

Two-pattern combining in the Reel Deckling Problem

Adolf Diegel¹, Olaf Diegel², Edouard Montocchio³, Sias van Schalkwyk³, George Pritchard⁴

ABSTRACT

The one-dimensional cutting stock or deckling problem arises in making narrow reels from wide rolls (paper) or short pieces from long ones (pipes). Key objectives are to fill demand with the fewest possible integer stock units and setups. Setups matter because changing knives for a new pattern may be costly for any material, the more so for paper: it may not rewind smoothly. One will also be

concerned about short runs, order splits, knife changes and surplus vs waste or broke.

This paper focuses on reducing setups by simple means. It presents and explores methods of finding, from among the patterns in a given cutting plan, those two which can be replaced by a single one. A companion paper aims to reduce three or more patterns to a lower number.

Keywords: *Cutting stock, Deckling, Combinatorial analysis, Heuristics*

INTRODUCTION

Terms for our topic come from the paper industry. The task is to deckle (unwind, slit and rewind) a master or stock roll, a “jumbo”, into small units. On a modern automatic machine, this may be done seamlessly, but for a manual change one must stop the winder, reset knives, then feed the band of paper in anew. The band being quite wide, it may foul, so a change in knives for a new setup tends to be costly.

The Linear Programming formulation, “LP”, of this problem aims to find optimal cutting patterns by one or the other of two methods. The initial one was to generate all possible patterns, or at any rate a fair number, then to select those best filling demand. Work on this approach began in the 1950s, the possibly earliest paper being Kantorovich (1962 reprint).

The other method is to generate a dynamic pattern, one at a time, study its effect, then add only a good one to the current table (Gilmore and Gomory 1961 p. 849, 1963, 1964, Wagner 1970, Kemeny 1972, Allwood and

Goulimis 1988, Goulimis 1995, Diegel 1987 to 2012).

This second approach was called “column generation” by early authors, as they arranged their cutting patterns in columns rather than in rows, see below. We prefer demand in columns and patterns in rows so we can print two tables side by side to study the transition from one to the other.

For now, critical is whether we use **static**, anterior, even exhaustive pattern generation, or whether it is **dynamic**, posterior or selective: are patterns generated before their analysis or else, are they added, one at a time, until all orders are filled?

Gau and Wäscher (1995) consider dynamic pattern generation as the “*most prominent approach to the 1D-CSP*”, able to solve quite large problems in minutes, if not seconds. However, Diegel (1996) shows that the dynamic approach may retain as many patterns as there are orders, one for each reel width: in principle, this means a high rather than a low number of setups. Nor does the dynamic method focus on setups: by its very nature, it aims to minimise stock usage rather than setups.

¹Cutline Research, 66 Edmonds Road, Durban 4001

²Massey University, Auckland, NZ, o.diegel@massey.ac.nz, www.massey.ac.nz

³Mondi Business Paper South Africa, Durban and Richards Bay ed.montocchio@mondigroup.co.za, sias.vanschalkwyk@mondigroup.co.za

⁴Mpact Limited, gpritchard@mpact.co.za

Table 1. Column-based static Linear Programming formulation

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	>	0	stock	
5	4	4	3	3	3	2	2	2	2	1	1	1	1	1	0	0	0	0	0	0	>	90	of 700	
0	1	0	2	1	0	3	2	1	0	4	3	2	1	0	6	5	4	3	2	1	>	85	of 600	
0	0	1	0	1	3	0	2	3	4	1	2	3	4	5	0	1	2	3	4	6	7	>	90	of 500
100	200	300	300	400	0	400	0	100	200	0	100	200	300	400	0	100	200	300	400	0	100	>	0	waste

Table 2. Rows and columns interchanged, classic format and ours

Stock	700	600	500	Loss	0	45	Loss	700	600	500	3600
.00	90	85	90	0	-106	.00	-90	-85	-90	-45	
1	3	0	3	0	-206	.00	3.00	.00	3.00	1.00	
1	2	2	2	0	-306	.00	2.00	2.00	2.00	1.00	
1	1	4	1	0	-406	.00	1.00	4.00	1.00	1.00	
1	0	6	0	0	-507	.00	.00	6.00	.00	1.00	
1	0	1	6	0			.00	1.00	6.00	1.00	

The static approach, on the other hand, can aim to minimise stock usage and setups. Yet choosing patterns for minimal stock usage entails the same rules as the dynamic method, hence easily as many setups as orders. Rather, to succeed in minimising setups, one must generate all possible patterns; then, since setup-friendly ones are neither known nor found intuitively, each pattern requires a conditional constraint, to allow it to run or not. Diegel (1996) and Foerster (1998, p. 79) explain why such constraints quadruple the space used for the patterns themselves.

So, if it is hardly practical to store all possible patterns for genuine data, it is even less so for tables four times as large.

The static approach may be impractical for large sets of data. Yet for those which can be handled, it may not only reduce setups, but also show that it may be as few as one half if not one third of orders. Seeing a number that low, yet realising that one can hardly enforce it formally, one may reasonably ask whether setups can be reduced otherwise, after the fact, and by fairly simple and rapid methods. Once a cutting plan has been found in some way, even manually or with fractional runs, can we identify patterns whose elements combine to reduce setups?

INITIAL DATA FOR PATTERN COMBINING

Johnston (1984, 1986) defined a rule for finding, among active patterns, those two which can be combined into one:

“Testing ... is very simple as it can be (trivially) shown that two batches having size ratio of p:q (p, q integer) can be strictly combined only if the two patterns differ, element by element, by 0 mod (p + q).”

Johnston does not (trivially) support this rule by a case study, so we propose an approach similar in at least one respect. It eventually leads to 3-combining (three patterns into two or one), even more.

We begin with a small example, “cut 90 reels of widths 700 and 500 mm as well as 85 600s from 3600 mm stock”. We chose 90 reels rather than 100 or more to have narrow columns, below, and fewer 600s to get fractional vs integer solutions, the latter usually with surplus.

Table 1 presents this seemingly simple case for the classic or static approach, with all possible patterns in one table.

The objective 1, 1, 1, ... in the first row means that each pattern, if utilised, takes 1 stock unit or jumbo for each run. This row appears on top of the table, as the main objective: minimise stock usage while yet filling demand, at right. Pattern loss, waste or broke are shown at the bottom of the table: it is merely recorded. If one were to exchange that row with the top one, broke would be minimal, but stock usage could be higher than necessary: low-loss patterns may still entail many rolls.

Cutting patterns appear in successive columns of the classic LP-table or “Tableau”. A pattern holds the number of reels it yields. Thus the first column/pattern has 5 reels of 700 mm and no others. 700*5 uses 3500 mm in 3600 with 100 mm broke in the last row. Stock width must be known to generate patterns, but, surprisingly, it is not in the classic Tableau.

Our example has only 3 orders and we could print all possible patterns across a page. Yet even so we had to compress their format and we cannot expect to print a larger set horizontally. This changes with patterns in rows and demand in columns, Table 2.

Patterns-in-rows allows printing, on one page, about 10 orders, enough to cover instructive examples. And for smaller sets we can show two tables side by side to compare them. Also, the number of pattern rows could reach about 50 on an A4-page.

Table 2 at left is an authentic transfer of the 0-loss patterns in Table 1, row-to-column. It could well serve a manual solution, for example, if one clicks on 3 in the first row to use that pattern to produce 700s and 500s by 90/3 or 30 runs. Yet the classic tableau did not include stock width nor minimal stock usage: $700*90 + 600*85 + 500*90 = 159,000$ mm; dividing by 3600 makes 45 rolls, integer round-up. This is -45 at right, "stock as yet unused", just as negative demand is "as yet unfilled".

Stock usage dominates our approach. If it finally remains negative, it becomes surplus; else new patterns may reduce it. Thus we put stock in the last column, just like another demand. (Early authors did not seem to improve a plan beyond filling orders or, if they did, they did not publish their approach, see Diegel 2012b, 2012c.)

With stock at right, we move pattern loss to the left, column 2, and add pattern labels. These include consecutive numbers*100, with the last two digits as "n-across", the number of reels in that pattern (which may be limited by available knives or stacks, Diegel 2011b, 2012d).

The consistency of this layout is confirmed by initial output, Table 3.

Table 3. Initial Tableau and effect of first manual pivot

Tableau 0	700	600	500	3600	Tableau 1	-106	600	500	3600
.00	-90.00	-85.00	-90.00	-45.00	.00	30.00	-85.00	.00	-15.00
-106	.00	3.00	.00	3.00	700	.00	.33	.00	1.00
-206	.00	2.00	2.00	2.00	-206	.00	-.67	2.00	.00
-306	.00	1.00	4.00	1.00	-306	.00	-.33	4.00	.00
-406	.00	.00	6.00	.00	-406	.00	.00	6.00	.00
-507	.00	.00	1.00	6.00	-507	.00	.00	1.00	6.00

(Continued on page 22)

Table 4. First plan filling demand, but with a fractional run

Tableau 2	-106	-406	500	3600	Cutting plan				
.00	30.00	14.17	.00	-.83	Width mm	700	600	500	3600
700	.00	.33	.00	1.00	Demand n	90	85	90	Loss
600	.00	.00	.17	.00	-106)	30.00	3	0	3
-206	.00	-.67	-.33	.00	-406)	14.17	0	6	0
-306	.00	-.33	-.67	.00	Sum	44.17	90	85	90
-507	.00	.00	-.17	6.00	pattern yet to be used, below				0

Table 5. Initial integer plan with 3 setups, 0 loss, but single run

Tableau 3	-106	-406	500	-507	Cutting plan				
.00	30.00	14.00	6.00	1.00	Width mm	700	600	500	3600
3600	.00	.00	-.20	7.20	Demand n	90	85	90	Loss
700	.00	.33	.07	-1.40	-106)	30	3	0	3
600	.00	.00	.20	-1.20	-406)	14	0	6	0
-206	.00	-.67	-.33	.00	-507)	1	0	1	6
-306	.00	-.33	-.67	.00	Sum	45	90	85	96+

If one were to pivot intuitively, one may well start with pattern -106 to cover 700s and 500s with 30 integer runs. Then the remaining unfilled order is 85*600; again the intuitive choice is -406, to fill that order without any loss and without disturbing the others, Table 4.

Of course, an integer run of 30*(3 0 3) yields 90 700s & 500s, exactly. Also, we get 85 600s by running -406 14.1667 times (if this is practical and if we can discard .83 of a roll partially used).

If not, there are surprisingly many alternative solutions. As -507 is already in the Tableau, clearly it can use idle stock for .83/.83 or 1 run: this yields 6 surplus 500s without any loss, while -406 changes to 14 runs, Table 5.

A single run is hardly practical, but we show this plan for two reasons: firstly, 3 orders may well imply 3 patterns; secondly, it will not be reduced to fewer patterns by the procedure explained below.

So what is the immediate answer? The most obvious one is, in Tableau 2, to replace idle stock by .83/.17 or 5 surplus 600s (instead of 6 500s), then -106 or -406 by 45 runs of -206, Table 6.

One pattern -206 can fill the orders previously done by two and yield as much surplus, so one may ask why we did not use it to begin with. Our optimiser would, using stock in Table 3 for 45/1 runs of what was -206 so far, but manually one could well have obtained any other plan.

Table 6. Intermediate and final Tableau for simple data

Run	3		-106	-406	500	600	Run	4		-206	-406	500	600
	0		30.00	15.00	.00	5.00		0		45.00	.00	.00	5.00
3600	0		.00	1.00	.00	6.00	3600	0		.00	1.00	.00	6.00
700	0		.33	-.33	1.00	-2.00	700	0		.50	-.50	1.00	-2.00
-206	0		-.67	-.33	.00	.00	-106	0		-1.50	.50	.00	.00
-306	0		-.33	-.67	.00	.00	-306	0		-.50	-.50	.00	.00
-507	0		.00	-1.00	6.00	-5.00	-507	0		.00	-1.00	6.00	-5.00

Professionals may want to explore alternative plans for our simple data, notably go for an exact, no-surplus solution. They will see that it does not exist (nor does it usually, **so one should apply zero tolerance only for academic reasons**). Here an exact plan occurs if demand for 600s is reduced from 85 to 84 -- and then exactness need not be enforced.

Anyway, for us the critical task was to document that a single pattern may replace two. So the question is, "How do we recognise the combining patterns and how do we transform them into one?"

TWO PATTERNS COMBINED INTO ONE

To continue with our example, if, manually or otherwise, we obtained the two-setup plan at left in Table 6, how can we find out that its patterns can combine into the single-run pattern (2 2 2), -206 previously?

So far all relevant patterns were in the table from the beginning: -206 has been there, but was not chosen because it did not fill demand as neatly as -106 & -406, Table 4. Yet once demand is filled, including the "demand" for 45 rolls, the effect of -206 on its predecessors is clear, as shown again at left in Table 7.

If -206 had been entered initially with the other patterns, it would now have coefficients -.67 & -.33 or 2/3 and 1/3, to replace either one of the active patterns by 45 runs, reducing the other one to 0, Table 6.

But, of course, we generate patterns dynamically, one at a time, so the new pattern would not yet have entered, nor would any unused patterns, while the current Tableau would be the one at right in Table 7, with the previous initial patterns numbered -106 & -206.

So, if pattern (2 2 2) is now generated dynamically, it would be -306. How do we find out that it can replace its two predecessors?

One way is to update (2 2 2 1) to (-.67 -.33 0 0) like any new pattern, as described in our companion papers (Diegel 2012b, 2012c). But for now, not knowing that the new pattern will be successful, the approach more easily understood is as described in Table 8.

45 runs of the new pattern produce the same output as its predecessors, so it can replace them. In turn, now that we know that pattern's value, it is worth updating (2 2 2 1) to (-.67 -.33 0 0) and do the final pivot as we did above, 30/.67 or 15/.33 making 45 runs either way. Or better yet, as we said after Table 6, enter 45 runs of (2 2 2) to begin with.

Table 7. Repeating many-patterns approach and dynamic version

Run	3		-106	-406	500	600	Run	3		-106	-206	500	600
	0		30.00	15.00	.00	5.00		0		30.00	15.00	.00	5.00
3600	0		.00	1.00	.00	6.00	3600	0		.00	1.00	.00	6.00
700	0		.33	-.33	1.00	-2.00	700	0		.33	-.33	1.00	-2.00
-206	0		-.67	-.33	.00	.00	-306, dynamic combining pattern						

Table 8. Checking whether one pattern can replace two

Hold the current runs, 30 and 15, of the 2 active patterns.

Recover the raw data for these patterns (they were stored anyway to be shown in the summary). Here it is (3 0 3) & (0 6 0), plus stock.

Multiply each pattern's reels by its runs:

$$30*(3\ 0\ 3) = 90\ 0\ 90, \text{ then } 15*(0\ 6\ 0) = 0\ 90\ 0$$

Add up the sums, 90 90 90 and divide by pattern (2 2 2), 45 45 45.

Table 9. Two fractional plans where one pattern can replace two

Width mm	700	600	500	3600	Width mm	700	600	500	3600		
Demand n	90	85	90	Loss	Demand n	100	100	100	Loss		
-106)	30.00	3	0	3	0	-106)	33.33	3	0	3	0
-406)	14.17	0	6	0	0	-406)	16.67	0	6	0	0
Sum	44.17	90	85	90	0	Sum	50.00	100	100	100	0

Nevertheless, since the optimal pattern may not be obvious initially, it may be worth reporting that the above rules may also solve fractional plans. For example, Table 9 at left has the plan from Table 4, then at right a similar result for demand raised to 100 for all reels.

At left, 14.17 would normally be rounded up to 15, then $30*(3\ 0\ 3)$ plus $15*(0\ 6\ 0) = 90\ 90\ 90$ or 45 runs of (2 2 2). Similarly, 50 runs of that pattern yield $33.3333*(3\ 0\ 3) + 16.6667*(0\ 6\ 0)$ or 100 reels each. So the good news is that 2-combining may define single-run patterns. The bad news is that it does not cover plans such as the one with 3 setups in Table 5. We turn to that topic in a companion paper.

SUMMARY

Re-combining two active patterns is a simple and useful tool in reducing the number of setups in the deckling problem. Indeed, it is astonishing to see how often old patterns combine into new ones, with fewer setups. Yet dealing with more than two patterns is still to be explained.

REFERENCES

Allwood, J. M. and C. N. Goulimis, Reducing the number of patterns in one-dimensional cutting stock problems, Electrical Engineering Dept., Imperial College, London SW7 2BT, June 1988.

Diegel, Adolf (1987). Cutting paper in Richards Bay: Dynamic local or global optimization in the trim problem? *ORiON, Journal of the Operations Research Society of South Africa* 3:42-55.

Diegel, Adolf (1988a). Integer LP solution for large trim problems. Joint International Conference EURO/TIMS, Paris, France, 6-8 July 1988.

Diegel, Adolf (1988b). Critical sizes in integer trim problems. Fourth South African Pulp and Paper Technical Conference, Durban, South Africa, 27-28 October 1988.

Diegel, Adolf (1990). Absolute integer solution for very large trim problems. IFORS-90, Athens, Greece, 25-29 June 1990.

Diegel, Adolf, Edouard Montocchio, Edward Walters, Sias van Schalkwyk and Spurs Naidoo (1996). Setup minimising conditions in the trim loss problem. *European Journal of Operational Research* 95:631-640.

Diegel, Adolf, Olaf Diegel, Edouard Montocchio, Sias van

Schalkwyk (2011b). Order Spread and Order Splits in the cutting stock problem. *TAPPSA Journal* Volume 2:38-47.

Diegel, Adolf, Olaf Diegel, Edouard Montocchio, Sias van Schalkwyk and George Pritchard (2012b). Compact linear programming for the reel cutting stock problem. *TAPPSA Journal* Volume 2:46-54.

Diegel, Adolf, Olaf Diegel, Edouard Montocchio, Sias van Schalkwyk and George Pritchard (2012c). Dynamic pattern generation for the Reel Cutting Stock Problem. *TAPPSA Journal* Volume 3:34-40.

Diegel, Adolf, Olaf Diegel, Edouard Montocchio, Sias van Schalkwyk and George Pritchard (2012d). Optimising order splits and run length in the Reel Deckling Problem. *TAPPSA Journal* Volume 4:27-32.

Foerster, Hildegard (1998). Fixkosten- und Reihenfolgeprobleme in der Zuschnittplanung, Peter Lang, Europäischer Verlag der Wissenschaften, Frankfurt am Main.

Gau, Thomas and Gerhard Wäscher (1995). CUTGEN1 - A problem generator for the one-dimensional cutting stock problem. *European Journal of Operational Research* 44:572-579. Special Interest Group Cutting and Packing (<http://prodlog.wiwi.uni-halle.de/sicup/>).

Gilmore, P. and R. Gomory (1961). A linear programming approach to the cutting stock problem. *Operations Research* 9:849-859; 1963, 11:863-888.

Gilmore, P. and R. Gomory (1964). Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research* 12:94-120.

Goulimis, C. N. (1995). Optimal solutions for the cutting stock problem. *European Journal of Operational Research* 44:197-208.

Johnston, R. E. (1984). Cutting patterns and cutter schedules. *Asia Pacific Operational Research Journal* 4:3-14.

Johnston, R. E. (1986). Rounding algorithms for cutting stock problems. *Asia Pacific Operational Research Journal* 3:166-171.

Kantorovich, I. W. (1962 reprint). Mathematical Methods of Organising and Planning Production. *Management Science* 6:366-422.

Kemeny, John G. et al. (1972). Finite Mathematics With Business Applications. Second Edition, Prentice-Hall, Englewood Cliffs, p. 282. Cf. also 1962 edition p. 387.

Wagner, Harvey M. (1970). Principles of Management Science. Prentice-Hall, Englewood Cliffs, p. 103. Cf. also 1975 edition p. 108.