Improved Gomory Cuts for Primal Cutting Plane Algorithms

S. Dey J-P. Richard

Industrial Engineering
Purdue University

INFORMS, 2005

Outline

1. Motivation
   - The Basic Idea
   - Set up the Lifting Problem
   - How to Solve this Lifting Problem
2. Primal Algorithm
   - Glover's Algorithm
   - Implementation
3. Results
   - Some Computational Results
   - Summary

Information From More than One Constraint?

Most cutting planes are generated using one constraint or are based on specific structures.

Examples of cuts based on single constraint:

- Gomory's Fractional Cut
- Cover Cuts

Example of cuts based on specific structure:

- Cuts based on Variable upper bound
- Cuts based on Generalized upper bounds
- Cuts based on mixing polytopes

Information From More than One Constraint?

Some examples of cuts generation based on multiple rows.

- Johnson, E. 1974. On the group problem with for mixed integer programming. *Math. Programming Study* 2.137 - 179.
- Koppe, M., Weismantel, R. 2002. Cutting Plane from a mixed interval farkas Lemma. *Operations Research Letters* 32. 207 -211.
- Marchand, H., Wolsey, L.A. 2001. Aggregation and mixed integer rounding to Solve MIPS. *Operations Research*.49. 363-371.
- Martin, A., Weismantel, R. 1998. The intersection of knapsack polyhedron and extensions. *Lecture Notes in Computer Science*1412, 243-256.

Assumptions

- The feasible region is:

$$\sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\} \tag{1}$$

$$x_j \in \mathbb{Z}_+ \quad \forall j \in N \tag{2}$$

- We already have a cut:

$$\sum_{j \in N} \alpha_j x_j \leq r \tag{3}$$

$$\alpha_j \in \mathbb{Z} \quad \forall j \in N$$

$$r \in \mathbb{Z} \tag{4}$$

- The feasible region is bounded.
- There exists a valid point with $x_k = 0$ for the LP relaxation.

Improving Coefficients of a Cut

We are looking for the best value of β:

$$\sum_{j \in N \setminus \{k\}} \alpha_j x_j + \beta x_k \leq r$$

which is valid for

$$\sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\}$$

$$x_j \in \mathbb{Z}_+ \quad \forall j \in N$$

Improving Coefficients of a Cut

The value can be determined as:

$$\beta = \quad min_{x_k \in \mathbb{Z}, x_k \geq 1} \quad \frac{r - \sum_{j \in N \setminus \{k\}} \alpha_j x_j}{x_k}$$

$$s.t. \quad \sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\}$$

$$x_j \in \mathbb{Z}_+ \quad \forall j \in N$$

This is difficult to solve!

A More Tractable Problem

We relax the feasible region:

$$\bar{\beta} = \quad min_{x_k \in \mathbb{Z}, x_k \geq 1} \quad \frac{r - \sum_{j \in N \setminus \{k\}} \alpha_j x_j}{x_k}$$

$$s.t. \qquad \sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\}$$

$$x_j \geq 0 \quad \forall j \in N$$

A little Improvement to the Relaxation

The approximate $\bar{\beta}$ can improved as:

$$\widetilde{\beta} = \quad min_{x_k \in \mathbb{Z}, x_k \geq 1} \quad \frac{\lceil r - \sum_{j \in N \setminus \{k\}} \alpha_j x_j \rceil}{x_k}$$

$$s.t. \qquad \sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\}$$

$$x_j \geq 0 \quad \forall j \in N$$

The Lifting Problem

There is one LP for each integer value of x_k.

$$\begin{aligned} \widetilde{\beta_\lambda} = \quad & min \quad \frac{\lceil r - \sum_{j \in N \setminus \{k\}} \alpha_j x_j \rceil}{\lambda} \\ & s.t. \quad \sum_{j \in N} a_{ij} x_j = b_i \quad i \in T = \{1, 2, 3, ..., m\} \\ & \quad x_j \geq 0 \quad \forall j \in N \\ & \quad x_k = \lambda \end{aligned} \tag{5}$$

Observation : $\widetilde{\beta} = min_{\lambda \in \mathbb{Z}, \lambda \geq 1} \widetilde{\beta_\lambda}$

Is it possible to reduce the number of LPs to be solved?

Definition

- Variant of the Value Function :

$$\begin{aligned} \Gamma(\lambda) &= r + min\left(-\sum_{j\in N\setminus\{k\}} \alpha_j x_j\right) \\ s.t. & \sum_{j\in N\setminus\{k\}} a_{ij}x_i + a_{ik}\lambda = b_i \quad \forall i \in T \\ & x_j \geq 0 \quad \forall j \in N \setminus \{k\} \end{aligned} \tag{6}$$

Whenever the problem is feasible.

- Observe : $\widetilde{\beta_\lambda} = \frac{\lceil\Gamma(\lambda)\rceil}{\lambda}$
- Define $\mu(\lambda)$ to be the optimal dual variables corresponding to the minimization problem in $\Gamma(\lambda)$.

Some Results

- **Proposition 1:** The function Γ is piecewise linear and convex over the domain of λ, and $-\mu(\lambda)^T a_k$ is a subgradient for the function Γ at λ.
- **Lemma 2:** If $\Gamma(0) \leq 0$, then $\widetilde{\beta_j} \geq \widetilde{\beta_1} - 1$.
- **Lemma 3:** Assume that $\Gamma(1)$ exists. If $\mu(1)^T a_k \leq -\lceil \Gamma(1) \rceil$, then $\widetilde{\beta_j} \geq \widetilde{\beta_1}$.

$\widetilde{\beta}$ can be approximated by solving 1 LP

Theorem 4: If $\Gamma(0) \leq 0$, then, $\widetilde{\beta} \geq \widetilde{\widetilde{\beta}}$, where

$$\widetilde{\widetilde{\beta}} = \begin{cases} \widetilde{\beta_1} & \text{if} \quad \Gamma(1) \text{ exists and } \mu(1)^T a_k \leq -\lceil \Gamma(1) \rceil \\ \widetilde{\beta_1} - 1 & \text{if} \quad \Gamma(1) \text{ exists and} \mu(1)^T a_k > -\lceil \Gamma(1) \rceil \\ 0 & \text{if} \quad \Gamma(1) \text{ does not exist} \end{cases} \tag{7}$$

13

Improving Gomory's Fractional cut

For a tableau row: $x_{B_u} + \sum_{j \in NB} \bar{a}_{uj} x_j = \bar{b}_u$,
Gomory's Fractional cut is: $x_{B_u} + \sum_{j \in NB} \lfloor \bar{a}_{uj} \rfloor x_j \leq \lfloor \bar{b}_u \rfloor$.

- All the coefficients are integer in the cut.
- We can fix a non-basic variable to zero.
- Since the cut is guaranteed to cut off the fractional point, $\Gamma(0) < 0$.
- Note : Not working with GMIC, because no known primal algorithm which converges using GMIC.

Primal Algorithm using Cutting Planes

- Ben-Israel, A., Charnes A. 1962. On some problems of diophantine programming. *Cahiers du Centre d'Etudes de Recherche Operationelle*. 4, 215 - 280.
- Young, R.D. 1965. A primal (all-integer) integer programming algorithm. *J. of Res. of the National Bureau of Standards* 69B, 213-250.
- Young, R.D. 1968. A simplified primal(all-integer) integer programming agorithm. *Oper. Res.* 16, 750 - 782.
- Glover, F. 1968. A new foundation for a simplified primal integer programming algorithm. *Oper. Res.*16, 724- 740.
- Sharma, S., Sharma, B., 1997. New technique for solving primal all-integer linear programming. *Opsearch* 34. 62 - 68.
- Lechford, A., Lodi, A. 2002. *Math. Methods of Oper. Res.* 56(1), 67-81.

Glover's algorithm

- Initialization:
  1. Start with an existing integer solution with an integer dictionary.
  2. A special row is added.
     - Different types of special rows can be added. The one used is $\sum_{i \in NB} x_i \leq M$. M was obtained by adding the bounds for each of the nonbasic variables.
- Entering Column Selection: Select the entering column based on a lexicographical ordering rule.

Glover's Algorithm - Contd.

- Pivoting: If the pivot element is 1 a normal simplex pivot is done. If the pivot element is greater than or equal to 2, a cut is added.
- Cut Generation
  1. The row for cut generation is selected by a specific rule from Glover's algorithm.
  2. The Gomory's Fractional cut is generated. We attempt to improve this cut.
  3. After adding the cut (improved or otherwise) and pivoting if the new basic variable is a slack variable of some previous added cut, the pivot row is dropped.
- Optimality Check : If the reduced cost are of the right sign stop else go to step 3.

Convergence Result

Theorem 5: The variant of Glover's algorithm with improved cuts is convergent.

Details on Cut Improvement

1. First the Gomory's fractional cut is generated.
2. Temporary pivoting is done after adding the Gomory's cut. The new reduced costs are found.
3. If the reduced cost is negative then it is divided by the entry in the special row.
4. The resulting vector of reduced costs is sorted in an increasing order. We attempt to improve the coefficients in the above order.
5. This improving process involves finding $\Gamma(1)$ by solving the Linear program for (6). Then the new coefficient $\tilde{\tilde{\beta}}$ is found according to Theorem 4. If $\tilde{\tilde{\beta}}$ is greater than the value in the Gomory's cut it is replaced.

One method to reduce computations of LPs corresponding to each variable

- Solved:

$$\min \sum_{j \in N \setminus \{k_1\}} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N \setminus \{k_1\}} a_j x_j = b - a_{k_1}$$

- Primal Simplex

$$\min \sum_{j \in N} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N} a_j x_j = b - a_{k_1}$$

- Dual Simplex

$$\min \sum_{j \in N \setminus \{k_2\}} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N \setminus \{k_2\}} a_j x_j = b - a_{k_2}$$

One method to reduce computations of LPs corresponding to each variable

- Solved:

$$\min \sum_{j \in N \setminus \{k_1\}} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N \setminus \{k_1\}} a_j x_j = b - a_{k_1}$$

- Primal Simplex

$$\min \sum_{j \in N} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N} a_j x_j = b - a_{k_1}$$

- Dual Simplex

$$\min \sum_{j \in N \setminus \{k_2\}} -\alpha_j x_j \qquad \text{s.t.} \sum_{j \in N \setminus \{k_2\}} a_j x_j = b - a_{k_2}$$

One method to reduce computations of LPs corresponding to each variable

- Solved:

$$\min \sum_{j \in N \setminus \{k_1\}} -\alpha_j x_j \quad \text{s.t.} \sum_{j \in N \setminus \{k_1\}} a_j x_j = b - a_{k_1}$$

- Primal Simplex

$$\min \sum_{j \in N} -\alpha_j x_j \quad \text{s.t.} \sum_{j \in N} a_j x_j = b - a_{k_1}$$

- Dual Simplex

$$\min \sum_{j \in N \setminus \{k_2\}} -\alpha_j x_j \quad \text{s.t.} \sum_{j \in N \setminus \{k_2\}} a_j x_j = b - a_{k_2}$$

Computational Experiment - Problem Instances

Random Problems based on Lechford and Lodi [1].

- $n \in \{5; 10; 15; 20; 25\}$
- $m \in \{5; 10\}$
- Five random instances making 50 total instances.
- The objective function coefficients are integers generated uniformly between 1 and 10.
- For the instances with $m = 5$, the left-hand side coefficients are also integers generated uniformly between 1 and 10.
- For the instances with $m = 10$ the matrix is 50% dense.
- In all cases the right-hand side of each constraint was set to half the sum of the left hand side coefficients.

Glover's Algorithm with Gomory Fractional cut

No.	Rows	Columns	Reach. Opt.	Proved Opt.
1	10	5	5	5
2	10	10	4	1
3	10	15	3	0
4	10	20	1	0
5	10	25	1	0
6	5	5	5	5
7	5	10	4	1
8	5	15	3	0
9	5	20	3	0
10	5	25	1	0

Glover's Algorithm with Improved Gomory Fractional cut

No.	Rows	Columns	Rh. Opt.	Prvd Opt.	Imp.	Gomory
1	10	5	5	5	2.6	1
2	10	10	5	5	4.6	0.4
3	10	15	5	5	13.2	0.8
4	10	20	5	5	28.4	0.8
5	10	25	4	4	168	2.2
6	5	5	5	5	3	0
7	5	10	5	5	7.2	0.4
8	5	15	5	5	10.4	0.2
9	5	20	5	5	15.6	0.4
10	5	25	5	5	23.2	0.8

25

Results at a Glance

- The results appear to be as good as results in Lechford and Lodi [1] without using cover cuts and primal heuristics.
- Out of 1416 cuts only 35 could not be improved. (2.5%)
- P0033 is solve by 65 improved cuts and 2 Gomory cuts.
- Could not reach optimal solution for P0201. Reached 500 cut limit.

Summary

- This method seems to be particularly suitable for Primal algorithm since we know which coefficients need to be improved.
- Even if such a technique is never used in practice - this definitely shows the usefulness of generating cuts based on multiple constraints.
- Since stronger cutting planes are needed for primal algorithms compared to dual algorithm, this kind of strengthening may be useful in general.
- Other cuts like cover cuts can also be improved.
- For MIP, similar results can be proven with slight modifications - but the improvement is only possible for integer variables.

Summary

- This method seems to be particularly suitable for Primal algorithm since we know which coefficients need to be improved.
- Even if such a technique is never used in practice - this definitely shows the usefulness of generating cuts based on multiple constraints.
- Since stronger cutting planes are needed for primal algorithms compared to dual algorithm, this kind of strengthening may be useful in general.
- Other cuts like cover cuts can also be improved.
- For MIP, similar results can be proven with slight modifications - but the improvement is only possible for integer variables.

Summary

- This method seems to be particularly suitable for Primal algorithm since we know which coefficients need to be improved.
- Even if such a technique is never used in practice - this definitely shows the usefulness of generating cuts based on multiple constraints.
- Since stronger cutting planes are needed for primal algorithms compared to dual algorithm, this kind of strengthening may be useful in general.
- Other cuts like cover cuts can also be improved.
- For MIP, similar results can be proven with slight modifications - but the improvement is only possible for integer variables.

Summary

- This method seems to be particularly suitable for Primal algorithm since we know which coefficients need to be improved.
- Even if such a technique is never used in practice - this definitely shows the usefulness of generating cuts based on multiple constraints.
- Since stronger cutting planes are needed for primal algorithms compared to dual algorithm, this kind of strengthening may be useful in general.
- Other cuts like cover cuts can also be improved.
- For MIP, similar results can be proven with slight modifications - but the improvement is only possible for integer variables.

Summary

- This method seems to be particularly suitable for Primal algorithm since we know which coefficients need to be improved.
- Even if such a technique is never used in practice - this definitely shows the usefulness of generating cuts based on multiple constraints.
- Since stronger cutting planes are needed for primal algorithms compared to dual algorithm, this kind of strengthening may be useful in general.
- Other cuts like cover cuts can also be improved.
- For MIP, similar results can be proven with slight modifications - but the improvement is only possible for integer variables.

Thank You.

References I

A. Lechford, A. Lodi.
Math. Methods of Oper. Res..
2002. 56(1), 67-81.