p$  is only polynomially large in the number of reel widths  $s$ . (None of the examples supplied by Greycon had any  $r_\alpha$  less than  $J/16$ .) Now suppose that one has a (minimum-waste) solution using  $n$  patterns  $P_1, \dots, P_n$ . Run through all triples  $(P_i, P_j, P_k)$ , where  $i, j, k = 1, \dots, n$  with  $i < j < k$  and calculate the contribution  $f_\alpha$  that they make to the total demand for customer width  $r_\alpha$ , namely

$$f_\alpha = c_i P_{i\alpha} + c_j P_{j\alpha} + c_k P_{k\alpha} \quad (\alpha = 1, \dots, s). \quad (1)$$

If any triple can be the subject of a  $3 \rightarrow 2$  reduction, there are patterns  $P_a$  and  $P_b$ , picked from the original set of  $p$  patterns, and an integer  $x$  satisfying  $0 < x \leq t$ , where  $t = c_i + c_j + c_k$ , such that

$$f_\alpha = x P_{a\alpha} + (t - x) P_{b\alpha} \quad (\alpha = 1, \dots, s). \quad (2)$$

There are  $O(n^3)$  triples in the starting solution and  $O(p^2)$  possibilities for  $(P_a, P_b)$ . However, the  $3 \rightarrow 2$  reductions can be identified by examining fewer than  $O(n^3 p^2)$  cases: rewrite (2) as

$$P_{a\alpha} = P_{b\alpha} + (f_\alpha - t P_{b\alpha})/x, \quad (3)$$

and then for each choice of  $P_b$  and  $x$ , calculate the right hand side of (3), checking whether it corresponds to a valid pattern. In other words, one has to examine only  $O(n^3 p t)$  cases to compile all  $3 \rightarrow 2$  reductions. Typically  $p \sim 10^3$ , while  $n, t \sim 10^1$  and so this difference is significant.

The same idea can be applied to higher reductions. All  $m_1 \rightarrow m_2$  reductions could be found by examining  $O(n^{m_1} p^{m_2-1} t^{m_2-1})$  cases. However, for  $m_2 > 2$ , exhaustive searches of this sort are probably impractical in real problems.

## 4 A Possible Algorithm

Having studied small examples of the various types of pattern reduction that might be important, it is possible to set about designing an algorithm that can be applied to much larger problems. It should be capable of catching  $2 \rightarrow 1$  and  $3 \rightarrow 2$  staircase reductions, but also more general possibilities. In addition, it should respect the computing budget of 1 minute of Pentium time for typical real problems. This section describes an initial implementation of one such algorithm, and also identifies ways in which it might be further refined.

## Overview

The operation of the algorithm is summarized in the following remarks:

- (a) The algorithm consists of a series of ‘rounds’. Each round selects a pair of patterns, say  $P_i$  and  $P_j$ , and makes a series of ‘moves’ in an attempt to make them coincide.
- (b) Each move involves a single interchange of customer reels between two of the patterns in the current solution, and has the effect of bringing the patterns  $P_i$  and  $P_j$  closer together by reducing the metric  $\delta(P_i, P_j) = \sum_{\alpha} |P_{i\alpha} - P_{j\alpha}|$ . There are two distinct types of move:
  - (I) The patterns involved in the interchange are  $P_i$  and  $P_j$  themselves;
  - (II) The patterns involved in the interchange are one of  $P_i$  and  $P_j$ , and a third, ‘catalyst’ pattern  $P_k$ .
- (c) Each round finishes when no further moves reducing  $\delta(P_i, P_j)$  are possible. If  $\delta(P_i, P_j)$  has been reduced to zero then a pattern reduction has been found. Whether or not this has been achieved, a new pair  $(P_i, P_j)$  are chosen for the next round.

## Remarks

- (1) The algorithm preserves all row and column sums in the table of patterns, and so  $P_i$  and  $P_j$  must be chosen to have the same row sum, *i.e.* the same number of customer reels, otherwise there is no possibility of making them the same.
- (2) The patterns involved in each interchange of reels should have the same multiplicity, to preserve the quantity of each customer width produced. Hence if  $c_i \neq c_j$  then only type II moves are available.
- (3) Type II moves are made only when no type I move is available.
- (4) Checks are needed to ensure that interchanges do not produce infeasible patterns, by substituting a large reel for a small one in a jumbo roll that is already full or nearly full.
- (5) If patterns  $P_i$  and  $P_j$  (of the same multiplicity) can take part in a  $2 \rightarrow 1$  reduction, this can be achieved by a sequence of type I moves.
- (6) A  $3 \rightarrow 2$  staircase reduction corresponds to a series of type II moves, in which the two patterns with repeated multiplicity take part in the interchange, one of them playing the catalyst to make the other two match.

## A small example

The following example shows how type I and type II moves can work together to effect a pattern reduction that would not be found by a  $2 \rightarrow 1$  or  $3 \rightarrow 2$  staircase procedure. It is also not difficult to check that no further reduction is possible.

| $r_1$ | $r_2$ | $r_3$ | $r_4$ |   |               | $r_1$ | $r_2$ | $r_3$ | $r_4$ |   |               | $r_1$ | $r_2$ | $r_3$ | $r_4$ |   |  |
|-------|-------|-------|-------|---|---------------|-------|-------|-------|-------|---|---------------|-------|-------|-------|-------|---|--|
| 3     | 2     | 0     | 0     | 1 |               | 2     | 2     | 1     | 0     | 1 |               | 2     | 2     | 1     | 0     | 1 |  |
| 0     | 3     | 2     | 0     | 1 | $\rightarrow$ | 1     | 3     | 1     | 0     | 1 | $\rightarrow$ | 2     | 2     | 1     | 0     | 1 |  |
| 0     | 0     | 3     | 2     | 1 |               | 0     | 0     | 3     | 2     | 1 |               | 0     | 0     | 3     | 2     | 1 |  |
| 2     | 0     | 0     | 3     | 1 |               | 2     | 0     | 0     | 3     | 1 |               | 1     | 1     | 0     | 3     | 1 |  |

It is assumed in this example that the customer widths are sufficiently close that no infeasible patterns will be encountered (for example, taking  $r_\alpha \leq J/5$  for all  $\alpha$  will do). Initially, all pairs of patterns are equally unmatched, so  $(i, j) = (1, 2)$  is chosen arbitrarily. The first move is type I, swapping reels of widths  $r_1$  and  $r_3$  between the first and second patterns. No further type I moves are then available. The second move is type II, with pattern  $P_4$  used as a catalyst. The interchange of reels occurs between patterns  $P_2$  and  $P_4$ , involving widths  $r_1$  and  $r_2$ .

## Choices of implementation

Within the general framework outlined above there are various ways in which the details may be implemented. Some of the choices to be made are the following:

- (1) Subject to the constraint that  $P_i$  and  $P_j$  have the same row sums in each round, their choice is arbitrary. However, it seems reasonable to choose them to be already as closely matched as possible, since, intuitively at least, fewer moves will then be required to reach a reduction. The initial implementation calculates the Hamming distance, *i.e.* the number of customer widths  $r_\alpha$  for which  $P_{i\alpha} \neq P_{j\alpha}$ , for each pair  $(i, j)$ . It then chooses the first pair in the lexicographic ordering of all pairs for which the minimum distance occurs. (The lexicographic ordering is taken to mean  $(1, 2), (1, 3), \dots, (1, n), (2, 3), (2, 4), \dots$ , where  $n$  is the current number of patterns.)
- (2) If a round does not end in a pattern reduction then to avoid being caught in a loop, the same  $(P_i, P_j)$  pair must not be chosen in the next round. This could be achieved by a tabu list (implemented in the initial version), or by some random choice among the possible candidates.
- (3) There is often freedom in choosing the reel widths involved in an interchange. The input data generally lists the items in increasing or decreasing order of



size, and the initial implementation simply takes the first possibility that does not generate an infeasible pattern.

- (4) There is no need to check for feasible patterns at each move, as initially implemented. An alternative would be to check only when each reduction is made, returning to the most recent feasible solution if the feasibility test fails. Allowing the collection of patterns to become temporarily infeasible might open up new possible reductions.

## Possible refinements

The main shortcoming of the algorithm appears to be that it does not allow moves between patterns of different multiplicity. As a result, it would miss some  $2 \rightarrow 1$  reductions (although these could easily be detected separately) and also examples such as the non-staircase  $3 \rightarrow 2$  reduction given in Section 1. On the positive side, it will catch all  $3 \rightarrow 2$  staircases, and also many  $n \rightarrow n - 1$  reductions with  $n > 3$ .

However, there is no reason why the method should not be extended to allow more moves. For example, the non-staircase in Section 1 can be effected by

| 1200 | 1150 | 1100 | 1000 | 900 | 800 |   |
|------|------|------|------|-----|-----|---|
| 0    | 2    | 1    | 0    | 0   | 1   | 2 |
| 1    | 0    | 1    | 1    | 0   | 1   | 2 |
| 0    | 0    | 0    | 2    | 2   | 0   | 1 |

↓

| 1200 | 1150 | 1100 | 1000 | 900 | 800 |   |
|------|------|------|------|-----|-----|---|
| 0    | 1    | 1    | 1    | 0   | 1   | 2 |
| 1    | 1    | 1    | 0    | 0   | 1   | 2 |
| 0    | 0    | 0    | 2    | 2   | 0   | 1 |

↓

| 1200 | 1150 | 1100 | 1000 | 900 | 800 |   |
|------|------|------|------|-----|-----|---|
| 0    | 1    | 1    | 1    | 0   | 1   | 2 |
| 0    | 1    | 1    | 1    | 0   | 1   | 2 |
| 2    | 0    | 0    | 0    | 2   | 0   | 1 |

Here the first move is performed by the current algorithm, but the second involves an interchange (using widths 1200 and 1000) between patterns  $P_2$  and  $P_3$ , which have different multiplicities. When making such moves, the relative multiplicities must be taken into account, in order to preserve the level of production of each customer width.

Finally, initial experimentation indicates that steps are needed to prevent the algorithm entering a repeating cycle of moves that does not lead to a reduction. The algorithm can be terminated by putting a very small limit on the maximum length of the tabu list, but this is not satisfactory, since it sometimes leads to reductions being missed.

## 5 Test Problems

Greycon provided several real problems, which have been used to make a preliminary evaluation of the effectiveness of the proposed algorithm. The input to each problem was a set of patterns generated by a waste-minimization algorithm accompanied by preliminary pattern reduction, although we found that not all  $2 \rightarrow 1$  and  $3 \rightarrow 2$  staircase reductions had been made.

Some samples are given below, giving just the input and final output, together with the number of moves used. Some of the lengths have been rescaled to reduce the numerical values of the  $r_\alpha$  and  $J$ . Execution times were well within the allocated computing budget, with typical problems requiring only a few seconds on a modest PC.

The scope for pattern reduction seems sensitive to the waste percentage of the input. If the waste percentage is small, say below 1%, then there are generally few feasible moves available, but if the percentage approaches 5% then the possible pattern reductions can be quite dramatic.

### Problem 1

| 170 | 180 | 200 | 220 | 250 | 260 | 266 |   |
|-----|-----|-----|-----|-----|-----|-----|---|
| 0   | 0   | 0   | 0   | 0   | 0   | 3   | 1 |
| 0   | 0   | 0   | 0   | 2   | 1   | 0   | 1 |
| 0   | 2   | 0   | 1   | 0   | 1   | 0   | 1 |
| 1   | 0   | 1   | 1   | 0   | 0   | 1   | 1 |
| 1   | 0   | 1   | 0   | 2   | 0   | 0   | 1 |

↓

| 170 | 180 | 200 | 220 | 250 | 260 | 266 |   |
|-----|-----|-----|-----|-----|-----|-----|---|
| 0   | 0   | 0   | 0   | 1   | 1   | 1   | 2 |
| 0   | 1   | 1   | 1   | 0   | 0   | 1   | 2 |
| 2   | 0   | 0   | 0   | 2   | 0   | 0   | 1 |

Jumbo size 870, 5.2% waste, 4 moves used

## Problem 2

| 850 | 900 | 1000 | 1100 | 1250 | 1270 | 1300 | 1330 | 1340 | 1345 |   |
|-----|-----|------|------|------|------|------|------|------|------|---|
| 0   | 0   | 0    | 0    | 0    | 0    | 1    | 2    | 0    | 0    | 2 |
| 0   | 0   | 0    | 0    | 0    | 0    | 0    | 3    | 0    | 0    | 1 |
| 0   | 0   | 0    | 0    | 2    | 0    | 1    | 0    | 0    | 0    | 1 |
| 0   | 0   | 0    | 0    | 0    | 1    | 0    | 0    | 1    | 1    | 1 |
| 1   | 0   | 1    | 0    | 2    | 0    | 0    | 0    | 0    | 0    | 1 |
| 0   | 0   | 2    | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 1 |
| 0   | 2   | 0    | 0    | 2    | 0    | 0    | 0    | 0    | 0    | 1 |
| 2   | 0   | 0    | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 1 |
| 1   | 0   | 1    | 1    | 0    | 0    | 0    | 1    | 0    | 0    | 1 |
| 0   | 2   | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 0    | 1 |

↓

| 850 | 900 | 1000 | 1100 | 1250 | 1270 | 1300 | 1330 | 1340 | 1345 |   |
|-----|-----|------|------|------|------|------|------|------|------|---|
| 0   | 0   | 0    | 0    | 1    | 0    | 1    | 1    | 0    | 0    | 4 |
| 0   | 0   | 0    | 0    | 0    | 1    | 0    | 0    | 1    | 1    | 1 |
| 1   | 0   | 1    | 1    | 0    | 0    | 0    | 1    | 0    | 0    | 4 |
| 0   | 2   | 0    | 0    | 2    | 0    | 0    | 0    | 0    | 0    | 2 |

Jumbo size 4350, 5.5% waste, 5 moves used

## Problem 3

This example is a case where no reductions were found. Lengths have been rescaled by a factor of 50, giving  $J = 85$ .

| 16 | 17 | 18 | 20 | 21 | 24 | 27 | 28 | 30 | 31 | 36 | 38 | 41 | 43 | 45 |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 3  |
| 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 4  |
| 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  |
| 0  | 0  | 2  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  |
| 0  | 0  | 0  | 0  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 2  |
| 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 1  |
| 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 3  |
| 1  | 0  | 0  | 0  | 1  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  |
| 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 10 |
| 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 11 |
| 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 0  | 2  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 5  |
| 0  | 0  | 0  | 0  | 4  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  |
| 0  | 0  | 0  | 0  | 0  | 2  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 7  |
| 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 9  |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  |
| 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 3  |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 4  |
| 0  | 0  | 0  | 0  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 2  |

Note that 12 of the 19 patterns completely fill the jumbo. Of the remainder, 5 patterns have length 84 and 2 have length 83. Overall, there is less than 0.2% waste.

## Problem 4

This was the largest problem, involving 91 customer widths, ranging from 1500 to 2490. The jumbo width was 4340 (although it could be taken to be 4300 without changing anything that follows), and so it is an example in which the first-fit-decreasing rule generates a minimum-waste solution. The initial solution supplied by Greycon is given below by listing the two widths in each pattern, in preference to the large, sparse matrix  $P$ . Overall, the waste is 4.1%.

| $c_i$ | Pattern |      | $c_i$ | Pattern |      | $c_i$ | Pattern |      |
|-------|---------|------|-------|---------|------|-------|---------|------|
| 4     | 2480    | 1820 | 4     | 2460    | 1840 | 16    | 2450    | 1850 |
| 4     | 2430    | 1870 | 16    | 2420    | 1880 | 4     | 2410    | 1890 |
| 8     | 2400    | 1900 | 4     | 2380    | 1920 | 6     | 2300    | 2000 |
| 4     | 2260    | 2040 | 5     | 2250    | 2050 | 10    | 2490    | 1800 |
| 2     | 2480    | 1810 | 9     | 2380    | 1900 | 4     | 2340    | 1940 |
| 9     | 2180    | 2100 | 2     | 2460    | 1810 | 4     | 2290    | 1980 |
| 2     | 2190    | 2080 | 4     | 2480    | 1720 | 16    | 2330    | 1930 |
| 4     | 2300    | 1960 | * 6   | 2130    | 2130 | 4     | 2320    | 1670 |
| 4     | 2310    | 1970 | * 4   | 2270    | 1980 | 11    | 2250    | 2000 |
| 4     | 2240    | 2010 | 2     | 2280    | 2020 | 4     | 2220    | 2030 |
| 1     | 2200    | 2050 | 16    | 2450    | 1780 | 11    | 2400    | 1830 |
| * 10  | 2150    | 2080 | 5     | 2390    | 1830 | 2     | 2190    | 2020 |
| 7     | 2180    | 2030 | 4     | 2460    | 1790 | 4     | 2170    | 2030 |
| * 6   | 2150    | 2050 | 4     | 2140    | 2060 | * 4   | 2130    | 2070 |
| 4     | 2120    | 2080 | 4     | 2110    | 2090 | * 4   | 2100    | 2100 |
| 10    | 2490    | 1700 | 4     | 2210    | 1980 | 4     | 2420    | 1760 |
| 16    | 2470    | 1680 | 16    | 2200    | 1950 | 4     | 2160    | 1990 |
| 4     | 2480    | 1660 | 4     | 2450    | 1690 | 6     | 2440    | 1700 |
| 5     | 2480    | 1650 | 4     | 2370    | 1610 | 4     | 2360    | 1740 |
| 16    | 2350    | 1750 | 7     | 2300    | 1800 | 17    | 2480    | 1600 |
| 4     | 2440    | 1640 | 1     | 2380    | 1700 | 16    | 2430    | 1630 |
| 4     | 2430    | 1620 | 11    | 2400    | 1650 | * 4   | 2050    | 1980 |
| 4     | 2470    | 1520 | 6     | 2410    | 1580 | 16    | 2430    | 1530 |
| 16    | 2230    | 1730 | 16    | 2400    | 1500 | 14    | 2280    | 1590 |
| 1     | 2030    | 1500 |       |         |      |       |         |      |

The algorithm reduced the number of patterns from 73 to 70, using 5 moves and leaving most of the original input unchanged. (Greycon subsequently revealed that a solution using 61 patterns has been found by A. Diegel; this would provide a useful target for any future algorithm refinements.) The seven patterns marked with an asterisk are those involved in some move of the reduction. Hence, in the matrix notation, the reduction may be condensed down to

| 1980 | 2050 | 2070 | 2080 | 2100 | 2130 | 2150 | 2270 |    |
|------|------|------|------|------|------|------|------|----|
| 0    | 0    | 0    | 0    | 0    | 2    | 0    | 0    | 6  |
| 1    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 4  |
| 0    | 0    | 0    | 1    | 0    | 0    | 1    | 0    | 10 |
| 0    | 1    | 0    | 0    | 0    | 0    | 1    | 0    | 6  |
| 0    | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 4  |
| 0    | 0    | 0    | 0    | 2    | 0    | 0    | 0    | 4  |
| 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 4  |

↓

| 1980 | 2050 | 2070 | 2080 | 2100 | 2130 | 2150 | 2270 |    |
|------|------|------|------|------|------|------|------|----|
| 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    | 16 |
| 1    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 8  |
| 0    | 1    | 0    | 1    | 0    | 0    | 0    | 0    | 10 |
| 0    | 0    | 1    | 0    | 0    | 0    | 0    | 1    | 4  |

## 6 Summary

Greycon's initial presentation to the Study Group posed several questions, which are listed below, along with (partial) answers arising from the work described above.

- (1) **Given a minimum-waste solution, what is the minimum number of patterns required?**

It has been shown in Section 2 that even when all the patterns appearing in minimum-waste solutions are known, determining the minimum number of patterns may be hard. It seems unlikely that one can guarantee to find the minimum number of patterns for large classes of realistic problems with only a few seconds on a PC available.

- (2) **Given an  $n \rightarrow n - 1$  algorithm, will it find an optimal solution to the minimum-pattern problem?**

There are problems for which  $n \rightarrow n - 1$  reductions are not possible although a more dramatic reduction is. For example, suppose there are three customer widths, all slightly smaller than  $J/3$ . Then

| $r_1$ | $r_2$ | $r_3$ |   |
|-------|-------|-------|---|
| 2     | 0     | 1     | 1 |
| 1     | 2     | 0     | 1 |
| 0     | 1     | 2     | 1 |

may be reduced to

| $r_1$ | $r_2$ | $r_3$ |   |
|-------|-------|-------|---|
| 1     | 1     | 1     | 3 |

but there is no set of two distinct patterns satisfying the same demands.

**(3) Is there an efficient  $n \rightarrow n - 1$  algorithm?**

In light of Question 2, Question 3 should perhaps be rephrased as ‘Is there an efficient algorithm to reduce  $n$  patterns?’ However, if an algorithm guaranteed to find some reduction whenever one existed then it could be applied iteratively to minimize the number of patterns, and we have seen this cannot be done easily.

**(4) Are there efficient  $5 \rightarrow 4$  and  $4 \rightarrow 3$  algorithms?**

**(5) Is it worthwhile seeking alternatives to greedy heuristics?**

In response to Questions 4 and 5, we point to the algorithm described earlier, or variants of it. Such approaches seem capable of catching many higher reductions.

**(6) Is there a way to find solutions with the smallest possible number of single patterns?**

The Study Group did not investigate methods tailored specifically to this task, but the algorithm proposed here seems to do reasonably well. It will not increase the number of singleton patterns under any circumstances, and when the number of singletons is high there will be many possible moves that tend to eliminate them.

**(7) Can a solution be found which reduces the number of knife changes?**

The algorithm will help to reduce the number of necessary knife changes because it works by bringing patterns closer together, even if this does not proceed fully to a pattern reduction. If two patterns are equal across some of the customer widths, the knives for these reels need not be changed when moving from one to the other.

## Contributors

|                |                 |               |
|----------------|-----------------|---------------|
| Chris Aldridge | Robert Leese    | Hans Tuentner |
| Jon Chapman    | Colin McDiarmid | Helen Wilson  |
| Rebecca Gower  | Mark Shepherd   | Alan Zinober  |

## Notes added

- (a) In the week following the Study Group, a paper [7] was presented at the CO96 conference by a group from the University of Lisbon, which addressed some of the issues discussed here.

- (b) The theorem in section 2 has been extended to the case  $J/3 < r_\alpha \leq J/2$  for each  $\alpha$ , so that no three customer reels fit on a single jumbo, but any two do. The proof is considerably more complicated.

## References

- [1] P. C. Gilmore and R. E. Gomory, *A linear programming approach to the cutting-stock problem*, Operations Research **9** (1961) 849–859; *ibid.* **11** (1963) 863–888.
- [2] V. Chvátal, *Linear Programming*, Freeman (1983), pp. 195–212.
- [3] C. N. Goulimis, *Optimal solutions for the cutting stock problem*, European Journal of Operations Research **44** (1990) 197–208.
- [4] P. E. Sweeney and E. R. Paternoster, *Cutting and packing problems: a categorized, application-orientated research bibliography*, Journal of the Operational Research Society **43** (1992) 691–706.
- [5] J. M. Allwood and C. N. Goulimis, *Reducing the number of patterns in the 1-dimensional cutting stock problem*, Imperial College report (1988).
- [6] M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman (1979).
- [7] A. Respicio, M. E. Captivo and J. Paixao, *Scheduling of cutting orders in paper rolls manufacturing*, presented at the Symposium on Combinatorial Optimization (CO96), Imperial College, March 1996.