Cutting Stock Problems

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

February 2018
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
The cutting stock problem is a prototypical example of a problem that can be attacked using a column generation approach.

We have multiple rolls of a raw material, for example lumber or silk or cellophane. Each original roll has the same length.

We want to slice the rolls up to give many shorter rolls of desired lengths.

We want to use as little raw material as possible while meeting demand.
Illustration for a single roll of raw material

raw material:

![Illustration of raw material with lengths](image)

finished product:

![Illustration of finished product with lengths](image)
Parameters:

- length of raw material: $r$
- desired final lengths: $l_i, i = 1, \ldots, m$
- desired quantities of final lengths: $b_i, i = 1 \ldots, m$
Outline

1. Introduction

2. Integer programming formulation (poor)

3. Column generation formulation (good)

4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
We can formulate the problem as an integer program, but this is a poor formulation. Introduce variables:

\[ z_j = \begin{cases} 
1 & \text{if use raw material roll } j \\
0 & \text{otherwise}
\end{cases} \]

\[ x_{ij} = \text{number of finals of length } l_i \text{ produced from raw } j \]

where \( j = 1, \ldots, R \) with \( R \) equal to an upper bound on the number of raw rolls needed, and \( i = 1, \ldots, m \).
IP formulation (poor)

The formulation is then:

\[
\min_{x,z} \quad \sum_{j=1}^{R} z_j \\
\text{subject to} \quad \sum_{i=1}^{m} l_i x_{ij} \leq r z_j \quad \text{for } j = 1, \ldots, R \\
\sum_{j=1}^{R} x_{ij} \geq b_i \quad \text{for demands } i = 1, \ldots, m \\
x_{ij} \geq 0, \text{ integer} \quad \text{for } i = 1, \ldots, m, j = 1, \ldots, R \\
z_j \quad \text{binary} \quad \text{for } j = 1, \ldots, R
\]

This can be solved using branch-and-bound. In practice it is a bad formulation because:

- **Symmetry**: As soon as we fix the \( x_{ij} \) values for one roll, the old fractional values can move to another roll.
- **Weak relaxation**: It is too easy to get fractional values for \( z_j \), because we throw something away from most rolls.
The formulation is then:

\[
\begin{align*}
\min_{x,z} & \quad \sum_{j=1}^{R} z_j \\
\text{subject to} & \quad \sum_{i=1}^{m} l_i x_{ij} \leq r z_j \quad \text{for } j = 1, \ldots, R \\
& \quad \sum_{j=1}^{R} x_{ij} \geq b_i \quad \text{for demands } i = 1, \ldots, m \\
& \quad x_{ij} \geq 0, \text{ integer} \quad \text{for } i = 1, \ldots, m, j = 1, \ldots, R \\
& \quad z_j \quad \text{binary} \quad \text{for } j = 1, \ldots, R
\end{align*}
\]

This can be solved using branch-and-bound. In practice it is a **bad formulation** because:

- **Symmetry**: As soon as we fix the \(x_{ij}\) values for one roll, the old fractional values can move to another roll.
- **Weak relaxation**: It is too easy to get fractional values for \(z_j\), because we throw something away from most rolls.
Outline

1. Introduction

2. Integer programming formulation (poor)

3. Column generation formulation (good)

4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
We work with **cutting patterns**. For example, a raw roll of length $r = 50$ can be cut into many different patterns when the desired lengths are 14, 18, and 21:

<table>
<thead>
<tr>
<th>pattern</th>
<th>waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,0,0)</td>
<td>8</td>
</tr>
<tr>
<td>(0,2,0)</td>
<td>14</td>
</tr>
<tr>
<td>(0,0,2)</td>
<td>8</td>
</tr>
<tr>
<td>(1,2,0)</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

...
IP cutting plane formulation

Let $a^j$ denote the $j$th cutting pattern, $j = 1, 2, \ldots, P$. We have a very large number of possible patterns, so generate patterns as needed.

Let $x_j$ denote the number of raw rolls cut into pattern $a^j$.

The formulation is:

$$
\text{min}_x \quad \sum_{j=1}^p x_j \\
\text{subject to} \quad \sum_{j=1}^p a^j x_j \geq b \quad \quad (CSP) \\
x_j \geq 0, \text{ integer, } j = 1, \ldots, p
$$

We solve the LP relaxation and then round at the end to get a feasible integer solution.
Outline

1. Introduction

2. Integer programming formulation (poor)

3. Column generation formulation (good)

4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Initial LP approximation

Let $r = 50$, $l_1 = 14$, $l_2 = 18$, $l_3 = 21$, $b_1 = 10$, $b_2 = 15$, $b_3 = 8$. Initialize with the three valid patterns:

$$a^1 = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}, \quad a^2 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \quad a^3 = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}.$$

After introducing slack variables, the initial LP is

$$\begin{align*}
\min_{x,s} & \quad x_1 + x_2 + x_3 \\
\text{subject to} & \quad 3x_1 + x_2 - s_1 = 10 \\
& \quad 2x_2 - s_2 = 15 \\
& \quad 2x_3 - s_3 = 8 \\
& \quad x \geq 0, \text{ integer}, \\
& \quad s \geq 0
\end{align*}$$

We take the three $x$ variables as the initial basic variables. Solving $Bx_B = b$ gives

$$x = \left(\frac{5}{6}, 7.5, 4\right).$$
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Find dual variables

We find the dual variables using complementary slackness, so we solve $B^T y = c_B$:

\[
\begin{align*}
3y_1 + y_1 + 2y_2 &= 1 \\
2y_3 &= 1
\end{align*}
\]

Thus we get

\[
\bar{y} = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{2} \right).
\]
The full dual problem

The general form of the dual problem is

$$\begin{align*}
\max_y & \quad b^T y \\
\text{subject to} & \quad a^T y \leq 1 \quad \text{for all patterns } a \\
& \quad y \geq 0
\end{align*}$$

primal: \quad \min \sum_j x_j

s.t. \quad \sum_j a_{ij} x_j \geq b_i \quad \forall i

\quad x_i \geq 0$$
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Subproblem

We need to determine if $\bar{y}$ is dual feasible. Thus we want to determine:

Is there a pattern $a$ with $a^T \bar{y} > 1$?

Note that the variable in this question is the pattern $a$. The dual variable $\bar{y}$ is fixed in the subproblem. We can write the subproblem mathematically as:

$$
\max_a \quad \bar{y}^T a
\text{subject to} \quad \sum_{i=1}^{m} l_i a_i \leq r \quad \forall i \geq 0, \text{integer for } i = 1, \ldots, m
$$

This is a knapsack problem.
An improving pattern

The knapsack is an NP-Complete problem. We don’t necessarily need to solve it to optimality: we only need to find a feasible solution with value strictly greater than one. For the example, the subproblem is

\[
\begin{align*}
\text{max}_a & \quad \frac{1}{3}a_1 + \frac{1}{3}a_2 + \frac{1}{2}a_3 \\
\text{subject to} & \quad 14a_1 + 18a_2 + 21a_3 \leq 50 \\
& \quad a_i \geq 0, \text{ integer}
\end{align*}
\]

One possible pattern is

\[
a^4 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix},
\]

which has value \(\frac{7}{6} > 1\) in the subproblem. Thus, \(\bar{y}\) is not dual feasible, and we can add the pattern to the primal problem (CSP).
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Updated LP approximation

The updated primal problem (CSP) is

\[
\begin{align*}
\min_{x,s} & \quad x_1 + x_2 + x_3 + x_4 \\
\text{subject to} & \quad 3x_1 + x_2 + 2x_4 - s_1 = 10 \\
& \quad 2x_2 - s_2 = 15 \\
& \quad 2x_3 + x_4 - s_3 = 8 \\
& \quad x \geq 0, \text{ integer, } s \geq 0
\end{align*}
\]

The incoming basic variable is \( x_4 \).
Determine pattern that leaves basis

We determine which variable leaves the basis using the minimum ratio test, so we need to calculate $d = B^{-1}a^4$, or solve the system

$$
\begin{align*}
3d_1 + d_2 &= 2 \\
2d_2 &= 0 \\
2d_3 &= 1
\end{align*}
\implies d = \begin{bmatrix} \frac{2}{3} \\ 0 \\ \frac{1}{2} \end{bmatrix}.
$$
Minimum ratio test

The minimum ratio test is then comparing

\[ d = \begin{bmatrix} \frac{2}{3} \\ 0 \\ \frac{1}{2} \end{bmatrix}, \quad B^{-1}b = \begin{bmatrix} \frac{5}{6} \\ 7.5 \\ 4 \end{bmatrix}, \quad \text{so ratios are} \quad \begin{bmatrix} \frac{5}{4} \\ - \\ 8 \end{bmatrix} \]

so pattern \( a^1 \) leaves the basis, after a step of length \( \frac{5}{4} \).
Updated basic feasible solution

The updated basis matrix and bfs are

\[
B = \begin{bmatrix}
2 & 1 & 0 \\
0 & 2 & 0 \\
1 & 0 & 2 \\
\end{bmatrix}, \quad B^{-1}b = \begin{bmatrix}
\frac{5}{4} \\
7.5 \\
\frac{27}{8} \\
\end{bmatrix} = \begin{bmatrix}
x_4 \\
x_2 \\
x_3 \\
\end{bmatrix}. 
\]
Updated dual solution

We find the dual variables using complementary slackness, so we solve $B^T y = c_B$:

\[
\begin{align*}
2y_1 + y_3 &= 1 \\
y_1 + 2y_2 &= 1 \\
2y_3 &= 1
\end{align*}
\]

\[\implies \bar{y} = \begin{bmatrix}
\frac{1}{4} \\
\frac{3}{8} \\
\frac{1}{2}
\end{bmatrix}.\]
Outline

1. Introduction
2. Integer programming formulation (poor)
3. Column generation formulation (good)
4. Example
   - Initial LP
   - Dual problem
   - Subproblem to determine dual feasibility
   - Second iteration
   - Subproblem in second iteration
Subproblem in second iteration

The subproblem to determine whether $\bar{y}$ is dual feasible is then

$$\max_a \quad \frac{1}{4}a_1 + \frac{3}{8}a_2 + \frac{1}{2}a_3$$
subject to $14a_1 + 18a_2 + 21a_3 \leq 50$
$$a_i \geq 0, \text{ integer}$$

All feasible patterns have value no larger than one in the subproblem.

Thus, the solution $\bar{y}$ is feasible in the dual (CSD) to the cutting stock problem, so we’ve solved the (LP relaxation of) the cutting stock problem (CSP).