Branch and Cut Algorithm

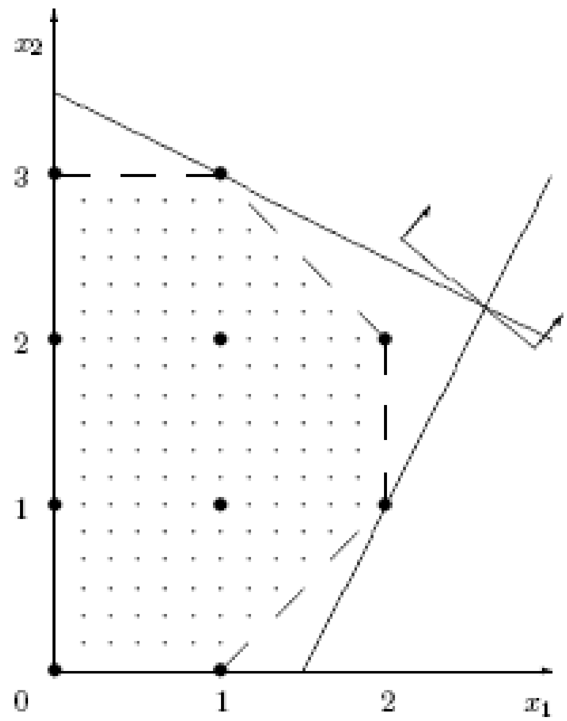
It is the combination of the Cutting Plane and Branch and Bound Algorithms. We show the principle below through an example.

A simple example.

Consider the integer programming problem

$$
\begin{array}{lrcrcl}
\min & -5x_1 & - & 6x_2 & & \\
\text{s.t.} & x_1 & + & 2x_2 & \leq & 7 \\
 & 2x_1 & - & x_2 & \leq & 3 \\
 & & & x_1, x_2 & \geq & 0, \text{ integer.}
\end{array}
$$

The first step in a branch-and-cut approach is to solve the linear programming relaxation, which gives the point $(2.6, 2.2)$, with value -26.2. There is now a choice: should the LP relaxation be improved by adding a cutting plane, for example, $x_1 + x_2 \leq 4$, or should the problem be divided into two by splitting on a variable?



A branch-and-cut example

Assume the algorithm makes the second choice, and further assume that the decision is to split on x_2, giving two new problems:

$$\begin{array}{llcrcl}
\text{min} & -5x_1 & - & 6x_2 & & \\
\text{s.t.} & x_1 & + & 2x_2 & \leq & 7 \\
 & 2x_1 & - & x_2 & \leq & 3 \\
 & & & \mathbf{x_2} & \geq & 3 \\
 & & & x_1, x_2 & \geq & 0, \text{ integer}
\end{array}$$

and

$$\begin{array}{llcrcl}
\text{min} & -5x_1 & - & 6x_2 & & \\
\text{s.t.} & x_1 & + & 2x_2 & \leq & 7 \\
 & 2x_1 & - & x_2 & \leq & 3 \\
 & & & \mathbf{x_2} & \leq & 2 \\
 & & & x_1, x_2 & \geq & 0, \text{ integer.}
\end{array}$$

The optimal solution to the original problem will be the better of the solutions to these two subproblems. The solution to the linear programming relaxation of the first subproblem is $(1, 3)$, with value -23. Since this solution is integral, it solves the first subproblem. This solution becomes the incumbent best known feasible solution. The optimal solution for the linear programming relaxation of the second subproblem is $(2.5, 2)$, with value -24.5. Since this point is nonintegral, it does not solve the subproblem. Therefore, the second subproblem must be attacked further.

We can then recursively repeat the algorithm for the subproblem.