

An Estimate of the Number of Iterations for the Fully Integer Gomory Algorithm

1970

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-197001.79966>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

Abstract

Full Text

UDC 512.25/26+519.3:330.115

Yu. Yu. Finkelstein

An Estimate of the Number of Iterations for the Fully Integer Gomory Algorithm

(Presented by Academician L. V. Kantorovich on 4 I 1970)

1. Many algorithms of integer linear programming are based on the idea of cutting planes (see ^(1,2)). These algorithms behave very “irregularly.” Problems of considerable size may be solved rather quickly, while at the same time some small problems are not solved in tens of thousands of iterations. Up to the present time, all published results on the effectiveness of the cutting-plane method have been based on machine experiments (for a survey of results see ⁽¹⁾ and ⁽²⁾, Ch. 9).

In the present paper we study one of the best-known algorithms of the cutting-plane method (the fully integer Gomory algorithm ⁽³⁾). For the Gomory algorithm an example of a “bad” problem (problem T_0) is constructed, i.e., a problem with a number of iterations growing rapidly with the growth of the coefficients and with an increase in the number of variables. This made it possible to obtain a nontrivial lower estimate for the maximum number of iterations (Theorem 2). In addition, it turned out (Theorem 3) that in solving the problem T_0 , large numbers appear in the simplex tableaux.

2. The terminology and notation relating to the fully integer Gomory algorithm are borrowed from the book ⁽²⁾ (where it is called Gomory’s third algorithm). Instead of Gomory’s third algorithm we shall study a certain algorithm (the G_3 -algorithm), which coincides in every respect with Gomory’s third algorithm except, perhaps, for the rule for choosing the row k (see ⁽²⁾, p. 174, bottom) according to which the integer proper cut is constructed. In the G_3 -algorithm, k at the p -th iteration is also chosen from the set $\{i \mid i \in \{1, 2, \dots, n\}; x_{i0}^{p-1} < 0\}$, but it is chosen according to some G_3 -rule. Thus Gomory’s third algorithm is also a G_3 -algorithm. All subsequent considerations refer to an arbitrary G_3 -algorithm (and, in particular, to Gomory’s third algorithm).
3. We consider an integer linear programming problem whose linear constraints are written in the form of an integer and l -normal simplex tableau T_0

$$x_0 = x_{00}^0 + \sum_{j \in N_0} x_{0j}^0(-x_j) \rightarrow \max, \quad (1)$$

$$x_i = x_{i0}^0 + \sum_{j \in N_0} x_{ij}^0(-x_j), \quad i = 1, \dots, n, \quad (2)$$

$$x_j \geq 0, \quad j = 1, \dots, n, \quad (3)$$

$$x_j \text{ integer}, \quad j = 1, \dots, n. \quad (4)$$

Here N_0 is the set of indices of the nonbasic variables corresponding to the simplex tableau T_0 . $N_0 \subset \{1, 2, \dots, n\}$. In what follows we call problem (1)–(4) problem T_0 .

4. By $r(T_0)$ we denote the number of nonbasic variables corresponding to the tableau T_0 (i.e., the number of elements of the set N_0). By $q(T_0)$ we denote the maximum of the absolute values of the elements of the tableau T_0

$$q(T_0) = \max\{|x_{ij}^0| \mid i = 0, 1, \dots, n; j \in N_0 \cup \{0\}\}. \quad (5)$$

By $I(T_0)$ we denote the number of iterations of the G_3 -algorithm corresponding to the initial tableau T_0 . By $K(r, q)$ we denote the set of integer linear programming problems specified in the form of integer

and l -normal simplex tableaux T_0 for which $r(T_0) = r$ and $q(T_0) = q$. Further, let

$$I(r, q) = \max\{I(T_0) \mid T_0 \in K(r, q)\}. \quad (6)$$

5. Theorem 1. *Let $q \geq 2$ and $r \geq 3$. Then*

$$I(r, q) = \infty. \quad (7)$$

The proof is based on the construction of a problem T_0^* for which the G_3 -algorithm gives an infinite number of iterations. This problem is written out below (the idea of constructing T_0^* and the proof of (7) are taken from (4)).

Problem.

$$x_0 = 0 + 1(-x_{r+1}),$$

$$x_1 = \begin{cases} -1 - q(-x_4) + 1(-x_5) + 1(-x_6), & \text{for } r = 3, \\ -1 - q(-x_{r+1}) + 1(-x_{r+2}), & \text{for } r \geq 4, \end{cases}$$

$$x_i = 0 - q(-x_{r+i}) + 1(-x_{r+i+1}), \quad 2 \leq i \leq r - 3,$$

$$x_{r-2} = 0 - q(-x_{2r-2}) + 1(-x_{2r-1}) + 1(-x_{2r}),$$

$$x_{r-1} = 0 + 1(-x_{r+1}) - q(-x_{2r-1}) + (q-1)(-x_{2r}),$$

$$x_r = 0 + \left(\sum_{j=r+1}^{2r+1} (q-1)(-x_j) \right) - q(-x_{2r}),$$

$$x_i = 0 + (-1)(-x_i), \quad i = r + 1, \dots, 2r,$$

$$0 \leq x_i \text{ integer}, \quad i = 1, 2, \dots, 2r.$$

For greater clarity, see the simplex tableau T_0 corresponding to the problem T_0^* for $r = 4$.

	1	$-x_5$	$-x_6$	$-x_7$	$-x_8$
x_0	0	1	0	0	0
x_1	-1	$-q$	1	0	0
x_2	0	0	$-q$	1	1
x_3	0	1	0	$-q$	$q-1$
x_4	0	$q-1$	$q-1$	$q-1$	$-q$
x_5	0	-1	0	0	0
x_6	0	0	-1	0	0
x_7	0	0	0	-1	0
x_8	0	0	0	0	-1

It is not difficult to see (on the basis of the same considerations as in ⁽⁴⁾) that the problem T_0^* is infeasible. Meanwhile, the greatest interest for investigation is represented by problems that have a solution (it is precisely for such problems that Gomory's third algorithm will be finite). Denote by $K'(r+1, q)$ the set of solvable integer linear programming problems given in the form of simplex tableaux $T_0 \in K'(r+1, q)$. Further, let

$$I'(r+1, q) = \max\{I(T_0) \mid T_0 \in K'(r+1, q)\}. \quad (8)$$

Theorem 2 (main). *Let $q \geq 2$ and $r \geq 3$. Then*

$$I'(r+1, q) \geq [q^{r-2} + (r-2)]/r + 1. \quad (9)$$

6. The proof of Theorem 2 is based on the construction of the feasible problem $T_0 \in K'(r+1, q)$ written below, for which one can show that

$$I(T_0) \geq [q^{r-2} + (r-2)]/r + 1. \quad (10)$$

It is obvious that

$$I'(r+1, q) \geq I(T_0), \quad (11)$$

so that (9) immediately follows from (10) and (11), which proves Theorem 2.

7. The problem T_0 has the following form:

Problem T_0 .

$$x_0 = 0 + 1(-x_{r+1}) + q(-x_{2r+1}),$$

$$x_1 = \begin{cases} -1 - q(-x_4) + 1(-x_5) + 1(-x_6) - 1(-x_7), & \text{for } r = 3, \\ -1 - q(-x_{r+1}) + 1(-x_{r+2}) - 1(-x_{2r+1}), & \text{for } r \geq 4, \end{cases}$$

$$x_i = 0 - q(-x_{r-i}) + 1(-x_{r+i+1}), \quad 2 \leq i \leq r-3,$$

$$x_{r-2} = 0 - q(-x_{2r-2}) + 1(-x_{2r-1}) + 1(-x_{2r}),$$

$$x_{r-1} = 0 + 1(-x_{r+1}) - q(-x_{2r-1}) + (q-1)(-x_{2r}),$$

$$x_r = 0 + \left(\sum_{j=r+1}^{2r-1} (q-1)(-x_j) \right) - q(-x_{2r}),$$

$$x_i = 0 + (-1)(-x_i), \quad i = r+1, \dots, 2r+1,$$

$$0 \leq x_i \text{ integer}, \quad i = 1, \dots, 2r+1.$$

In other words, the tableau T_0 is obtained from the tableau T_0^* by adjoining the row x_{2r+1} and the column x_{2r+1} . For greater clarity, the simplex tableau T_0 corresponding to $r = 4$ is written out below.

	1	$-x_5$	$-x_6$	$-x_7$	$-x_8$	$-x_9$
x_0	0	1	0	0	0	q
x_1	-1	$-q$	1	0	0	-1
x_2	0	0	$-q$	1	1	0
x_3	0	1	0	$-q$	$q-1$	0
x_4	0	$q-1$	$q-1$	$q-1$	$-q$	0
x_5	0	-1	0	0	0	0
x_6	0	0	-1	0	0	0
x_7	0	0	0	-1	0	0
x_8	0	0	0	0	-1	0
x_9	0	0	0	0	0	-1

By $X' = (x'_1, \dots, x'_{2r+1})$ we denote the desired l -optimal plan of the problem T_0 . The proof of formula (10) is carried out in several stages.

Lemma 1. $x'_{2r+1} \geq 1$.

Lemma 2. $(x'_1, \dots, x'_{2r}, x'_{2r+1}) = (0, \dots, 0, 1)$.

Following (2), denote by $T_0, T_1, \dots, T_k, \dots$ the simplex tableaus successively obtained when applying the G_3 -algorithm to the problem T_0 . To the tableau $T_k = \|x_{ij}^k\|_{i=0,1,\dots,2r+1; j \in N_k}$ there corresponds the set N_k of indices of nonbasic variables. Here $\bar{N}_k = N_k \cup \{0\}$. If the G_3 -algorithm solves the problem T_0 in a finite number of iterations, then we denote by δ the number of the last iteration. At the k -th iteration there is a transition from the tableau T_k to the tableau T_{k+1} .

Lemma 3. x_{2r+1} enters the basis only at the last (the δ -th) iteration, i.e.

$$(2r+1) \in N_k, \quad k = 0, 1, \dots, \delta; \quad 2r+1 \notin N_{\delta+1}.$$

Lemma 4. If $t \leq \delta$, then

$$\sum_{i=1}^r x_{i0}^t = -1; \quad x_{ij}^t = x_{ij}^0, \quad i = 0, 1, \dots, 2r+1, \quad j \in N_t.$$

Lemma 5. If $t \leq \delta$, then

$$-(r-1)(q-1) - 1 \leq x_{i0}^t \leq q-1, \quad i = 1, \dots, r.$$

Lemma 6. If $t \leq \delta$, then

$$0 = x_{i0}^0 \leq x_{i0}^1 \leq \dots \leq x_{i0}^t, \quad i = r+1, \dots, 2r.$$

Lemma 7. If $t \leq \delta$ and $\Delta_i^t = x_{i0}^t - x_{i0}^0$, $i = 1, \dots, r$, then

$$\Delta_1^t = \begin{cases} qx_{4,0}^t - x_{5,0}^t - x_{6,0}^t, & \text{for } r = 3, \\ qx_{r+1,0}^t - x_{r+2,0}^t, & \text{for } r \geq 4, \end{cases}$$

$$\begin{aligned}\Delta_i^t &= qx_{r+i,0}^t - x_{r+i+1,0}^t, \quad 2 \leq i \leq r-3, \\ \Delta_{r-2}^t &= qx_{2r-2,0}^t - x_{2r-1,0}^t - x_{2r,0}^t.\end{aligned}$$

Lemma 8. If $t \leq \delta$, then

$$\sum_{i=1}^r x_{r+i,0}^t = \frac{1}{q-1} \left(x_{r+1,0}^t (q^{r-1} - 1) - \sum_{i=1}^{r-2} \Delta_i^t (q^{r-i-1} - 1) \right).$$

Lemma 9. If: 1) $t \leq \delta$, 2) $x_{10}^t \leq -1$, 3) $t \geq 1$, then

$$\sum_{i=1}^r x_{r+i,0}^t \geq q^{r-2} + (r-2).$$

Lemma 10. If the G_3 -algorithm, in a finite number of iterations, gives a solution of problem T_0 , then there will be an iteration number $t \geq 1$ such that

$$t \leq \delta, \quad x_{10}^t \leq -1.$$

Lemma 11. If: 1) $1 \leq t \leq \delta$, 2) $x_{10}^t \leq -1$, then

$$I(T_0) \geq t + 1 \geq \frac{q^{r-2} + (r-2)}{[(r-1)(q-1) + 1]/q} + 1 \geq \frac{q^{r-2} + (r-2)}{r} + 1.$$

Here $\lceil Z \rceil$ denotes the least integer not less than Z .

It follows directly from Lemma 11 that formula (10) holds, which also makes it possible (see above) to prove Theorem 2.

8. Lemma 11 makes it possible to write down explicitly integer linear programming problems of small size and with small coefficients, for which the number of iterations of any G_3 -algorithm (including Gomory's third algorithm) is sufficiently large. For example, for $r = 10$ and $q = 10$ ($2r + 1 = 21$) we obtain $I(T_0) \geq 10^7 + 2$.
9. From Lemmas 9 and 10 it follows directly that

Theorem 3. If the G_3 -algorithm, in a finite number of iterations, gives a solution of problem T_0 , then there will be an iteration number $t \geq 1$ such that

$$\max\{x_{r+i,0}^t \mid i = 1, \dots, r\} \geq [q^{r-2} + (r-2)]/r.$$

Theorem 3 shows that problem T_0 is "bad" not only in the sense that it is solved in a large number of iterations, but also in the sense that, in the process of solving problem T_0 , large elements may occur in the simplex tableaux.

Central Economics and Mathematics Institute
Academy of Sciences of the USSR
Moscow

Received
1 XII 1969

REFERENCES

1. M. L. Balinski, *Manag. Sci.*, **12**, No. 3, 253 (1965).
2. A. A. Korbut, Yu. Yu. Finkelstein, *Discrete Programming*, Moscow, 1969.
3. R. E. Gomory, In: *Industrial Scheduling*, New Jersey, 1963, ch. 13;
Russian transl. R. E. Gomory, *Calendar Planning*, Moscow, 1966, p. 227.
4. Yu. Yu. Finkelstein, *Problems of Cybernetics*, no. 21, 1969, p. 249.

Note: Figure translations are in progress. See original paper for figures.

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.