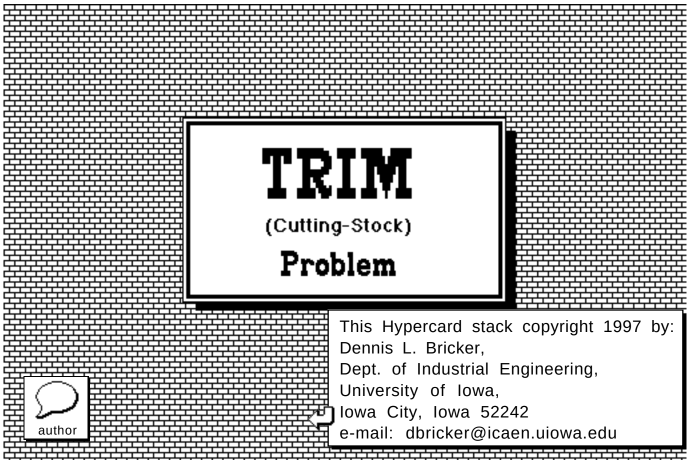
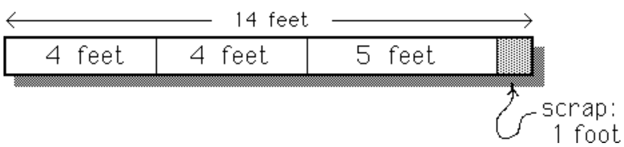



There are many valid "cutting patterns" which will provide the required lengths, e.g.:




**Example**

A certain material (e.g., lumber) is kept in stock in standard lengths:

| length (ft) | 9 | 14 | 16 |
|---|---|---|---|
| cost ($) | 5 | 9 | 10 |

An order is received:

| length (ft) | 4 | 5 | 7 |
|---|---|---|---|
| # pieces | 30 | 20 | 40 |

How should these required pieces be cut from the standard lengths, so as to minimize the cost?

**Notation**

$L_1, L_2, \ldots L_K$ standard lengths
$C_1, C_2, \ldots C_K$ costs

$\ell_1, \ell_2, \ldots \ell_I$ required lengths
$b_1, b_2, \ldots b_I$ # pieces required

$A^j$ cutting pattern #j
$k(j)$ index of standard length used by pattern #j
$A^j_i$ # pieces of length $\ell_i$ produced by pattern #j

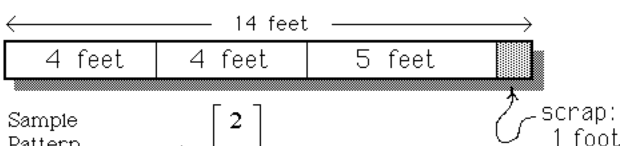
**Example**

In Stock:

| k | 1 | 2 | 3 |
|---|---|---|---|
| $L_k$ | 9 | 14 | 16 |
| $C_k$ | 5 | 9 | 10 |

Required:

| i | 1 | 2 | 3 |
|---|---|---|---|
| $\ell_i$ | 4 | 5 | 7 |
| $b_i$ | 30 | 20 | 40 |




Sample Pattern

$$A^j = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \qquad k(j) = 2$$

**Integer LP Model**

$X_j$ = # of times pattern j is used

$$\text{Minimize } \sum_{j=1}^{J} C_{k(j)} X_j$$

$$\text{s.t. } \sum_{j=1}^{J} A^j_i X_j \geq b_i, \; i=1,2,\cdots I$$

$$X_j \geq 0 \text{ and integer, } j=1,2,\cdots J$$

***If we ignore the integer restriction on Xi and solve the LP relaxation, we can often get a good integer solution by rounding, etc.***

In the LP relaxation, at most I patterns will be used (in a basic optimal solution)

In a typical problem, there may be an "astronomical" number of possible patterns, even though a relatively small number are used in the optimal solution.

**Column Generation Scheme for LP**

At each iteration of the revised simplex method, a ***simplex multiplier*** $\pi$ is computed.

A pattern $[a_1, a_2, \ldots a_I]$ may be entered into the basis if its ***reduced cost*** is negative, i.e., if

$$C_k - \pi a < 0 \quad \text{i.e.,} \quad \sum_{i=1}^{I} \pi_i a_i > C_k$$

and feasible, i.e., $\sum_{i=1}^{I} \ell_i a_i \leq L_k$

where $L_k$ is the length used by the cutting pattern

## Column Generation Scheme for LP

Given $\pi$, solve the knapsack problem, which (if $\pi a > C_k$) yields a pattern which may be entered into the solution.

Subproblem:

$$\text{Maximize } \sum_{i=1}^{I} \pi_i a_i$$
$$\text{s.t. } \sum_{i=1}^{I} \ell_i a_i \le L_k$$
$$a_1, a_2, \cdots a_I \ge 0 \text{ and integer}$$

## Solution of Knapsack Subproblems

If *dynamic programming* is used, then we will obtain a solution of the subproblem for each standard length. We can then add each pattern to the LP if the pattern's reduced cost is negative.

A ***heuristic*** method (which is faster than DP but doesn't guarantee the optimal solution to the knapsack problem) may also produce patterns with negative reduced cost to be added to the LP.

## Heuristic Knapsack Filling

$$\text{Maximize } \sum_{i=1}^{I} \pi_i a_i$$
$$\text{s.t. } \sum_{i=1}^{I} \ell_i a_i \le L_k$$
$$a_1, a_2, \cdots a_I \ge 0 \text{ and integer}$$

**Step 1**

Order the requested lengths according to the ratios of "payoff" per unit length, $\frac{\pi_{i_1}}{\ell_{i_1}} \ge \frac{\pi_{i_2}}{\ell_{i_2}} \ge \cdots \ge \frac{\pi_{i_I}}{\ell_{i_I}}$

## Heuristic Knapsack Filling

**Step 2**

Fill the knapsack with the maximum possible number of pieces of length $\ell_{i_1}$

$$a_{i_1} = \left\lfloor \frac{L_k}{\ell_{i_1}} \right\rfloor \quad \text{"floor" function}$$

**Step 3**

Fill what remains empty ($L_k - \ell_{i_1} a_{i_1}$) with as many pieces as possible of the next most "valuable" length ($\ell_{i_2}$), etc., until the knapsack is filled or the last length $\ell_{i_I}$ has been tried.

## DP Solution of Knapsack Problem

$$\text{Maximize } \sum_{i=1}^{I} \pi_i a_i$$
$$\text{s.t. } \sum_{i=1}^{I} \ell_i a_i \le L_k$$
$$a_1, a_2, \cdots a_I \ge 0 \text{ and integer}$$

$f_i(L)$ = maximum value which can be attained using lengths $\ell_1$ through $\ell_i$ when L units of capacity remain in the knapsack

$= \max \{ \pi_i a_i + f_{i-1}(L - \ell_i a_i) \}$ for i=1,2,... I

$f_0(L) = 0$

## DP Solution of Knapsack Problem

Compute $f_0, f_1, f_2, \ldots f_I$ recursively for all permissable values of L.

Thus, all subproblems have been solved by computing $f_I(L_k)$ for k=1, 2, ... K.

## Solution of Example

In Stock:

| k | 1 | 2 | 3 |
|---|---|---|---|
| $L_k$ | 9 | 14 | 16 |
| $C_k$ | 5 | 9 | 10 |

Required:

| i | 1 | 2 | 3 |
|---|---|---|---|
| $\ell_i$ | 4 | 5 | 7 |
| $b_i$ | 30 | 20 | 40 |

To begin our column generation algorithm, we shall (arbitrarily) propose the following three patterns cut from the 9-feet stock length:

$$\begin{bmatrix}2\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \& \begin{bmatrix}0\\0\\1\end{bmatrix}\begin{array}{l}\leftarrow \text{4-ft length}\\ \leftarrow \text{5-ft length}\\ \leftarrow \text{7-ft length}\end{array}$$

Our initial LP is

$$\begin{array}{l} \text{Minimize } 5X_1 + 5X_2 + 5X_3 \\ \text{s.t. } 2X_1 - S_1 = 30 \\ \quad X_2 - S_2 = 20 \\ \quad X_3 - S_3 = 40 \\ X_k \ge 0,\ k=1,2,3;\ S_i \ge 0,\ i=1,2,3 \end{array}$$

with the optimal solution:

$$\begin{cases} X_1=15,\ X_2=20,\ X_3=40,\ S_1=S_2=S_3=0 \\ \pi_1 = 2.5,\ \pi_2 = 5,\ \pi_3 = 5,\ \text{cost}= \$375 \end{cases}$$

## Heuristic Knapsack Filling

*Subproblem*

$$\begin{aligned}&\text{Max } 2.5a_1 + 5a_2 + 5a_3\\ &\text{s.t. } 4a_1 + 5a_2 + 7a_3 \le 16\\ &a_i \ge 0 \text{ \& integer, } i=1,2,3\end{aligned}$$

Let's try generating a cutting pattern from the longest (16-ft) stock using the heuristic method:

- sort the rqmts: $\frac{\pi_2}{\ell_2} > \frac{\pi_3}{\ell_3} > \frac{\pi_1}{\ell_1}$, i.e., $\frac{5}{5} > \frac{5}{7} > \frac{2.5}{4}$
- fill the knapsack with as many as possible of the 5-ft lengths, i.e., $a_2 = \left\lfloor \frac{16}{5} \right\rfloor = 3$
- after including 3 pieces of length 5 feet, $16 - 3\times5 = 1$ foot of capacity remains. This is insufficient for any of the other required pieces, i.e., $a_3 = a_1 = 0$.

***The heuristic method has produced the pattern*** $A^4=\begin{bmatrix}0\\3\\0\end{bmatrix}$ ***with reduced cost*** $10 - \pi A^4 = 10-15 = -5$

The reduced cost is negative, so it is worthwhile to add this pattern to the LP.

The new LP is

$$\begin{aligned}&\text{Minimize } 5X_1+5X_2+5X_3+10X_4\\ &\text{s.t. } 2X_1 - S_1 = 30\\ &\quad X_2 + 3X_4 - S_2 = 20\\ &\quad X_3 - S_3 = 40\\ &X_k\ge0,\ S_i\ge0\ \ k=1,2,3,4;\ s=1,2,3\end{aligned}$$

which has the optimal solution:

$$\begin{cases} X_1=15,\ X_3=40,\ X_4=\frac{20}{3},\ X_2=S_1=S_2=S_3=0\\ \pi_1 = 2.5,\ \pi_2 = \frac{10}{3},\ \pi_3 = 5,\ \text{cost}= \$341.67\end{cases}$$

Let's try generating another useful pattern by the use of the heuristic method on the 16-ft stock length:

$$\begin{aligned}&\text{Max } 2.5a_1 + \tfrac{10}{3}a_2 + 5a_3\\ &\text{s.t. } 4a_1 + 5a_2 + 7a_3 \le 16,\\ &a_i \ge 0 \text{ \& integer, } i=1,2,3\end{aligned}$$

Sort the required pieces:

$$\frac{\pi_3}{\ell_3} > \frac{\pi_2}{\ell_2} > \frac{\pi_1}{\ell_1},\ \text{i.e., } \frac{5}{7} > \frac{10/3}{5} > \frac{2.5}{4}$$

We fill the 16-ft knapsack with as many of the 7-ft required pieces as possible:

$$a_3 = \left\lfloor \frac{16}{7} \right\rfloor = 2$$

This leaves $16-2\times7 = 2$ feet capacity, which is insufficient for any of the other required lengths. The pattern generated by the heuristic method is therefore $\begin{bmatrix}0\\0\\2\end{bmatrix}$ with reduced cost $10 - 5\times2 = 0$.
i.e., this pattern is not useful in the LP.

If we apply the heuristic to the 14-ft stock length, we again obtain $a_1 = a_2 = 0$, $a_3 =2$, but because the cost of the 14-ft stock length is lower, the reduced cost is $9 - 5\times2 = -1 < 0$.
Therefore, we add the pattern $A^5 = \begin{bmatrix}0\\0\\2\end{bmatrix}$ (with k(5)=2)

$$\begin{aligned}&\text{Minimize } 5X_1 + 5X_2 + 5X_3 + 10X_4 + 9X_5\\ &\text{s.t. } 2X_1 - S_1 = 30\\ &\quad X_2 + 3X_4 - S_2 = 20\\ &\quad X_3 + 2X_5 - S_3 = 40\\ &X_k\ge0,\ S_i\ge0\ \ k=1,2,3,4,5;\ s=1,2,3\end{aligned}$$

LP Solution:

$$\begin{cases} X_1=15,\ X_4=\frac{20}{3},\ X_5=20\\ \pi_1 = 2.5,\ \pi_2 = \frac{10}{3},\ \pi_3 = 4.5,\ \text{cost}= \$321.6\end{cases}$$

The next subproblem is

$$\begin{aligned}&\text{Max } 2.5a_1 + \tfrac{10}{3}a_2 + 5a_3\\ &\text{s.t. } 4a_1 + 5a_2 + 7a_3 \le L\\ &a_i \ge 0 \text{ \& integer, } i=1,2,3\end{aligned}$$

where L is either 9, 14, or 16.

Heuristic pattern generation from 14-ft stock length

*(L=14)*

$$\begin{aligned}&\text{Max } 2.5a_1 + \tfrac{10}{3}a_2 + 5a_3\\ &\text{s.t. } 4a_1 + 5a_2 + 7a_3 \le 14\\ &a_i \ge 0 \text{ \& integer, } i=1,2,3\end{aligned}$$

Sorting the requred lengths:

$$\frac{\pi_2}{\ell_2} > \frac{\pi_3}{\ell_3} > \frac{\pi_1}{\ell_1},\ \text{i.e., } \frac{10/3}{5} > \frac{4.5}{7} > \frac{2.5}{4}$$

We can include 2 of the 5-ft pieces, leaving 14-10=4 feet capacity. The next length to try is 7-ft, but there is insufficient capacity; finally, we include 1 of the 4-ft pieces, exactly filling the knapsack.

We have thus generated the pattern $A^6 = \begin{bmatrix}1\\2\\0\end{bmatrix}$ from the 14-ft stock length. Because the reduced cost of this pattern is

$$C_2 - \sum_i \pi_i a_i = 9 - \left(2.5\times 1 + \frac{10}{3}\times 2 + 4.5\times 0\right) = -0.17 < 0$$

we will add the pattern to the LP:

$$\begin{array}{llllllll}
\textbf{Minimize } 5X_1 + 5X_2 + 5X_3 + 10X_4 + 9X_5 + 9X_6 \\
\textbf{s.t.} \quad 2X_1 & & & & + X_6 - S_1 & & = 30 \\
& X_2 & & + 3X_4 & + 2X_6 & - S_2 & = 20 \\
& & X_3 & + 2X_5 & & - S_3 & = 40 \\
X_k \geq 0,\ k=1,2,3,4,5,6; & & & S_i \geq 0,\ s=1,2,3
\end{array}$$

LP

$$\boxed{\begin{array}{llllllll}
\textbf{Minimize } 5X_1 + 5X_2 + 5X_3 + 10X_4 + 9X_5 + 9X_6 \\
\textbf{s.t.} \quad 2X_1 & & & & + X_6 - S_1 & & = 30 \\
& X_2 & & + 3X_4 & + 2X_6 & - S_2 & = 20 \\
& & X_3 & + 2X_5 & & - S_3 & = 40 \\
X_k \geq 0,\ k=1,2,3,4,5,6; & & & S_i \geq 0,\ s=1,2,3
\end{array}}$$

with solution

$$\begin{cases} X_1=10,\ X_5=20,\ X_6=10 \\ \pi_1 = 2.5,\ \pi_2 = 3.25,\ \pi_3 = 4.5,\ \text{cost}= \$320 \end{cases}$$

Heuristic solution of knapsack problem:

$$\boxed{\begin{array}{r} \textbf{Maximize } 2.5\, a_1 + \frac{39}{12} a_2 + 4.5\, a_3 \\ \textbf{s.t.} \quad 4\, a_1 + 5\, a_2 + 7\, a_3 \leq 14 \\ a_i \geq 0 \ \&\ \text{integer},\ i=1,2,3 \end{array}}$$

Sort the pieces: $\frac{\pi_2}{\ell_2} > \frac{\pi_3}{\ell_3} > \frac{\pi_1}{\ell_1}$, i.e., $\frac{3.25}{5} > \frac{4.5}{7} > \frac{2.5}{4}$

Using 14-ft stock length, we include $a_2$=2 of the 5-ft length, $a_3$=0 of the 7-ft length, and $a_1$=1 of the 4-ft length; ***but this was pattern #6*** (which now has reduced cost 0)

If we apply the heuristic pattern-generating method to the 9-ft stock length, we get the pattern

$A^7 = \begin{bmatrix}1\\1\\0\end{bmatrix}$ with reduced cost

$$C_1 - \sum_i \pi_i a_i = 5 - \left(2.5\times 1 + 3.25\times 1 + 4.5\times 0\right) = -0.75$$

and so we will add this pattern to the LP.

$$\boxed{\begin{array}{llllllll}
\textbf{Minimize } 5X_1 + 5X_2 + 5X_3 + 10X_4 + 9X_5 + 9X_6 + 5X_7 \\
\textbf{s.t.} \quad 2X_1 & & & & + X_6 + X_7 - S_1 & & = 30 \\
& X_2 & & + 3X_4 & + 2X_6 + X_7 & - S_2 & = 20 \\
& & X_3 & + 2X_5 & & - S_3 & = 40 \\
X_k \geq 0,\ k=1,2,3,4,5,6; & & & S_i \geq 0,\ s=1,2,3
\end{array}}$$

LP Solution:

$$\begin{cases} X_1=5,\ X_5=20,\ X_7=20 \\ \pi_1 = 2.5,\ \pi_2 = 2.5,\ \pi_3 = 4.5,\ \text{cost}= \$305 \end{cases}$$

Column-generating subproblem:

$$\boxed{\begin{array}{l} \textbf{Max } 2.5\, a_1 + 2.5\, a_2 + 4.5 a_3 \\ \textbf{s.t.} \quad 4a_1 + 5a_2 + 7a_3 \leq L \\ a_i \geq 0 \ \&\ \text{integer},\ i=1,2,3 \end{array}}$$

Applying the heuristic method to the knapsack problem:

$$\boxed{\begin{array}{l} \textbf{Max } 2.5\, a_1 + 2.5\, a_2 + 4.5 a_3 \\ \textbf{s.t.} \quad 4a_1 + 5a_2 + 7a_3 \leq L \\ a_i \geq 0 \ \&\ \text{integer},\ i=1,2,3 \end{array}}$$

Sorting the req'd pieces:

$\frac{\pi_3}{\ell_3} > \frac{\pi_1}{\ell_1} > \frac{\pi_2}{\ell_2}$, i.e., $\frac{4.5}{7} > \frac{2.5}{4} > \frac{4.5}{5}$

Using L=9, we get $a_3$ =1, $a_1$=0, $a_2$ =0, which is pattern #3 again, now with reduced cost 0

Using L=14, we get $a_3$ =2, $a_1$ = $a_2$ =0 with reduced cost = 0

Using L=16, we get $a_3$ =2, $a_1$=$a_2$=0, which has reduced cost = +1.

The heuristic method is unable to generate a pattern having a negative reduced cost... however, because it doesn't in general optimize the knapsack problem, there may be other patterns which do have negative reduced costs.

Therefore, we must now apply the DP (*dynamic programming*) algorithm.

```
Stock Piece 1 (Length 9)
*** Optimal knapsack value is 5 ***
*** There are 2 optimal patterns from this stock piece ***

   Optimal Pattern # 1 from this stock piece

            i  l  a
            -  -  -
            1  4  2     Scrap remaining: 1
            2  5  0
            3  7  0

   Optimal Pattern # 2 from this stock piece

            i  l  a
            -  -  -
            1  4  1     Scrap remaining: 0
            2  5  1
            3  7  0

 Reduced Cost: 0
```

Stock Piece 2 (Length 14)

*** Optimal knapsack value is 9 ***

| i | l | a |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 5 | 0 |
| 3 | 7 | 2 |

Scrap remaining: 0

Reduced Cost: 0

Stock Piece 3 (Length 16)

*** Optimal knapsack value is 10 ***

| i | l | a |
|---|---|---|
| 1 | 4 | 4 |
| 2 | 5 | 0 |
| 3 | 7 | 0 |

Scrap remaining: 0

Reduced Cost: 0

## Stage 1

| s \ x: | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | -999.99 | -999 | -999.99 | -999 |
| 1 | 0 | -999.99 | -999 | -999.99 | -999 |
| 2 | 0 | -999.99 | -999 | -999.99 | -999 |
| 3 | 0 | -999.99 | -999 | -999.99 | -999 |
| 4 | 0 | 2.50 | -999 | -999.99 | -999 |
| 5 | 0 | 2.50 | -999 | -999.99 | -999 |
| 6 | 0 | 2.50 | -999 | -999.99 | -999 |
| 7 | 0 | 2.50 | -999 | -999.99 | -999 |
| 8 | 0 | 2.50 | 5 | -999.99 | -999 |
| 9 | 0 | 2.50 | 5 | -999.99 | -999 |
| 10 | 0 | 2.50 | 5 | -999.99 | -999 |
| 11 | 0 | 2.50 | 5 | -999.99 | -999 |
| 12 | 0 | 2.50 | 5 | 7.50 | -999 |
| 13 | 0 | 2.50 | 5 | 7.50 | -999 |
| 14 | 0 | 2.50 | 5 | 7.50 | -999 |
| 15 | 0 | 2.50 | 5 | 7.50 | -999 |
| 16 | 0 | 2.50 | 5 | 7.50 | 10 |

## Stage 2

| s \ x: | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0.00 | -999.99 | -999.99 | -999.99 |
| 1 | 0.00 | -999.99 | -999.99 | -999.99 |
| 2 | 0.00 | -999.99 | -999.99 | -999.99 |
| 3 | 0.00 | -999.99 | -999.99 | -999.99 |
| 4 | 2.50 | -999.99 | -999.99 | -999.99 |
| 5 | 2.50 | 2.50 | -999.99 | -999.99 |
| 6 | 2.50 | 2.50 | -999.99 | -999.99 |
| 7 | 2.50 | 2.50 | -999.99 | -999.99 |
| 8 | 5.00 | 2.50 | -999.99 | -999.99 |
| 9 | 5.00 | 5.00 | -999.99 | -999.99 |
| 10 | 5.00 | 5.00 | 5.00 | -999.99 |
| 11 | 5.00 | 5.00 | 5.00 | -999.99 |
| 12 | 7.50 | 5.00 | 5.00 | -999.99 |
| 13 | 7.50 | 7.50 | 5.00 | -999.99 |
| 14 | 7.50 | 7.50 | 7.50 | -999.99 |
| 15 | 7.50 | 7.50 | 7.50 | 7.50 |
| 16 | 10.00 | 7.50 | 7.50 | 7.50 |

## Stage 3

| s \ x: | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0.00 | -999.99 | -999.99 |
| 1 | 0.00 | -999.99 | -999.99 |
| 2 | 0.00 | -999.99 | -999.99 |
| 3 | 0.00 | -999.99 | -999.99 |
| 4 | 2.50 | -999.99 | -999.99 |
| 5 | 2.50 | -999.99 | -999.99 |
| 6 | 2.50 | -999.99 | -999.99 |
| 7 | 2.50 | 4.50 | -999.99 |
| 8 | 5.00 | 4.50 | -999.99 |
| 9 | 5.00 | 4.50 | -999.99 |
| 10 | 5.00 | 4.50 | -999.99 |
| 11 | 5.00 | 7.00 | -999.99 |
| 12 | 7.50 | 7.00 | -999.99 |
| 13 | 7.50 | 7.00 | -999.99 |
| 14 | 7.50 | 7.00 | 9.00 |
| 15 | 7.50 | 9.50 | 9.00 |
| 16 | 10.00 | 9.50 | 9.00 |

## Optimal Solution of LP