Branch and Cut algorithms for Combinatorial Optimization Problems

Avijit Sarkar
Branch & Cut algorithms modify the basic Branch & Bound strategy by attempting to strengthen the linear programming relaxation (LPR) of an IP with new inequalities before branching a partial solution.

- Basically, Branch & Cut = Branch & Bound + Cutting Planes
- Pure Branch & Bound can be considerably sped up by employing cutting planes either at the top of a Branch & Bound tree or at every node of the tree, because cutting planes considerably reduce the size of the tree.
- Branch & Cut can be used in conjunction with heuristics to obtain a lower bound on the optimal value, using the Branch & Bound algorithm.
Consider the following IP:

P0: \[ \min z = -6x_1 - 5x_2 \]
subject to \[ 3x_1 + x_2 \leq 11 \]
\[ -x_1 + 2x_2 \leq 5 \]
\[ x_1, x_2 \geq 0, \text{ integer} \] (refer to fig 1 here)

- Ignoring integrality constraints, we solve the LPR.

- Optimal solution is \((2 \frac{3}{7}, 3 \frac{5}{7})\), \(z_{opt} = -33 \frac{1}{7}\)
Sub-problems generated by branching on \( x_1 \)

**P1:**

\[
\begin{align*}
\text{min } z &= -6x_1 - 5x_2 \\
\text{subject to} \quad &3x_1 + x_2 \leq 11 \\
&-x_1 + 2x_2 \leq 5 \\
&x_1 \geq 3 \\
&x_1, x_2 \geq 0, \text{ integer}
\end{align*}
\]

Optimal Solution of \( \text{LPR}_{P1} \)

\((3,2), \ z_{\text{opt}} = -28\)

**P2:**

\[
\begin{align*}
\text{min } z &= -6x_1 - 5x_2 \\
\text{subject to} \quad &3x_1 + x_2 \leq 11 \\
&-x_1 + 2x_2 \leq 5 \\
&x_1 \leq 2 \\
&x_1, x_2 \geq 0, \text{ integer}
\end{align*}
\]

Optimal Solution of \( \text{LPR}_{P2} \)

\((2,3.5), \ z_{\text{opt}} = -29.5\)

(refer to fig 2 here)

(refer to fig 3 here)
Add a cut to P2

P3: \( \min z = -6x_1 - 5x_2 \)
subject to \( 3x_1 + x_2 \leq 11 \) \hspace{1cm} \text{Optimal Solution of LPR}_{P3}
\( -x_1 + 2x_2 \leq 5 \) \hspace{1cm} (1.8,3.4), \( z_{opt} = -27.8 \)
\( x_1 \leq 2 \)
\( 2x_1 + x_2 \leq 7 \) \textit{(added cut)}
\( x_1 , x_2 \geq 0, \text{integer} \) \hspace{1cm} \text{(refer to fig 4 here)}

Questions that arise :

➢ Is the added inequality a valid inequality?
➢ How to generate a valid inequality?
➢ Whether to branch or to cut?
Progress of Branch & Cut on a 2D IP

PROBLEM P0
SOLN TO RELAXATION: \((2^{\frac{3}{7}}, 3^{\frac{5}{7}}), z_{opt} = -33\)

Branch on \(x_1\)
- \(x_1 \geq 3\)
- \(x_1 \leq 2\)

Problem P1
SOLN TO RELAXATION
\((3,2), z_{opt} = -28\)

Problem P2
SOLN TO RELAXATION
\((2,3.5), z_{opt} = -29.5\)

Add cut \(2x_1 + x_2 \leq 7\)

Problem P3
SOLN TO RELAXATION
\((1.8,3.4), z = -27.8\)
Branch & Cut algorithm

1. **Initialization**: Denote the initial IP problem by $ILP^0$ & set the active nodes to be $L=\{ILP^0\}$. Set the upper bound to be $\overline{z}=+\infty$. Select one problem $l \in L$ and set its lower-bound on the optimal value, $\underline{z}_l=-\infty$.

2. **Termination**: If $L=\emptyset$, then the solution $x'$ which yielded the incumbent objective value $z$ is optimal. If no such $x'$ exists, i.e. $z=+\infty$ the ILP is infeasible.

3. **Problem selection**: Select and delete a problem $ILP^l$ from $L$.

4. **Relaxation**: Solve the LPR of $ILP^l$. If the relaxation is infeasible, set $\underline{z}_l=+\infty$ and go to step 6. Let $\overline{z}_l$ denote the optimal objective value of the relaxation, if it is finite and let $x^{IR}$ be an optimal solution; otherwise, set $\underline{z}_l=-\infty$.

5. **Add cutting planes**: If desired, search for cutting planes, if any are found, add them to the relaxation and return to step 4.

6. **Fathoming & Pruning**:
   (a) If $\overline{z}_l \geq \overline{z}$ go to step 2
   (b) If $\underline{z}_l < \overline{z}$ and $x^{IR}$ is integral feasible, update $z=\underline{z}_l$, delete from $L$ all problems with $\underline{z}_l \geq z$, and go to step 2.

7. **Partitioning**: Let $\left\{ S^{lj} \right\}_{j=1}^{k}$ be a partition of the constraint set $S^l$ of problem $ILP^l$. Add problems $\left\{ ILP^{lj} \right\}_{j=1}^{k}$ to $L$, where $ILP^{lj}$ is $ILP^l$ with feasible region restricted to $S^{lj}$ and $\underline{z}_{lj}$ for $j=1, \ldots, k$ is set to the value of $\underline{z}_l$ for the parent problem $l$. Go to step 2.
Generating Cutting Planes

- Take a weighted combination of the inequalities from the current LPR.
- Exploit the fact that variables must be integral, process known as integer rounding.
- Cutting planes generated in this way are called Chvatal-Gomory cutting planes.
- Example:

\[
\frac{1}{6} (3x_1 + x_2 \leq 11) + \frac{5}{12} (-x_1 + 2x_2 \leq 5)
\]

gives

\[
\frac{1}{12} x_1 + x_2 \leq 3 \frac{11}{12}
\]

LHS of the inequality is rounded down, which gives

\[
x_2 \leq 3 \frac{11}{12}
\]

In any feasible solution to an IP, the LHS must take an integer value, so the RHS is rounded down.

Finally we have the valid inequality: \( x_2 \leq 3 \)
When to Generate Cuts?

- If there is a situation, in which the cutting plane loop in steps 4 & 5 tails off, i.e. the solution to a current LPR is not much better than the solutions to recent previous LPRs, then a node can be fathomed.
A Generalized Assignment Problem with SOS2

- Problem uses Special Ordered Sets of Type 2.
- Each set has 20 variables.
- 120 – 220 constraints, excluding non-negativity.
- Problem solved using:
  - (A) IBM RS6000/590
  - (B) MINTO 3.0 as branch-and-bound algorithm
  - (C) CPLEX 6.0 as LP solver
- Branching tree limited to 50,000 nodes
Comparison of CPU time (secs), using B&B and B&C (Defarias et al.)

<table>
<thead>
<tr>
<th>SETS</th>
<th>CONS</th>
<th>VARS</th>
<th>TIME (NO CUTS)</th>
<th>TIME (W/CUTS)</th>
<th>FIRST SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>120</td>
<td>2000</td>
<td>2833</td>
<td>90</td>
<td>2590</td>
</tr>
<tr>
<td>110</td>
<td>130</td>
<td>2200</td>
<td>3312</td>
<td>2290</td>
<td>2999</td>
</tr>
<tr>
<td>120</td>
<td>140</td>
<td>2400</td>
<td>2397</td>
<td>83</td>
<td>2241</td>
</tr>
<tr>
<td>130</td>
<td>150</td>
<td>2600</td>
<td>2311</td>
<td>104</td>
<td>2204</td>
</tr>
<tr>
<td>140</td>
<td>160</td>
<td>2800</td>
<td>2314</td>
<td>227</td>
<td>2278</td>
</tr>
<tr>
<td>150</td>
<td>170</td>
<td>3000</td>
<td>956</td>
<td>73</td>
<td>457</td>
</tr>
<tr>
<td>160</td>
<td>180</td>
<td>3200</td>
<td>2998</td>
<td>731</td>
<td>2913</td>
</tr>
<tr>
<td>170</td>
<td>190</td>
<td>3400</td>
<td>1967</td>
<td>158</td>
<td>1561</td>
</tr>
<tr>
<td>200</td>
<td>220</td>
<td>4000</td>
<td>1310</td>
<td>43</td>
<td>982</td>
</tr>
</tbody>
</table>
Comparison of nodes evaluated with & without cuts

<table>
<thead>
<tr>
<th>SETS</th>
<th>NODES(NO CUTS)</th>
<th>NODES(W/CUTS)</th>
<th>CUTS</th>
<th>NODES(NO CUTS)</th>
<th>NODE(W/CUTS)</th>
<th>CUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>38967</td>
<td>881</td>
<td>138</td>
<td>32351</td>
<td>456</td>
<td>135</td>
</tr>
<tr>
<td>110</td>
<td>33933</td>
<td>15429</td>
<td>641</td>
<td>31251</td>
<td>7151</td>
<td>510</td>
</tr>
<tr>
<td>120</td>
<td>30523</td>
<td>1067</td>
<td>130</td>
<td>26662</td>
<td>898</td>
<td>101</td>
</tr>
<tr>
<td>130</td>
<td>27679</td>
<td>1147</td>
<td>151</td>
<td>25094</td>
<td>654</td>
<td>121</td>
</tr>
<tr>
<td>140</td>
<td>24191</td>
<td>1735</td>
<td>143</td>
<td>23674</td>
<td>1499</td>
<td>96</td>
</tr>
<tr>
<td>150</td>
<td>18180</td>
<td>792</td>
<td>117</td>
<td>8243</td>
<td>421</td>
<td>110</td>
</tr>
<tr>
<td>160</td>
<td>29708</td>
<td>4645</td>
<td>201</td>
<td>28020</td>
<td>3239</td>
<td>118</td>
</tr>
<tr>
<td>170</td>
<td>21707</td>
<td>1462</td>
<td>138</td>
<td>20532</td>
<td>900</td>
<td>95</td>
</tr>
<tr>
<td>200</td>
<td>14260</td>
<td>344</td>
<td>107</td>
<td>10437</td>
<td>276</td>
<td>103</td>
</tr>
</tbody>
</table>
Figure 1:

\[ 3x_1 + x_2 \leq 11 \]

\[-x_1 + 2x_2 \leq 5 \]

\[ x_1 \geq 3 \]

\[ 3x_1 + x_2 = 11 \]

Figure 2:
Figure 3:

Figure 4: