Reports on Economics and Finance, Vol. 9, 2024, no. 1, 27 - 38
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ref.2024.91419

Generalization of the Method Based on Parameterization Developed for Solving Integer Programming Problems

Kadriye Simsek Alan

Department of Mathematical Engineering
Yildiz Technical University, Davutpasa, Istanbul, Turkey

Abstract

This study presents the generalization of methods based on parameterization proposed for solving integer programming problems with linear objectives and linear constraints. Initially, the method based on parameterization, proposed for solving 2, 3, and 4-variable integer linear programming problems, is generalized to n-variable integer linear programming problems. The proposed method relies on the parameterization obtained from Diophantine equations. Using this parameterization, the original problem is reformulated as another integer linear programming problem that can be solved more efficiently using simple mathematical programming. The method can be used regardless of the number of constraints of the problem and also provides all alternative solutions to the decision maker. It is demonstrated with some examples that this method provides an efficient algorithm.

Keywords: Linear integer programming, Linear Diophantine equations, Optimal hyperplane, Pure integer programming problems, Optimal solution

1 Introduction

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. Integer programming is also known as discrete optimization or combinatorial optimization. Due to the nature of many problems encountered in practice, there are many situations where all or some of the variables in the problem must be integer values. For example, since it would be invalid in practice to define some products produced in enterprises as fractional in terms of quantity, it is necessary to express the transactions that are considered as variables in terms of production as integers. Integer Programming is needed to solve the problems encountered in many fields such as planning, transportation, production, telecommunications, security, modeling, foresight, finance, investment, insurance and so on, which branches of science such as business, economics, engineering, statistics, and mathematics deal with. Many studies have been done in the field of integer programming. Many studies have been made in the field of Integer Programming. Notable contributions include the works of Gomory, Dantzig, and Land and Doig [4, 6, 7]. Gomory [6] outlined a finite algorithm for obtaining integer solutions to linear programs. Dantzig [4] reviewed some of the successes in the use of linear programming methods for solving discrete variable extremum problems. Markowitz and Manne [8] presented a general approach susceptible to individual variations, depending upon the problem and the judgment of the user. Land and Doig [7] developed the branch boundary method, which is one of the most preferred methods in solving integer programming (IP) problems. Joseph [9] presented a linear programming formulation and a parametric formulation of ILP to determine the contribution of the endpoints to the optimal ILP solution. It has been shown that this formulation is equivalent to the standard formulation and thus can be used to solve the ILP problem. Pandian and Jayalakshmi [12] developed a method, also called variable reduction method, based on mathematical concepts for classes of pure integer linear programming problems. Tsai et al. [21] developed a global optimization approach to find all solutions of a general ILP problem and presented an algorithm for finding all alternative optimum solutions of an ILP problem. Mohammed et al. [11] formulated a general personnel scheduling problem with hourly requirement patterns as a linear programming problem. Genova and Guliashki [5] reviewed methods and approaches for solving integer linear problems belonging to the class of NP-hard optimization problems in the last 50 years. Hossain and Hasan [10] proposed an algorithm for solving a large-scale integer programming problem based on the column generation method and used the proposed algorithm to solve capital budgeting and planning. Shinto and Sushama [15] provided a test that examines whether the approximate solution from Relaxed Linear Programming (RLP) is an optimal solution in ILP, using the concept of the Linear Diophantine Equation in their study. In addition, they proposed a modification of the branch-bound method to reach the optimal solution of the ILP problem even when the approximate solution obtained from RLP does not satisfy the optimality conditions. Bertsimas et al. [1] proposed a method for solving ILP problems, which provides a natural generalization for Farkas Lemma. The method is based on algebraic geometry and gives a natural way to do sensitivity analysis. This path also gives the systematic ordering of all viable solutions and the structural information of the appropriate Integer Programming set. Dang and Ye [3] proposed an alternative method for solving integer programming problems called the "fixed point iterative method. Pedrosa [13] presented an evolutionary algorithm for solving linear integer programs based on the strategy of separating variables into integer subset and continuous subset. Simsek Alan et al. [16] developed an alternative method based on parameterization obtained from Diophantine equations for solving integer linear programming (ILP) problems with two variables. Later, this method was developed to solve ILP problems with three variables and four variables [17-18]. Simsek Alan [19] proposed an algorithm to find solutions for an ILP problem using basic computer programming, considering the lower and upper bounds of decision variables. Although this method is easily applicable and effective, as the number of variables increases, the processing time becomes longer and its applicability decreases. Therefore, the importance of developing more efficient and useful new methods and algorithms to overcome ILP problems is obvious.

In this study, the method based on the parametrization derived from Diophantine equations, developed for two-variable Integer Linear Programming (ILP) problems in Simsek Alan et al. [16], for three-variable ILP problems in Simsek Alan [17], and four-variable ILP problems in Simsek Alan [18], has been generalized for n-variable ILP problems. The generalized method reformulates the given ILP problem using this parametrization and can be solved with basic computer programming techniques. A numerical example is provided to demonstrate the application of the method, and it is implemented 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 algorithm is given in section 4. Our numerical examples and conclusions are presented in Section 5 and Section 6, respectively.

2 Preliminaries

In this section, brief requried information are presented.

Definition 1 [2]: The mathematical formulation of an ILP problem is described below:

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

Definition 2 **[14]:** 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 $=$ $gdc(c_j, c_j \neq 0, j = 1,2, \dots, n)$. 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 (Schrijver 1986).

Theorem 1[2]: $(x_1, x_2, \ldots, x_n,\ z)$ is a solution of the problem if and only if $(x_1, x_2, \ldots, x_n)$ satisfies all constraints of P_1 .$(x_1,\ x_2, \ldots, x_n)$ satisfies the objective function (maximum or minimum) defined in Definition 1.

3 Solution method

ILP algorithms take advantage of the computational methods and results of linear programming that yield successful results. Likewise, our algorithm uses these computational methods to find solutions to ILP problems. It starts with solving the LP problem.

Let x denote the optimum solution of RLP and z_{LP} the optimum value of the objective function. That is $z_{LP} = c^T x$. Let x^* denotes the integer solution of the ILP. If z^* denotes the value of the objective function corresponding to the integer solution x^* of the ILP. Then z^* is an integer less than or equal to z_{LP}. That is $z^* \leq c^T x$. Since $z^* \leq c^T x$, let $z^* = \lfloor z_{LP} \rfloor$, the greatest integer less than or equal to z_{LP}. If $x^* \in Z^n$, then the integer solution x^* has been found. If $x^* \notin Z^n$, then according to [20], the ILP problem P_1 is reformulated as the following ILP problem P_2.

$$P_2\colon \begin{cases} \displaystyle\sum_{j=1}^{n} c_j y_j = \lfloor z_{LP} \rfloor \\ \quad \displaystyle\sum_{j=1}^{n-1} d_{ij} y_j \leq e_i \\ \quad y_j \geq 0 \ \text{ and integer}, \big(i = 1, 2, \ldots, m,\ \ j = 1, 2, \ldots, (n-1)\big) \end{cases} \tag{2}$$

As a result of the proposition given above, If the problem P_2 is solved, the optimal solution of the ILP problem P_1 is obtained. For this, first, parameterization is made according to the optimal $\lfloor z_{LP} \rfloor$ value. Then, by taking into account the lower and upper bounds of the decision variables, it is investigated whether there are integer points that satisfy the equations $\sum_{j=1}^{n} c_j y_j = \lfloor z_{LP} \rfloor$, and $\sum_{j=1}^{n-1} d_{ij} y_j \leq e_i$. If such integer points exist, the desired optimal solution is found. If there is more than one optimal solution, they are alternative solutions. If no integer points are satisfying the equations $\sum_{j=1}^{n} c_j y_j = \lfloor z_{LP} \rfloor$, and $\sum_{j=1}^{n-1} d_{ij} y_j \leq e_i$, This process is repeated for maximum (minimum) problems by decreasing (increasing) the optimal z^* value by one unit until optimal solutions are found.

The proposed algorithm for solving ILP problems consists of the following steps.

Step 0: Load LP problem P_1.

Step 1: Solve the relaxed LP problem P_1 to find the optimal solution x^*.

Step 2: If the optimal solution $(x_1, x_2, \dots, x_n)$ is an integer, then the IP problem has been solved. Stop. Otherwise, go to Step 3.

Step 3: Set $c^T x = \lfloor z_{Lp} \rfloor$.

Step 4: Replace $y_1, y_2, \dots, y_n$ by $x_1, x_2, \dots, x_n$ in the equation $c^T x = z^*$, respectively.

Step 5: Get an arbitrary variable y_i from the variables $y_1, y_2, \dots, y_n$ in the equation $c^T y = \lfloor z_{LP} \rfloor$.

Step 6: Replace $y_1, y_2, \dots, y_i, \dots, y_n$ by the variables $x_1, x_2, \dots, x_n$ in constraints, respectively.

Step 7: Determine the domain interval of the variables $y_1, y_2, \dots, y_n$.

Step 8: Is the domain interval meaningful? If it is meaningful, go to step 9. Otherwise, replace$\lfloor z_{LP} \rfloor - 1$ $(\lfloor z_{LP} \rfloor + 1)$ by $\lfloor z_{LP} \rfloor$, in the maximization (minimization) problem and return to step 3.

Step 9: If there is an integer solution $(y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$ that satisfies the system of inequalities, go to step 10. Otherwise, replace $\lfloor z_{LP} \rfloor - 1$ $(\lfloor z_{LP} \rfloor + 1)$ by $\lfloor z_{LP} \rfloor$ in the maximization (minimization) problem and return to step 3.

Step 10: Calculate the value of y_i using the integer points $(y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$.

Step 11: If $y_i \in Z^+$, then the IP problem has been solved. go to step 12. If the solution found is more than one, they are the alternative optimal solutions. If $y_i \notin Z^+$, replace $\lfloor z_{LP} \rfloor$ by $\lfloor z_{LP} \rfloor - 1$ $(\lfloor z_{LP} \rfloor + 1)$ in the maximization (minimization) problem and return to step 3.

Step 12: Write integer points $(y_1, y_2, \dots, y_i, \dots, y_n)$.

Step13: Stop.

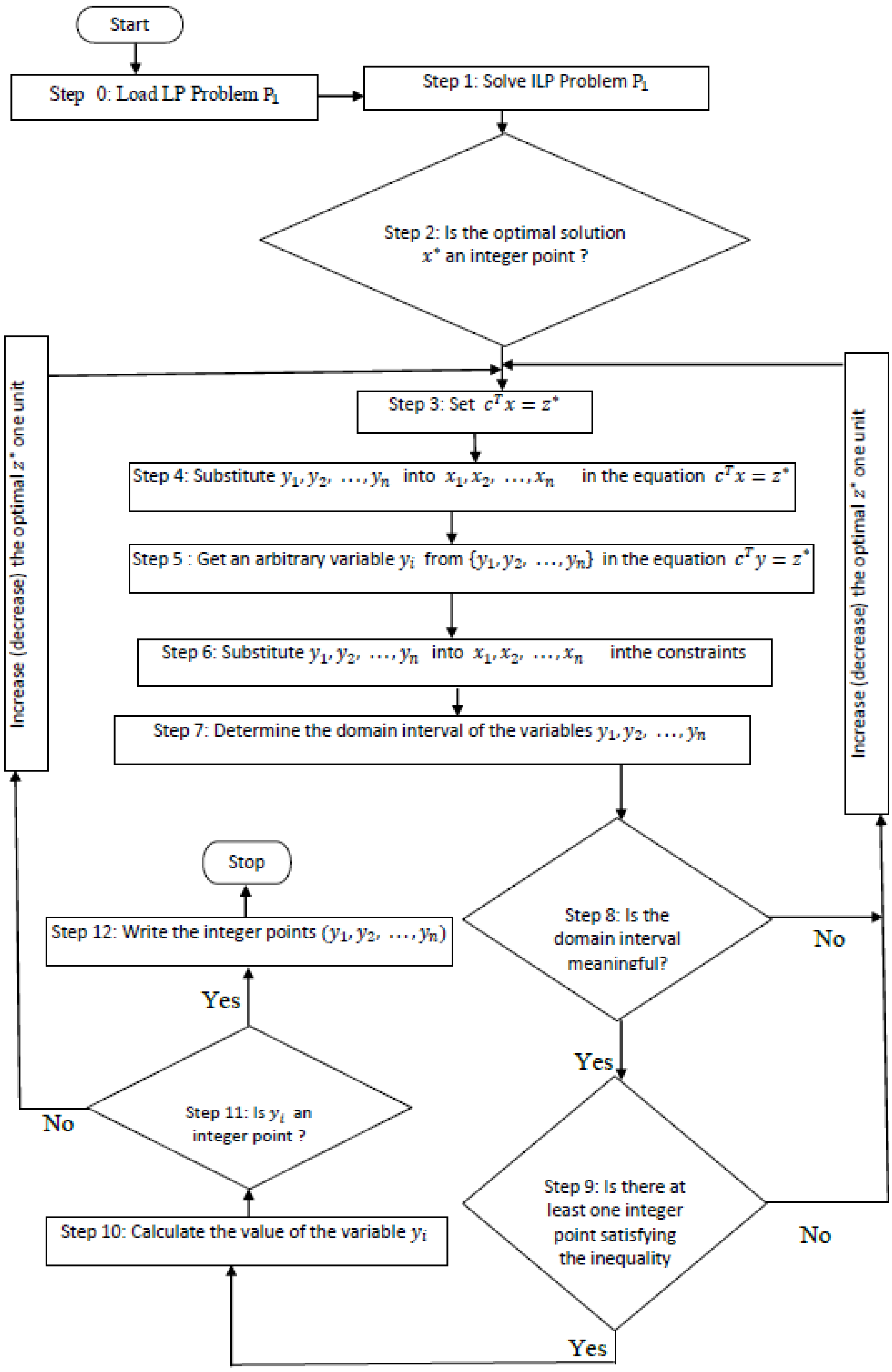

Figure 1: Flowchart of the Solution Method for Integer Linear Programming Problems

4 Numerical experiment

In this section, we solve a maximization problem using our method, and we code this example in the MAPLE programming language.

Example 4.1. Solve the following ILP problem.

Step 0:

P_1 Max $35x_1 + 85x_2 + 135x_3 + 10x_4 + 25x_5 + 2x_6 + 94x_7$
Subject to

$$2x_1 + 3x_2 + 9x_3 + 0.5x_4 + 2x_5 + 0.1x_6 + 4x_7 \leq 25$$

$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0,\ x_5 \geq 0,\ x_6 \geq 0,\ x_7 \geq 0$ and integer.

Step 1: If the relaxed LP problem is solved, $P_1: (x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6, x_7) = (0\ ,\ 8.3333, 0,\ 0,\ 0,\ 0,\ 0)$ and the optimum value $z^* = 708$ is obtained.

Step 2: There is no integer solution, goto step 3.

Step 3: Set $35x_1 + 85x_2 + 135x_3 + 10x_4 + 25x_5 + 2x_6 + 135x_3 + 94x_7 = 708$.

Step 4: If is $y_1, y_2, \ldots, y_7$ replaced by $x_1, x_2, \ldots, x_7$ in the Diophantine equation
$35x_1 + 85x_2 + 135x_3 + 10x_4 + 25x_5 + 2x_6 + 135x_3 + 94x_7 = 708$,
the equation $5y_1 + 85y_2 + 135y_3 + 10y_4 + 25y_5 + 94y_7 = 708$ is obtained.

Step 5: From the equation $5y_1 + 85y_2 + 135y_3 + 10y_4 + 25y_5 + 94y_7 = 708$,

$$y_6 = \frac{708 - (35y_1 + 85y_2 + 135y_3 + 10y_4 + 25y_5 + 94y_7)}{2}$$

is found.

Step 6: If $y_1,\ y_2,\ y_3,\ y_4,\ y_5,\ \frac{708-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2},\ y_7$ are replaced by $x_1, x_2,\ x_3, x_4,\ x_5,\ x_6,\ x_7$ in the constraints, respectively, the inequality $-5y_1 + 25y_2 - 45y_3 + 5y_5 + 14y_7 \geq 208$ is obtained.

*Step 7-8:*The domain interval of the variables is found as $0 \leq y_1 \leq 12,\ 0 \leq y_2 \leq 8,\ 0 \leq y_3 \leq 2,\ 0 \leq y_4 \leq 50,\ \ 0 \leq y_5 \leq 12,\ 0 \leq y_6 \leq 250,\ 0 \leq y_7 \leq 6$.

Step 9: There is no integer point $(y_1,\ y_2,\ y_3,\ y_5,\ y_7)$ that satisfies the inequality $-5y_1 + 25y_2 - 45y_3 + 5y_5 + 14y_7 \geq 208$. Therefore, Optimal value 708 is replaced by optimal value 707 and return to step 3 and the steps of the given algorithm are applied. If this process continues until the optimal solution is found, for the optimal value 700, $35x_1 + 85x_2 + 135x_3 + 10x_4 + 25x_5 + 2x_6 + 135x_3 + 94x_7 = 700$ is obtained. If the parametric variable y_6 is obtained from this equation, $y_6 = \frac{700-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$ found. If y_1, y_2, y_3, y_4, y_5, $\frac{708-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$, y_7 are replaced by x_1, x_2, x_3, x_4, x_5, x_6, x_7, rescpectively, in the constraints, the inequality $-5y_1 + 25y_2 - 45y_3 + 5y_5 + 14y_7 \geq 200$ is obtained. The integer points (0, 8, 0, 0, 0, 10, 0), (0, 8, 0, 1, 0, 5, 0), (0, 8, 0, 2, 0, 0, 0) satisfy this inequality, and y_6 also positive integers. Consequently, the optimal value of our problem is 700, and the optimal solutions are (0, 8, 0, 0, 0, 10, 0), (0, 8, 0, 1, 0, 5, 0), (0, 8, 0, 2, 0, 5, 0). The summarized results of Example 4.1 are presented in Table 4.1.

```
Maple Codes for example 4.1;
restart;
aa:=1:
for maximum from 0 to 708  while (aa=1) do;
  #print ("Optimal value", 708-maximum);
    for y1 from 0 to 12 do;
    for y2 from 0 to 8 do;
    for y3 from 0 to 2 do;
    for y4 from 0 to 50 do;
    for y5 from 0 to 12 do;
    for y7 from 0 to 6 do;
      y6:=(708-maximum-(35*y1+85*y2+135*y3+10*y4+25*y5+94*y7))/2;
      A:=2*y1+3*y2+9*y3+(0.5)*y4+2*y5+(0.1)*y6+4*y7;
        if (y6=(708-maximum-
(35*y1+85*y2+135*y3+10*y4+25*y5+94*y7))/2 and y6>=0
          and type(y6,integer) and type(maximum,integer)and A<= 25 )
then aa:=0;
          print("Optimal value is ",708-maximum,"Optimal solution
          is"(y1,y2,y3,y4,y5,y6,y7));
        else end if;
    end do;
    end do;
    end do;
    end do;
    end do;
    end do;
  end do;
```

"Optimal value is ", 700, "Optimal solution is"(0, 8, 0, 0, 0, 10, 0)

"Optimal value is ", 700, "Optimal solution is"(0, 8, 0, 1, 0, 5, 0)

"Optimal value is ", 700, "Optimal solution is"(0, 8, 0, 2, 0, 0, 0)

Table 4.1. Summarized results of example 4.1

Optimal value z^*	y_6	Reconstructed constraints	Is there an integer point that satisfy the reconstructed constraints ?	Is $(y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ an integer ?
708 (Iteration 1)	$\frac{708-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 208$	No	
707 (Iteration 2)	$\frac{707-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 207$	No	–
706 (Iteration 3)	$\frac{706-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 206$	No	–
705 (Iteration 4)	$\frac{705-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 205$	No	–
704 (Iteration 5)	$\frac{704-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 204$	No	–
703 (Iteration 6)	$\frac{703-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 203$	No	–
702 (Iteration 7)	$\frac{702-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 202$	No	–
701 (Iteration 8)	$\frac{701-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 201$	No	–
700 (Iteration 9) Optimal value z*	$\frac{700-(35y_1+85y_2+135y_3+10y_4+25y_5+94y_7)}{2}$	$-5y_1+25y_2-45y_3+5y_5+14y_7 \geq 200$	**YES**	**(0, 8, 0, 2, 0, 0, 0) (Integer Solution), { (0 , 8 , 0 , 0 , 0 , 10 , 0) , (0 , 8 , 0 , 1 , 0 , 5 , 0)} (Alternative Integer Solution)**

5 Conclusion

In this study, a novel iterative method for solving general Integer Linear Programming (ILP) problems is proposed using parameterization. The method introduces a simple and effective algorithm, employing basic algebraic operations and elementary mathematical programming. Compared to other methods for solving ILP problems, our approach offers several advantages, which are as follows:

1. **All Alternative Solutions**: Unlike many traditional methods that focus only on finding the optimal solution, our approach systematically identifies all alternative solutions to ILP problems.
2. **Speed and Efficiency**: The method solves larger problems in less time and uses resources more efficiently.
3. **Memory Usage**: It manages memory efficiently for large problems.
4. **Flexibility**: The method can be applied to various ILP problems, unlike other methods that are often limited to specific problem types.

Acknowledgements. This paper is an extension of its presentation at the 12th International Conference on Applied Analysis and Mathematical Modeling ICAAMM 2024, 2024, July, 19-23, Biruni Universty, Istanbul, Turkey.

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

References

[1] D.G. Bertsimas, G. Perakis, and S. Tayur, A New Algebraic Geometry Algorithm for Integer Programming, *Management Science*, **46** (2000), no. 7, 999-1008. https://doi.org/10.1287/mnsc.46.7.999.12033

[2] Chen, R.G. Batson, and Y. Dang, *Applied Integer Programming: Modeling and Solution*, John Wiley & Sons, New York, 2011.

[3] C. Dang and Y. Ye, A Fixed Point Iterative Approach to Integer Programming and its Distributed Computation, *Fixed Point Theory and Applications*, **2015** (2015), 1, 182. https://doi.org/10.1186/s13663-015-0429-8

[4] G.B. Dantzig, Discrete-variable Extremum Problems, *Journal of the Operations Research Society of America*, **5** (1957), no. 2. https://doi.org/10.1287/opre.5.2.266

[5] K. Genova and V. Guliashki, Linear Integer Programming Methods and Approaches – A Survey, *Journal of Cybernetics and Information Technologies*, **11** (2011), no. 1, 1-23.

[6] R.E. Gomory, Outline of an Algorithm for Integer Solution to Linear Programs, *Bulletin of the American Mathematical Society*, **64** (1958), 275-278. https://doi.org/10.1090/s0002-9904-1958-10224-4

[7] A.H. Land and A.G. Doig, An Automatic Method for Solving Discrete Programming Problems, *Econometrica*, **28** (1960), 497-520. https://doi.org/10.2307/1910129

[8] H.M. Markowitz and A.S. Manne, *On the Solution of Discrete Programming*, Rand Report P-885, June 1996. https://doi.org/10.7249/p711

[9] A. Joseph, Parametric Formulation of the General Integer Linear Programming Problem, *Computers & Operations Research*, **22** (1995), no. 9, 883-892. https://doi.org/10.1016/0305-0548(94)00077-l

[10] M.I. Hossain and M.B. Hasan, A Decomposition Technique for Solving Integer Programming Problems, *GANIT: Journal of Bangladesh Mathematical Society*, **33** (2013), 1-11. https://doi.org/10.3329/ganit.v33i0.17649

[11] N.H. Mohamad and F. Said, Integer Linear Programming Approach to Scheduling Toll Booth Collectors Problem, *Indian Journal of Science and Technology*, **6** (2013), no. 5, 4416-4421. https://doi.org/10.17485/ijst/2013/v6i5.7

[12] P. Pandian and M.A. Jayalakshmi, New Approach for Solving a Class of Pure Integer Linear Programming Problems, *Journal of Advanced Engineering Technology*, **3** (2012), 248-251.

[13] J.P. Pedroso, An Evolutionary Solver for Pure Integer Linear Programming, *International Transactions in Operational Research*, **9** (2002), no. 3, 337-352. https://doi.org/10.1111/1475-3995.00359

[14] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons Ltd, 1986.

[15] K.G. Shinto and C.M. Sushama, An Algorithm for Solving Integer Linear Programming Problems, *International Journal of Research in Engineering and Technology*, **2** (2013), 107-112. https://doi.org/10.15623/ijret.2013.0207012

[16] K. Simsek Alan, I. Albayrak, M. Sivri, and C. Güler, An Alternative Algorithm for Solving Linear Programming Problems Having Two Variables, *International Journal of Applied Information Systems*, **12** (2019), no. 25, 6-9.

[17] K. Simsek Alan, A Novel Alternative Algorithm for Solving Linear Programming Problems Having Three Variables, *Journal of Cybernetics and Information Technologies*, **20** (2020), 27-35. https://doi.org/10.2478/cait-2020-0045

[18] K. Simsek Alan, A Novel Alternative Algorithm for Solving Linear Integer Programming Problems with Four Variables, *European Journal of Science and Technology*, **29** (2021), 81-86. https://doi.org/10.31590/ejosat.1020212

[19] K. Simsek Alan, A Novel Alternative Algorithm for Solving Linear Integer Programming Problems, *European Journal of Science and Technology*, **29** (2021), 93-98.

[20] K Simsek Alan, A Novel Alternative Algorithm to Find All Multiple Solutions of General Integer Linear Program, *Sigma Journal of Engineering and Natural Sciences*, **42** (2024), no. 5, 1532-1541. https://doi.org/10.14744/sigma.2024.00118

Received: October 25, 2024; Published: November 15, 2024