A Cutting-Stock Problem

Consider a paper mill. The paper-making machine produces "raw rolls" of width 100 inches, which are then cut by adjustable knives into "final rolls" of various widths. The mill has received orders for certain numbers of final rolls of various widths, and would like to fill these orders using as few raw rolls as possible.

There are various patterns (potentially a very large number of them) into which the raw rolls can be cut. Suppose the raw roll has width w, and the final roll widths are w_i for $i = 1 \ldots m$. A possible pattern consists of p_i finals of width w_i for each i, where p_i are nonnegative integers and $\sum_{i=1}^{m} p_i w_i \leq w$. Fortunately, we will not need to actually construct all the possible patterns: the method of "delayed column generation" will generate the patterns we need as we go along.

Let's say there are n possible patterns: for each one we will have a variable x_j, representing the number of raw rolls cut according to pattern number j. This pattern consists of p_{ij} finals of width w_i for $i = 1 \ldots m$. Then the total number of finals of width w_i we obtain will be $\sum_{j=1}^{n} p_{ij} x_j$, and we need this to be at least the number of orders b_i for this type of final. We obtain a linear programming problem:

$$\begin{aligned} \text{minimize } & \sum_{j=1}^{n} x_j \\ \text{subject to } & \sum_{j=1}^{n} p_{ij} x_j \geq b_i \text{ for } i = 1 \ldots m \\ & \text{all } x_j \geq 0 \end{aligned}$$

Actually, to make LINDO's "dual prices" positive rather than negative, I'll make this into a "standard primal" problem:

$$\begin{aligned} \text{maximize } & \sum_{j=1}^{n} -x_j \\ \text{subject to } & \sum_{j=1}^{n} -p_{ij} x_j \leq -b_i \text{ for } i = 1 \ldots m \\ & \text{all } x_j \geq 0 \end{aligned}$$

We will start out with a few patterns, enough to produce a feasible solution, and solve the problem with those patterns. The dual variable values y_i obtained from the solution can then be used to "price out" any proposed new pattern, to see if it would enter the basis. We have

$$\eta_j = \sum_{i=1}^{m} y_i(-p_{ij}) - (-1) = 1 - \sum_{i=1}^{m} y_i p_{ij}$$

In order for x_j to be chosen to enter the basis, we need $\eta_j < 0$, and we typically like to choose the most negative of these. So we will search for the best new pattern to enter the basis, i.e. the one that maximizes $\sum_{i=1}^{m} y_i p_{ij}$. If we find one we introduce it into the problem, and continue solving. When there are no patterns where this sum is greater than 1, we must have the optimal solution.

For the pattern search, I'll use LINDO's integer linear programming abilities: we want to maximize $\sum_{i=1}^{m} y_i p_i$ subject to $\sum_i w_i p_i \leq 100$, all $p_i \geq 0$ where p_i are integers. The restriction to integers makes this an integer linear programming problem, not just a linear programming problem.

For example suppose we have orders for the following finals (these numbers were suggested by the class):

Final width	Quantity
13"	21
17"	45
25"	11
33"	18
42"	15
63"	34

We can start with the following patterns, which contain all the final widths:

Pattern 1: $33'' + 63''$ (total width 96", waste 4")
Pattern 2: $13'' + 17'' + 25'' + 42''$ (total width 97", waste 3")

Our first LINDO file (`cutstock.ltx`) looks like this:

```
max -x1 -x2
st
c13  )  - x2 <= -21
c17  )   -x2 <= -45
c25  )   -x2 <= -11
c33  )   -x1 <= -18
c42  )   -x2 <= -15
c63  )   -x1 <= -34
end
```

LINDO's result is

```
        OBJECTIVE FUNCTION VALUE
        1)      -79.00000
  VARIABLE         VALUE          REDUCED COST
        X1         34.000000          0.000000
        X2         45.000000          0.000000
       ROW    SLACK OR SURPLUS     DUAL PRICES
      C13)         24.000000          0.000000
      C17)          0.000000          1.000000
      C25)         34.000000          0.000000
      C33)         16.000000          0.000000
      C42)         30.000000          0.000000
      C63)          0.000000          1.000000
```

With these two patterns, the best we can do is to use 79 raw rolls, 34 in pattern 1 and 45 in pattern 2. The dual prices are 1 for 17-inch and 63-inch finals, 0 for the others. Thus with these prices for those finals, we look for a pattern that will fit on the 100-inch raw roll and have total price more than 1.

The LINDO file (`cutgen.ltx`) for this pattern-search problem is

```
max 0.0 p13 + 1.0 p17 + 0.0 p25 + 0.0 p33 + 0.0 p42 + 1.0 p63
st
13 p13 +  17 p17 + 25 p25 + 33 p33 + 42 p42 + 63 p63 <= 100
end
gin 6
```

where the "`gin 6`" tells LINDO that the variables must be integers. LINDO's solution for this one is

```
        OBJECTIVE FUNCTION VALUE
        1)      5.000000
  VARIABLE        VALUE          REDUCED COST
       P13          0.000000          0.000000
       P17          5.000000         -1.000000
       P25          0.000000          0.000000
       P33          0.000000          0.000000
       P42          0.000000          0.000000
       P63          0.000000         -1.000000
       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         15.000000          0.000000
```

Thus we have our new pattern:

Pattern 3: $5 \times 17''$ (total width 85", waste 15")

We modify our problem file `cutstock.ltx` to include the new pattern with variable `x3` having coefficient -1 in the objective and -5 in the `c17` row:

```
max -x1 -x2 -x3
st
c13  )   -x2 <= -21
c17  )   -x2 -5 x3 <= -45
c25  )   -x2 <= -11
c33  )   -x1 <= -18
c42  )   -x2 <= -15
c63  )   -x1 <= -34
end
```

In some systems we might introduce this variable and continue from the basis that was optimal for the previous problem, which might only require one more pivot. In LINDO we will just solve the system from scratch; this doesn't matter here because for such a small problem the solution takes practically no time. The LINDO solution is

```
        OBJECTIVE FUNCTION VALUE

        1)      -59.80000

  VARIABLE        VALUE          REDUCED COST
        X1        34.000000          0.000000
        X2        21.000000          0.000000
        X3         4.800000          0.000000

       ROW   SLACK OR SURPLUS     DUAL PRICES
      C13)         0.000000          0.800000
      C17)         0.000000          0.200000
      C25)        10.000000          0.000000
      C33)        16.000000          0.000000
      C42)         6.000000          0.000000
      C63)         0.000000          1.000000
```

Now the number of raw rolls has decreased to 59.8, and the dual prices have changed. We enter the new dual prices in the pattern-search file, and solve, obtaining a new pattern:

Pattern 4: $7 \times 13''$ (total width 91", waste 9")

We continue in this way, entering each new pattern into the main problem file, solving, entering the dual prices into the pattern-search file and solving:

Pattern 5: $13'' + 2 \times 42''$ (total width 97", waste 3")
Pattern 6: $2 \times 17'' + 63''$ (total width 97", waste 3")
Pattern 7: $3 \times 33''$ (total width 99", waste 1")
Pattern 8: $4 \times 25''$ (total width 100", waste 0")
Pattern 9: $5 \times 13'' + 2 \times 17''$ (total width 99", waste 1")
Pattern 10: $25'' + 33'' + 42''$ (total width 100", waste 0")

At this point the optimal objective value in the pattern-search problem is 1.000002; presumably the extra .000002 is roundoff error and this should really be exactly 1, which means we are done. The optimal pattern is a new one, though, and there's no harm in trying it in the main problem:

Pattern 11: $5 \times 13'' + 33''$ (total width 98", waste 2")

However, this new pattern doesn't enter in the solution of the main problem, and the dual prices remain unchanged. The optimal solution for the main problem is:

```
        OBJECTIVE FUNCTION VALUE
        1)      -48.15909
  VARIABLE        VALUE          REDUCED COST
        X1        14.545455          0.000000
        X2         0.000000          0.022727
        X3         0.000000          0.204545
        X4         0.000000          0.045455
        X5         5.772727          0.000000
        X6        19.454546          0.000000
        X7         0.000000          0.045455
        X8         1.886364          0.000000
        X9         3.045455          0.000000
       X10         3.454545          0.000000
       X11         0.000000          0.000000
       ROW   SLACK OR SURPLUS     DUAL PRICES
      C13)         0.000000          0.136364
      C17)         0.000000          0.159091
      C25)         0.000000          0.250000
      C33)         0.000000          0.318182
      C42)         0.000000          0.431818
      C63)         0.000000          0.681818
```

Thus we need 48.15909 raw rolls, with 14.545455 in pattern 1, 5.772727 in pattern 5, 19.454546 in pattern 6, 1.886364 in pattern 8, 3.045455 in pattern 9 and 3.454545 in pattern 10.

The fact that the solution is not in integers could be a problem for a real paper mill. However, we can ask LINDO for the best integer solution (using the patterns generated so far) by adding the command `gin 11` at the end of the file. The result is a solution using 49 raw rolls (obviously the best we could hope for, since 48 is impossible without the restriction to integers): 17 in pattern 1, 2 in pattern 2, 1 in pattern 3, 1 in pattern 4, 6 in pattern 5, 17 in pattern 6, 2 in pattern 8, 2 in pattern 9 and 1 in pattern 10.