Dynamic pattern generation for the Reel Cutting Stock Problem

Adolf Diegel[1], Olaf Diegel[2], Edouard Montocchio[3], Sias van Schalkwyk[3], George Pritchard[4]

Abstract

Narrow reels are to be deckled from wide rolls so as to fill the orders on hand with minimal stock, the main task. We also consider setups, run length, order splits, surplus or deficit, pattern or reel multiples and related aspects: any one may be given priority even if this entails more stock. Either way, we are to combine reel widths in patterns, respecting stock and machine related factors, "pattern generation".

The static or classic way of generating patterns is to do so before any optimisation begins and, indeed, to list all possible patterns. We do so in a companion paper, but know that it is feasible only with few orders, small stock rolls or if knife settings or narrow waste disposal chutes preclude many patterns. Otherwise, orders may combine in astronomical numbers and computations become too complex to be practical or else, in omitting some patterns, we run the risk of missing critical ones.

This is where dynamic pattern generation enters. "Dynamic" now means "as one goes along, step by step" rather than "beforehand, all at once". We examine one pattern at a time, assess its effect on the plan so far and retain only a good one. We do this first to fill demand, then, with a new formula, to improve a solution. This dynamic, re-iterative approach is essential for realistic data. We aim to explain it in plain English.

Keywords: Compact Linear Programming, Column/Pattern Generation, CSP

Introduction

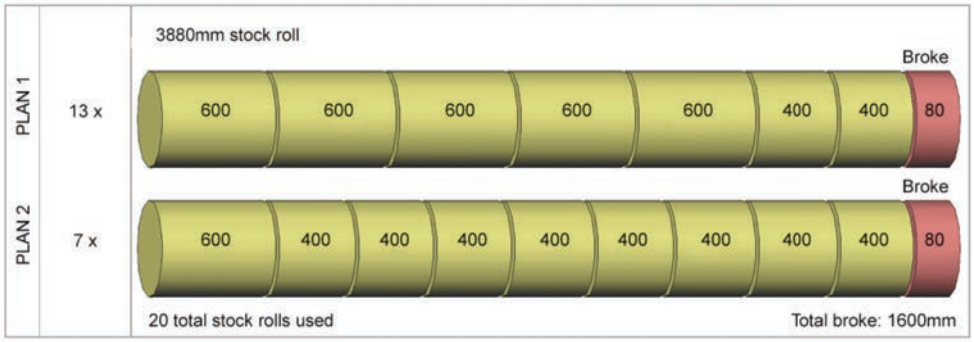
Our topic goes back to Gilmore and Gomory (1961 p. 849, 1963, 1964): "A linear programming approach ... enables one to compute always with a matrix which has **no more columns than it has rows**." Rows initially held orders, a fixed number, so "no more columns" concerns patterns, those added, one by one, in a new column to cover as yet unfilled demand, row by row (Diegel 1996). Thus orders started in rows, patterns in columns.

Yet "dynamic" covers either layout. We prefer (many) patterns in rows and (few) orders in columns, to keep tables narrow, feasible to print. Also, reels appear by column in a final cutting plan, Figure 1.

To appreciate such a plan, one should observe a paper machine. Wet pulp enters, dry paper exits, so fast that it can be wound, but not slit. The output is a machine roll, pope-reel or jumbo, a unit so wide, thick and heavy that it cannot be shipped. So the first task is to rewind a jumbo into smaller sets: stock units or rolls of a given diameter, to be slit and rewound as narrow reels (or cutreels, to be processed again).

This is done on a winder, in first-stage deckling. It has been called "cutting", yet it involves several steps. Cores are supplied and blades set by a winder crew or by automated devices. The winder unwinds a jumbo and, while slitting the paper, rewinds it onto the cores. This happens so fast that the winder can only deckle (unwind, slit and rewind). It cannot chop sheets nor slit reels below its knife settings. Narrow reels and sheets are combined as parent reels or cutreels for the winder, to be processed on a second machine, two-stage (Diegel 2012a)

[1] Cutline Research, 66 Edmonds Road, Durban 4001, South Africa. Corresponding author, cutworks7@iafrica.com, www.cutworks7.com
[2] Massey University, Auckland, New Zealand. Email: o.diegel@massey.ac.nz, mechatronics.massey.ac.nz
[3] Mondi Business Paper South Africa, Durban and Richards Bay. Email: ed.montocchio@mondigroup.co.za, sias.vanschalkwyk@mondigroup.co.za
[4] Mpact Limited, gpritchard@mpact.co.za, South Africa



(Top) Figure 1. Simple picture-type cutting plan

Width mm	600	400	3880
Demand n	72	80	Loss
-207) 13	5	2	80
-609) 7	1	8	80
Sum 20	72	82+	1600
Slits	2	2	4

Width mm	600	400	3880
Demand n	72	80	Loss
-106) 4	6	0	280
-408) 16	3	5	80
Sum 20	72	80	2400
Slits	2	1	2

(Bottom) Table 1. Quantitative plans: reels & roll, pattern runs and output

To return to the picture plan at hand, it tells the winder crew all it needs to know: where to set knives and how often a pattern is to run. But for a planner it omits crucial data. How many reels are wanted for each width? How many are produced? Are there deficit or surplus, split orders, short runs, wide offcuts? Of several plans, which one is best?

A quantitative plan provides answers based on demand, a pattern's label (negative and with n-across to separate it from reels, see Table 2), run length, reels for each order (n-up, "Number up") and output, Table 1

Patterns in rows make it easy to evaluate plans. At left the shortest run is 7, with 2 surplus 400s and 1600 mm loss. Plan B fills demand exactly, but with more waste and a shortest run of 4; yet order splits fell from 4 to 2, all 400s coming from 1 run. Will a paper mill profit from surplus? Will clients want it and wait until all runs are finished? Or do they prefer to have early delivery, after at most 16 vs 20 runs for 400s?

It is not for us to decide, nor need we show 8 other plans for the same data (Diegel 2012b). We do imply that good plans may entail testing many patterns. We aim to do so for all of them, but store only the good ones, those which fill more demand with less loss and longer runs.

A simple example

A dynamic approach may retain few final patterns, but it is more easily understood in the context of all possible ones. To list them, we use an example so small that, on one page, we can show tables vertically (with many patterns) and horizontally (for few orders). We also want tables side by side to study them before and after testing a pattern. Table 1 thus came from Table 2, the LP-part usually called a "Tableau".

Row 1 has a Tableau's Number, 0 initially. Next comes 20, minimal stock usage (which may be exceeded by poor fits or if extra rolls are used). 20 comes from the first two rows, reel width and negative, unfilled, demand for 72 reels of 600 mm and 80 400s: these add up to 75200 mm, so 3880 mm stock makes 20 rolls, integer round-up. 3880*20 is 77600 mm, with 2400 mm as slack for pattern loss (or surplus in a good plan). "20" appears twice on top of the Tableau, first as the initial estimate, then for the rolls still available. Finally, we use Compact LP, without Identity Matrix, as in our companion paper (Diegel 2012b).

Patterns appear in subsequent rows, with negative labels to distinguish them from orders. Labels include the reels in each pattern, "n-across", 06 to 09. Then we list pattern

0	20 Loss	600	400	3880	Table number, usage, reels & roll
	.00	-72	-80	-20	Loss, demand, minimal stock rolls
-106	280.00	6.00	.00	1.00	
-207	80.00	5.00	2.00	1.00	Pattern labels, pattern loss,
-307	280.00	4.00	3.00	1.00	reels-in-pattern and 1 roll
-408	80.00	3.00	5.00	1.00	for each potential run
-508	280.00	2.00	6.00	1.00	
-609	80.00	1.00	8.00	1.00	More patterns, down to -709 with
-709	280.00	.00	9.00	1.00	9 400s and 280 mm loss

Table 2. Initial Tableau with all full-cut patterns

	0	20 Loss	600	400	3880
		.00	-72.00	-80.00	-20.00
-408		80.00	3.00	5.00	1.00
3880		80.00	-1.50	9.50	2.50
-207		.00	2.00	-3.00	-1.00

	1	20 Loss	600	400	-408
		1600.00	-12.00	20.00	20.00
3880		80.00	3.00	5.00	1.00
600		.00	.50	-1.50	-.50

	1	20 Loss	600	400	-408
		1600.00	-12.00	20.00	20.00
3880		80.00	3.00	5.00	1.00
-207		.00	2.00	-3.00	-1.00

	2	20 Loss	-207	400	-408
		1600.00	6.00	2.00	14.00
3880		80.00	-1.50	9.50	2.50
600		.00	.50	-1.50	-.50

Table 3. Initial dynamic solution with pattern labels from Table 2

	3	20 Loss	-207	400	3880
		1152.00	14.40	-51.20	-5.60
-408		32.00	-.60	3.80	.40
600		16.00	.20	.40	.20

	4	20 Loss	600	400	3880
		.00	-72.00	-80.00	-20.00
-408		80.00	3.00	5.00	1.00
-207		80.00	5.00	2.00	1.00

Table 4. Returning to the raw data, "pivot backwards"

Width mm		600	400	3880
Demand n		72	80	Loss
-408)	14	3	5	80
-207)	6	5	2	80
Sum	20	72	82+	1600

Width mm		600	400	3880
Demand n		72	80	Loss
-207)	13	5	2	80
-609)	7	1	8	80
Sum	20	72	82+	1600

Table 5. First dynamic solution and a better plan with pattern -609

loss. It so happens that, for 2 orders, one less of one reel allows one more of the other: pattern loss changes by 600 - 400 or 200 mm, from 280 to 80 mm and back again.

Before turning to programming details, one may want to see a solution. Table 3 shows a dynamic initial plan for patterns –408, then –207, from Table 2 (with their static labels; dynamic ones would be consecutive).

If we treat the "demand" for 20 rolls as we do a client's, 20 runs of pattern -408 use all expected stock in Tableau 1. This yields surplus 400s, but does reduce remaining demand to 12 600s, more than any other pattern. Then -207 added to Tableau 1 fills that demand in Tableau 2: its 6 runs reduce -408's to 14 and surplus from 20 to 2 400s.

Anybody can see that 20 runs of -408 affect 600s and 400s. Yet without having studied LP, one may be surprised by zero loss and negative data in row -207: it was revised by entering -408 and we now see its **nett effect**; similarly for fractions and other numbers in Tableau 2. Before explaining this, we show that it is consistent by reversing what we did in Table 3: we "pivot backwards" from Tableau 2, Table 4.

Indeed, returning the stock column in Tableau 3 shows that -207 leaves more unfilled demand than -408, 51.20 400s against 12 600s. Then Tableau 4 restores demand and our two dynamic patterns with their raw data. This confirms that we were numerically correct and consistent.

Is a low number of dynamic patterns a valid objective? It is initially, just to fill demand. But usually we want to check and possibly improve other aspects of a plan. For example, we compare the current plan with an alternative, Table 5.

Excepting run length, results are the same. The winder crew likes longer shortest runs, so we can objectively prefer Plan B to A (though other aspects may dominate run length, see Table 1). For now, critical is that 2 orders use 2 patterns, yet in distinct plans. As critical is that we do not normally get the second plan initially, but only after accepting the one at left: we then add -609 in a third round, replacing -408. We should now explain how to do this, how a pattern can be evaluated dynamically and enter a Tableau only after it was found to be good.

Dynamic initial patterns

The secret of LP (and of solving simultaneous equations, Carl Friedrich Gauss 1777 to 1855) is that, upon activating a new pattern, others are revised. We explained in our companion paper (Diegel 2012b) how this is done in a static approach with all patterns initially in Tableau 0. Now, generating dynamic patterns, we ask, "How do we revise a pattern which has been generated tentatively?"

We answer this question by solving our example step by step as we do for genuine data: we do not normally begin with the stock column as we did above, but choose a pattern for an as yet unfilled demand and continue to do so until all orders are filled. Only then do we check stock: if it is zero, nothing needs to be done; if negative, "unused", it may become surplus; else we generate more patterns to try to reduce stock usage.

To document this, we begin with the static Tableau from Table 2. The raw data are at left in Table 6. At right are the objective function and all patterns as they would appear after activating -408 to produce 400s.

Indeed, only pattern -108 is the same at left and at right

(because of 0 in the new -408 column). Yet all other rows are revised, in all columns. Obviously, then, entering a pattern dynamically, one at a time, should get the same result. To document that it does, the Intermediate Tableau at left in Table 7 prepares the revision of rows 1 and 2, then of pattern -408: exchange labels, save raw data, then replace those cells, -80 by 0 in row 2 and 5 by 1 in the pattern (to simulate Gauss's Identity Matrix).

Obviously, we obtain the same result as in the top rows of Table 6 for static patterns. This confirms what we expected:

dynamic patterns use the same rules as static ones.

So does revising a second pattern. For example, it would start with the raw data in row -207 at left in Table 8. Then we revise it dynamically by the rules below to obtain a new row -207 at right: indeed, it is the co-responding row in Table 6.

Once -207 is revised, it fills demand with the rules specified in Table 7, "exchange labels" and so on. One could fill 600s with -24/3.80 or 6.316 runs for the first Tableau 2 in Table 9, with -.211 unused rolls. Yet, once a pattern is revised, one may check other pivots: -4 rolls in Tableau 1 could make -4/.60 or 6.67 runs of -207 and 1.33 surplus 600s, second Tableau 2.

Finally, still in Tableau 1 of Table 9, we could exchange unused rolls for 20 400s, third Tableau 2 (also obtainable from the second Tableau 2 by exchanging -207 for 6.67/.33 or 20 400s). Creating surplus instead of filling demand may seem futile, but here it does yield integer runs for both patterns, Tableau 3 in Table 10. The same plan results from the first Tableau 2 in Table 9 by exchanging stock for .211/.105 or 2 400s, but why risk intermediate fractions?

Our example showed how to revise two dynamic patterns and, by chance, we had an integer plan. This is less likely

Initial Tableau 0

0	20 Loss	600	400	3880
	.00	-72	-80	-20
-108	280.00	6.00	.00	1.00
-207	80.00	5.00	2.00	1.00
-307	280.00	4.00	3.00	1.00
-408	80.00	3.00	5.00	1.00
-508	280.00	2.00	6.00	1.00
-609	80.00	1.00	8.00	1.00
-709	280.00	.00	9.00	1.00

Tableau 1, after entering -408

1	20 Loss	600	-408	3880
	1280.00	-24	16.00	-4.00
400	16.00	.60	.20	.20
-108	280.00	6.00	.00	1.00
-207	48.00	3.80	-.40	.60
-307	232.00	2.20	-.60	.40
-508	184.00	-1.60	-1.20	-.20
-609	-48.00	-3.80	-1.60	-.60
-709	136.00	-5.40	-1.80	-.80

Table 6. Raw data for all patterns, then after entering -408

At left: Exchange pivot row & column labels, -408 & 400.

Still at left in new row: Save PIVOT = 5 & set to 1, then

save RATIO = DEMAMD/PIVOT, -80/5 = -16 & set Row 2 to 0.

Intermediate 0/1 and labels

0	20 Loss	600	-408	3880
	.00	-72	**-80/0**	-20
400	80.00	3.00	**5/1**	1.00

Tableau 1, after updates

1	20 Loss	600	-408	3880
	1280.00	-24	16.00	-4.00
400	16.00	.60	.20	.20

Update objective: **New Row 2 = Old Row 2 - RATIO * Old Pivot Row**

Loss: 0 - -16*80 = 1280
Run -408: 0 - -16*1 = 16
600: -72 - -16*3 = -24
Stock: -20 - -16*1 = -4

New Pivot Row = Old Row / Saved PIVOT, 80/5 = 16, 3/5, 1/5, 1/5

Table 7. Revising initial Tableau via Intermediate Table

Simulated pattern -207

1	20 Loss	600	-408	3880
	1280.00	-24	16.00	-4.00
400	16.00	.60	.20	.20
-207	80.00	5.00	**2/0**	1.00

Same pattern revised for -408

1	20 Loss	600	-408	3880
	1280.00	-24.00	16.00	-4.00
400	16.00	.60	.20	.20
-207	48.00	3.80	-.40	.60

For -207, save n-up in cell of pivot column, **PIVOT = 2 & set to 0.**

Revise -207 at right: **New = Old - PIVOT * Previous-pivot-row**

Loss: 80 - 2*16 = 48
-408: 0 - 2*.20 = -.40
600: 5 - 2*.60 = 3.80
Stock: 1 - 2*.20 = .60

Table 8. Raw data for new -207, then revised for previous -408

-207 revised for -408					New pattern fills demand				
1	20 Loss	600	-408	3880	2	20 Loss	-207	-108	3880
	1280.00	-24.00	16.00	-4.00		1583.16	6.316	13.474	-.211
400	16.00	.60	.20	.20	400	8.42	-.158	.263	.105
-207	48.00	3.80	-.40	.60	600	12.63	.263	-.105	.158

-207 used for surplus 600s					Integer plan with surplus 400s				
2	20 Loss	600	-408	-207	2	20 Loss	600	-408	400
	1600.00	1.33	13.33	6.67		1600.00	-12.00	20.00	20.00
3880	80.00	6.33	-.67	1.67	3880	80.00	3.00	1.00	5.00
400	.00	-.67	.33	-.33	-207	.00	2.00	-1.00	-3.00

Table 9. Alternative plans after updating pattern -207

2	20 Loss	600	-408	400	3	20 Loss	-207	-408	400
	1600.00	-12.00	20.00	20.00		1600.00	6.00	14.00	2.00
3880	80.00	3.00	1.00	5.00	3880	80.00	-1.50	2.50	9.50
-207	.00	2.00	-1.00	-3.00	600	.00	.50	-.50	-1.50

Table 10. Initial integer solution with two patterns

Width mm		600	400	3880
Demand	n	72	80	Loss
-207)	8	5	2	80
-408)	8	3	5	80
-508)	4	2	6	280
Sum	20	72	80	2400

Width mm		600	500	400	3880
Demand	n	72	60	80	Loss
-108)	16	2	2	4	80
-207)	10	4	2	1	80
-308)	2	1	4	3	80
Sum	28	74+	60	80	2240

Table 11. 3-pattern plans for the previous 2, then for 3 orders

with more data, so we should generate patterns not merely to fill demand, but to improve plans. Yet revision after using the stock column requires a new rule (to the best of our knowledge, not published elsewhere).

Initial revision of further patterns

Before defining a new rule, we should explain dynamic revision for more than 2 patterns. 3 can occur for 2 orders as shown at left in Table 11, one of 7 such plans in our companion paper. Yet these were found after filling demand, so they cannot help to explain before/after revision. Rather, we propose 3 orders with 3 initial patterns, Plan B.

The plan at left is consistent, but does not compete with previous ones. At right, we added 60 500s for 3 initial patterns. Each has 3 orders, so new patterns will be revised by their predecessors. To show how this is done we start at the beginning, then revise -207. Either pattern is to minimise remaining demand, Table 12.

Clearly, a first pattern involves mere division, "Old/Pivot". The second pattern requires "Old - Pivot*Previous row", two steps, yet still simple arithmetic. This is so also for all subsequent patterns needed to fill demand, but **each is revised by all its non-zero-pivot predecessors**.

In other words, a dynamic pattern is added one at time, but revising it entails the same procedure as a static approach (with all patterns in Tableau 0): a dynamic one reflects its predecessors as if it had been in the raw data table to begin with. So initial patterns involve not magic, but consistent programming: for any previous and still active pattern, recover its pivot and proceed as we did for pattern 2. We document this for Pattern 3 in Table 13, that is, after -108 & -207 were used to fill demand for 400S & 600s.

0	28 Loss	600	500	400	3880	Save PIVOT = 4,
	.00	-72.	-60.	-80.	-28.	set to 1, then
-108	80	2	2	**4/1**	1	**New = Old/PIVOT**
1	28 Loss	600	500	-108	3880	80/4= 20 2/4=.50
	1600.00	-32.000	-20.000	20.000	-8.000	2/4=.50 1/4=.25
400	20.00	.500	.500	.250	.250	1/4=.25
		Adding raw data for -207				
1	28 Loss	600	500	-108	3880	Save PIVOT = 1 under
	1600.00	-32.000	-20.000	20.000	-8.000	-108, set to 0, then
400	20.00	.500	.500	.250	.250	**New = Old - PIVOT ***
-207	80.00	4	2	**1/0**	1	**Previous row cell**
1	28 Loss	600	500	-108	3880	Revised -207:
	1600.00	-32.000	-20.000	20.000	-8.000	80-20=60 4-.50=3.50
400	20.00	.500	.500	.250	.250	2-.50=1.5 0-2.5=-.25
-207	60.00	3.500	1.500	-.250	.750	1 - .250 = .750

Table 12. First two patterns (2 2 4) and (4 2 1) for 3-order plan

Once -308 is revised, remaining stock makes 1.143/.357 = 3.20 runs. This affects previous runs and yields 2.40 surplus 500s, Tableau 3 in Table 14. We exchange fractional 500s for 2.40/1.20 600s in Tableau 4.

The first 3-pattern integer plan has minimal loss, but if we go beyond filling demand, we obtain several alternatives and finally Plan B. It has 2 patterns, 1 split order, yet 2 surplus 600s are wasted. A practitioner will decide which plan is best. For us, critical is that results are so consistent that dynamic revision must have been correct. It is also that one should go beyond an initial plan and try to improve it. So we must explain how to revise patterns after demand was filled.

Revising patterns after filling demand

The difference between initial and subsequent revision is this: once we pivoted on the stock column, **a new pattern is adapted by data in the stock row rather than by previous patterns**. The corresponding author programmed this in the 1980s (Diegel 1984, 1987, 1988) and has not found it published, but we can say "it works" and present it as follows.

"New = Old - Pivot*Previous pattern row" still defines the arithmetic, yet now revising data do not come from active patterns, but **from the stock row**, after pivoting on the stock column, and "pivot" is a raw number in the new pattern. To illustrate, we return to our 2-order data. The initial solution and the raw data for the new pattern -609 appear in Tableau 3 of Table 15. At right we show its cutting plan to verify runs of 6 & 14. Then we repeat Tableau 3, but with -609 revised so as to yield Tableau 4 with 13 & 7 runs: with the same loss, clearly these beat 6 & 14.

We can see the reason for a new approach by noting that -609 no longer fits into the old table: its data do not

		Raw data for -308 after -207				Save -308 pivots in
2	28 Loss	-207	500	-108	3880	active pattern columns:
	2148.57	9.143	-6.286	17.714	-1.143	P108 = 3 and P207 = 1,
400	11.43	-.143	.286	.286	.143	set each to 0, then:
600	17.14	.286	.429	-.071	.214	**New = Old - PIVOT ***
-308	80.00	**1/0**	4	**3/0**	1	**Previous row cell**

Revising -308 for -108: 80-3*11.43 = 45.71; 0-3*-.143 = .429;
4-3*.286 = 3.143; 0-3*.286 = -.857; 1-3*.143 = .571

-308	45.71	.429	3.143	-.857	.571	**Revised for -108 & 400**

P207 = 1, so simply subtract row 600 from revised -308:

45.71 - 17.14 = 28.57; .429 - .286 = .143; 3.143 - .429 = 2.714;
-.857 - -.071 = -.786; .571 - .214 = .357

-308	28.57	.143	2.714	-.786	.357	**Revised also for -207**

Table 13. Entering and revising pattern (1 4 3) after -108 & -207

3	28 Loss	-207	500	-108	-308
	2240.00	9.60	2.40	15.20	3.20
3880	80.00	.40	7.60	-2.20	2.80
400	.00	-.20	-.80	.60	-.40
600	.00	.20	-1.20	.40	-.60

	Width mm	600	500	400	3880
	Demand n	72	60	80	Loss
-108)	16	2	2	4	80
-207)	10	4	2	1	80
-308)	2	1	4	3	80
	Sum 28	74+	60	80	2240

4	28 Loss	-207	600	-108	-308
	2240.00	10.000	2.000	16.000	2.00
3880	80.00	1.667	6.333	.333	-1.00
500	.00	-.167	-.833	-.333	.50
400	.00	-.333	-.667	.333	.00

	Width mm	600	500	400	3880
	Demand n	72	60	80	Loss
-608)	16	3	0	5	80
-707)	12	2	5	0	180
	Sum 28	72	60	80	3440
	Splits	2	1	1	2

Table 14. Initial fractional solution and integer plans for 3 orders

3	20 Loss	-207	-408	400
	1600.00	6.00	14.00	2.00
3880	80.00	-1.50	2.50	9.50
600	.00	.50	-.50	-1.50
-609	80	1?	8?	1?

	Width mm	600	400	3880
	Demand n	72	80	Loss
-207)	6	5	2	80
-408)	14	3	5	80
	Sum 20	72	82+	1600

3	20 Loss	-207	-408	400
	1600.00	6.00	14.00	2.00
3880	80.00	-1.50	2.50	9.50
600	.00	.50	-.50	-1.50
-609	.00	1.00	-2.00	.00

4	20 Loss	-207	-609	400
	1600.00	13.00	7.00	2.00
3880	80.00	-.25	1.25	9.50
-408	.00	-.50	-.50	.00
600	.00	.25	-.25	-1.50

Table 15. Initial integer solution, pattern revised and its effect

3	20 Loss	-207	-408	400	Current plan still has a reel
	1600.00	6.00	14.00	2.00	width in the top row, here 2
3880	80.00	-1.50	2.50	9.50	400s, but 0 would also count.
600	.00	.50	-.50	-1.50	
Temp	80.00	1.50	-2.50	-9.50	**Stock row with +/- n-up reversed**

Subtract pattern loss from Temp, 80 - 80 = 0,
then check Row 1 of current Tableau for Reel Labels, 400:

Raw pattern:		600	400	3880	**in a reel column**, add raw n-up in
-609	80	1	8	1	pattern to that column in Temp,
Temp	.00	1.50	-2.50	-1.50	here **-9.50 + 8 = -1.50**
600	.00	.50	-.50	-1.50	Current data in revising row 600
New	.00	1.00	-2.00	.00	-609, after **Temp - Pivot*Revisor**

Table 16. Revising new pattern after using stock column

refer to the labels in the top row, but to reels and stock width. So we keep (80 1 8 1) at hand, but do not add it to Tableau 3. Rather, we **copy the stock row's loss, then its numbers** with their opposite +/- signs as in the "Temp"-row in Table 16.

-609 loss is subtracted from the initial Temp, 80 - 80 or 0. Then the last Temp-cell has -9.50 400s. This reel is still in the top row, so add its raw n-up in the 400-column, -9.50 + 8 = -1.50, updating Temp for surplus. Finally, the first active reel number in (1 8) is the revising Pivot: 1.50 - 1*.50 = 1, -2.50 + .50 = -2, while the 400-column becomes 0. Surplus is not affected by -609, as indeed we saw in Tableau 4.

In short, beyond creating the Temp-row from the current Stock row and selecting a pivot from the new pattern's raw data, one can use the same subroutine to revise a pattern after filling demand. This would apply to static patterns as well as to dynamic ones.

We could repeat revision for the 3-order data above or any example, but space is short. Also, an interested person can download our optimiser at www.cutworks7.com and run it with Output Option 2, "full printout".

Conclusion

We showed that dynamic pattern generation uses the same LP-procedure to fill demand as the static case, with all possible patterns in Tableau 0. We then went beyond the initial phase by defining how to revise further patterns, to improve a plan, static or dynamic.

Computations are the same and we still check all possible patterns, but dynamically we retain only good ones, usually fewer than the orders at hand. This makes good planning practical even for many reel widths and wide stock rolls.

References

Diegel, Adolf (1984). Optimal dimensions of virgin stock in cutting glass to order. Decision Sciences, *The Journal for the American Institute of Decision Sciences* 15 2:260-274.

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 1:42-55. First presented at Annual Meeting, ORSA, University of Cape Town, 1986.

Diegel, Adolf (1988). Integer LP solution for large trim problems. Paper presented at EURO/TIMS, Paris, 6-8 July 1988.

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

Diegel, Adolf and Olaf Diegel (2012a). Formulating two-stage, two-machine deckling with compact LP. *TAPPSA Journal* Volume 1:31-39.

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.

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 (1963). A linear programming approach to the cutting stock problem, Part 2. *Operations Research* 11:863-888.

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