DISCUSSION PAPER NO. 111

ON THE INTERRELATIONSHIPS BETWEEN GOMORY AND DANTZIG CUTTING PLANES

By

Avinoam Perry

Revised June 1975

* Assistant Professor, Graduate School of Management, Northwestern University, Evanston, Illinois.

ABSTRACT

In this paper we developed a cutting plane formula and used it for solving the all integer program. The cut is based on properties of the Gomory mixed integer cut but, nevertheless, its associated slack is restricted to be integer. Also we showed that the new cut is uniformly deeper than the modified Dantzig cut, and under circumstances discussed in the paper it may be deeper than the Gomory mixed integer cut for the all integer program along one or more dimensions. Computational experience for this new cut is contrasted with other codes, and the results (as shown in this paper) are encouraging.

On the Interrelationships Between Gomory and Dantzig Cutting Planes

Consider the ILP

(1) $$\text{Max} \quad Z = \sum_{j\in J} c_j x_j$$

$$\text{s.t.} \quad \sum_{j\in J} B_{ij} x_j = B_{i0} \quad i = 1,2,\ldots,m$$

$$x_j \geq 0 \text{ integer} \;, \qquad j \in \overline{J}$$

The ith row corresponding to the LP solution of (1) is given by

(2) $$x_B = B_0 - \sum_{j\in J} B_j t_j$$

where t_j is used for signifying a nonbasic variable, and x_B is a basic variable

The Gomory Cut [5] derived from (2) is given by

(3) $$\sum_{j\in J} b_j t_j \geq b_0$$

where: $b_j = B_j - [B_j]$ and $b_0 = B_0 - [B_0] > 0$

An alternative complementary cut is given by

(4) $$\sum_{j\in J} (1-b_j) t_j \geq 1 - b_0$$

Convergence was proved for (3) and (4) in [5]. An intuitive cut derived by Dantzig [3] is given by

(5) $$\sum_{j\in J} t_j \geq 1$$

As noted by Gomory and Hoffman [7], an algorithm based on (5), does not converge in a finite number of steps.

A combination of (3) and (4) produce the modified Dantzig cut [2] which is given by

(6) $$\sum_{j \in K} t_j \geq 1 \qquad K \subseteq J$$

where K includes only those nonbasic variables for which $b_j \neq 0$. Note that (6) is at least as deep as (5). Bowman and Nemhauser [1] proved that algorithms applying (6) are finite.

Rubin and Graves [9] used (3) to strengthen (6) and derived what they call MD2, which is given by

(7) $$\sum_{j \in K} t_j \geq 2$$

This cut is realized under the condition that $b_0 \neq b_j > 0$ for all nonbasic variables in the corresponding source row.

Salkin [10] suggested that the Gomory mixed integer cut [6] given by:

(8) $$\sum_{j \in R} b_j t_j + \sum_{j \in Q} \frac{b_0}{1-b_0} (1-b_j) t_j \geq b_0$$

where $R \cup Q = K$ and the assignment of any t_j to either R or Q is such that:

$$\text{if } b_j > b_0 \quad t_j \to Q$$

$$\text{if } b_j \leq b_0 \quad t_j \to R$$

is deeper than (3), (4), (5), and (6).

Garfinkel and Nemhauser [4] noted that in large problems it would appear that (8) is undesirable for ILP's because continuous slack variables are required.

Perry [8] tested 3 different versions of (8) on some 15 well known test problems and obtained encouraging results, but admitting difficulties in coding due to severe rounding error problems.

To facilitate the following discussion we simplify (8) as

(9) $$\sum_{j \in K} f_j t_j \geq f_0$$

$$\text{where } f_j = \begin{cases} b_j & \text{if } j \in R \\ \frac{b_0}{1-b_0}(1-b_j) & \text{if } j \in Q \end{cases}$$

$$f_0 = b_0$$

Note that $f_j \leq f_0$ for all $j \in K$ and the slack variable associated with (9) is continuous. It follows that (9) is always deeper than (or equivalent to) (3), (4), (5), and (6).

Let $f_m = \max_{j \in K} f_j$ for the ith source row, then from (9) it follows that

(10) $$f_m \sum_{j \in K} t_j \geq f_0$$

or

(11) $$\sum_{j \in K} t_j \geq \langle f_0/f_m \rangle^*$$

Since $\sum_{j \in K} t_j$ is an integer and since $f_m \leq f_0$ (11) is equal to (6) if there is at least one $b_j = b_0$. (11) is equal to (7) if $1 < f_0/f_m \leq 2$. (11) becomes $\sum_{j \in K} t_j \geq M$ if $(M-1) < f_0/f_m \leq M$ where M is an integer number.

Note that if f_0/f_m is an integer (9) is consistently deeper than (11) but if f_0/f_m is noninteger than (11) is deeper than (9) along one or more dimensions. Also note that (11) assumes an integer slack while (9) assumes a continuous slack.

Taking into account the procedure for deriving (8) we can now use it for strengthening (11). (8) was derived by requiring

(12) $$\sum_{j \in R} b_j t_j \geq b_0$$

---

*This notation is used to signify the smallest integer $\geq f_0/f_m$

or

(13) $$\sum_{j \in Q} \frac{b_0}{1-b_0} (1-b_j) t_j \geq b_0 \quad .$$

let $f_R = \max_{j \in R} b_j$ and $f_Q = \max_{j \in Q} \frac{b_0}{1-b_0} (1-b_j)$, then from (12) and (13) respectively, it follows that either

(14) $$f_R \sum_{j \in R} t_j \geq b_0$$

or

(15) $$f_Q \sum_{j \in Q} t_j \geq b_0 \quad .$$

From (14) and (15) we have

(16) $$\sum_{j \in R} t_j \geq \langle b_0/f_R \rangle = M_1 = \text{integer}$$

or

(17) $$\sum_{j \in Q} t_j \geq \langle b_0/f_Q \rangle = M_2 = \text{integer}$$

Combining (16) and (17) we have

(18) $$M_2 \sum_{j \in R} t_j + M_1 \sum_{j \in Q} t_j \geq M_1 \cdot M_2$$

If $M_1 \neq M_2$ then (18) is deeper than (11) which is deeper than (6) and therefore an algorithm applying (18) converges in a finite number of steps. (18) preserves the integrality requirement of all the variables involved and is less sensitive to rounding errors than (8).

The problems used for testing purposes are those developed and reported by J. Haldi [5] to test the LIP1 computer code. Further comparisons were made with respect to Trauth and Woolsey's study in computational efficiency [9] who tested the LIP1, IPM3, and ILP2-1 codes. The results are presented in tables

I and II and are self explanatory. All times were computed from the first executed instruction of the program to the end of the minimum output needed to interpret the results. All times are given in seconds. The word "iteration" refers to a single matrix pivot operation. MD and SGV2 were run on the CDC 6400 computer.

The first ten problems in the computational summary tables are Haldi's fixed charge problems. They are followed by IBM integer programming test problems also in [5]. The results are summarized in the following tables:

Table I

Fixed Charge Problems

* Code	MD		SGV2		LIP1		IPM-3		ILP2-1	
Problem	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.
1	1.979	37	1.902	20	1.833	24	3.117	54	0.852	36
2	2.508	52	1.401	13	1.350	15	3.767	81	0.935	47
3	1.996	31	1.430	14	1.883	26	3.033	37	1.384	104
4	1.001	10	0.966	6	1.483	18	4.100	91	0.674	18
5	3.765	48	2.414	16	9.012	158		+7000		+7000
6	3.708	45	2.819	24	7.507	123		+7000	3.273	311
7	3.401	46	2.497	16	7.883	159		+7000		+7000
8	3.322	45	2.310	14	6.417	126		+7000	3.033	306
9	1.917	15	1.282	9	3.233	42	5.183	118	3.598	298
10	8.670	86		+5000	9.150	102	71.100	1396		+7000

*MD is Applying the Cuts in (18) to reach an integer solution
SGV2 is utilizing the strengthened Gomory Mixed Integer Cut (8) Version 2 [8]
LIP1, IPM-3, ILP2-1 are codes tested by Trauth and Woolsey [11]

Table II

Haldi's IBM Problems

Code	MD		SGV2		LIP1		IPM-3		ILP2-1	
Problem	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.	Time	Itr.
1	2.005	12	1.512	8	1.866	11	2.300	8	1.010	9
2	2.623	25	2.785	23	3.016	32	2.833	17	1.056	13
3	2.300	41	2.617	41	2.866	53	2.633	22	0.705	23
4	12.988	85	8.882	40	11.666	73	5.933	24	3.492	41
5	81.650	402	31.566	149	66.483	351	51.600	1144		+7000
9	302.795	683	357.002	841	473.100	953	633.313	6758		+7000

Acknowledgement

I wish to thank A. Tamir for his useful comments on an early draft of this paper.

References

[1] Bowman, V. J. and G. L. Nemhauser, "A Finiteness Proof for the Modified Dantzig Cuts in Integer Programming," Naval Research Logistics Quarterly, 17, 1970, pp. 309-313.

[2] Charnes, A., and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, 1961, John Wiley & Sons.

[3] Dantzig, G. B., "Notes on Solving Linear Programs in Integers," Naval Research Logistics Quarterly, 6, 1959, pp. 75-76.

[4] Garfinkel, R. S. and G. L. Nemhauser, Integer Programming, 1972, John Wiley & Sons, Inc.

[5] Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs," Princeton IBM Mathematical Research Report, November 1958. Also in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill, New York, 1963, pp. 269-302.

[6] Gomory, R. E., "An Algorithm for the Mixed Integer Problem," RM-2597, Rand Corp., 1960.

[7] Gomory, R. E. and A. J. Hoffman, "On the Convergence of an Integer Programming Process," Naval Research Logistics Quarterly, 10, 1963, pp. 121-123.

[8] Perry, A., "The Strengthened Gomory Mixed Integer Cut for the All Integer Program," Discussion Paper No. 93, 1974, The Center for Mathematical Studies in Economics and Management Science, Northwestern University

[9] Rubin, D. S. and R. L. Graves, "Strengthened Dantzig Cuts for Integer Programming," Operations Research, 20, 1972, pp. 173-177.

[10] Salkin, H. M., "A Note on Gomory Fractional Cuts," Operations Research, 19, 1971, pp. 1538-1541.

[11] Trauth, C. A. and R. E. Woolsey, "Integer Linear Programming: A Study in Computational Efficiency," Management Science, 15, May 1969, pp. 481-493.