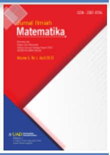
UAD Universitas Ahmad Dahlan

Jurnal Ilmiah Matematika
Vol. 11, No. 1, Mei 2024, pp. 6-18
ISSN 2774-3241
http://journal.uad.ac.id/index.php/Konvergensi



IMPLEMENTATION OF BRANCH AND BOUND ALGORITHM AND VARIABLE REDUCTION ALGORITHM IN PRODUCTION PROFIT OPTIMIZATION

Hanny Puspha Jayanti [a,1,*], Muchammad Abrori [b,2]

[*a] Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga, Indonesia
[b] Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga, Indonesia
[1] hanny.puspha27@gmail.com; [2] muchammad.abrori@uin-suka.ac.id
*Correspondent Author

Received: *Revised:* *Accepted:*

KATAKUNCI

Algoritma Branch and Bound
Integer Linear Programming
Algoritma Reduksi Variabel

ABSTRAK

Linear Programming (LP) tidak dapat menjawab permasalahan produksi yang mengharuskan variabel keputusan berbentuk bilangan bulat. Oleh karena itu, Integer Linear Programming (ILP) ada sebagai kasus khusus LP dimana variabel keputusannya adalah bilangan bulat. Penelitian ini bertujuan untuk mengetahui perbedaan nilai keluaran dan jumlah iterasi yang digunakan pada Algoritma Branch and Bound dan Variable Reduction untuk menyelesaikan masalah pemaksimalan keuntungan produksi. Algoritma Branch and Bound membagi masalah menjadi submasalah yang mengarah pada solusi dengan membentuk struktur pohon pencarian dan menerapkan batasan untuk mencapai solusi optimal. Sedangkan Algoritma Reduksi Variabel melibatkan pemindahan variabel keputusan dari sisi kiri ke sisi kanan fungsi kendala. Penelitian ini menggunakan data dari Industri Susu Rembang, dengan permasalahan ingin memaksimalkan keuntungan produksi. Dengan bantuan Maple, penyelesaian dengan Algoritma Branch and Bound dan Variable Reduction menghasilkan keuntungan yang sama yaitu Rp 14.786.548. Namun proses perhitungan menggunakan Algoritma Variable Reduction memerlukan iterasi yang lebih banyak dibandingkan dengan Algoritma Branch and Bound.

KEYWORDS

Branch and Bound Algorithm
Integer Linear Programming
Variable Reduction Algorithm

Implementation of Branch and Bound Algorithm and Variable Reduction Algorithm in Production Profit Optimization

Linear Programming (LP) cannot answer production problems that require decision variables to be integers. For this reason, Integer Linear Programming (ILP) exists as a special case of LP where the decision variables are integers. This research is intended to determine the difference in output values and the number of iterations used in the Branch and Bound Algorithm and Variable Reduction to solve the problem of maximizing production profits. The Branch and Bound algorithm divides the problem into sub-problems that lead to a solution by forming a search tree structure and applying restrictions to achieve an optimal solution. Meanwhile, the Variable Reduction Algorithm involves moving the decision variables from the left side to the right side of the constraint function. This study uses data from the Rembang Dairy Industry, with the problem of wanting to maximize production profits. Using Maple's assistance, the settlement using the Branch and Bound Algorithm and Variable Reduction yields the same profit, which

ISSN 2774-3241

is IDR 14,786,548. However, the calculation process using the Variable Reduction Algorithm requires more iterations than the Branch and Bound Algorithm.

Introduction

The rapid industrial growth causes operations research to be needed in the business world. This was because the problems that arose were basically the same even though the context was different from that of the military at that time. Entering the 1950s, operations research has developed rapidly in the world of business, government, and private institutions. During this period, linear and dynamic programming techniques were discovered and developed. Major developments have taken place in the study of queuing and inventory problems [1].

To meet the needs of life, humans often think of getting big profits even though the resources they have are limited. Consciously or not, humans often use economic principles. Therefore operations research exists to solve optimization problems related to these economic principles. The optimization problem that arises is to maximize production profits or minimize production costs with limited resources. To solve this problem, a method called Linear Programming was developed. Linear Programming is a mathematical method with a linear character to find an optimal solution by maximizing or minimizing the objective function with constraints as limits [2].

In some cases, especially those related to production results, the resulting decision variable must have an integer value. According to Mulyono [3] one of the assumptions of linear programming techniques is divisibility or fractionality. In other words, each model variable can occur in all non-negative values, meaning that the value of the solution obtained does not have to be an integer. In certain situations, this assumption is unrealistic and unacceptable. Many problems in industry and business require model variables to be integers. Therefore, integer programming is a linear programming with the additional requirement that all or some of the variables have non-negative integer values, but model parameters are not required to have integer values.

Based on the description above, this research is intended to solve the Integer Linear Programming problem by using the Branch and Bound Algorithm and Variable Reduction. According to Supatimah and friends in their journal entitled " Optimasi Keuntungan dengan Metode Branch and Bound", the Branch and Bound Algorithm is a method that can solve Integer Linear Peogramming problems, this method divides the problem into sub-problems (branching) that lead to to a solution by forming a search tree structure and bounding to reach the optimal solution [4]. Furthermore, P. Pandian and M. Jayalakshmi introduced a new method for solving Integer Linear Programming problems called the Variable Reduction Algorithm, this algorithm involves moving a decision variable to the other side of a constraint function [5].

Branch and Bound Algorithm and Variable Reduction Algorithm are appropriate methods in optimizing production to obtain maximum profit, with the resulting variables being integer values. In this study, for the same case, the results of the two methods will be compared to see which algorithm produces the fewest iterations with the most optimum results. The number of iterations produced determines the length of the Integer Linear Programming problem solving process to get the maximum benefit.

Methods

The methods used in this research are literature studies and applied research. Literature study is done by looking for references in the form of books or journals. Applied research is carried out by looking for cases in everyday life which are then solved using the Branch and Bound Algorithm and Variable Reduction. The problem in this study is limited by the problem of maximizing which will be discussed is optimizing the amount of production for better profits.

Linear Programming

The following is a general form of integer linear programming with the objective function to be maximized:

$$z = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \tag{1}$$

constraints

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \text{ , for } i = 1, 2, \ldots, \text{n} \tag{2}$$

$$x_j \ge 0$$

Simplex Method

The optimization solution in the real world does not only involve two variables. The more complex and advanced the times, the more complex the variables that play a role in achieving optimization. The use of Linear Programming which only uses two variables is not enough. For that we need a new tool known as the simplex method. This method can be used for optimization problems involving multivariate variables [6].

The steps in solving Linear Programming problems using the Simplex Method are as follows:

1. Converting Linear Programming problems into canonical form.
2. Arrange the canonical model equations into a simplex table.
3. Specifies the column and row keys.
4. Determines the key element, which is the intersection of the key column and key row.
5. Dividing the key row with key elements, in the key column the value other than the value in the key row becomes zero.
6. Changing the values outside the key row

$$\text{New row} = \text{old row} - \left(\frac{\text{the value of the key row}}{\text{key element}} \times \text{the value of the key column}\right)$$

7. Seeing the value of is optimal or not. If not then the iteration is continued. In maximizing, optimal results are obtained when or there is no positive value of $c_j - z_j$.

Integer Linear Programming

The following is a general form of integer linear programming with the objective function to be maximized:

$$z = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \quad (3)$$

constraints

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{, for } i = 1, 2, \ldots, \text{n} \quad (4)$$

$x_j \geq 0$, integer

Branch and Bound Algorithm

The Branch and Bound algorithm was first introduced in 1960 by A.H. Land and A.G. Dogg. This algorithm is a method that functions to find optimal solutions in cases of linear programming that produces decision variables in the form of integers. In line with its name, the Branch and Bound Algorithm limits the optimal solution by forming an upper and lower branch on each decision variable that has a fractional value to a whole value so that each restriction will form a new branch [2].

The steps in solving the Integer Linear Programming problem by utilizing the Branch and Bound Algorithm are:

1. Solve Integer Linear Programming problems using the simplex method (for cases with more than two variables) or the graphical method (for cases that only have 2 variables) as in solving Linear Programming problems.
2. Observing the optimum solution from step 1. If the resulting variable is integer, then the optimal solution has been reached. However, if the resulting variable is not an

integer value, then proceed to step 3.

3. Choose the variable with the difference in the largest fraction of the integers in each variable in step 2 to be branched into sub-problems. Form two new constraints of the variable with boundary batasan and .
4. Set the solution in step 1 as the upper limit and variable solutions that have been rounded up to be the lower limit.
5. Solve the sub-problem in the same way as in step 1 by adding the new constraints obtained from the previous step. If the solution has an integer value, then proceed to step 6. But if not, then return to step 3.
6. If one of the sub-problems has an integer value and the others have no resolution (infeasible), then the branching is stopped.
7. In choosing the optimum solution, if there are several sub-problems with integer-valued solutions, then for the objective function to maximize the value chosen is largest of Meanwhile, for the objective function to be minimized, the smallest value is chosen.

Variable Reduction Algorithm

The Variable Reduction Algorithm is based on a simple mathematical concept. Variable Reduction Algorithm can serve as an effective tool to find the optimum solution of the Integer Linear Programming problem. This algorithm is considered better than the Cutting Plane and Branch and Bound Algorithms in solving Integer Linear programming problems because in its application it is simpler and easier to understand [5].

The steps in solving the Integer Linear Programming problem by utilizing the Variable Reduction Algorithm for n variables are as follows:

1. Determine the minimum value of the largest integer value in each constraint function with variables , notated with .

$$x_j^* = \min\left\{\left[\frac{b_i}{a_{ij}}\right]\right\}, \text{ for } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n \quad (5)$$

With condition:

- If the results of the minimum value are the same, then one is chosen.
- If the minimum value results are different, then the smallest positive value is chosen.
- If there is a result that has a value of 0, then it does not need to be selected. Because 0 is a neutral value.

Then, $x_r \in \{0, 1, 2, \ldots, U_r\}$, for U_r is x_j^* means the largest integer value of x_r.

2. For each $x_r \in \{0, 1, 2, \ldots, U_r\}$ involve $n-1$ on variables x_j with $j \neq r$. So that the

Integer Linear Programming problem becomes:

$$z = \sum_{\substack{j=1 \\ j \neq r}}^{n} c_j x_j \quad (6)$$

constraints $\sum_{\substack{j=1 \\ j \neq r}}^{n} a_{ij} x_j \leq b_i - a_r x_r$ (7)

$x_j \geq 0$, dengan i = 1, 2,..., m dan j = 1, 2, ..., n

$j \neq r \in$ integer

3. Determine the minimum value

$$x_j^* = \min \left\{ \left[\frac{b_i - a_r x_r}{a_{ij}} \right] \right\}, \text{ with } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n \quad (8)$$

Then substitute $x_r = x_j \in \{0, 1, 2, \ldots, U_r\}$ into the equation 8

4. Repeating steps 2 and 3, so that an Integer Linear Programming equation is formed:

$$z = \sum_{\substack{j=1 \\ j \neq r}}^{n} c_j x_j \quad (9)$$

constraints $\sum_{\substack{j=1 \\ j \neq r}}^{n} a_{ij} x_j \leq b_i - a_r x_r - \ldots - a_k x_k$ (10)

$x_j \geq 0$, with i = 1, 2, ..., m and j = 1, 2, ..., n

$j \neq r, \ldots, k \in$ integer

5. Determine the optimal solution candidate. For the maximizing case, is chosen with the largest positive value. As for the minimizing case, is chosen with the smallest positive value.

Results and Discussion

This research uses data that has been presented in the journal owned by Laras Vegi Nuranggraini, Agustina Pradjaningsih, and Abduh Riski entitled "Optimasi Produksi Susu Dengan Algoritma Affine Scaling (Studi Kasus Pada Industri Susu Rembangan Jember)". The journal uses data from Rembang Dairy Industry in Jember, East Java. From observations obtained in the form of data on product types, the amount of raw materials and supporting materials used, the purchase price of raw materials and supporting materials, as well as the selling price of products and production time which will be carried out in January 2021. The data is used as constraints and profit data as the objective function in maximizing profits [7].

Table 1 raw material constraints (cow's milk)

Types of products	Raw Material(ml)
Original Milk	250
Tiramisu Milk	218
Red Velvet Milk	225
Caramel Coffee Milk	227
Durian Milk	230
Vanilla Milk Late	227
Inventory of Fresh Milk Raw Materials	**1,800,000**

Table 2 Production cost constraints (in rupiah)

Types of products	Raw material	Food Additives	Packaging	Etc	Production
Original Milk	2,000	0	1,350	2,000	5,350
Tiramisu Milk	1,740	1,344	1,350	2,000	6,343
Red Velvet Milk	1,800	1,050	1,350	2,000	6,200
Caramel Coffee Milk	1,816	966	1,350	2,000	6,132
Durian Milk	1,840	840	1,350	2,000	6,030
Vanilla Milk Late	1,816	966	1,350	2,000	6,132
Production Cost Inventory					**40,000,000**

Table 3 Production time constraints (in minutes)

Types of products	Boiling	Added Flavor	Cooling	Packaging	Production
Original Milk	0.25	0	0.5	1.5	2.25
Tiramisu Milk	0.25	0.2	0.5	1.5	2.45
Red Velvet Milk	0.25	0.15	0.5	1.5	2.4
Caramel Coffee Milk	0.25	0.2	0.5	1.5	2.45
Durian Milk	0.25	0.125	0.5	1.5	2.375
Vanilla Milk Late	0.25	0.125	0.5	1.5	2.375
Working Hours (minutes)					**14,400**

Table 4 Constraints of supporting materials (in ml)

Types of products	Flavor Ingredients	One Month Needs
Original Milk	0	0
Tiramisu Milk	32	24,000
Red Velvet Milk	25	24,000
Caramel Coffee Milk	23	24,000
Durian Milk	20	24,000
Vanilla Milk Late	23	24,000

 ISSN 2774-3241

Table 5 Selling price and production costs (in rupiah)

Types of products	Selling Price	Production Cost
Original Milk	7,500	5,350
Tiramisu Milk	9,000	6,343
Red Velvet Milk	9,000	6,200
Caramel Coffee Milk	8,000	6,132
Durian Milk	8,000	6,030
Vanilla Milk Late	8,000	6,132

Discussion*:*

For Variables for example:

z = The objective function sought (maximize)
x_1 = Original Milk
x_2 = Tiramisu Milk
x_3 = Red Velvet Milk
x_4 = Caramel Coffee Milk
x_5 = Durian Milk
x_6 = Vanilla Milk Late

Integer Linear Programming (ILP) in the case of Rembang milk production can be written as follows:

Objective function: Maximize

$$z = 2{,}150x_1 + 2{,}66x_2 + 2{,}800x_3 + 2{,}368x_4 + 2{,}470x_5 + 2{,}368x_6$$

constraint

$$250x_1 + 218x_2 + 225x_3 + 227x_4 + 230x_5 + 227x_6 \leq 1{,}800{,}000$$

$$5{,}350x_1 + 6{,}343x_2 + 6{,}200x_3 + 6{,}132x_4 + 6{,}030x_5 + 6{,}132x_6 \leq 40{,}000{,}000$$

$$2.25x_1 + 2.45x_2 + 2.4x_3 + 2.45x_4 + 2.375x_5 + 2.375x_6 \leq 14{,}400$$

$$32x_2 \leq 24{,}000$$

$$25x_3 \leq 24{,}000$$

$$23x_4 \leq 24{,}000$$

$$20x_5 \leq 24{,}000$$

$$23x_6 \leq 24{,}000$$

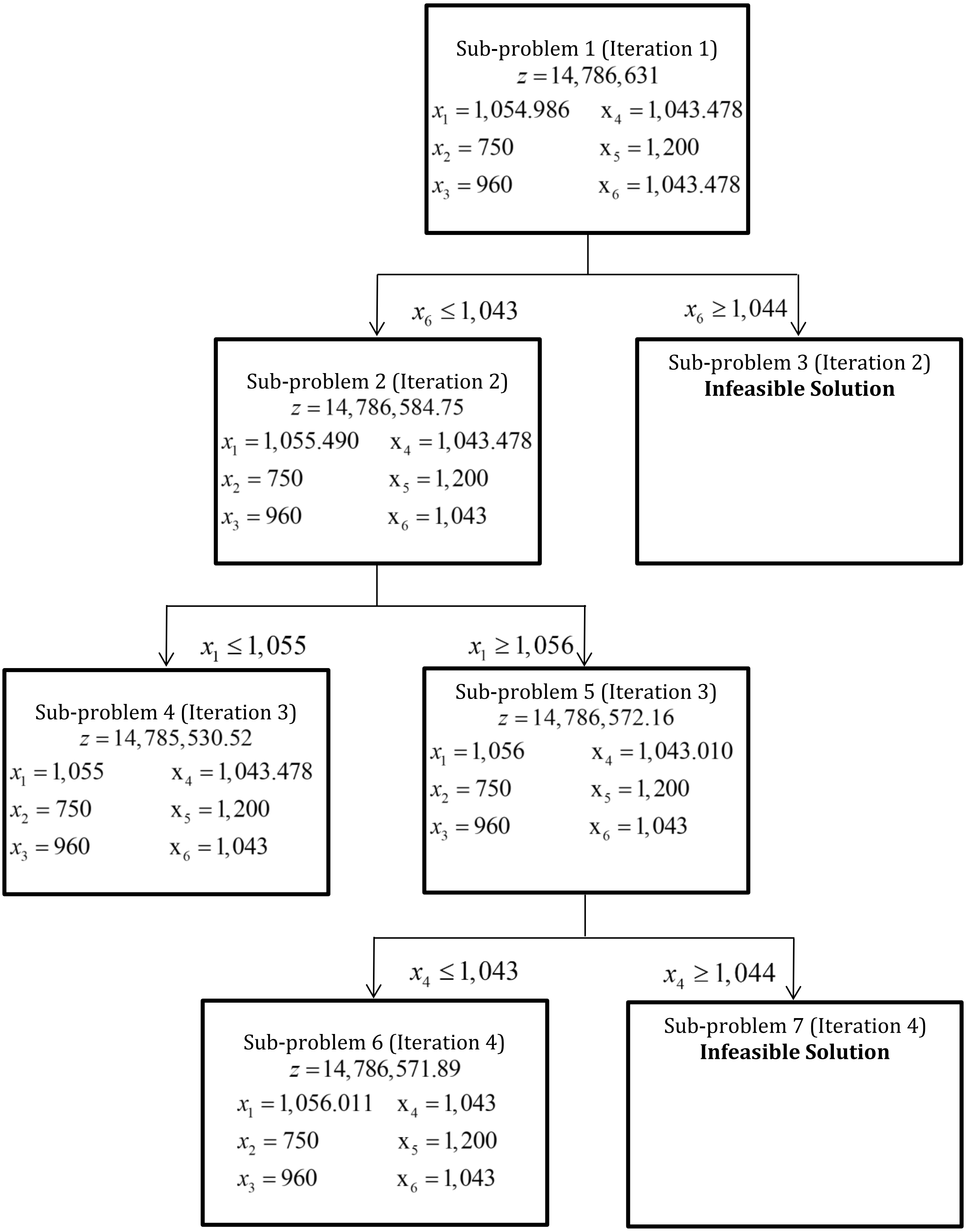
$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0, \ \textit{integer}$$

 ISSN 2774-3241

Completion with the Branch and Bound Algorithm

Based on the solution using the simplex method with the help of Maple Software, the optimal solution is obtained $x_1 = 1,054.986;$ $x_2 = 750;$ $x_3 = 960;$ $x_4 = 1,043.478;$ $x_5 = 1,200;$ and $x_6 = 1,043.478$. Because there are x_1, x_4, and x_6 which is not integer, then the variable with the difference in the largest fraction of the integers in each variable is selected, obtained x_1 have differences 0,014 and x_4 along x_6 have a difference of 0,478. Therefore one of them can be selected x_4 and x_6. Selected $x_6 = 1,043.478$ as a branching variable in iteration 2, namely sub-problem 2 by adding constraints $x_6 \le 1.043$ and sub-problem 3 by adding constraints $x_6 \ge 1.044$. Then, solved using the simplex method for each sub-problem. This branching process is repeated until all the decision variables obtained have integer values. After doing 5 branches with 9 sub-problems, the optimal solution is obtained as follows:

ISSN 2774-3241



ISSN 2774-3241

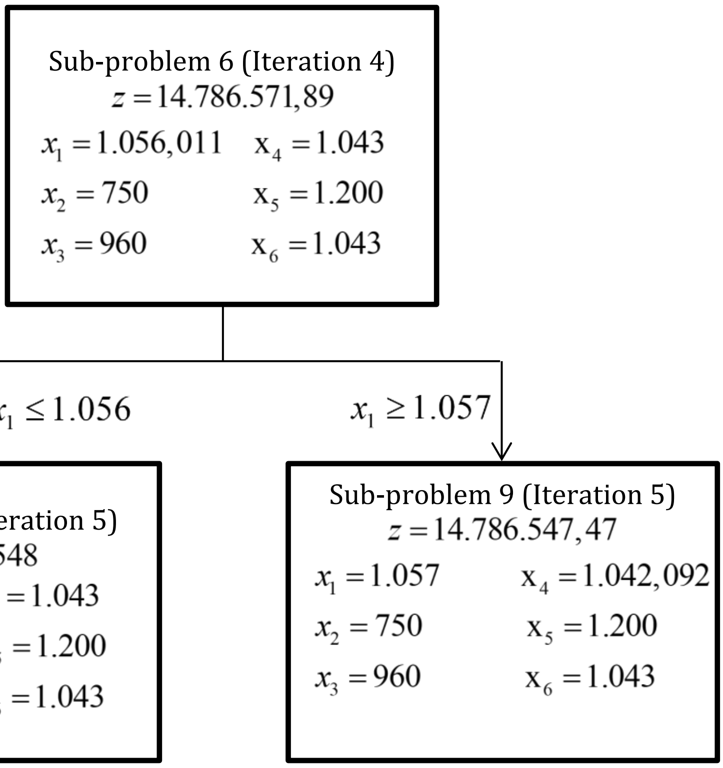


Figure 1 Branching iterations 1 – 5

Because there is already a sub-problem where all the variables are positive integer with the largest value z, so that the branching process is stopped. Then, the optimal solution is obtained in iteration 5 (sub-problem 8) with the optimal solution $x_1 = 1,056$; $x_2 = 750$; $x_3 = 960$; $x_4 = 1,043$; $x_5 = 1,200$; $x_6 = 1,043$; with $z = 14,786,548$.

Completion with Variable Reduction Algorithm

Calculations with the help of Maple Software in iteration 1, by determining the minimum value of the integer values for each variable, the minimum value of $x_1 = 6,400$; $x_2 = 750$; $x_3 = 960$; $x_4 = 1,043$; $x_5 = 1,200$; and $x_6 = 1,043$. Therefore, minimum value of selected integer values is $x_2 = 750$. Then, $x_r = x_2 \in \{0,1,\ldots,750\}$, Which 750 is the largest integer value of x_2.

Next, in iteration 2, for each $x_r = x_2 \in \{0,1,\ldots,750\}$ implicate $6-1$ on variables x_j with $j \neq r$. Because the optimal solution is obtained when exactly one constraint variable is on the left, then the iteration is continued. Specifies the minimum value by substituting $x_r = x_2 \in \{0,1,\ldots,750\}$. Because if $x_2 \in \{0,1,\ldots,749\}$ substituted into the equation in the previous step will produce a minimum value that is greater than the result of the substitution $x_2 = 750$, better continue computation simply by substituting $x_2 = 750$. With notes, substitution results $x_2 = 750 \neq 0$ and $x_2 = 750$ positive round value. If the value is 0 or negative, so the replaced value is the value before 750, and continues until it gets the minimum value that is not 0 and has a positive integer value. This applies to all iterations. After doing 6 iterations, the optimal solution obtained is defined as:

Table 6 Iterations 1 – 6 of the Variable Reduction Algorithm

Iteration	Find the minimum value	The decision variable obtained	Selected variable
1	$x_j^* = \min\left\{\left[\frac{b_i}{a_{ij}}\right]\right\}$	$x_1 = 6,400$ $x_4 = 1,043$ $x_2 = 750$ $x_5 = 1,200$ $x_3 = 960$ $x_6 = 1,043$	$x_2 = 750$
2	$x_j^* = \min\left\{\left[\frac{b_i - x_2}{a_{ij}}\right]\right\}$	$x_1 = 5,583$ $x_5 = 1,200$ $x_3 = 960$ $x_6 = 1,043$ $x_4 = 1,043$	$x_3 = 960$
3	$x_j^* = \min\left\{\left[\frac{b_i - x_2 - x_3}{a_{ij}}\right]\right\}$	$x_1 = 4,559$ $x_6 = 1,043$ $x_4 = 1,043$ $x_5 = 1,200$	$x_4 = 1,043$
4	$x_j^* = \min\left\{\left[\frac{b_i - x_2 - x_3 - x_4}{a_{ij}}\right]\right\}$	$x_1 = 3,424$ $x_5 = 1,200$ $x_6 = 1,043$	$x_6 = 1,043$
5	$x_j^* = \min\left\{\left[\frac{b_i - x_2 - x_3 - x_4 - x_6}{a_{ij}}\right]\right\}$	$x_1 = 2,323$ $x_5 = 1,200$	$x_5 = 1,200$
6	$x_j^* = \min\left\{\left[\frac{b_i - x_2 - x_3 - x_4 - x_6 - x_5}{a_{ij}}\right]\right\}$	$x_1 = 1,056$	$x_1 = 1,056$

Because in iteration 6 there is only 1 variable left on the left, then the iteration is stopped. Therefore, based on the results obtained in iterations 1 – 6, $x_2 = 750$; $x_3 = 960$; $x_4 = 1.043$; $x_6 = 1,043$; $x_5 = 1,200$; and $x_1 = 1,056$; then the resulting optimal solution is

$$z = 2,150x_1 + 2,566x_2 + 2,800x_3 + 2,368x_4 + 2,470x_5 + 2,368x_6$$
$$= 2,150(1,056) + 2,566(750) + 2,800(960) + 2,368(1,043) +$$
$$2,470(1,200) + 2,368(1,043)$$
$$= 14,786,548$$

Conclusions

The Branch and Bound algorithm requires fewer iterations. Meanwhile, solving using the Variable Reduction Algorithm requires more iterations. However, both methods produce the same optimum value, that is $x_1 = 1,056$; $x_2 = 750$; $x_3 = 960$; $x_4 = 1,043$; $x_5 = 1,200$; and $x_6 = 1,043$ with $z = 14,786,548$. Then, if done manually, The Branch and Bound algorithm takes longer. This is because that algorithm in each iteration is divided into two sub-problems and requires solving using the simplex method, which the simplex method also requires iteration in its completion.

Daftar Pustaka

[1] Meflinda, Astuti, and Mahyarni, *Operations Research (Riset Operasi)*. Riau: UNRI PRESS, 2011.

[2] Siswanto, *Operation Research, Jilid 1*. Jakarta: Erlangga, 2007.

[3] S. Mulyono, *Operations Research*. Jakarta: Fakultas Ekonomi Universitas Indonesia, 1991.

[4] S. S. Supatimah, Farida, and S. Andriani, "Optimasi Keuntungan dengan Metode Branch and Bound," *AKSIOMA J. Mat. dan Pendidik. Mat.*, vol. 10, no. 1, pp. 13–23, 2019. Available: http://surl.li/fgxeg

[5] P. Pandian and M. Jayalakshmi, "A New Approach For Solving A Class Of Pure Integer Linear Programming Problems," *Int. J. Adv. Eng. Technol.*, vol. III, no. I, pp. 248–251, 2012. Available: http://surl.li/fgxdt

[6] V. R. Parmono, R. Kristiawan, and H. A. Hutahaean, *Riset Operasi*, Pertama. Jakarta: Universitas Terbuka, 2007.

[7] L. V. Nuranggraini, A. Pradjaningsih, and A. Riski, "Optimasi Produksi Susu Dengan Algoritma Affine Scaling (Studi Kasus Pada Industri Susu Rembangan Jember)," *Pros. Semin. Nas. Integr. Mat. dan Nilai Islam.*, vol. 4, no. 1, pp. 13–18, 2021. Available: http://surl.li/fgxcw