Journal of the Nigerian Association of Mathematical Physics
Volume 54 **(January** *2020 Issue*), **pp129 – 136**
© J. of NAMP

INTEGER LINEAR PROGRAMMING AN OBJECT ORIENTED APPROACH

[1]E. Ebojoh and [2]J.A. Akpobi

Department of Production Engineering, University of Benin, P.M.B. 1154 Benin City, Edo State, Nigeria.

Abstract

In this work we consider Integer Linear Programming problems (ILP) and develop computerized solutions to these problems. The computational process involved using object oriented approach to develop accurate solutions to Integer Linear Programming problems. The solutions were implemented with Microsoft Visual Basic.Net programming language. In conducting the analysis, ILP models involving: All Integer linear programming (AILP), Mixed Integer linear programming (MILP) and Fuzzy Integer linear programming (FILP) were considered. The accuracy of the software was tested with a number of bench mark examples and the results obtained, showed that the software developed was robust to parameter variations and very accurate. Instead obtaining software for the individual ILP presented a single software has been developed to handle the three cases.

***Keywords*:** Gomory's cut; Object-oriented Programming (OOP); Computer aided Design (CAD)

1. Introduction

In many practical problems, the decision variables make sense only if they are assigned integer values; this is most applicable in linear programming. For example, it is often necessary to assign men, machines and vehicles to activities in integer quantities. This is of particular interest in business and industries where quite often, the fractional solutions are unrealistic because the units are not divisible. An objective scaling ensemble approach a novel two-phase heuristic for integer linear programming problems was used on a wide variety of integer linear programming problems and was found to be effective[1]. The technique identities and aggregates multiple partial solutions to modify the problem formulation and significantly reduce the search space. Furthermore, a modified method for unpromising variants at solving the problem of the integer linear programming with Boolean variables was adopted [2]. This method can help to improve, i.e. to decrease a calculated complexity of an algorithm of a solution of the integer linear programming problem with Boolean variables based on the proposed set of some clipping strategies. Also a new scalable algorithm called NSLP, for solving high-dimensional, non-stationary linear programming problems on modern cluster computing systems was formulated[3].The algorithm consists of two phases: Quest and Targeting. The Quest phase calculates a solution of the system of inequalities defining the constraint system of the linear programming problem under the condition of dynamic changes in input data, Fejer mappings is used. The Targeting phase forms a special system of points having the shape of an n-dimensional axisymmetric cross. The cross moves in the n-dimensional space in such a way that the solution of the linear programming problem is located all the time in an -vicinity of the central point of the cross. It was explained that Linear Programming Problems have numerous and outstanding significance in Decision making viz. transportation model, travelling sales-man problems as well as large-scale problems[5]. Furthermore, in many applications that are practical the information in the system under consideration may not be precise, in such situations, it is appropriate to use the Fuzzy LPP [5,6]. In serving managers by providing the best solutions to a variety of integer linear programming problems with fuzzy variables in a simple and effective manner classical integer linear programming was proposed [7]. Further, a new decomposition method of solving mixed integer linear programming (MILP) with fuzzy variables was also proposed[4,8].Mixed Integer Linear Programming techniques are also effective not only for mixed problems, but also for pure-integer problems, pure-binary problems, or in fact any combination of real-, integer-, and binary-valued variables[9]. In the paper solving fuzzy linear programming by using revised tsao method [10, 11],it was suggested using this method in the fuzzy primal simplex where we need to evaluate the optimality condition in the iteration. The mentioned approach will be useful when one has to use the fuzzy dual simplex too and gave a systematic and a convenient method for comparing of the fuzzy numbers, by the use of ranking functions. Other theoretical studies done on integer linear programming, which cover a wide range of problems and approach were considered for modeling the three approaches mentioned above. Equations that served as guides for developing this software are presented in this work at

Corresponding Author**:** Ebojoh E., Email: voke.ebojoh@uniben.edu, Tel: +2347065474779, +2348055040348 (JAA)

sections 2.0. Examples to illustrate the use of the software was presented in section 4.0, discussion in section 5.0 and the conclusion of the work in section 6.0

2.0 Design of Software Model

In designing the relevant algorithm for this software, standard design methodologies well treated in a number of texts were followed. A number of variables were put into consideration in this work which includes: fractional cut (s_g), variables (x_i) and dual method of solving simplex method.

2.1 All Integer Linear Programming

All integer programming problems in which all decision variables are restricted to integer values

Maximize $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$

Subject to the constraints

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$. \quad . \quad . \quad . \quad .$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$
$$x_1x_2, \ldots\ldots x_n \geq 0 \; and \; are \; integers.$$

(1)

2.2 Gomory's All (PURE) Integer Cutting Plane

The first cutting plane algorithm introduced by Gomory to solve pure IP (PIP) problems is known as Gomory's fractional cuts because all the nonzero coefficients of the generated cuts are less than one. It is sometimes referred to as the method of integer form. This is also the first cutting plane algorithm that was proven to be finitely convergent.

In solving the above equation, the integer linear programming problem is first relaxed and solved for using the optimum solution with linear programming methods. If an integer solution is achieved, then the problem is optimal. Using the simplex method to solve a linear program produces a set of equations of the form $x_r + \sum_{j\neq r} a_{rj}\, x_j = br$

Suppose the basic variable x_r has the largest fractional value among all basic variables, restricted to be integers. Then the rth constraints equation (row) from the simplex table can be written as:

$$x_{Br}(= b_r) = 1.x_r + (a_{r1}x_1 + a_{r2}x_2 + \cdots) = x_1 + \sum_{j\neq r} a_{rj}\, x_j \qquad (2)$$

Where $x_j\,(j = 1,2,3, \ldots\ldots)$ represents all the non-basic variables in the rth constant (row), except the variable $x_r$and $b_r(= x_{Br})$ are the non- integer value of variable x_j.let us decompose the coefficients of x_j, x_rvariables and$x_{Br}$to integer and non-negative fractional parts in Equation (2) as shown blow

$$[x_{Br}] + f_r = (1 + 0)x_r + \sum_{j\neq r}\{(a_{rj}) + f_{rj}\}x_j \qquad (3)$$

Where (x_{Br} and (a_{rj}) denote the largest integer obtained by truncating the fractional parts from x_{Br} and a_{rj} respectively.

Rearranging Equation (3) so that all the integer coefficients appear on the left-hand side, we get

$$f_r + \{(x_{Br}) - x_r - \sum_{j\neq r}(a_{rj})x_j\} = \sum_{j\neq r} f_{rj}\;x_j \qquad (4)$$

Where f_r is strictly a positive fraction $(0 < f_f < 1)$ while f_{rj} is a non- negative fraction $(0 \leq f_r \leq 1)$. Since all the variable (including slack) are require to assume integer values, the terms in the left hand side in Equation (4) is f_r plus a non-negative number, we may write it in the form of the following inequalities:

$$f_r \leq \sum_{j\neq r} f_{rj}\;x_j \qquad (5)$$

Or $\quad \sum_{j\neq r} f_{rj}\, x_j = f_r + s_g \quad or \quad - f_r = s_g - \sum_{j\neq r} f_{rj}\; x_j \qquad (6)$

Where $s_g$is a non-negative slack variable and is called the Gomory's slack variable.

Equation (6) represents Gomory's cutting plane constant. When this new constant is added to the bottom of optimal simplex table, it would create an additional row in the table, along with a column for the new variable s_g.

If the above fractional equation is added to the collection of constraints in the tableau, it is the same as if beginning with the previous integer equation as an initial condition. This is the essence of the cutting plane method of solving integer linear programming problems.

2.3 Mixed Integer Linear Programming

Maximize $\quad \boldsymbol{\sum_{j=1}^{n} c_j x_j}$

Subject to $\quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1, . . . , m \qquad (7)$

$$X_j \geq 0, \quad j \in k$$
$$X_j \leq 0, \quad j \in k'$$

Where k and k' are the set of integer variables and set of continuous variables respectively.

This can also be expressed as**:**

Maximize $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$ (8)
Subject to the constraint
$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$
$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$
$. \quad . \quad . \quad . \quad .$
$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$
And $x_j\ are\ integers; where\ j = 1,2,3, \ldots .\ k(k < n)$

2.4 Gomory Mixed Integer Cutting Plane

Suppose that the basic variable x_r is restricted to be integer and has a largest fractional value among all those basic variables that are restricted to take integer value then rewrite the rth constraint (row) from the optimal simplex table as follows (same as Equation (2)):

$x_{Br} = x_r + \sum_{j \neq r} a_{rj}\, x_j$ (9)

Where x_j represents all the non-basic variables in the rth row except variablex_r and x_{Br} is the non-integer value as of variable x_r. Decompose coefficient of x_j, x_r variables and x_{Br} into integer and non-negative fractional parts, as shown below:

$x_{Br} = (x_{Br}) + f_r$ (10)

And $R+ = \{j : a_{rj} \geq 0\}$ set of subscripts j (columns in simplex table) for which $a_{rj} \geq 0$
$R- = \{j : a_{rj} < 0\}$, Set of subscripts j (columns in simplex table) for which $a_{ij} < 0$,Then Equation (9) can be rewritten as,

$[x_{Br}] + f_r = (1 + 0)x_r + \sum_{j \in R\_} a_{rj}\ x_j$ (11)

Rearrange the terms in Equation (11) so that all of the integer coefficients appear on the right-hand side. This gives

$\sum_{j \in R+} a_{rj}\, x_j + \sum_{j \in R-} a_{rj}\, x_j = f_r + \{[x_{Br}] - x_r\} = f_r + 1$ (12)

Where f_r is strictly positive number ($i.e. 0 < f_r < 1$), and I is the value

Since the terms in the bracket on the right-hand side of Equation (12) are integers, left-hand side in Equation (12) is either positive or negative according as $f_r + 1$ is positive or negative.

CASE 1: Let $f_r + 1$ be positive. Then this must be $f_r, 1 + f_r, 2 + f_r, \ldots$ we shall then have

$\sum_{j \in R+} a_{rj}\, x_j + \sum_{j \in R-} a_{rj}\, x_j \geq f_r$ (13)

Since$a_{rj} \in R_-$ are non-positive and $x_j \geq 0$

$\sum_{j \in R+} a_{rj}\, x_j \geq \sum_{j \in R+} a_{rj}\, x_j + \sum_{j \in R-} a_{rj}\, x_j$ (14)

And hence $\sum_{j \in R+} a_{rj}\, x_j \geq f_r$

CASE 2 let $f_r + 1$ be negative, then it must be $f_r, -1 + f_r, -2 + f_r, \ldots.$ and we shall have

$\sum_{j \in R-} a_{rj}\, x_j \geq \sum_{j \in R+} a_{rj}\, x_j + \sum_{j \in R-} a_{rj}\, x_j \geq -1 + f_r$ (15)

Multiply both sides of Equation (15) by the negative number$\frac{f_r}{f_r - 1}$, we have

$(\frac{f_r}{f_r - 1}) \sum_{j \in R-} a_{rj}\, x_j \geq f_r$ (16)

Either of inequalities (15) and (16) holds since in the both cases the left-hand side is non- negative and one of these is greater than or equal tof_r. Thus, any feasible solution to mixed integer programming must satisfies the inequality:

$\sum_{j \in R+} a_{rj}\, x_j + (\frac{fr}{fr - 1}) \sum_{j \in R-} a_{rj}\, x_j \geq f_r$ (17)

Inequality (17) is not satisfied by the optimal solution of the LP problem without integer requirement. This is because by putting $x_j = 0$ for all $j,$ the left-hand side becomes zero and right-hand side become positive. Thus, inequality (17) defines the cut.
Adding a non-negative slack variable, we can rewrite Equation (17) as

$s_g = -f_r + \sum_{j \in R+} a_{rj}\, x_j + (\frac{f_r}{f_r - 1}) \sum_{j \in R-} a_{rj}\, x_j$ (18)

Equation (18) represents the require Gomory's cut
For generating the cut Equation (18) it was assumed that the basic variable $x_r$should take integer value f_r .
But if one or more $x_j, j \in R +$ anda_{rj} $\{\frac{fr}{(f_r - 1)}\}$, j ∈ $R -$ are desired to be as small as possible, we can proceed as follows. The value of the coefficient of x_r can be increased or decreased by an integral amount in Equation (12) so as to get a term with smallest coefficient in Equation (17) in order to reduce the feasibility region as much as possible through cutting planes, the coefficients of integer variables x_r must be as small as possible. The smallest positive coefficient for $x_r$in Equation (12)$\{ f_{rj}, \frac{f_r}{1 - f_r}(1 - f_{rj})\}$is the smaller of the two coefficients would be considered to make the cut penetrate deeper into the original feasible region. A cut is said to be deep if the intercepts of the hyper plane represented by a cut with the
x-axis are larger. Obviously,

$f_r \leq \frac{f_r}{1 - f_r}(1 - f_{rj}); f_r \leq f_rAndf_r > \frac{fr}{1 - fr}(1 = f_{rj});\ f_{rj} > f_r$ (19)

Thus, the new cut can be express as

$$s_g = -f_r + \sum_{j \in R} f^*_{rj} x_j \qquad (20)$$

Where

$$f_{rj} = \begin{cases} a_{rj} & a_{rj} \geq 0 \; and \; x_j \; \text{non}-\text{integer} \\ \left(\dfrac{\text{f}_\text{r}}{f_r - 1}\right) a_{rj} & a_{rj} < 0 \; \text{and } \text{x}_\text{j} \text{ non}-\text{integer} \\ \text{f}_{\text{rj}} & f_{rj} \leq f_r \; and \; x_j \; \text{Non-integer} \\ \left(\dfrac{\text{f}_\text{r}}{f_r - 1}\right)(1 - f_{rj}) & f_{rj} > f_r \; and \; x_j \; \text{Non-integer} \end{cases} \qquad (21)$$

2.5 Fuzzy Integer Linear Programming

The following integer linear programming problem with fuzzy variables:

(P) Maximize $z^{\sim} = cx^{\sim}$

subject to $Ax\sim \leq b\sim$ (22)

$x\sim \geq 0$ and are integers (23)

where the coefficient matrix A = $(a_{ij})_{m*n}$is a nonnegative real crisp matrix, the cost vector
c = $(c_1, ..., c_n)$is nonnegative crisp vector and $x\sim = (x_j^{\sim})_{n*1}$ and
b =$(b_i)_{m*1}$are nonnegative real fuzzy vectors such that $x_j^{\sim}$, $b_i^{\sim} \in F(R)$ for all $1 \leq j \leq n$ and$1 \leq i \leq m$.

Definition 1. A fuzzy vector x ~ is said to be a feasible solution of the problem (P)
if x ~ satisfies (22) and (23).

Definition 2. A feasible solution x ~ of the problem (P) is said to be an optimal
solution of the problem (P) if there exists no feasible u $^{\sim}$ $(u_j^{\sim})_{n*1}$of (P) such that $cu^{\sim} > cx^{\sim}$.

Using these definitions, the arithmetic operations of fuzzy numbers, we can obtain the following result.

(P1) Maximize $z_2 = cx_2$

subject to $Ax_2 \leq b_2$

$x_2 \geq 0$ are integers; (24)

(P2) Maximize $z_1 = cx_1$

subject to $Ax_1 \leq b_1$

$x_1 \geq 0$

$x_1 \leq x_2^o$and are integers (25)

and

(P3) Maximize $z_3 = cx_3$

subject to $Ax_3 \leq b_3$

$x_3 \geq 0$

$x_3 \geq x_2^o$and are integers. (26)

Suppose that x_1^o, x_2^o and $x_3^o$are optimal solutions of the crisp integer linear Programming problems (P2), (P1) and (P3)with optimal values z_2^o, z_1^o *and* z_3^orespectively. This implies that $x_o^{\sim} = (x_1^o, x_2^o, x_3^o)$ is an optimal solution of theProblem (P) with optimal value $z_o^{\sim} = (z_1^o, z_2^o, z_3^o)$.

3.0 Program Description

The program was created using the basic equations (Equations. (1) – (28)) modeled with the Visual Basic. Net object oriented programming language[12]. Accompanied in this work are the program Algorithm shown below and the flowchart. The module of operation of this program is basically in the three phases; input, analysis and output.

3.1 Algorithm (Pseudo code)

```
Start
Select type of Integer problem
        Problem type1: all integer linear programming
        Problem type2: mixed integer linear programming
        Problem type3: integer programming with fuzzy number
If problem type= all integer linear programming then
        Input known parameters (coefficient) of $A \; and \; Cof \; Xi$ using (1)
        Input known parameter (constant) b using (1)
                Compute desired parameter $Z \; and \; Xi$ using (2)
10:     If desired parameter($Xi$) = Integer value then
                $Z$ Is optimal solution
        End if
Elseif desired parameter ($Xi$) = non integer value then
        Select desired parameter$Xi$ with highest fraction $fr$ using (3)
                Generate cutting plane $s_g$ using eqs. (4) – (7)
                Add cutting plane to the bottom of optimal solution
        Compute desired parameter(s) $Z \; and \; Xi$ using (2)
        If desired parameter ($Xi$) = Integer value then        goto 10
        End if
```

```
End
If problem type= mixed integer linear programming then
        Input known parameters of $A\ and\ Cof\ Xi$ using eqs. (8) and (9)
        Input known parameter (constant) b using eqs.(8) and (9)
                Compute desired parameter $Z\ and\ Xi$ using (11)
25:     If desired parameter $(Xi)$ = Integer value then
                $Z$ Is optimal solution
        End if
Elseif$Xi$ = non-integer value then
        Select desired parameter$Xi$ with highest fraction $fr$ using eqs.(12) - (14)
                Test individual fractional part with equation (19)
        Generate cutting plane $s_g$ using eqs. (15) –(18)
        Add cutting plane to the bottom of optimal solution
                Compute desired parameter $Z\ and\ Xi$ using (11)
        If$Xi$ = Integer value then goto 25
        End if
End if
If problem type= Fuzzy integer linear programming then
        Input known parameters of $A\ and\ Cof\ Xi$ using eqs. (22) and (23)
        Input known parameter (constant) b using eqs. (22) and (23)
                Compute desired parameter $Z\ and\ Xi$ using eqs. (24) – (28)
42:     If desired parameter $(Xi)$ = Integer value then
                $Z$ Is optimal solution
        End if
Elseif desired parameter $(Xi)$ = non integer value then
        Select desired parameter$Xi$ with highest fraction $fr$ using (3)
                Generate cutting plane $s_g$ using eqs. (4) – (7)
                Add cutting plane to the bottom of optimal solution
        Compute desired parameter(s) $Z\ and\ Xi$ using (2)
        If$Xi$ = Integer value then                                goto 42
        End if
End if
End program
```

Flow Chart

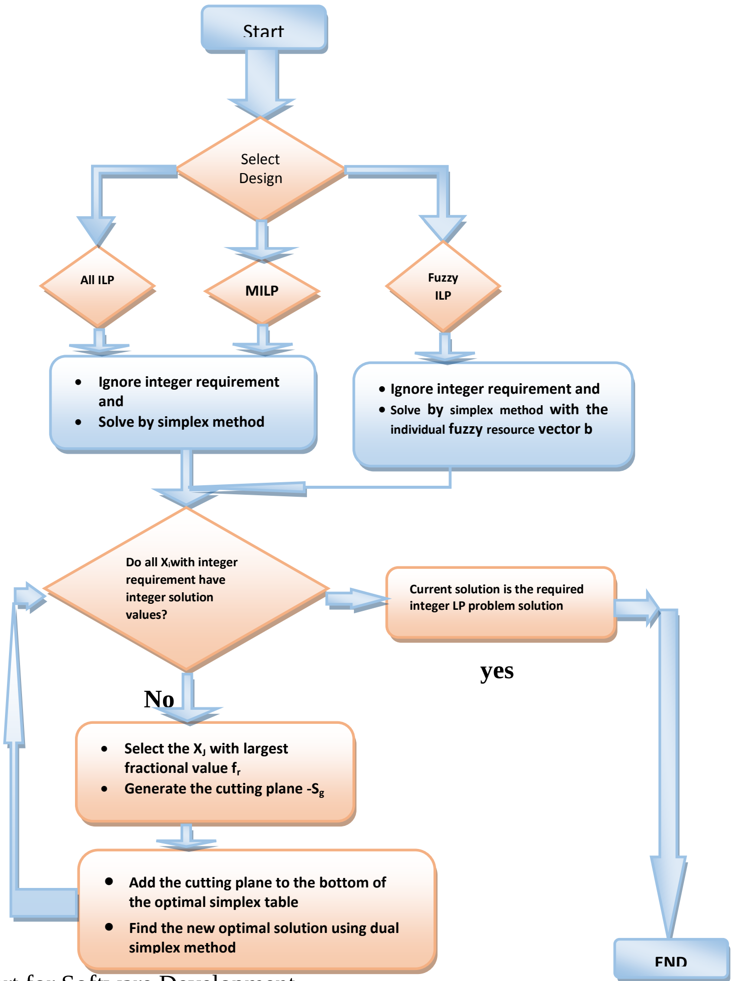

Start
Select Design
All ILP
MILP
Fuzzy ILP
• Ignore integer requirement and
• Solve by simplex method
• Ignore integer requirement and
• Solve by simplex method with the individual fuzzy resource vector b
Do all Xᵢ with integer requirement have integer solution values?
Current solution is the required integer LP problem solution
yes
No
• Select the Xⱼ with largest fractional value fᵣ
• Generate the cutting plane -Sg
• Add the cutting plane to the bottom of the optimal simplex table
• Find the new optimal solution using dual simplex method
END

Fig. 1: Flow chart for Software Development

4 Examples of the Running Program

Illustrating the working efficiency of this software the following examples were considered as shown below

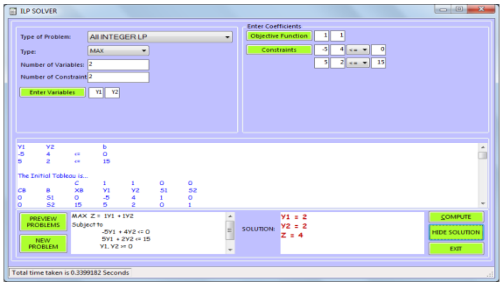
Example 1: **Maximize** $Z = y_1 + y_2$

Subject to $-5y_1 + 4y_2 \le 0$

$5y_1 + 2y_2 \le 15$

$\boldsymbol{x_1, x_2 \geq 0}$

Solution



Type of Problem: All INTEGER LP
Type: MAX
Number of Variables: 2
Number of Constraint: 2
Enter Variables: Y1 Y2

Objective Function: 1 1
Constraints: -5 4 <= 0
5 2 <= 15

```
Y1      Y2              b
-5      4       <=      0
5       2       <=      15

The Initial Tableau is...
                C       1       1       0       0
CB      B       XB      Y1      Y2      S1      S2
0       S1      0       -5      4       1       0
0       S2      15      5       2       0       1
```

```
MAX Z = 1Y1 + 1Y2
Subject to
        -5Y1 + 4Y2 <= 0
        5Y1 + 2Y2 <= 15
        Y1, Y2 >= 0
```

SOLUTION:
```
Y1 = 2
Y2 = 2
Z = 4
```

Total time taken is 0.3399182 Seconds

Fig. 2: Snapshot All ILPProblem

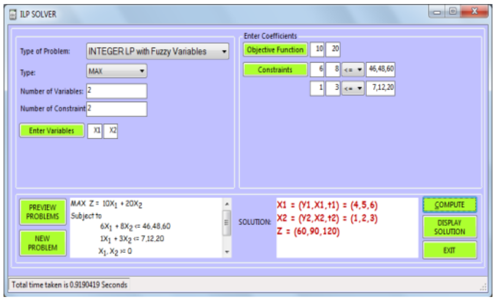
Example2: **Maximize** $\boldsymbol{z} = 10x_1 + 20x_2$

Subject to: $6x_1 + 8x_2 \le (46{,}48{,}60)$

$x_1 + 3x_2 \le (7{,}12{,}20)$

And $\boldsymbol{x_1, x_2 \geq 0}$ ***and are integer***

Solution:



Type of Problem: INTEGER LP with Fuzzy Variables
Type: MAX
Number of Variables: 2
Number of Constraint: 2
Enter Variables: X1 X2

Objective Function: 10 20
Constraints: 6 8 <= 46,48,60
1 3 <= 7,12,20

```
MAX Z = 10X1 + 20X2
Subject to
        6X1 + 8X2 <= 46,48,60
        1X1 + 3X2 <= 7,12,20
        X1, X2 >= 0
```

SOLUTION:
```
X1 = (Y1,X1,t1) = (4,5,6)
X2 = (Y2,X2,t2) = (1,2,3)
Z = (60,90,120)
```

Total time taken is 0.9190419 Seconds

Fig. 3: Snapshot of a Fuzzy decision variable Problem

Example 3: **Maximize**$Z = 3j_1 + j_2 + 3j_3$

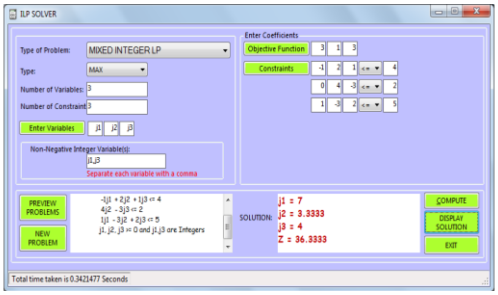
Subject to $-j_1 + 2j_2 + j_3 \le 4$

$4j_2 - 3j_3 \le 2$

$j_1 - 3j_2 + 2j_3 \le 5$

$\boldsymbol{j_2, \geq 0; j_1 , \quad j_3}$ **are non-negative integers**

Solution:



Type of Problem: MIXED INTEGER LP
Type: MAX
Number of Variables: 3
Number of Constraint: 3
Enter Variables: j1 j2 j3
Non-Negative Integer Variable(s): j1,j3
Separate each variable with a comma

Objective Function: 3 1 3
Constraints: -1 2 1 <= 4
0 4 -3 <= 2
1 -3 2 <= 5

```
        -1j1 + 2j2 + 1j3 <= 4
        4j2 - 3j3 <= 2
        1j1 - 3j2 + 2j3 <= 5
        j1, j2, j3 >= 0 and j1,j3 are Integers
```

SOLUTION:
```
j1 = 7
j2 = 3.3333
j3 = 4
Z = 36.3333
```

Total time taken is 0.3421477 Seconds

Fig.4: Snapshot of a Mixed ILP Problem

Other Examples of solved problems both manually and with the software are shown below and tabulated in Table 1.

Example 4: **Maximize**$z = 3M_1 + 6M_2 + 2M_3$

Subject to $4M_1 - 4M_2 \le 5$

$$-M_1 + 6M_2 \leq 5$$
$$-M_1 + M_2 + M_3 \leq 5$$
$\boldsymbol{M_1, M_2, M_3} \geq \mathbf{0}$**and** $\boldsymbol{M_1, M_3}$ non-negative integers

Example 5: **Maximize**$z = 4x_1 + 3x_2$

Subject to: $x_1 + 2x_2 \leq (4,8,12)$

$2x_1 + x_2 \leq (6,9,12)$

And $\boldsymbol{x_1, x_2 \geq 0}$ ***and are integer***

Example 6: **Minimize**$z = 5P_1 + 4P_2$

Subject to $3P_1 + 2P_2 \geq 5$

$2P_1 + 3P_2 \geq 7$

$\boldsymbol{P_1, P_2 \geq 0}$

5. Discussion

The software functionality was tried on a number of practical problems and the solution obtained both from manual computation and software computation was carefully entered in the Table 1.Having tried the software on a number of problems, the software accuracy is evident in the consistency of results with that obtained by manual computation available in established texts; we were able to ascertain the percentage proximity in terms of accuracy of the software computation to the manual computation evaluated as percentage error. The percentage error was obtained as shown in Table 1.

Table 1: Showing Results Obtained By Manual and Software Computation

S/N	**TYPE PROBLEM**	**TYPE**	**SOFTWARE COMPUTATION**	**MANUAL COMPUTATION**	**% ERROR**
1	AILP	Max	y_1=2,y_2=2, z=4	X_1=2,X_2=2, z=4	0.000
2	FILP	Max	X_1=(4,5,6) x_2=(1,2,3) Z=(60,90,120)	X_1=(4,5,6) x_2=(1,2,3) Z=(60,90,120)	0.000
3	MILP	Min	J_1=7,J_2=3.3333,J_3=4,Z=36.3333	X_1=7,X_2=3.3333,X_3=4,Z=36.3333	0.000
4	MILP	Max	M_1=2,M_2=2, M_3=6,Z=24	X_1=0,X_2=1, $X_3$2,Z=7	0.000
5	FILP	Max	X_1=(3,4,5) X_2=(0,1,2) Z=(12,19,28)	X_1=(3,4,5) X_2=(0,1,2) Z=(12,19,28)	0.000
6	AILP	Min	P_1=0.5,p_2=2, Z=10.5	X_1=0.5,X_2=2, Z=10.5	0.000

6. Conclusion

Knowing that manual computation of problems modeled would require some pretty time of about 30 minutes or more, the software was designed to solve speedily and give visual display of run time, it also provides visual display of the solution, so as to easily and speedily interpret the solutions to the problems.

References

(1) Zhanga W. and Nicholsona C.D. (2018), Objective Scaling Ensemble Approach for Integer Linear Programming. Journal of Heuristics. (Researchgate arXiv:1808.10263v1 (math.OC) 11 Aug 2018

(2) Kavun S., Daradkeh Y.B. and Aldhaifallah M. (2015), Method of the Integer Linear Programming. MITTEILUNGEN KLOSTERNEUBURG. 64(2014) XX-XX.www.mitt-klosterneuburg.com

(3) Sokolinskaya I. and Sokolinsky L. B. (2017), On the Solution of Linear Programming Problems in the Age of Big Data. DOI: 10.1007/978-3-319-67035- 5_7. https://www.researchgate.net/publication/318107000

(4) Sharma, J.K. (2010), "Operations Research, Theory and Application", 4th Edition Macmillan Publishers, India Ltd. ISBN 13: 970-0230-63885-3

(5) Sudhagar C. *and* Ganesan K. (2010), "Fuzzy Integer Linear Programming with fuzzy Decision variables". Applied Mathematical Sciences, vol. 4, No. 70 (2010), pp. 3493-3502.

(6) Sudhagar C. and Ganesan K. (2009) "Ranking fuzzy numbers based on their scores", Proc. of International Conference on Mathematical Methods and Computation, Tiruchirappalli, India (2009), pp. 383 – 392

(7) Pandian, P. and Jayalakshmi, M. (2010), "A New Method for Solving Integer Linear Programming Problems with Fuzzy variables". *Applied mathematical sciences* vol. (4) No. 20, (2010) pp. 997 -1004.

(8) Sharma,S.C. and Abha, B. (2011), "Mixed Integer Linear Programming Problem with Fuzzy variables". *International Journal of Mathematical Archives-2(4), (2011) pp.438-443. ISSN: 2229-5046*

(9) Chinneck J. W. (2004), "Practical Optimization: a Gentle Introduction". http://www.sce.carleton.ca/faculty/chinneck/po.html

(10) Nasseri, S.H. and Sohrabi, M. (2010), "Solving fuzzy Linear Programming by using Revised Tsao's Method". *Austrialian Journal of Basic and Applied Sciences*, 4(10): pp. 4592 – 4600 (2010). ISSN 1991 – 8178

(11) Nasseri, S.H., Ardil, E. Y. A. and Zaefarian, R. (2005), "Simplex Method for solving Linear Programming Problems with Fuzzy Numbers". *World academy of science, Engineering and Technology*, pp. 10 (2005)

(12) Paul D. and Harvey D. (2011), Visual Basic 2010. How to program. By pearson Education Inc., Publishing as Prentice Hall Upper Saddle River, New Jersey 07458 pp. 104-079.