The **Cutting Stock Problem**, and solving it by the **Gomory-Gilmore algorithm**.

Given the next data: C=16, and

size	#
3	20
4	50
5	40

The meaning of the data is as follows: We have a small workshop where we do some carpentry work. More exactly, we have lumber with length of C=16 feet. Also, we have a demand for items with the sizes given in the table above, with the given cardinality. That is, we must produce 20 items (or pieces) of length 3; 50 items of length 4; and 40 items of length 5. We will produce them so that the stock material will be cut into pieces. For example, a lumber can be cut into 4 pieces of small items, each of length 4. We need to provide all items that are ordered, but we want to use as few number of lumbers as possible. This is the so called **Cutting Stock Problem**.

Let us try to find all “patterns” that are of sense:

	1	2	3a	3	4	5	6	7	8	9	10	11	
3	5	4	3	3	2	2	2	1	1	0	0	0	
4	0	1	1	0	2	1	0	3	2	4	1	0	
5	0	0	0	1	0	1	2	0	1	0	2	3	
	x1	x2		x3	x4	x5	x6	x7	x8	x9	x10	x11	

(Possibly) these are all patterns of interest.

For example pattern 3a will not needed, as pattern 3 dominates it!

So, we can list all not dominated patterns, that can be used, if we are “lucky”.

For example, if we will use pattern 2, then from some raw material unit we will gain 4 pieces of size 3 and one piece of size 4. If we cut, e.g. two such lumber stock using this pattern 2, we will gain 2*4=8 pieces of size 3 and 2*1=2 of size 4, and so on.

Let us write up the model of the problem (for the above input, supposing that these are the patterns of interest). Let x_i be the number of stocks that are cutter according to pattern i. We gain the next model:

$$5x1 + 4x2 + 3x3 + 2x4 + 2x5 + 2x6 + x7 + x8 \geq 20$$

$$x2 + 2x4 + x5 + 3x7 + 2x8 + 4x9 + x10 \geq 50$$

$$x3 + x5 + 2x6 + x8 + 2x10 + 3x11 \geq 40$$

$$x\_i \geq 0, \text{ integer, for all } 1 \leq i \leq 11$$

$$z = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 \quad \rightarrow \min$$

Are we done? Can we solve it?

One problem is, that the variables are integers. So not a simple LP, but an integer model that we should solve. This is not a big issue, as we can solve the relaxation (ignoring the integrality constraints), and after getting the optimal solution for the LP, all variables that are not integer in the optimal solution, will be rounded up. Since there are only few constraints (i.e. 3 constraints), we loose not too much from the optimality. (That is, suppose we need to cut 1500.8 stocks in the relaxed optimal solution, but in this relaxed optimal solution there are 3 variables, all are fractional numbers. Like x_i=500.1, x_j=300.2; x_k=700.5. We round all up, getting a new solution x_i'=501, x_j'=301; x_k'=701. Then we loose not more than 3, the optimal solution is not less than 1501 (as it must be integer), and our solution is 1503, the "gap" is only 2. Not too big loss in practical problems (if the number of different types is relatively small). But the main problem is something other:

But the main problem is, that if there are more different sizes (types), then the number of appropriate patters grow exponentially. The problem is treated, and solved in these publications:

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

The authors say, that if there are e.g. 40 item types (40 different sizes), then the number of patterns that should be considered, may be about 100 million! **We are not able to list them all**! (Recall that in 1961 the case was much harder, in sense of computability.)

The proposed solution (by Gomory-Gilmore) is:

- (we take = instead of ≥ , usually this also works)
- (we solve the relaxation, and the optimal relaxed solution is rounded up)
- **column generation** (solving a slave problem which is a knapsack problem to find a good pattern, which will enter the basis)
- **revised simplex**

The combination of the above two things makes possible to solve a considerably large problem also. We will demonstrate the solution for the above input.

Phase 0.

We start with the next patterns (columns):

$B=\{a^1,a^2,a^3\}$:

a^1	a^2	a^3
5	0	0
0	4	0
0	0	3

Then it is easy to see that B^{-1} is

1/5	0	0
0	1/4	0
0	0	1/3

The pricing vector is $\pi=cB*B^{-1}=[1,1,1]*B^{-1}=[1/5; 1/4; 1/3]$.

We look for a good column (that can enter the basis).

Generally it looks like $\underline{a}(a1,a2,a3)$. For example $a^1(5,0,0)$, $a^2(0,4,0)$, these are at moment in the basis. Some other can be like $\underline{a}(3,1,0)$ (see the initial table of possible patterns). Or $\underline{a}(2,2,0)$, and so on. But usually we are not able to list all needed column. So we take it generally: $\underline{a}(a1,a2,a3)$, where a1, a2 and a3 mean, respectively, that how many are cut from the size of 3, from the size of 4, and from the size of 5. This is the general sub-problem for the choice of an appropriate column (of the unknown simplex tableau):

$$\pi*\underline{a}-1 \rightarrow \max$$

$$\text{st. } 3a1 + 4a2 + 5a3 \leq 16$$

$$a1,a2,a3 \geq 0, \text{ integer.}$$

Here the constraint $3a1 + 4a2 + 5a3 \leq 16$ means that it is possible to gain so many pieces from the stock material. Naturally, a1, a2, a3 are all non-negative integers. And finally, recall, that the reduced cost can be calculated as the a(a1,a2,a3) vector is multiplied by the pricing vector, and the cost function coefficient is subtracted, i.e. $z_k-c_k = \pi *a_k-c_k$ (where a_k is the k-th column vector in the coefficient matrix A). Now our column vector is denoted above in the general model of the slave problem as $\underline{a}$ =a(a1,a2,a3) .

Now, all coefficients in the c vector are 1. So the reduced cost is $\pi*\underline{a}-1$. If this is positive, it is advantageous if the a vector in consideration enters the basis. This is why we maximize $\pi*\underline{a}-1$. In fact, if we find any a vector for which this objective is positive, it still applies.

So, now, our slave problem looks as follows:

$$1/5\ a1 + 1/4\ a2 + 1/3\ a3 - 1 \rightarrow \max$$

st. $3a1 + 4a2 + 5a3 \le 16$

$a1, a2, a3 \ge 0$, integer.

We realize that this is an upper bounded knapsack problem.

Let us calculate the gain/weight ratios:

$$1/5\ a1 + 1/4\ a2 + 1/3\ a3 - 1 \rightarrow \max$$

st. $3a1 + 4a2 + 5a3 \le 16$

$a1, a2, a3 \ge 0$, integer.

g/w: 1/15, 1/16, 1/15.

That is, optimal cost will be given, if the knapsack is completely full, and we pack only from the first type and third type!

That is, $\mathbf{a^4(2,0,2)}$ is an optimal solution.

Let us calculate the corresponding objective value: It is $z=2*1/5+0*1/4+2*1/3-1 = 2/5+2/3-1 > 0$, so it is advantageous to take this new vector a^4 into the basis.

At moment we know that $a^4 \rightarrow B \rightarrow$?

But, what vector will leave the basis?

We need to calculate the transformed form of a^4 (let us denote it by y), and the basis solution, usually denoted by x_B. Recall that $x_B=B^{-1}*b$ and $y= B^{-1}* a^4$. (Recall also so that **b(20,50,40)** is the b vector, and the basis inverse is:

1/5	0	0
0	1/4	0
0	0	1/3

After making the calculation we get $x_B(4;\ 25/2;\ 40/3)$ and $y(2/5;\ 0;\ 2/3)$.

Now we apply the minimum rule to decide what vector leaves the basis.

Min $\{4/(2/5);\ (40/3)/(2/3)\}=4/(2/5)$. The corresponding number is chosen to the pivot value in the y vector, that is: y(**2/5**;0;2/3). That is, the first row is chosen in the column vector y. This means that vector will leave the basis, which is in the first position. The current basis is $B=\{a^1,a^2,a^3\}$, so the first vector in the basis is a^1.

So we know that $a^4 \rightarrow B \rightarrow a^1$

Comes Phase 1.

Phase 1.

Now B={a^4,a^2,a^3} where the already considered columns are

a^1	a^2	a^3	a^4
5	0	0	2
0	4	0	0
0	0	3	2

Recall that y is as follows: y(**2/5**;0;2/3), where the first value is the pivot. It means that the basis transformation should be made so that

- the first row is divided by 2/5
- the second row remains as here we already do have a 0
- we subtract (5/3)-times of the first row from the third row (to eliminate the 2/3 by the 2/5)

But, **we make the transformation only on the basis inverse**.

The old basis inverse is:

1/5	0	0
0	1/4	0
0	0	1/3

The new basis inverse will be:

1/2	0	0
0	1/4	0
-1/3	0	1/3

Applying this new basis inverse, the new pricing vector is (the numbers in the matrix are added in each column to get): π(1/6; 1/4; 1/3).

With this new pricing vector the new slave problem looks like:

$$\frac{1}{6} a1 + \frac{1}{4} a2 + \frac{1}{3} a3 - 1 \rightarrow \max$$

$$\text{st. } 3a1 + 4a2 + 5a3 \leq 16$$

$$a1, a2, a3 \geq 0, \text{ integer.}$$

g/w: 1/18, 1/16, 1/15

Now, by some calculation (B&B or Dynamic Programming, or elementary calculation or Brute Force) we can confirm that the **maximum of z is 0**. It means that we already arrived to an optimal solution of the relaxed model. Still we need to calculate that what is the optimal solution. We get it as $x_B = B^{-1} * b$. After making the calculation we get

x_B(10; 12.5; 6.66). The optimum value (of the relaxed problem) is so 29.16.

Since the optimum value of the cutting stock problem is some integer number (the number of stock material that is cut), we know that the optimal solution of the cutting stock problem is at least 30 (29.16 is rounded up).

Let us try to round up also the components of the x_B vector. We get a new vector as x(10; 13; 7).

Since the current basis is $B=\{a^4,a^2,a^3\}$, this means that so many stock are cut from the patterns: We cut 10 stock according to column 4, we cut 13 according to pattern 2, we cut 7 according to pattern 3.

“We are lucky” in the sense, that the sum of the components is just 30, which means that we do have a “proven” optimal solution of the cutting stock problem. Let us summarize finally that what patterns are used, and how many pieces are cut.

size	a^1	a^2	a^3	a^4	Produced:
3	5	0	0	2	20
4	0	4	0	0	52
5	0	0	3	2	41
#	-	13	7	10	

We conclude that all demand is produced (and a bit more).