AN ALGORITHM FOR INTEGER LINEAR BILEVEL PROGRAMMING PROBLEMS USING TRANSFORMATION

Anuradha Sharma

(Associate Professor Delhi University)

Abstract—In this paper an algorithm is developed to solve the integer linear bilevel programming (ILBP) problem. The ILBP is transferred into a single level programming which can be solved by the algorithm proposed here. This algorithm deals by changing the randomly generated initial population into an initial population satisfying the constraints to improve the efficiency of the algorithm to deal with the constrains.

Keywords- Linear bilevel programming; Integer programming; genetic algorithm; Kuhn-Tucker conditions.

I. INTRODUCTION

The mathematical programming models deal with a single decision maker and a single objective function which are used for centralized planning systems. Bilevel programming deals with decision processes involving two decision makers with a hierarchical structure. The first level decision maker is termed as the leader and the decision maker at the second level pertains to the objective of the the follower. Each decision maker optimizes his own objective function and is affected by the actions of the other planner. Integer Linear bilevel programming (ILBP) problem is a special case of bilevel programming problems in which the objective functions as well as the constraints are all linear, and it can be formulated as follows[3]:

(ILBP) \[\text{max } f(x,y) = p_1 x + q_1 y\]

where \(y\) solves

\[\text{max } g(x,y) = p_2 x + q_2 y\] \hspace{1cm} (1)

\[\text{s.t. } A_1 x + A_2 y \leq b,\]

\[x, y \geq 0,\]

\[x, y \text{ integers}\]
where \( f, g \) are the objective functions of the leader and follower, respectively; \( x \) is an \( n_1 \)-dimensional column vector (the variables under the control of first level) and \( y \) is an \( n_2 \)-dimensional column vector (the variables under the control of second level); \( p_1 \) and \( p_2 \) are \( n_1 \)-dimensional row vectors, \( q_1 \) and \( q_2 \) are \( n_2 \)-dimensional row vectors; \( A_1 \) is an \( m \times n_1 \)-matrix and \( A_2 \) is an \( m \times n_1 \)-matrix and \( b \) is an \( m \)-dimensional column vector. We assume that the polyhedron \( S \) defined by the common constraints is nonempty and bounded.

Many researchers tackled the ILBP problem by presenting both theoretical results and application [4], [5]. The algorithmic approaches developed so far for ILBP problem can be classified into four categories [5]:

1. meta heuristics approach
2. Kuhn-Tucker conditions approach
3. fuzzy approach
4. vertex enumeration approach

The ILBP is neither continuous everywhere nor convex because the objective function of the first level problem is decided by the solution function of the second level problem which is neither linear nor differential. J. F. Bard [1] proved that ILBP is a NP-Hard problem. The algorithms proposed include genetic algorithms [2], [5], [7], simulated annealing algorithms [9], particle swarm optimization (PSO) [6] etc.

In this paper an algorithm is developed for the ILBP problem. DE proposed by Storn and Price [10] optimized real valued multi-modal objective functions. Besides its good convergence properties the algorithm is very simple to implement and finds solution to the problem.

II. ALGORITHMIC DEVELOPMENT

The algorithm proposed in this paper is a parallel direct search method which utilizes NP D-dimensional parameter vectors

\[
p_{iG} = (p_{i1G}, p_{i2G}, \ldots, p_{iDG}) \quad i=1,2,\ldots,\text{NP}
\]

as a population for each generation \( G \). To be more precise, the basic approach in the algorithm [8],[10] is described in the following steps:

Step 1: The initial vector population is chosen randomly from within a user-defined...
range and should cover the entire parameter space.

**Step 2: Mutation.** In this step, the mutated vectors \( (i = 1, 2, \ldots, NP) \) are generated according to

\[
g_{i}^{G+1} = p_{r1}^{G} + F \cdot (p_{r2}^{G} - p_{r3}^{G})
\]

with random indexes \( r1, r2, r3 \in \{1, 2, \ldots, NP\} \), integer, mutually different and \( F > 0 \). The randomly chosen integers \( r1, r2 \) and \( r3 \) are also chosen to be different from the running index \( i \), so that \( NP \) must be greater or equal to four to allow for this condition. Weighting factor \( F \in [0, 2] \) is a real and constant factor which controls the amplification of the differential variation \( (p_{r2}^{G} - p_{r3}^{G}) \).

**Step 3:** To increase the diversity of the perturbed parameter vectors, crossover is introduced. To this end, the trial vector

\[
 h_{i}^{G+1} = (h_{i1}^{G+1}, h_{i2}^{G+1}, \ldots, h_{iD}^{G+1})
\]

is formed, where

\[
h_{ij}^{G+1} = \begin{cases} 
g_{ij}^{G}, & \text{if } (\text{randb}(j) \leq CR) \text{ or } j = \text{rnbr}(i) \\
p_{ij}^{G}, & \text{otherwise}, \end{cases}
\]

In (5), \( \text{randb}(j) \in [0, 1] \) is the \( j \)-th evaluation of a uniform random number generator with outcome. \( CR \in [0, 1] \) is the crossover constant which has to be determined by the user. \( \text{rnbr}(i) \in \{1, 2, \ldots, D\} \) is a randomly chosen index which ensures that \( h_{ij}^{G+1} \) gets at least one parameter from \( g_{ij}^{G+1} \).

**Step 4:** To decide whether or not it should become a member of generation \( G+1 \), the trial vector \( g_{ij}^{G+1} \) is compared to the target vector \( p_{i}^{G} \) using the greedy criterion.

**Step 5:** Stop the algorithm if the condition is satisfied or iteration number is larger than the maximal iteration number. Then the best generated solution, which can be obtained by the current best vector kept in all iterations in the earliest time, is reported as the solution for the LBP problem by the proposed algorithm.

### III. REDUCING ILBP PROBLEM TO A SINGLE LEVEL PROGRAM

At first the ILBP problem can be transferred into a single level programming of the form

\[
(P1) \quad \max f(x,y) = p_{1}x + q_{1}y \\
\]

\[
x
\]

\[
s.t. \quad A_{1}x + A_{2}y + u = b \\
\]

\[
z A_{2} - v = d_{2},
\]

\[\text{(6)}\]
\[ uz = 0, \ n y = 0, \]
\[ x, y , z , u, v \geq 0. \]
\[ x, y \text{ integer} \]

Problem (6) is transformed into the following problem

\[
\text{(P2)} \quad \max p_1 x \\
\text{s.t.} \quad A_1 x = b - A_2 y^G - u^G \\
\quad z'A' - v' = d_2 \\
\quad x, z', v' \geq 0 \]
\[ x \text{ integer} \quad (8) \]

In problem (8), \( z' \) and \( v' \) are the variables of \( z \) and \( v \) that are greater than or equal to zero. Also, \( A'_2 \) is the rows of \( A_2 \) which is associated with the variables \( z' \).

Furthermore, problem (8) can be decomposed into two separate problems as follows:

\[
\text{(P3)} \quad z'A'_2 - v' = d_2 \\
\quad z', v' \geq 0 \quad (9) \\
\]

\[
\text{(P4)} \quad \max p_1 x \\
\text{s.t.} \quad A_1 x = b - A_2 y^G - u^G \quad (10) \\
\quad x \geq 0 \\
\]

Such decomposition is allowed, because problems (9) and (10) do not have common variables. Problem (9) is solved first. If this problem is infeasible, then the target vector is not achievable, otherwise problem (10) is solved. If this problem is infeasible, then \( p_1^G \) is unaccessible.
IV. NUMERICAL EXAMPLE

Max $-2x + 11y + 2$

Max $-x - 3y + 3$

s.t. $x - 2y \leq 4$

$2x - y \leq 24$

$3x + 4y \leq 96$

$x + 7y \leq 126$

$-4x + 5y \leq 65$

$x + 4y \geq 8$

$x, y \geq 0$

The best solution for the method proposed in this paper is given by $(x, y) = (17.4545, 10.9091)$ and objective function value is $f = 85.0909$ and $g = -50.181$.

IV. CONCLUSIONS

This procedure designs the algorithm for solving ILBP problems in which the optimal solution of the lower level problem is dependent on the upper level problem. The solution approach makes use of simple transformation method which takes up less computational work.

ACKNOWLEDGEMENTS: The author is highly grateful to the referees for their valuable suggestions which helped in the improvement of the paper.

REFERENCES


