Solving interval linear programming problem using generalized interval LU decomposition method

K Nirmala¹, T Nirmala² and K Ganesan³
Department of Mathematics, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur – 603 203, Chennai, Tamilnadu, India.

E-mail: nirmalak@srmist.edu.in , nirmalat@srmist.edu.in, ganesank@srmist.edu.in

Abstract. In this paper, we propose a method for solving interval linear programming problem by using generalized interval LU decomposition method based on Kaucher arithmetic for generalized interval numbers. Numerical example is also been provided.

1. Introduction
Linear programming problem (LPP) is used in the field of optimization. Uncertainties are common in real world situations. To tackle this uncertainty in Linear Programming, Interval Linear Programming Problem (ILPP) has been introduced. ILPP is used to characterize uncertain parameters in decision making problems. ILPP is a LP problem (that has an objective function subject to the constraints) whose co-efficient are generalized interval numbers.

Elif Garajova [1] had used some transformation on ILPP suitably and converted the problem as ordinary LP. Huiling Duan [2] used the technique of converting ILP as ordinary LP by finding the bounds (lower and upper) of an objective function. MATLAB programming has been used to solve ordinary LPP. Jeyalaksmi [3] split interval linear programming into upper LP, middle LP and lower LP problem and solved the three classical linear problem by using break and bound method which becomes more time consuming. H. Mishmast Nehi et al. [4] have discussed few methods of solving ILPP.

LU decomposition of a given square matrix factorizes the matrix into upper triangular matrix (by Gauss Elimination method) and lower triangular matrix (with diagonal elements as 1), such that the product of these two triangular matrices arrives as a same matrix and it is easy to do manipulations in the process of finding the solution. This method of factorizing a matrix into two triangular matrices has its uses in various fields of engineering especially to solve discrete dynamical system problems, to solve complex circuit designing problems. It is also used to solve linear system of equations, to find inverses, determinant of a matrix. Abdulraheem M Z et al. [5] showed the effectiveness of LU decomposition method to solve a system of classical linear equations using MATLAB. Nirmala et.al [6] has given an algorithm for LU factorization for an interval matrix which is invertible by using Gaussian elimination method based on generalized interval arithmetic.

Kaucher introduced the concept of dual which in turn lead a new arithmetic for the generalized intervals. Alexandre Goldsztejn and Gilles Chabert [7] introduced a new method of LU decomposition for a generalized interval matrix by using Kaucher arithmetic and also by using Gaussian Elimination method. KaucherPy, an interval package can also be used to simplify the calculations.
In this paper, we propose an algorithm for solving ILPP. Here we convert the problem as a system of interval linear inequalities and arrive as an interval matrix. The interval matrix in this system can be split into lower triangular and upper triangular interval matrices by using the generalized interval LU decomposition method [8]. Further, LU decomposition technique has been used for finding the solution of interval linear system.

This paper comprises of the following: In section 2, we have discussed the basic concepts of generalized interval and its arithmetic proposed by Kaucher. The aim of this section is to introduce some notations and notions from Nirmala et al. [9] which are of useful in our further considerations. In section 3, we recall the standard form of interval linear programming problem and conversion of ILPP to an interval matrix inequality system. In section 4, we have proposed an algorithm for finding the solution of interval linear programming problem (ILP). In section 5, Numerical example has been provided.

2. Preliminaries

Let $\mathbb{IR} = \{\bar{a} = [a, a]: a \leq a \text{ and } a, a \in \mathbb{R}\}$ denotes the set of all proper intervals and $\mathbb{IR} = \{\bar{a} = [a, a]: a > a \text{ and } a, a \in \mathbb{R}\}$ denotes the set of all improper intervals on the real line $\mathbb{R}$.

If $\bar{a}$ and $\bar{a}$ then are equal, then ‘interval’ will become a real number. The term “interval” can be used for an “interval number” and vice-versa. Let $D = \mathbb{IR} \cup \mathbb{IR} = \{[a, a]: a, a \in \mathbb{R}\}$ be the set of all generalised intervals (both proper and improper). The set $D$ forms a group under addition and multiplication operations for zero free intervals, with the inclusion monotonicity. The “dual” is an important monadic operator which was introduced by Kaucher [9] that reverses the end-points of the interval and expresses an element to element symmetry between proper and improper intervals in $D$. For example, if $\bar{a} = [a, a] \in D$, its dual is defined by dual $\bar{a} = [a, a] = [a, a]$. The opposite of an interval $\bar{a} = [a, a]$ is opp $[a, a] = [a, a]$ which is the additive inverse of $[a, a]$ and $\frac{1}{a} = \frac{1}{a}$ is the multiplicative inverse of $[a, a]$, provided $0 \notin [a, a]$. That is,

\[ \bar{a} + (\text{dual } \bar{a}) = \bar{a} - \text{dual}(\bar{a}) = [a, a] - \text{dual}([a, a]) = [a, a] - [a, a] = [a - a, a - a] = [0, 0] \]

and $\bar{a} \times \left( \frac{1}{\text{dual}(\bar{a})} \right) = [a, a] \times \frac{1}{[a, a]} = [a, a] \times \left[ \frac{1}{a}, \frac{1}{a} \right] = \left[ \frac{a}{a}, \frac{a}{a} \right] = [1, 1]$.

2.1. Kaucher interval arithmetic

The arithmetic was introduced by Kaucher for generalized intervals which is also known as Kaucher interval arithmetic to overcome the disadvantages of classical interval arithmetic. If $\bar{a} = [a, a], \bar{b} = [b, b] \in D$, are any two interval numbers, then

(i) Addition: $\bar{a} + \bar{b} = [a, a] + [b, b] = [a + b, a + b]$

(ii) Subtraction: $\bar{a} - \bar{b} = [a, a] - [b, b] = [a - b, a - b]$

(iii) Multiplication:

Let $P = \{\bar{a} = [a, a] \in D / a \geq 0, a \geq 0\}$ be a set of non-negative intervals, $-P = \{\bar{a} = [a, a] \in D / a \leq 0, a \leq 0\}$ be a set of non-positive intervals, $Z = \{\bar{a} = [a, a] \in D / a < 0 < a\}$ be a set of intervals containing zero and dual $Z = \{\bar{a} = [a, a] \in D / a < 0 < a\}$ be a set of intervals containing zero.
Then the multiplication of two intervals is defined as

\[
\tilde{a} \tilde{b} = [\min\{a b, a \bar{b}, \bar{a} b, \bar{a} \bar{b}\}, \max\{a b, a \bar{b}, \bar{a} b, \bar{a} \bar{b}\}]
\]

(iv) Division: \(1 + \bar{a} = \frac{1}{\tilde{a}} = \frac{1}{[a, \bar{a}]} = \left[ \frac{1}{\bar{a}}, \frac{1}{a} \right]\) where \(0 \notin [a, \bar{a}]\)

3. Standard form of Interval Linear Programming Problem

A standard form of an ILLP is to maximize the given objective function with respect to the constraints:

\[
\text{Maximize } Z = \sum_{i=1}^{n} c_i \tilde{x}_i
\]

such that

\[
\sum_{j=1}^{n} \tilde{a}_{ij} \tilde{x}_j \leq \tilde{b}_i, \quad i = 1, \ldots, m
\]

\[
\tilde{x}_j \leq 0, \quad j = 1, \ldots, n
\]

where \(\tilde{c}_j = [c_j, \bar{c}_j]\), \(\tilde{x}_j = [\tilde{x}_j, \bar{\tilde{x}}_j]\), \(\tilde{b}_j = [\bar{b}_j, \tilde{b}_j]\)

and \(\tilde{a}_{ij} = [\tilde{a}_{ij}, \bar{a}_{ij}]\)

3.1. Construction of interval linear programming problem as interval matrix inequality system

Consider the interval linear programming problem as

\[
\text{Maximize } \tilde{Z} = \tilde{c}_1 \tilde{x}_1 + \tilde{c}_2 \tilde{x}_2 + \tilde{c}_3 \tilde{x}_3 + \ldots + \tilde{c}_{n-1} \tilde{x}_{n-1}
\]

subject to the constraints

\[
\tilde{a}_{21} \tilde{x}_1 + \tilde{a}_{31} \tilde{x}_1 + \tilde{a}_{22} \tilde{x}_2 + \tilde{a}_{32} \tilde{x}_2 + \tilde{a}_{23} \tilde{x}_3 + \ldots + \tilde{a}_{3,n-1} \tilde{x}_{n-1} \leq \tilde{b}_2
\]

\[
\tilde{a}_{31} \tilde{x}_1 + \tilde{a}_{32} \tilde{x}_2 + \tilde{a}_{33} \tilde{x}_3 + \ldots + \tilde{a}_{3,n-1} \tilde{x}_{n-1} \leq \tilde{b}_3
\]

where \(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1} \geq 0\).

To find \(\tilde{Z}\), we will rewrite the above ILLP problem in the form of less than or equal (\(\leq\)) inequalities as follows:
\[-\tilde{c}_1 \tilde{x}_1 - \tilde{c}_2 \tilde{x}_2 - \tilde{c}_3 \tilde{x}_3 - \ldots - \tilde{c}_{n-1} \tilde{x}_{n-1} + \tilde{Z} \leq \tilde{0} \]
\[\tilde{a}_{21} \tilde{x}_1 + \tilde{a}_{22} \tilde{x}_2 + \tilde{a}_{23} \tilde{x}_3 + \ldots + \tilde{a}_{2,n-1} \tilde{x}_{n-1} \leq \tilde{b}_2 \]
\[\tilde{a}_{31} \tilde{x}_1 + \tilde{a}_{32} \tilde{x}_2 + \tilde{a}_{33} \tilde{x}_3 + \ldots + \tilde{a}_{3,n-1} \tilde{x}_{n-1} \leq \tilde{b}_3 \]
\[\vdots \]
\[\tilde{a}_{n1} \tilde{x}_1 + \tilde{a}_{n2} \tilde{x}_2 + \tilde{a}_{n3} \tilde{x}_3 + \ldots + \tilde{a}_{n,n-1} \tilde{x}_{n-1} \leq \tilde{b}_n \]

where \(-\tilde{x}_1, -\tilde{x}_2, \ldots, -\tilde{x}_{n-1}, -\tilde{Z} \leq \tilde{0}\).

Note that now, in the above system, the objective function is included as one of the constraint and \(\tilde{Z}\) is now treated as a variable. Also note, here the no. of inequalities is as same as the no. of variables. Hence we get the interval matrix form for ILPP as \(\tilde{A}\tilde{X} \leq \tilde{B}\) where

\[
\tilde{A} = \begin{bmatrix}
\tilde{a}_{11} & \tilde{a}_{12} & \ldots & \tilde{a}_{1,n-1} & 1 \\
\tilde{a}_{21} & \tilde{a}_{22} & \ldots & \tilde{a}_{2,n-1} & 0 \\
\tilde{a}_{31} & \tilde{a}_{32} & \ldots & \tilde{a}_{3,n-1} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\tilde{a}_{n1} & \tilde{a}_{n2} & \ldots & \tilde{a}_{n,n-1} & 0
\end{bmatrix}, \quad \tilde{a}_{ii} = -\tilde{c}_i \text{ for } 1 \leq i \leq n-1
\]

\[
\tilde{X} = \begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\vdots \\
\tilde{x}_{n-1} \\
\tilde{Z}
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
\tilde{0} \\
\tilde{b}_2 \\
\tilde{b}_3 \\
\vdots \\
\tilde{b}_n
\end{bmatrix}
\]

Further, by using generalized interval LU decomposition method proposed by Alexandre Goldsztejn [7] using Kaucher interval arithmetic, the arrived interval matrix \(\tilde{A}\) can be decomposed into lower and upper triangular interval matrix [10,11].

4. Algorithm to solve Interval Linear Programming Problem

The stepwise procedure to solve maximization of ILPP is as follows:

Step 1: Rewrite the given ILPP including the objective function in the form of “less than or equal to inequalities”.

Step 2: Check whether the no. of inequalities is equal to the no. of variables. If so, continue to step 4.

Step 3: Introduce a dummy variables or dummy inequalities to form the system with equal number of inequalities & variables.

Case 1: If the no. of inequalities is less than the no. of variables, add the inequalities in the system till they become equal. These inequalities can be formed by multiplying any non-zero coefficient with the variable in one of the inequality.

Case 2: If the no. of inequalities is greater than no. of variables, then introduce that much of dummy (slack) variables in the appropriate inequalities.

Case 3: If there is a zero row in upper triangular interval matrix \(\tilde{U}\), then LU decomposition method fails.

Step 4: Consider the matrix system of interval linear equations \(\tilde{A}\tilde{X} = \tilde{B}\).
Step 5: Find lower triangular interval matrix $\tilde{L}$ and upper triangular interval matrix $\tilde{U}$ such that $\tilde{A} = \tilde{L}\tilde{U}$ by the method of generalized interval LU decomposition.

Step 6: Let $\tilde{Y} = \tilde{U}\tilde{X}$ and solve $\tilde{Y}$ from $\tilde{L}\tilde{Y} = \tilde{B}$ by using forward substitution method.

Step 7: Solving $\tilde{U}\tilde{X} = \tilde{Y}$ for $\tilde{X}$ by backward substitution method, we get the solution for system of interval linear programming problem.

5. Numerical Example

Example 5.1:

Consider the following ILP problem discussed by MishmastNehi H, Ashayerinasab H.A. and Allahdad M [4]

Maximise $\tilde{Z} = [3, 3.5]x_1 - [1, 1.2]x_2$
subject to $[1, 1.1]x_1 + [1.6, 1.8]x_2 \leq [11.6, 12]$
$[3, 4]x_1 - [2, 3]x_2 \leq [5, 7],$
where $x_1, x_2 \geq 0$

By the proposed algorithm, the given interval linear programming problem can be formulated as

$$
\begin{pmatrix}
-3.5 & -3 & 1.12 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
\tilde{x}_1
\end{pmatrix}
\leq
\begin{pmatrix}
0
\end{pmatrix}

[1, 1.1]
[1.6, 1.8]
[3, 4]

\begin{pmatrix}
1
0
0
\end{pmatrix}

\begin{pmatrix}
\tilde{Z}
\end{pmatrix}
\leq
\begin{pmatrix}
11.6, 12
5, 7
\end{pmatrix}

Consider the interval linear system as $\tilde{A}\tilde{X} = \tilde{B}$, and by using the generalised interval LU decomposition method, we represent $\tilde{A}$ as $\tilde{A} = \tilde{L}\tilde{U}$ where

\[
\tilde{L} = \begin{pmatrix}
1 & 0 & 0 \\
-0.3142 & 0.3333 & 0 \\
-1.1428 & -0.7634 & -0.5058
\end{pmatrix}
\begin{pmatrix}
1 & 1 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

and

\[
\tilde{U} = \begin{pmatrix}
-3.5 & -3 & 1.12 & 1 & 1 \\
0 & 0 & 1.97704 & 2.1333 & 0.3142 & 0.3333 \\
0 & 0 & 0 & 0.8864 & 0.8411
\end{pmatrix}
\]

Now applying the LU decomposition technique that is, $\tilde{A}\tilde{X} = \tilde{B}$, solve $\tilde{L}\tilde{Y} = \tilde{B}$, by forward substitution method, we get

$\tilde{Y} = ([0, 0], [11.6, 12], [14.1608, 12.8673])^T$

By backward substitution, solving $\tilde{U}\tilde{X} = \tilde{Y}$, we get

$\tilde{X} = ([3.5336, 3.86], [4.9729, 4.324], [5.6280, 8.322])^T$

Thus the solution set is $\tilde{x}_1 = [3.5336, 3.86]$ and $\tilde{x}_2 = [4.9729, 4.324]$

$\therefore \tilde{Z} = [5.628, 8.322]$
6. Conclusion
By using the arithmetic operation introduced by Kaucher on generalized interval numbers, we have given an algorithm for solving interval linear programming problem by applying the concept of LU decomposition method. It can be seen that number of computations are less compared to other known techniques. Numerical example is also provided to show the efficiency of the given method.

7. References
[1] Elif Garajova, Milan Hladik and Miroslav Rada 2019 Central European Journal of Operations Research 27 601-614
[2] Huiling Duan and Tao Peng 2014 International Mathematical Forum 9 1333 – 1340
[3] Jayalakshmi M 2018 International Journal of Pure and Applied Mathematics 118 363-370
[4] MishmastNehi H and Ashayerinasab H.A, Allahdad M 2018 Oper Res Int J
[5] Abdulraheem MZ and Mohammad K 2018 Journal of Applied and Computational Mathematics 7 1-5
[6] Nirmala T, Dutta D, Kushwaha H S and Ganesan K 2016 Global journal of Pure and Applied Mathematics 12 414-423
[7] Alexandre Goldsztejn and Gilles Chabert 2007 Springer- Verlag Berlin Heidelberg 312-319
[8] Kaucher E 1980 Computing Suppl 2, 33-49
[9] Nirmala T, Dutta D, Kushwaha H S and Ganesan K 2011 Applied Mathematical Sciences 5 607-624
[10] Chinchole S MandBhadane A P2014 International Journal of Emerging Technology and Advanced Engineering 4 176-180
[11] DirceuMaraschin , LucasM.Tortelli, Alice F. Finger and Aline B. Loret 2019 International Journal of Latest Engineering Research and Applications 4 34-45