# Hybrid Optimization with Integer Constraints: Modeling and Solving Problems Using Simplex Techniques

**Mark Laisin & Collins Edike**

**Abstract**

This paper presents a hybrid optimization framework for solving linear integer programming (LIP) problems using Simplex-based techniques. Building on the classical Simplex method and integrating it with branch-and-bound, cutting-plane, and heuristic approaches, the proposed methodology addresses the complexity of integer-constrained decision-making in areas such as scheduling, logistics, and resource allocation. A brief historical overview of the Simplex method and its evolution into LIP solutions is provided, along with theoretical insights into feasibility, constraint preservation, and solution structure. The effectiveness of the hybrid approach is demonstrated through a real-world application at Innoson Vehicle Manufacturing, showcasing how the integrated method improves solution accuracy and computational efficiency in practical settings. This study contributes to the broader field of optimization by offering a unified, adaptable approach for modeling and solving complex integer programming problems.

*Keywords*: Simplex method, Linear Integer Programming (LIP), Branch-and-bound, Resource allocation

## I. INTRODUCTION

Optimization techniques have become essential tools in mathematics, economics, operations research, and computer science. Among these, linear programming (LP) has played a foundational role in modeling and solving a wide array of real-world challenges—from resource allocation to logistics. The introduction of the Simplex method by George Dantzig in 1947 marked a significant milestone, offering an efficient algorithm for solving LP problems by navigating the vertices of the feasible region (Dantzig, 1947).

However, many practical problems—such as workforce scheduling, transportation planning, and network design—require decision variables to take on integer values, which standard LP techniques cannot directly handle. This limitation led to the development of Linear Integer Programming (LIP), an extension of LP that combines continuous optimization methods with discrete variable constraints (Wolsey, 1998).

Despite the theoretical appeal of LIP, solving such problems efficiently remains a challenge due to the combinatorial explosion of possible integer solutions. To overcome this, hybrid optimization techniques have emerged, integrating the Simplex algorithm with methods like branch-and-bound, cutting planes, and heuristic strategies to handle integrality constraints more effectively (LaisinandEdike, 2025a). These hybrid approaches maintain the mathematical rigor of LP while enabling robust, efficient solutions to discrete decision-making problems in logistics, energy systems, and scheduling.

The historical roots of LP date back to the 19th century, with Joseph Fourier's development of the Fourier-Motzkin elimination method for solving systems of inequalities (Fourier, 1824). Later, János Farkas introduced Farkas' Lemma (1902), which enhanced theoretical understanding of feasibility in linear systems. These foundational contributions culminated in Dantzig's Simplex Method (1947), which remains widely used in both academia and industry.

The adaptation of the Simplex method for LIP, however, introduced new difficulties. Integer constraints disrupted the convex geometry exploited by Simplex, prompting researchers to explore hybrid strategies. Techniques like branch-and-bound (Land & Doig, 1960), cutting-plane methods (Gomory, 1958), and heuristic-guided iterations have since been combined with the Simplex framework to tackle LIP more effectively.

Such hybrid Simplex-based LIP models have gained traction in recent years across diverse sectors:

- Supply Chain Management: Hybrid optimization is used to manage inventory, production, and transportation decisions involving integer constraints (Chauhan & Dey, 2022).
- Transportation and Logistics: Integer-constrained problems such as vehicle routing and delivery scheduling benefit from integrated Simplex and branch-and-bound methods (López et al., 2021).
- Scheduling: In job-shop and workforce scheduling, Simplex-based models are hybridized with metaheuristics to enhance feasibility and reduce solution time (Zhao & Kim, 2020).
- Energy Distribution: Hybrid models optimize decisions in smart grids and renewable energy networks, where binary and integer variables

are critical (Cui et al., 2023).

Furthermore, recent computational studies have demonstrated the advantages of hybrid optimization frameworks in terms of both convergence speed and solution stability. For example, Laisin and Adigwe (2025b) showed that hybrid methods—particularly those integrating gradient-based algorithms with integer constraints—consistently outperform classical techniques in complex solution spaces. Additionally, Laisin and Edike (2025c), in their paper titled *"Characterizing Boundedness and Solution Size in Rational Linear Programming and Polyhedral Optimization,"* established that all feasible solutions within such rational linear systems are bounded and exhibit guaranteed convergence properties.

This paper builds upon these developments by presenting a modeling framework and solution methodology for solving Integer-Constrained Linear Programs using Simplex-based hybrid optimization. The approach is demonstrated across multiple real-world applications, reinforcing its utility as a unified and adaptable solution paradigm.

## II. PRELIMINARIES AND METHODS

Linear Integer Programming (LIP) involves solving combinatorial constrained optimization problems with integer variables, where both the objective function and constraints are linear relationships. This mathematical framework can be represented as follows:

$$Maximize\ Z = \sum_{i=1}^{n} c_i x_i$$

$$subject\ to: Ax = \sum_{i=1}^{n} a_{ij} x_i \leq b_j, j = 1,2, \dots, m,$$

$$x_i \in \mathbb{Z}^n,$$

Where the solution $x_i \in Z^n$ is a vector of n integer variables: $x_i = (x_1, x_2, \dots, x_n)^T$ and the data are rational and are given by the $m \times n$ matrix A, the $1 \times n$ matrix c, and the $m \times 1$ matrix b. This formulation also includes equality constraints as each equality constraint can be represented using two inequality constraints like $a_{ij} x_i \leq b$. There are two types of Integer Programming Problems: An **Integer Programming Problem (IPP)** involves finding optimal solutions to mathematical models where some or all decision variables are constrained to integer values. When all variables are restricted to non-negative integers, it is called the **Pure Integer Programming Problem** or **All Integer Programming Problem (All IPP)**. Conversely, if only some variables are constrained to be integers while others can take any non-negative values, it is termed a **Mixed Integer Programming Problem (Mixed IPP)**.

Integer programming is a fundamental tool in optimization, enabling the modeling of complex decision-making processes in various fields such as operations research, economics, and engineering. The distinction between pure and mixed integer programming allows for flexibility in modeling scenarios where certain decisions are discrete (e.g., the number of machines to purchase) while others are continuous (e.g., the amount of raw material to use).

Recent advancements focus on improving algorithms for solving these problems efficiently. For example, Del Pia (2023) explores the complexity of mixed integer convex quadratic programming, providing insights into optimization algorithms.

Additionally, the development of algorithms utilizing Graver bases have enhanced solving nonlinear integer programming problems in variable dimensions. This approach has been particularly effective in addressing block-structured and n-fold integer programming challenges, contributing to the broader theory of sparse and bounded tree-depth integer programming.

These advancements have significant implications for practical applications, enabling more efficient and effective solutions to complex optimization problems across various industries.

**Definitions**

- **General Description:** Simplex Linear Integer Programming involves solving optimization problems where the objective function and constraints are linear, and the decision variables are restricted to integer values, using the Simplex algorithm as the core solution technique *(Winston, 2004;* Glover *et al*., 2021*).*

- **Mathematical Perspective:**The Simplex Method is an iterative algorithm for solving linear programming problems. It operates on a feasible region defined by linear inequalities and progresses toward an optimal solution by moving along the edges of this region (*Nemhauser, & Wolsey, 1999;* Dantzig, 2021*).*

- **Algorithmic Focus:** Simplex Linear Integer Programming applies the Simplex algorithm as a basis to navigate feasible regions of linear programming, extended to accommodate integer variable constraints

*(Bertsimas, &Tsitsiklis, 1997;*Klein, 2018*).*

- **Practical Context:**This class of optimization problems employs the Simplex method to optimize linear objective functions over decision variables restricted to integer values. These models are widely utilized in scheduling, resource allocation, and logistics, where decisions often involve discrete quantities (Schrijver, 1986; Schrijver, 2017). However, Laisin et al. (2024), in their study titled *"The Construction of Rational Polyhedra on an n × n Board with Some Applications to Integral Polyhedral Theory,"* demonstrated that certain polyhedra—particularly those analyzed graphically—necessitate integer-valued solutions to accurately define feasible regions. Their findings underscore the importance of incorporating integer constraints when modeling real-world problems where fractional solutions are not practical or permissible.

- **Hybrid Approach:** Simplex Linear Integer Programming integrates the principles of linear programming and combinatorial optimization to solve integer-restricted problems, balancing computational efficiency and precision. (*Hillier, & Lieberman, 2020;* Klose &Drexl, 2022*).*

- Linear Integer Programming (LIP): A mathematical optimization problem where the objective function is linear, and the decision variables are required to take integer values. These problems are commonly subject to linear constraints, including bounds and relationships between the variables (Vanderbei, 2022).

- Branch-and-Bound: A general algorithm for solving integer and combinatorial optimization problems. It systematically explores the solution space by dividing it into subproblems and eliminating suboptimal solutions using bounds (Land & Doig, 2021).

- Feasibility and Optimality Conditions: In linear programming, feasibility refers to whether there exists a set of values for the decision variables that satisfies all the constraints. A solution is feasible if it lies within the feasible region, which is the set of all points that satisfy the constraints. (Dantzig, 1947).

The Cutting-Plane method is particularly useful in situations where the solution space is continuous but needs to be restricted to integer points. The cutting-plane algorithm, often combined with other techniques like branch-and-bound, is used in large-scale ILP problems, especially when dealing with complex or non-convex feasible regions (Balas, 2022).

Duality in linear programming refers to the relationship between two optimization problems: the primal problem and its dual. The primal problem involves the direct optimization of an objective function subject to constraints, while the dual problem involves the optimization of a related objective function subject to dual constraints. The solutions to the primal and dual problems are related by the duality theorem, which states that under certain conditions, the optimal values of the primal and dual problems are equal. Duality plays a crucial role in understanding the properties of optimization problems, and it has important applications in sensitivity analysis and the development of algorithms like interior-point methods (Vanderbei, 2022).

The integration of the Simplex method with modern computational advancements has expanded its utility. Hybrid algorithms, combining branch-and-bound with cutting-plane techniques, harness the strengths of the Simplex method while leveraging computational power to tackle increasingly complex problems. Machine learning techniques have also been integrated with LIP to predict better initial solutions, speeding up the optimization process.

This paper successfully aimed to unify diverse requirement policies from various domains into a cohesive general model, utilizing the Simplex linear integer technique to ensure the optimal solution remains within the feasible region. This approach demonstrates a significant step forward in creating a unified framework while maintaining mathematical rigour.

## III. Results

**3.1 Adding, Subtracting, Or Scaling**: Let Z be the feasible solution set of an Integer Programming Problem (I.P.P.), defined by a system of linear constraints. If a new constraint is obtained by adding, subtracting, or scaling any of the existing constraints by a non-zero scalar, then every solution in Z satisfies the new constraint.

**Proof:**

Consider Z as the feasible solution set of an Integer Programming Problem (I.P.P.), defined by a system of linear constraints:

$$Ax \leq b, \qquad x \in Z^n$$

where A is an $m \times n$ matrix of coefficients, b is a vector of size m, and x is the vector of integer decision variables.

Assume a new constraint $c^T x \leq d$ is derived by manipulating the existing constraints through addition, subtraction, or scaling. This implies that $c^T$ and d are obtained as linear combinations of rows of A and elements of b, respectively. Specifically, the new constraint can be represented as:

$$c^T = \lambda_1 a_1^T + \lambda_2 a_2^T + \cdots + \lambda_m a_m^T$$

$$d = \lambda_1 b_1 + \lambda_2 b_2 + \cdots + \lambda_m b_m$$

where $\lambda_1, \lambda_2, \cdots, \lambda_m$, are scalars (not all zero), and $a_i^T$ are rows of A, with $b_i$ being the corresponding elements of b.

We now assess the Feasibility of Z under $Ax \leq b$. By definition, each solution $x \in Z$ satisfies:

$$a_i^T \leq b_i, for\ i = 1,2, \dots, m$$

Therefore, the feasible set Z consists of all $x \in Z^n$ that satisfies all these m constraints.

The new constraint $c^T x \leq d$ is derived from a valid linear combination of the original constraints:

$$\lambda_1(a_1^T \leq b_1), \lambda_2(a_2^T \leq b_2), \cdots, \lambda_m(a_m^T \leq b_m)$$

Adding or subtracting these constraints results in:

$$\lambda_1 a_1^T + \lambda_2 a_2^T + \cdots + \lambda_m a_m^T \leq \lambda_1 b_1 + \lambda_2 b_2 + \cdots + \lambda_m b_m$$

Which simplifies to:

$$c^T x \leq d$$

Since this new inequality is a linear combination of the original constraints, it is satisfied by any x that satisfies the original constraints.

Hence, everysolution $x \in Z$ satisfies all original constraints $Ax \leq b$. Because the new constraint $c^T x \leq d$ is a consequence of these constraints, every $x \in Z$ must also satisfy $c^T x \leq d$.

Therefore, the introduction of $c^T x \leq d$ does not eliminates any solutions in Z as the constraint is already satisfied by all $x \in Z$. Therefore, the feasible set Z remains unchanged.

We have shown that any new constraint derived by adding, subtracting, or scaling the existing constraints by a non-zero scalar is satisfied by every solution in Z. Hence, the theorem is proven. **Q.E.D.**

**3.2 Constraint Preservation in Integer Programming Problems:** Let Z be the feasible solution set of an Integer Programming Problem (I.P.P.), defined by a system of linear constraints $Ax \leq b$, where A is an $m \times n$ matrix, b is a vector of size m, and $x \in Z^n$. Suppose $x^*$ is a solution obtained by the Simplex technique that includes fractional parts for one or more variables. Then:

a. The solution $x^*$ satisfies all constraints in the original system $Ax \leq b$.

b. Any new constraint $c^T x \leq d$ derived by:

   - Adding or subtracting two or more constraints from the original system, or
   - Scaling any constraint by a non-zero scalar, is also satisfied by $x^*$ and all feasible integer solutions in Z

**Construction**

To proceed, we first introduce two key concepts as follows: Suppose $\rho = a\frac{1}{b}$

Let $\lceil \rho \rceil =$largest integral part of number $a$, i.e. the greatest integer less than $a$,

and $\mu =$positive fractional part of number $a$,

thus, we have $\rho = \lceil \rho \rceil + \mu$, clearly $0 \leq \mu < 1$, for example

(i) If $\rho = a\frac{1}{b}$ then $\lceil \rho \rceil = a$ and $\mu = \frac{1}{b}$ so that $a\frac{1}{b} = a + \frac{1}{b}$. $\forall\, b \in \mathbb{Z}^+$ and

(ii) If $\rho = -a\frac{1}{b}$ then $\lceil \rho \rceil = -(a+1)$ and $\mu = \frac{2}{b}$ so that

$$-a\frac{1}{b} = -\left(a + \frac{1}{b}\right) \geq -(a+1), \;\; \forall\, a, b \in \mathbb{Z}^+$$

Now, we proceed with the construction of the Gomory constraint as follows.

Let the optimal solution of the maximization Linear Programming Problem (L.P.P.), ignoring the integer requirements of the variables, be expressed in the following table: 3.1.

In this table: 3.1, the basic variables $x_{B_1}, x_{B_2}, \ldots, x_{B_m}$ are arranged in a convenient order for clarity.

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_1$ | $\cdots$ | $Y_i$ | $\cdots$ | $Y_m$ | $Y_{m+1}$ | $\cdots$ | $Y_n$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Y_1$ | $c_{B_1}$ | $x_{B_1}$ | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 0 | $Y_{1,m+1}$ | $\cdots$ | $Y_{1n}$ |
| $Y_2$ | $c_{B_2}$ | $x_{B_2}$ | 0 | 1 | $\cdots$ | 0 | $\cdots$ | 0 | $Y_{2,m+1}$ | $\cdots$ | $Y_{2n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $Y_i$ | $c_{B_i}$ | $c_{B_i}$ | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 0 | $Y_{i,m+1}$ | $\cdots$ | $Y_{in}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $Y_m$ | $c_{B_m}$ | $x_{B_m}$ | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 1 | $Y_{m,m+1}$ | $\cdots$ | $Y_{mn}$ |
| | | $x_J$ | $x_{B_1}$ | $x_{B_2}$ | $\cdots$ | $x_{B_i}$ | $\cdots$ | $x_{B_m}$ | 0 | $\cdots$ | 0 |

**Table: 3.1**

Let the $ith$ basic variable $x_{B_i}$ be a non-integer

Note that $1 \leq i \leq m$. Therefore, using $ith$ row of Table: 3.1, we have

$$x_{B_i} = 0.x_1 + 0.x_2 + \cdots + 0.x_m + y_{i,m+1}x_{m+1} + \cdots + y_{in}x_n$$

$$= x_i + \sum_{j=m+1}^{n} y_{ij}x_j$$

.

$$x_i = x_{B_i} - \sum_{j=m+1}^{n} y_{ij}x_j \quad \forall i = 1, 2, \ \cdots, n \qquad 3.1$$

Let $x_{B_i} = \lceil x_{B_i} \rceil + \mu_{B_i}$ and $y_{ij} = \lceil y_{ij} \rceil + \mu_{ij}$ where $\lceil x_{B_i} \rceil$ and $\lceil y_{ij} \rceil$ are the largest integral parts of $x_{B_i}$ and $y_{ij}$ respectively, while $\mu_{B_i}$ and $\mu_{ij}$ are the positive fractional parts of $x_{B_i}$ and $y_{ij}$ respectively.

Now, $x_{B_i} \geq \lceil x_{B_i} \rceil$and $y_{ij} \geq \lceil y_{ij} \rceil$, $0 \leq \mu_{B_i} < 1, 0 \leq \mu_{ij} < 1$ then, by equation 3.1, we have;

$$x_i = \lceil x_{B_i} \rceil + \mu_{B_i} - \sum_{j=m+1}^{n} \left(\lceil y_{ij} \rceil + \mu_{ij}\right)x_j \quad \forall i = 1, 2, \ \cdots, n$$

$$\mu_{B_i} - \sum_{j=m+1}^{n} \mu_{ij}x_j = x_i - \lceil x_{B_i} \rceil + \sum_{j=m+1}^{n} \lceil y_{ij} \rceil x_j, \forall i = 1, 2, \ \cdots, n \qquad 3.2$$

Thus, if $x_i, \ \forall i = 1, 2, \ \cdots, n$ and $x_j, \ \forall j = m + 1, \cdots, n$ are all integers then,

$$x_i - \lceil x_{B_i} \rceil + \sum_{j=m+1}^{n} \lceil y_{ij} \rceil x_j, \forall i = 1, 2, \ \cdots, n \text{ , thus,}$$

$$\mu_{B_i} - \sum_{j=m+1}^{n} \mu_{ij} x_j, \qquad \forall i = 1, 2, \ \cdots, n, \text{is an integer}$$

It follows that;

$$\sum_{j=m+1}^{n} \mu_{ij} x_j, \qquad \forall i = 1, 2, \ \cdots, n, \qquad \text{is positive, therefore,}$$

$$\mu_{B_i} - \sum_{j=m+1}^{n} \mu_{ij} x_j \leq \mu_{B_i} < 1, \qquad \forall i = 1, 2, \ \cdots, n, \qquad \textit{that is,}$$

$$\mu_{B_i} - \sum_{j=m+1}^{n} \mu_{ij} x_j, \qquad \forall i = 1, 2, \ \cdots, n, \qquad \text{is an integer less than one.}$$

Thus, it is either a zero or a negative integer. Hence,

$$\mu_{B_i} - \sum_{j=m+1}^{n} \mu_{ij} x_j \leq 0, \quad \text{or}$$

$$- \sum_{j=m+1}^{n} \mu_{ij} x_j \ \leq -\mu_{B_i} \ \forall i = 1, 2, \ \cdots, n, \qquad 3.3$$

Equation 3.3 is called the Gomory constraint. Applying the non-negative slack variable, we have;

$$- \sum_{j=m+1}^{n} \mu_{ij} x_j \ + x_{G1} \leq -\mu_{B_i} \ \ \forall i$$
$$= 1, 2, \ \cdots, n, \qquad 3.4$$

Adding equation 3.4, Table 3.1, we have; Table: 3.2

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_1$ | $\cdots$ | $Y_i$ | $\cdots$ | $Y_m$ | $Y_{m+1}$ | $\cdots$ | $Y_n$ | $x_{G1}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Y_1$ | $c_{B_1}$ | $x_{B_1}$ | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 0 | $Y_{1,m+1}$ | $\cdots$ | $Y_{1n}$ | 0 |
| $Y_2$ | $c_{B_2}$ | $x_{B_2}$ | 0 | 1 | $\cdots$ | 0 | $\cdots$ | 0 | $Y_{2,m+1}$ | $\cdots$ | $Y_{2n}$ | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ | $\vdots$ |
| $Y_i$ | $c_{B_i}$ | $c_{B_i}$ | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 0 | $Y_{i,m+1}$ | $\cdots$ | $Y_{in}$ | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ $\cdots$ | $\vdots$ | $\vdots$ $\cdots$ | $\vdots$ | $\vdots$ | | $\vdots$ | $\vdots$ |
| $Y_m$ | $c_{B_m}$ | $x_{B_m}$ | 0 | 0 | | 0 | | 1 | $Y_{m,m+1}$ | | $Y_{mn}$ | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\vdots$ |
| $Y_{G1}$ | 0 | $-\mu_{B_i}$ | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 0 | $\mu_{i,m+1}$ | | $\mu_{in}$ | 1 |
| | | $x_J$ | $x_{B_1}$ | $x_{B_2}$ | $\cdots$ | $x_{B_i}$ | $\cdots$ | $x_{B_m}$ | 0 | $\cdots$ | 0 | $-\mu_{B_i}$ |

**Table: 3.2**

Since $-\mu_{B_i}$ is negative, the optimal solution derived from Table: 3.2 is not feasible. Therefore, we apply the dual simplex algorithm to obtain the optimal feasible solution. If all the variables in the resulting solution are integers, the process ends. Otherwise, we construct the second Gomory constraint from the resulting Simplex-Gomory table (Table: 3.2), introduce it into the next iterative table, and solve it using the dual simplex algorithm. This process is repeated until an integer solution is obtained.

## IV. NUMERICAL APPLICATION

### Application 4.1

The Innoson Vehicle manufacturing plant in Umudim Nnewi, Anambra state, has a factory that produces cars and buses, as well as a showroom for selling these vehicles. The company earns a profit of \$3,000.00 and \$12,000.00 per car and bus, respectively. The company has two subdirectors, C and B, who can each work a maximum of 7 hours and 15 hours per day, respectively. Both subdirectors are responsible for supervising both cars and buses. Subdirector C spends 2 hours supervising a car and 4 hours supervising a bus, while subdirector B spends 5 hours supervising a car and 3 hours supervising a bus. To maximize daily profit, how many cars and buses should be supervised? (a) Formulate and solve this problem as an LP problem. (b) If the optimal solution is not integer-valued, use the technique in equation 3.1 to derive the optimal integer solution.

### Mathematical formulation

Let $x_1$ and $x_2$ be the number of cars and buses to be supervised daily, respectively. Thus, the mathematical model of the LP problem is stated as:

$$\text{Maximize } Z = \$3000x_1 + \$12000x_2$$

Subject to the time constraints

$$\text{supervisor C}: \ 2x_1 + 4x_2 \leq 7$$

$$\text{supervisor B}: \ 5x_1 + 3x_2 \leq 15$$

$$x_1, x_2 \geq 0 \text{ and are integers.}$$

### Solution

(a) Adding slack variable $x_3$ and $x_4$, the given LP problem is follows:

$$\text{Maximize } Z = 3x_1 + 12x_2 + 0x_3 + 0x_4$$

Subject to the time constraints

$$2x_1 + 4x_2 + x_3 + 0x_4 = 7$$

$$5x_1 + 3x_2 + 0x_3 + x_4 = 15$$

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | Min. Ratio |
|---|---|---|---|---|---|---|---|
| $Y_3$ | 0 | 7 | 2 | 4 | 1 | 0 | $7/4 \rightarrow$ |
| $Y_4$ | 0 | 15 | 5 | 3 | 0 | 1 | 5 |
| | | $\Delta_J$ | 3000 | 12000 $\uparrow$ | 0 | 0 | |

Table: 4.1

Now, using the Simplex method we have; Table: 4.1

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ |
|---|---|---|---|---|---|---|
| $Y_2$ | 12000 | 1.75 | 0.5 | 1 | 0.5 | 0.25 |
| $Y_4$ | 0 | 9.75 | 3.5 | 0 | $-0.75$ | 1 |
| | | $\Delta_J$ | $-3000$ | 0 | $-3000$ | 0 |

Table: 4.2 (Optimal solution is not integer-valued)

The non-integer optimal solution shown in Table: 4.2 is $x_1 = 0$; $x_2 = \frac{7}{4}$; $x_4 = 0$ *and* $MaxZ = 21{,}000$

b) To obtain the integer-valued solution, we proceed to construct Gomory's Fractional cut, with the help of $x_2$ - row (because it has largest fraction value) s follows.

$$\frac{1}{2}x_1 + x_2 + \frac{1}{4}x_3 + 0x_4 = \frac{7}{4}$$

$$\left(\frac{1}{2} + 0\right)x_1 + (1 + 0)x_2 + \left(\frac{1}{4} + 0\right)x_3 + 0x_4 = \left(\frac{3}{4} + 1\right)$$

$$\frac{3}{4} + (1 - x_2) - \frac{1}{2}x_1 - \frac{1}{4}x_3 = 0$$

$$\frac{3}{4} \leq \frac{1}{2}x_1 + \frac{1}{4}x_3$$

On adding Gomory Slack variable $x_{g_1}$, the required Gomory's fractional cut becomes:

$$\frac{3}{4} = \frac{1}{2}x_1 + \frac{1}{4}x_3 + x_{g_1}$$

Now, by introducing the new constraint equation to the bottom of the optimal simplex Table-4.2, we have a new infeasible Simplex Gomory iterative table: 4.2.1.

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_{g_1}$ |
|---|---|---|---|---|---|---|---|
| $Y_2$ | 12000 | 1.75 | 0.5 | 1 | 0.25 | 0 | 0 |
| $Y_4$ | 0 | 9.75 | 3.5 | 0 | −0.75 | 1 | 0 |
| $Y_{g_1}$ | 0 | −0.75 | 0.5 | 0 | −0.25 | 0 | 1 |
| | | Min. Ratio | 6000 | 0 | 3000 | 0 | 0 |

Iteration Table: 4.2.1:

The outgoing vector is $Y_{g_1}$ from the basis and entering vector $Y_1$ into the basis by applying the dual simplex method. The new solution is shown in

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_{g_1}$ |
|---|---|---|---|---|---|---|---|
| $Y_2$ | 12000 | 1 | 0 | 1 | 0 | 0 | 1 |
| $Y_4$ | 0 | 4.5 | 0 | 0 | −2.5 | 1 | 7 |
| $Y_1$ | 3000 | 0.75 | 1 | 0 | 0.5 | 0 | 2 |
| | | $\Delta_J$ | −3000 | 0 | 1500 | 0 | − 6000 |

Iteration Table: 4.2.2;

Still, the optimal solution shown in Table: 4.2.2 is non-integer. Therefore, by adding one more fractional cut, with the help of the$Y_1$-row, we have:

$$\frac{3}{2} = x_1 - \frac{1}{2}x_3 - \frac{1}{4}x_3 - 2x_{g_1}$$

$$\left(\frac{1}{2} + 1\right) = (1+0)x_1 - \left(\frac{1}{2} + 0\right)x_3 - (-2+0)x_{g_1}$$

$$\frac{1}{2} + \left(1 - x_1 + 2x_{g_1}\right) = \frac{1}{2}x_3$$

$$\frac{1}{2} \leq \frac{1}{2}x_3$$

On adding Gomory slack variable $Y_{g_2}$, the required Gomory's fractional cut becomes:

$$-\frac{1}{2} = -\frac{1}{2}x_3 + 2x_{g_2}$$

Adding this cut to the optimal simplex table-4.2.3, the new table so obtained is shown in table: 4.2.3

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_{g_1}$ | $Y_{g_2}$ |
|---|---|---|---|---|---|---|---|---|
| $Y_2$ | 12000 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| $Y_4$ | 0 | 4.5 | 0 | 0 | $-2.5$ | 1 | 7 | 0 |
| $Y_1$ | 3000 | 0.75 | 1 | 0 | 0.5 | 0 | 2 | 0 |
| $Y_{g_2}$ | 0 | $-0.5$ | 0 | 0 | $-0.5$ | 0 | 0 | 1 → |
| | | $\Delta_J$ | $-3000$ | 0 | $-1500$ | 0 | $-6000$ | 0 |
| | | Min. Ratio | 0 | 0 | 1500 ↑ | 0 | 0 | 0 |

Iteration Table: 4.2.3

The outgoing vector $Y_{g_2}$ from the basis and entering vector is $Y_3$ into the basis by applying the dual simplex method. The new solution is shown in table:4.2.4.

| $B$ | $C_B$ | $X_B$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_{g_1}$ | $Y_{g_2}$ |
|---|---|---|---|---|---|---|---|---|
| $Y_2$ | 12000 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| $Y_4$ | 0 | 7 | 0 | 0 | 0 | 1 | 7 | $-5$ |
| $Y_1$ | 3000 | 1 | 1 | 0 | 0 | 0 | 2 | 1 |
| $Y_3$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | $-2$ |
| | | $\Delta_J$ | 0 | 0 | 0 | 0 | $-6000$ | $-3000$ |

Table: 4.2.4

Therefore, since all the variables in Table: 4.2.4 have integer values and all $\Delta_J \leq 0$, the solution is an integer optimal solution. It is recommended that the company should produce one Car and one Bus each day to yield Max (profit)$Z = \$15{,}000$. Similarly, we can solve different examples with the help of the algorithm of Gomory's mixed integer cutting plane and zero-one integer programming.

**Conclusion**

The evolution of linear programming (LP) optimization techniques illustrates the dynamic relationship between theoretical innovation and practical application. From its mathematical foundations to modern computational advancements, LP has become an essential tool for solving optimization problems. The development of efficient algorithms, such as the Simplex and interior-point methods, has played a crucial role in making LP both widely accessible and impactful. The advancement of Simplex-based approaches for linear integer programming highlights the synergistic connection between mathematical theory and practical application. From its origins in linear programming to its pivotal role in solving discrete optimization problems, the Simplex method has proven to be remarkably adaptable and enduringly relevant. The continuous evolution of computational techniques ensures that the legacy of the Simplex method will continue to shape the field of optimization, enabling solutions to increasingly complex challenges in science, industry, and beyond.

## REFERENCES

Balas, E. (2022). Cutting-plane algorithms for integer programming. Mathematical Programming, 165(1), 1-20. https://doi.org/10.1007/s10107-021-01515-x

Bertsimas, D., &Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization. Athena Scientific.*

Chauhan, S., & Dey, A. (2022). Application of integer programming in supply chain optimization: A case study. Operations Research Perspectives, 9, 100-112. https://doi.org/10.1016/j.orp.2021.100112

Cui, X., Zhang, L., & Liu, S. (2023). Optimization of renewable energy distribution using Simplex-based linear integer programming. Energy Systems, 14(4), 123-135. https://doi.org/10.1007/s12667-023-00415-6

Dantzig, G. B. (1947). Maximization of a linear function of variables subject to linear inequalities. U.S. Air Force Project RAND.

Dantzig, G. B. (1947). Linear programming and extensions. Princeton University Press.

Del Pia, A. (2023). Convex quadratic sets and the complexity of mixed integer convex quadratic programming. Mathematics, Computer Science. TLDR; Corpus ID: 264832926

Farkas, J. (1902). Über die Theorie der linearenUngleichungen. Journal of the Austrian Academy of Sciences, 111(1), 1-25.

Fourier, J. (1824). The analytical theory of heat. Cambridge University Press.

Gomory, R. E. (1958). Outline of an algorithm for integer solutions to linear programs. Bulletinof the American Mathematical Society, 64(5), 275–287. https://doi.org/10.1090/S0002-9904-1958-10223-3

Hillier*, F. S., & Lieberman, G. J. (2020). Introduction to Operations Research. McGraw Hill Education.*

Laisin, M., Edike, C. and Bright O. Osu (2024); The construction of rational polyhedron on an $n \times n$ board with some application on integral polyhedral. TIJER, ISSN 2349-9249, Vol 11, Issue 11, www.tijer.org.

Laisin, M., & Edike, C. (2025a). Hybrid Optimization with Integer Constraints: Modeling and Solving Problems Using Simplex Techniques. *Journal of Medicine, Engineering & Physical Sciences (JOMEEPS)*. https://klamidas.com/jomeeps-v3n1-2025-01/

Laisin, M., &Adigwe, R. U. (2025b). Implementation and comparative analysis of AMGT method in Maple 24: Convergence performance in optimization problems. *Global Online Journal of Academic Research (GOJAR)*, *4*(52), 26–40. https://klamidas.com/gojar-v4n1-2025-02/

Laisin, M., &Edike, C. (2025c) Characterizing Boundedness and Solution Size in Rational Linear Programming and Polyhedral Optimization. *Journal of Medicine, Engineering & Physical Sciences (JOMEEPS)*. https://klamidas.com/jomeeps-v3n1-2025-01/

Land, A. H., & Doig, A. G. (1960). An automatic method for solving discrete programming problems. *Econometrica, 28*(3), 497–520. https://doi.org/10. 2307/1910129

Land, A. H., & Doig, A. G. (2021). The computational complexity of integer programming. Operations Research, 19(3), 423-431. https://doi.org/10.1287/ opre.19.3.423

López, M., García, J., & Sánchez, P. (2021). A hybrid Simplex approach for large-scale transportation problems. Journal of Optimization Theory and Applications, 190(2), 45-60. https://doi.org/10.1007/s10957-020-01746-1

Nemhauser, *G. L., & Wolsey, L. A. (1999). Integer and Combinatorial Optimization. New York: Wiley.Schrijver, A. (1986). Theory of Linear and Integer Programming. Wiley-Interscience.*

Vanderbei, R. J. (2022). Linear programming: Foundations and extensions (5th ed.). Springer. https://doi.org/10.1007/978-3-030-54928-4

Wolsey, L. A. (1998). Integer Programming. Wiley-Inter science

Winston*, W. L. (2004). Operations Research: Applications and Algorithms. Belmont: Thomson Brooks/Cole.)*

Zhao, T., & Kim, J. (2020). Hybrid approaches for scheduling in manufacturing: A Review. Computers & Industrial Engineering, 150, https://doi.org/10.1016/j.cie.2020.106116

**Author Information:** Mark Laisin is a Professor of Applied Mathematics at Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria. *Email*: laisinmark@gmail.com

Collins Edike is of the Department of Mathematics, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria. *Email*: edikecollins505@gmail.com

**CITING THIS ARTICLE**

**APA**

Laisin, M. & Edike, C. (2025). Hybrid Optimization with Integer Constraints: Modeling and Solving Problems Using Simplex Techniques. *Global Online Journal of Academic Research* (*GOJAR*), 4(2), 22-38. https://klamidas.com/gojar-v4n2-2025-02/.

**MLA**

Laisin, Mark and Edike, Collins. "Hybrid Optimization with Integer Constraints: Modeling and Solving Problems Using Simplex Techniques". *Global Online Journal of Academic Research* (*GOJAR*), vol. 4, no. 2, 2025, pp. 22-38. https://klamidas.com/gojar-v4n2-2025-02/.