Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2024.00000

Research Article

# A novel alternative algorithm to find all multiple solutions of general integer linear program

Kadriye ŞİMŞEK ALAN[1]

[1]Department of Mathematical Engineering, Yildiz Technical University, Istanbul, 34220, Türkiye

ARTICLE INFO

*Article history*
Received: 13 June 2023
Revised: 23 September 2023
Accepted: 15 November 2023

**Keywords:**
Alternative Solutions; Integer Linear Programming; Multiple Solutions; Linear Diophantine Equations; Optimal Hyperplane; Optimal Solution

ABSTRACT

Integer linear programming (ILP) is often used to model and solve real-life problems. In practice, alternative solutions are very useful as they significantly increase flexibility for the decision-maker. In this study, an alternative method based on parameterization obtained from the Diophantine equation is developed to find all alternative solutions to ILP problems, and an easy-to-implement, efficient, and reliable algorithm is presented. The proposed method was used without being affected by the number of variables and constraints in the problem. Numerical examples are presented to demonstrate the usefulness of the proposed method. In addition, these examples are coded in the MAPLE programming language according to the proposed algorithm.

**Cite this article as:** Şimşek Alan K. A novel alternative algorithm to find all multiple solutions of general integer linear program. Sigma J Eng Nat Sci 2024;42(5):1532−1541.

## INTRODUCTION

Operations research, mathematical physics, numerical analysis, and differential equations are vital fields dedicated to solving complex mathematical problems. These disciplines involve formulating intricate real-world issues into mathematical models, analyzing them, and then working toward their resolution and optimization. The solutions derived from these efforts have wide-ranging applications in industries, economics, sciences, and various other fields. The cross-pollination between these domains also leads to the emergence of novel mathematical theories [1-5]. Integer Programming is a linear programming problem in which some or all of the variables take integer (or discrete) values by adding the condition of being integer to the linear programming model. These applications are extensively reviewed in [6-10].

Although the solution of the ILP problem seems easier at first glance than the solution of the LP problem, it is much more difficult to reach the optimal solution in the IP model. Optimization approaches in IP models can basically be classified as stochastic and deterministic. Stochastic methods are easy to implement and require little prior knowledge to solve the optimization problem [10]. The study [11] proposed a simulated annealing (SA) metaheuristic method, incorporating two constructive heuristic approaches: the Clark and Wright (CW) algorithm, as well as the nearest neighborhood (NN) search algorithm. The quality of the final solutions was directly associated with the quality of the initial solutions. Therefore, two

***Corresponding author.**
*E-mail address: ksimsek@yildiz.edu.tr

*This paper was recommended for publication in revised form by Regional Editor Ahmet Selim Dalkilic* heuristics were recommended for generating initial solutions Stochastic methods are suitable for problems where function evaluations are inexpensive but cannot guarantee rigorous global optimality. In addition, as the problem size increases, the probability of finding the global solution decreases. For this reason, it seems very reasonable to prefer deterministic methods in the solution of IP problems. In deterministic methods, Gomory [12] first developed a method called Gomory cutting plane for solving ILP problems. The method starts with the simplex solution of an LP problem. If the solution at the starting point is an integer value, it is the optimal solution. If not, an integer solution is sought by adding a linear constraint created on the fractional variables in the solution. Land and Doig [13] developed the branch-boundary method to solve IP problems. The LP-based branch and bound method is the most widely used algorithm to solve MILP problems [14]. Biegler and Grossmann have generally reviewed many deterministic methods for IP problems [10,15]. Studies [16-21] can be given as examples of studies in this field. Integer Linear Programming problems and Linear Diophantine equations play a significant role in the processes of mathematical modeling and problem optimization. The deep relationship between these two fields encompasses critical practical applications in various areas, ranging from the efficient utilization of natural resources to the management of production processes.

In the study [22], a method was developed for solving ILP problems using a parameterization obtained from a linear Diophantine equation to solve two-variable ILP problems, and an alternative algorithm was presented. Later, this method was developed to solve ILP problems with three variables [23], and four variables [24] respectively. Brimkov et al.[25] presented results connected with the theory of computation over real numbers, recently developed by Blum, Shub, and Smale. The subjects of the investigation were variations of the well-known ILP Problem and the Linear Diophantine Equation, where the coefficients were real numbers and we sought an integer solution. Results regarding the solvability and complexity of these problems were demonstrated, and algorithms for their solution were developed in the presence of additional information. In the paper [26], the lattice basis reduction algorithm, which was a general method for solving linear Diophantine equations with bounds on the variables, was used. Computational results were demonstrated by solving both the feasibility and optimization versions of market split instances, involving up to 7 equations and 60 variables. Additionally, various branching strategies were discussed, and their effects on the number of enumerated nodes were evaluated. In [27], a representation of the solutions of a system of m linear integer inequalities in n variables, where m was less than or equal to n, with a full-rank coefficient matrix, was given by using the ABS algorithm for solving linear Diophantine equations. This result was applied to solve linear integer programming problems with m being less than or equal to n inequalities. In the study [28], structural results on solutions to the Diophantine system that had the smallest number of nonzero entries were presented. The tools employed were of algebraic and number theoretic nature, including the utilization of Siegel's lemma, generating functions, and commutative algebra. In the study [29], an integer programming model was developed to assist in determining the segmentation of assembly lines for creating efficient and economical assembly lines. Additionally, worker timetabling decisions can be made by solving the same mathematical model.

Simsek Alan [30] presented an algorithm for solving ILP problems with a large number of integer variables using simple computer programming. Alonso-Pecina et al. [31] used threshold accepting and tabu search metaheuristic methods to solve the Label Printing Problem. These heuristic methods significantly improved the results with the assistance of Hill Climbing with a double neighborhood. Recently, the importance of integer linear programming has continued to grow, and significant advancements have been made in this field. In the study [32], the researchers introduced the first fast and exact solver for the Minimum Flow Decomposition (MFD) problem on acyclic flow networks, relying on Integer Linear Programming (ILP). Their approach's cornerstone was the encoding of exponentially many solution paths using only a quadratic number of variables. Furthermore, they broadened their ILP formulation to encompass numerous practical variations, including the integration of longer or paired-end reads, as well as the minimization of flow errors. In [33], due to the complexity of the problems stated in [31], an integer linear programming model was proposed for obtaining optimal solutions for instances where optimal solutions could not be found. Aurricchio et al.[34] proposed an integer linear programming model that provides solutions for asymmetric rhythm tiling with a specific rhythm A. They demonstrated how this model can be used as an iterative algorithm to find all the rhythms that tile with a given rhythm A for a given period n. Additionally, they efficiently checked the necessity of the Coven-Meyerowitz condition (T2). In conclusion, they conducted various experiments to validate the time efficiency of the model.

If there is more than one optimal solution that satisfies both the objective function and the constraints, they are alternative optimal solutions to an ILP problem. It is beneficial for the decision maker to choose among many solutions without any deterioration in the objective function, as it gives the decision maker many advantages over its competitors in practice. Therefore, it is important to find all alternative solutions to an ILP problem. However, in Integer Programming problems, the feasible solution region is neither continuous nor convex. The feasible solution points of the problem may not fall onto the endpoints of the solution region or even on the boundaries of the region. Therefore, linear programming algorithms based on the logic of searching for solutions at the endpoints cannot be directly applied to Integer Programming (IP) problems. Consequently, it is crucial to limit the optimal solution region of the LIP problem. In this study, for this purpose, a parameterization has been developed to help us determine the feasible solution region containing all optimal solutions. A method is devised to find all alternative solutions to a given ILP problem by utilizing this parametrization. Additionally, an easily applicable, practical algorithm is presented. Therefore, this study proposes an alternative method to find all alternative solutions to a pure integer ILP problem. First, using the parametrization from the Diophantine equation, the original ILP problem is reformulated as another ILP problem that can be easily solved by simple mathematical programming. Then, an algorithm is developed that finds all alternative optimal solutions. In addition, some numerical examples are given and these examples given using this algorithm are coded in the MAPLE programming language.

The rest paper is organized as follows: Required information is presented in Section 2. The solution method is handled in Section 3. The proposed method is given in Section 4. Our numerical examples and conclusions are given in Section 5 and Section 6, respectively.

## Preliminaries

In this section, the notation used in this study and the necessary information for obtaining all alternative optima of an ILP problem are provided.

The set of real (integral) numbers is denoted by $R$ (Z). Nonnegative integral numbers are represented by $Z^+$. For $n \in N$, the set of vectors with n components (n -tuples or n -vectors) with entries in $R$ (Z) is denoted by $R^n$ ($Z^n$). Transposition is indicated by the superscript "$T$". Therefore, for $x \in R^{n,}$ $x^T$ forms a row vector. The set of $mxn$ matrices with entries in $R$ is denoted by $R^{mxn}$. When it comes to a matrix $A \in R^{mxn}$, it is typically assumed that the row index set of A is $\{1, \dots, m\}$ and the column index set is $\{1, \dots, n\}$. The elements or entries of $a \in R^{mxn}$ are represented by $a_{ij}$, where $1 \le i \le m$ and $1 \le i \le n$. The components of vectors are also regarded as $nx1$ matrices.

If A is real $mxn$-matrices and $b \in R^n$, then $Ax \le b$ is called a system of ( linear) inequalities, and $Ax = b$ a system of (linear) equations. The solution set $\{x \in R^n | Ax \le b\}$ of a system of inequalities is called a polyhedron. If $a \in R^n \setminus \{0\}$ and $a_0 \in R$, then the polyhedron $\{x \in R^n | a^T x \le a_0\}$ is called a halfspace, and the polyhedron $\{x \in R^n | a^T x = a_0\}$ a hyperplane. Every polyhedron is the intersection of finitly many halfspaces.

Integer linear programming (ILP) investigates linear programming problems in which the variables are restricted to integers: the general problem, an $mxn$-matrix A, vectors $b \in R^m$, $c \in R^n$, is usually written the following form:

$$\max\{c^T x | Ax \le b; x \text{ integer}\} \tag{1}$$

The mathematical formulation of an ILP problem is given in Definition 1.

**Definition 1** [35]**:** The mathematical formulation of an ILP problem is described below:

$$P_1: \begin{cases} Max(Min) \sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij} x_j \le b_i \\ \quad x_j \ge 0 \text{ and integer}, \ (\mathrm{i} = 1,2, \dots \mathrm{m}, \mathrm{j} = 1,2, \dots \mathrm{n}) \end{cases} \tag{2}$$

In the case of (1), any vector $x$ satisfying $Ax \le b$ is called a feasible solution (for the maximum). The set of feasible solutions is called the feasible region. If the feasible region is nonempty, the problem is feasible, otherwise infeasible. The function $(x_1, x_2, \dots, x_n) \to c_1x_1 + c_2x_2 + \dots + c_nx_n$ is called the objective function or cost function, and $c_1x_1 + c_2x_2 + \dots + c_nx_n$ is the objective value or cost of $x$. A feasible solution with an objective value equal to the maximum is called an optimum (or optimal) solution. If we replace the term 'maximizing' in (1) with 'minimizing', the resulting integer linear program problem uses the same terminology.

There is a close relationship between optimal solutions and hyperplanes, and this relationship is provided in the following theorem 1.

**Theorem 1[36]:** Let z be an integer. Let S denote the set of all feasible solutions to the ILP. If $S \cap \{x | cx = z\}$

is non-empty, then the optimum solution to the ILP will lie on the hyperplane $cx = z$.

Linear Diophantine equations are linear equations over natural numbers and are widely used in mathematical modeling. For more detailed information and important results regarding the Linear Diophantine Equation, refer to [37]. The Linear Diophantine equation is defined as follows.

**Definition 2 [37]:** An equation of the form,

$$a_1x_{1+} \dots + a_nx_n = c, \tag{3}$$

Where $a_1, a_2, \dots, a_n$, $b$ are fixed integers, this is called a *linear Diophantine equation*. We assume that $n \ge 1$ and that coefficients $a_1, a_2, \dots, a_n$ are all different from zero.

The greatest common divisor (gcd) used in expressing the results on Linear Diophantine equations is presented in Definition 3.

**Definition 3 [37]:** If $\alpha_1, \alpha_2, \dots, \alpha_n$ are rational numbers, not all equal to 0, then the largest rational number$\gamma$ dividing each of $\alpha_1, \alpha_2, \dots, \alpha_n$ exists and is called the *greatest common divisor* or gcd of $\alpha_1, \alpha_2, \dots, \alpha_n$ denoted by

$gcd\{\alpha_1, \alpha_2, \dots, \alpha_n\}$.

The ILP problem, hyperplane, and linear Diophantine equation have a closely related relationship with each other. This relationship is expressed by the following definition 4.

**Definition 4 [36]:** Consider the objective hyperplane

$\sum c_j x_j = z$,

where each $c_j \in Z$, which is a linear Diophantine equation in integers. Let $d = gcd(c_j, c_j \ne 0, j = 1,2, \dots, n)$.

It has an integer solution if and only if $d | z$.

Additionally, if a linear Diophantine equation has an integer solution, there will be infinitely many solutions for this equation.

**Theorem 2** [37]**.** The equation (3) is solvable if and only if $\gcd(a_1, a_2, \dots, a_n)|c$. In the case of solvability, one can choose $(n-1)$ solutions such that each solution is an integer linear combination of those $(n-1)$ solutions.

**Solution Method**

Let $x^*$ denote the integer solution of the ILP. If $z^*$ denotes the value of the objective function corresponding to the integer solution $x^*$ of the ILP. If the Diophantine equation $\sum_{j=1}^{n} c_j x_j = z^*$ is set, and then $y_1, y_2, \dots, y_n$ are replaced by $x_1, x_2, \dots, x_n$ in this equation $\sum_{j=1}^{n} c_j x_j = z^*$, $\sum_{j=1}^{n} c_j y_j = z^*$ is obtained. From the equation $\sum_{j=1}^{n} c_j y_j = z^*$, $y_k = \frac{[z^*-(c_1y_1+\dots+c_{k-1}y_{k-1}+c_{k+1}y_{k+1}+\dots+c_ny_n)]}{c_k}$ is found for each $k = 1, 2, \dots, n$. Then $y_1, y_2, \dots, y_{k-1}$, $\frac{[z^*-(c_1y_1+\dots+c_{k-1}y_{k-1}+c_{k+1}y_{k+1}+\dots+c_ny_n)]}{c_k}$, $y_{k+1}, \dots, y_n$ are replaced by $x_1, x_2, \dots, x_n$ in the constraints respectively. As a result of this process, the given ILP problem $P_1$ is reformulated as the following ILP problem $P_2$.

$$P_2: \begin{cases} \sum_{j=1}^{n} c_j y_j = z^* \\ \sum_{j=1}^{n-1} d_{ij} y_j \le (\ge, =) e_i \\ \quad y_j \ge 0 \text{ and integer}, (i = 1, 2, \dots, m, \; j = 1, 2, \dots, n) \end{cases} \quad (4)$$

**Proposition:** $(x_1, x_2, \dots, x_k, \dots, x_n)$ is a solution for ILP problem $P_1$ if and only if $(y_1, y_2, \dots, y_{k-1}, y_{k+1}, \dots, y_n)$ is a solution of the ILP problem $P_2$where $y_k = \frac{[z^*-(c_1y_1+\dots+c_{k-1}y_{k-1}+c_{k+1}y_{k+1}+\dots+c_ny_n)]}{c_j}$ for each $k = 1, 2, \dots, n$.

**Proof:** It is clear that $(x_1, x_2, \dots, , x_k, \dots, x_n)$ is a solution of ILP problem $P_1$ if and only if $(y_1, y_2, \dots, y_{k-1}, y_{k+1}, \dots, y_n)$ is a solution of the problem $P_2$ where $y_k = \frac{[z^*-(c_1y_1+\dots+c_{k-1}y_{k-1}+c_{k+1}y_{k+1}+\dots+c_ny_n)]}{c_j}$ for each $k = 1, 2, \dots, n$.

If the problem $P_2$ given in equation (4) is solved, the result of the original ILP problem is found.

## PROPOSED ALGORITHM

Our solution algorithm for finding all alternative optima of the ILP problem consists of the following steps.

***Step 0:*** Load ILP problem $P_1$.

***Step 1:*** Solve the ILP problem $P_1$ to find the optimal value $z^*$.

***Step 2:*** Set $\sum_{j=1}^{n} c_j x_j = z^*$.

***Step 3:*** Replace $y_1, y_2, \dots, y_n$, by $x_1, x_2, \dots, x_n$ in the equation $\sum_{j=1}^{n} c_j x_j = z^*$ respectively.

***Step 4:*** Get an arbitrary variable $y_k$ from the variables $y_1, y_2, \dots, y_n$ in the equation $\sum_{j=1}^{n} c_j y_j = z^*$.

***Step 5:*** Replace $y_1$, $y_2$, $\dots$, $\frac{[z^*-(c_1y_1+\dots+c_{k-1}y_{k-1}+c_{k+1}y_{k+1}+\dots+c_ny_n)]}{c_k}$, $\dots, y_n$ by the variables $x_1, x_2, \dots, x_n$ in constraints, respectively.

***Step 6:*** Determine the domain interval of the variables $y_1, y_2, \dots, y_n$.

***Step 7:*** Find integer points $(y_1, y_2, \dots, y_{k-1}, y_{k+1}, \dots, y_n)$ that both satisfy the reconstructed constraints and satisfy $y_k \in Z^+$.

***Step 8:*** Write $(y_1, y_2, \dots, y_k, \dots, y_n)$ integer points.

***Step 9:*** Stop.

The flowchart of our solution algorithm is illustrated in Figure 1.

**Numerical Experiments**

In this section, the proposed algorithm is applied to a maximization and minimization problem. In addition, the examples given are coded in the MAPLE programming language.

**Example 1.** Consider the following fixed-charge problem from [38].

Maximize z= $x_1 + x_2 + x_3$

Subject to $20y_1 + 30y_2 + x_1 + 2x_2 + 2x_3 \le 180$ $30y_1 + 20y_2 + 2x_1 + x_2 + 2x_3 \le 150$

$-60y_1 + x_1 \le 0$

$-75y_2 + x_2 \le 0$

$x_1, x_2, x_3 \ge 0, x_1, x_2, x_3 \in Z, y_1, y_2 \in \{0, 1\}$

***Step 0:***

$P_1$: Max $x_1 + x_2 + x_3$

Subject to $20y_1 + 30y_2 + x_1 + 2x_2 + 2x_3 \le 180$ $30y_1 + 20y_2 + 2x_1 + x_2 + 2x_3 \le 150$

$-60y_1 + x_1 \le 0$

$-75y_2 + x_2 \le 0$

$x_1, x_2, x_3 \ge 0, x_1, x_2, x_3 \in Z, y_1, y_2 \in \{0, 1\}$.

***Step 1***: Solving this problem by LINGO [39], the globally optimal solution (23, 53, 0, 1, 1), and objective value $z^*$ =76 are obtained.

***Step 2***: Set $x_1 + x_2 + x_3 = 76$.

***Step3***: If is $m, n, r, s, t$ replaced by $x_1, x_2, x_3, y_1, y_2$ in the Diophantine equation, $m + n + r = 76$ is obtained.

***Step 4***: From the equation $m + n + r = 76$, $m = 76 - n - r$ is found.

***Step 5:*** If $76 - n - r$, $n$ $r$, $s$, $t$ by are replaced by $x_1, x_2, x_3, y_1, y_2$ in the constraints, respectively, the inequality systems

$$\begin{gathered} n + r + 20s + 30t \le 104 \\ n - 30s - 20t \ge 2 \\ n + r + 60s \ge 76 \\ n - 75t \le 0 \end{gathered} \quad (5)$$

is obtained.

***Step 6:*** From the given problem and the Diophantine equation $x_1 + x_2 + x_3 = 76$, the definition ranges of

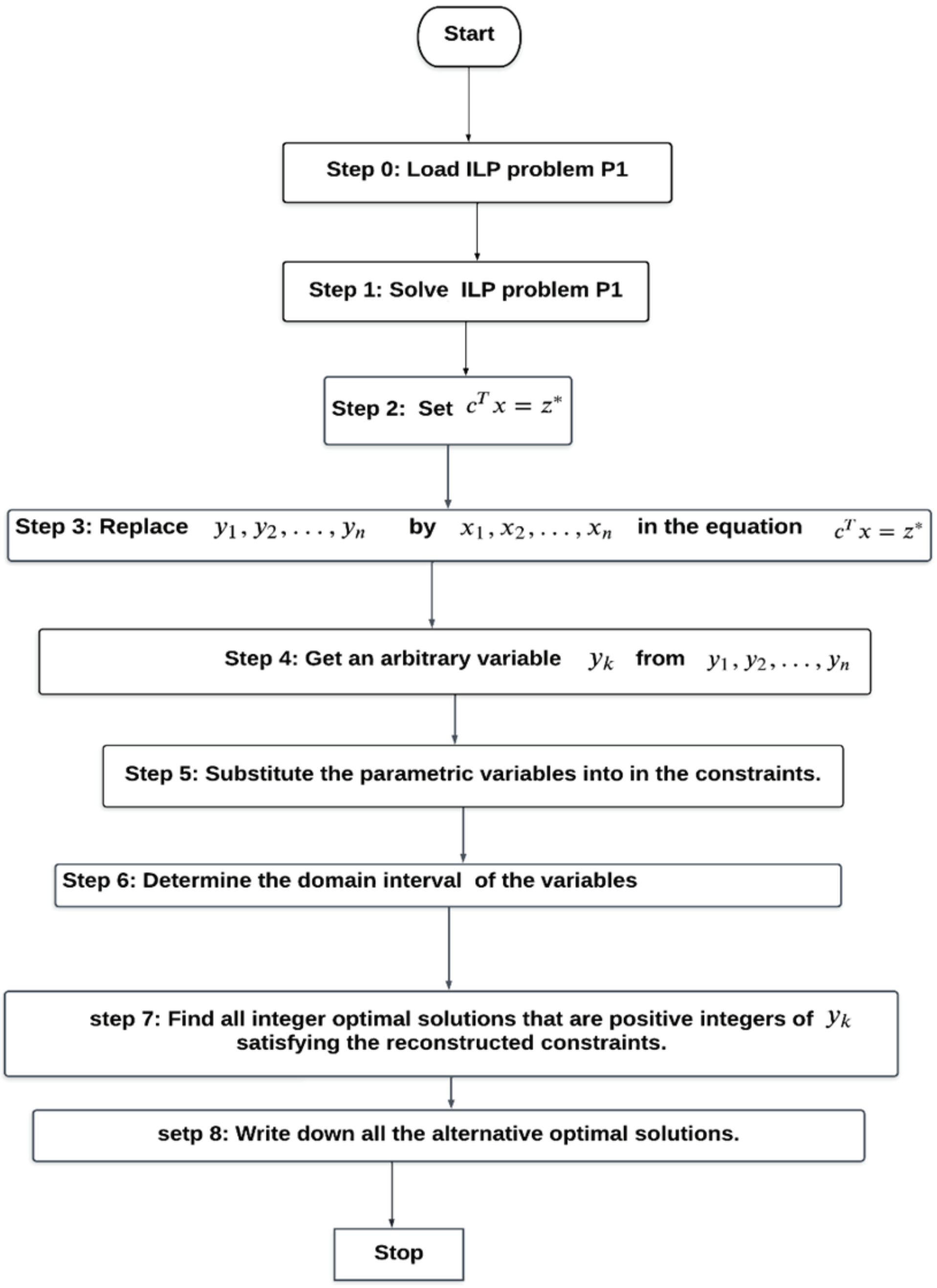
Start
Step 0: Load ILP problem P1
Step 1: Solve ILP problem P1
Step 2: Set $c^T x = z^*$
Step 3: Replace $y_1, y_2, \ldots, y_n$ by $x_1, x_2, \ldots, x_n$ in the equation $c^T x = z^*$
Step 4: Get an arbitrary variable $y_k$ from $y_1, y_2, \ldots, y_n$
Step 5: Substitute the parametric variables into in the constraints.
Step 6: Determine the domain interval of the variables
step 7: Find all integer optimal solutions that are positive integers of $y_k$ satisfying the reconstructed constraints.
setp 8: Write down all the alternative optimal solutions.
Stop

**Figure1.** The flowchart of the solution algorithm for finding all alternative optima of the Integer Linear Programming Problem.

**Table 1.** Summarized results of example 1

| **Optimal value $z^*$** | ***m*** | **Reconstructed constraints** | **All Optimal solutions** |
|---|---|---|---|
| 76 | $76 - n - r$ | $n + r + 20s + 30t \leq 104$ | (24, 52, 0, 1, 1) |
| | | $n - 30s - 20t \geq 2$ | (23, 52, 1, 1, 1) |
| | | $n + r + 60s \geq 76$ | (22, 52, 2, 1, 1) |
| | | $n - 75t \leq 0$ | (23, 53, 0, 1, 1) |
| | | | (22, 53, 1, 1, 1) |
| | | | (22, 54, 0, 1, 1) |

parametric variables are obtained as $0 \le m \le 76$, $0 \le n \le 76$, $0 \le r \le 1$, $0 \le s \le 1$, $0 \le t \le 1$.

***Step 7:*** There are five alternative optimal solutions (23, 52, 1, 1, 1), (24, 52, 0, 1, 1), (22, 52, 2, 1, 1), (23, 53, 0, 1, 1), (22, 53, 1, 1, 1), (22, 54, 0, 1, 1).

***Step 8:*** The integer points (24, 52, 0, 1, 1), (23, 52, 1, 1, 1), (22, 52, 2, 1, 1), (23, 53, 0, 1, 1 ), (22, 53, 1, 1, 1), (22, 54, 0, 1, 1) are all alternative integer optimal solutions.

***Step 9***: Stop.

Example 1 is summarized in Table 1. Additionally, the pseudo-code for Example 1 is given in Figure 2.

Table 1 presents the parametrization applied for the ILP problem given in Example 1, which has an optimal value of 76. It shows the reconstructed constraints and all obtained alternative optimal solutions. This problem was also provided as an example in the study [18], and identical results were obtained. However, in [18], 460 iterations were required to find all optimal solutions, along with the addition of 74 new constraints. In our proposed method, only one reconstructed constraint is used without the need for iterations. The computational and timing advantages provided to the user by our proposed method can be clearly observed in this example.

**Algorithm 1:** The solution algorithm for Example 1

```
1.  Begin
2.      Input ILP problem P₁
3.      Solve ILP problem P₁ to find optimal value z*
4.      Print "Optimal value z*", 76
5.      $x_1 \leftarrow m, x_2 \leftarrow n, x_3 \leftarrow r, y_1 \leftarrow s, y_2 \leftarrow t$
6.      $m \leftarrow z* - n - r$
7.      DefineVariableIntervals:
            Set $m$ interval: $[m_1, m_2]$
            Set $r$ interval: $[r_1, r_2]$
            Set $n$ interval: $[n_1, n_2]$
            Set $s$ interval: $[s_1, s_2]$
            Set $t$ interval: $[t_1, t_2]$
8.      End Define VariableIntervals
9.      $aa \leftarrow 1$
10.     While $aa = 1$ do
11.     For i from 0 to 76 do
12.      For m from 0 to 76 do
13.        For n from $n_1$ to $n_2$ do
14.          For r from $r_1$ to $r_2$ do
15.            For s from $s_1$ to $s_2$ do
16.              For t from $t_1$ to $t_2$ do
17.                $m \leftarrow 76 - i - n - r, k \leftarrow m + n + r$
18.                  If ($m = 76 - i - n - r$ and k=76 and $m \ge 0$ and type (m, integer) and
                       $n + r + 20s + 30t \le 104$ and $n - 30s - 20t \ge 2$ and
                       $n + r + 60s \ge 76$ and $n - 75t \le 0$)
    then
19.                    Print "Optimal value is", $76 - i$, "Optimal Solution is", $(m, n, r, s, t)$
20.                    $aa = 0$
21.                  End If
22.                End do
23.              End do
24.            End do
25.          End do
26.        End do
27.     End While
28.     End
```

**Figure 2.** Pseudo-code for example 1.

Maple programming language coding for Example 1 is as follows.

```
> #Maple Codes for Example 1;
> restart;
aa:=1:
for maximum from 0 to 76 while (aa=1) do;
 for n from 0 to 76 do;
 for r from 0 to 76 do;
 for s from 0 to 1 do;
for t from 0 to 1 do;
m:=(76-maximum-n-r);
K:=m+n+r;
if ( m=(76-maximum-n-r) and K=76 and m>=0 and type(m,integer) and 20*s+30*t+n+r<= 104 and -30*s-20*t+n>=2 and
60*s+n+r>=76 and -75*t+n<=0) then aa:=0 ;
print(“Optimal value is”,76-maximum,”Optimal Solution is”(m,n,r,s,t));
else end if;
end do;
end do;
end do;
end do;
end do;
“Optimal value is”, 76 “optimal Solution is” (24, 52, 0, 1, 1)
“Optimal value is”, 76 “optimal Solution is” (23, 52, 1, 1, 1)
“Optimal value is”, 76 “optimal Solution is” (22, 52, 2, 1, 1)
“Optimal value is”, 76 “optimal Solution is” (23, 53, 0, 1, 1)
“Optimal value is”, 76 “optimal Solution is” (22, 53, 1, 1, 1)
“Optimal value is”, 76 “optimal Solution is” (22, 54, 0, 1, 1)
```

**Example 2.** Consider the following ILP problem

***Step 0:***

Minimize z = $x_1 + x_2 - x_3 - x_4$

Subject to $x_1 - 3x_2 - 4x_3 \leq 4$

$x_2 + x_3 - x_4 \leq 5$

$x_1 + x_2 + x_3 + x_4 + x_5 \leq 6$

$x_1, x_2, x_3, x_4, x_5 \geq 0, x_1, x_2, x_3, x_4, x_5 \in Z$

***Step 1*:** Solving this problem by POM-QM and the optimal solution (0, 0, 0, 5, 1) objective value $z^*$ =6 is obtained.

***Step 2*:** Set $x_1 + x_2 - x_3 - x_4 = 6$.

***Step 3:*** If is $m, n, r, s, t$ replaced by $x_1, x_2, x_3, x_4, x_5$ in the Diophantine equation $m + n - r - s = 6$ is obtained.

***Step 4:*** From the equation $m + n - r - s = 6$, $m = 6 - n + r + s$ is found.

***Step 5:*** If $6 - n + r + s, n, r, s, t$ by are replaced by $x_1$, $x_2$, $x_3$, $x_4$, $x_5$ in the constraints, respectively, the inequality systems,

$$\begin{aligned} -4n - 3r + s &\leq 10 \\ n + r - s &\leq 5 \\ 2r + 2s + t &\leq 1 \end{aligned} \qquad (6)$$

is obtained.

***Step 6:*** The definition ranges of parametric variables are found as $0 \leq m \leq 6, 0 \leq n \leq 6, 0 \leq r \leq 6, 0 \leq s \leq 6, 0 \leq t \leq 6$.

***Step 7:*** There are six alternative optimal solutions (0, 0, 0, 6, 0), (0, 0, 1, 5, 0 ), (0, 0, 2, 4, 0), (0, 0, 3, 3, 0 ), (0, 0, 4, 2, 0 ), (0, 0, 5, 1, 0).

***Step 8:*** The integer points (0, 0, 0, 6, 0), (0, 0, 1, 5, 0 ), (0, 0, 2, 4, 0), (0, 0, 3, 3, 0 ), (0, 0, 4, 2, 0 ), (0, 0, 5, 1, 0) are all alternative integer optimal solutions.

***Step 9:*** Stop.

Example 2 is summarized in Table 2. Additionally, the pseudo-code for Example 2 is given in Figure 3.

Table 2 displays the applied parameterization, reconstructed constraints, and all obtained alternative optimal solutions for Example 2, where the optimal value of the ILP problem provided is 6. As seen in the table, all alternative solutions are obtained without the need for iterations, utilizing only one reconstructed constraint.

Maple programming language coding for Example 2 is as follows.

```
#Maple Codes for example 2;
> restart;
aa:=1:
for minimum from 6 by 1 while (aa=1)do;
for n from 0 to 6 do;
for r from 0 to 6 do;
for s from 0 to 6 do;
for t from 0 to 6 do;
 m:=(-6+(-1)*n+r+s);
 K:=m+n-r-s;
 if (K=-6 and m>=0 and type (m, integer),(-4)*n+(-3)*r+s<= 10 and n+r-s<=5 and
 2*r+2*s+t<=12) then aa:=0 ;
 print( “Optimal value is”,minimum, “Optimal solution is”(m,n,r,s,t) );
else end if;
end do;
end do;
```

**Table 2.** Summarized results of example 2

| **Optimal value $z^*$** | ***m*** | **Reconstructed constraints** | **All Optimal solutions** |
|---|---|---|---|
| 6 | $6 - n + r + s$ | $-4n - 3r + s \leq 10$ | (0, 0, 0, 6, 0) |
| | | $n + r - s \leq 5$ $2r + 2s + t \leq 1$ | (0, 0, 1, 5, 0 ) |
| | | | (0, 0, 2, 4, 0) |
| | | | (0, 0, 3, 3, 0 ) |
| | | | (0, 0, 4, 2, 0 ) |
| | | | (0, 0, 5, 1, 0) |

**Algorithm 2 :** The solution algorithm for Example 2

```
1.  Begin
2.      Input ILP problem P₁
3.      Solve ILP problem P₁ to find optimal value z*
4.      Print "Optimal value z*", 6
5.      $x_1 \leftarrow m, x_2 \leftarrow n,\ x_3 \leftarrow r\ , y_1 \leftarrow s\ , y_2 \leftarrow t$
6.      $m \leftarrow z * - n + r - s$
7.      Define VariableIntervals:
            Set $m$ interval: $[m_1, m_2]$
            Set $r$ interval: $[r_1, r_2]$
            Set $n$ interval: $[n_1, n_2]$
            Set $s$ interval: $[s_1, s_2]$
            Set $t$ interval: $[t_1, t_2]$
8.      End Define VariableIntervals
9.      $aa \leftarrow 1$
10.     While $aa = 1$ do
11.     For i from 0 to 76 do
12.       For m from $m_1$ to $m_2$ do
13.         For n from $n_1$ to $n_2$ do
14.           For r from $r_1$ to $r_2$ do
15.             For s from $s_1$ to $s_2$ do
16.               For t from $t_1$ to $t_2$ do
17.                 $m \leftarrow 6 - i - n + r + s, k \leftarrow m + n - r - s$
18.                   If ($m = 6 - i - n - r$ and k=6 and $m \geq 0$ and type (m, integer) and
                        $-4n - 3r + s \leq 10$ and $n + r - s \leq 5$ and $2r + 2s + t \leq 1$
     then
19.                     Print "Optimal value is", $6 - i$, "Optimal Solution is", $(m, n, r, s, t)$
20.                   $aa = 0$
21.                   End If
22.                 End do
23.               End do
24.             End do
25.           End do
26.         End do
27.       End While
28.     End
```

**Figure 3.** Pseudo-code for example 2.

```
end do;
end do;
end do;
"Optimal value is", 6, "Optimal solution is" (0, 0, 0, 6, 0)
"Optimal value is", 6, "Optimal solution is" (0, 0, 1, 5, 0)
"Optimal value is", 6, "Optimal solution is" (0, 0, 2, 4, 0)
"Optimal value is", 6, "Optimal solution is" (0, 0, 3, 3, 0)
"Optimal value is", 6, "Optimal solution is" (0, 0, 4, 2, 0)
"Optimal value is", 6, "Optimal solution is" (0, 0, 5, 1, 0)
```

## CONCLUSION

In this study, a method is proposed to find all optimal solutions of an ILP problem by developing a simple parameterization that enables the determination of a feasible region containing all optimal solutions. Additionally, an efficient algorithm is presented. First, the original ILP problem is reformulated as another ILP problem using the parameterization obtained from the Diophantine equation, making it easily solvable with simple mathematical programming. Our method is applicable regardless of the number of variables and constraints in the problem. No new constraints are added to the given ILP problem, and all optimal solutions to the problem are found in a very short time. Thanks to these advantages, the proposed algorithm can be utilized in solving ILP problems and addressing some real-life problems.

## ACKNOWLEDGEMENTS

This paper is an extension of its presentation at the 6th International HYBRID Conference on Mathematical Developments and Applications, ICOMAA 2023, 10-13 May 2023, Yıldız Technical University, Istanbul, Turkey.

This work was supported by Yildiz Technical University Scientific Research Projects Coordination Unit. Project Number: FBA-2021-4032.

## AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

## DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

## CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## ETHICS

There are no ethical issues with the publication of this manuscript.

## REFERENCES

[1] Maayah B, Moussaoui A, Bushnaq S, Arqub OA. The multistep Laplace optimized decomposition method for solving fractional-order coronavirus disease model (COVID-19) via the Caputo fractional approach. Demonstr Math 2022;55:963−977. [CrossRef]

[2] Momani S, Arqub OA, Maayah B. Piecewise optimal fractional reproducing kernel solution and convergence analysis for the Atangana-Baleanu-Caputo model of the Lienard's equation. Fractals 2020;28:2040007. [CrossRef]

[3] Arqub OA, Rashaideh H. The RKHS method for numerical treatment for integrodifferential algebraic systems of temporal two-point BVPs. Neural Comput Appl 2018;30:2595−2606. [CrossRef]

[4] Arqub OA. Computational algorithm for solving singular Fredholm time-fractional partial integrodifferential equations with error estimates. J Appl Math Comput 2019;59:227−243. [CrossRef]

[5] Arqub OA, Maayah B. Adaptive the Dirichlet model of mobile/immobile advection/dispersion in a time-fractional sense with the reproducing kernel computational approach: Formulations and approximations. Int J Mod Phys B 2022;18. [CrossRef]

[6] Djerdjour M. An enumerative algorithm framework for a class of nonlinear integer programming problems. Eur J Oper Res 1997;101:104−121. [CrossRef]

[7] Salkin HM, Mathur K. Foundations of Integer Programming. Amsterdam: North-Holland;1989.

[8] Simons R. How OR improved smelter performance. MP in Action. The Newsletter of Mathematical Programming in Industry and Commerce. 1989.

[9] Taha HA. Operations Research. New York, US: Macmillan; 2003.

[10] Biegler LT, Grossmann IE. Retrospective on optimization. Comput Chem Eng 2004;28:1169−1192. [CrossRef]

[11] Mustafa D. A metaheuristic solution approach with two construction heuristics for vehicle routing problem with simultaneous linehauls and backhauls. Sigma J Eng Nat Sci 2021;39:226−236.

[12] Gomory RE. Outline of an algorithm for integer solution to linear programs. Bull Am Math Soc 1958;64:275−278. [CrossRef]

[13] Land AH, Doig AG. An automatic method for solving discrete programming problems. Econometrica 1960;28:497−520. [CrossRef]

[14] Dakin RJ. A tree search algorithm for mixed integer programming problems. Comput J 1965;8:250−255. [CrossRef]

[15] Grossmann IE, Biegler LT. Future perspective on optimization. Comput Chem Eng. 2004;28:1193−1218. [CrossRef]

[16] Joseph A. Parametric formulation of the general integer linear programming problem. Comput Oper Res 1995;22:883−892. [CrossRef]

[17] Pandian P, Jayalakshmi MA. New approach for solving a class of pure integer linear programming problems. J Adv Eng Technol 2012;3:248−251.

[18] Tsai JF, Lin MH, Hu YC. Finding multiple solutions to general integer linear programs. Eur J Oper Res 2008;184:802−809. [CrossRef]

[19] Genova K, Guliashki V. Linear integer programming methods and approaches-a survey. J Cybern Inf Technol 2011;11:1−23.

[20] Shinto KG, Sushama CM. An algorithm for solving integer linear programming problems. Int J Res Eng Technol 2013;37−47.

[21] Bertsimas D, Perakis G, Tayur S. A new algebraic geometry algorithm for integer programming. Manag Sci 2000;46:999−1008. [CrossRef]

[22] Simsek Alan K, Albayrak I, Sivri M, Guler C. An alternative algorithm for solving linear programming problems having two variables. Int J Appl Inf Syst 2019;12:6−9.

[23] Simsek Alan K. A novel alternative algorithm for solving linear programming problems having three variables. J Cybern Inf Technol 2020;20:27−35. [CrossRef]

[24] Simsek Alan K. A novel alternative algorithm for solving linear integer programming problems with four variables. Eur J Sci Technol 2021;29:81−86. [CrossRef]

[25] Brimkov VE, Danchev SS. Real data-integer solution problems within the Blum-Shub-Smale computational model. J Complex 1997;13:279−300. [CrossRef]

[26] Aardal K, Bixby RE, Cor AJ, Hurkens CAJ, Lenstra AK, Job W, et al. Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances. Integer Program Combin Optim 7th Int IPCO Conf Graz Austria. 1999. [CrossRef]
[27] Esmaeili H, Amiri MN, Spedicato E. ABS solution of a class of linear integer inequalities and integer LP problems. Optim Methods Softw 2001;179−192. [CrossRef]
[28] Aliev I, De Loera JA, Oertel T, O'Neill C. Sparse solutions of linear Diophantine equations. SIAM J Appl Algebra Geom. 2017;1:239−253. [CrossRef]
[29] Yılmaz OF. An integer programming model for disassembly system configuration. Sigma J Eng Nat Sci. 2019;37:813−825.
[30] Simsek Alan K. A novel alternative algorithm for solving linear integer programming problems. Eur J Sci Technol 2021;29:93−98. [CrossRef]
[31] Alonso PF, Arellano-VJ J, Diego-Celis R. Two heuristics for the label printing problem. Int Trans Oper Res 2022;29:2841−2854. [CrossRef]
[32] Fernando HCD, Williams L, Mumey B, Tomescu AI. Efficient minimum flow decomposition via integer linear programming. J Comput Biol 2022;29:1252−1267. [CrossRef]
[33] Dehua X, Yuan S, Yu C, Limin X, Fengzhao Y, Yunzhou X. An integer linear programming model for the label printing problem. Int Trans Oper Res 2023;1−19.
[34] Auricchio G, Ferrarini L, Lanzarotto G. An integer linear programming model for tilings. J Math Music 2023;17:514−530. [CrossRef]
[35] Chen DS, Batson RG, Dang Y. Applied Integer Programming: Modeling and Solution. John Wiley & Sons; 2011.
[36] Schrijver A. Theory of Linear and Integer Programming. Chichester, UK: John Wiley & Sons, Ltd; 1986.
[37] Andreescu T, Andrica D, Cucurezeanu I. An Introduction to Diophantine Equations. New York, US: Springer, Ltd; 2010. [CrossRef]
[38] Zionts S. Linear and Integer Programming. New Jersey, US: Prentice-Hall; 1974.
[39] LINGO. Chicago: Release 9.0. LINDO System Inc; 2004.