A Cutting Plane Algorithm for Integer Programs With an Easy Proof of Convergence

Bell, D.E.

IIASA Working Paper

WP-73-015

1973
1. **Introduction**

Let $B$ be the optimal LP basis for a given problem with $m$ rows and $n+m$ variables.

\[
\begin{align*}
\min & \quad c_B^T y + c_N^T x \\
\text{s.t.} & \quad B y + N x = b \\
& \quad y, x > 0 \text{ integer,}
\end{align*}
\]

where $B^{-1} b \geq 0$, $c_N \geq c_B B^{-1} N$, and all coefficients are assumed to be integral.

**Lemma 1** If $c_B B^{-1} b$ is not integral, the constraint

\[
 c_B y + \lfloor c_B B^{-1} N \rfloor x \geq \lfloor c_B B^{-1} b \rfloor
\]

is a valid cut for the I.P., where $\lfloor t \rfloor$ is the lowest integer not less than $t$.

**Proof** Since $B$ is optimal

\[
 c_B y + c_N^T x \geq c_B B^{-1} b
\]

for all feasible $y$, $x$, and for any value of $c_N$ satisfying $c_N \geq c_B B^{-1} b$. In particular

\[
 c_B y + \lfloor c_B B^{-1} N \rfloor x \geq c_B B^{-1} b
\]
Since $c_B$, $[c_B^{-1}N]$ are integral, for all feasible integral values of $(y, x)$

$$c_B y + [c_B^{-1}N] x \geq [c_B^{-1}b]$$

which is therefore a valid cut.

**Lemma 2** If the cut of Lemma 1 is added to the LP, the optimal objective value increases to at least $[c_B^{-1}b]$.  

**Proof** Let $(y^*, x^*)$ be the optimal solution to the new LP, then

$$c_B y^* + c_N x^* \geq c_B y^* + [c_B^{-1}N] x^* \geq [c_B^{-1}b]$$

since $c_N \geq [c_B^{-1}N]$ because $c_N$ is integral and because of the new cut.

//
2. The Algorithm

Step 1  Solve the L.P.

Step 2a  If the value of the objective is not integral, add the cut of lemma 1 and return to step 1.

Step 2b  If the value of the objective is integral, create a subproblem with added constraint

\[ c_B y + c_N x = c_B B^{-1} b \]

Togeter with a new objective function \((d_B, d_N)\) chosen
only to be independent of the existing m+1 rows. Implement this algorithm on the subproblem. If the subproblem has a feasible solution, it is optimal. If it has no feasible solution, add the cut

\[ c_B^y + c_N^r \geq c_B^{B^{-1}} + 1 \]

to the original L.P. and go to step 1.
3. **Convergence**

**Theorem** The algorithm of section 2 produces the optimal solution, or shows there is none, after solving only a finite number of L.P. problems, if the L.P. feasible region is bounded.

**Proof** By induction on \( n \) the number of non basic variables.

For \( n = 0 \) the algorithm produces the solution or the information that no solution exists after solving at most two L.P.'s.

Assume that the algorithm converges for all programs having up to \( n \) non basic variables and now consider a problem having \( n + 1 \).

If the subproblem is created, it has only \( n \) non basic variables and hence can be solved finitely so that each repetition of step 1 occurs after a finite number of L.P. solutions. Note that the objective value increases by at least 1 every two iterations of step 1. If the L.P. region is bounded, the algorithm must converge finitely.