Discussion Paper No. 95

A STRENGTHENED MODIFIED DANTZIG CUT FOR THE ALL INTEGER PROGRAM

by

Avinoam Perry*

August 1974

* Assistant Professor of Decision Sciences, Northwestern University, Evanston, Illinois 60201

ABSTRACT

A strengthened modified Dantzig cut may be derived from the strengthened mixed integer cut for the all integer program [7]. This cut has the form: $\sum t_j \geq N$ where t_j is a nonbasic variable in the current basis and N is an integer ≥ 1. A cut selection rule based on the properties of this cut has been introduced and tested. The summary of computational experience indicate a good potential of this approach.

I. Introduction

This paper introduces a cutting plane method for solving the all integer program. The cut has the form: $\sum t_j \geq N$ where t_j is a nonbasic variable variable whose coefficient in the current optimum is not an integer and N is an integer number ≥ 1. This cut is always deeper or equal to the modified Dantzig cut and, therefore, an algorithm employing this cut must converge in a finite number of steps [1]. This cut is derived from the strengthened mixed integer cut of the all integer program, nevertheless, it is less sensitive to rounding errors, and its slack variable is an integer; factors which make it attractive computationally.

II. Derivation of the Strengthened Gomory Mixed Integer Cut of the All Integer Program

Consider the following L.P. problem

$$\max \sum_j c_j x_j$$

$$\text{s.t.} \sum_j B_{ij} x_j + t_i = B_{i0}$$

$$x_j, t_i = \text{integer}$$

The optimum solution to the problem has the following form:

(1) $$x_i = B_{i0} + \sum_k B_{ik} t_{ik} - \sum_{k'} B_{ik'} t_{ik'}$$

where: x_i is a basic variable

t_i is a nonbasic variable

B_{i0} is the value of the basic variable x_i at the current optimum solution

k is the set of all negative coefficients of the nonbasic variables t_{ik}

k' is the set of all positive coefficients of the nonbasic variables $t_{ik'}$

B_{ik} is the coefficient of the nonbasic variables t_{ik}

$B_{ik'}$ is the coefficient of the nonbasic variable t_{ik}'

$$(x_i, t_{ik}, B_{i0}, B_{ik}, B_{ik'} \geq 0).$$

Dividing the set k into two sets Q and R, and dividing the k′ sets into two sets Q′ and R′, (1) is extended to:

(2) $$x_i = B_{i0} + \sum_Q B_{iQ}t_{iQ} + \sum_R B_{iR}t_{iR} - \sum_{Q'} B_{iQ'}t_{iQ'} - \sum_{R'} B_{iR'}t_{iR'}$$

The assignment of a variable $t_i \in k$ to either Q or R, or the assignment of a variable $t_i \in k'$ to either Q′ or R′ is arbitrary. (2) may be extended to the following form:

(3) $$x_i = [B_{i0}] + b_{i0} + \sum_Q [B_{iQ}]t_{iQ} + \sum_Q b_{iQ}t_{iQ} + \sum_R ([B_{iR}] + 1)t_{iR}$$

$$- \sum_R (1 - b_{iR})t_{iR} - \sum_{Q'} [B_{iQ'}]t_{iQ'} - \sum_{Q'} b_{iQ'}t_{iQ'}$$

$$- \sum_{R'} ([B_{iR'}] + 1)t_{iR'} + \sum_{R'} (1 - b_{iR'})t_{iR'}$$

where: $[B_{i0}]$, $[B_{iQ}]$, $[B_{iQ'}]$, $[B_{iR}]$, $[B_{iR'}]$ are the integer part of B_{i0}, B_{iQ}, $B_{iQ'}$, B_{iR}, $B_{iR'}$, respectively, and b_{i0}, b_{iQ}, $b_{iQ'}$, b_{iR}, $b_{iR'}$ are the fractional part of B_{i0}, B_{iQ}, $B_{iQ'}$, B_{iR}, $B_{iR'}$, respectively. $(b_{i0}, b_{iQ}, b_{iQ'}, b_{iR}, b_{iR'} \geq 0)$

From the integrality requirement on all x_i's and t_i's it follows that:

(4) $$b_{i0} + \sum_Q b_{iQ}t_{iQ} - \sum_R (1 - b_{iR})t_{iR} - \sum_{Q'} b_{iQ'}t_{iQ'}$$

$$+ \sum_{R'} (1 - b_{iR'})t_{iR'}$$

is an integer which must be either ≥ 1 or ≤ 0.

If (4) is ≥ 1 then:

(4a) $$b_{i0} + \sum_{Q} b_{iQ}t_{iQ} + \sum_{R'} (1 - b_{iR'})t_{iR'} - 1$$

must be true.

If (4) is ≤ 0 then:

(4b) $$b_{i0} - \sum_{R} (1 - b_{iR})t_{iR} - \sum_{Q'} b_{iQ'}t_{iQ'} \leq 0$$

must be true.

The equivalent convexity cut form of (4a) is:

(4c) $$\sum_{Q} \frac{b_{iQ}}{1 - b_{i0}} t_{iQ} + \sum_{R'} \frac{1 - b_{iR'}}{1 - b_{i0}} t_{iR'} \geq 1.$$

The equivalent convexity cut form of (4b) is:

(4d) $$\sum_{R} \frac{1 - b_{iR}}{b_{i0}} t_{iR} + \sum_{Q'} \frac{b_{iQ'}}{b_{i0}} t_{iQ'} \geq 1.$$

At least one of the (4c) or (4d) must be true for (4) to be true, and since by definition (4c) and (4d) are ≥ 0 we have:

(5) $$\sum_{Q} \frac{b_{iQ}}{1 - b_{i0}} t_{iQ} + \sum_{R} \frac{1 - b_{iR}}{b_{i0}} t_{iR} + \sum_{Q'} \frac{b_{iQ'}}{b_{i0}} t_{iQ'} + \sum_{R'} \frac{1 - b_{iR'}}{1 - b_{i0}} t_{iR'} \geq 1$$

which is the strengthened mixed integer cut of the all integer program [7].

We may now summarize the derivation of the strengthened cut.

1. Solve the L. P. by ignoring the integrality requirements.
2. Derive a Gomory cut in the convexity cut form: $\sum_{j} (1/t_j^*)t_j \geq 1$

where: $t_j^* = b_0/b_j$ for all j in R'

$t_j^* = b_0/(1 - b_j)$ for all j in k.

3. If there is any $t_j^* < 1$ replace it by its complement. For example:

if $b_0/b_j < 1$ replace it by $(1 - b_0)/(1 - b_j) > 1$

if $b_0/(1 - b_j) < 1$ replace it by $(1 - b_0)/b_j > 1$.

4. The new cut is $\sum_j (1/t_j^*)t_j \geq 1$ where $t_j^* \geq 1$.

III. The Strengthened Modified Dantzig Cut

One significant disadvantage of the strengthened Gomory mixed integer cut of the all integer program is the fact that the new slack variable is not necessarily an integer. After the first cut is employed, the original problem becomes a mixed integer problem a fact which may cause a slow convergence if relatively many cuts are needed for solving the problem. It is possible, though, to use (5) for the derivation of a cut, the slack variable of which is an integer and, nevertheless, is deeper or equal to the modified Dantzig cut.

Consider the cut

$$\sum_j (1/t_j^*)t_j \geq 1 \qquad (t_j^* \geq 1)$$

$$\text{Define } N = \begin{cases} \min t_j^* & \text{if } \min_j t_j^* \text{ is an integer} \\ \min_j [t_j^*] + 1 & \text{if } \min t_j^* \text{ is not an integer} \end{cases}$$

Then the following cut is a valid cut:

$$(6) \qquad \sum_j t_j \geq N$$

Proof:

(7) Let $\min_j t_j^* = t_1^*$

then the cut $\sum_j (1/t_j^*)t_j \geq 1$ may be written as

(8) $t_1 + \sum_{j=2}^{n} (t_1^*/t_j^*)t_j \geq t_1^*$

Since $t_1^*/t_j^* \leq 1$ and every t_j is an integer, the solution to the problem:

(9) $\min \sum_j t_j$

(10) s.t. $t_1 + \sum_{j=2}^{n} (t_1^*/t_j^*)t_j \geq t_1^*$

t_j is an integer

is: $t_1 = t_1^*$, $\sum_{j=2}^{n} (t_1^*/t_j^*)t_j = 0$

if t_1^* is not an integer then $t_1 = [t_1^*] + 1$ is the solution to the above problem.

When $\min t_j^*$ is an integer (5) is deeper than or equal to (6) when $\min t_j^*$ is not an integer (6) is deeper than (5) at least along one dimension.

The main properties of the strengthened modified Dantzig cut are:

1. The slack variable of the cut is an integer.
2. No fractional variables are used and therefore the danger of rounding the errors is reduced significantly; a property which proves to be attractive computationally.

IV. Cut Selection Rule

Consider the following strengthened modified Dantzig cut: $\sum_j t_j \geq N$. The larger N the deepter the cut.

$$N = \begin{cases} \min t_j^* & \text{if } t_j^* \text{ is an integer.} \\ \min [t_j^*] + 1 & \text{if } t_j^* \text{ is not an integer} \end{cases}$$

Among several alterna tives select the one which maximizes N and use it as the source row for deriving the cut $\sum t_j \geq N$.

Notice that the cut selection rule in this algorithm almost always guarantees the selection of the best cut among several alternatives. This factor by itself may cause a relatively fast convergence.

V. Strengthening the Strengthened Modified Dantzig Cut

It is possible to strengthen (6) by using the properties of the strengthened mixed integer cut of the all integer program without violating the integrality property of the new slack variable and the coefficients of the nonbasic variables. In deriving (5) we used two alternative conditions (4c) and (4d) at least one of which must have been true for (5) to be true. An alternative presentation of (4c) and (4d) is:

(4ca) $\sum_{c} (1/t_j^*)\ t_j \geq 1$

(4da) $\sum_{d} (1/t_j^*)\ t_j \geq 1.$

By the same reasoning used for (6) we define

$$N = \begin{cases} \min_{c} t_j^* \text{ if } t_j^* \text{ is an integer} \\ \min_{c} [t_j^*] + 1 \text{ if } t_j^* \text{ is not an integer} \end{cases}$$

$$M = \begin{cases} \min_{d} t_j^* \text{ if } t_j^* \text{ is an integer} \\ \min_{d} [t_j^*] + 1 \text{ if } t_j^* \text{ is not an integer.} \end{cases}$$

Then we replace (4ca) and (4da) by (4cb) and (4db), respectively,

(4cb) $\sum_{c} t_j \geq N$

(4db) $\sum t_j \geq M$

and the cut becomes:

(7) $M \cdot \sum_{c} t_j + N \sum_{d} t_j \geq N \cdot M.$

If $N \neq M$ then (7) is deeper than (6).

Since M and N are integers the new slack variable is an integer and the original constraints set remains all integer.

VI. Computational Experience with the Strengthened Modified Dantzig Cut

Computational experience with cutting plane algorithms for integer programs do not, in general, yield a reliable solution. Many times they fail to converge in a reasonable number of steps and other times they cut off integer points which might have been otherwise candidates for an optimum solution of the optimization problem. Experiments with the Strengthened Gomory Mixed Integer Cut of the All Integer Program [7] proved the point just made. The main reasons for those difficulties lie in the high sensitivity for machine rounding errors which is built into most of the cutting plane algorithms. Cuts derived by the Strengthened Mixed Integer are based on the numerical fractions of the tableau coefficients which are already subjected to some previous rounding errors in their fractional part. The relatively good results obtained by employing this type of an algorithm were due to the fact that the truncating parameters (epsilon) were changed from one test problem to another so as to yield a minimum rounding error effect. At times when the truncating param meters were not modified from one test problem to another, the program failed to converge or yielded a solution different from the real optimum. The Strengthened Modified Dantzig cut algorithm has been, on the average, slower to converge than the Strengthened Mixed Integer Cut, but, nevertheless, it has been significantly more reliable and less sensitive to machine rounding errors; a fact which is caused, mainly, by the integer rather then fractional, coefficients of the additional constraint.

The problems used for testing pruposes 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. All programs 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	7 833	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 the Strengthened Modified Dantzig Cut
SGV2 is the Strengthened Gomory Mixed Integer Cut - Version 2 [7]
LIP1, IPM-3, ILP2-1 are codes tested by Trauth and Woolsey [9]

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

VII. Conclusions

While SGV2 was the fastest algorithm in most of these test problems it suffered from high sensitivity to machine rounding errors, therefore, every problem was solved several times with different truncating parameters until a solution was reached. MD, on the other hand, was slower but its reliability was proved to be of importance. It solved all the test problems and never failed to converge.

References

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

2. Glover, F., "Convexity Cuts", Working Paper, School of Business, The University of Texas, Austin, December 1969.

3. 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.

4. Gomory, R. E., "An Algorithm for the Mixed Integer Problem", RAND Report P-1885, February, 1960.

5. Haldi, John, "25 Integer Programming Test Problems", Working Paper No. 43, Graduate School of Business, Stanford University.

6. Martin, G. T., "An Accelerated Euclidean Algorithm for Integer Linear Programming", Recent Advances in Mathematical Programming, McGraw-Hill, New York, 1963, pp. 311-318.

7. Perry, A., "The Strengthened Gomory Mixed Integer Cut for the All Integer Program", Discussion Paper No 93, July 1974, The Center For Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, Illinois.

8. Rubin, D. S., and Graves, R. L., "Strengthened Dantzig Cuts for Integer Programming, ORSA, Volume 20, January 1972, pp. 178-182.

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