Aligarh Journal of Statistics
Vol. 29 (2009), 1-6

A SIMULATION BASED ALGORITHM FOR INTEGER LINEAR PROGRAMMING PROBLEMS

Rajiv V. Parikh and K. Muralidharan

ABSTRACT

In this article, we present a computer simulation based algorithm for solving all kinds of *IPP*. The suggested algorithm has been found to be efficient and works well with Zero-One *IPP* and Mixed type of *IPP*. As such no any mathematical computations are required to execute this algorithm.

1. INTRODUCTION

Integer Linear Programming (ILP) are linear program in which some or all the variables are restricted to Integer (or discrete) values. Many authors have pointed out that to get an exhaustive solution for an ILP is so far not available. Among the methods available Branch and Bound method gives a better solution than any other method as far as computational experience is concerned. Though this method allows solutions for mixed problems, it also has limitations in deciding branching variables. In furtherance to this as a special case of General Branch and Bound algorithm, there is Zero-One implicit enumeration algorithm, which also works on heuristics background. As many authors have suggested that only using new technological advances in computers one can offer an efficient solution for ILP algorithm. [See Taha (2007)]. The other exhaustive reference of interest is due to the book by Hastie *et. al* (2001) where they have given genetic algorithm for solving ILP. Motivated by the above facts and difficulties, we present a simple computer simulation based algorithm to solve a Zero-One ILP. The proposed algorithm has the following benefits:

i) Any number of Integer variables can be treated / accommodated.

ii) No approximation for solutions.

iii) No any restriction for constraints.

iv) No pre or post mathematical computations.

This algorithm works under the assumption that the constraint coefficient matrix has non-negative entries

The only pre-requisite for using this algorithm is the knowledge of any Programming knowledge preferably "C" or "C^{++}" language. In the next section, we present the algorithm for Zero-One Integer problem. In section 3, we slightly modify the algorithm and discuss the same for general ILP. The complete program written in "C" is available with the authors.

Throughout this paper we assume an ILP of the following kind: wherever changes are required they have been mentioned at the right places.

Maximize $Z = CX$

Subject to $AX \leq B$, $X \geq 0$,

where A, B and C have the usual meaning.

2. ZERO-ONE INTEGER PROGRAMMING PROBLEM

Algorithm:

Step 1 Input A, B, C, n and SN, where n and SN are the number of variables and simulation numbers respectively.

Step 2 Generate a Bernoulli random variable and store it in $x[i]$, $i = 1, 2, \ldots, n$.

Step 3 Use these $x[i]$, $i = 1, 2, \ldots, n$ in the constraints, if the values satisfy all the constraints then compute objective function, Z else repeat Step 2.

Step 4 Let $y[i]$, $i = 1, 2, \ldots, n$ and 'Z_0' be the corresponding feasible solution and objective function value based on the previous simulation.

Step 5 If $Z \geq Z_0$ then $Z = Z_0$ and set $y[i] = x[i]$, $i = 1, 2, \ldots, n$.

Step 6 Repeat step 2 to 5 for a suitable choice of SN.

The above algorithm is seemed to converge to the optimum solution very fast irrespective of the type of constraints. The stopping rule depends on the number of variables in the program and can be checked after running a few simulations. This can be done in a simple way like this: Put a counter for Z and if the value of Z remains same for frequent iterations after m $(m < SN)$ simulations then the optimum values are achieved. In Table 1 we present some of the famous Zero–One ILP previously discussed by many authors along with their solution and simulation number at which the final solution is attained. The solution in each case is exactly the same as obtained by the authors.

Table 1:

No.	Integer Linear programming Problem	Solution as per our algorithm	Simulation number where solution is obtained
1	$Max Z = 20X_1 + 40X_2 + 20X_3 + 15X_4 + 30X_5$ Subject to $5X_1 + 4X_2 + 3X_3 + 7X_4 + 8X_5 \leq 25$	$X_1 = 1$, $X_2 = 1$, $X_3 = 1$, $X_4 = 1$, $X_5 = 0$, $Z = 95$	87

	$X_1+7X_2+9X_3+4X_4+6X_5 \le 25$ $8X_1+10X_2+2X_3+X_4+10X_5 \le 25$ $X_i=0$ or 1, $i=1,2,3,4,5$ (Ref. Taha (2001), Eg: 9.3b-3, page no. 398)		
2	$MaxZ=3X_1+2X_2+5X_3+2X_4+3X_5$ Subject to $X_1+X_2+X_3+2X_4+X_5 \le 4$ $7X_1+3X_3+4X_4+3X_5 \le 8$ $11X_1+6X_2+3X_4-3X_5 \le 3$ $X_i=0$ or 1, $i=1,2,3,4,5$ (Ref. Taha (2001), Eg: 9.3-3, page no. 393)	$X_1=1$, $X_2=1$, $X_3=0$, $X_4=0$, $X_5=0$, $Z=5$	30
3	$MaxZ=20X_1+40X_2+20X_3+15X_4+30X_5$ $+12X_6+13X_7+9X_8+12X_9$ Subject to $13X_1+50X_2+7X_3+6X_4+36X_5+6X_6$ $+46X_7+38X_8+18X_9 \le 50$ $3X_1+8X_2+6X_3+2X_4+34X_5+6X_6$ $+4X_7+7X_8+3X_9 \le 20$ $X_i=0$ or 1, $i=1,2,\ldots,9$ (Ref. Srivastava, *et. al* (1997), Illustration: 3.3b-3, page no. 727)	$X_1=1$, $X_2=0$, $X_3=1$, $X_4=1$, $X_5=0$, $X_6=1$, $X_7=0$, $X_8=0$, $X_9=1$, $Z=72$	333

We have also tried this algorithm for many Zero – One ILP problems in the literature and all problems discussed in Taha (2001) and obtained identical solutions in each case.

3. GENERALISED INTEGER PROGRAMMING PROBLEM

Pure Integer Programming Problem:

Here we slightly modify the above algorithm to incorporate the general setup of IPP.

Algorithm:

Step 1 Input A, B, C, m, n and SN, where m, n and SN are the number of constraints, variables and simulation numbers respectively.

Step 2 Find the value of variables in the following way

For $i=1$ to n

For $j=1$ to m

$S[i]\,[j]=\text{int}(B[j]/A[j][i])$

Next j

Next i

Step 3 Find the Minimum value for all variables; denote it by $m[i]$.

$$m[i] = \min(s[i][1], s[i][2], \ldots, s[i][m])$$

Step 4 Generate an Integer random number in the interval $[0, m[i]]$ and store it in $x[i]$, $i = 1, 2, \ldots, n$ {In 'C' language the code rand () % $(m[i]+1)$ will generate the Integer number in $[0, m[i]]$}

Step 5 Use these $x[i]$, $i = 1, 2, \ldots, n$ in the constraints, if the values satisfy all the constraints then compute objective function, Z else repeat Step 4

Step 6 Let $y[i]$, $i = 1, 2, \ldots, n$ and 'Z_0' be the corresponding feasible solution and objective function value based on the previous simulation

Step 7 If $Z \geq Z_0$ then $Z_0 = Z$ and set $y[i] = x[i]$, $i = 1, 2, \ldots, n$

Step 8 Repeat step 4 to 7 for a suitable choice of SN

Based on the above algorithm, we obtained solutions for some General *IPP* presented in many standard textbooks. They are presented in Table 2. The solutions have been checked using the software *TORA* given by Taha (2001). The interesting thing about this algorithm is that, we can have alternate solutions also for suitably running with different simulations.

Table 2:

No.	Integer Linear programming Problem	Solution as per our algorithm	Simulation number where final solution is obtained
1	$MaxZ = 3X_1 + 3X_2 + 4X_3 + 7X_4$ Subject to $X_1 + X_2 + X_3 + X_4 \leq 9$ $5X_1 + 3X_2 + 2X_3 + X_4 \leq 60$ $X_1 + 3X_2 + 5X_3 + 5X_4 \leq 50$	$X_1 = 2$, $X_2 = 0$, $X_3 = 0$, $X_4 = 6$, $Z = 48$	375
2	$MaxZ = 6X_1 + 13X_2 + 10X_3 + 20X_4 + 25X_5$ Subject to $15X_1 + 3X_2 + 5X_3 + 6X_4 + 10X_5 \leq 15$ $X_1 + 3X_2 + 2X_3 + 5X_4 + X_5 \leq 3$ $11X_1 + 6X_2 + 3X_4 - 3X_5 \leq 3$	$X_1 = 0$, $X_2 = 0$, $X_3 = 0$, $X_4 = 0$, $X_5 = 1$, $Z = 25$	10

	$X_5 \geq 0.25$ $X_1 + 3X_2 + X_3 + 5X_4 + X_5 = 1$		
3	$MaxZ = 16X_1 + 15X_2 + 17X_3 + 15X_4 + 40X_5 + 12X_6 + 13X_7 + 9X_8 + 12X_9$ Subject to $13X_1 + 50X_2 + 7X_3 + 6X_4 + 36X_5 + 6X_6 + 46X_7 + 38X_8 + 18X_9 \leq 50$ $3X_1 + 8X_2 + 6X_3 + 2X_4 + 34X_5 + 6X_6 + 4X_7 + 7X_8 + 3X_9 \leq 20$	$X_1 = 0$, $X_2 = 0$, $X_3 = 1$, $X_4 = 7$, $X_5 = 0$, $X_6 = 0$, $X_7 = 0$, $X_8 = 0$, $X_9 = 0$, $Z = 122$	17834

We declare to state that the IPP having negative coefficient in the constraints is not treated in this paper and is under investigation.

Mixed Integer Programming Problem:

Here we make the following changes in Step 4 of the above algorithm, given in section 3. Other steps remain same.

Step 4 Generate a random number in the interval $[0, m[i]]$ and store it in $x[i]$, $i = 1, 2, \ldots, n$ (In 'C' language the code rand () % $(m[i]+1)$ will generate the Integer number in $[0, m[i]]$ and the code (rand $()/RAND\_MAX)^*$ $m[i]$ will generate real numbers.)

Table 3:

No.	Integer Linear programming Problem	Solution as per our algorithm	Simulation number where final solution is obtained
1	$MaxZ = 10X_1 + 8X_2$ Subject to $2X_1 + 4X_2 \leq 25$ $4X_1 + 6X_2 \leq 27$ X_1 is integer	$X_1 = 6$, $X_2 = 0.4988$, $Z = 63.99$	3805
2	$MaxZ = X_1 + X_2$ Subject to $2X_1 + 5X_2 \leq 16$ $6X_1 + 5X_2 \leq 30$ X_1 is integer	$X_1 = 4$, $X_2 = 1.1988$, $Z = 5.1988$	10718
3	$MaxZ = 9X_1 + 3X_2 + 5X_3$	$X_1 = 3$, $X_2 = 0$, $X_3 = 1.2843$,	945

	Subject to $2X_1 + X_2 + 7X_3 \leq 15$ $9X_1 + 3X_2 + 2X_3 \leq 30$ $3X_1 + 11X_2 + 13X_3 \leq 45$ $X_1,\ X_2$ are integer	$Z = 33.4217$	

Acknowledgement

Both the authors thank the referee and the Editor for their valuable suggestions.

REFERENCES

Hastie, T., Tibshirani, A., and Friedman, J.H. (2001): *The Elements of Statistical Learning*. Springer-Verlag.

Srivastava, U.K., Shenoy, G.V. and Sharma, S.C. (1997): *Quantitative Techniques for Managerial Decisions*. New Age International (P) Ltd. Publishers.

Taha, H.A. (2001): *Operations Research*. Prentice Hall.

Taha, H A. (2007): *Operations Research: An Introduction*. Prentice Hall.

Received : 22-05-2005

Revised : 16-09-2009

Rajiv V. Parikh
Department of Statistics,
Bhavnagar University, Bhavnagar-364 002. India.

K. Muralidharan
Department of Statistics,
M.S. University of Baroda,
Vadodara-390 002. India.
e-mail: lmv_murali@yahoo.com