forecasting

Article

Solving Linear Integer Models with Variable Bounding

Elias Munapo [1], Joshua Chukwuere [2] and Trust Tawanda [3,*]

[1] Department of Business Statistics and Operations Research, School of Economic Sciences, North West University, Mafikeng Campus, Mmabatho 2745, South Africa; elias.munapo@nwu.ac.za

[2] Department of Information Sciences, School of Economic Sciences, North West University, Mafikeng Campus, Mmabatho 2745, South Africa; josh.chukwuere@nwu.ac.za

[3] Department of Statistics and Operations Research, National University of Science and Technology, Ascot, Bulawayo P.O. Box AC 939, Zimbabwe

* Correspondence: trust.tawanda@nust.ac.zw

Abstract: We present a technique to solve the linear integer model with variable bounding. By using the continuous optimal solution of the linear integer model, the variable bounds for the basic variables are approximated and then used to calculate the optimal integer solution. With the variable bounds of the basic variables known, solving a linear integer model is easier by using either the branch and bound, branch and cut, branch and price, branch cut and price, or branch cut and free algorithms. Thus, the search for large numbers of subproblems, which are unnecessary and common for NP Complete linear integer models, is avoided.

Keywords: continuous optimal solution; linear integer model; basic variable bound; NP complete and sub-problem

1. Introduction

It is challenging to solve the NP Complete linear integer programming (LIP) model using the available methods [1–3]. Linear integer models are used when the continuity assumption in linear programming (LP) does not hold. For discrete quantities such as the number of computer machines, aircraft, automobiles, personnel, warehouses, or stations, continuity is not a strong assumption. Several approaches can be used to solve this model, including branch and bound (BB) [4,5], branch and price (BP) [6], branch and cut (BC) [7], branch cut and price (BCP) [8], and branch cut and free (BCF) [9] algorithms. In addition, reformulation techniques that can help reduce the complexity of LIP models exist [10]. The LIP models can also be solved by using a characteristic equation [1]. A hypermetaheuristic approach to solving mixed integer optimization problems was proposed by [11]. Data envelopment analysis (DEA)-related problems were solved to prove the efficiency and validity of the method; however, proving the validity of DEA-related problems only limited the study. A new strategy to solve linear integer programming problems with simplex directions was presented by [12]. To prove the validity of the method, the results were compared with CPLEX software results. With this method, the authors managed to compute accurate solutions for 34 out of 100 problems. Mixed integer linear programming was used by [13] to solve the station capacity allocation problem of a star-tree pipe network. The results of the model proved the validity of the method. Integer linear programming was applied in [14] to optimize energy-aware high-level synthesis and reliability. The results were compared with a genetic algorithm for several problems, and the integer linear programming approach resulted in optimal solutions. Mixed integer models were applied in [15] to optimize a vanadium redox flow battery with electrolyte maintenance, variable efficiencies, and capacity fade. The results proved that the problem was solved in a more accurate way when using this method as compared with other simpler methods. The branch and bound approach was enhanced in [16] to solve the knapsack linear integer

check for updates

Citation: Munapo, E.; Chukwuere, J.; Tawanda, T. Solving Linear Integer Models with Variable Bounding. *Forecasting* **2023**, *5*, 443–452. https://doi.org/10.3390/forecast5020024

Academic Editor: Sonia Leva

Received: 4 October 2022
Revised: 28 April 2023
Accepted: 4 May 2023
Published: 5 May 2023

problem. The problems were solved as subproblems that were less complex and required fewer iterations to obtain the optimal solution.

We present a way to find a bound such that a predetermined number of integers is accommodated in a calculated variable range. Using the continuous optimal solution of the linear integer model, the integer bounds for the basic variables are approximated and then used to calculate the optimal integer solution. The search for large numbers of subproblems, which are unnecessary and common for NP Complete linear integer models, is avoided. Linear integer models are NP-hard problems; thus, computing the optimal solution in a reasonable computational time is remarkably difficult. Combining metaheuristics and exact solution approaches is an efficient and effective solution. The variable-bounding method that we propose helps one to determine the variable bounds for linear integer models and thus reduces the number of iterations when applying exact-solution approaches such as branch and bound and branch and cut, among others, to solve the linear integer models. The approximated integer bounds for the variables are utilized when calculating the optimal integer solutions. The exact methods require fewer iterations when the integer bounds for the variables are approximated first; thus, the search for a large number of subproblems, which are unnecessary, is avoided. Essentially, the main contribution of this research is the development of the variable-bounding method, which can be used to approximate the variable integer bounds for the variables; the approximated bounds will be utilized to reduce the complexity and the number of iterations when applying the exact-solution approaches.

2. The Linear Integer Programming (LIP)

$$\begin{array}{c}\text{Maximize } Y_o = c_1y_1 + c_2y_2 + \ldots + c_ny_n,\\ \text{Such that :}\\ a_{11}y_1 + a_{12}y_2 + \ldots + a_{1n}y_n \le b_1,\\ a_{21}y_1 + a_{22}y_2 + \ldots + a_{2n}y_n \le b_2,\\ \ldots\\ a_{m1}y_1 + a_{m2}y_2 + \ldots + a_{mn}y_n \le b_m,\end{array} \tag{1}$$

where a_{ij}, b_j and c_i are integers, Y_o is the objective value, $y_i \ge 0$ and an integer, i = 1,2,…,m and $j = 1, 2, \ldots, n$.

3. The Continuous Optimal Table

The integer restrictions on the LIP model, which restrict integers that are difficult to use, can be relaxed. This is performed by changing the LIP model into an ordinary linear program (LP), as the LP is easier to solve than the LIP version. The relaxed LP form is solved to obtain a continuous optimal solution, which is provided in Table 1.

Table 1. LP continuous optimal table.

	y_1	y_2	…	y_m	s_1	s_2	…	s_m	**R.H.S**
Y_o	0	0	…	0	λ_1	λ_2	…	λ_m	R
y_1	1	0	…	0	π_{11}	π_{12}	…	π_{1m}	β_1
y_2	0	1	…	0	π_{21}	π_{22}	…	π_{2m}	β_2
…	…	…	…	…	…	…	…	…	…
y_m	0	0	…	1	π_{m1}	π_{m2}	…	π_{mm}	β_m

where $\lambda_1, \lambda_2, \ldots, \lambda_m, \beta_1, \beta_2, \ldots, \beta_m$ are non-negative constants, whereas π_{ij} and R are constants. The variables $y_1, y_2, \ldots, y_m$ are made basic and $s_1, s_2, \ldots, s_m$ are made nonbasic for convenience. Any other arrangement is also possible.

4. Determining the Expressions for the Basic Variable Limits

$$\text{Let } \mathrm{\lambda_1 s_1 + \lambda_2 s_2 + \ldots + \lambda_m s_m \ge \Delta Y_0} \tag{2}$$

where ΔY_0 is a reduction in the optimal objective value R. By introducing a slack variable s_{m+1}, the constraint becomes Equation (3)

$$-\lambda_1 s_1 - \lambda_2 s_2 - \ldots - \lambda_m s_m + s_{m+1} = -\Delta Y_0 \tag{3}$$

By adding this extra row to the current optimal solution, it becomes Table 2.

Table 2. Adding a new constraint.

	y_1	y_2	$\ldots$	y_m	s_1	s_2	$\ldots$	s_m	s_{m+1}	**R.H.S**
Z	0	0	$\ldots$	0	λ_1	λ_2	$\ldots$	λ_m	0	R
y_1	1	0	$\ldots$	0	π_{11}	π_{12}	$\ldots$	π_{1m}	0	β_1
y_2	0	1	$\ldots$	0	π_{21}	π_{22}	$\ldots$	π_{2m}	0	β_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
y_m	0	0	$\ldots$	1	π_{m1}	π_{m2}	$\ldots$	π_{mm}	0	β_m
s_{m+1}	0	0	$\ldots$	0	$-\lambda_1$	$-\lambda_2$	$\ldots$	$-\lambda_m$	1	$-\Delta Y_0$

We then applied the dual simplex procedure, and n possible solutions were found. The first basic variable in the first solution is of the form provided in Equation (4):

$$y_1 = \beta_1 + h_1^1 \Delta Y_0, \forall i. \tag{4}$$

The second basic variable in the first solution is of the form provided in Equation (5):

$$y_2 = \beta_2 + h_2^1 \Delta Y_0, \forall i. \tag{5}$$

The mth basic variable in the mth solution is of the form provided in Equation (6):

$$y_m = \beta_m + h_m^1 \Delta Y_0, \forall i. \tag{6}$$

where $h_{i}^{\bar{i}}$ is a constant and $\bar{i} = 1, 2, ..., m$

Thus,

$$\begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_m \end{bmatrix} = \begin{bmatrix} \beta_1 + h_1^1 \Delta Z \\ \beta_2 + h_2^1 \Delta Z \\ \ldots \\ \beta_m + h_m^1 \Delta Z \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_m \end{bmatrix} = \begin{bmatrix} \beta_1 + h_1^2 \Delta Z \\ \beta_2 + h_2^2 \Delta Z \\ \ldots \\ \beta_m + h_m^2 \Delta Z \end{bmatrix}, \ldots, \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_m \end{bmatrix} = \begin{bmatrix} \beta_1 + h_1^m \Delta Z \\ \beta_2 + h_2^m \Delta Z \\ \ldots \\ \beta_m + h_m^m \Delta Z \end{bmatrix} \tag{7}$$

From these m possible solutions, the bounds for the variables that are nonbasic can be determined using the formula provided in Equation (8):

$$\beta_i + \ell_i^L \Delta Y_0 \leq y_i \leq \beta_i + \ell_i^U \Delta Y_0 \tag{8}$$

where

$$\ell_i^L = \min\left[h_i^1, h_i^2, \ldots, h_i^m\right] \tag{9}$$

and

$$\ell_i^U = \max\left[h_i^1, h_i^2, \ldots, h_i^m\right] \tag{10}$$

5. Arranging Variables in the Order of Their Restrictions

Arranging the variables in an order that begins with the most restricted variables will reduce the number of iterations during the searching process. The approaches that are branch-and-bound-related converge fast if the first iteration begins with the most-restricted variables.

5.1. Ordering Branching Variables

Branching must always begin with the most-restricted variables. In this case, the variable with the most restrictions is defined as the variable with the lowest number of possible integral values.

Proof. Let the number of possible integer values for variable x_i be provided by I_i.

Thus,

$$\mathrm{n}(\mathrm{x_i}) = \omega_\mathrm{i} \tag{11}$$

Thus, the number of possible integers for the various variables are provided in Equation (12):

$$\begin{gathered}\mathrm{n}(\mathrm{x_1}) = \omega_1,\\ \mathrm{n}(\mathrm{x_2}) = \omega_2,\\ \ldots\\ \mathrm{n}(\mathrm{x_m}) = \omega_\mathrm{m}.\end{gathered} \tag{12}$$

Additionally, let the number of branches in ascending order be Equation (13):

$$\omega_1 \leq \omega_2 \leq \ldots \leq \omega_\mathrm{m} \tag{13}$$

Thus, variable y_1 is the most restricted variable, followed by y_2 and then variable y_3 in that order up to variable y_m. The following number of nodes are visited when constructing the search tree. In this type of search process, the worst-case scenario is assumed. □

5.2. Starting with the Most Restricted Variable

Step 1: Number of iterations taken $= \omega_1$
Step 2: Number of iterations taken $= \omega_1\omega_2$
Step m: Number of iterations taken $= \omega_1\omega_2\ldots\omega_\mathrm{m}$
Let the total number of iterations $(\sum \text{iterations}(\mathrm{A}))$ be provided by Equation (14):

$$\sum \text{iterations}(\mathrm{A}) = \omega_1 + \omega_1\omega_2 + \ldots + \omega_1\omega_2\ldots\omega_\mathrm{m} \tag{14}$$

5.3. Starting with the Variable with the Least Restriction

Step 1: Number of iterations taken $= \omega_m$
Step 2: Number of iterations taken $= \omega_m\omega_{m-1}$
Step m: Number of iterations taken $= \omega_m\omega_{m-1}\ldots\omega_1$
Let the total number of iterations $(\sum \text{iterations}(\mathrm{B}))$ be provided by Equation (15):

$$\sum \text{iterations}(\mathrm{B}) = \omega_\mathrm{m} + \omega_\mathrm{m}\omega_\mathrm{m-1} + \ldots + \omega_\mathrm{m}\omega_\mathrm{m-1}\ldots\omega_1 \tag{15}$$

Therefore,

$$\sum \text{iterations}(\mathrm{A}) \leq \sum \text{iterations}(\mathrm{B}) \tag{16}$$

More information on the search process can be found in [17], and in this search process, the worst-case scenario is assumed. Let the restriction index on variable y_i be provided by γ_i; then, we have Equations (17) and (18):

$$\gamma_\mathrm{i} = \ell_\mathrm{i}^\mathrm{U}\Delta\mathrm{Z} - \ell_\mathrm{i}^\mathrm{L}\Delta\mathrm{Z} \tag{17}$$

$$\gamma_\mathrm{i} = \varphi_\mathrm{i}\Delta\mathrm{Z} \tag{18}$$

where $\varphi_i = \ell_i^U - \ell_i^L$. Thus, the variables can be arranged in their restricted order by arranging their indices in ascending order. The smaller the value of φ_i, the more restricted the variable y_i is. We present a way to find a bound such that a predetermined number of integers is accommodated in a calculated variable range.

6. Calculating the Basic Variable Integral Bounds

The decrease in the objective value (ΔY_0) can be obtained by solving the following single variable LP:

$$\begin{gathered}\text{Minimize} = \Delta Y_0, \\ \text{Such that}: \\ \varphi_1 \Delta Y_0 \geq \omega, \\ \varphi_2 \Delta Y_0 \geq \omega, \\ \ldots \\ \varphi_m \Delta Y_0 \geq \omega\end{gathered} \tag{19}$$

where ω is a range that accommodates at least an integer. The optimal solution to Equation (19) is readily available and is provided by Equation (20):

$$\Delta \mathrm{Y}_0 = \mathrm{Max}\left[\frac{\omega}{\varphi_1}, \frac{\omega}{\varphi_2}, \ldots, \frac{\omega}{\varphi_m}\right] \tag{20}$$

6.1. Proof

The LP model in Equation (19) can be rewritten as Equation (21):

$$\begin{gathered}\text{Minimize} = \Delta \mathrm{Y}_0, \\ \text{such that} \\ \Delta \mathrm{Y}_0 \geq \frac{\omega}{\varphi_1}, \\ \mathrm{Y}_0 \geq \frac{\omega}{\varphi_2}, \\ \ldots \\ \mathrm{Y}_0 \geq \frac{\omega}{\varphi_m}\end{gathered} \tag{21}$$

The dual variable of Equation (21) becomes that provided in Equation (22):

$$\begin{gathered}\text{Maximize} \frac{\omega}{\varphi_1} x_1 + \frac{\omega}{\varphi_1} x_2 + \ldots + \frac{\omega}{\varphi_1} x_m, \\ \text{suchthat} \\ x_1 + x_2 + \ldots + x_m \leq 1\end{gathered} \tag{22}$$

where x_i is the dual variable.

6.2. Determining the Value of ω

The number ω can be determined in many ways. One of these ways is to use the interval provided in Equation (23):

$$1 < \omega \tag{23}$$

This interval is valid because the shortest distance between any two integers is one. If the distance is less than one, then we may not find an integer in the interval:

$$2 \leq \omega \tag{24}$$

It may seem logical to start from two onwards:

$$\omega = 2, 3, 4, \ldots \tag{25}$$

The range ω is not necessarily an integer; in this case, we are using integers for convenience. Any value of ω greater than one may be used. Once the restriction expressions for the basic variables are known, the variable bounds can be determined. From the most restricted variable y_i, the decrease in the objective value can be calculated given that at least two integers are accommodated in the bounds as provided in Equation (26):

$$\Delta \mathrm{Y}_0 = \mathrm{Min}\left[\frac{1 + \mathrm{f}[\beta_\mathrm{i}]}{\ell_\mathrm{i}^\mathrm{U}} \text{ or } \frac{2 - \mathrm{f}[\beta_\mathrm{i}]}{\ell_\mathrm{i}^\mathrm{U}}, \frac{1 + \mathrm{f}[\beta_\mathrm{i}]}{\ell_\mathrm{i}^\mathrm{L}} \text{ or } \frac{2 - \mathrm{f}[\beta_\mathrm{i}]}{\ell_\mathrm{i}^\mathrm{L}}\right] \tag{26}$$

where $f[\beta_i]$ is the fractional part of β_i. $\frac{1+f[\beta_i]}{\ell_i^U}$ and $\frac{1+f[\beta_i]}{\ell_i^L}$ are used when $\ell_i^U < 0$ and $\ell_i^L < 0$, respectively, and $\frac{2-f[\beta_i]}{\ell_i^U}$ and $\frac{2-f[\beta_i]}{\ell_i^L}$ are used when $\ell_i^U > 0$ and $\ell_i^L > 0$, respectively.

Proof. The expression $\beta_i + \ell_i^L \Delta Y_0 \leq y_i \leq \beta_i + \ell_i^U \Delta Y_0$, provided in Equation (8), can be written as Equation (27):

$$\bar{\beta}_i + f[\beta] + \ell_i^L \Delta Y_0 \leq y_i \leq \bar{\beta}_i + f[\beta] + \ell_i^U \Delta Y_0 \tag{27}$$

where $\bar{\beta}_i$ is the integer and $f[\beta_i]$ is the fractional part of β_i.

For the bounds $\bar{\beta}_i + f[\beta] + \ell_i^L \Delta Y_0$ and $\bar{\beta}_i + f[\beta] + \ell_i^U \Delta Y_0$ to be integers, $f[\beta] + \ell_i^L \Delta Y_0$ and $f[\beta] + \ell_i^U \Delta Y_0$ must be integers. This is possible as per Equations (28) and (29).

$$f[\beta] + \ell_i^L \Delta Y_0 = 1, 2, 3, \ldots \tag{28}$$

$$[\beta] + \ell_i^U \Delta Y_0 = 1, 2, 3, \ldots \tag{29}$$

For $\ell_i^U > 0$ and $\ell_i^L > 0$, respectively, at least two integers are within the bounds of the most restricted variable provided in Equation (30):

$$f[\beta] + \ell_i^L \Delta Y_0 = 2 \tag{30}$$

and

$$f[\beta] + \ell_i^U \Delta Y_0 = 2 \tag{31}$$

Thus, $\Delta Y_0 = \frac{2-f[\beta_i]}{\ell_i^U}$ and $\Delta Y_0 = \frac{2-f[\beta_i]}{\ell_i^L}$.

Similarly,

$$f[\beta] - \ell_i^L \Delta Y_0 = 0, -1, -2, \ldots \tag{32}$$

$$f[\beta] - \ell_i^U \Delta Y_0 = 0, -1, -2, \ldots \tag{33}$$

because $\ell_i^U < 0$ and $\ell_i^L < 0$.

Additionally, at least two integers are present in between the bounds provided in Equations (34) and (35):

$$f[\beta] - \ell_i^L \Delta Y_0 = -1 \tag{34}$$

$$f[\beta] - \ell_i^U \Delta Y_0 = -1 \tag{35}$$

Thus, $\Delta Y_0 = \frac{1+f[\beta_i]}{\ell_i^U}$ and $\Delta Y_0 = \frac{1+f[\beta_i]}{\ell_i^L}$.

The value of ΔY_0 is substituted to obtain the bounds for all the basic variables, and once these are available, they can easily be changed to integers, as shown in Equation (36):

$$I_i^L \leq y_i \leq I_i^U \tag{36}$$

where I_i^L and I_i^U are the lower and upper integer limits, respectively. □

7. Determining the Bound for the Liner Integer Model

A linear integer model can usually be solved with the approximated integer bounds with fewer subproblems than in the case where no variable bounds are present. If no feasible solution exists, then $\omega_1 s_1 + \omega_2 s_2 + \ldots + \omega_n s_n \geq \Delta Y_0$ or Equation (37):

$$Y_0 \leq R - \Delta Y_0 \tag{37}$$

Equation (37) depicts a strong bound for the linear integer model. This is then used as an upper bound for the branch and bound, branch and cut, branch and price, branch cut and price, and branch cut and free algorithms, or any selected suitable procedure. If a feasible solution exists, then the upper bound is optimal.

Justification

The whole area

$$R - \Delta Y_0 \leq Y_0 \leq R \tag{38}$$

is searched for integer points, and the feasible optimal integer solution for this small region is the global optimal one.

7.1. Procedure for Finding a Strong Bound

Step 1: Relax and solve the model so that a continuous optimal solution can be obtained.

Step 2: Express the basic variables in terms of their approximated integer variable bounds.

Step 3: Solve the LIP model now with the approximated variable bounds. When branching, start with the most restricted variables.

Step 4: If the solution is feasible, then it is optimal; otherwise, an upper bound is obtained as $Y_0 \leq R - \Delta Y_0$ and is used in the selected approach for the LIP model.

7.2. Numerical Illustration

Consider the three-variable linear integer model provided in Equation (39) that is used to prove the validity and applicability of the variable bounding approach. Without the introduction of the variable bounds, the branch and bound method requires 9123 iterations to solve the problem, and only 3 iterations are required when variable bounds are computed first. This numerical illustration clearly shows the importance and advantages of approximating the integer variable bounds.

$$\begin{gathered} \text{Maximize} Y_0 = 17y_1 + 19y_2 + 22y_3, \\ \text{Such that} \\ 9y_1 + 15y_2 - 15y_3 \leq 87, \\ 10y_1 - 12y_2 + 6y_3 \leq 45, \\ -7y_1 + 13y_2 + 15y_3 \leq 99,008. \end{gathered} \tag{39}$$

where $y_1, y_2, y_3 \geq 0$ and integer.

The continuous optimal table is provided in Table 3.

Table 3. Continuous optimal table.

	y_1	y_2	y_3	s_1	s_2	s_3	**R.H.S**
Y_o	0	0	0	1.1918	1.9440	1.8809	186,410.8961
y_1	1	0	0	0.0438	0.0713	0.0153	1519.3595
y_3	0	0	1	−0.0078	0.0377	0.0438	4336.3971
y_2	0	1	0	0.0326	−0.0051	0.0346	3430.5815

Whereby the variables are expressed in terms of ΔY_0. From the continuous optimal table provided in Table 3, we have Equation (40):

$$1.1918y_1 + 1.944y_2 + 1.8809y_3 \geq \Delta Y_0 \tag{40}$$

Because $n = 3$, the possible solutions in terms of ΔY_0 are provided as follows:

When variable s_1 enters the basis:

$$y_1 = 1519.3595 - 0.0368\Delta Y_0,$$

$$y_2 = 3430.5815 - 0.0274\Delta Y_0,$$

$$y_3 = 4336.3971 - 0.0065\Delta Y_0.$$

When variable s_2 enters the basis:

$$y_1 = 1519.3595 - 0.0367\Delta Y_0,$$

$$y_2 = 3430.5815 - 0.0026\Delta Y_0,$$

$$y_3 = 4336.3971 - 0.0194\Delta Y_0.$$

When variable s_3 enters the basis:

$$y_1 = 1519.3595 - 0.0081\Delta Y_0,$$

$$y_2 = 3430.5815 - 0.0184\Delta Y_0,$$

$$y_3 = 4336.3971 - 0.0233\Delta Y_0.$$

By arranging the variables in the order of their descending restriction, the variable bounds become that provided in Equation (41):

$$\begin{array}{c} 1519.3595 - 0.0368\Delta Y_0 \le y_1 \le 1519.3595 - 0.0081\Delta Y_0 \\ 3430.5815 - 0.0274\Delta Y_0 \le y_2 \le 3430.5815 - 0.0026\Delta Y_0 \\ 4336.3971 - 0.0233\Delta Y_0 \le y_3 \le 4336.3971 - 0.0065\Delta Y_0 \end{array} \tag{41}$$

The variable ranges can be calculated as provided in Equation (42):

$$\begin{array}{c} \text{Variable } y_1, \tau_1 = -0.0081\Delta Y_0 + 0.0368\Delta Y_0 = 0.0287\Delta Y_0, \\ \text{Variable } y_2, \tau_2 = 0.0026\Delta Y_0 + 0.0274\Delta Y_0 = 0.03\Delta Y_0, \\ \text{Variable } y_3, \tau_3 = 0.0065\Delta Y_0 + 0.0233\Delta Y_0 = 0.0298\Delta Y_0. \end{array} \tag{42}$$

By arranging the variables in their approximated descending order of restriction, we have the following:

Descending order $\xrightarrow{y_1 y_3 y_2}$

Calculating the basic variable limits

The most restricted basic variable is y_1; thus,

$$1519.3595 - 0.0368\Delta Y_0 \le y_1 \le 1519.3595 - 0.0081\Delta Y_0 \tag{43}$$

Because $\Delta Y_0 = \text{Min}\left[\frac{1+f[\beta_i]}{\ell_i^U} \text{ or } \frac{2-f[\beta_i]}{\ell_i^U}, \frac{1+f[\beta_i]}{\ell_i^L} \text{ or } \frac{2-f[\beta_i]}{\ell_i^L}\right]$, the smallest ΔY_0 (i.e., ΔY_1) is calculated by using the equation provided in Equation (44):

$$\Delta Y_0 = \text{Min}\left[\frac{1.3595}{0.0368} = 36.9428, \frac{1.3595}{0.0081} = 167.8395\right] = 36.9428 \tag{44}$$

By replacing ΔY_0 with ΔY_1, we have Equation (45):

$$\begin{array}{c} 1518.00 \le y_1 \le 1519.06 \\ 3429.57 \le y_2 \le 3430.68 \\ 4335.53 \le y_3 \le 4336.63 \end{array} \tag{45}$$

After adjusting to the integer bounds, we have Equation (46):

$$\begin{array}{c} 1518 \le y_1 \le 1519 \\ 3430 \le y_2 \le 3430 \\ 4336 \le y_3 \le 4336 \end{array} \tag{46}$$

The upper bound is provided by $Y_0 \leq 186,410.8961 - 36.9428$; thus,

$$Y_0 \leq 186,373.9533 \tag{47}$$

Solving the LIP model with the approximated integer bounds

By selecting an automated branch and bound algorithm and solving the LIP model with the approximated integer bounds, a feasible solution is obtained and verified in only three iterations. The optimal solution obtained is

$$\begin{aligned} Y_0 &= 186,368, \\ y_1 &= 1518, \\ y_2 &= 3430, \\ y_3 &= 4336 \end{aligned} \tag{48}$$

Without the bounds, the branch and bound method (automated version) requires 9123 iterations to obtain the same optimal solution provided in Equation (48).

8. Conclusions

We solved linear integer models by using the variable bounding method. Finding a bound is important and is supposed to be the first attack method that is required for any linear integer programming problem. The computational complexity and many iterations that are required to search the NP Complete LIP problems may be significantly reduced by using bounds. One can independently generate *n* possible solutions, which are used to calculate limits. These independent calculations allow one to use parallel processors [18]. Using parallel processing alongside the available approaches for linear integer programming has been unsuccessful. A stronger bound can be found by increasing the number of integers accommodated within the limits of the most restricted variable. However, there should be a balance between the strength of the bound and the resulting number of iterations required to generate it.

Author Contributions: Methodology, E.M., J.C. and T.T.; writing—original draft preparation, E.M.; writing—review and editing, J.C. and T.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding the North West University, South Africa.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this research are contained in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, S.; Munapo, E.; Nyamugure, P. An Insight into the Characteristic Equation for an Integer Program. *Int. J. Math. Eng. Manag. Sci.* **2021**, *6*, 611–620. [CrossRef]
2. Munapo, E.; Kumar, S. Linear Integer Programming Theory and Applications: Some Recent Innovative Developments. De Gruyter. Available online: https://play.google.com/store/books/details?id=Vw5QEAAAQBAJ&gl=us (accessed on 6 December 2021).
3. Schrijver, A. *Theory of Linear and Integer Programming*; John Wiley: New York, NY, USA, 1986.
4. Alrabeeah, M.; Kumar, S.; Al-Hasani, A.; Munapo, E.; Eberhard, A. Computational Enhancement in the Application of the Branch and Bound Method for Linear Integer Programs and Related Models. *Int. J. Math. Eng. Manag. Sci.* **2019**, *4*, 1140–1153. [CrossRef]
5. Hillier, F.S. *Introduction to Operations Research*, 8th ed.; McGraw-Hill: New York, NY, USA, 2005.
6. Salvelsbergh, M.W.P. A branch and price algorithm to solve the generalized assignment problem. *Oper. Res.* **1997**, *45*, 381–841.
7. Mitchell, J.E. Branch and cut algorithms for integer programming. In *Encyclopedia of Optimization*; Floudas, C.A., Pardalos, P.M., Eds.; Kluwer Academic Publisher: Amsterdam, The Netherlands, 2001.
8. Fukasawa, R.; Longo, H.; Lysgaard, J.; Poggi de Aragao, M.; Uchoa, E.; Werneck, R.F. Robust branch-and-cut-price for the Capacitated vehicle routing problem. *Math. Program. Ser. A* **2006**, *106*, 491–511. [CrossRef]

9. Munapo, E. Branch Cut and Free Algorithm for the General Linear Integer Problem. In *Intelligent Computing and Optimization, Proceedings of the 3rd International Conference on Intelligent Computing and Optimization 2020 (ICO 2020), Hua Hin, Thailand, 8–9 October 2020*; Advances in Intelligent Systems and Computing; Vasant, P., Zelinka, I., Weber, G.W., Eds.; Springer: Cham, Switzerland, 2021; Volume 1324, p. 1324. [CrossRef]
10. Munapo, E.; Kumar, S. Reducing the complexity of the knapsack linear integer problem by reformulation techniques. *Int. J. Syst. Assur. Eng. Manag.* **2021**, *12*, 1087–1093. [CrossRef]
11. Gonzalez, M.; López-Espín, J.J.; Aparicio, J.; Talbi, E. A hyper-matheuristic approach for solving mixed integer linear optimization models in the context of data envelopment analysis. *PeerJ Comput. Sci.* **2022**, *8*, e828. [CrossRef] [PubMed]
12. Cereser, B.L.H.; Oliveira, A.R.L.D.; Moretti, A.C. A new strategy to solve linear integer problems with simplex directions. *Pesqui. Oper.* **2022**, *42*, 1–19. [CrossRef]
13. Zhou, J.; Chen, Y.; Peng, J.; Zhou, L.; Liang, G. A Mixed-Integer Linear Programming Model for the Station Capacity Allocation Problem of a Star-Tree Pipe Network. *J. Pipeline Syst. Eng. Pract.* **2023**, *14*, 04022053. [CrossRef]
14. Dilek, S.; Tosun, S. Integer linear programming-based optimization methodology for reliability and energy-aware high-level synthesis. *Microelectron. Reliab.* **2022**, *139*, 114849. [CrossRef]
15. Cremoncini, D.; Frate, G.F.; Bischi, A.; Ferrari, L. Mixed Integer Linear Program model for optimized scheduling of a vanadium redox flow battery with variable efficiencies, capacity fade, and electrolyte maintenance. *J. Energy Storage* **2023**, *59*, 106500. [CrossRef]
16. Munapo, E. Improvement of the Branch and Bound Algorithm for Solving the Knapsack Linear Integer Problem. *East.-Eur. J. Enterp. Technol.* **2020**, *2*, 104. [CrossRef]
17. Kumar, S.; Munapo, E.; Jones, B.C. An integer equation controlled descending path to a protean pure integer program. *Indian J. Math.* **2007**, *49*, 211–237.
18. Padberg, M.; Rinaldi, G. A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.* **1991**, *33*, 60–100. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.