OPTIMIZATION ALGORITHM STUDY : MIXED INTEGER LINEAR PROGRAMMING

Tittawat Fongchantuk

A Thesis Submitted in Partial Fulfilment of the Requirements
for the Degree of Master of Science
The Petroleum and Petrochemical College, Chulalongkorn University
in Academic Partnership with
The University of Michigan, The University of Oklahoma,
Case Western Reserve University, and Institut Français du Pétrole
2018

Thesis Title: Optimization Algorithm Study : Mixed Integer Linear Programming

By: Tittawat Fongchantuk

Program: Petrochemical Technology

Thesis Advisor: Asst. Prof. Kitipat Siemanond

Accepted by The Petroleum and Petrochemical College, Chulalongkorn University, in partial fulfilment of the requirements for the Degree of Master of Science.

.... College Dean

(Prof. Suwabun Chirachanchai)

Thesis Committee:

....

(Asst. Prof. Kitipat Siemanond)

....

(Prof. Thirasak Rirksomboon)

....

(Assoc. Prof. Vissanu Meeyoo)

ABSTRACT

5971015063: Petrochemical Technology Program
Tittawat Fongchantuk: Optimization Algorithm Study: Mixed Integer Linear Programming.
Thesis Advisor: Asst. Prof. Kitipat Siemanond 216 pp.
Keywords: Optimization solver/ Supply chain/ Simplex/ Branch-and-Bound/ MILP

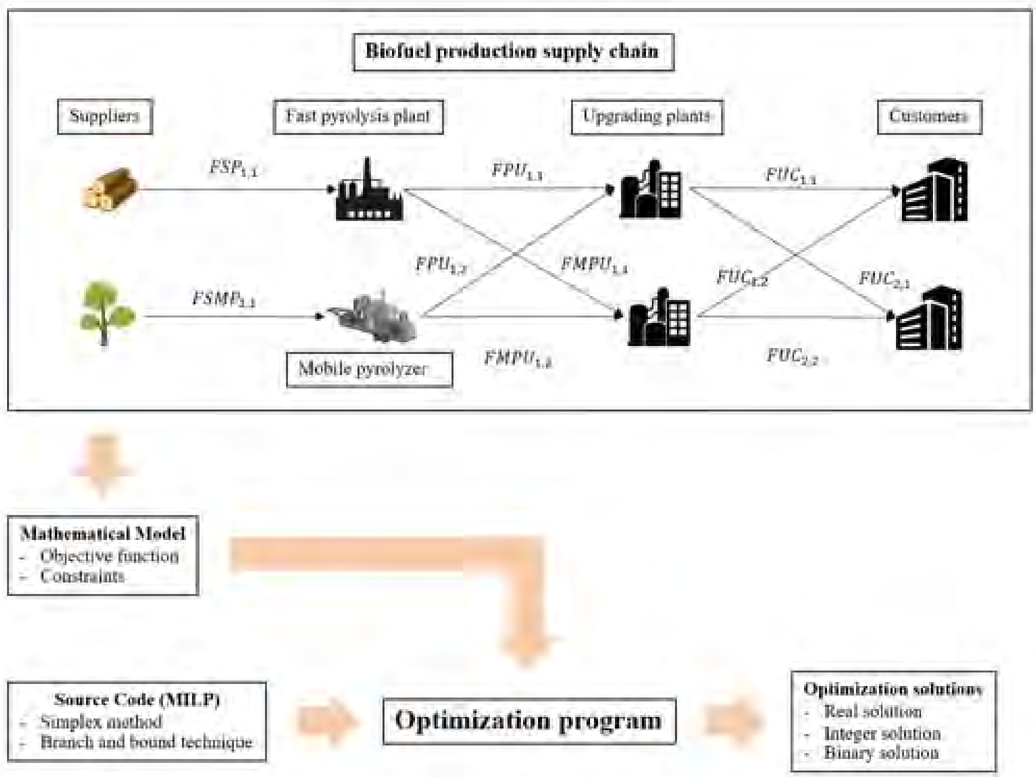
In the present, many petrochemical and petroleum products (engine fuel, solvent, plastic and synthetic rubber. etc.) are important. They are transported by ship, pipe line or train every day. So, minimizing transportation cost and time or maximizing profit is an important key of every company. In this optimization process, the optimal supply chain must be well developed in mathematical programming with the objective function of minimum cost or maximum profit and be solved by using optimization solvers. Objective functions and their constraints are solved by suitable types of optimization solvers: Linear programming (LP) and mixed integer linear programming (MILP). MILP is one of the most widely used optimization technique for designing supply chain, and developed by using linear programming and branch-and-bound technique. In this research, MILP algorithm is studied and developed on FORTRAN 4.0 to study MILP procedure, and then it is applied with case study of biofuel production supply chain. Finally, the results are validated with solver in Microsoft Excel. The result shows that the MILP algorithm can show many solutions with one optimum point for multiple optimum problems for benefit of alternative solutions, while only one solution is obtained from solver in Microsoft Excel. Moreover, the optimal SCNPV from the algorithm is close to one from Microsoft Excel. It shows that the MILP algorithm is high efficient for accuracy solution and benefit for alternative of suitable solutions in optimizing supply chain problems.

บทคัดย่อ

ฐิตวัฒน์ ฟองจันทึก : การศึกษาอัลกอริทึมของ Optimization: Mixed Integer linear Programming (Optimization Algorithm Study: Mixed Integer Linear Programming) อ. ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ กิติพัฒน์ สีมานนท์ 216 หน้า

ในปัจจุบันเคมีภัณฑ์และผลิตภัณฑ์จากปิโตรเลียมและปิโตรเคมีต่างๆ เช่น เชื้อเพลิงเครื่องยนต์ สารทำละลาย พลาสติก และยางสังเคราะห์ เป็นต้น มีความสำคัญอย่างมาก ซึ่งมีการขนส่งอย่างต่อเนื่องเป็นประจำทุกวันโดยทางเรือ ท่อขนส่ง และรถไฟ เป็นต้น ด้วยเหตุนี้การลดต้นทุนและเวลาในการขนส่งผลิตภัณฑ์ต่างๆ ให้น้อยที่สุด และการเพิ่มกำไรให้มากที่สุดจึงเป็นปัจจัยสำคัญปัจจัยหนึ่งในอุตสาหกรรม ในกระบวนการการหาจุดที่ดีที่สุดนี้ (Optimization) ห่วงโซ่อุปทาน (Supply chain) จะถูกพัฒนาขึ้นให้อยู่ในรูปของสมการทางคณิตศาสตร์สำหรับการลดต้นทุนหรือเพิ่มกำไร จากนั้นจึงแก้สมการโดยใช้ Optimization solver ที่เหมาะสม เช่น Linear Programming (LP) และ Mixed Integer Linear Programing (MILP) เป็นต้น MILP เป็นหนึ่งใน solver ที่ถูกใช้อย่างแพร่หลายที่สุดสุดที่ใช้สำหรับแก้ปัญหาห่วงโซ่อุปทานซึ่งประกอบด้วย 2 เทคนิค ได้แก่ LP และ Branch-and-Bound ในงานวิจัยนี้อัลกอริทึมของ MILP ถูกพัฒนาขึ้นบนโปรแกรม Fortran 4.0 เพื่อศึกษากระบวนการทำงานของ MILP solver จากนั้นกรณีศึกษาของห่วงโซ่อุปทานในการผลิตเชื้อเพลิงชีวภาพจะถูกใช้เพื่อประเมินการทำงานของโปรแกรม ในที่สุดผลจากการทดลองจะถูกนำไปประเมินร่วมกับ solver ของ Microsoft excel จากผลการทดลองพบว่า โปรแกรมสามารถแสดงคำตอบที่มีค่า optimum เท่ากันได้หลายคำตอบสำหรับปัญหาที่เป็น multiple optimum ซึ่งเป็นประโยชน์อย่างมากสำหรับการเลือกใช้คำตอบที่เหมาะสมในขณะที่ solver ของ Microsoft excel แสดงได้เพียงคำตอบเดียวเท่านั้น และนอกจากนี้ค่า SCNPV จากโปรแกรมยังมีค่าใกล้เคียงกับค่า SCNPV จาก solver ของ Microsoft excel อีกด้วย ซึ่งถือได้ว่า อัลกอริทึมสามารถทำงานได้อย่างมีประสิทธิภาพแม่นยำ และมีประโยชน์ทางด้านให้หลายตัวเลือกที่เหมาะสมสำหรับปัญหาห่วงโซ่อุปทาน

GRAPHICAL ABSTRACT



ACKNOWLEDGMENTS

I would first like to thank my thesis advisor (Asst. Prof. Kitipat Siemanond) for every advice in my research. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. He gave me the right direction whenever he thought I needed it.

I would also like to thank to Prof. Thirasak Rirksonboon and Assoc. Prof. Vissanu Meeyoo for being committees of my thesis. Every recommendation and comments are very useful for this research.

I am grateful for the partial scholarship and partial funding of the thesis work provided by the Petroleum and Petrochemical College.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

		PAGE
Title Page		i
Abstract (in English)		iii
Abstract (in Thai)		iv
Acknowledgements		vi
Table of Contents		vii
List of Tables		ix
List of Figures		xi
CHAPTER		
I	**INTRODUCTION**	1
II	**THEORETICAL BACKGROUD**	4
	2.1 Linear Programming (LP)	4
	2.1.1 Problem That Has ≤ Form In All Constraints	5
	2.1.2 Problem That Has = Or ≥ In Their Constraints	7
	2.2 Branch-And-Bound	11
	2.2.1 Branch-And-Bound For Integer And Mixed-Integer Linear Solution	11
	2.2.2 Branch-And-Bound For Binary Solution (0 And 1)	16
III	**EXPERIMENTAL**	27
	3.1 Developing LP Algorithm	27
	3.1.1 Simplex Algorithm	27
	3.1.2 Simplex Algorithm Evaluation	29
	3.2 Developing MILP Algorithm	30
	3.2.1 Branch-And-Bound Algorithm	30
	3.2.2 Branch-And-Bound Algorithm Evaluation	32
	3.3 Biofuel Production Supply Chain Case Study	34

CHAPTER **PAGE**

IV RESULTS AND DISCUSSIONS 36

4.1 Simplex Algorithm Evaluation 36

4.1.1 Results From The Developed Simplex Algorithm 40

4.1.2 The Developed Simplex Algorithm Evaluation 41

4.2 Branch-And-Bound Algorithm Evaluation 51

4.2.1 BIP Algorithm Evaluation 51

4.2.1.1 Results Of Small Supply Chain With BIP 54

4.2.1.2 Other Solutions Of Small Supply Chain With BIP 55

4.2.2 MILP Algorithm Evaluation 59

4.2.2.1 Result From MILP Algorithm Evaluation 60

4.2.2.2 MILP Algorithm Evaluation 61

4.3 Biofuel Production Supply Chain Case Study 64

4.3.1 MILP Evaluation For Biofuel Production Supply Chain 80

4.3.1.1 Result Of MILP Evaluation For Biofuel Production 80

4.3.1.2 The MILP Evaluation For Biofuel Production 82

4.3.2 BIP Evaluation For Biofuel Production Supply Chain 85

4.3.2.1 Result Of BIP Evaluation For Biofuel Production 85

4.3.2.2 BIP Evaluation For Biofuel Production 87

4.3.3 Full MILP Evaluation For Biofuel Production Supply Chain 91

4.3.3.1 Result Of Full MILP Evaluation For Biofuel Production 91

4.3.3.2 Full MILP Evaluation For Biofuel Production 93

V CONCLUSION 97

CHAPTER **PAGE**

REFFERENCES 98

APPENDICES 100

Appendix A The Developed Algorithm Evaluation 100

Appendix B Degenerate Problem Description 122

Appendix C Simplex Evaluation For Small Supply Chain 124

Appendix D Branch-And-Bound Evaluation 126

Appendix E Nomenclature For The Variables In BioFuel Production Supply Chain Case Study 128

Appendix F Result Of Biofuel Production Supply Chain 131

Appendix G Additional Complex Supply Chain Case 141

CIRRICULUM VITAE 216

LIST OF TABLES

TABLE		PAGE
3.1	The project problem data	33
4.1	Results for small supply chain problem from developed simplex Algorithm	40
4.2	Results for small supply chain problem from non-basic variable	42
4.3	Results for small supply chain problem with x_{11} = 50	43
4.4	Results from the different entering variable	44
4.5	Results of small supply chain with x_{21}= 50	49
4.6	Results validated with the algorithm by fixing $x_{1,1}$ = 100	50
4.7	Results of small supply chain with BIP	54
4.8	Results of small supply chain with BIP and non-basic variable	56
4.9	Results of small supply chain with BIP and y_{22} = 250	57
4.10	Results of small supply chain with BIP and x_{12} = 200	58
4.11	Result of the project problem from MILP algorithm	60
4.12	Conversion factor of the reactions	66
4.13	The costs of feedstock	73
4.14	The fuel consumption parameters for transportation cost and truck	78
4.15	Biofuel production supply chain result from MILP evaluation	81
4.16	Biofuel production supply chain result from BIP evaluation	86
4.17	Other biofuel production supply chain result from BIP evaluation	88
4.18	Biofuel production supply chain result from full MILP evaluation	92
4.19	Other biofuel production result from full MILP evaluation	94
C1	Results for small supply chain problem from Microsoft excel	124
D1	Result of small supply chain from excel (validated with BIP)	126
D2	Result of the project problem from Microsoft excel	127
F1	Specification of x to supply chain variables	132
F2	Biofuel production result from excel (validated with MILP)	133
F3	Biofuel production result from excel (validated with BIP)	136

TABLE **PAGE**

F4 Biofuel production result from excel (validated with full MILP) 139

LIST OF FIGURES

FIGURE **PAGE**

2.1	Bounding and branching example for the two new subproblem.	13
2.2	Branch and bound technique for integer linear program.	15
2.3	Branch and bound technique for integer linear program.	17
2.4	The example for BIP (mixed binary real).	19
3.1	Flowchart of simplex algorithm for inequality (≤) constraints.	28
3.2	The small supply chain problem.	29
3.3	Algorithm for BIP and MILP.	31
3.4	The graphic model for biofuel production supply chain in case study.	34
3.5	Biofuel production case study.	35
4.1	Mass flow variables of the supply chain.	36
4.2	Result for small supply chain problem from developed simplex Algorithm.	40
4.3	Procedure to obtain new solution from non-basic variable.	42
4.4	Results for small supply chain problem from non-basic variable.	43
4.5	Results for small supply chain problem with x_{11} = 50.	44
4.6	Results from the different entering variable.	45
4.7	Flowchart for choosing the most negative value.	46
4.8	The code for checking the most negative value.	47
4.9	The code for fixing the first one of the most negative values.	47
4.10	The code for fixing the last one of the most negative values.	48
4.11	Results for small supply chain problem with x_{21} = 50.	49
4.12	Results validated with the algorithm by fixing $x_{1,1}$ = 100.	50
4.13	Results of small supply chain with BIP.	55
4.14	Results of small supply chain with BIP and non-basic variable.	56
4.15	Results of small supply chain with BIP and y_{22} = 250.	57
4.16	Results of small supply chain with BIP and x_{12} = 200.	58
4.17	Flowchart for fathoming step in branch-and-bound algorithm.	62

FIGURE **PAGE**

4.18 Branch-and-bound step of the developed branch-and-bound algorithm. 63

4.19 Subproblem from Microsoft excel in branch-and-bound evaluation. 63

4.20 The mass flow variables biofuel production supply chain. 64

4.21 Biofuel production supply chain mass flow rate from MILP Evaluation. 82

4.22 Subproblem from excel in MILP evaluation for biofuel production. 83

4.23 MILP and fathoming algorithm. 83

4.24 MILP procedure for biofuel production supply chain. 84

4.25 Biofuel production supply chain mass flow rate from BIP evaluation. 87

4.26 Other Biofuel production mass flow rate from BIP evaluation. 89

4.27 Subproblem from excel in BIP evaluation for biofuel production. 90

4.28 BIP procedure for biofuel production supply chain. 90

4.29 Biofuel production mass flow rate from full MILP evaluation. 93

4.30 Other Biofuel production mass flow rate from full MILP evaluation. 95

4.31 Subproblem from excel in full MILP evaluation for biofuel Production. 95

A1 Simplex method algorithm. 100

A2 The result of $\leq constraint$ problem from the algorithm. 108

A3 The algorithm for big-M method. 109

A4 The result of $= constraint$ problem from the algorithm. 110

A5 The result of $\geq constraint$ problem from the algorithm. 110

A6 The algorithm for branch-and-bound. 111

A7 The result of MILP problem from the algorithm. 120

A8 The result of pure binary problem from the algorithm. 120

A9 The result of BIP (mixed binary real) problem from the algorithm. 121

B1 Graphical solution of normal problem. 122

B2 Graphical solution of degenerate problem. 123

C1 Simplex evaluation for small supply chain. 124

C2 Results for small supply chain problem from Microsoft excel. 125

D1 BIP evaluation of small supply chain. 126

FIGURE **PAGE**

D2 Result of small supply chain from excel (validated with BIP). 127
D3 Result of the project problem from BIP algorithm. 127
F1 Result of biofuel production supply chain with MILP. 131
F2 Biofuel production result from excel (validated with MILP). 134
F3 Result of biofuel production supply chain with BIP. 135
F4 Biofuel production result from excel (validated with BIP). 137
F5 Result of biofuel production supply chain with full MILP. 138
F6 Biofuel production result from excel (validated with full MILP). 140
G1 Detail of the supply chain 5-6-2. 141
G2 Variables of mass flow in supply chain 5-6-2. 142
G3 All 'x' variables of mass flow in supply chain 5-6-2. 142
G4 Results of supply chain 5-6-2 by simplex algorithm (Case 1). 145
G5 Main source code of algorithm. 146
G6 Subroutine of inputting values. 155
G7 The value of problem 5-6-2 on notepad. 156
G8 Simplex subroutine. 161
G9 Results of supply chain 5-6-2 from the program. 169
G10 Subroutine source code for alternative solution. 170
G11 Other results of supply chain 5-6-2 by simplex algorithm (Case 2). 173
G12 Other results of supply chain 5-6-2 by simplex algorithm (Case 3). 174
G13 Detail of the supply chain 5-6-2 MILP. 175
G14 Binary variables in supply chain 5-6-2 175
G15 All 'x' variables of binary variables in supply chain 5-6-2. 176
G16 Mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 1). 180
G17 Main source code for MILP. 181
G18 Subroutine of inputting value (MILP case). 190
G19 The values of problem 5-6-2 MILP on notepad. 191
G20 Simplex subroutine (MILP case). 202
G21 Results of supply chain 5-6-2 with binary from the program (Case 1). 210

FIGURE **PAGE**

G22 Subroutine source code for alternative solution (MILP case). 212

G23 Other mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 2). 215

G24 Other mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 3). 215

CHAPTER I
INTRODUCTION

Normally in production processes and supply chains, many procedures such as transportation, warehouse, production process or demand market, would be managed efficiently (Sharifizadeh, 2015). However due to complexity with their networks and many routings, targets of the systems such as profit, cost and production time are difficult to be managed. Examples for production processes and supply chains are LNG supply chain, biofuel transportation and supply chain and supply chain network in toothbrush industry. Moreover, production processes contain many network processes such as network of heat exchangers, oil blending and water or substance controlling. These supply chains and production processes would be developed in mathematical model that involve many variables and constraints. However due to complexity of the constraints and variables, the problems are difficult to be solved without computer programming

Optimization solver is computer program to find optimum point of objective function with its constraints. Algorithm of the solver is designed to solve specific problems: Linear Programming (LP) to solve linear equation problems, Integer Programming (IP) to solve linear equation problems with integer solutions, Mixed Integer Linear Programming (MILP) to solve linear equation problems with mixed-integer real solutions and Mixed Integer Non-linear Programming (MINLP) to solve non-linear equation problems with mixed-integer real solutions (Edgar, 1989). To select types of optimization solver, many factors would be considered, such as type of the problems, number of variables and constraints or license price.

Before optimization solver is applied, mathematical models of the problems are formulated including with objective function and constraints. However, due to complexity and many numbers of variables and constraints, computational time and different solutions would be obtained from different solvers. Furthermore, many problems have often occurred in solving complex problems, such as a long computational time, program error and variable and constraint limitation. To study solver algorithm would help to understand solving algorithm. Therefore, the solver algorithm would be studied for understanding and reducing the problems.

However, only one optimum solution of the problem is obtained from commercial optimization solver. This makes users don’t have choice to choose suitable solution. Improving the mathematical model is only one way to obtain new suitable solution, which makes users too long and too complicated to obtain other solutions. So, obtaining multiple optimum solutions from multiple optimum problems with only one solving is much benefit for alternative solutions and to eliminate time improving mathematical model. In this research, the optimization solver is developed and studied to find the way of multiple optimum solutions also.

Moreover, for the programming most of mathematical models used in industry and supply chain are developed on mixed integer linear programming (MILP) form and mixed integer non-linear equation (MINLP) form because they consist decision variables and continuous variables (integer variables is substituted by decision variables and the other is substituted by continuous variables) (Hillier, 2010). So in this research, optimization solver algorithm is studied on mixed integer linear programming (MILP). The algorithm is developed with simplex method and branch-and-bound technique, which are used in commercial solvers such as OSL, CPLEX, LINDO and ZOOM. Consequently, MILP is the most interesting programming to study for the problems of supply chains, which consists of 2 methods: simplex method and branch-and-bound.

However, for programs which are used to develop optimization solvers, there are many languages such as C, C++, Java, Matlab and Fortran. In developing optimization solver, the programs used to develop must efficiently work on complex source code with many array variables and complex calculation, because optimization algorithm uses very complex calculation and many array variables which is difficult to check error or to develop and improve the algorithm. Moreover, the program must be used on every computer processor for that it can be used on both high and low efficient processor computers. Most of the programs such as Matlab have many functions to help work, but that makes them to be too much memory, which only highly efficient processor computers can be used. Moreover, many programs such as C and C++ can not be used for complex source codes and complex calculation. However, Fortran is a program that is developed for scientific work which can be efficiently used for complex source codes and complex calculation and can be used on both high and low efficient processor computers. So, Fortran is the most interesting for developing thee optimization solvers.

In this research, MILP is studied and developed on Fortran 4.0 to study the calculation procedures for supply chain problems to be benefit for understanding the optimization solver work and be advantage for using optimization solver efficiently for supply chain problems and improving optimization solver in the future. After developing MILP algorithm, biofuel production supply chain is used to evaluate the algorithm, and validated with solver of Microsoft Excel.

CHAPTER II
THEORETICAL BACKGROUD

MILP is used to solve linear problem with linear constraints to find mixed integer-continuous variables at optimum point. The best choice and the best value of the variables will be binary (0-1), where 0 is for unselected decision variables and 1 is for selected decision variables (Edgar, 1989).

To study MILP algorithm, 2 procedures: simplex method and branch-and-bound technique, are focused. Simplex method is used to solve linear programming problem, and branch-and-bound technique is used to solve IP and MILP problem.

2.1 Linear Programming (LP)

Linear programming is used to find the optimal point of linear objective function with linear constraints. Generally, there are many methods used for LP such as graphical method and simplex method, dependent on the problems (Edgar, 1989).

Problems that can be solved by simplex method must be in following form:

1. Objective function must be in linear equation

   Maximize/Minimize $f(x) = \sum_{i=1}^{r} c_1 x_i$

2. Constraints of problems must be in linear equation or non-equation

   Subject to $\sum_{i=1}^{r} a_{ji} x_i = b_j$ j = 1, 2, . . . , m

   $\sum_{i=1}^{r} a_{ji} x_i \geq b_j$ j = m + 1, . . . , p

3. All variables must be greater than or equal to 0

   Subject to $x_i \geq 0$ i = 1, 2, . . . , r

   (All r of the variables are nonnegative)

There are 3 procedures in simplex method.

1. Express the standard form of LP in table
2. Select a starting basic feasible solution (BFS)
3. Generate a new BFS until optimal solution is found

Examples for simplex method are shown in sections 2.1.1 and 2.1.2.

2.1.1 Problems That Have (≤) Form In All Constraints

Example Minimize $z = -3X_1 - 2X_2 + 5X_3$

Subject to $X_1 + 2X_2 + X_3 \leq 430$

$3X_1 + 2X_3 \leq 450$

$X_1 + 4X_2 \leq 420$

Give $X_1, X_2, X_3 \geq 0$

Solution

1. Set right hand side of objective function to be number

2. Set right hand side of all constraints to be number and ≥ 0, then change ≤ to = with adding slack number (S) to left hand side of all constraints.

$$z + 3X_1 + 2X_2 - 5X_3 = 0$$

$$X_1 + 2X_2 + X_3 + S_1 = 430$$

$$3X_1 + 2X_3 + S_2 = 450$$

$$X_1 + 4X_2 + S_3 = 420$$

Give $X_1, X_2, X_3, S_1, S_2, S_3 \geq 0$

3. Set a standard LP table

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	3	2	-5	0	0	0	0	
S_1	0	1	2	1	1	0	0	430	
S_2	0	3	0	2	0	1	0	450	
S_3	0	1	4	0	0	0	1	420	

4. In the first table, S_1, S_2 and S_3 are basic feasible solutions (BFS), which are 430, 450 and 420, respectively. The z value in the first table is 0. However, it is not the best solution because there are positive values in the first row, which are 3 and 2. In minimization, all values in the first row must be negative or 0 to obtain the optimum z (in maximization, all values in the first row must be positive or 0).

5. The column that has the most positive value in the first row is called pivot column. The variable of pivot column is an entering variable, and in this table entering variable is X_1. The value in b column is divided by the value in pivot column for each row to find ratio. The row that has the least ratio is called pivot row. The variable of pivot row is a leaving variable, and in this table leaving variable is S_2. The leaving variable (S_2) is substituted by the entering variable (X_1) in basic column. The variable which is intersection of the pivot column and pivot row is called pivot number, now it is 3.

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	3	2	-5	0	0	0	0	
S_1	0	1	2	1	1	0	0	430	430/1
X_1	0	3	0	2	0	1	0	450	450/3
S_3	0	1	4	0	0	0	1	420	420/1

6. All of values in pivot row will be divided by pivot number

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	3	2	-5	0	0	0	0	
S_1	0	1	2	1	1	0	0	430	430/1
X_1	0	1	0	2/3	0	1/3	0	150	450/3
S_3	0	1	4	0	0	0	1	420	420/1

7. Make all of values in pivot column except pivot number to be 0 with the equation of Gauss-Jordan below:

$$\text{New row} = \text{Old row} - (\text{Old Pivot Column} \times \text{New Pivot row}) \qquad (1)$$

For example, in the first row,

$3 - 3\times1 = 0$

$2 - 3\times0 = 2$

$-5 - 3\times(2/3) = -7$

$0 - 3\times0 = 0$

$0 - 3\times(1/3) = -1$

$0 - 3\times0 = 0$

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	0	2	-7	0	-1	0	-450	
S_1	0	0	2	1/3	1	-1/3	0	280	
X_1	0	1	0	2/3	0	1/3	0	150	
S_3	0	0	4	-2/3	0	-1/3	1	270	

8. New BFS values are S_1, X_1 and S_3 = 280, 150 and 270, respectively. However, there are positive values in the first row, so the same previous procedure is used to obtain new BFS. New entering variable is X_2, new leaving variable is S_3 and new pivot number is 4.

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	0	2	-7	0	-1	0	-450	
S_1	0	0	2	1/3	1	-1/3	0	280	280/2
X_1	0	1	0	2/3	0	1/3	0	150	
S_3	0	0	4	-2/3	0	-1/3	1	270	270/4

The equation (1) is used to obtain new table shown below.

Basic	z	X_1	X_2	X_3	S_1	S_2	S_3	b	Ratio
z	1	0	0	-20/3	0	-5/6	-1/2	-585	
S_1	0	0	0	2/3	1	-1/6	-1/2	145	
X_1	0	1	0	2/3	0	1/3	0	150	
X_2	0	0	1	-1/6	0	-1/12	1/4	67.5	

The new BFS values are S_1, X_1 and X_2 = 145, 150 and 67.5, respectively. Now there are no positive values in the first row, so this solution is the best solution. The solution for this example are $X_1 = 150$, $X_2 = 67.5$, $X_3 = 0$, $S_1 = 145$, $S_2 = 0$, $S_3 = 0$ and $z = -585$

2.1.2 Problems That Have (=) Or (≥) In Their Constraints

In this problem type, special variable will be added into left hand side of constraints, called artificial variable (R). +R will be added into constraints having (=), and –S+R will be added into constraints having (≥). Then in left hand side of objective function, R variable will be added into with M coefficient (+M for maximization, -M for minimization), M is 1 million. Moreover, R variable will not be entering variable, if it was ever leaving variable for only one time. Finally, in the last standard table, all of R variables must be 0; if not, there is no solution for that problem.

Example Maximize $z = 3X_1 + 5X_2$

Subject to $X_1 \leq 4$

$2X_2 \leq 12$

$3X_1 + 2X_2 = 18$

Give $X_1, X_2 \geq 0$

Solution

1. Set right hand side of objective function to be number

2. Set right hand side of all constraints to be number and ≥ 0. Then if there is $\geq$ in the constraints, $\geq$ must be changed to = and –S+R will be added into left hand side. If it is =, only +R is added into left hand side.

$$z - 3X_1 - 5X_2 + MR = 0$$
$$X_1 + S_1 = 4$$
$$2X_2 + S_2 = 450$$
$$3X_1 + 2X_2 + R = 18$$

Give $X_1, X_2, S_1, S_2, R \geq 0$

3. Eliminate R variable in objective function by substitute $R = 18 - 3X_1 - 2X_2$. Then new objective function is $z - (3M + 3)X_1 - (2M + 5)X_2 = -18M$.

4. Set standard LP table

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	-3M-3	-2M-5	0	0	0	-18M	
S_1	0	1	0	1	0	0	4	
S_2	0	0	2	0	1	0	12	
R	0	3	2	0	0	1	18	

5. In the first table, S_1, S_2 and R are Basic feasible solutions (BFS). The value of z is 0. However, in the first row there are negative values, so they are not the best solution.

6. In the standard table, X_1 is Entering Variable, S_1 is Leaving Variable and Pivot number is 1.

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	-3M-3	-2M-5	0	0	0	-18M	
X_1	0	1	0	1	0	0	4	4/1
S_2	0	0	2	0	1	0	12	
R	0	3	2	0	0	1	18	18/3

7. Do the same with 6, 7 in 2.1.1 section.

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	0	-2M-5	3M+3	0	0	-6M+12	
X_1	0	1	0	1	0	0	4	
S_2	0	0	2	0	1	0	12	
R	0	0	2	-3	0	1	6	

8. They are not the best solution because there is negative value in the first row, so we must find new BFS. New entering variable is X_2, new leaving variable is R and pivot number is 2.

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	0	-2M-5	3M+3	0	0	-6M+12	
X_1	0	1	0	1	0	0	4	
S_2	0	0	2	0	1	0	12	12/2
X_2	0	0	2	-3	0	1	6	6/2

9. Do the same with 6, 7 in 2.2.1 section.

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	0	0	-9/2	0	M+5/2	27	
X_1	0	1	0	1	0	0	4	4/1
S_1	0	0	0	3	1	-1	6	6/3
X_2	0	0	1	-3/2	0	1/2	3	

10. In the first row, there is negative value, so we do the same with previous procedure.

Basic	z	X_1	X_2	S_1	S_2	R	b	Ratio
z	1	0	0	0	3/2	M+1	36	
X_1	0	1	0	0	-1/3	1/3	2	
S_1	0	0	0	1	1/3	-1/3	2	
X_2	0	0	1	0	1/2	0	6	

This is the best solution because there is no negative value in the first row. So $X_1 = 2$, $X_2 = 6$, $S_1 = 2$, $S_2 = 0$, $R = 0$ and $z = 36$.

For the problem that have $\geq$ in constraints, $-S + R$ will be add into the left hand side of that constraint, but only R variable is the first BFS.

<u>Example</u> Minimize $z = 4X_1 + X_2$

Subject to $3X_1 + X_2 = 3$

$4X_1 + 3X_2 \geq 6$

$X_1 + 2X_2 \leq 3$

Give $X_1, X_2 \geq 0$

Then we obtain $z + (7M - 4)X_1 + (4M-1)X_2 - MS_1 = 9M$

$3X_1 + X_2 + R1 = 3$

$4X_1 + 3X_2 - S1 + R2 = 6$

$X_1 + 2X_2 + S2 = 3$

Give $X_1, X_2, S_1, S_2, R_1, R_2 \geq 0$

Basic	z	X_1	X_2	S_1	R_1	R_2	S_2	b	Ratio
z	1	7M-4	4M-1	-M	0	0	0	9M	
R_1	0	3	1	0	1	0	0	3	
R_2	0	4	3	-1	0	1	0	6	
S_2	0	1	2	0	0	0	1	3	

The steps to solve this problem are the same as the previous examples. All of the tables show below.

Basic	z	X_1	X_2	S_1	R_1	R_2	S_2	b	Ratio
z	1	0	(5M+1)/3	-M	(4-7M)/3	0	0	4+2M	
X_1	0	1	1/3	0	1/3	0	0	1	3
R_2	0	0	5/3	-1	-4/3	1	0	2	6/5
S_2	0	0	5/3	0	-1/3	0	1	2	6/5

Basic	z	X_1	X_2	S_1	R_1	R_2	S_2	b	Ratio
z	1	0	0	1/5	8/5-M	-1/5-M	0	18/5	
X_1	0	1	0	1/5	3/5	-1/5	0	3/5	3
X_2	0	0	1	-3/5	-4/5	3/5	0	6/5	-
S_2	0	0	0	1	1	-1	1	0	0

Basic	z	X_1	X_2	S_1	R_1	R_2	S_2	b	Ratio
z	1	0	0	0	7/5-M	-M	-1/5	18/5	
X_1	0	1	0	0	2/5	0	-1/5	3/5	
X_2	0	0	1	0	-1/5	0	3/5	6/5	
S_1	0	0	0	1	1	-1	1	0	

The results from the solving are 3.6 for the minimum point. X_1 and X_2 are 0.6 and 1.2, respectively.

2.2 Branch-And-Bound

Branch-and-bound is technique used to solve both of mixed-integer (MILP) and pure-integer (IP) problems. This technique is very advantage to decrease the procedures in searching for the optimal point. Generally, numbers of solution in IP and MILP problems are calculated with C^n (C is number of decisions and n is number of integer variables). But in this branch-and-bound technique, the feasible solution will be branched to current subproblems to find the best solution, which reduce solving procedures. This technique is separated to 2 types: binary integer programming (BIP), and mixed integer linear programming (MILP) (Hillier, 2010).

2.2.1 Branch-And-Bound For Integer And Mixed-Integer Linear Solution

Mixed integer linear programming (MILP) is used to solve linear problem with mixed-integer-real variables. The problem that has only integer variables is called integer linear programming (IP). Normally, general form of MILP problem is

Maximize/Minimize $Z = \sum_{j=1}^{n} c_j x_j,$

Subject to $\sum_{j=1}^{n} a_{ij} x_j \leq b_i,$ for i = 1, 2, . . . , m,

And $x_j \geq 0,$ for j = 1, 2, . . . , n,

x_j is integer, for j = 1, 2, . . . , I; $I \leq n$.

(If $I = n$, the problem is pure IP problem.)

For MILP, the general integer-restricted variable could have very large numbers of possible integer values. Therefore, two ranges of the subproblems would be specified. Simplex method will be used for LP relaxation in current subproblem. Then, the integer-restricted variable will be taken upper bound and lower bound and branched to other nodes for integer solution.

To descript how this is done, x_j is identified as the current branching variable, and x_j^* is identified as its value from the current subproblem. Using square brackets to denote

$[x_j^*]$ = greatest integer $\leq x_j^*$,

the range of the two new subproblems would be specified

$x_j \leq [x_j^*]$ and $x_j \geq [x_j^*] + 1,$

respectively. The inequalities will be added to constraint of the two new subproblem. Finally, x_j would be fixed to one integer value. The example shows in Fig. 2.1

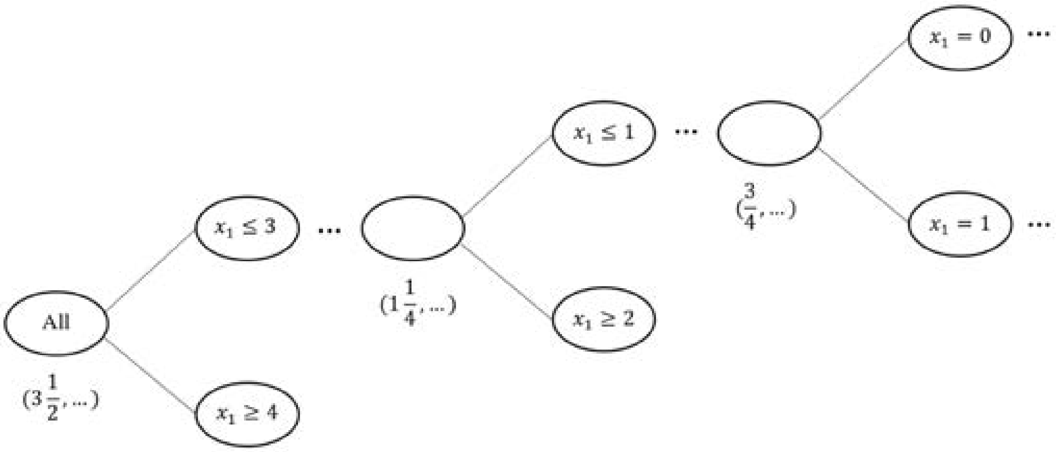


Figure 2.1 Bounding and branching example for the two new subproblem.

From Fig. 2.1, if $x_j^* = 3\frac{1}{2}$, then

$x_j \le 3$ and $x_j \ge 4$,

are the additional constraints for the new subproblems. When solution from the subproblem is $1\frac{1}{4}$, the additional constraints $x_1 \le 1$ and $x_1 \ge 2$ would be added in the two new subproblem ($x_1 \le 3$ is still additional constraint). When the $x_1 \le 1$ subproblem is solve, the solution is $\frac{3}{4}$. The new subproblem would be created as $x_1 \le 0$ and $x_1 \ge 1$, so $x_1 = 0$ and 1. The MILP branch-and-bound could be summarized as below:

1. Branching: For branching to the new subproblem, let x_j be integer-restricted variable from the current subproblem and x_j^* is its value from LP relaxation. Identified the two new subproblems as $x_j^* \le [x_j^*]$ and $x_j^* \ge [x_j^*] + 1$ respectively, when $[x_j^*]$ = greatest integer $\le x_j^*$.

2. Bounding: For each new subproblem, used simplex method to solve, and obtain Z value for resulting optimal solution.

3. Fathoming: For each new subproblem, discard the subproblems that are fathomed by any of following tests.

Test 1: Its bound $\le Z^*$, where Z^* is Z from the current incumbent.

Test 2: The subproblem has no feasible solution.

Test 3: All of the integer-restricted variables are integer value.

Example Maximize $Z = 4X_1 – 2X_2 + 7X_3 – X_4$

Subject to

$$
\begin{aligned}
X_1 + 5X_3 &\leq 10 \\
X_1 + X_2 - X_3 &\leq 1 \\
6X_1 - 5X_2 &\leq 0 \\
-X_1 + 2X_3 – 2X_4 &\leq 3
\end{aligned}
$$

$X_1, X_2, X_3, X_4 \geq 0$

X_1, X_2, X_3 are integer.

Before using MILP technique, the continuous solutions are be obtained by LP optimization (simplex method). The branch-and-bound performance for the example shows in Fig. 2.2 and is descripted as follow:

Node 1: The continuous solutions are obtained by LP with non-integer solutions. Node 2 is branched from Node 1 with $X_1 \geq 2$, and is solved by LP.

Node 2: There are no feasible solutions in this node because of non-balance of the constraints. Therefore, there is no branching from this node with the second fathoming.

Node 3: This node is branched from Node 1 with $X_1 \leq 1$, and is solved by LP. X_1 is 1, which is integer. However, X_2 and X_3 are not. So this solution is not the correct solution. Node 4 would be branched from this node with $X_2 \geq 2$ because X_2 is 1.2.

Node 4: X_2 is 2, which is integer, but the others are not. So it is not the correct solution. However, X_1 become non-integer because of the constraint $X_1 \leq 1$, it is not fixed as integer value. Therefore, Node 5 would be branched with $X_1 \geq 1$.

Node 5: With non-balance of constraints, this node has no the feasible solution.

Node 6: This node is branched from Node 4 with $X_1 \leq 0$. In this node, all integer-restricted variables are integer: X_1, X_2, and X_3 = 0, 2 and 2, respectively. The Z value is 9.5. The Node branch stops with the third fathoming.

Node 7: Equation ($X_2 \leq 1$) is added to the constraints of this node with branching from Node 3. The solutions are not the correct solutions because of non-integer values. Node 8 and Node 9 are branched with $X_1 \geq 1$ and $X_1 \leq 0$, respectively.

Node 8 has no feasible solution. And all integer-restricted variables are integer in Node 9. The Z value is 13.5, and Node branching stops.

From 9 nodes, the nodes that are the correct solution (all integer-restricted variables are integer), are Node 6 and Node 9. Z from Node 9 is 13.5, while one from Node 6 is 9.5. So the best solution for this example is 13.5 with X_1, X_2, X_3 and X_4 are 0, 0, 2 and 0.5, shown in Node 9 in Fig. 2.2.

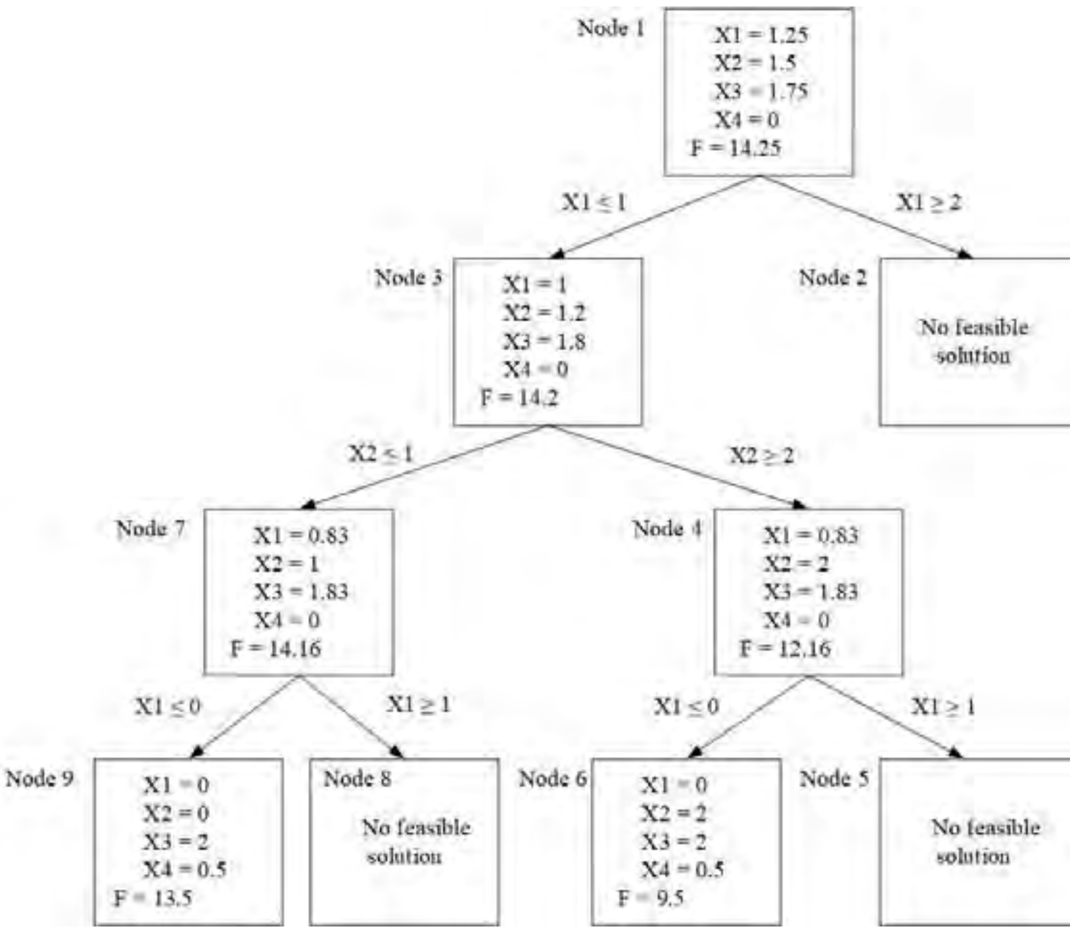


Figure 2.2 Branch and bound technique for integer linear program.

2.2.2 Branch-And-Bound For Binary Solution (0 And 1)

Binary integer programming (BIP) is used to solve linear problem with binary (0, 1) solution. Normally, BIP is used for decision problems. The solving procedure (branch-and-bound) is the same with MILP, but it is different with adding (≤) equation to the constraint to fix range of the binary variable value as 0 and 1.

Example Maximize $f = 86X_1 + 4X_2 + 40X_3$

Subject to $774X_1 + 76X_2 + 42X_3 \leq 875$

$67X_1 + 27X_2 + 53X_3 \leq 875$

Give $X_1, X_2, X_3 = 0, 1$

The example is pure integer problem. The problem will be solved by branch-and-bound technique. Three new constraints are added to the constraints:

$X_1 \leq 1$

$X_2 \leq 1$

$X_3 \leq 1.$

The BIP procedures show in Fig. 2.3 and are described as follows:

Node 1: Obtain continuous solution by LP technique, and all variables are ≤ 1. $X_1 = 1$, $X_2 = 0.776$, $X_3 = 1$.

Node 2: At the first node, X_2 is not integer, so Node 2 will be branched from the first node and X_2 will be set to ≥ 1. Then, LP is used to solve. (For $X_2 \leq 0$ will be branched for solving later)

Node 3: At Node 2, X_1 is 0.978, not integer. So it branches to Node 4 and sets $X_1 \geq 1$, and then does LP relaxation.

Node 4: At Node 3, X_3 is 0.595, not integer. Node 4 is generated for $X_3 \geq 1$. Because of non-balance in constraints, so Node 4 is not feasible solution and the node branching stops.

Node 5: this is branched from Node 3 with $X_3 \leq 0$. All integer-restricted variables are integer. The f value is 90.0. And the node branching stops because of this.

Node 6: this node is branched from Node 2 with $X_1 \leq 0$. All integer-restricted variable are integer with f = 44.0.

Note 7: This Node is branched from Node 1 with $X_2 \leq 0$. The objective function is 126 with integer solutions.

From 7 Nodes, 3 Nodes: Note 4, 5 and 6, give feasible solutions. Node 6 gives the greatest value of f, 126.0. So this node is the best solution. X_1, X_2 and X_3 = 1, 0 and 1 respectively, shown in Node 7 in Fig. 2.3.

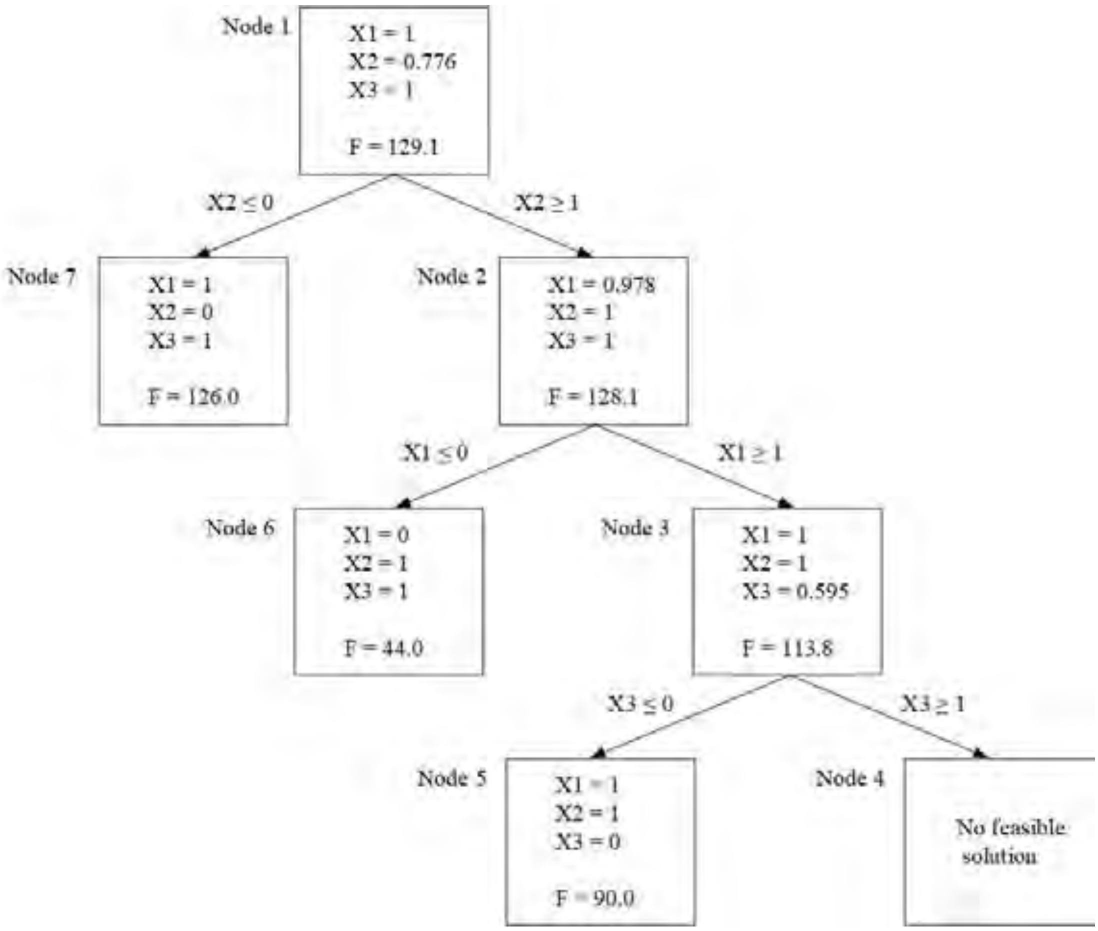


Figure 2.3 Branch and bound technique for integer linear program.

So, BIP can be concluded to 2 steps:

1. Add $X_j \leq 1$ into the constraints for integer-restricted variables.
2. Do the next procedures the same as MILP procedures.

Moreover, most of the decision problems consist of both decision variables (0, 1) and continuous variables (real value). For this case, the example is shown below.

<u>Example</u> Maximize $Z = 5X_1 + 7X_2 + 3X_3$

Subject to

$$
\begin{aligned}
X_1 &\leq 7 \\
X_2 &\leq 5 \\
X_3 &\leq 9 \\
X_1 - 100y_1 &\leq 0 \\
X_2 - 100y_2 &\leq 0 \\
X_3 - 100y_3 &\leq 0 \\
y_1 + y_2 + y_3 &\leq 2 \\
3X_1 + 4X_2 + 2X_3 - 100y_4 &\leq 30 \\
4X_1 + 6X_2 + 2X_3 - 100y_4 &\leq 4{,}000
\end{aligned}
$$

Give X_1, X_2 and $X_3 \geq 0$

y_1, y_2, y_3 and y_4 are binary.

The example is solved by using branch-and-bound technique. In the first step, the linear programming is used to solve for the continuous solution. The branch-and-bound technique is used to find the BIP solution. The procedure for solving shows in Fig. 2.4.

In the first node, the continuous solution is obtained by linear programming. All of integer-restricted variables are not binary. So the second node is branched from the first node by adding $y_1 \geq 1$.

In the second node, the LP relaxation is used to solve the problem. The y_1 and y_4 are 1, binary. But y_2 and y_3 are not binary, so the third node is branched by adding $y_2 \geq 1$ to the constraints.

After the solution is obtained from the LP relaxation in the third node, all of integer-restricted variables are binary. The maximum point for this node is 70.

For the forth node and the fifth node, the maximum points are 62. These maximum points are less than 70, so these nodes stop the branching. So the best maximum point for the problem is 70 with 1, 1, 0 and 1 for y_1, y_2, y_3 and y_4, respectively, shown in subproblem 2 in Fig. 2.4.

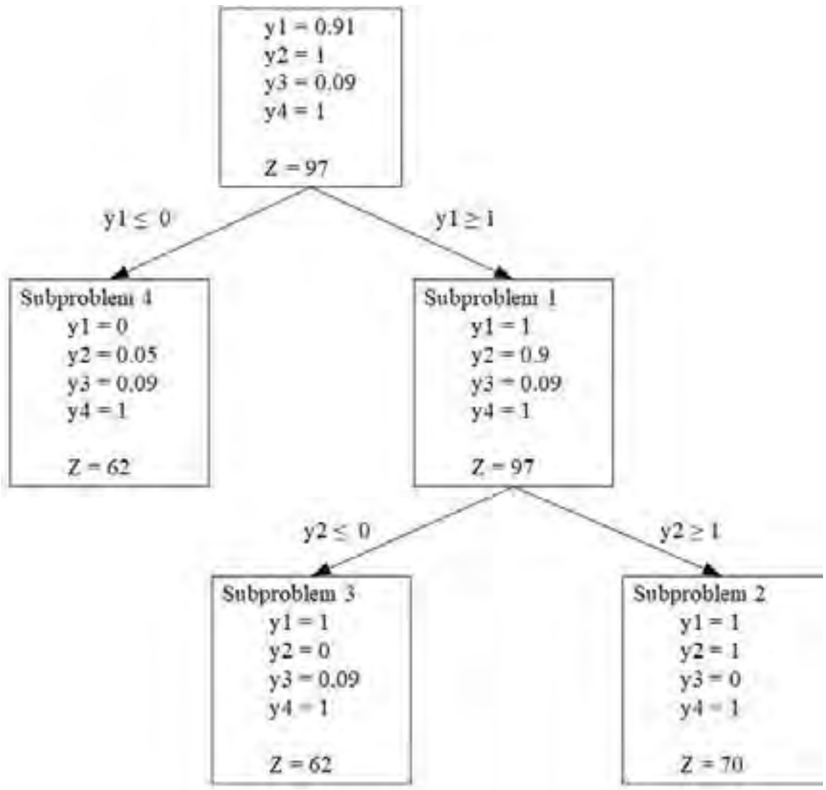


Figure 2.4 The example for BIP (mixed binary real).

LITERATURE REVIEW

1. An MILP model for optimization of a small-scale LNG supply chain along a coastline

Jokinen *et al.* (2015) generally, natural gas is transported in liquid form, called liquefied natural gas (LNG). It is transported through small terminal networks that combine with sea and land by ship. However, the supply chain network building is complicated and has high cost.

In this research, mathematical model of optimization for minimizing the cost is presented for LNG supply chain. For the case study, energy requirement of a country is designed under different cost structures for LNG and for land-based transportation.

The IBM ILOG CPLEX Optimization Studio 12.5 optimization software was performed on a computer running a 64-bit Windows 7 operating system with a 3.5 GHz Intel Core i7 processor and 16 GB of RAM. The test case problem included 630 binary, 360 integer and 1924 continuous variables.

2. Deployment of a hydrogen supply chain by multi-objective/multi-period optimization at regional and national scales

Almaraz *et al.* (2015) in this paper, a methodological framework for the design of a five-echelon hydrogen supply chain (HSC) (energy source, production, storage, transportation and fuelling station) are developed by considering the geographic level of implementation. Three objectives have to be optimised simultaneously, i.e., cost, global warming potential and safety risk are studied based on mixed integer linear programming. Roadmaps in the French is considered for data collection and demand scenarios. The results between different geographic scale cases is compared.

The model is formulated within GAMS environment, and solved using CPLEX 12.

3. Supply chain optimization of residual forestry biomass for bioenergy production: The case study of Portugal

Paulo *et al.* (2015) normally, renewable energies is good alternative energy. However, for the renewable energies, problems in the present is sourcing problems. The bioenergy can be an attractive solution if effectively managed.

In this paper, the supply chain design is presented by mathematical programming to study the design of the residual forestry biomass to bioelectricity production in the Portuguese context. The design and planning of the bioenergy supply chain is optimized by a mixed integer linear programming (MILP) model. The model consist of the optimal selection of biomass amounts and sources, the transportation modes selection, and links that must be established for biomass transportation and products delivers to markets. Results illustrate the positive contribution of the mathematical programming approach to achieve viable economic solutions.

4. Optimization-based approach for strategic design and operation of a biomass-to-hydrogen supply chain

Woo *et al.* (2016) design and operation of a renewable hydrogen system are optimized from various types of biomass. Mixed integer linear programming model is developed to determine the optimal logistics decision-making to minimize the total annual cost for a comprehensive biomassto-hydrogen (B2H2) supply chain with import and inventory strategies. The optimal design of the supply chain and main cost-drivers, manage logistics operations against fluctuations of biomass availability and hydrogen demand, and making strategic decisions for planning the B2H2 system such as capital investment and energy import planning is identified in this paper. A case study of an upcoming B2H2 supply chain for the transportation sector at Jeju Island, Korea, is analyzed.

The proposed MILP model is executed in General Algebraic Modeling System (GAMS) software for computational experiment. A B2H2 supply chain with operational schedule is designed considering the available parameters in 2040, Jeju Island (see session 4.). The case is solved using CPLEX 12.4.0.1 Solver.

5. Supply chain network design and operation: Systematic decision making for centralized, distributed, and mobile biofuel production using mixed integer linear programming (MILP) under uncertainty

Sharifzadeh *et al.* (2015) the biofuel is one good alternative energy with cheap process and can be conducted in centralized, decentralizes, or mobile configurations. Generally, wastes or lignocellulose is used in the process without human food, so it is not overlap with the human food supply chain. And several advantages is obtained for biofuel production from fast pyrolysis. However, biomass resources are dispersed and subject to seasonal and geographical uncertainties. So in this research, a mixed integer (piece-wise) linear program (MILP) was developed to determine the optimal supply chain design and operation, under uncertainty. Rigorous process modelling and detailed economic analysis were coupled with exhaustive search of potential production locations and biomass resources in order to enhance the fidelity of the solution.

The optimisation problem was programmed as a mixed integer linear program (MILP), implemented in GAMS and was solved with CPLEX 12.1.0.

6. Cost optimization of biofuel production – The impact of scale, integration, transport and supply chain configurations (Jong, 2017)

Jong *et al.* (2017) in this research, the optimization model is presented to analyze the impact of four cost reduction strategies for biofuel production: economies of scale, intermodal transport, integration with existing industries, and distributed supply chain configurations (i.e. supply chains with an intermediate pre-treatment step to reduce biomass transport cost). The model assessed biofuel production levels ranging from 1 to 150 PJ a^{-1} in the context of the existing Swedish forest industry. Biofuel was produced from forestry biomass using hydrothermal liquefaction and hydroprocessing. Disabling the benefits of integration favors large-scale centralized production, while intermodal transport networks positively affect the benefits of economies of scale. As biofuel production costs still exceeds the price of fossil transport fuels in Sweden after implementation of all cost reduction strategies, policy support and stimulation of further technological learning remains essential to achieve cost parity with fossil fuels for this feedstock/technology combination in this spatiotemporal context. The model was written in GAMS using a CPLEX solver.

7. Optimal design and planning of biodiesel supply chain considering non-edible feedstock

Babazadeh *et al.* (2017) in the present, first-generation biodiesel production from vegetable edible oils and animal fats has triggered a sense of concern among policymakers and development practitioners about farm land allocation, food supply, and food market equilibrium. Utilization of second-generation biodiesel from nonedible feedstocks has been attracted many interests in recent years. To accelerate transition towards large-scale and economic viable biofuels, systematic design and optimization of entire biofuel supply chains is crucial. The proposed model is capable to determine the optimum numbers, locations, capacity of facilities, suitable transportation modes, appropriate technology at bio-refinery, material flow, and production planning in different periods. The proposed model is applied in a real case in Iran. They consider Jatropha seeds and waste cooking oil as non-edible feedstocks for second-generation biodiesel production in the studied case.

Due to MILP structure of the proposed model, CPLEX solver of the GAMS optimization software is used for solving the model. The optimum solution was achieved after 1659 s. The presented model has 50166 equations, 701924 continuous, 173 binary variables.

8. Design of regional and sustainable bio-based networks for electricity generation using a multi-objective MILP approach

Perez-Fortes *et al.* (2012) this work is focused on a mathematical programming approach applied to bio-based supply chains that use locally available biomass at or near the point of use in order to produce electricity or other bioproduct. The problem of designing and planning a regional biomass supply chain is formulated as a MO-MILP (multiobjective mixed integer linear program), which takes into account three main objectives: economic, environmental and social criteria. The model supports decision-making about location and capacity of technologies, connectivity between the supply entities, biomass storage periods, matter transportation and biomass utilisation. The advantages of this approach are highlighted by solving a case study of a specific district in Ghana. The aim is to determine the most suitable biomass and electricity network among the different communities. The technology considered to transform the biomass into electricity is gasification combined with a gas engine.

The mathematical model has been written in GAMS and solved using CPLEX 11.0 on a PC Intel(R) Core(TM) i7-2620M CPU 2.70 GHz and 4.00 Gb of RAM. The optimisation model contains 27122 equations, 818215 continuous and 1115 discrete variables.

9. Supply chain optimization of sugarcane first generation and eucalyptus second generation ethanol production in Brazil

Jonker *et al.* (2016) the expansion of the ethanol industry in Brazil faces two important challenges: to reduce total ethanol production costs and to limit the greenhouse gas (GHG) emission intensity of the ethanol produced. The objective of this study is to economically optimize the scale and location of ethanol production plants given the expected expansion of biomass supply regions. A linear optimization model is utilized to determine the optimal location and scale of sugarcane and eucalyptus industrial processing plants given the projected spatial distribution of the expansion of biomass production in the state of Goiás between 2012 and 2030. Three expansion approaches evaluated the impact on ethanol production costs of expanding an existing industry in one time step (one-step), or multiple time steps (multi-step), or constructing a newly emerging ethanol industry in Goiás (greenfield).

10. Optimal supply chain network design with process network and BOM under uncertainties: A case study in toothbrush industry

Pham *et al.* (2017) design of supply chain network significantly affects supply chain performance for long period. Since each industry has a unique set of characteristics which evidently drive the design supply chain network, a number of various models have been formulated to meet the needs of such business contexts. Even though many models have been proposed for manufacturing industry context, most of them are based on the facility location model. It tends to lead the supply chain network design model to be complicated. Therefore, the purpose of this research is to propose an alternative approach to formulate manufacturing network design problem. Features, such as multi-echelon, multi-commodity, products structure, and manufacturing process, are taken into consideration as characteristics of the studied environment. Two models, deterministic and fuzzy models, have been explored in the study and both of them have demonstrated the validity of the proposed formulation method.

The OPL models are solved by MIP solver in IBM ILOG 12.6 system. The computer which is used is AMD Quad-core Processor, 1.9 GHz, 4 GB of Ram. The case problem has 112 binary and 64,188 integer variables, and 1,492 constraints.

11. An engine oil closed-loop supply chain design considering collection risk

Paydar *et al.* (2017) manufacturers are devising new methods to make production systems more efficient and effective. Designing an optimized supply chain can support the corresponding processes to integrate the resources. However, one of the most important obstacles is the resource limitation. Recycling the second-hand products is one of the approaches to cope with this issue. Reverse logistics is a system of collecting products from end-users to the manufacturing centers for obtaining values from collected materials. In this research, the collection and distribution process of engine oil is considered. A MILP model for a closed-loop supply chain of used engine oil is proposed and a case study of an oil refinery company is proposed to explore the applicability of the model. Two objective functions of maximizing profit and minimizing the risk of the collection are considered.

There are 12,031 variables, 9698 constraints and 57,477 nonzeros (parameters) in the model. The mathematical model was solved using LINGO 15.0 on a laptop computer equipped with Intel® CoreTM i5-4800M CPU @ 2.66 GHz and 4.00 GB RAM.

CHAPTER III
METHODOLOGY

Materials And Equipment

Equipment:

Lenovo (Intel® Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, 4.00 GB of RAM, Windows 10 Pro, 64-bit Operating system)

Software:

1. Fortran 4.0
2. Microsoft Office 2013: Excel

Experimental Procedures

3.1 Develop LP Algorithm

3.1.1 <u>Simplex Algorithm</u>

For inequality (≤) constraint, all of the constraints have to be converted to equality by adding slack number (s). The right hand side of both objective function and constraints must be positive or zero value. The simplex would be applied as below procedures.

1. Add the coefficient to standard table.
2. Specify the first basic feasible solution.
3. Choose the most negative value in the first row, and identify it as pivot column and entering variable.
4. Divide the right hand side values by the values in the pivot column to select the least positive ratio, and identify it as pivot row and leaving variable. The value in intersection between pivot column and pivot row, is called pivot number.
5. Use Gauss-Jordan equation to obtain the new solution.
6. Check the first row to find the negative value, and follow it as step 1-5. If there is on negative value in the first row, stop the algorithm and obtain the best solution.

If it is minimization, the objective function would be multiplied by -1. For equality and inequality (≥) constraints, slack number and artificial number are added to the constraints to become equality. Million time of artificial number would be added to the objective function. Finally, the 1-6 step as the previous would be followed. This method is called Big - M. The algorithm shows in Fig. 3.1.

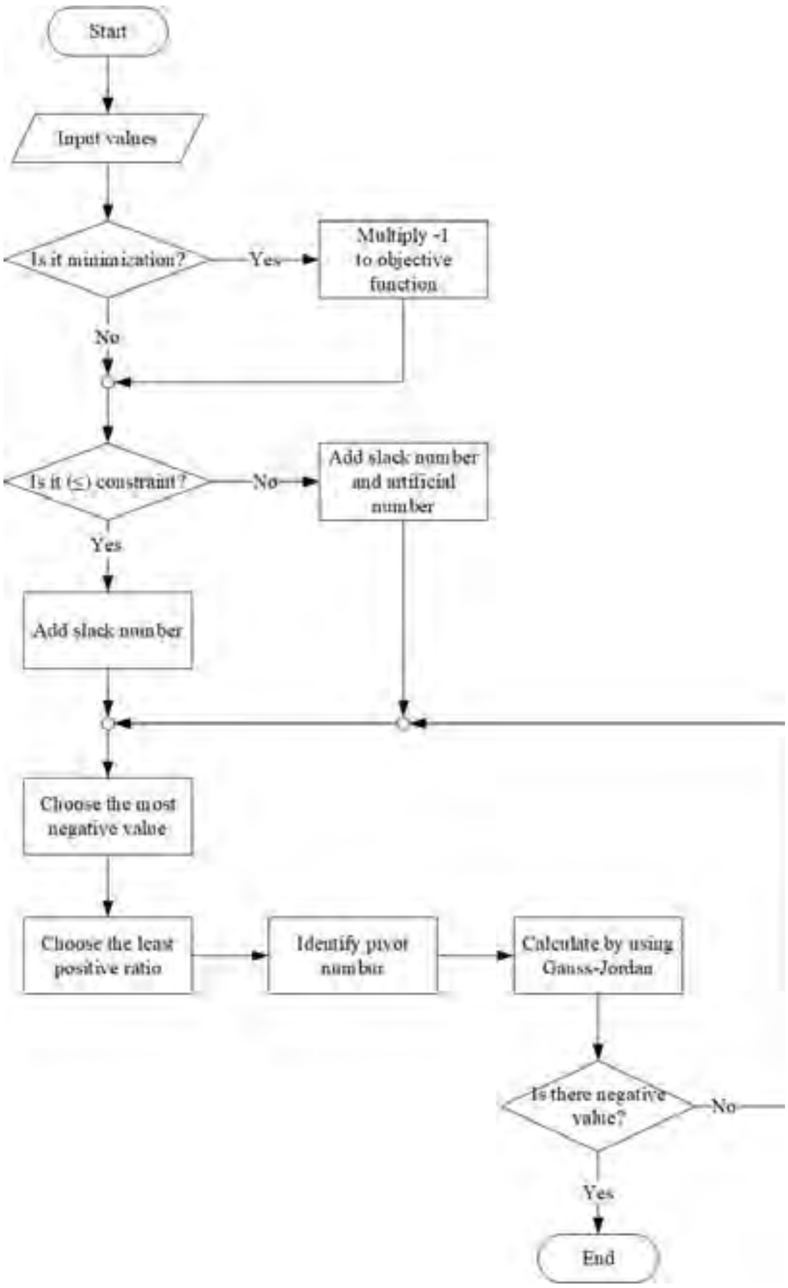


Figure 3.1 Flowchart of simplex algorithm for inequality (≤) constraints.

3.1.2 Simplex Algorithm Evaluation

The simplex algorithm would be evaluated with small supply chain problem as show in Fig. 3.2. The profit of the problem is maximized.

For the problem description, some products are produced in 3 plants, stocked in 2 warehouses and then sent to 3 markets. The operating plants and the warehouses have different capacities and the markets have different demands with difference of transportation distance. The mass balance of items would be managed to obtain the maximum profit. The developed simplex algorithm would be validated with Microsoft excel.

The transportation cost = 90 $ per kmile per item, operating in warehouse cost = 300 $ per item, operating in plant cost = 500 $ per item and product price = 1,500 $ per item.

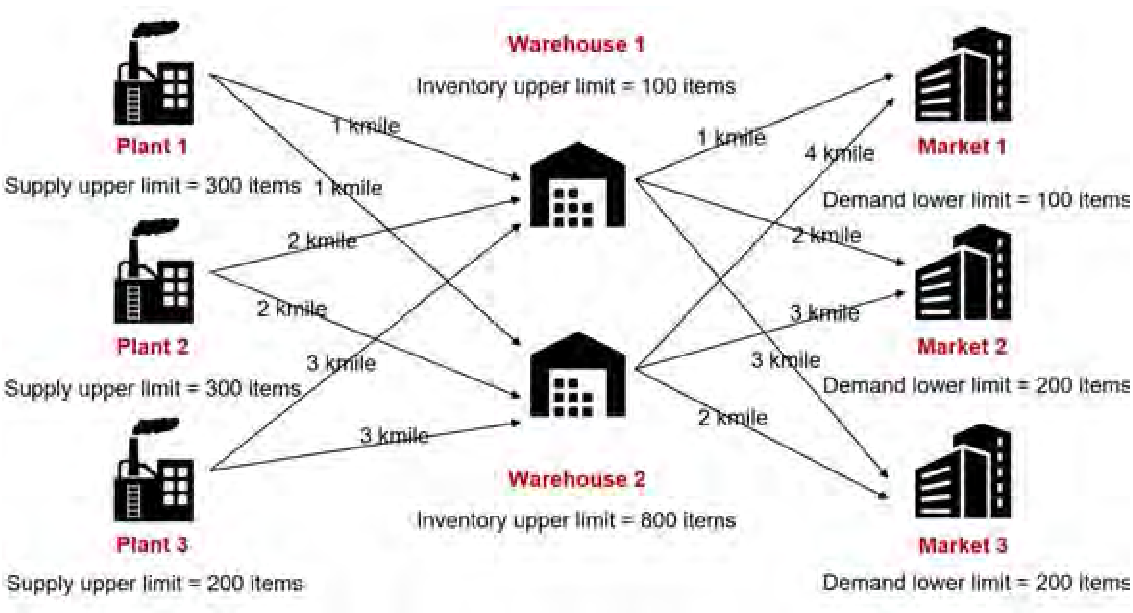


Figure 3.2 The small supply chain problem.

3.2 Develop MILP Algorithm

3.2.1 Branch-And-Bound Algorithm

Branch-and-bound algorithm is divided to binary programing algorithm (BIP) and Mixed integer linear programming algorithm (MILP). The algorithms are explained respectively, and shown in Fig. 3.3. For MILP algorithm, the procedures are summarized and shown below.

1. Branching: For branching to the new subproblem, let x_j be integer-restricted variable from the current subproblem and x_j^* is its value from LP relaxation. Identified the two new subproblems as $x_j^* \leq [x_j^*]$ and $x_j^* \geq [x_j^*] + 1$ respectively, when $[x_j^*]$ = greatest integer $\leq x_j^*$.

2. Bounding: For each new subproblem, used simplex method to solve, and obtain Z value for resulting optimal solution.

3. Fathoming: For each new subproblem, discard the subproblems that are fathomed by any of following tests.

Test 1: Its bound $\leq Z^*$, where Z^* is Z from the current incumbent.

Test 2: The subproblem has no feasible solution.

Test 3: All of the integer-restricted variables are integer value.

For BIP algorithm, the procedure could be concluded to 5 steps below.

1. Add $X_j \leq 1$ into the constraints for integer-restricted variables.

2. Do the next procedures the same as MILP procedures.

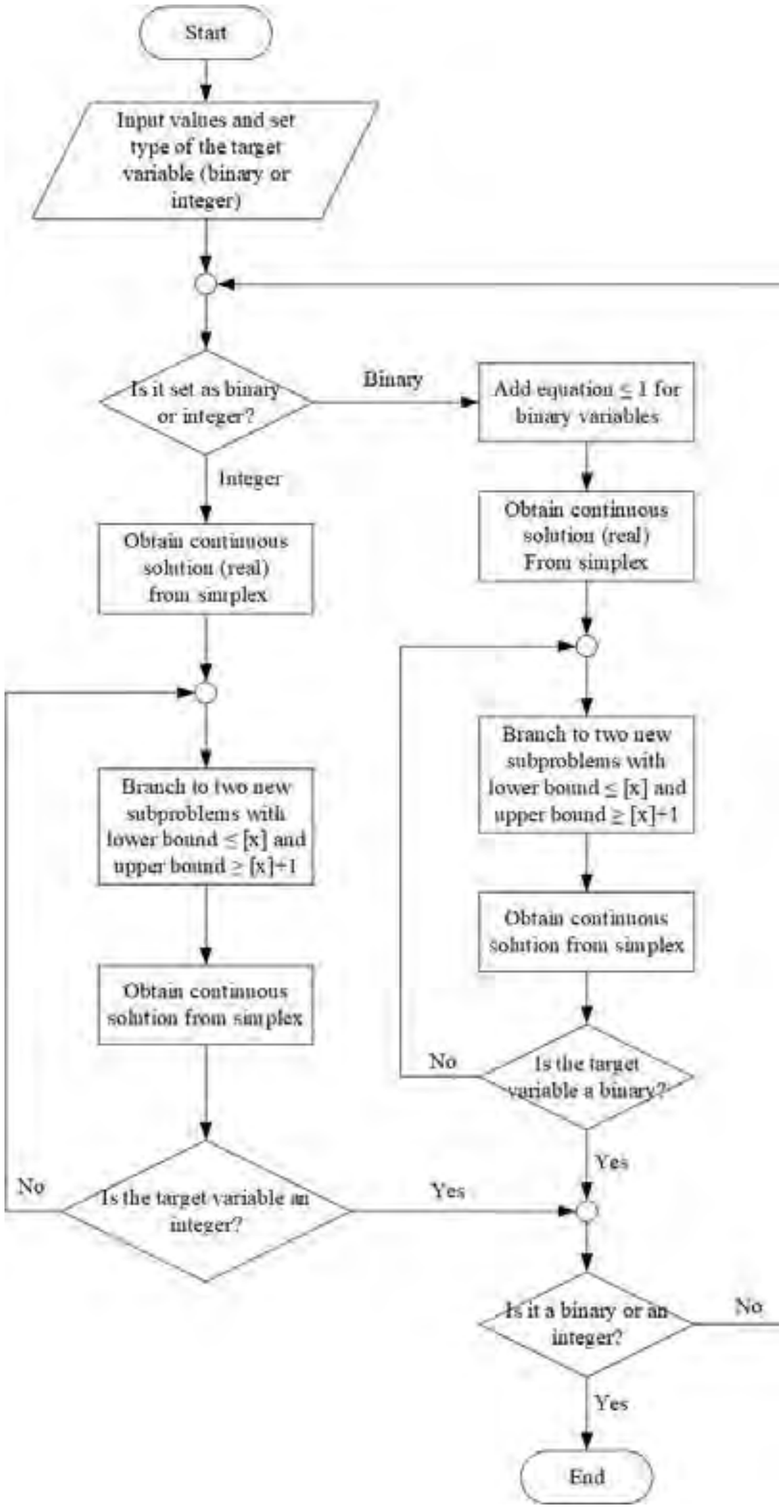


Figure 3.3 Algorithm for BIP and MILP.

3.2.2 Branch-And-Bound Algorithm Evaluation

The branch-and-bound algorithm evaluation is divided into 2 parts: BIP evaluation, and MILP evaluation.

The problem used in BIP evaluation is the same one that used in simplex evaluation (Fig. 3.2). The value of items is used as continuous variable, and a number of the chosen ways is considered decision variables. The cost and the profit are not used as constants, but they are found after optimizing.

For the problem used in MILP evaluation, the project description is used. A microelectronics manufacturing facility is considering six projects to improve operations as well as profitability. However, not all of these projects can be implemented due to expenditure limitations as well as engineering manpower constraints. Table 3.1 gives projected cost, manpower, and profitability data for each project.

Table 3.1 The project problem data (Edgar, 1989)

Project	Description	First year expenditure	Second year expenditure	Engineering hours	Net present value
1	Modify existing production line with new etchers	\$300,000	0	4000	\$100,000
2	Build new production line	\$100,000	\$300,000	7000	\$150,000
3	Automate new production line	0	\$200,000	2000	\$35,000
4	Install plating line	\$50,000	\$100,000	6000	\$75,000
5	Build waste recovery plant	\$50,000	\$300,000	3000	\$125,000
6	Subcontract waste disposal	\$100,000	\$200,000	600	\$60,000

The resource limitations are:

First year expenditure: \$450,000

Second year expenditure: \$400,000

Engineering hours: 10,000

A new or modernized production line must be provided (Project 1 or 2). Automation is feasible only for the new line. Either project 5 or project 6 can be selected, but not both. The total net present value subject would be maximized by using the developed algorithm and Microsoft excel.

3.3 Biofuel Production Supply Chain Case Study

Biofuel is a fuel from agriculture or lignocellulosic feedstock. Normally, two steps are used to produce biofuel from lignocellulosic feedstock that are fast pyrolysis reaction and chemical treatment (hydrothermal upgrading and hydrodeoxygenation) (Sharifzaden, 2015).

In this case, two strategies are considered for biofuel production: (1) Centralized processing strategy where fast pyrolysis plant is used to produce pyrolysis oil and sends to upgrading center, benefiting from economic of scale and low transportation cost because of operating plant location near resource, and (2) Remote processing strategy where pyrolysis oil produced by the mobile pyrolyzer is sent to upgrading center for biofuel production, benefiting from cheap price of supplier.

The case study would be simulated to produce biofuel to two customers, Liverpool and London, by using the 2 different types of resources: Supplier type 1 is commercial resource, and supplier type 2 is forest resource. The fast pyrolysis plant and upgrading plant would be separated to install near resources and customers. The 2 locations would be simulated for upgrading centers near customers. And mobile pyrolyzer would be used for forest resources. The biofuel production supply chain graphic shows in Fig. 3.4.

Figure 3.4 The graphic model for biofuel production supply chain case study.

From Fig. 3.4, resource from Hereford is identified as suppler type 1, Bristol is supplier type 2, fast pyrolysis plant is located in Birmingham and finally Liverpool and London are used to locate upgrading plants and customers.

The case study is divided to three parts for evaluation, MILP evaluation, BIP evaluation and Full MILP (integer and binary) evaluation, which are validated with Microsoft excel. The simulation model shows in Fig. 3.5.

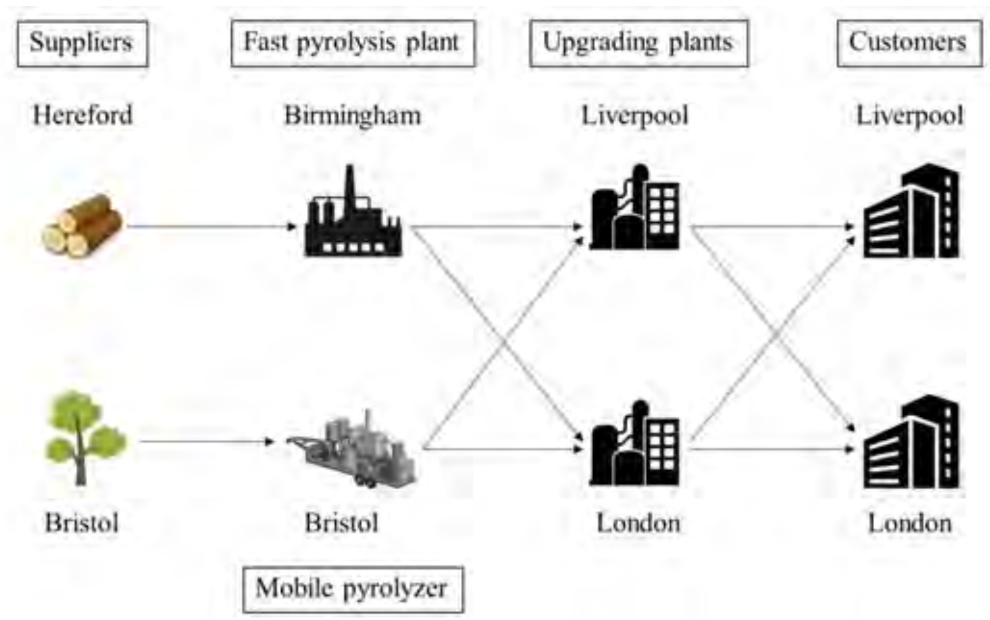


Figure 3.5 Biofuel production case study.

CHAPTER IV
RESULTS AND DISCUSSIONS

4.1 Simplex Algorithm Evaluation

The simplex algorithm is developed in Fortran 4.0, evaluated with the small supply chain problem comparing with Microsoft excel.

From Fig. 3.2, the products produced by three plants would be transferred to two warehouses and then transferred to three markets. The upper limit for plant 1, 2 and 3 is 300, 300 and 200 respectively. The capacity for warehouse 1 and 2 is 100 and 800. The lower limit for demand 1, 2 and 3 is 100, 200 and 200 respectively. For the problem, the maximum profit would be found.

The mass flow from the plant to the warehouses is specified as $x_{i,j}$ and from the warehouses to the markets is specified as $y_{j,k}$, when i is number of the plants, j is number of the warehouses, k is number of the markets. Therefore, the mass flow variables are identified as show in Fig. 4.1.

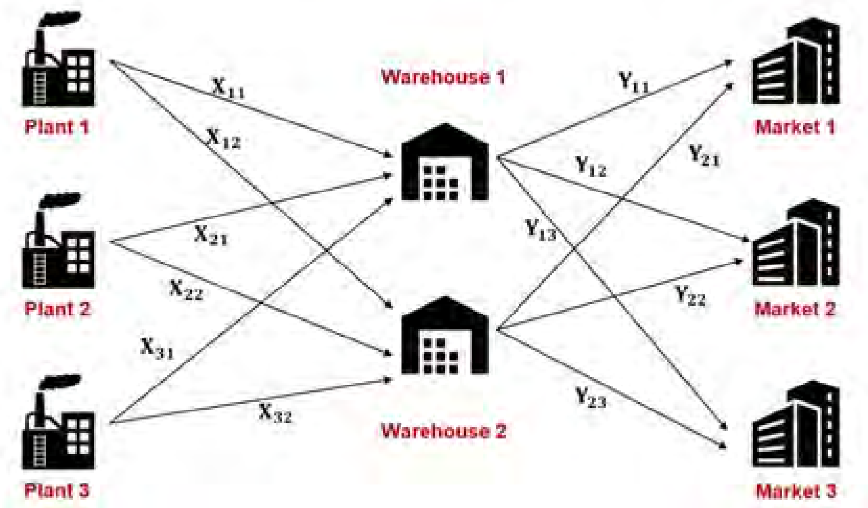


Figure 4.1 Mass flow variables of the supply chain.

The distance between the nodes shows in Fig. 3.2. To maximize profit of the problem, mathematical model would be simulated.

The objective function is specified as difference between summation of income and summation of cost, as show below.

$$Maximize \quad profit = \sum income - \sum cost$$

So, the objective function would be specified below.

$$Maximize \quad profit = totat\ income - total\ transportation\ cost$$
$$-total\ operating\ cost$$

The constraints would be divided to mass balance, transportation cost, operating in warehouse cost, operating in plant cost and total income. The mass balance constraints are taken around every node and around the system (overall mass balance). Equation (1) is production limit for each plant, limiting with amounts of suppliers.

$$\sum_j x_{i,j} \leq Supplier^i \qquad \forall i \tag{1}$$

The value substitution for equation (1) shows as equations (1A) – (1C).

$$x_{1,1} + x_{1,2} \leq 300 \tag{1A}$$

$$x_{2,1} + x_{2,2} \leq 300 \tag{1B}$$

$$x_{3,1} + x_{3,2} \leq 200 \tag{1C}$$

Equation (2) is minimum demand for each market.

$$\sum_j y_{j,k} \geq Demand^k \qquad \forall k \tag{2}$$

The value substitution for equation (2) shows as equations (2A) – (2C).

$$y_{1,1} + y_{2,1} \geq 100 \tag{2A}$$

$$y_{1,2} + y_{2,2} \geq 200 \tag{2B}$$

$$y_{1,3} + y_{2,3} \geq 200 \tag{2C}$$

The capacity of the warehouse is limited by equation (3).

$$\sum_i x_{i,j} \leq Inventory^j \qquad \forall j \tag{3}$$

The value substitution for equation (3) shows as equations (3A) – (3B).

$$x_{1,1} + x_{2,1} + x_{3,1} \leq 100 \tag{3A}$$

$$x_{1,2} + x_{2,2} + x_{3,2} \leq 800 \tag{3B}$$

And equation (4) is overall mass balance.

$$\sum_i x_{i,j} = \sum_k y_{j,k} \qquad \forall j \tag{4}$$

The value substitution for equation (4) shows as equations (4A) – (4B).

$$x_{1,1} + x_{2,1} + x_{3,1} = y_{1,1} + y_{1,2} + y_{1,3} \tag{4A}$$

$$x_{1,2} + x_{2,2} + x_{3,2} = y_{2,1} + y_{2,2} + y_{2,3} \tag{4B}$$

The constraints for transportation cost (TC), the operating in warehouse cost (OWC), the operating in plant cost (OPC) and total income (TI) are developed as equations (5) – (8), respectively.

$$TC = \sum_i \sum_j (x_{i,j} \times Transportation\ cost\ \times Distance) + \sum_j \sum_k (y_{j,k} \times Transportation\ cost\ \times Distance) \quad (5)$$

The value substitution for equation (5) shows as equations (5A).

$$TC = (90x_{1,1} + 90x_{1,2} + 180x_{2,1} + 180x_{2,2} + 270x_{3,1} + 270x_{3,2}) + (90y_{1,1} + 180y_{1,2} + 270y_{1,3} + 360y_{2,1} + 270y_{2,2} + 180y_{2,3}) \quad (5A)$$

$$OWC = \sum_i \sum_j (x_{i,j} \times Operting\ in\ warehouse\ cost) \quad (6)$$

The value substitution for equation (6) shows as equations (6A).

$$OWC = (300x_{1,1} + 300x_{2,1} + 300x_{3,1}) + (300x_{1,2} + 300x_{2,2} + 300x_{3,2}) \quad (6A)$$

$$OPC = \sum_i \sum_j (x_{i,j} \times Operting\ in\ plant\ cost) \quad (7)$$

The value substitution for equation (7) shows as equations (7A).

$$OPC = (500x_{1,1} + 500x_{2,1} + 500x_{3,1}) + (500x_{1,2} + 500x_{2,2} + 500x_{3,2}) \quad (7A)$$

$$TI = \sum_j \sum_k (y_{j,k} \times Product\ price) \quad (8)$$

The value substitution for equation (8) shows as equations (8A).

$$TI = (1{,}500y_{1,1} + 1{,}500y_{1,2} + 1{,}500y_{1,3}) + (1{,}500y_{2,1} + 1{,}500y_{2,2} + 1{,}500y_{2,3}) \quad (8A)$$

4.1.1 Results From The Developed Simplex Algorithm

The maximum profit is 272,000$ and a number of transportation is 7. The results show in Table 4.1 and Fig. 4.2.

Table 4.1 Results for small supply chain problem from developed simplex algorithm

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	100 items
x_{12}	300 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	0 items
x_{31}	100 items	y_{22}	200 items
x_{32}	100 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

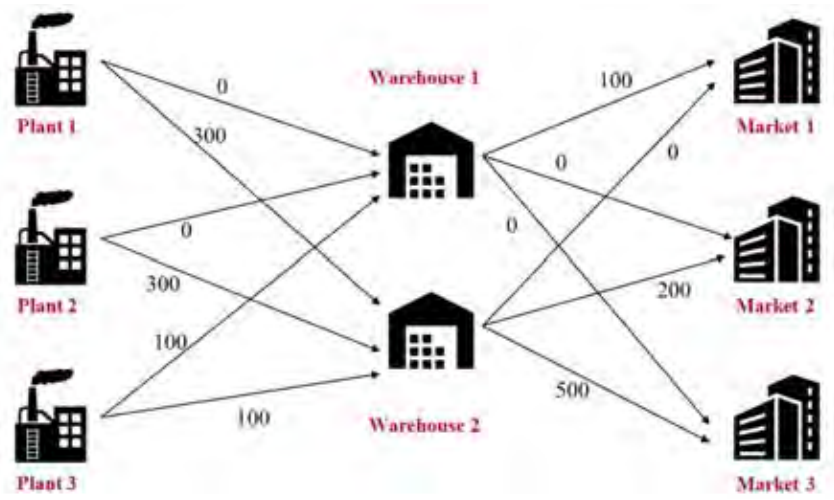


Figure 4.2 Result for supply chain problem from developed simplex algorithm.

From Table 4.1 and Fig. 4.2, x_{11} and x_{21} are zero because of low capacity of warehouse 1. Moreover, x_{11} is zero because of low transportation cost of x_{12} (limit of plant 1 = 300 items). The values of y_{12} and y_{13} are zero because of limit of warehouse 1, but y_{21} is zero because of high transportation cost (4 kmile, the longest distance).

From validating with excel solver (shows in appendix), the objective values (the profit) are the same (272,000 \$), but the mass flow values are not. It indicates that this problem is multi-solution problem. There are many solutions from the problem with only one of objective function (272,000 \$).

4.1.2 The Developed Simplex Algorithm Evaluation

From the result of the both solvers, the maximum profit is 272,000\$, this result is normal for the linear solver because normally the linear problem has one solution of the objective function. However from the mass flow value, the results are different. It shows that this problem is degenerate problem, a non-unique solution. However, only one solution is obtained from excel solver. Other solutions can be obtained from the developed algorithm.

From the section 2.1 (linear programming), the value in the first row of the non-basic variable in the final table is not zero, when the problem is not degenerate problem. If it is zero, it shows that the problem is degenerate problem. That non-basic variable would be used as entering variable, and Gauss-Jordan is used to obtain new solution. This procedure shows in Fig. 4.3. In this small supply chain, there are 3 non-basic variable as zero: x_{11}, x_{21} and y_{12}, so it indicates that this problem is degenerate problem. x_{11} is used as entering variable to continue Gauss-Jordan calculation. The new solution shows in Table 4.2 and Fig. 4.4.

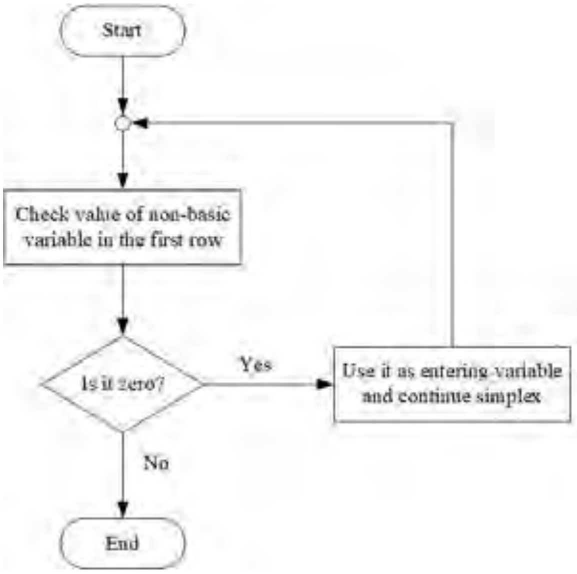


Figure 4.3 Procedure to obtain new solution from non-basic variable.

From Table 4.2 and Fig. 4.4, the profit is 272,000$ and a number of transportation is 7. x_{21} and x_{31} are zero because of high transportation cost (and very long distance) and low capacity of warehouse 1. The values of y_{12} and y_{13} are zero because of limit of warehouse 1, but y_{21} is zero because of high transportation cost (4 kmile, the longest distance).

Table 4.2 Results for small supply chain problem from non-basic variable

Variable	Value	Variable	Value
x_{11}	100 items	y_{11}	100 items
x_{12}	200 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	0 items
x_{31}	0 items	y_{22}	200 items
x_{32}	200 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

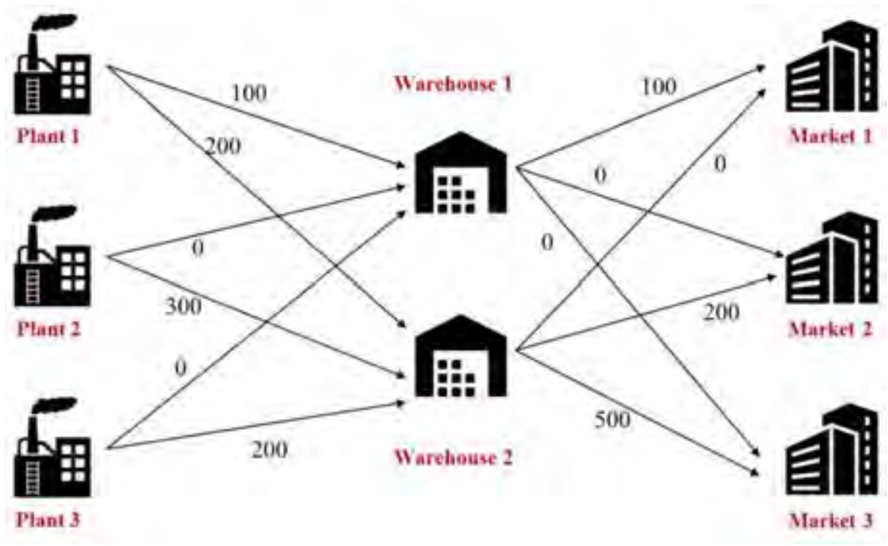


Figure 4.4 Results for small supply chain problem from non-basic variable.

From Table 4.1 and Table 4.2, x_{11} is 0 and 100 respectively, so it is probably that x_{11} is value from 0 to 100. Other solutions could be obtained by adding new constraint that is in range 0 and 100. So x_{11} is added as 50 in new constraint. The new solution shows in Table 4.3 and Fig. 4.5. The profit is 272,000$ and a number of transportation is 8.

Table 4.3 Results for small supply chain problem with x_{11} = 50

Variable	Value	Variable	Value
x_{11}	50 items	y_{11}	100 items
x_{12}	250 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	0 items
x_{31}	50 items	y_{22}	200 items
x_{32}	150 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

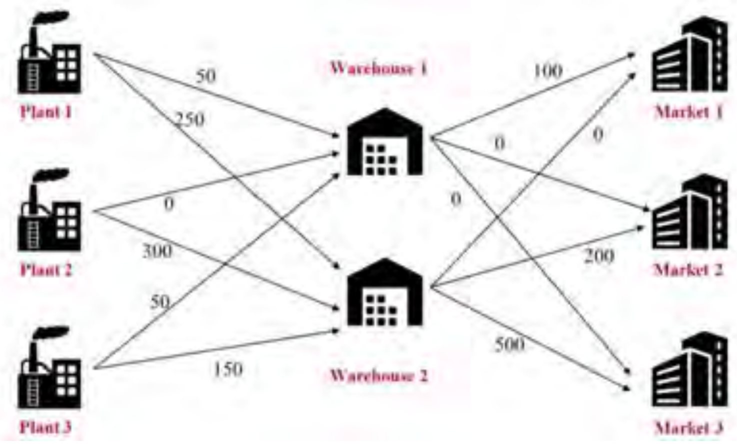


Figure 4.5 Results for small supply chain problem with x_{11} = 50.

From the section 2.1, the first step of simplex calculation is choosing the most negative value in the first row. However for the degenerate problem, the first row has many values that are the most negative value. The different value that is chosen gives different solutions. In this small supply chain problem, there are 3 the most negative variables: TC, OWC and OPC, which are -999,999. For Table 4.1, TC is chosen, so the other would be used to obtain new solution. In this case, OPC is chosen to be the entering variable and the solution is in Table 4.4 and Fig. 4.6. The profit is 272,000$ and a number of transportation is 7.

Table 4.4 Results from the different entering variable

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	100 items
x_{12}	300 items	y_{12}	0 items
x_{21}	100 items	y_{13}	0 items
x_{22}	200 items	y_{21}	0 items
x_{31}	0 items	y_{22}	200 items
x_{32}	200 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

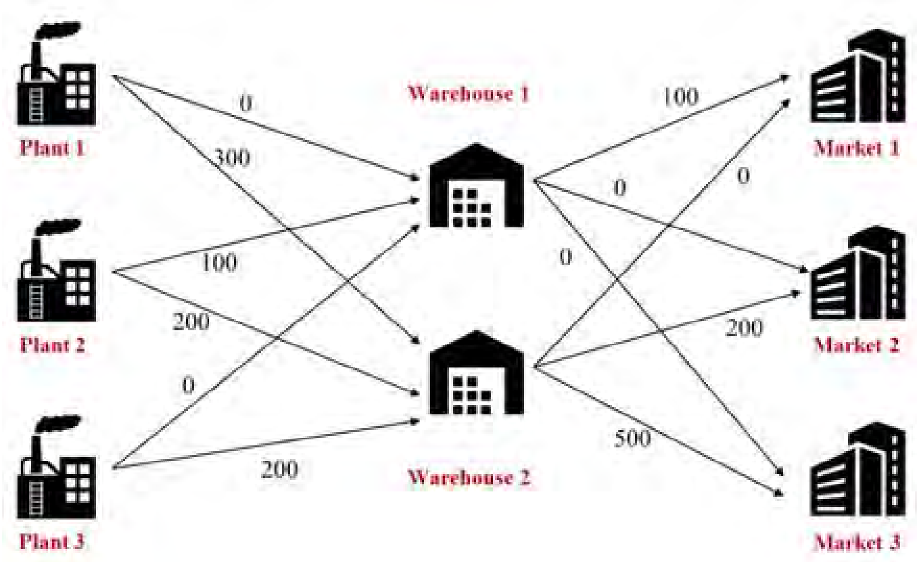


Figure 4.6 Results from the different entering variable.

Difference between Table 4.1 and Table 4.4 is value of x_{21}, x_{22}, x_{31} and x_{32}, from 0 to 100, 300 to 200,100 to 0 and 100 to 200, respectively.

Normally, some of the variables could be chosen to obtain the solution, but it would give different solution. For Microsoft excel or commercial solver, the calculation code would developed by using random function. Therefore, some of the variables would be chosen randomly. But for large problem, choosing the different variable on the first row in standard LP table would give little different objective function solution. So the code for choosing the most negative values is very important. For this developed algorithm, the code is not random function, but it is fixed to one solution benefiting from the short computation time.

The procedure of the code for choosing the most negative value in the first row shows in Fig. 4.7 as flowchart. The most negative value is checked as first. If the most negative values are more than 1, the program would remember the number of the values and the variables in case of emergency. When Gauss-Jordan calculation has finished, the feasible solution would be checked. If the solution is not feasible, the other of the most negative value would be used instead of the first one until the feasible solution is obtained.

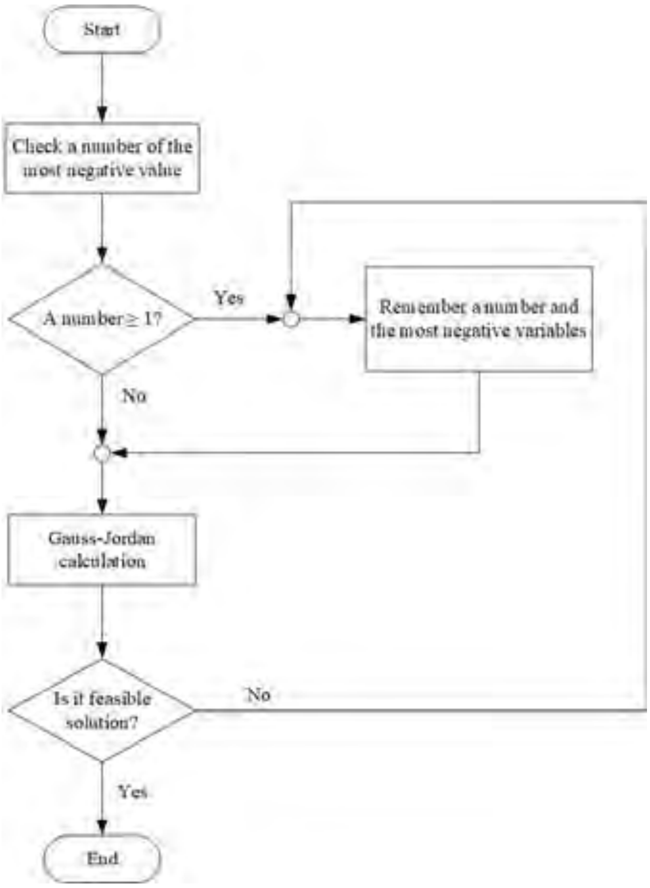


Figure 4.7 Flowchart for choosing the most negative value.

Moreover, Fig. 4.8 shows the algorithm for checking the most negative value. From Fig. 4.8, x (i, j) is specified as x value in i row and j column, i is specified as number of row, j is specified as number of column. The i element is 1, because of the first row of the standard LP table. NLV symbol is a number of the most negative value. During checking, if the program find that the most negative value is more than one value, NLV would be increase to a number of the most negative value and the program would remember the value and the variables for using in case of non-feasible solution.

However in this problem, the most negative value in the first standard table is 1,000,001, one value of the TI variable. So for the first standard table, NLV would be 1, and then is sent to calculate Gauss-Jordan correctly. There is no remembering the most negative value and the variable in this standard table. This is performance of the algorithm for checking the most negative value in the first standard table.

```
c       GOTO 106
        if (o.NE.1) goto 106
        LV=0
        NLV=1
        j=1
100     continue
               if (x(1,j).GE.0) goto 101
                 if (j.EQ.a) goto 102
                   do 103 l=j+1,a
                        if (x(1,j).GT.x(1,l)) goto 104
                           if (x(1,j).LT.x(1,l)) goto 103
                               LV = x(1,j)
                               NLV = NLV+1
103                continue
102              continue
                   do 105 l=1,b
                        if (x(1,j).GT.s(1,l)) goto 107
                           if (x(1,j).LT.s(1,l)) goto 105
                              LV = x(1,j)
                              NLV = NLV+1
105                continue
                   if (NLV.GT.1) goto 114
                   write (*,*) 'x',j,x(1,j)
                   goto 11
101     continue
        if (j.EQ.a) goto 108
        j=j+1
        goto 100
104     continue
        j=1
        LV=0
        NLV=1
        goto 100
```

Figure 4.8 The code for checking the most negative value.

But in the second standard table, the most negative value is fixed to choose the first value of all the most negative value in the first row benefiting from short computation time. The procedure shows in Fig. 4.9.

```
106     continue
        j=1
7       continue
               if (x(1,j).GE.0) goto 70
                 if (j.EQ.a) goto 8
                   do 9 l=j+1,a
                        if (x(1,j).GT.x(1,l)) goto 71
9                  continue
8                continue
                     do 10 l=1,b
                        if (x(1,j).GT.s(1,l)) goto 72
10                 continue
                   goto 11
70      continue
        if (j.EQ.a) goto 73
        j=j+1
        goto 7
71      continue
        j=1
        goto 7
```

Figure 4.9 The code for fixing the first one of the most negative values.

From Fig. 4.9, the symbol (GT) indicates that if there are many the most negative values, the first value would be chosen for Gauss-Jordan calculation. In this case, The TC variable would be chosen because it is the first value of the most negative value in the second standard table, that are TC, OWC and OPC. The result of this case shows in Table 4.1.

Moreover, the result that shows in Table 4.4 is obtained by using the algorithm shown in Fig. 4.10. The symbol (GT) is changed to be (GE) to use the last one instead of the first one. The last value of the most negative vale is chosen in the second standard table, so the different solution is obtained.

```
106 continue
    j=1
  7 continue
          if (x(1,j).GE.0) goto 70
            if (j.EQ.a) goto 8
              do 9 l=j+1,a
                if (x(1,j).GE.x(1,l)) goto 71
  9           continue
  8         continue
                do 10 l=1,b
                  if (x(1,j).GE.s(1,l)) goto 72
 10           continue
              goto 11
 70 continue
    if (j.EQ.a) goto 73
    j=j+1
    goto 7
 71 continue
    j=1
    goto 7
```

Figure 4.10 The code for fixing the last one of the most negative values.

In this small supply chain problem, the objective function solution would be the same as all of the case of the most negative value. The efficiency of the code would be show in section 4.3.

Moreover in Table 4.1 and Table 4.4, x_{21} is 0 and 100, respectively. So other solutions can be obtained by adding x_{21} as new constraint between 0 and 100. In this case, x_{21} is added as 50. The solution shows in Table 4.5 and Fig. 4.11. The profit is 272,000$ and a number of transportation is 8.

Table 4.5 Results of small supply chain with x_{21}= 50

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	100 items
x_{12}	300 items	y_{12}	0 items
x_{21}	50 items	y_{13}	0 items
x_{22}	250 items	y_{21}	0 items
x_{31}	50 items	y_{22}	200 items
x_{32}	150 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

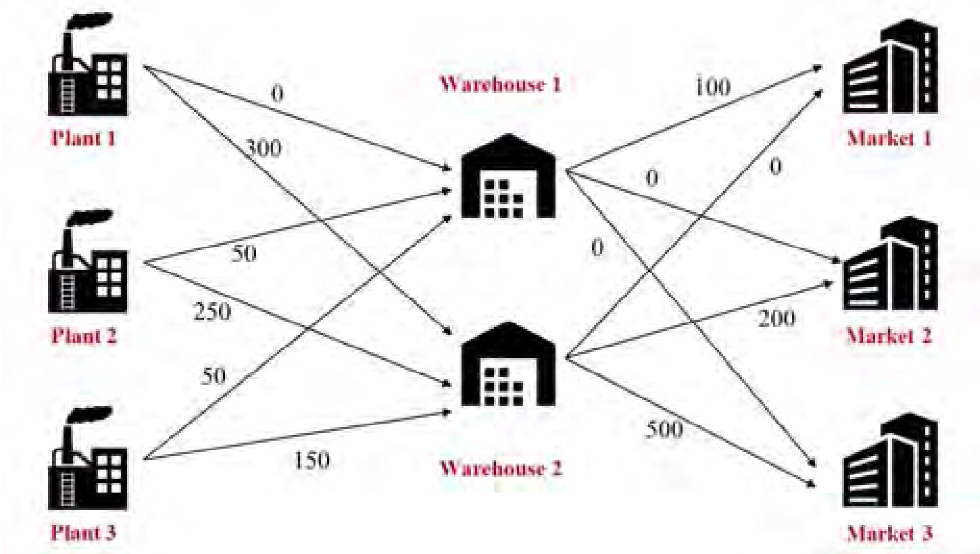


Figure 4.11 Results for small supply chain problem with x_{21} = 50.

Finally, the same solutions from 2 solvers could be obtained by fixing some variables. For this case, $x_{1,1}$ is added in now constraint as 100. The results show in Table 4.6 and Fig. 4.12.

Table 4.6 Results validated with the algorithm by fixing $x_{1,1}$ = 100

Variable	Value	Variable	Value
x_{11}	100 items	y_{11}	100 items
x_{12}	200 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	0 items
x_{31}	0 items	y_{22}	200 items
x_{32}	200 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

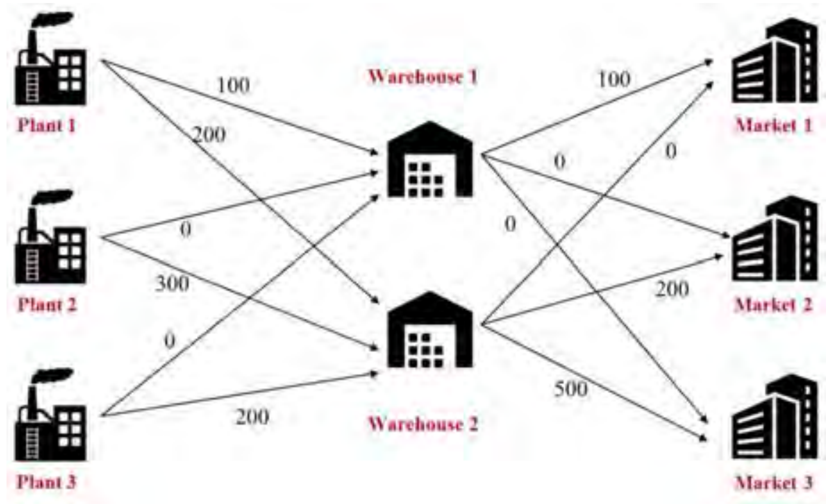


Figure 4.12 Results validated with the algorithm by fixing $x_{1,1}$ = 100.

These show only continuous solutions of the small supply chain. The MILP form shows in section 4.2.1.

4.2 Branch-And-Bound Algorithm Evaluation

For branch-and-bound algorithm, the BIP algorithm and the MILP algorithm are developed in the same code (branch-and-bound algorithm code). Each of the both codes would be used dependent on setting variable type. If the variable is set as binary, BIP code would execute, but if the variable is set as integer, MILP code would execute, as follow by Fig. 3.3.

4.2.1 BIP Algorithm Evaluation

In section 4.1, only continuous solutions are shown. In this section, the transportation number is added as Integer part of the small supply chain. However, the cost would be calculated after optimization of transportation number. The objective function and constraints show below.

The objective function is specified as a number of transportation, as show below.

$$Minimize \quad Total\ transportation = \sum_i \sum_j Zx_{i,j} + \sum_j \sum_k Zy_{j,k}$$

Zx is transportation from plant to warehouse, and Zy is transportation from warehouse to market. The both are binary.

The constraints would be divided to mass balance and transportation part. The mass balance constraints are taken around every node and around the system (overall mass balance). Equation (1) is production limit for each plant, limiting with amounts of suppliers.

$$\textstyle\sum_j x_{i,j} \leq Supplier^i \qquad \forall i \tag{1}$$

The value substitution for equation (1) shows as equations (1A) – (1C).

$$x_{1,1} + x_{1,2} \leq 300 \tag{1A}$$

$$x_{2,1} + x_{2,2} \leq 300 \tag{1B}$$

$$x_{3,1} + x_{3,2} \leq 200 \tag{1C}$$

Equation (2) is minimum demand for each market.

$$\sum_j y_{j,k} \geq Demand^k \qquad \forall k \tag{2}$$

The value substitution for equation (2) shows as equations (2A) – (2C).

$$y_{1,1} + y_{2,1} \geq 100 \tag{2A}$$

$$y_{1,2} + y_{2,2} \geq 200 \tag{2B}$$

$$y_{1,3} + y_{2,3} \geq 200 \tag{2C}$$

The capacity of the warehouse is limited by equation (3).

$$\sum_i x_{i,j} \leq Inventory^j \qquad \forall j \tag{3}$$

The value substitution for equation (3) shows as equations (3A) – (3B).

$$x_{1,1} + x_{2,1} + x_{3,1} \leq 100 \tag{3A}$$

$$x_{1,2} + x_{2,2} + x_{3,2} \leq 800 \tag{3B}$$

And equation (4) is overall mass balance.

$$\sum_i x_{i,j} = \sum_k y_{j,k} \qquad \forall j \tag{4}$$

The value substitution for equation (4) shows as equations (4A) – (4B).

$$x_{1,1} + x_{2,1} + x_{3,1} = y_{1,1} + y_{1,2} + y_{1,3} \tag{4A}$$

$$x_{1,2} + x_{2,2} + x_{3,2} = y_{2,1} + y_{2,2} + y_{2,3} \tag{4B}$$

The transportation from plant to warehouse is equation (5).

$$x_{i,j} \leq MZx_{i,j} \qquad \forall i, j \tag{5}$$

The value substitution for equation (5) shows as equations (5A) – (5F).

$$x_{1,1} \leq 1{,}000Zx_{1,1} \tag{5A}$$

$$x_{1,2} \leq 1{,}000Zx_{1,2} \tag{5B}$$

$$x_{2,1} \leq 1{,}000Zx_{2,1} \tag{5C}$$

$$x_{2,2} \leq 1{,}000Zx_{2,2} \tag{5D}$$

$$x_{3,1} \leq 1{,}000Zx_{3,1} \tag{5E}$$

$$x_{3,2} \leq 1{,}000Zx_{3,2} \tag{5F}$$

The transportation from warehouse to market is equation (6).

$$y_{j,k} \leq MZy_{j,k} \qquad \forall j, k \tag{6}$$

The value substitution for equation (6) shows as equations (6A) – (6F).

$$y_{1,1} \leq 1{,}000Zy_{1,1} \tag{6A}$$

$$y_{1,2} \leq 1{,}000Zy_{1,2} \tag{6B}$$

$$y_{1,3} \leq 1{,}000Zy_{1,3} \tag{6C}$$

$$y_{2,1} \leq 1{,}000Zy_{2,1} \tag{6D}$$

$$y_{2,2} \leq 1{,}000Zy_{2,2} \tag{6E}$$

$$y_{2,3} \leq 1{,}000Zy_{2,3} \tag{6F}$$

4.2.1.1 Results Of Small Supply Chain With BIP

From the small supply chain with BIP, a number of transportation is 5 and the profit is 195,000$. The results show in Table 4.7 and Fig. 4.13.

Table 4.7 Results of small supply chain with BIP

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	0 items
x_{12}	300 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	100 items
x_{31}	0 items	y_{22}	200 items
x_{32}	0 items	y_{23}	300 items
Zx_{11}	0	Zy_{11}	0
Zx_{12}	1	Zy_{12}	0
Zx_{13}	0	Zy_{13}	0
Zx_{14}	1	Zy_{14}	1
Zx_{15}	0	Zy_{15}	1
Zx_{16}	0	Zy_{16}	1

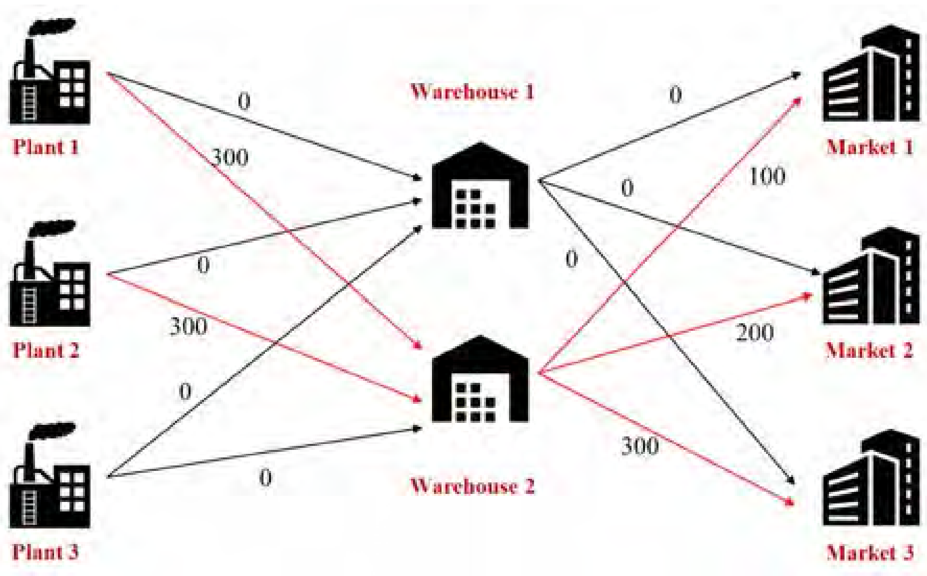


Figure 4.13 Results of small supply chain with BIP.

From Table 4.7, a number of transportation is fewer than that from Table 4.1 because in this section the minimum of a transportation number is focused. Moreover, the profit in this section is the less also. The transportation lines that are chosen, are Zx_{12}, Zx_{14}, Zy_{14}, Zy_{15} and Zy_{16} with 300, 300, 100, 200 and 300, respectively. However in validating with excel solver, the solutions are not similar (with the same transportation number). Therefore, it indicates that this problem is degenerate problem.

4.2.1.2 Other Solutions Of Small Supply Chain With BIP

With being degenerate problem, the non-basic variable that is zero in the first row in the final LP table, is used as entering variable to obtain new solution. The solution shows in Table 4.8 and Fig. 4.14. A number of transportation is 5 and the profit is 186,000$.

Table 4.8 Results of small supply chain with BIP and non-basic variable

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	0 items
x_{12}	300 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	100 items
x_{31}	0 items	y_{22}	300 items
x_{32}	0 items	y_{23}	200 items
Zx_{11}	0	Zy_{11}	0
Zx_{12}	1	Zy_{12}	0
Zx_{13}	0	Zy_{13}	0
Zx_{14}	1	Zy_{14}	1
Zx_{15}	0	Zy_{15}	1
Zx_{16}	0	Zy_{16}	1

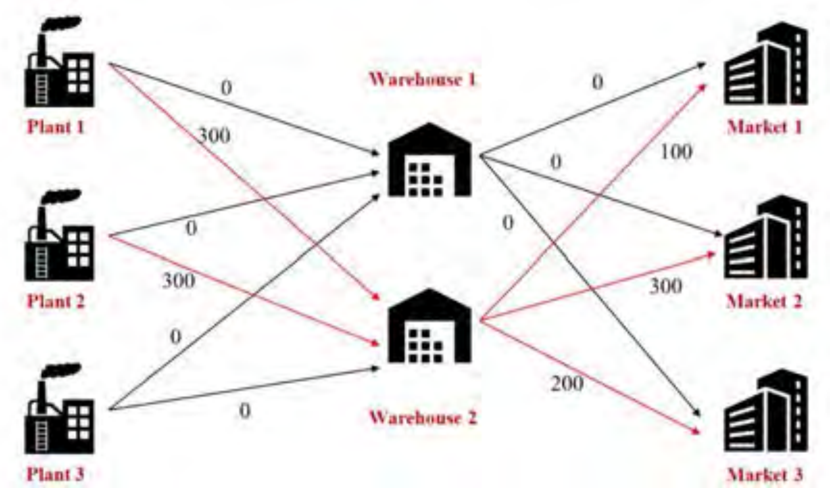


Figure 4.14 Results of small supply chain with BIP and non-basic variable.

Moreover from Table 4.7 and Table 4.8, y_{22} is 200 and 300 respectively. It is possible that y_{22} is in range 200 and 300. So y_{22} is added in new constraint as 250 to obtain other solution of this problem. The solution shows in Table 4.9 and Fig. 4.15. A number of transportation is 5 and the profit is 190,500$.

Table 4.9 Results of small supply chain with BIP and y_{22} = 250

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	0 items
x_{12}	300 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	100 items
x_{31}	0 items	y_{22}	250 items
x_{32}	0 items	y_{23}	250 items
Zx_{11}	0	Zy_{11}	0
Zx_{12}	1	Zy_{12}	0
Zx_{13}	0	Zy_{13}	0
Zx_{14}	1	Zy_{14}	1
Zx_{15}	0	Zy_{15}	1
Zx_{16}	0	Zy_{16}	1

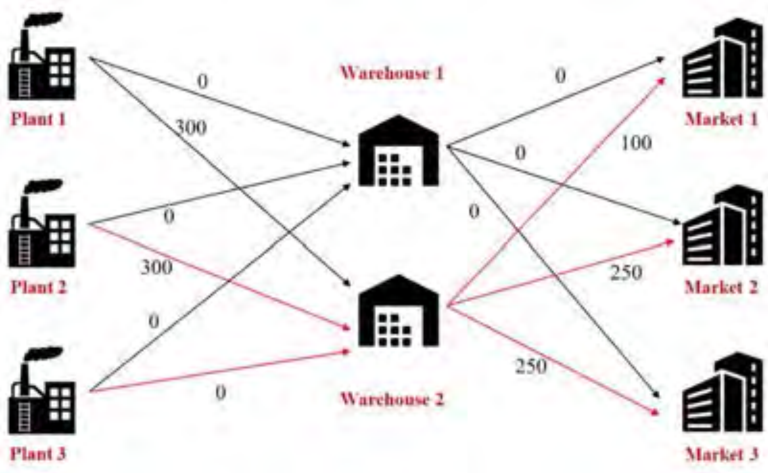


Figure 4.15 Results of small supply chain with BIP and y_{22} = 250.

Finally, the same solutions from 2 solvers could be obtained by fixing some variables. For this case, $y_{22}$is added in now constraint as 250. The results show in Table 4.10 and Fig. 4.16.

Table 4.10 Results of small supply chain with BIP and x_{12} = 200

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	0 items
x_{12}	200 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	100 items
x_{31}	0 items	y_{22}	200 items
x_{32}	0 items	y_{23}	200 items
Zx_{11}	0	Zy_{11}	0
Zx_{12}	1	Zy_{12}	0
Zx_{13}	0	Zy_{13}	0
Zx_{14}	1	Zy_{14}	1
Zx_{15}	0	Zy_{15}	1
Zx_{16}	0	Zy_{16}	1

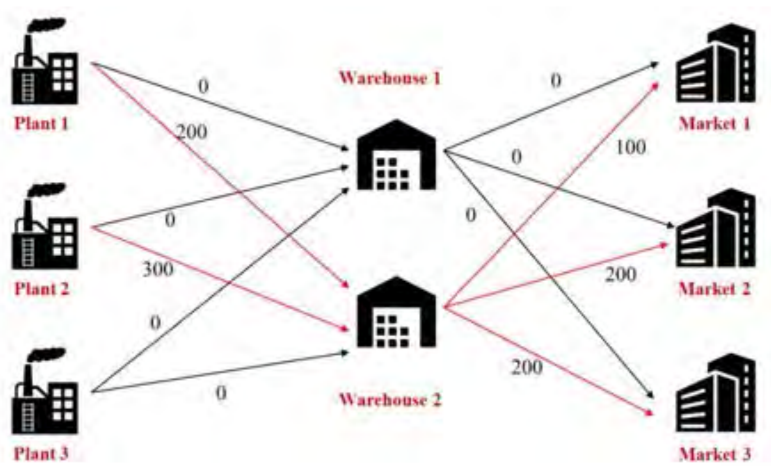


Figure 4.16 Results of small supply chain with BIP and x_{12} = 200.

4.2.2 MILP Algorithm Evaluation

To evaluate the algorithm, the project problem is section 3.2.2 would be found the maximum the total net present value. The result would be compared with Microsoft excel.

From Table 3.1, the data of the problem show in this table. The mathematical model would be simulated for using in the solver. For the problem, the objective function is specified as summation of every net present value. x_i is identified to be decision of project i, 1 or 0. If the project i is chosen, x_i is 1, or not, x_i is 0. The mathematical model would be simulated as below.

$$Maximize \qquad Total\ net\ present\ value = \sum_i (net\ present\ value) \times x_i$$

The objective function is substituted by adding the value below.

$$Max\ TNPV = 100{,}000x_1 + 150{,}000x_2 + 35{,}000x_3 + 75{,}000x_4 + 125{,}000x_5 + 60{,}000x_6$$

The constraints are developed involving with each year expenditure and engineering hours and resource limitations. The equations (1) – (3) refer to the first year expenditure, second year expenditure and engineering hours, respectively.

$$\textstyle\sum_i First\ year\ expenditure \times x_i \leq Total\ first\ year\ expenditure \qquad (1)$$

The value substitution for equation (1) shows as equations (1A).

$$300{,}000x_1 + 100{,}000x_2 + 50{,}000x_4 + 50{,}000x_5 + 100{,}000x_6 \leq 450{,}000 \qquad (1A)$$

$$\textstyle\sum_i Second\ year\ expenditure \times x_i \leq Total\ second\ year\ expenditure \qquad (2)$$

The value substitution for equation (2) shows as equations (2A).

$$300{,}000x_2 + 200{,}000x_3 + 100{,}000x_4 + 300{,}000x_5 + \ 200{,}000x_6 \leq 400{,}000 \quad \text{(2A)}$$

$$\textstyle\sum_i (Engineering\ hours) \times x_i \leq Total\ engineering\ hours \quad (3)$$

The value substitution for equation (3) shows as equations (3A).

$$4{,}000x_1 + 7{,}000x_2 + 2{,}000x_3 + 6{,}000x_4 + 3{,}000x_5 + \ 600x_6 \leq 10{,}000 \quad \text{(3A)}$$

The equation (4) shows that one of the project 1 and 2 is selected.

$$x_1 + x_2 \leq 1 \quad (4)$$

The project 3 would be not used, if the project 2 is not used, it refers to the equation (5). The project 5 and 6 could be used only one of its (the equation (6)).

$$x_2 - x_3 \geq 0 \quad (5)$$

$$x_5 + x_6 \leq 1 \quad (6)$$

The problem would be solve by using Microsoft excel and the developed branch-and-bound algorithm. The result of both solvers would be discussed and compared.

4.2.2.1 Result From MILP Algorithm Evaluation

The maximum net present value is optimized to 225,000$. And the results show in Table 4.11.

Table 4.11 Result of the project problem from MILP algorithm

Variable	Value	Variable	Value
x_1	1	x_4	0
x_2	0	x_5	1
x_3	0	x_6	0

From Table 4.11, the project 1 and 5 are selected. The result shows that the final solution is the same as Microsoft excel. This result indicates that the problem is possible to be general-solution problem (not multi-solution problem).

4.2.2.2 MILP Algorithm Evaluation

For MILP algorithm, fathoming step is used for 3 cases:

Case 1: Its bound $\leq Z^*$, where Z^* is Z from the current incumbent,

Case 2: The subproblem has no feasible solution,

Case 3: All of the integer-restricted variables are integer value.

This step of branch-and-bound technique helps to decrease a number of subproblem to reduce executing time in finding the optimal point. This fathoming step is developed on the algorithm following the flowchart in Fig. 4.17.

From flowchart in Fig. 4.17, after the solution is obtained from MILP and LP, the solution would be checked the artificial variable values because in the feasible solutions the artificial variable values are zero in the solution. If it is zero, it transfers to the next step, if not; it indicates that it is not the feasible solution. In case of feasibility, the solution would be substituted to the objective function. If the optimal value from the executing and one from substituting are the same, the optimal solution is correct. For the correct optimal solution, it would transfer to the next step. But in case of non-feasible solution, the subproblem stops. When the solution is feasible solution, it would be checked the integer value. If all integer-restricted variables are integer, the subproblem stops, but not; it transfers to the next step. The optimal solution would be compared with the current incumbent. If it is greater than the current incumbent, the node branching would continue, if not; the subproblem stops.

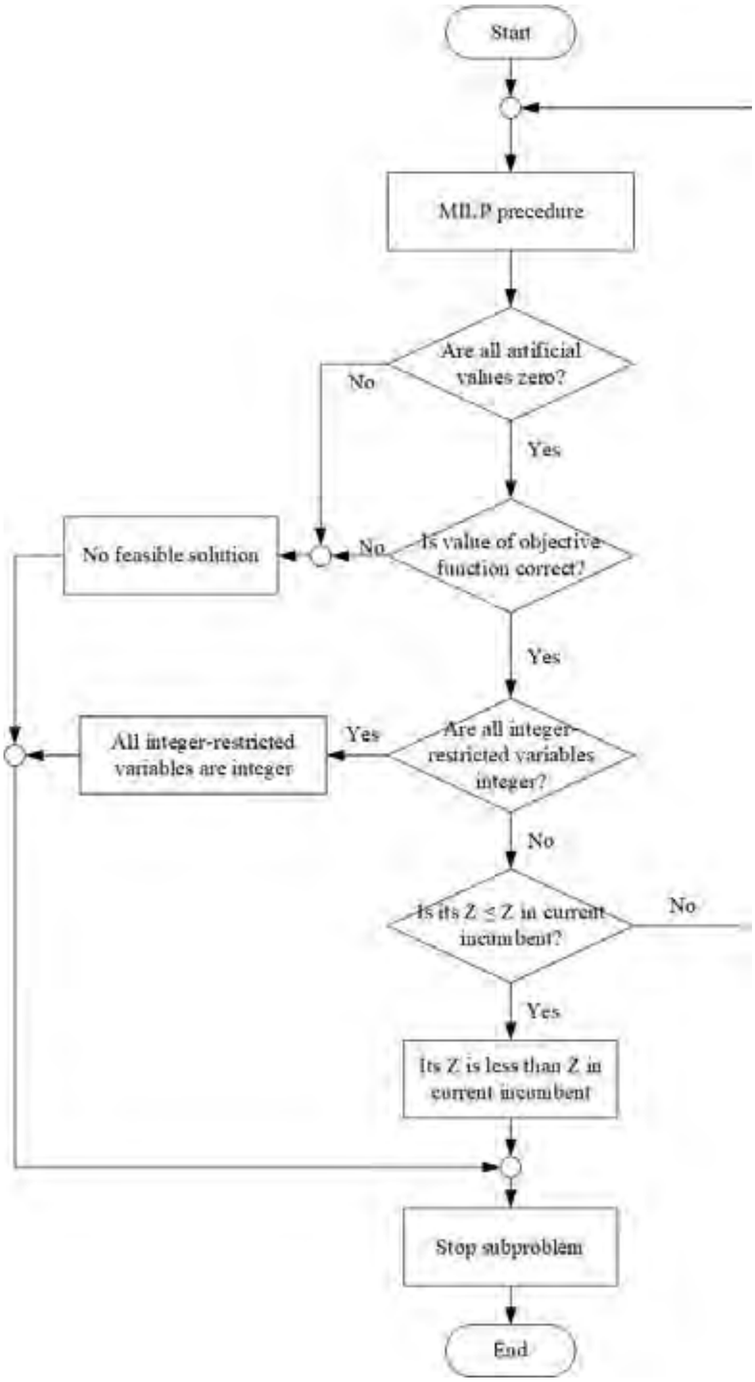


Figure 4.17 Flowchart for fathoming step in branch-and-bound algorithm.

With these branch-and-bound and fathoming steps, this project problem shows only 4 subproblems. The 4 supproblems show in Fig. 4.18.

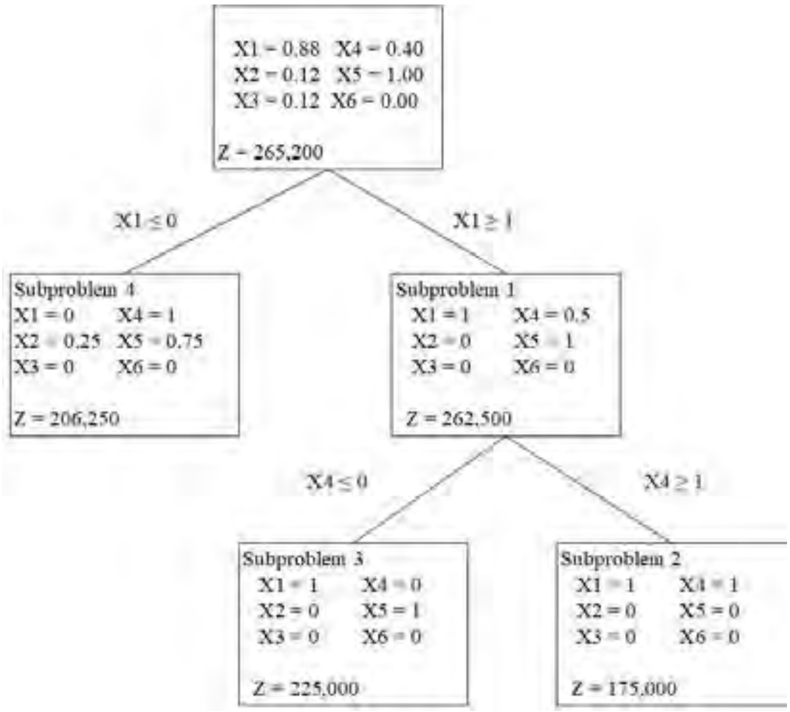


Figure 4.18 Branch-and-bound step of the developed branch-and-bound algorithm.

From Fig. 4.18, the supproblems 2 and 3 stop branching because all variables are integer, and subproblem 4 stops because Its bound $\leq Z^{*}$(Z from current incumbent), when Z^{*} = 225,000 (subproblem 3). In this case, 4 subproblems are obtained from Microsoft excel the same, as show in Fig 4.19.

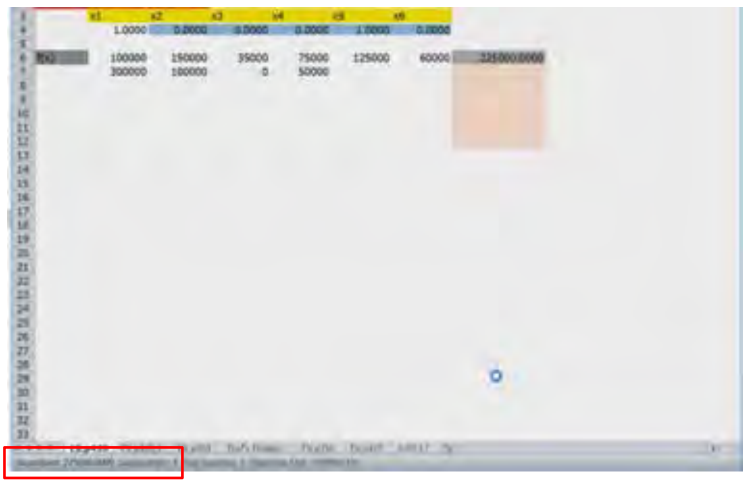


Figure 4.19 Subproblem from Microsoft excel in branch-and-bound evaluation.

From this comparison, it shows that the fathoming technique makes the algorithm to be good efficiency in finding an optimum point for MILP case with the same optimum value.

4.3 Biofuel Production Supply Chain Case Study

The biofuel production supply chain in the United Kingdom case study, descripted in section 3.3, is used to evaluate the developed algorithm comparing with Microsoft excel.

The algorithm evaluation for biofuel production supply chain case study is divided to 3 parts;

Part 1: Integer restricted variables are evaluated in integer case

Part 2: Binary restricted variables are evaluated in binary case

Part 3: Both integer and binary restricted variables are evaluated together.

The total annual gross profit (TNGP) would be considered as objective function. But the total supply chain net present value (SCNPV) would be considered as the target for this case study. So the total annual gross profit that is obtained from the solvers would be used to calculate the total supply chain net present value with 10 years in operating time.

The model of the biofuel production supply chain shows in Fig. 3.5 in section 3.3. That model is specified the mass flow variables shown in Fig. 4.20.

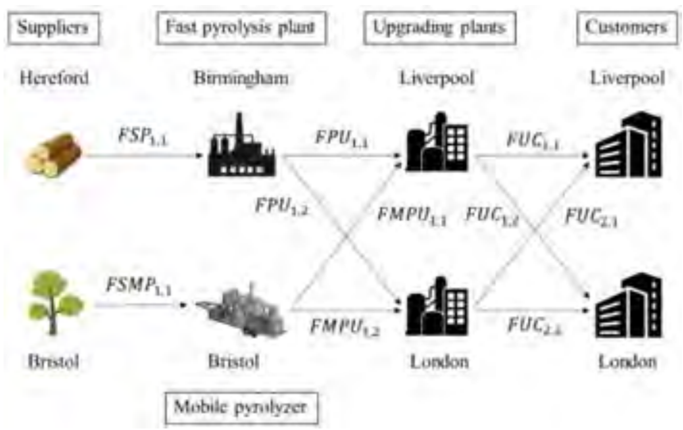


Figure 4.20 The mass flow variables biofuel production supply chain.

All of equations for objective function and constraints are descripted below. Total annual gross profit (TAGP) is considered as the objective function of the problem showing in equation (1).

$$TAGP = \sum_{fp,u,mp} vp^{fp,u,mp} - \sum_{fp,u,mp} TPC^{fp,u,mp} - df \times \sum_{fp,u,mp} DP^{fp,u,mp} \quad (1)$$

The value substitution for the objective function shows in equation (1A).

$$TAGP = (vp^{fp} + vp^{u1} + vp^{u2} + vp^{mp}) - (TPC^{fp} + TPC^{u1} + TPC^{u2} + TPC^{mp})$$
$$-0.1 \times (DP^{fp} + DP^{u1} + DP^{u2} + DP^{mp}) \quad (1A)$$

TAGP is the total annual gross profit. The product price, the total production cost and the depreciation are identified as vp, TPC and DP, respectively. The fast pyrolysis plant, the upgrading plant 1 and 2, and the mobile pyrolyzer are identified as fp, u1, u2 and mp, respectively.

The constraints would be divided to mass balance, fast pyrolysis and upgrading plant costs, operating costs, transportation costs and mobile pyrolysis costs. The equations (2) – (4) are taken as mass balance for fast pyrolysis plant, upgrading plants and mobile pyrolyzer, respectively. Each equation is multiplied by conversions (cf) that shows in Table 4.12.

$$\sum_u FPU_{fp,u} = FSP_{s,fp} \times cf_{b,o} \quad (2)$$

$$\sum_c FUC_{u,c} = (FPU_{fp,u} + FMPU_{mp,u}) \times cf_{o,gd} \qquad \forall u \quad (3)$$

$$\sum_u FMPU_{mp,u} = FSMP_{s,mp} \times cf_{b,o} \quad (4)$$

Table 4.12 Conversion factor of the reactions

Description	Conversion factor (cf)	Units
Pyrolysis oil on dry biomass for fast pyrolysis plant	0.768	$Kg\ Kg^{-1}$
Gasoline and diesel on pyrolysis oil	0.528	$dm^3\ Kg^{-1}$
Pyrolysis oil on dry biomass for mobile pyrolyzer	0.583	$Kg\ Kg^{-1}$

The value substitution for mass balance in the fast pyrolysis plant shows as equations (2A).

$$FPU_{1,1} + FPU_{1,2} = 0.768 \times FSP_{1,1} \tag{2A}$$

The value substitution for mass balance in the upgrading plants 1 and 2 shows as equations (3A) – (3B), respectively.

$$FUC_{1,1} + FUC_{1,2} = 0.528 \times (FPU_{1,1} + FMPU_{1,1}) \tag{3A}$$

$$FUC_{2,1} + FUC_{2,2} = 0.528 \times (FPU_{1,2} + FMPU_{1,2}) \tag{3B}$$

The value substitution for mass balance in the mobile pyrolyzer shows as equations (4A).

$$FMPU_{1,1} + FMPU_{1,2} = 0.583 \times FSMP_{1,1} \tag{4A}$$

The equation (5) is mass balance for demands (both customers). The biofuel demands are 80.46 $dam^3\ y^{-1}$ in Liverpool and 180.54 $dam^3\ y^{-1}$ in London.

$$\textstyle\sum_u FUC_{u,c} = D_c \qquad \forall c \tag{5}$$

The value substitution for demand of the both customers shows as equations (5A) – (5B).

$$FUC_{1,1} + FUC_{2,1} = 80.46 \tag{5A}$$

$$FUC_{1,2} + FUC_{2,2} = 180.54 \tag{5B}$$

The equations (6) – (7) are limitations of supplier type 1 and 2. The suppliers are 1,500 $Mg\ d^{-1}$ in supplier 1 (Hereford) and 900 $Mg\ d^{-1}$ in supplier 2 (Bristol).

$$FSP_{s,fp} \leq AV_s^{st1} \tag{6}$$

$$FSMP_{s,mp} \leq AV_s^{st2} \tag{7}$$

The value substitution for limitation for the both suppliers shows as equations (6A) – (7A).

$$FSP_{1,1} \leq 1{,}500 \tag{6A}$$

$$FSMP_{1,1} \leq 900 \tag{7A}$$

For the fast pyrolysis and upgrading plant cost constraints, the plants are divided to small scale and large scale. Each scale is taken as linear equation by using piece-wise linearization of equipment cost.

The equations (8) – (9) are considered as linear equation of equipment cost for fast pyrolysis plants and upgrading plants, respectively. The first term refers to small scale, 200 – 850 $Mg\ d^{-1}$ of dry biomass, and the second term refers to large scale, 850 – 2,000 $Mg\ d^{-1}$ of dry biomass. The slope and intersection values show in equations (8A) – (9B).

$$EC^{fp} \geq \sum_u FPU1_{fp,u} \times SLP1_{fp} + INP1_{fp} \times y1^{fp} + \sum_u FPU2_{fp,u} \times SLP2_{fp} + INP2_{fp} \times y2^{fp} \quad (8)$$

For fast pyrolysis plant:

y = 0.0189x + 2.4721 $\quad 200 \leq x \leq 850$ (8A)

y = 0.0137x + 5.8201 $\quad 850 \leq x \leq 2{,}000$ (8B)

The value substitution for equipment cost for the fast pyrolysis plant shows as equation (8C).

$$EC^{fp} \geq 0.0189(FPU1_{1,1} + FPU1_{1,2}) + 2.4721 \times y1^{fp} + 0.0137(FPU2_{1,1} + FPU2_{1,2}) + 5.8201 \times y2^{fp} \quad (8C)$$

$$EC^{u} \geq \sum_c FUC1_{u,c} \times SLU1_{u} + INU1_{u} \times y1^{u} + \sum_c FUC2_{u,c} \times SLU2_{u} + INU2_{u} \times y2^{u} \quad \forall u \quad (9)$$

For upgrading plant:

y = 0.0639x + 18.029 $\quad 200 \leq x \leq 850$ (9A)

y = 0.0468x + 23.938 $\quad 850 \leq x \leq 2{,}000$ (9B)

The value substitution for equipment cost for the upgrading plants 1 and 2 shows as equation (9C) – (9D).

$$EC^{u1} \geq 0.0639(FUC1_{1,1} + FUC1_{1,2}) + 18.029 \times y1^{u1} + 0.0468(FUC2_{1,1} + FUC2_{1,2}) + 23.938 \times y2^{u1} \quad (9C)$$

$$EC^{u2} \geq 0.0639(FUC1_{1,1} + FUC1_{1,2}) + 18.029 \times y1^{u2} + 0.0468(FUC2_{1,1} + FUC2_{1,2}) + 23.938 \times y2^{u2} \quad (9D)$$

The equations (10) – (11) show that the plants are divided to small scale and large scale. And one of two scales would be chosen for each plant with equation (12).

$$FPU_{fp,u} = FPU1_{fp,u} + FPU2_{fp,u} \qquad \forall u \tag{10}$$

$$FUC_{u,c} = FUC1_{u,c} + FUC2_{u,c} \qquad \forall u, c \tag{11}$$

$$y1^{fp,u} + y2^{fp,u} = 1 \qquad \forall u \tag{12}$$

The value substitution for scale separation from fast pyrolysis plant to the both upgrading plants shows as equations (10A) – (10B).

$$FPU_{1,1} = FPU1_{1,1} + FPU2_{1,1} \tag{10A}$$

$$FPU_{1,2} = FPU1_{1,2} + FPU2_{1,2} \tag{10B}$$

The value substitution for scale separation from the both upgrading plants to the customers 1 and 2 shows as equations (11A) – (11D).

$$FUC_{1,1} = FUC1_{1,1} + FUC2_{1,1} \tag{11A}$$

$$FUC_{1,2} = FUC1_{1,2} + FUC2_{1,2} \tag{11B}$$

$$FUC_{2,1} = FUC1_{2,1} + FUC2_{2,1} \tag{11C}$$

$$FUC_{2,2} = FUC1_{2,2} + FUC2_{2,2} \tag{11D}$$

The value substitution for choosing one of the two scales for the fast pyrolysis plant and the both upgrading plants are expressed as equations (12A) – (12C).

$$y1^{fp} + y2^{fp} = 1 \tag{12A}$$

$$y1^{u1} + y2^{u1} = 1 \tag{12B}$$

$$y1^{u2} + y2^{u2} = 1 \tag{12C}$$

The equations (13) – (20) are considered as upper limit and lower limit for each plant scale. Equations (13) – (14) are upper limit of mass flow rate for small scale of fast pyrolysis plant and upgrading plants, respectively.

$$\sum_u FPU1_{fp,u} \leq SCLP \times y1^{fp} \tag{13}$$

$$\sum_c FUC1_{u,c} \quad \leq SCLU \times y1^{u} \qquad \forall u \tag{14}$$

The value substitution for the upper limits for small scale of the fast pyrolysis plant and the both upgrading plants shows as equations (13A) – (14B).

$$FPU1_{1,1} + FPU1_{1,2} \leq 850 \times y1^{fp} \tag{13A}$$

$$FUC1_{1,1} + FUC1_{1,2} \leq 850 \times y1^{u1} \tag{14A}$$

$$FUC1_{2,1} + FUC1_{2,2} \leq 850 \times y1^{u2} \tag{14B}$$

Equations (15) – (16) are lower limit for large scale of fast pyrolysis plant and upgrading plants.

$$\sum_u FPU2_{fp,u} \geq SCLP \times y2^{fp} \tag{15}$$

$$\sum_c FUC2_{u,c} \quad \geq SCLU \times y2^{u} \qquad \forall u \tag{16}$$

The value substitution for the lower limits for large scale of the fast pyrolysis plant and the both upgrading plants shows as equations (15A) – (16B).

$$FPU2_{1,1} + FPU2_{1,2} \geq 850 \times y2^{fp} \tag{15A}$$

$$FUC2_{1,1} + FUC2_{1,2} \geq 850 \times y2^{u1} \tag{16A}$$

$$FUC2_{2,1} + FUC2_{2,2} \geq 850 \times y2^{u2} \tag{16B}$$

Equations (17) – (18) are lower limit for small scale of fast pyrolysis plant and upgrading plants.

$$\sum_u FPU1_{fp,u} \geq mSCLP \times y1^{fp} \tag{17}$$

$$\sum_c FUC1_{u,c} \quad \geq mSCLU \times y1^{u} \qquad \forall u \tag{18}$$

The value substitution for the lower limits for small scale of the fast pyrolysis plant and the both upgrading plants shows as equations (17A) – (18B).

$$FPU1_{1,1} + FPU1_{1,2} \geq 200 \times y1^{fp} \tag{17A}$$

$$FUC1_{1,1} + FUC1_{1,2} \geq 200 \times y1^{u1} \tag{18A}$$

$$FUC1_{2,1} + FUC1_{2,2} \geq 200 \times y1^{u2} \tag{18B}$$

Finally, equations (19) – (20) are upper limit for small scale of fast pyrolysis plant and upgrading plants.

$$\sum_u FPU2_{fp,u} \leq maxSCLP \times y2^{fp} \tag{19}$$

$$\sum_c FUC2_{u,c} \quad \leq maxSCLU \times y2^{u} \qquad \forall u \tag{20}$$

The value substitution for the upper limits for large scale of the fast pyrolysis plant and the both upgrading plants shows as equations (19A) – (20B).

$$FPU2_{1,1} + FPU2_{1,2} \leq 2{,}000 \times y2^{fp} \tag{19A}$$

$$FUC2_{1,1} + FUC2_{1,2} \leq 2{,}000 \times y2^{u1} \tag{20A}$$

$$FUC2_{2,1} + FUC2_{2,2} \leq 2{,}000 \times y2^{u2} \tag{20B}$$

The depreciation (DP) is functions of the equipment cost, expressed as equation (21).

$$DP^{fp,u,mp} = df^{fp,u,mp} \times EC^{fp,u,mp} \qquad \forall u \tag{21}$$

The value substitution for the depreciation costs for the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (21A) – (21D).

$$DP^{fp} = 0.1 \times EC^{fp} \tag{21A}$$

$$DP^{u1} = 0.1 \times EC^{u1} \tag{21B}$$

$$DP^{u2} = 0.1 \times EC^{u2} \tag{21C}$$

$$DP^{mp} = 0.1 \times EC^{mp} \tag{21D}$$

For operating cost constraints, the values of products (vp) are dependent on the product selling price (sp), expressed as equations (22) – (24). For the selling prices from fast pyrolysis plant, mobile pyrolyzer and upgrading plants, biochar selling price is 40.8 \$ Mg^{-1}, and biofuel selling price is 1,541 \$ m^{-3}.

$$vp^{fp} = sp^{fp} * \sum_u FPU_{fp,u} \tag{22}$$

$$vp^{mp} = sp^{mp} * \sum_u FMPU_{mp,u} \tag{23}$$

$$vp^{u} = sp^{u} * \sum_c FUC_{u,c} \qquad \forall u \tag{24}$$

The value substitution for the product prices for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (22A) – (24B).

$$vp^{fp} = 40.8 \times (FPU_{1,1} + FPU_{1,2}) \tag{22A}$$

$$vp^{mp} = 40.8 \times (FMPU_{1,1} + FMPU_{1,2}) \tag{23A}$$

$$vp^{u1} = 1{,}541 \times (FUC_{1,1} + FUC_{1,2}) \tag{24A}$$

$$vp^{u2} = 1{,}541 \times (FUC_{2,1} + FUC_{2,2}) \tag{24B}$$

The costs of raw materials (CRM), equations (25) – (27), depend on the raw material cost (RMC) and biomass cost (bC). The feedstock cost of mobile pyrolyzer is 25$ Mg^{-1}. And the costs of feedstock for the fast pyrolysis plants and upgrading plants show in Table 4.13.

$$CRM^{fp} = (RMC^{fp} + bC) \times \sum_u FPU_{fp,u} \tag{25}$$

$$CRM^{u} = RMC^{u} \times \sum_c FUC_{u,c} \qquad \forall u \tag{26}$$

$$CRM^{mp} = RMC^{mp} \times \sum_u FMPU_{mp,u} \tag{27}$$

Table 4.13 The costs of feedstock

Material	Cost
Hardwood biomass	83.33 \$ Mg^{-1} of dry biomass
Catalysts	38.62 \$ m^{-3} of product
Natural gas	85.19 \$ m^{-3} of product
Waste disposal	18.00 \$ Mg^{-1} of dry biomass

The value substitution for the raw material costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (25A) – (27B).

$$CRM^{fp} = 101.33 \times (FPU_{1,1} + FPU_{1,2}) \tag{25A}$$

$$CRM^{u1} = 123.81 \times (FUC_{1,1} + FUC_{1,2}) \tag{26A}$$

$$CRM^{u2} = 123.81 \times (FUC_{2,1} + FUC_{2,2}) \tag{27A}$$

$$CRM^{mp} = 25 \times (FMPU_{1,1} + FMPU_{1,2}) \tag{27B}$$

The operating labour (OL) is function of the number of workers at each plant, expressed as equations (28) – (30). The operating labour is calculated on 3 shifts per day. 4 workers and 12 workers per shift are necessary for the fast pyrolysis plant and upgrading plant, respectively. These numbers of works are used for plant throughput of 2 $Gg\ d^{-1}$. The average salary for the workers is 57.678 \$ h^{-1}.

$$OL^{fp} = olf^{fp} \times \sum_u FPU_{fp,u} \tag{28}$$

$$OL^{u} = olf^{u} \times \sum_c FUC_{u,c} \qquad \forall u \tag{29}$$

$$OL^{mp} = olf^{mp} \times \sum_u FMPU_{mp,u} \tag{30}$$

The value substitution for the operating labor costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (28A) – (30B).

$$OL^{fp} = 0.00119 \times (FPU_{1,1} + FPU_{1,2}) \tag{28A}$$

$$OL^{u1} = 0.0068 \times (FUC_{1,1} + FUC_{1,2}) \tag{29A}$$

$$OL^{u2} = 0.0068 \times (FUC_{2,1} + FUC_{2,2}) \tag{30A}$$

$$OL^{mp} = 0.018 \times (FMPU_{1,1} + FMPU_{1,2}) \tag{30B}$$

The costs of utilities (UT) are equations (31) – (33). The utility factor for the fast pyrolysis plant and upgrading plant are 1.2 $M\$\ y^{-1}$ and 5.5 $M\$\ y^{-1}$. For mobile pyrolyzer, the amount of propane to purchase is 263.83 $m^3\ y^{-1}$ for temperature controlling in mobile pyrolyzer and the propane price is 602 \$ m^{-3}.

$$UT^{fp} = utf^{fp} \times \sum_u FPU_{fp,u} \tag{31}$$

$$UT^{u} = utf^{u} \times \sum_c FUC_{u,c} \qquad \forall u \tag{32}$$

$$UT^{mp} = utf^{mp} \times \sum_u FMPU_{mp,u} \tag{33}$$

The value substitution for the utility costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (31A) – (33B).

$$UT^{fp} = 0.00142 \times (FPU_{1,1} + FPU_{1,2}) \tag{31A}$$

$$UT^{u1} = 0.0123 \times (FUC_{1,1} + FUC_{1,2}) \tag{32A}$$

$$UT^{u2} = 0.0123 \times (FUC_{2,1} + FUC_{2,2}) \tag{33A}$$

$$UT^{mp} = 0.0032 \times (FMPU_{1,1} + FMPU_{1,2}) \tag{33B}$$

The variable costs (VC) are the summation of the labour costs, the utility costs and the raw material costs, equation (34). The operating supervision is 95% of operating labour.

$$VC^{fp,u,mp} = OL^{fp,u,mp} \times smr^{fp,u,mp} + UT^{fp,u,mp} + CRM^{fp,u,mp} \qquad \forall u \tag{34}$$

The value substitution for the variable costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (34A) – (34D).

$$VC^{fp} = 0.95 \times OL^{fp} + UT^{fp} + CRM^{fp} \tag{34A}$$

$$VC^{u1} = 0.95 \times OL^{u1} + UT^{u1} + CRM^{u1} \tag{34B}$$

$$VC^{u2} = 0.95 \times OL^{u2} + UT^{u2} + CRM^{u2} \tag{34C}$$

$$VC^{mp} = 0.95 \times OL^{mp} + UT^{mp} + CRM^{mp} \tag{34D}$$

Finally, the total production cost is expressed as equation (35). The trucks are rented at the price of 25,000 $ y^{-1}. And the average salary for track divers is 40,000 $ y^{-1}.

$$TPC^{fp,u,mp} = VC^{fp,u,mp} + TC^{fp,u,mp} + TT^{fp,u,mp} \times (LD + RP) \qquad \forall u \tag{35}$$

The value substitution for the total production costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (35A) – (35D).

$$TPC^{fp} = VC^{fp} + TC^{fp} + 0.065 \times TT^{fp} \tag{35A}$$

$$TPC^{u1} = VC^{u1} + TC^{u1} + 0.065 \times TT^{u1} \tag{35B}$$

$$TPC^{u2} = VC^{u2} + TC^{u2} + 0.065 \times TT^{u2} \tag{35C}$$

$$TPC^{mp} = VC^{mp} + TC^{mp} + 0.065 \times TT^{mp} \tag{35D}$$

For transportation cost constraints, the transportation costs (TC) depend on distance and mass flow rate, equations (36) – (38). The fuel consumption parameters show in Table 4.14.

$$TC^{fp} = \sum_s(DS \times FSP_{s,fp} \times tcf_b) + \sum_u(DS \times FPU_{fp,u} \times tcf_o \tag{36}$$

$$TC^u = \sum_c(DS \times FUC_{u,c} \times tcf_{gd}) \qquad \forall u \tag{37}$$

$$TC^{mp} = \sum_u(DS \times FMPU_{mp,u} \times tcf_o) \tag{38}$$

The value substitution for the transportation costs for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (36A) – (38B).

$$TC^{fp} = 0.0035 \times FSP_{1,1} + (0.0011 \times FPU_{1,1} + 0.0014 \times FPU_{1,2}) \tag{36A}$$

$$TC^{u1} = 0.00032 \times FUC_{1,1} + 0.0035 \times FUC_{1,2} \tag{37A}$$

$$TC^{u2} = 0.0035 \times FUC_{2,1} + 0.00032 \times FUC_{2,2} \tag{38A}$$

$$TC^{mp} = 0.002 \times FMPU_{1,1} + 0.0013 \times FMPU_{1,2} \tag{38B}$$

The transportation truck numbers (TT) are integer numbers and function of distance, mass flow rate, the capacity of the trucks (CT), the average velocity of the trucks (VL), 45 $km\,h^{-1}$, a number of working hours per day of a truck driver (WH), 10 hrs., and the working days per year (DY), 254 $d\,y^{-1}$, equations (39) – (41).

$$TT^{fp} \geq \sum_s\left(DS \times \frac{FSP_{s,fp}}{CT_b * VL * WH * DY}\right) + \sum_u\left(DS \times \frac{FPU_{fp,u}}{CT_o * VL * WH * DY}\right) \tag{39}$$

$$TT^{mp} \geq \sum_u\left(DS \times \frac{FMPU_{mp,u}}{CT_o * VL * WH * DY}\right) \tag{40}$$

$$TT^{u} \geq \sum_{c}\left(DS \times \frac{FUC_{u,c}}{CT_{gd}*VL*WH*DY}\right) \qquad \forall u \qquad (41)$$

The value substitution for the transportation trucks for the products from the fast pyrolysis plant, the both upgrading plants and the mobile pyrolyzer are expressed as equations (39A) – (41B).

$$TT^{fp} \geq 0.034FSP_{1,1} + 0.011 \times FPU_{1,1} + 0.014 \times FPU_{1,2} \qquad (39A)$$

$$TT^{mp} \geq 0.02 \times FMPU_{1,1} + 0.013 \times FMPU_{1,2} \qquad (40A)$$

$$TT^{u1} \geq 0.0026 \times FUC_{1,1} + 0.028 \times FUC_{1,2} \qquad (41A)$$

$$TT^{u2} \geq 0.028 \times FUC_{1,1} + 0.0026 \times FUC_{1,2} \qquad (41B)$$

Table 4.14 The fuel consumption parameters for transportation cost and truck

Material	**Density**	**Unit**
Pyrolysis oil	1,170	$kg\ m^{-3}$
Chipped biomass	220	$kg\ m^{-3}$
Biofuel	660	$kg\ m^{-3}$
Type of truck	**Volume**	
Freight truck	40	m^3
Tank lorry	40	m^3
Material	**Cost**	
Pyrolysis oil	19.3	\$ $km^{-1}\ Gg^{-1}$
Chipped biomass	103.3	\$ $km^{-1}\ Gg^{-1}$
Biofuel	27.3	\$ $km^{-1}\ dam^{-1}$

For mobile pyrolysis cost constraint, equation (42) refers to the purchased equipment cost of mobile pyrolyzers (EC). The equation cost is fixed at 3.60 M$.

$$EC^{mp} = MPP \tag{42}$$

The value substitution for the equipment cost for the mobile pyrolyzer is expressed as equation (42A).

$$EC^{mp} = 3.60 \tag{42A}$$

After the maximum TAGP is obtained from the optimization solvers, the maximum TAGP is used to find the total supply chain net present value (SCNPV). The total supply chain net present value is considered for 10 years in this case. The total annual gross profit (TAGP), the objective function, becomes the total annual net profit (TANP) by minus tax, 20%, equation (43).

$$TANP = TAGP \times (1 - tax) \tag{43}$$

The value substitution for the total annual net profit (TANP) shows as equation (43A).

$$TANP = TAGP \times 0.8 \tag{43A}$$

And the total annual cash flow (TACF) is the summation of the total annual net profit and the depreciation, equation (44).

$$TACF = TANP + df \times \sum_{fp,u,mp} DP^{fp,u,mp} \tag{44}$$

The value substitution for the total annual cash flow (TACF) shows as equation (44A).

$$TACF = TANP + 0.1 \times (DP^{fp} + DP^{u1} + DP^{u2} + DP^{mp}) \quad (44A)$$

Finally, the total supply chain net present value is the total annual cash flow multiplied by present worth factor, equation (45).

$$SCNPV = \sum_t^{10} TACF \times pwf_t \quad (45)$$

The value substitution for the total supply chain net present value (SCNPV) shows as equation (45A).

$$SCNPV = \sum_t^{10} TACF \times (1 + 1.2)^t \quad (45A)$$

The case study is solved by using Microsoft excel and the developed algorithm, respectively.

4.3.1 MILP Evaluation (Only Integer) For Biofuel Production Supply Chain

In this case, 4 integer-restricted variables are considered to be evaluated, which are transportation truck from supplier 1 to the upgrading plant 1 (TTfp), transportation truck from mobile pyrolyzer to upgrading plant 2 (TTmp), transportation truck from upgrading plant 1 to the both customers (TTu1) and transportation truck from upgrading plant 2 to the both customers (TTu2).

4.3.1.1 Results Of MILP Evaluation (Only Integer) For Biofuel Production Supply Chain

From the developed algorithm evaluation, the total annual gross profit is maximized to 317.48 M\$ y^{-1}. From this total annual gross profit, the total supply chain net present value for 10 years of plant operating is 2,723.34 M\$. The results of mass flows and transportation trucks show in Table 4.15.

Table 4.15 Biofuel production supply chain result from integer case

Variable	Value	Variable	Value
FSP_{11}	1,080.44 $Mg\ d^{-1}$	$FMPU_{12}$	495.27 $Mg\ d^{-1}$
$FSMP_{11}$	898.16 $Mg\ d^{-1}$	FUC_{11}	192.37 $m^3\ d^{-1}$
FPU_{11}	369.96 $Mg\ d^{-1}$	FUC_{12}	17.93 $m^3\ d^{-1}$
FPU_{12}	459.81 $Mg\ d^{-1}$	FUC_{21}	27.62 $m^3\ d^{-1}$
$FMPU_{11}$	28.35 $Mg\ d^{-1}$	FUC_{22}	476.66 $m^3\ d^{-1}$
TTfp	47	TTu2	2
TTu1	1	TTmp	7

From Table 4.15, transportation trucks from supplier 1 to upgrading plant are used in many amounts, 47 trucks, because of many amounts of suppliers 1 (FSP_{11}) and pyrolysis oil from pyrolysis plant 1 and 2 (FPU_{11} *and* FPU_{12}). From the equations (39) – (41) in section 4.3, a number of transportation trucks depends on distance between locations and mass flow rate. From distance from supplier 1 to upgrading plants; 93.34 km from supplier 1 to the fast pyrolysis plant 1, 157.7 km from the fast pyrolysis plant 1 to upgrading plant 1, and 204.4 km from the fast pyrolysis plant 1 to upgrading plant 2, the distance summation is 455.44 km and the mass flow summation is 1,910.21 $Mg\ d^{-1}$. So many transportation trucks from supplier 1 to upgrading plants are used to transfer biomasses and pyrolysis oil. The transportation truck from upgrading plant 1 to the both customers is 1 truck because only 192.37 $m^3\ d^{-1}$ from upgrading plant 1 to customer 1, and 17.93 $m^3\ d^{-1}$ from upgrading plant 1 to customer. However, the total of distance from upgrading plant 1 to the both customers is very large, 387.7 km, but a number of transportation truck is only 1 with small amounts of mass flow rate. For the transportation truck from upgrading plant 2 to the both customers, the distance is the same with one from upgrading plant 1 to the both customers, but the transportation truck is greater because of 504.28 $m^3\ d^{-1}$ of total mass flow rate from upgrading plant 2 to the both customers. It is greater. The transportation truck from mobile pyrolyzer to the both upgrading plants is 7, greater than ones from upgrading plants to customer, because of large amounts of mass flow rate; 28.35 $Mg\ d^{-1}$ from mobile pyrolyzer to upgrading plant 1 and 495.27 $Mg\ d^{-1}$ from mobile pyrolyzer to upgrading plant 2, and very long distance, 479.7 km. Moreover, the mass flow rate is shown as node model in Fig. 4.21.

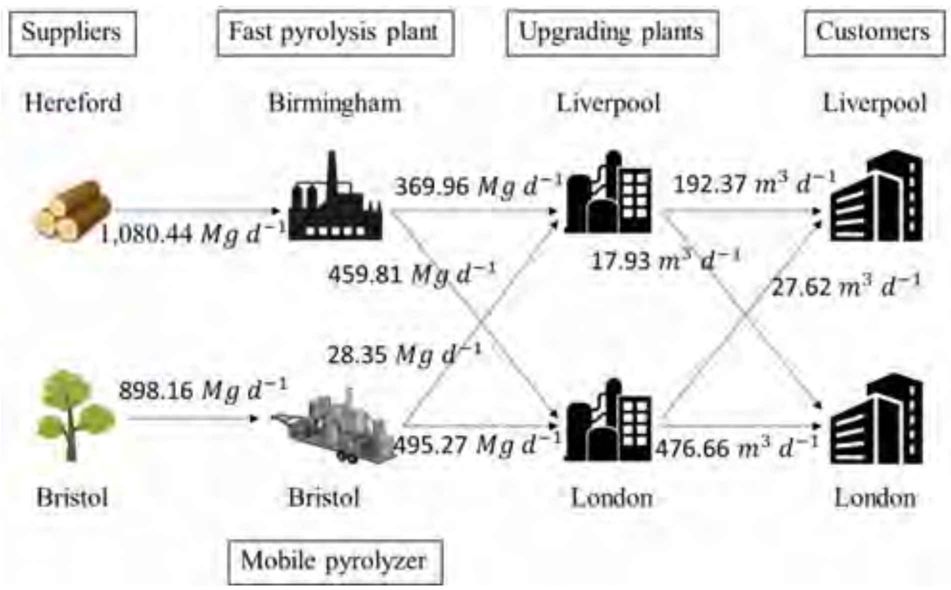


Figure 4.21 Biofuel production supply chain mass flow rate from integer case.

The mass flow rate from mobile pyrolyzer to the upgrading 1 is small amount because the demand of customer 2 (London) is large (180.54 $dam^3\ y^{-1}$). The large amounts of biomasses and pyrolysis oil are transferred to the customer 2. Moreover, the distance from mobile pyrolyzer to upgrading plant 1 is greater than one to upgrading plant 2. All of the suppliers 2 (forest) are used because of cheap price of feedstock (25 \$ Mg^{-1}).

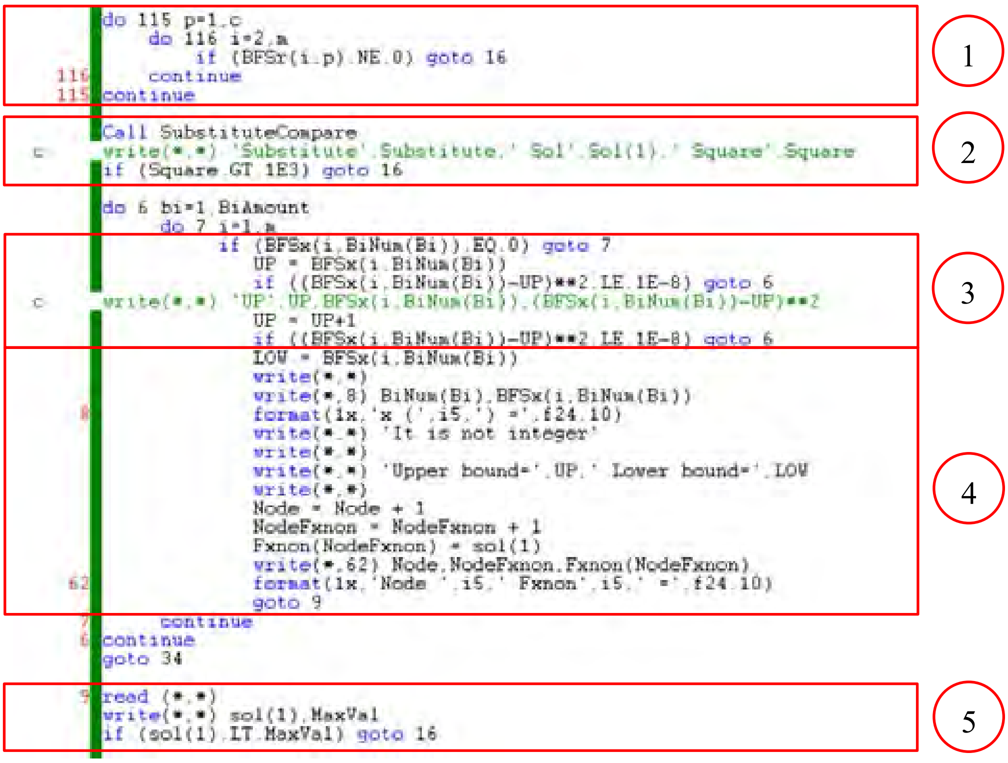
However from results of the both solvers, the total supply chain net present value and the other results are the same. It indicates that this problem evaluated by integer evaluation is not multi-solution problem (is general problem). But the executing time for the both solvers is different.

4.3.1.2 MILP Evaluation (Only Integer) For Biofuel Production

From the results, the solver in Microsoft excel uses 8 subproblems to find the optimum point, as shows in Fig. 4.22.

Figure 4.22 Subproblem from excel in integer case for biofuel production.

In the same time, our algorithm which uses fathoming algorithm to find the optimum point and branching to make new subproblems, shows 8 subproblems the same. The procedure of the algorithm shows in Fig. 4.23.



```
      do 115 p=1,c
         do 116 i=2,m
            if (BFSr(i,p).NE.0) goto 16
116      continue
115   continue

      Call SubstituteCompare
c     write(*,*) 'Substitute',Substitute,' Sol',Sol(1),' Square',Square
      if (Square.GT.1E3) goto 16

      do 6 bi=1,BiAmount
         do 7 i=1,m
            if (BFSx(i,BiNum(Bi)).EQ.0) goto 7
               UP = BFSx(i,BiNum(Bi))
               if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-8) goto 6
c     write(*,*) 'UP',UP,BFSx(i,BiNum(Bi)),(BFSx(i,BiNum(Bi))-UP)**2
               UP = UP+1
               if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-8) goto 6
               LOW = BFSx(i,BiNum(Bi))
               write(*,*)
               write(*,8) BiNum(Bi),BFSx(i,BiNum(Bi))
8              format(1x,'x (',i5,') =',f24.10)
               write(*,*) 'It is not integer'
               write(*,*)
               write(*,*) 'Upper bound=',UP,' Lower bound=',LOW
               write(*,*)
               Node = Node + 1
               NodeFxnon = NodeFxnon + 1
               Fxnon(NodeFxnon) = sol(1)
               write(*,62) Node,NodeFxnon,Fxnon(NodeFxnon)
62             format(1x,'Node ',i5,' Fxnon',i5,' =',f24.10)
               goto 9
7        continue
6     continue
      goto 34

9     read (*,*)
      write(*,*) sol(1),MaxVal
      if (sol(1).LT.MaxVal) goto 16
```

1

2

3

4

5

Figure 4.23 MILP and fathoming algorithm.

From the algorithm in Fig. 4.23, parts 1, 2, 3 and 5 are fathoming and branching to new subproblem is in part 4. From flowchart of fathoming step in Fig. 4.17, the steps of checking non-feasible solution are part 1 and 2 in Fig 4.23. Part 3 is step of checking integer value of integer-restricted variables in fathoming. And part 5 is step of checking Z value comparing with Z in current incumbent.

Part 4 in Fig. 4.23 is step of branching to new subproblem of MILP algorithm. The integer-restricted variables that are not integer would be set to be ≥ the least value that is greater that it, and be ≤ the most value that is less that it. Then the new subproblem would be calculated by LP algorithm.

From this MILP algorithm, the optimum point would be found by 8 subproblems the same as solver on Microsoft excel. The result from MILP algorithm for the problem shows in Fig. 4.24.

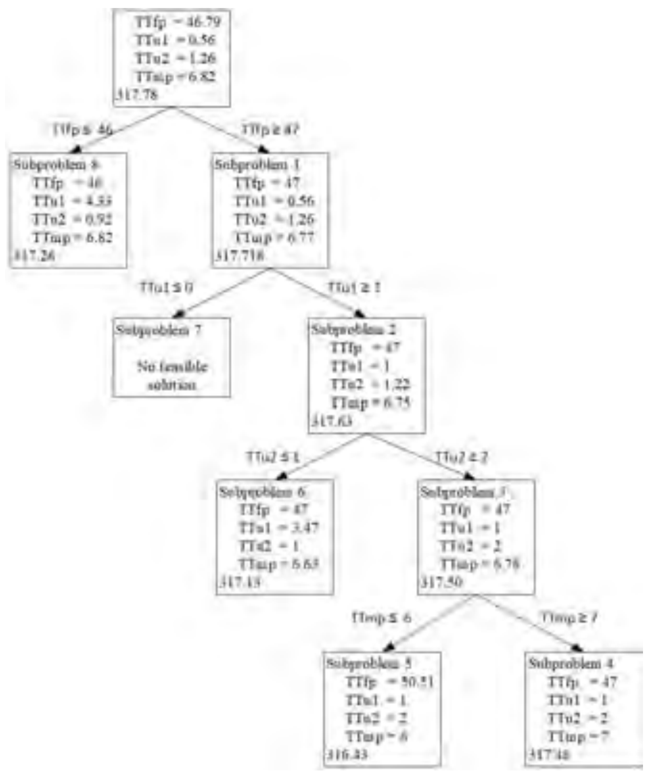


Figure 4.24 MILP procedure for biofuel production supply chain.

From Fig. 4.24, the subproblem 4 stops branching because all integer-restricted variables are integers, which is current incumbent. So, the algorithm in part 3 in Fig. 4.23 would execute. The subproblems 5, 6 and 8 stop branching because its bound is less than Z of the current incumbent. So, the part 5 in Fig. 4.23 would execute. The subproblem 7 stops branching because it is non-feasible solution, which is executed by part 1 and 2 in Fig. 4.23.

This performance indicates that the MILP algorithm (only integer) for this non-multiple problem shows the same result and the same numbers of subproblems as solver on Microsoft excel. However, for multiple problem, the different numbers of subproblems and many solutions from our algorithm are shown. this is discussed in section 4.3.2 and 4.3.3 later.

4.3.2 MILP Evaluation (Only Binary) For Biofuel Production Supply Chain

In this case, 6 integer-restricted variables are considered to be evaluated, that are the decision variable for small scale of the fast pyrolysis plant ($y1^{fp}$), the decision variable for large scale of the fast pyrolysis plant ($y2^{fp}$), the decision variable for small scale of the upgrading plant 1 ($y1^{u1}$), the decision variable for large scale of the upgrading plant 1 ($y2^{u1}$), the decision variable for small scale of the upgrading plant 2 ($y1^{u2}$), the decision variable for large scale of the upgrading plant 2 ($y2^{u2}$). The 6 variables are set to be integer in the solver to use BIP algorithm. However, some constraints of the problem would set the 6 variables to be less than 1, that are equations (12) in section 4.3.

4.3.2.1 Results Of MILP Evaluation (Only Binary) For Biofuel Production

From the developed algorithm evaluation, the total annual gross profit is maximized to 317.18 M\$ y^{-1}. From this total annual gross profit, the total supply chain net present value for 10 years of plant operating is 2,722 M\$. The results of mass flows and the decision variables show in Table 4.16.

From Table 4.16, the decision variable for small scale of the fast pyrolysis plant is zero. From the equation (12) in section 4.3, it is necessary to choose one between the both scales of each plant (pyrolysis plant, upgrading plant 1 and upgrading plant 2). So the decision variable for large scale of the fast pyrolysis plant is 1. It indicates that the large scale of the fast pyrolysis plant is used. From the equations (8A) and (8B), it shows that if the throughput biomass is greater than 850 $Mg\ d^{-1}$, the large plant would be chosen. From Table 4.13, the mass flow from the fast pyrolysis plant to the upgrading plants is about 828.71 $Mg\ d^{-1}$ or about 1,079.05 $Mg\ d^{-1}$ of throughput biomass, so the decision variable for the large scale of the fast pyrolysis plant is 1.

Table 4.16 Biofuel production supply chain result from binary case

Variable	Value	Variable	Value
FSP_{11}	1,079.05 $Mg\ d^{-1}$	$FMPU_{12}$	524.7 $Mg\ d^{-1}$
$FSMP_{11}$	900.00 $Mg\ d^{-1}$	FUC_{11}	220.0 $m^3\ d^{-1}$
FPU_{11}	652.80 $Mg\ d^{-1}$	FUC_{12}	124.68 $m^3\ d^{-1}$
FPU_{12}	175.91 $Mg\ d^{-1}$	FUC_{21}	0 $m^3\ d^{-1}$
$FMPU_{11}$	0 $Mg\ d^{-1}$	FUC_{22}	369.92 $m^3\ d^{-1}$
$y1^{fp}$	0	$y2^{u1}$	1
$y2^{fp}$	1	$y1^{u2}$	0
$y1^{u1}$	0	$y2^{u2}$	1

Moreover, from the equations (9A) and (9B), the large scale of the upgrading plants are used when the throughput biomass is greater than 850 $Mg\ d^{-1}$. From Table 4.10, the mass flow of biofuel from the upgrading plant1 and 2 is about 344.68 $m^3\ d^{-1}$ and 369.92 $m^3\ d^{-1}$, that are about 850 $Mg\ d^{-1}$ and 912.25 $Mg\ d^{-1}$ of the throughput biomass, respectively. So the decision variables for large scale of the both upgrading plants are 1. Besides, the mass flow rate would be shown as node model in Fig. 4.25.

From Fig. 4.25, the mass flow rate from mobile pyrolyzer to the upgrading plant 1 is zero because of large scale of the fast pyrolysis plant. So the mass flow from the fast pyrolysis plant to the upgrading plant 1 is high enough. Moreover, the mass flow rate from the upgrading plant 2 to the customer 1 is zero because mass flow from the large scale of the upgrading plant 1 to customer 1 is high enough.

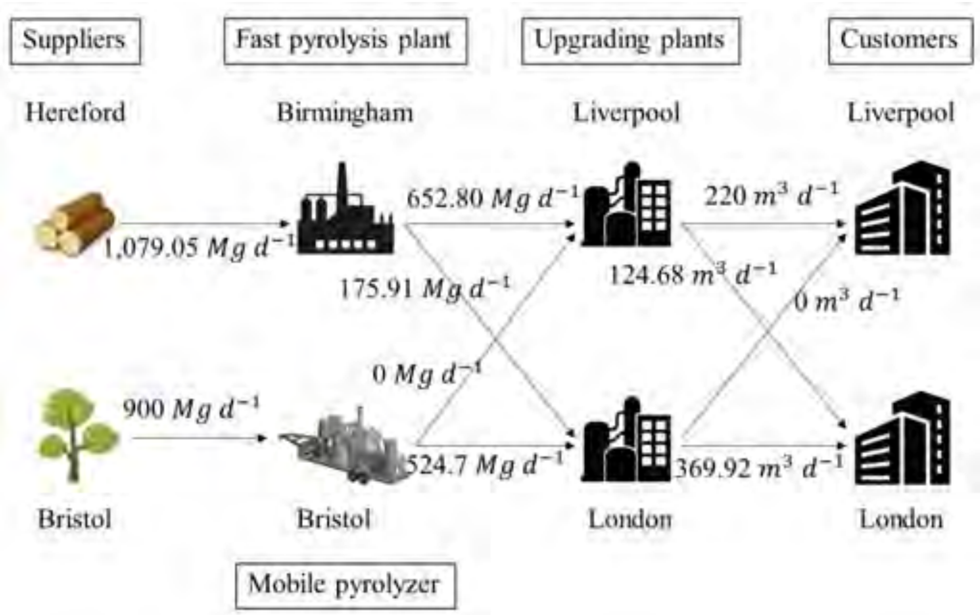


Figure 4.25 Biofuel production supply chain mass flow rate from binary case.

Besides, the results from the both solvers are not the same. So this problem is muti-solution problem.

4.3.2.2 MILP Evaluation (Only Binary) For Biofuel Production

From the results from the both solvers, it indicates that the problem is multi-solution problem. In section 4.1.2, the multi-solution problem depends on the algorithm of choosing the most negative value in the first row for the developed algorithm. In this case, from Fig. 4.9 the symbol (GT) is used for the problem, this result is obtained. For the symbol (GT), it means that the first one of all the most negative values is used. So, if the last one of all the most negative values is used (symbol (GT) become symbol (GE)), the results would change. Table 4.17 and Fig. 4.26 show other results of the problem.

Table 4.17 Other biofuel production supply chain results from binary case

Variable	Value	Variable	Value
FSP_{11}	1,079.05 $Mg\ d^{-1}$	$FMPU_{12}$	524.7 $Mg\ d^{-1}$
$FSMP_{11}$	900.00 $Mg\ d^{-1}$	FUC_{11}	220.0 $m^3\ d^{-1}$
FPU_{11}	416.67 $Mg\ d^{-1}$	FUC_{12}	0 $m^3\ d^{-1}$
FPU_{12}	412.04 $Mg\ d^{-1}$	FUC_{21}	0 $m^3\ d^{-1}$
$FMPU_{11}$	0 $Mg\ d^{-1}$	FUC_{22}	494.6 $m^3\ d^{-1}$
$y1^{fp}$	0	$y2^{u1}$	0
$y2^{fp}$	1	$y1^{u2}$	0
$y1^{u1}$	1	$y2^{u2}$	1

From Table 4.17, the decision variable for large scale of the fast pyrolysis plant is 1 because large amount of mass flow from the fast pyrolysis plant to the both upgrading plant, 828.71 $Mg\ d^{-1}$ of pyrolysis oil or 1,079.05 $Mg\ d^{-1}$ of the throughput biomass. The decision for small scale of the upgrading plant 1 is 1 because mass flow from it to the both customers is 220 $m^3\ d^{-1}$ of biofuel or 542.53 $Mg\ d^{-1}$ of the throughput biomass. So the small scale of the upgrading plant 1 is used. Finally, the decision variable for large scale of the upgrading plant 2 is 1 because mass flow from it to the both customers is 494.6 $m^3\ d^{-1}$ of biofuel or 1,219.72 $Mg\ d^{-1}$ of the throughput biomass. Moreover, the mass flow rate could be shown as node model, as shows in Fig. 4.26.

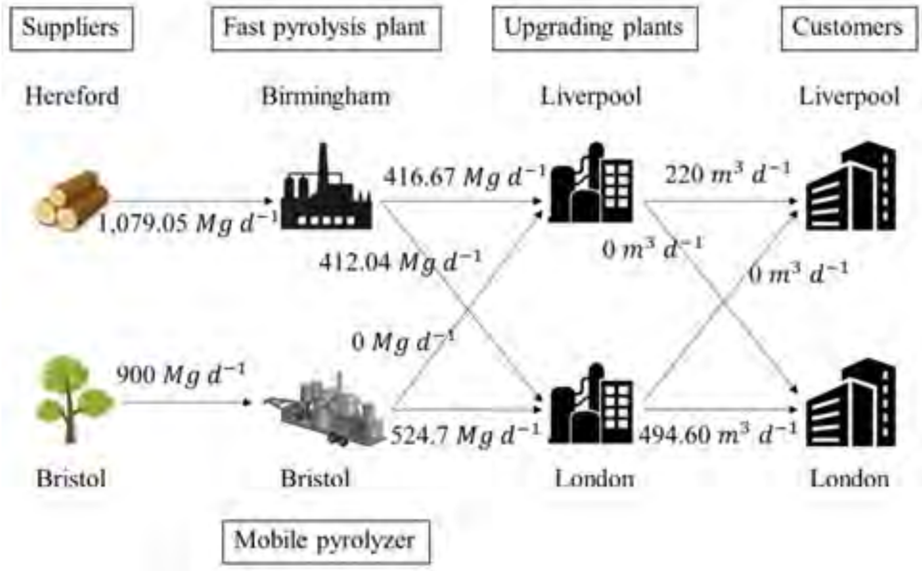


Figure 4.26 Other Biofuel production mass flow rates from binary case.

From Fig. 4.26, the mass flow rate from mobile pyrolyzer to the upgrading plant 1 is zero because large scale of the fast pyrolysis plant is used. So the mass flow from the fast pyrolysis plant to the upgrading plant 1 is high enough. Moreover, the mass flow rate from the upgrading plant 2 to the customer 1 is zero because mass flow from the large scale of the upgrading plant 2 to customer 2 is used in many amounts. So, all of mass flow rates from the upgrading plant 2 are used for the customer 2. Finally, the mass flow rate from the upgrading plant 1 to the customer 2 is zero because mass flow from the large scale of the upgrading plant 2 to customer 2 is high enough. The total annual gross profit is maximized to 317.69 M\$ y^{-1}. From this total annual gross profit, the total supply chain net present value for 10 years of plant operating is 2,726.12 M\$. With many ways for the solutions, it indicates that this problem is multi-solution problem. Fig. 4.27 shows that the Microsoft excel uses 8 subproblems in this problem.

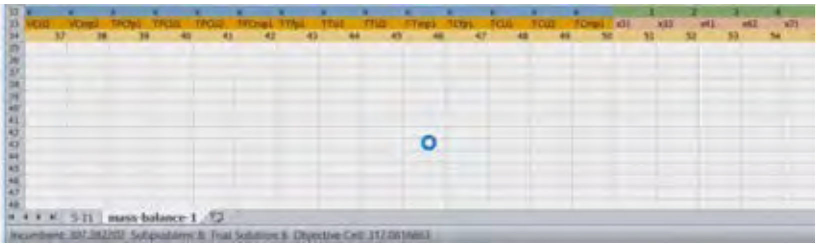


Figure 4.27 Subproblems from excel in binary case for biofuel production.

However, the subproblems used in our algorithm are 6. All subproblems show in Fig. 4.28.

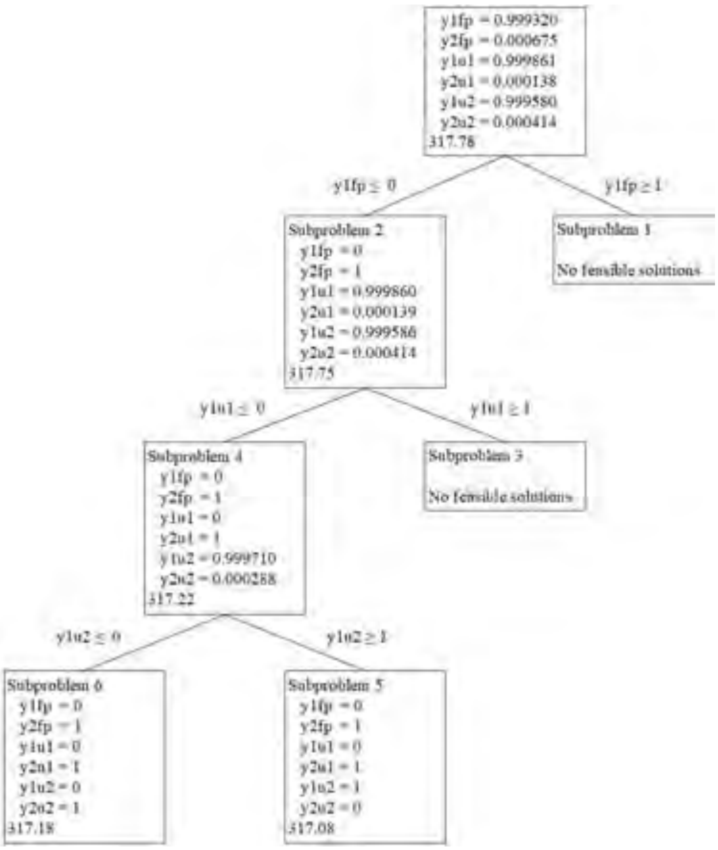


Figure 4.28 Subproblems in binary case for biofuel production supply chain.

The step of fathoming is the same as step in integer case. So, the procedure in Fig. 4.28 is descripted by using Fig. 4.23. The subproblems 1 and 2 are executed by the algorithm in parts 1 and 2. And the subproblems 5 and 6 are executed by the algorithm in part 3.

4.3.3 MILP Evaluation (Both Integer And Binary) For Biofuel Production Supply Chain

For this case, both integer and binary variables would be evaluated together. 4 integer-restricted variables are considered to be evaluated, that are transportation truck from supplier 1 to the upgrading plant 1 (TTfp), transportation truck from mobile pyrolyzer to upgrading plant 2 (TTmp), transportation truck from upgrading plant 1 to the both customers (TTu1) and transportation truck from upgrading plant 2 to the both customers (TTu2). 6 binary-restricted variables are considered to be evaluated, that are the decision variable for small scale of the fast pyrolysis plant ($y1^{fp}$), the decision variable for large scale of the fast pyrolysis plant ($y2^{fp}$), the decision variable for small scale of the upgrading plant 1 ($y1^{u1}$), the decision variable for large scale of the upgrading plant 1 ($y2^{u1}$), the decision variable for small scale of the upgrading plant 2 ($y1^{u2}$), the decision variable for large scale of the upgrading plant 2 ($y2^{u2}$).

4.3.3.1 Results Of MILP Evaluation (Both Integer And Binary) For Biofuel Production

From the developed algorithm evaluation, the total annual gross profit is maximized to 316.97 M\$ y^{-1}. From this total annual gross profit, the total supply chain net present value for 10 years of plant operating is 2,720.19 M\$. The results of mass flows, transportation trucks and the decision variables show in Table 4.18.

From Table 4.18, the transportation trucks from supplier 1 to the upgrading plants are 46 which are very large, because many mass flows from supplier 1 to the upgrading plants are about 1,908.81 $Mg\ d^{-1}$, very large, and because of a long distance from supplier 1 to the upgrading plants, 455.44 km. The transportation trucks from the upgrading plant 1 to the both customers are greater than one from the upgrading plant 2 because of higher amounts in mass flow, 371.86 $m^3\ d^{-1}$ from the upgrading plant 1, and 342.74 $m^3\ d^{-1}$ from the upgrading plant 2.

The transportation trucks from mobile pyrolyzer to the upgrading plants are 7 with 549.50 $Mg\ d^{-1}$ of mass flows and a long distance, 479.7 km. from mobile pyrolyzer to the upgrading plants.

Table 4.18 Biofuel production supply chain result from integer and binary case

Variable	Value	Variable	Value
FSP_{11}	1,079.64 $Mg\ d^{-1}$	$FMPU_{12}$	522.32 $Mg\ d^{-1}$
$FSMP_{11}$	899.22 $Mg\ d^{-1}$	FUC_{11}	215.12 $m^3\ d^{-1}$
FPU_{11}	677.11 $Mg\ d^{-1}$	FUC_{12}	156.74 $m^3\ d^{-1}$
FPU_{12}	152.06 $Mg\ d^{-1}$	FUC_{21}	4.88 $m^3\ d^{-1}$
$FMPU_{11}$	27.18 $Mg\ d^{-1}$	FUC_{22}	337.86 $m^3\ d^{-1}$
Integer variables			
TTfp	46	TTu2	1
TTu1	5	TTmp	7
Binary variables			
$y1^{fp}$	0	$y2^{u1}$	1
$y2^{fp}$	1	$y1^{u2}$	1
$y1^{u1}$	0	$y2^{u2}$	0

Moreover, for the binary variables, the decision variable for large scale of the fast pyrolysis plant is 1 because the mass flow from the fast pyrolysis plant to the upgrading plants is 829.17 $Mg\ d^{-1}$ benefiting from equation (8B) in section 4.3. The decision variable for large scale of the upgrading plant 1 is 1 because of benefit from equation (9B) with large amounts of biofuel from the upgrading plant to the customers. And the decision variable for small scale of the upgrading plant 2 is 1 because of small amounts of biofuel from the upgrading plant 2 to the customers benefiting from the equation (9A). Besides, the mass flow rate results show in Fig. 4.29 as the node model.

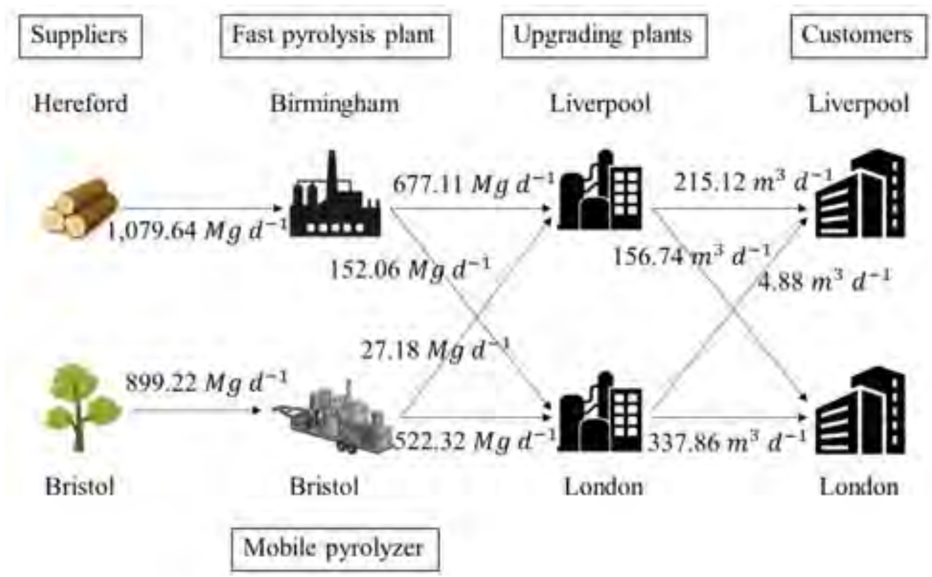


Figure 4.29 Biofuel production mass flow rate from integer and binary case.

From Fig. 4.29, the mass flow rate from mobile pyrolyzer to the upgrading plant 1 is small because large scale of the fast pyrolysis plant is used. So the mass flow from the fast pyrolysis plant to the upgrading plant 1 is high enough. Moreover, the mass flow rate from the upgrading plant 2 to the customer 1 is quite zero because mass flow from the small scale of the upgrading plant 2 to customer 2 is used in many amounts. So, all of the mass flow rates from the upgrading plant 2 are used for the customer 2. Moreover from the different results of the both solvers, it indicates that the problem is multi-solution problem.

4.3.3.2 MILP Evaluation (Both Integer And Binary) For Biofuel Production

From multi-solution problem of this case, this problem would have many solutions with only one objective function value. These different solutions are obtained from the algorithm of choosing the most negative value in the first row, which are discussed in section 4.1.2. When the full MILP algorithm changes the symbol (GT) to the symbol (GE), other solutions would show in Table 4.19.

Table 4.19 Other biofuel production result from integer and binary case

Variable	Value	Variable	Value
FSP_{11}	1,099.97 $Mg\ d^{-1}$	$FMPU_{12}$	451.59 $Mg\ d^{-1}$
$FSMP_{11}$	872.43 $Mg\ d^{-1}$	FUC_{11}	215.12 $m^3\ d^{-1}$
FPU_{11}	647.26 $Mg\ d^{-1}$	FUC_{12}	156.74 $m^3\ d^{-1}$
FPU_{12}	197.52 $Mg\ d^{-1}$	FUC_{21}	4.88 $m^3\ d^{-1}$
$FMPU_{11}$	57.03 $Mg\ d^{-1}$	FUC_{22}	337.86 $m^3\ d^{-1}$
Integer variables			
TTfp	47	TTu2	1
TTu1	5	TTmp	7
Binary variables			
$y1^{fp}$	0	$y2^{u1}$	1
$y2^{fp}$	1	$y1^{u2}$	1
$y1^{u1}$	0	$y2^{u2}$	0

From Table 4.19, all of the decision variables for small and large scale of the plant are the same as one from Table 4.18. But the transportation trucks from supplier 1 to the both upgrading plants change from 46 to 47. This shows the example of the different solutions of this multi-solution problem. The results of mass flow rate shows in Fig. 4.30.

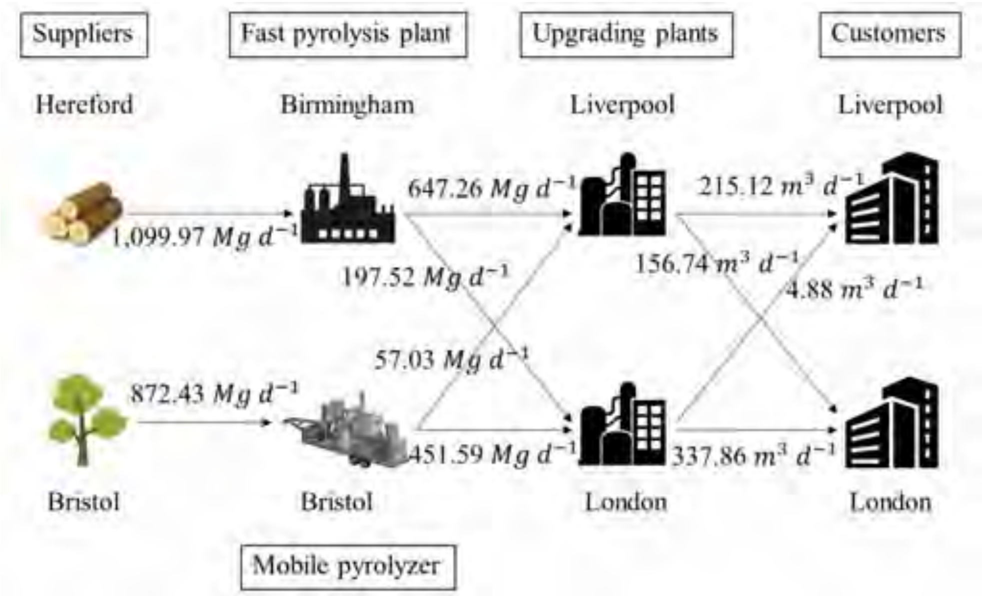


Figure 4.30 Other Biofuel production mass flow rate from integer and binary case.

18 subproblems are used with 1-4 trial solutions for each subproblem in Microsoft excel, as show in Fig. 4.31. And for our algorithm, 54 subproblems are used. The procedures of our algorithm show in Appendix.

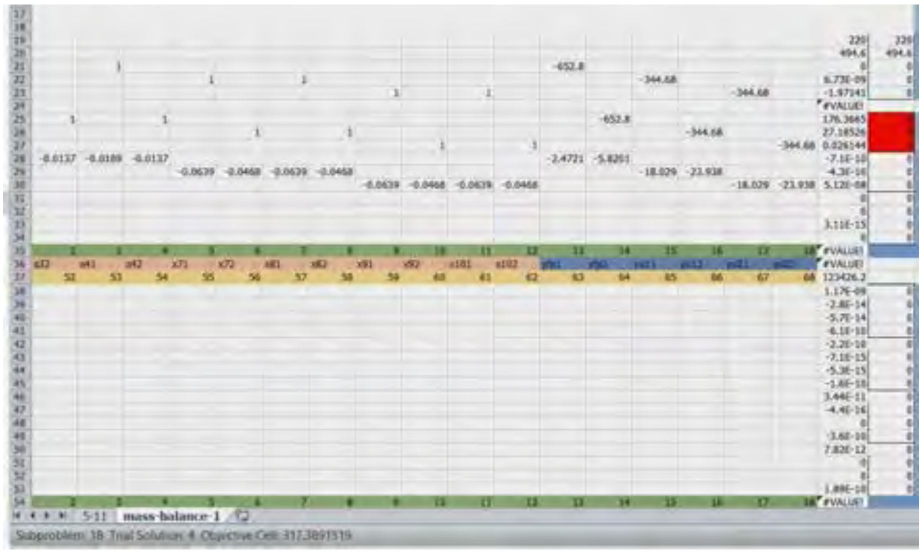


Figure 4.31 Subproblems from Microsoft excel in integer and binary case for biofuel production.

From all results, it indicates that the solver can find many solutions of multi-solution problems from the algorithm of choosing the most negative value.

CHAPTER V
CONCLUSION

The MILP algorithm is developed on Fortran 4.0, and is validated by solver in Microsoft excel. The algorithm is evaluated into 3 parts: (1) Simplex evaluation, (2) Branch-and-bound evaluation, and (3) Biofuel production supply chain.

From simplex evaluation part, the optimum points from the algorithm and solver in Microsoft excel are the same, but the solution are not. This shows that this supply chain problem is multi-optimum problem, which can show many solutions. The algorithm can show many solutions by 3 ways: (1) using the non-basic variable that is 0 in the final LP table as entering variable to obtain new solution, (2) Choosing different the most negative/positive value in the first row, and (3) Adding some new constraints to obtain new solution.

Moreover, the results from the algorithm and the solver show that the optimum points from both algorithm and solver are the same, and many solutions can be obtained from the algorithm also. It indicates that the algorithm has good efficiency to find the optimum point the same as the solver, and can show many solutions to be benefit for alternative solutions. Fathoming is used in the algorithm to increate efficiency in finding the optimum point also.

Finally, in complex supply chain case, the biofuel production supply chain problem is used to evaluate the algorithm. The result shows that the algorithm has good efficiency in finding the optimum point and can show many solutions in complex problem also. It indicates that in more complex supply chain problem, the algorithm still has good efficiency.

REFERENCES

Almaraz, S.D.L., Azzaro-Pantel, C., Montastruc, L., Boix, M. (2015) Deployment of a hydrogen supply chain by multi-objective/multi-period optimization at regional and national scales. Chemical Engineering Research and Design, 104, 11-31.

Babazadeh, R. (2017) Optimal design and planning of biodiesel supply chain considering non-edible feedstock. Renewable and Sustainable Energy Reviews, 75, 1089-1100.

Edgar, T.F., and Himmelblau, D.M., (1989) Optimization Of Chemical Processes. The United State: McGrew-Hill.

Hillier, F.S., and Lieberman, G.J., (2010) Introduction to operations research. New York: McGraw-Hill.

Jokinen, R., Petterson, F., Saxen, H. (2015) An MILP model for optimization of a small-scale LNG supply chain along a coastline. Applied Energy, 138, 423-431.

Jonker, J.G.G., Junginger, H.M., Verstegen, J.A., Lin, T., Rodriguez, L.F., Ting, K.C., FaaiJ, A.P.C. (2016) Supply chain optimization of sugarcane first generation and eucalyptus second generation ethanol production in Brazil. Applied Energy, 173, 494-510.

Jong, S.D., Hoefnagels, R., Wetterlund, E., Pettersson, K., Faaij, A., Junginger, M. (2017) Cost optimization of biofuel production-The impact of scale, integration, transport and supply chain comfigurations. Applied Energy, 195, 1055-1070.

Paulo, H., Azcue, X., Barbosa-Povoa, A.P., Relvas, S. (2015) Supply chain optimization of residual forestry biomass for bioenergy production: The case study of Portugal. Biomass and Bioenergy, 83, 245-256.

Paydar, M.M., Babaveisi, V., Safaei, A.S. (2017) An engine oil closed-loop supply chain design considering collection risk. Computers and Chemical Engineering, 104, 38-55.

Perez-Fortes, M., Lainez-Aguirre, J.M., Arranz-Piera, P., Velo, E., Puigjiner, L. (2012) Design of regional and sustainable bio-based networks for electricity generation using a muti-objective MILP approach. Energy, 44, 79-95.

Pham, T., Yenradee, P. (2017) Optimal supply chain network design with process network and BOM under uncertainties: A case study in toothbrush industry. Computers & Industrial Engineering, 108, 177-191.

Sharifzadeh, M., Garcia, M.C., Shah, N. (2015) Supply chain network design and operation: Systematic decision making for centralized, distributed, and mobile biofuel production using mixed integer linear programming (MILP) under uncertainty. Biomass and Bioenergy, 81, 401-414.

Woo, Y.B., Cho, S., Kim J., Kim, B.S. (2016) Optimization-based approach for strategic design and operation of a biomass-to-hydrogen supply chain. International Journal of Hydrogen Energy, 41, 5405-5418.

APPENDICES

Appendix A The Developed Algorithm Evaluation

The algorithm is developed on Fortran 4.0 and is divided in 2 parts, simplex method part and branch-and-bound part. The both parts would be descripted and evaluated by using the examples in section 2.1 and 2.2.

The simplex method algorithm is shown in Fig. A1. This algorithm is divided to 2 parts, general simplex method and big-M method.

```
C-----------------------------------------------------------------------------------------
C                                          Simplex subroutine
C-----------------------------------------------------------------------------------------
      Subroutine Simplex
      implicit none

      character name*20

      integer a,b,c,m,i,j,k,l,o,p,MaxMin,Binary,BiAmount,BiNum,Bi
      integer PivotCol,PivotRow,NUMx,NUMs,NUMr,Br
      integer BranchFor,BranchBack,BiNUMBranchFor,BiNumBranchBack
      integer Node,NodeFx,NodeFxnon,F,FF
      integer operator,BackRow,BranchBackNumber,BranchBackLock,Ba,BaR
      integer up,low,branchup,branchlow
      integer Valm,Valb,Valc
      integer NonUNI,NonUNInode,aa,jj,NodeNUM
      integer MostX1,RatioX,RatioS,MostS,MostX2,Round

      double precision x,s,r,sol,BFSx,BFSs,BFSr,ratio,PivotNum
      double precision Valx,Vals,Valr,Valsol,Fx,Fxnon
      double precision Xf,Sf,Rf
      double precision Substitute,Square,OriX
      double precision MaxVal

      dimension x(1000,1000),s(1000,1000),r(1000,1000),sol(1000)
      dimension NUMx(1000),NUMs(1000),NUMr(1000),BFSr(1000,1000)
      dimension BFSx(1000,1000),BFSs(1000,1000),ratio(1000)
      dimension Valx(1000,1000),Vals(1000,1000),Valr(1000,1000)
      dimension Valsol(1000),BiNum(1000)
      dimension BiNumBranchFor(1000),BiNumBranchBack(1000,1000)
      dimension Fx(1000),Fxnon(1000),Xf(1000,1000),Sf(1000,1000)
      dimension Rf(1000,1000),branchBack(1000),BranchBackLock(1000)
      dimension branchup(1000),branchlow(1000,1000),OriX(1000)

      common a,b,c,m,i,j,k,l,o,p,MaxMin,PivotCol,PivotRow,NUMx,NUMs,NUMr
      common X,s,r,sol,BFSx,BFSs,BFSr,ratio,PivotNum
      common Valx,Vals,Valr,Valsol,Binary,BiAmount,BiNum,Bi,Br
```

```
      common BranchFor,branchBack,BiNumBranchFor,BiNumBranchBack
      common Node,NodeFx,NodeFxnon,Fx,Fxnon,Xf,Sf,Rf,F,FF
      common operator,BackRow,BranchBackNumber,BranchBackLock,Ba,BaR
      common up,low,branchup,branchlow
      common Valm,Valb,Valc,name
      common Substitute,Square
      common OriX,MaxVal
      common NonUNI,NonUNInode,aa,jj,NodeNUM
      common MostX1,RatioX,RatioS,MostS,MostX2,Round


      MostX1=0
      RatioX=0
      RatioS=0
      MostS =0
      MostX2=0
      Round =1


      if (NonUNI.EQ.1) goto 11
      if (NonUNI.EQ.2) goto 13

  117 continue

      do 6000 i=2,m
        if (sol(i).GE.0) goto 6000
         if (c.EQ.0) goto 6006
            do 6001 p=1,c
               if (r(i,p).EQ.0) goto 6001
                  do 6002 k=1,b
                     if (s(i,k).EQ.0) goto 6002
                     s(1,k) = s(1,k) - (-1)*s(i,k)*1000000
                     s(i,k) = s(i,k)*(-1)
                     r(i,p) = 0
                        do 6003 j=1,a
                        x(1,j) = x(1,j) - (-1)*x(i,j)*1000000
                        x(i,j) = x(i,j)*(-1)
6003                    continue
                        sol(1) = sol(1) - (-1)*sol(i)*1000000
                        sol(i) = sol(i)*(-1)
                        goto 6000
6002              continue
                  do 6004 j=1,a
                  x(i,j) = x(i,j)*(-1)
6004              continue
                  sol(i) = sol(i)*(-1)
                  goto 6000
6001        continue
            c = c+1
            r(i,c) = 1
            do 6010 k=1,b
            if (s(i,k).EQ.0) goto 6010
            s(i,k) = s(i,k)*(-1)
            s(1,k) = s(1,k) + (-1)*s(i,k)*1000000
6010        continue
            do 6005 j=1,a
            x(i,j) = x(i,j)*(-1)
            x(1,j) = x(1,j) + (-1)*x(i,j)*1000000
6005        continue
            sol(i) = sol(i)*(-1)
            sol(1) = sol(1) + (-1)*sol(i)*1000000
            goto 6000
6006      continue
          c = c+1
          r(i,c) = 1
          do 6011 k=1,b
          if (s(i,k).EQ.0) goto 6011
          s(i,k) = s(i,k)*(-1)
          s(1,k) = s(1,k) + (-1)*s(i,k)*1000000
6011      continue
          do 6007 j=1,a
          x(i,j) = x(i,j)*(-1)
          x(1,j) = x(1,j) + (-1)*x(i,j)*1000000
6007      continue
          sol(i) = sol(i)*(-1)
          sol(1) = sol(1) + (-1)*sol(i)*1000000
6000 continue
```

```
c ---------------------------------------------------------------------------------------------

      Call StartingBFS

c ---------------------------------------------------------------------------------------------

      o=0
 2000 continue
      o=o+1
      write(*,*)
      write(*,5006) o
 5006 format(1x,'The table No.',i6)

      do 2507 j=1,a
      Write(*,*) 'X',j,X(1,j)
 2507 continue
      write(*,*) 'sol',sol(1)

C------------------------------------ Select the least negative value --------------------------

      GOTO 106

C----------------------------------------------------------------------------------

  106 continue
      j=1
    7 continue
           if (x(1,j).GE.0) goto 70
             if (j.EQ.a) goto 8
               do 9 l=j+1,a
                    if (MostX1.EQ.1) goto 2500
                    if (x(1,j).GT.x(1,l)) goto 71
                    goto 9
 2500               continue
                    if (x(1,j).GE.x(1,l)) goto 71
    9          continue
    8        continue
                 do 10 l=1,b
                    if (MostX2.EQ.1) goto 2501
                    if (x(1,j).GT.s(1,l)) goto 72
                    goto 10
 2501               continue
                    if (x(1,j).GE.s(1,l)) goto 72
   10          continue
               goto 11
   70 continue
      if (j.EQ.a) goto 73
      j=j+1
      goto 7
   71 continue
      j=1
      goto 7


   72 continue
      k=1
      goto 74
   73 continue
      k=1
   74 continue
   12 continue
            if (s(1,k).GE.0) goto 75
                if (k.EQ.b) goto 13
                  do 14 l=k+1,b
                     if (MostS.EQ.1) goto 2502
                     if (s(1,k).GT.s(1,l)) goto 76
                     goto 14
 2502                continue
                     if (s(1,k).GE.s(1,l)) goto 76
   14             continue
                  goto 13
   75 continue
      if (k.EQ.b) goto 77
      k=k+1
      goto 12
   76 continue
```

```
      k=1
      goto 12
C---------------------------------------- Obtain the Answer --------------------------------

   77 continue
      Call ObtainAnswer

      do 120 p=1,c
          do 121 i=2,m
              if (BFSr(i,p).NE.0) goto 122
  121     continue
  120 continue

      Call SubstituteCompare
      if (Square.GT.1E3) goto 122

      write(*,*) 'Complete'
      goto 2505
  122 continue
      write(*,*) 'Non complete'
 2505 continue

      write(*,*) 'Round ',Round
      Round = Round + 1
      if (Round.GT.32) goto 2506
      Call InputValue
      Call SettingForNewBFS
      Call RunGTGE
      read (*,*)
      goto 117

 2506 continue
      NodeNUM = NodeNUM + 1
      write(*,*) NodeNUM


      goto 2007
C------------------------------------------- X zone --------------------------------

   11 continue
 5010 format(1x,'The least negative value is',f24.10)
 5011 format(1x,'So the entering variable is X (',i5,')')

      PivotCol = j
      do 15 i=2,m
            if (X(i,j).LE.0) goto 16
                ratio(i) = sol(i)/x(i,j)
 5012           format(1x,'Ratio (',i5,') =',E30.20,1x,E30.20,E30.20)
                goto 15
   16       continue
                ratio(i) = -1
 5013          format(1x,'Ratio (',i5,') is infinity',15x,E30.20,E30.20)

   15 continue

C ------------------------------------------------------------------------------------------

      GOTO 129

C ------------------------------------------------------------------------------------------

  129 continue
      i=2
  130 continue
      if (ratio(i).LT.0) goto 131
         if (i.EQ.m) goto 19
             do 133 l=i+1,m
                  if (ratio(l).LT.0) goto 133
                      if (RatioK.EQ.1) goto 2503
                      if (ratio(i).GT.ratio(l)) goto 132
                      goto 133
 2503                 continue
                      if (ratio(i).GE.ratio(l)) goto 132

  133        continue
```

```
                    goto 19
 131 continue
     if (i.EQ.m) goto 123
     i=i+1
     goto 130
 132 continue
     i=1
     goto 130

c ------------------------------------------------------------------------------------------

 123 continue
     write(*,*) 'there is no counting ratio'
     goto 77

c ------------------------------------------------------------------------------------------

  19          continue
 140 continue

     PivotRow = i
     PivotNum = x(PivotRow,PivotCol)
5014 format(1x,'The pivot row is',i5)
5015 format(1x,'The pivot number is',f24.10)

     if (ratio(72).NE.ratio(87)) goto 9000
9000 continue
     if (x(72,j).NE.x(87,j)) goto 9001
9001 continue
     if (sol(72).NE.sol(87)) goto 9002
9002 continue
     if (sol(87).NE.(ratio(72)*x(87,j))) goto 9003
9003 continue

     do 21 j=1,a
           x(PivotRow,j) = x(PivotRow,j)/Pivotnum
5016       format(1x,'x (',i5,') =',f24.10)
  21 continue

     do 22 k=1,b
           s(PivotRow,k) = s(PivotRow,k)/PivotNum
5017       format(1x,'s (',i5,') =',f24.10)
  22 continue

     do 1006 p=1,c
           r(PivotRow,p) = r(PivotRow,p)/PivotNum
5552       format(1x,'r (',i5,') =',f24.10)
1006 continue

     sol(PivotRow) = sol(PivotRow)/PivotNum
5018 format(1x,'Sol =',f24.10)
     PivotNum = x(PivotRow,Pivotcol)

     do 23 i=1,PivotRow-1
5019 format(1x,'The new values in the row No.',i5)
           if (PivotCol.EQ.1) goto 24
                do 25 j=1,PivotCol-1
                x(i,j) = x(i,j) - x(i,PivotCol)*x(PivotRow,j)
5020            format(1x,'x (',i5,') =',f24.10)
  25            continue
  24       continue
                   if (PivotCol.EQ.a) goto 26
                       do 27 j=Pivotcol+1,a
                       x(i,j) = x(i,j) - x(i,PivotCol)*x(PivotRow,j)
  27                   continue
  26               continue
           do 28 k=1,b
           s(i,k) = s(i,k) - x(i,PivotCol)*s(PivotRow,k)
5021       format(1x,'s (',i5,') =',f24.10)
  28       continue

           do 1007 p=1,c
           r(i,p) = r(i,p) - x(i,PivotCol)*r(PivotRow,p)
5550       format(1x,'r (',i5,') =',f24.10)
1007       continue
```

```
             sol(i) = sol(i) - x(i,PivotCol)*sol(PivotRow)
             x(i,PivotCol) = x(i,PivotCol) - x(i,PivotCol)*PivotNum
 5022        format(1x,'sol =',E30.22)
   23 continue

      if (PivotRow.EQ.m) goto 2008
         do 29 i=PivotRow+1,m
            if (PivotCol.EQ.1) goto 30
                 do 31 j=1,PivotCol-1
                 x(i,j) = x(i,j) - x(i,PivotCol)*x(PivotRow,j)
   31            continue
   30       continue
                    if (PivotCol.EQ.a) goto 32
                        do 33 j=PivotCol+1,a
                        x(i,j) = x(i,j) - x(i,PivotCol)*x(PivotRow,j)
   33                   continue
   32               continue
            do 34 k=1,b
            s(i,k) = s(i,k) - x(i,PivotCol)*s(PivotRow,k)
   34       continue

            do 1008 p=1,c
            r(i,p) = r(i,p) - x(i,PivotCol)*r(PivotRow,p)
 1008       continue

            sol(i) = sol(i) - x(i,PivotCol)*sol(PivotRow)
            x(i,PivotCol) = x(i,PivotCol) - x(i,PivotCol)*PivotNum
   29 continue

 2008 continue
        NUMx(PivotRow) = PivotCol
        NUMs(PivotRow) = 0
        NUMr(PivotRow) = 0
        BFSx(PivotRow,PivotCol) = sol(PivotRow)
            if (PivotCol.EQ.1) goto 55
               do 56 j=1,PivotCol-1
               BFSx(PivotRow,j) = 0
   56          continue
   55       continue
            if (PivotCol.EQ.a) goto 57
               do 58 j=PivotCol+1,a
               BFSx(PivotRow,j) = 0
   58          continue
   57       continue
            do 59 k=1,b
            BFSs(PivotRow,k) = 0
   59       continue

            do 1009 p=1,c
            BFSr(PivotRow,p) = 0
 1009       continue

        goto 2009
C-------------------------------------------------S zone --------------------------------

   13 continue
 5023 format(1x,'So the entering Variable is s (',i5,')')

      PivotCol = k
      do 35 i=2,m
            if (s(i,k).LE.0) goto 36
                ratio(i) = sol(i)/s(i,k)
                goto 35
   36       continue
                ratio(i) = -1
   35 continue

      do 38 i=2,m
            if (ratio(i).LT.0) goto 38
               if (i.EQ.m) goto 39
                  do 40 l=i+1,m
                     if (ratio(l).LT.0) goto 40
                        if (RatioS.EQ.1) goto 2504
                        if (ratio(i).GT.ratio(l)) goto 38
                        goto 40
 2504                   continue
                        if (ratio(i).GE.ratio(l)) goto 38
```

```
 40             continue
                goto 39
 38 continue
    Write(*,*) 'there is no counting number'

    goto 77

 39           continue

    PivotRow = i
    PivotNum = s(PivotRow,PivotCol)

    do 41 j=1,a
          x(PivotRow,j) = x(PivotRow,j)/Pivotnum
 41 continue

    do 42 k=1,b
          s(PivotRow,k) = s(PivotRow,k)/PivotNum
 42 continue

    do 1010 p=1,c
          r(PivotRow,p) = r(PivotRow,p)/PivotNum
1010 continue

    sol(PivotRow) = sol(PivotRow)/PivotNum
      PivotNum = s(PivotRow,Pivotcol)

      do 43 i=1,PivotRow-1
          do 44 j=1,a
          x(i,j) = x(i,j) - s(i,PivotCol)*x(PivotRow,j)
 44       continue
             if (PivotCol.EQ.1) goto 45
                do 46 k=1,PivotCol-1
                s(i,k) = s(i,k) - s(i,PivotCol)*s(PivotRow,k)
 46             continue
 45       continue
                  if (PivotCol.EQ.b) goto 47
                      do 48 k=Pivotcol+1,b
                      s(i,k) = s(i,k) - s(i,PivotCol)*s(PivotRow,k)
 48                   continue
 47               continue

          do 1011 p=1,c
          r(i,p) = r(i,p) - s(i,PivotCol)*r(PivotRow,p)
1011      continue

          sol(i) = sol(i) - s(i,PivotCol)*sol(PivotRow)
          s(i,PivotCol) = s(i,PivotCol) - s(i,PivotCol)*PivotNum
 43 continue

    if (PivotRow.EQ.m) goto 2010
       do 49 i=PivotRow+1,m
           do 50 j=1,a
           x(i,j) = x(i,j) - s(i,PivotCol)*x(PivotRow,j)
 50        continue

               if (PivotCol.EQ.1) goto 51
                  do 52 k=1,PivotCol-1
                  s(i,k) = s(i,k) - s(i,PivotCol)*s(PivotRow,k)
 52               continue
 51            continue
                  if (PivotCol.EQ.b) goto 53
                      do 54 k=Pivotcol+1,b
                      s(i,k) = s(i,k) - s(i,PivotCol)*s(PivotRow,k)
 54                   continue
 53               continue

          do 1012 p=1,c
          r(i,p) = r(i,p) - s(i,PivotCol)*r(PivotRow,p)
1012      continue

          sol(i) = sol(i) - s(i,PivotCol)*sol(PivotRow)
          s(i,PivotCol) = s(i,PivotCol) - s(i,PivotCol)*PivotNum
 49    continue

2010 continue
      NUMx(PivotRow) = 0
```

```
          NUMs(PivotRow) = PivotCol
          NUMr(PivotRow) = 0
          BFSS(PivotRow,PivotCol) = sol(PivotRow)
             do 60 j=1,a
             BFSx(PivotRow,j) = 0
  60         continue

             if (PivotCol.EQ.1) goto 61
                do 62 k=1,PivotCol-1
                BFSs(PivotRow,k) = 0
  62            continue
  61         continue
             if (PivotCol.EQ.b) goto 63
                do 64 k=PivotCol+1,b
                BFSs(PivotRow,k) = 0
  64            continue
  63         continue

             do 1013 p=1,c
             BFSr(PivotRow,p) = 0
 1013        continue

          goto 2009
C------------------------------------- Obtain new BFS zone ------------------------------

 2009 continue
      do 2011 i=2,m
          if (NUMx(i).EQ.0) goto 2012
          BFSx(i,NUMx(i)) = sol(i)
          goto 2011
 2012     continue
              if (NUMs(i).EQ.0) goto 1014
              BFSs(i,NUMs(i)) = sol(i)
              goto 2011
 1014         continue
                 BFSr(i,NUMr(i)) = sol(i)
 2011 continue

 5024 format(1x,'The answer of table No.',i5)

      do 2017 j=1,a
          do 2018 i=2,m
             if (BFSx(i,j).LE.0) goto 2019
 5007        format(1x,'BFSx (',i5,') =',f24.10)
             goto 2017
 2019        continue
                if (i.LT.m) goto 2018
 2018     continue
 2017   continue
      do 2020 k=1,b
          do 2021 i=2,m
             if (BFSs(i,k).LE.0) goto 2022
 5008        format(1x,'BFSs (',i5,') =',f24.10)
             goto 2020
 2022        continue
                if (i.LT.m) goto 2021
 2021     continue
 2020 continue

      do 1015 p=1,c
          do 1016 i=2,m
             if (BFSr(i,p).LE.0) goto 1017
 5551        format(1x,'BFSr (',i5,') =',f24.10)
             goto 1015
 1017        continue
                if (i.LT.m) goto 1016
 1016     continue
 1015 continue

 5025 format(1x,'Maximum of this turn is',f24.10)
      goto 2000

C------------------------------------------ goto 2000 -----------

 2007 continue

      end
```

Figure A1 Simplex method algorithm.

From Fig. A1, the simplex method is developed by steps descripted in section 2.1.1 for general simplex method ($\leq constraint$) without big-M method. The example for simplex method with ($\leq constraint$) is descripted in section 2.1.1.

Example for ($\leq constraint$) problem

Minimize $z = -3X_1 - 2X_2 + 5X_3$

Subject to $X_1 + 2X_2 + X_3 \leq 430$

$3X_1 + 2X_3 \leq 450$

$X_1 + 4X_2 \leq 420$

Give $X_1, X_2, X_3 \geq 0$

In the first step, a number of equations and variable would be input into the program followed by coefficient of each variable. The result from the algorithm shows in Fig. A2.

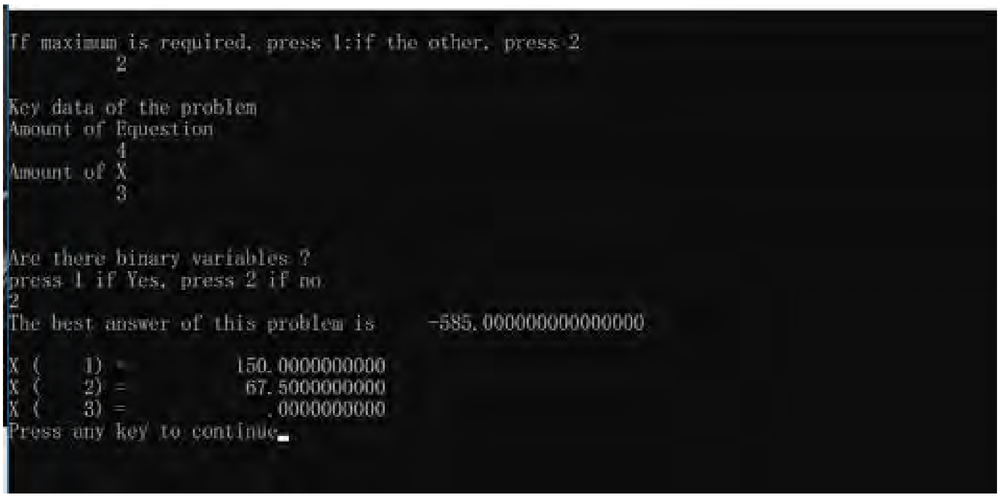


Figure A2 The result of $\leq constraint$ problem from the algorithm.

The simplex method algorithm for big-M method shows in Fig. A3. The technique to solve by big-M is descripted in section 2.1.2.

```
            if (operator.NE.2) goto 7005
            write(*,7003) i
            read (1,*) Valsol(i)
            b=b+1
            c=c+1
            Vals(i,b) = -1
            Valr(i,c) = 1
               do 7006 j=1,a
               Valx(1,j) = Valx(1,j) - Valx(i,j)*1000000
7006           continue
               do 7007 k=1,b
               Vals(1,k) = Vals(1,k) - Vals(i,k)*1000000
7007           continue
               Valsol(1) = Valsol(1) - Valsol(i)*1000000
               goto 7012
7005        continue
                  write(*,7003) i
                  read (1,*) Valsol(i)
                  c=c+1
                  Valr(i,c) = 1
                  do 7009 j=1,a
                  Valx(1,j) = Valx(1,j) - Valx(i,j)*1000000
7009              continue
                  do 7010 k=1,b
                  Vals(1,k) = Vals(1,k) - Vals(i,k)*1000000
7010              continue
                  Valsol(1) = Valsol(1) - Valsol(i)*1000000
7012 continue
```

Figure A3 The algorithm for big-M method.

From Fig. A3, the algorithm is developed for the first table of the problem and then the algorithm in Fig. A1 would be used to solve. The example for big-M method is divided to 2 cases; $(= constraint)$ problem and $(\geq constraint)$ problem, descripted in section 2.1.2.

Example for $(= constraint)$ problem

Maximize $z = 3X_1 + 5X_2$

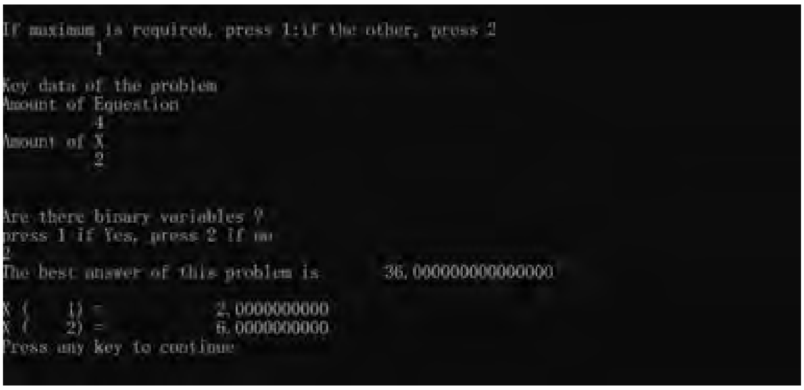
Subject to $X_1 \leq 4$

$2X_2 \leq 12$

$3X_1 + 2X_2 = 18$

Give $X_1, X_2 \geq 0$

In the first step, the algorithm in Fig. A3 is used to set the first table of the problem and them the algorithm in Fig. A1 is used to solve. The result shows in Fig. A4.



```
If maximum is required, press 1:if the other, press 2
        1

Key data of the problem
Amount of Equestion
        4
Amount of X
        2

Are there binary variables ?
press 1 if Yes, press 2 if no
2
The best answer of this problem is        36.000000000000000

X (    1) =            2.0000000000
X (    2) =            6.0000000000
Press any key to continue
```

Figure A4 The result of $= constraint$ problem from the algorithm.

Example for ($\geq constraint$) problem

Minimize $\quad z = 4X_1 + X_2$

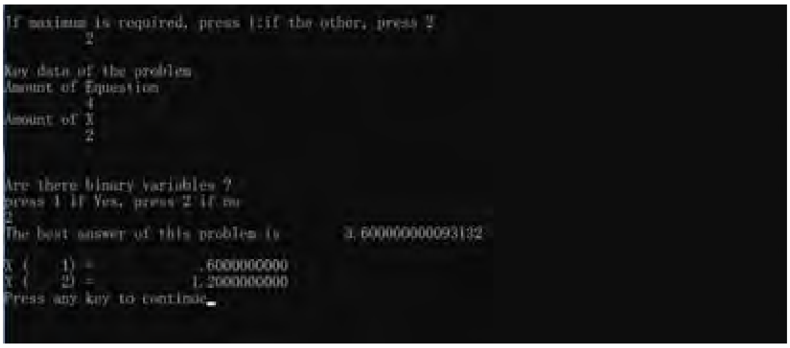
Subject to $\quad 3X_1 + X_2 = 3$

$4X_1 + 3X_2 \geq 6$

$X_1 + 2X_2 \leq 3$

Give $X_1, X_2 \geq 0$

The result of the ($\geq constraint$) problem shows in Fig. A5.



```
If maximum is required, press 1:if the other, press 2
        2

Key data of the problem
Amount of Equestion
        4
Amount of X
        2

Are there binary variables ?
press 1 if Yes, press 2 if no
2
The best answer of this problem is        3.600000000093132

X (    1) =            .6000000000
X (    2) =            1.2000000000
Press any key to continue
```

Figure A5 The result of $\geq constraint$ problem from the algorithm.

For branch-and-bound algorithm, it is shown in Fig. A6. The examples for this algorithm are divided to 3 examples; MILP problem, pure binary problem and BIP (mixed binary real) problem.

```
      program Simplex Method
      implicit none

      character name*20

      integer a,b,c,m,i,j,k,l,o,p,MaxMin,Binary,BiAmount,BiNum,Bi
      integer PivotCol,PivotRow,NUMx,NUMs,NUMr,Br
      integer BranchFor,BranchBack,BiNuMBranchFor,BiNumBranchBack
      integer Node,NodeFx,NodeFxnon,F,FF
      integer operator,BackRow,BranchBackNumber,BranchBackLock,Ba,BaR
      integer up,low,branchup,branchlow
      integer Valm,Valb,Valc
      integer NonUNI,NonUNInode,aa,jj,NodeNUM
      integer MostX1,RatioX,RatioS,MostS,MostX2,Round

      double precision x,s,r,sol,BFSx,BFSs,BFSr,ratio,PivotNum
      double precision Valx,Vals,Valr,Valsol,Fx,Fxnon
      double precision Xf,Sf,Rf
      double precision Substitute,Square,OriX
      double precision MaxVal

      dimension x(1000,1000),s(1000,1000),r(1000,1000),sol(1000)
      dimension NUMx(1000),NUMs(1000),NUMr(1000),BFSr(1000,1000)
      dimension BFSx(1000,1000),BFSs(1000,1000),ratio(1000)
      dimension Valx(1000,1000),Vals(1000,1000),Valr(1000,1000)
      dimension Valsol(1000),BiNum(1000)
      dimension BiNumBranchFor(1000),BiNumBranchBack(1000,1000)
      dimension Fx(1000),Fxnon(1000),Xf(1000,1000),Sf(1000,1000)
      dimension Rf(1000,1000),branchBack(1000),BranchBackLock(1000)
      dimension branchup(1000),branchlow(1000,1000),OriX(1000)

      common a,b,c,m,i,j,k,l,o,p,MaxMin,PivotCol,PivotRow,NUMx,NUMs,NUMr
      common x,s,r,sol,BFSx,BFSs,BFSr,ratio,PivotNum
      common Valx,Vals,Valr,Valsol,Binary,BiAmount,BiNum,Bi,Br
      common BranchFor,branchBack,BiNumBranchFor,BiNumBranchBack
      common Node,NodeFx,NodeFxnon,Fx,Fxnon,Xf,Sf,Rf,F,FF
      common operator,BackRow,BranchBackNumber,BranchBackLock,Ba,BaR
      common up,low,branchup,branchlow
      common Valm,Valb,Valc,name
      common Substitute,Square
      common OriX,MaxVal
      common NonUNI,NonUNInode,aa,jj,NodeNUM
      common MostX1,RatioX,RatioS,MostS,MostX2,Round

C-----------------------------------------------------------------------------
C                                          Input values in box.dat
C-----------------------------------------------------------------------------

      open (1,file='simplex evaluation-5-6-2.dat')

C-----------------------------------------------------------------------------
C                                          Choose one of the both; Maximize, Minimize
C-----------------------------------------------------------------------------
      write(*,*)
      write(*,*) 'If maximum is required, press 1;if the other, press 2'
      read (1,*) MaxMin
      write(*,*) MaxMin
      write(*,*)

C--------------------------------------- Key data ---------------------------------

      write(*,*) 'Key data of the problem'
      write(*,*) 'Amount of Equestion'
      read (1,*) m
      write(*,*) m
      write(*,*) 'Amount of X'
      read (1,*) a
      write(*,*) a
      b=0
      c=0

      i=1
      write(*,*)
      read (1,*) name
      write(*,*) name
      read (1,*) aa
```

```
      do 7002 jj=1,aa
            read (1,*) j,Valx(i,j)
            OriX(j) =  Valx(i,j)
            write(*,7001) i,j
7001        format(1x,'X (',i5,',',i5,')')
            write(*,*) Valx(i,j)
            Valx(i,j) = Valx(i,j)*(-1)
7002 continue
            read (1,*) Valsol(i)
            write(*,7003) i
7003        format(1x,'sol (',i5,')')

      if (MaxMin.EQ.1) goto 7014
      do 7015 j=1,a
      ValX(1,j) = ValX(1,j)*(-1)
7015 continue
      Valsol(1) = Valsol(1)*(-1)
7014 continue

      do 7012 i=2,m
      read (1,*) name
      write(*,*) name
      read (1,*) aa
5000 format(1X,'Equestion',i5)
            do 7013 jj=1,aa
            read (1,*) j,Valx(i,j)
            write(*,5001) i,j
5001        format(1x,'x (',i5,',',i5,')')
7013        continue
      read (1,*) operator
      write(*,*) operator
             if (operator.NE.1) goto 7004
             write(*,7003) i
             read (1,*) Valsol(i)
             b=b+1
             Vals(i,b) = 1
             goto 7012
7004         continue
                if (operator.NE.2) goto 7005
                Write(*,7003) i
                read (1,*) Valsol(i)
                b=b+1
                c=c+1
                Vals(i,b) = -1
                Valr(i,c) = 1
                   do 7006 j=1,a
                   Valx(1,j) = Valx(1,j) - Valx(i,j)*1000000
7006               continue
                   do 7007 k=1,b
                   Vals(1,k) = Vals(1,k) - Vals(i,k)*1000000
7007               continue
                   Valsol(1) = Valsol(1) - Valsol(i)*1000000
                   goto 7012
7005            continue
                      write(*,7003) i
                      read (1,*) Valsol(I)
                      c=c+1
                      Valr(i,c) = 1
                      do 7009 j=1,a
                      Valx(1,j) = Valx(1,j) - Valx(i,j)*1000000
7009                  continue
                      do 7010 k=1,b
                      Vals(1,k) = Vals(1,k) - Vals(i,k)*1000000
7010                  continue
                      Valsol(1) = Valsol(1) - Valsol(i)*1000000
7012 continue

      Valm=m
      Valb=b
      Valc=c

      write(*,*)
      write(*,*) 'Are there binary variables ?'
      write(*,*) 'press 1 if Yes, press 2 if no'
      read (*,*) Binary
```

```
      if (Binary.EQ.2) goto 4
      write(*,*)
      write(*,*) 'How many binary variables?'
      read (*,*) BiAmount

      write(*,*)
      write(*,*) 'Which X are binary?'
         do 5 Bi=1,BiAmount
            read (*,*) BiNum(Bi)
    5    continue
C---------------------------------------------- Obtain the first BFS ------------------------------
    4 continue
      Call InputValue

C----------------------------------------------------------------------------------------------------
C                                              Mixed Integer Linear Programming
C----------------------------------------------------------------------------------------------------
      if (binary.EQ.1) goto 61

  125 continue
      Call Simplex

  128 continue
      Call XfSfRf
      if (MaxMin.EQ.1) goto 500
      if (NonUNI.GT.0) goto 500
      Sol(1) = Sol(1)*(-1)
  500 continue
      write(*,*) 'The best answer of this problem is ',sol(1)
      Call XSRforFx

      do 120 j=1,a
         do 121 i=2,m
            if (NUMx(i).EQ.j) goto 120
  121    continue
         if (x(1,j).NE.0) goto 120
         Write(*,*) 'x',j
  120 continue
      do 122 k=1,b
         do 123 i=2,m
            if (NUMs(i).EQ.k) goto 122
  123    continue
         if (s(1,k).NE.0) goto 122
         Write (*,*) 's',k
  122 continue

      write(*,*) 'press 0 to skip: 1 for x: 2 for s'
      read (*,*) NonUNI
      if (NonUNI.EQ.0) goto 10
         if (NonUNI.EQ.2) goto 124
         write(*,*) 'choose the variable, x'
         read (*,*) j
         goto 125
  124    continue
           write(*,*) 'choose the variable, s'
           read (*,*) k
           goto 125

      goto 10
c ------------------------------------------------------------------------------------------------

   61 continue
      write(*,*) 'Press 1 for doing non unique: if not, press 0'
      read (*,*) NonUNI
      if (NonUNI.EQ.0) goto 126
      write(*,*) 'Press node number'
      read (*,*) NonUNInode

  126 continue
      BranchFor=0
      BackRow=0
      BranchBack(BackRow)=0
      BranchBackNumber=0
      BranchBackLock(BranchBackNumber)=0
```

```
      Node=0
      NodeFx=0
      NodeFxnon=0
   17 Call Simplex

      do 115 p=1,c
           do 116 i=2,m
                if (BFSr(i,p).NE.0) goto 16
 116       continue
 115  continue

      Call SubstituteCompare
      if (Square.GT.1E3) goto 16

      do 6 bi=1,BiAmount
           do 7 i=1,m
                if (BFSx(i,BiNum(Bi)).EQ.0) goto 7
                   UP = BFSx(i,BiNum(Bi))
                   if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-16) goto 6
                   UP = UP+1
                   if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-16) goto 6
                   LOW = BFSx(i,BiNum(Bi))
                   write(*,8) BiNum(Bi),BFSX(i,BiNum(Bi))
    8              format(1x,'x (',i5,') =',f24.10)
                   write(*,*) 'It is not integer'
                   write(*,*) 'Upper bound=',UP,' Lower bound=',LOW
                   Node = Node + 1
                   NodeFxnon = NodeFxnon + 1
                   Fxnon(NodeFxnon) = sol(1)
   62              format(1x,'Node ',i5,' Fxnon',i5,' =',f24.10)
                   goto 9
    7      continue
    6 continue
      goto 34

    9 continue
      if (NodeFx.EQ.0) goto 135
      if (Maxmin.EQ.1) goto 130
      sol(1) = sol(1)*(-1)
      if (sol(1).GT.MaxVal) goto 16
      goto 135
  130 continue
      if (sol(1).LT.MaxVal) goto 16

  135 continue
      Call InputValue
      Call SettingForNewBFS


      BranchFor = BranchFor+1
      BackRow = BackRow+1
      BranchBack(BackRow) = 1
      BiNumBranchFor(BranchFor) = BiNum(Bi)
      BiNumBranchBack(BackRow,BranchBack(BackRow)) = BiNum(Bi)
      BranchUP(BranchFor) = UP
      BranchLOW(BackRow,BranchBack(BackRow)) = LOW

   82 continue
      do 19 Br=1,BranchFor
      m=m+1
      b=b+1
      c=c+1

         do 11 j=1,a
         x(m,j) = 0
   11    continue
         do 83 k=1,h
         s(m,k) = 0
   83    continue
         do 84 p=1,c
         r(m,p) = 0
   84    continue
         sol(m) = 0

         do 110 i=1,m
         s(i,b) = 0
         r(i,c) = 0
```

```
110    continue

       x(m,BiNumBranchFor(Br)) = 1
       sol(m) = BranchUP(Br)
       s(m,b) = -1
       r(m,c) = 1

       do 85 j=1,a
       x(1,j) = x(1,j) - x(m,j)*1000000
 85    continue
       do 86 k=1,b
       s(1,k) = s(1,k) - s(m,k)*1000000
 86    continue
       sol(1) = sol(1) - sol(m)*1000000
 19 continue

    do 71 Ba=1,BranchBackNumber
       do 72 BaR=1,BranchBack(BranchBackLock(Ba))-1
       m=m+1
       b=b+1

          do 87 j=1,a
          x(m,j) = 0
 87       continue
          do 88 k=1,b
          s(m,k) = 0
 88       continue
          do 89 p=1,c
          r(m,p) = 0
 89       continue
          sol(m) = 0

          do 111 i=1,m
          s(i,b) = 0
111       continue

          x(m,BiNumBranchBack(BranchBackLock(Ba),BaR))=1
          sol(m) = BranchLOW(BranchBackLock(Ba),BaR)
          s(m,b) = 1

 72    continue
 71 continue

          do 12 i=2,m
             do 13 j=1,a
                if (x(i,j).EQ.0) goto 13
                goto 12
 13          continue
                   if (c.EQ.0) goto 18
                      do 14 p=1,c
                         if (r(i,p).EQ.0) goto 14
                            do 15 k=1,b
                               if (s(i,k).EQ.0) goto 15
                               if (sol(i).GT.0) goto 16
                               Call NewSettingXSR
                               goto 12
 15                         continue
                                  if (sol(i).NE.0) goto 16
                                  Call NewSettingXSR
                                  goto 12
 14                   continue
                                  if (sol(i).LT.0) goto 16
                                  Call NewSettingXSR
                                  goto 12
 18                continue
                                  if (sol(i).LT.0) goto 16
                                  Call NewSettingXSR
                                  goto 12
 12       continue
    Goto 17

 16 continue
    Node = Node + 1
    write(*,*) 'No feasible solution'
    Goto 34

 34 continue
```

```
      write(*,*) 'All of integer variables are integer'
      Node = Node + 1
      NodeFx = NodeFx + 1
      if (NonUNInode.EQ.0) goto 127
      if (NodeFx.NE.NonUNInode) goto 127
      goto 128
  127 continue
      FX(NodeFX) = sol(1)
   74 format(1X,'Node ',i5,' FX',i5,' =',f24.10)
      Call XfSfRf
      if (Maxmin.EQ.1) goto 131
      sol(1) = sol(1)*(-1)
      if (NodeFx.EQ.1) goto 143
      if (sol(1).GT.MaxVal) goto 33
  143 continue
      MaxVal = sol(1)
      goto 33
  131 continue
      if (NodeFx.EQ.1) goto 144
      if (sol(1).LT.MaxVal) goto 33
  144 continue
      MaxVal = sol(1)

   33 continue

      if (BranchFor.EQ.0) goto 51
      Call InputValue
      Call SettingForNewBFS

      BranchFor = BranchFor-1

      if (BranchBack(BackRow).NE.1) goto 75
      BranchBackNumber = BranchBackNumber+1
      BranchBackLock(BranchBackNumber) = BackRow
   75 continue

      do 20 Br=1,BranchFor
      m=m+1
      b=b+1
      c=c+1

         do 92 j=1,a
         x(m,j) = 0
   92    continue
         do 93 k=1,b
         s(m,k) = 0
   93    continue
         do 94 p=1,c
         r(m,p) = 0
   94    continue
         sol(m) = 0

         do 112 i=1,m
         s(i,b) = 0
         r(i,c) = 0
  112    continue

         x(m,BiNumBranchFor(Br)) = 1
         sol(m) = BranchUP(Br)
         s(m,b) = -1
         r(m,c) = 1

         do 95 j=1,a
         x(1,j) = x(1,j) - x(m,j)*1000000
   95    continue
         do 96 k=1,b
         s(1,k) = s(1,k) - s(m,k)*1000000
   96    continue
         sol(1) = sol(1) - sol(m)*1000000
   20 continue

      do 76 Ba=1,BranchBackNumber-1
         do 77 BaK=1,BranchBack(BranchBackLock(Ba))-1
         m=m+1
         b=b+1

            do 97 j=1,a
```

```
          x(m,j) = 0
 97       continue
          do 98 k=1,b
          s(m,k) = 0
 98       continue
          do 99 p=1,c
          r(m,p) = 0
 99       continue
          sol(m) = 0

          do 113 i=1,m
          s(i,b) = 0
113       continue

          x(m,BiNumBranchBack(BranchBackLock(Ba),BaR))=1
          sol(m) = BranchLOW(BranchBackLock(Ba),BaR)
          s(m,b) = 1
 77    continue
 76 continue

    do 79 BaR=1,BranchBack(BackRow)
       m=m+1
       b=b+1

          do 102 j=1,a
          x(m,j) = 0
102        continue
          do 103 k=1,b
          s(m,k) = 0
103        continue
          do 104 p=1,c
          r(m,p) = 0
104        continue
          sol(m) = 0

          do 114 i=1,m
          s(i,b) = 0
114       continue

          x(m,BiNumBranchBack(BranchBackLock(BranchBackNumber),BaR))=1
          sol(m) = BranchLOW(BranchBackLock(BranchBackNumber),BaR)
          s(m,b) = 1
 79 continue

          do 28 i=2,m
             do 29 j=1,a
                if (x(i,j).EQ.0) goto 29
                goto 28
 29          continue
                   if (c.EQ.0) goto 32
                      do 30 p=1,c
                         if (r(i,p).EQ.0) goto 30
                            do 31 k=1,b
                               if (s(i,k).EQ.0) goto 31
                               if (sol(i).GT.0) goto 47
                               Call NewSettingXSR
                               goto 28
 31                         continue
                               if (sol(i).NE.0) goto 47
                               Call NewSettingXSR
                               goto 28
 30                   continue
                               if (sol(i).LT.0) goto 47
                               Call NewSettingXSR
                               goto 28
 32                 continue
                               if (sol(i).LT.0) goto 47
                               Call NewSettingXSB
                               goto 28
 28       continue

    Call Simplex

    do 117 p=1,c
       do 118 i=2,m
          if (BFSr(i,p).NE.0) goto 47
118    continue
```

```
117 continue
    Call SubstituteCompare
    if (Square.GT.1E3) goto 47

    do 37 bi=1,BiAmount
         do 38 i=1,m
            if (BFSx(i,BiNum(Bi)).EQ.0) goto 38
               UP = BFSx(i,BiNum(Bi))
               if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-16) goto 37
               UP = UP+1
               if ((BFSx(i,BiNum(Bi))-UP)**2.LE.1E-16) goto 37
               LOW = BFSx(i,BiNum(Bi))
               write(*,8) BiNum(Bi),BFSx(i,BiNum(Bi))
               write(*,*) 'It is not integer'
               write(*,*) 'Upper bound=',UP,' Lower bound=',LOW
               Node = Node + 1
               NodeFxnon = NodeFxnon + 1
               FXnon(NodeFxnon) = sol(1)
               goto 81
 38      continue
 37 continue
    goto 40

 81 continue

    if (NodeFX.EQ.0) goto 137
    if (Maxmin.EQ.1) goto 138
    sol(1) = sol(1)*(-1)
    if (sol(1).GT.MaxVal) goto 47
    goto 137
138 continue
    if (sol(1).LT.MaxVal) goto 47

137 continue
    Call InputValue
    Call SettingForNewBFS
    BranchFor = BranchFor+1
    BranchBack(BackRow) = BranchBack(BackRow)+1
    BiNumBranchFor(BranchFor) = BiNum(Bi)
    BiNumBranchBack(BackRow,BranchBack(BackRow)) = BiNum(Bi)
    BranchUP(BranchFor) = UP
    BranchLOW(BackRow,BranchBack(BackRow)) = LOW
    goto 82

 47 continue
    Node = Node + 1
    write(*,*) 'No feasible solution'
    goto 50

 40 write(*,*)
    write(*,*) 'All of integer variables are integer'
    Node = Node + 1
    NodeFx = NodeFx + 1
    if (NonUNInode.EQ.0) goto 129
    if (NodeFx.NE.NonUNInode) goto 129
    goto 128
129 continue
    Fx(NodeFx) = sol(1)
    Call XfSfRf
    if (Maxmin.EQ.1) goto 132
    sol(1) = sol(1)*(-1)
    if (NodeFx.EQ.1) goto 145
    if (sol(1).GT.MaxVal) goto 50
145 continue
    MaxVal = sol(1)
    goto 50
132 continue
    if (NodeFX.EQ.1) goto 146
    if (sol(1).LT.MaxVal) goto 50
146 continue
    MaxVal = sol(1)

 50 continue
    BackRow = BackRow-1
    BranchBackNumber = BranchBackNumber-1

    if (BackRow.EQ.0) goto 51
```

```
      goto 31

   51 continue

      if (NodeFx.NE.0) goto 147
      write(*,*) 'No Answer'
      goto 10
  147 continue

      if (Maxmin.EQ.1) goto 140
         do 141 F=1,NodeFx
           Fx(F) = Fx(F)*(-1)
  141    continue
  140 continue

      write(*,*) 'A number of nodes',NodeFx
      write(*,*)
      if (NodeFx.GT.1) goto 63
      write(*,*) 'The best answer of this problem is ',Fx(NodeFx)
      Call XSRforFx
      goto 10
   63 continue

      do 64 F=1,NodeFx
         if (F.LT.NodeFx) goto 65
         write(*,*) 'The best answer of this problem is ',FX(NodeFX)
         Call XSRforFx
         goto 10
   65    continue
            do 66 FF=F+1,NodeFx
               if (Maxmin.EQ.1) goto 136
                  if ((FX(F).LT.Fx(FF)).OR.(Fx(FF).EQ.0)) goto 66
                  goto 64
  136          continue
                  if (FX(F).GE.Fx(FF)) goto 66
                  goto 64
   66       continue
            write(*,*) 'node',F
            write(*,*) 'The best answer of this problem is ',FX(F)
            Call XSBforFx2
            goto 10
   64 continue

   10 END
```

Figure A6 The algorithm for branch-and-bound.

From Fig. A6, the algorithm is used after the simplex execution. The procedure is descripted in section 2.2. The examples for this algorithm evaluation are below, descripted in section 2.2.

Example for MILP problem

Maximize $Z = 4X_1 - 2X_2 + 7X_3 - X_4$

$$\begin{array}{lll} \text{Subject to} & X_1 + 5X_3 & \leq 10 \\ & X_1 + X_2 - X_3 & \leq 1 \\ & 6X_1 - 5X_2 & \leq 0 \\ & -X_1 + 2X_3 - 2X_4 & \leq 3 \end{array}$$

$X_1, X_2, X_3, X_4 \geq 0$

X_1, X_2, X_3 are integer.

In the first step, a number of equations, a number of variables and a number of integer-restricted variables would be input into the program followed by coefficient of each variable. The result of the MILP problem shows in Fig. A7.

Figure A7 The result of MILP problem from the algorithm.

Example for pure binary problem

Maximize $f = 86X_1 + 4X_2 + 40X_3$

Subject to $774X_1 + 76X_2 + 42X_3 \leq 875$

$67X_1 + 27X_2 + 53X_3 \leq 875$

Give $X_1, X_2, X_3 = 0, 1$

The result of the pure binary problem shows in Fig. A8.

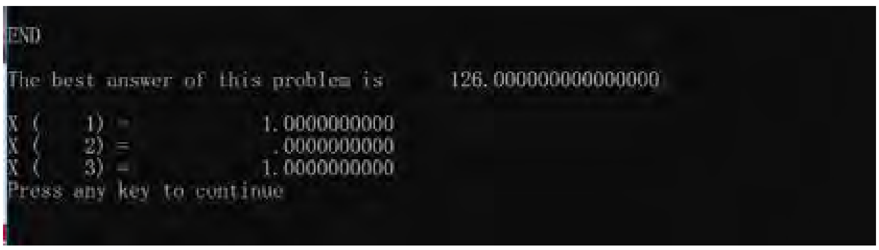


Figure A8 The result of pure binary problem from the algorithm.

Example for BIP (mixed binary real) problem

Maximize $Z = 5X_1 + 7X_2 + 3X_3$

Subject to

$$X_1 \leq 7$$
$$X_2 \leq 5$$
$$X_3 \leq 9$$
$$X_1 - 100y_1 \leq 0$$
$$X_2 - 100y_2 \leq 0$$
$$X_3 - 100y_3 \leq 0$$
$$y_1 + y_2 + y_3 \leq 2$$
$$3X_1 + 4X_2 + 2X_3 - 100y_4 \leq 30$$
$$4X_1 + 6X_2 + 2X_3 - 100y_4 \leq 4{,}000$$

Give X_1, X_2 and $X_3 \geq 0$

y_1, y_2, y_3 and y_4 are binary.

The result of the BIP (mixed binary real) problem shows in Fig. A9.

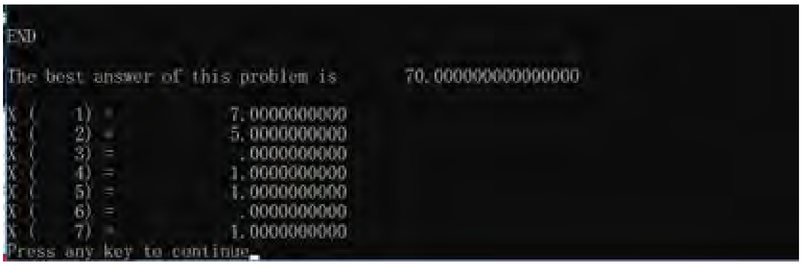


Figure A9 The result of BIP (mixed binary real) problem from the algorithm.

Appendix B Degenerate Problem Description

The solution from degenerate problem could be obtained by many ways with one optimal objective function solution. The graphical solutions in Fig. B1 and B2 show the difference between the normal problem solution and the degenerate problem solution, respectively.

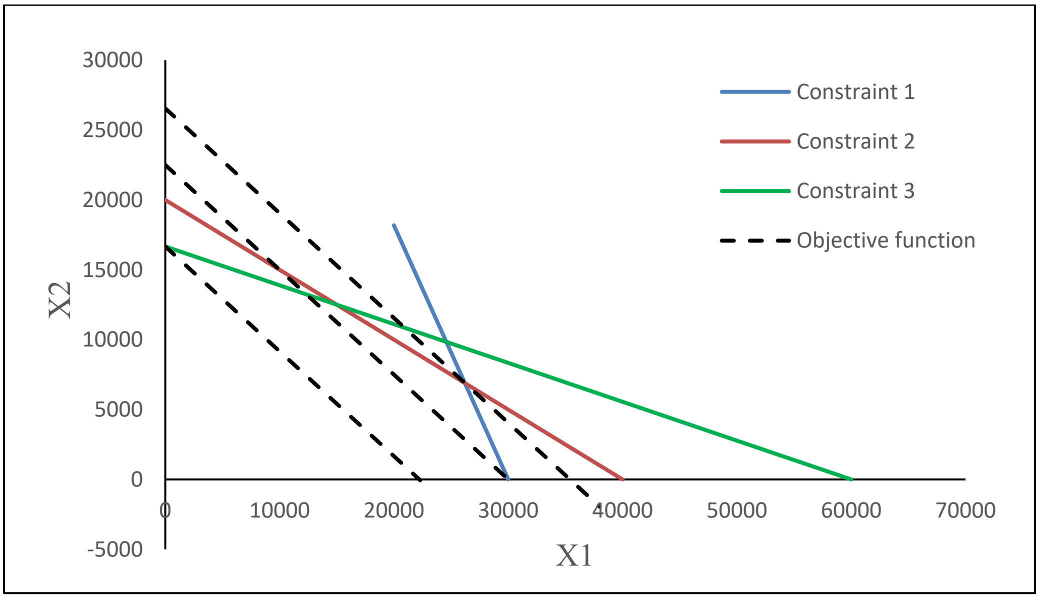


Figure B1 Graphical solution of normal problem.

Fig. B1 shows that one optimal solution could occur in normal problem because each point of objective function line on constraint line gives different solution.

However for degenerate problem, only one of optimal objective function solution would occur, but the many solutions can occur because many point of objective function line on constraint 2 line would give many solution (values of x_1 *and* x_2) in one value of objective function.

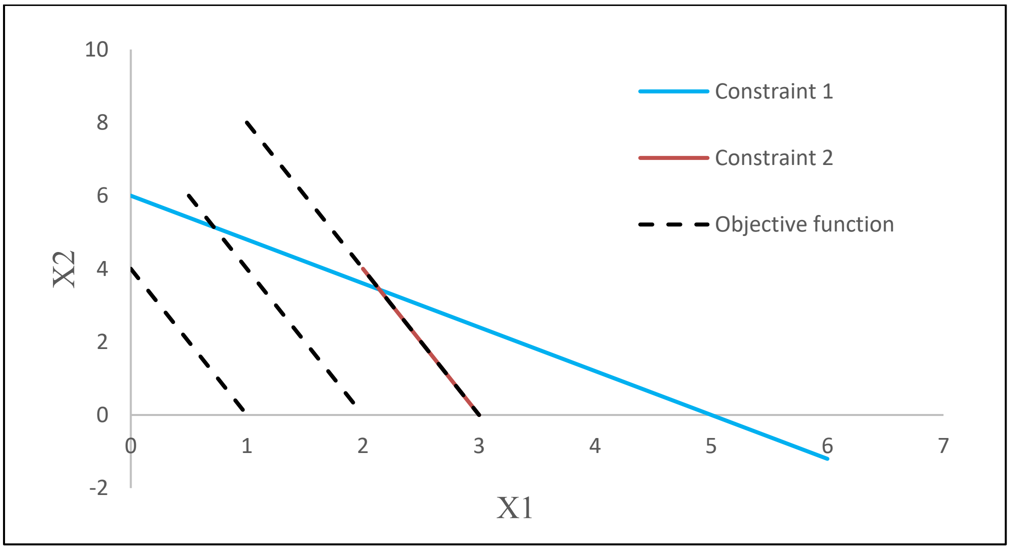
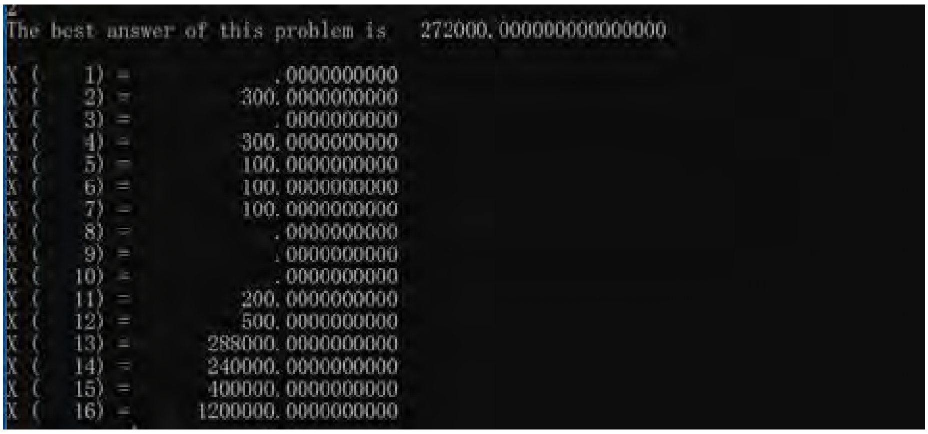


Figure B2 Graphical solution of degenerate problem.

Appendix C Simplex Evaluation For Small Supply Chain

The result from the developed algorithm shows in Fig. C1.



```
The best answer of this problem is   272000.000000000000000

X (   1) =           .0000000000
X (   2) =        300.0000000000
X (   3) =           .0000000000
X (   4) =        300.0000000000
X (   5) =        100.0000000000
X (   6) =        100.0000000000
X (   7) =        100.0000000000
X (   8) =           .0000000000
X (   9) =           .0000000000
X (  10) =           .0000000000
X (  11) =        200.0000000000
X (  12) =        500.0000000000
X (  13) =     288000.0000000000
X (  14) =     240000.0000000000
X (  15) =     400000.0000000000
X (  16) =    1200000.0000000000
```

Figure C1 Simplex evaluation for small supply chain.

The results from Microsoft excel show in Table C1, and Fig. C2. The maximum profit is 272,000$.

Table C1 Results for small supply chain problem from Microsoft excel

Variable	Value	Variable	Value
x_{11}	100 items	y_{11}	100 items
x_{12}	200 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	0 items
x_{31}	0 items	y_{22}	200 items
x_{32}	200 items	y_{23}	500 items
TC	288,000$	OPC	400,000$
OWC	240,000$	TI	1,200,000$

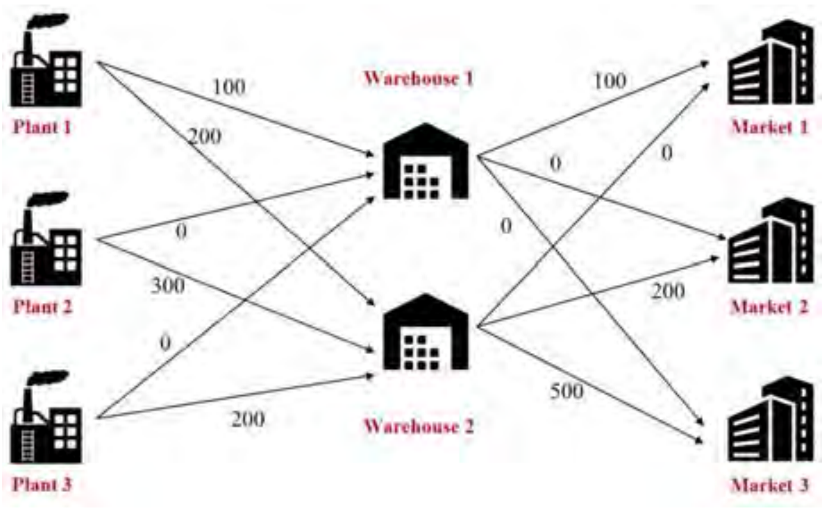
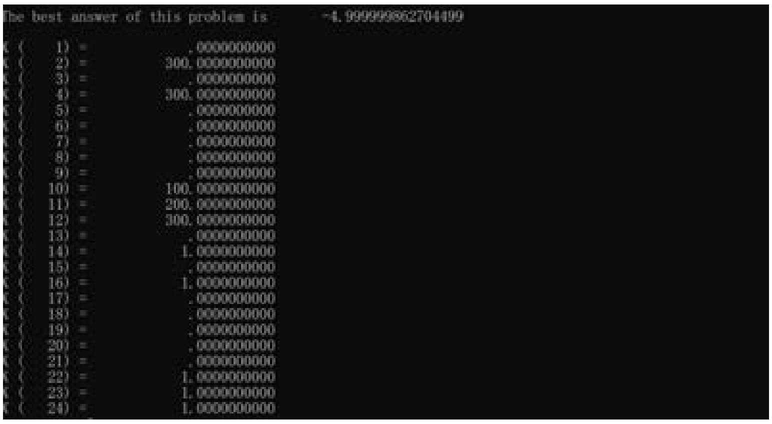


Figure C2 Results for small supply chain problem from Microsoft excel.

From Table C1 and Fig. C2, x_{21} and x_{31} are zero because of high transportation cost (and very long distance) and low capacity of warehouse 1. The values of y_{12} and y_{13} are zero because of limit of warehouse 1, but y_{21} is zero because of high transportation cost (4 kmile, the longest distance).

Appendix D Branch-And-Bound Evaluation

Result of small supply chain from developed algorithm shows in Fig. D1. And result from excel shows in Table D1 and Fig. D2.



```
The best answer of this problem is      -4.999999862704499
X (    1) =           .0000000000
X (    2) =        300.0000000000
X (    3) =           .0000000000
X (    4) =        300.0000000000
X (    5) =           .0000000000
X (    6) =           .0000000000
X (    7) =           .0000000000
X (    8) =           .0000000000
X (    9) =           .0000000000
X (   10) =        100.0000000000
X (   11) =        200.0000000000
X (   12) =        300.0000000000
X (   13) =           .0000000000
X (   14) =          1.0000000000
X (   15) =           .0000000000
X (   16) =          1.0000000000
X (   17) =           .0000000000
X (   18) =           .0000000000
X (   19) =           .0000000000
X (   20) =           .0000000000
X (   21) =           .0000000000
X (   22) =          1.0000000000
X (   23) =          1.0000000000
X (   24) =          1.0000000000
```

Figure D1 BIP evaluation of small supply chain.

Table D1 Result of small supply chain from excel (validated with BIP)

Variable	Value	Variable	Value
x_{11}	0 items	y_{11}	0 items
x_{12}	200 items	y_{12}	0 items
x_{21}	0 items	y_{13}	0 items
x_{22}	300 items	y_{21}	100 items
x_{31}	0 items	y_{22}	200 items
x_{32}	0 items	y_{23}	200 items
Zx_{11}	0	Zy_{11}	0
Zx_{12}	1	Zy_{12}	0
Zx_{13}	0	Zy_{13}	0
Zx_{14}	1	Zy_{14}	1
Zx_{15}	0	Zy_{15}	1
Zx_{16}	0	Zy_{16}	1

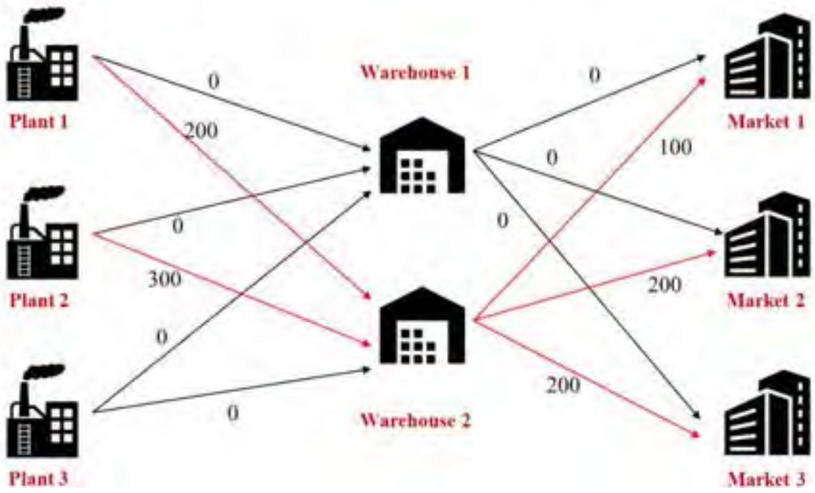


Figure D2 Result of small supply chain from excel (validated with BIP).

For MILP evaluation, the result from the developed algorithm shows in Fig. D3, and the result from excel shows in Table D2. The maximum total net present value is 225,000$.

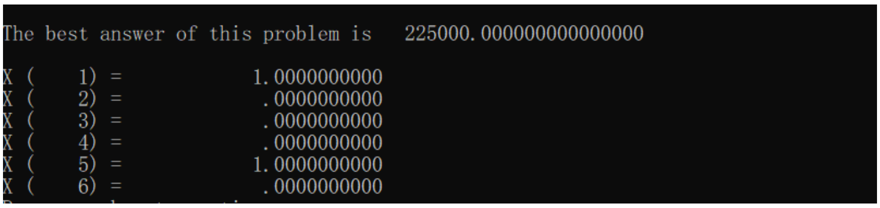


Figure D3 Result of the project problem from BIP algorithm.

Table D2 Result of the project problem from Microsoft excel

Variable	Value	Variable	Value
x_1	1	x_4	0
x_2	0	x_5	1
x_3	0	x_6	0

Appendix E Nomenclature For The Variables In Biofuel Production Supply Chain Case Study

Continuous variables

CRM	Cost of raw material
DP	Depreciation
EC	Equipment cost
FMPU	Flow of material between mobile pyrolyzer to upgrading plants
FPU	Flow of material between pyrolysis plant to upgrading plants
FPU1	Flow of material between small pyrolysis plant to upgrading plants
FPU2	Flow of material between large pyrolysis plant to upgrading plants
FSMP	Flow of material between suppliers to mobile pyrolyzer
FSP	Flow of material between suppliers to pyrolysis plant
FUC	Flow of material between upgrading plants to consumers
FUC1	Flow of material between small upgrading plants to consumers
FUC2	Flow of material between large upgrading plants to consumers
OL	Operating labour
TACF	Total annual cash flow
TAGP	Total annual gross profit
TANP	Total annual net profit
TC	Transportation cost
TPC	Total production cost
UT	Utility cost

Integer variables

TT	Total number of transportation trucks

Binary variables

y1	Existence of small plant
y2	Existence of large plant

Parameters

AV	Available biomass

bC	Biomass cost
cf	Conversion factor
c	Consumer
CT	Truck capacity
D	Demand
df	Depreciation factor
DS	Distance
DY	Working days per year
fp	Fast pyrolysis plant
INP1	Intercept equipment cost of pyrolysis plant vs small plant capacity
INP2	Intercept equipment cost of pyrolysis plant vs large plant capacity
INU1	Intercept equipment cost of upgrading plant vs small plant capacity
INU2	Intercept equipment cost of upgrading plant vs large plant capacity
LD	Labour truck driver
mp	Mobile pyrolyzer
MPP	Mobile pyrolyzer price
mSCLP	Minimum scale pyrolysis plant
mSCLU	Minimum scale upgrading plant
olf	Operating labour factor
pwf	Present worth factor
RMC	Raw material purchasing cost
RP	Truck rental price
s	Supplier
SCLP	Scale change from small to large in pyrolysis plant
SCLU	Scale change from small to large in upgrading plants
SCNPV	Supply Chain Net present value
SLP1	Slope equipment cost of pyrolysis plant vs small plant capacity
SLP2	Slope equipment cost of pyrolysis plant vs large plant capacity
SLU1	Slope equipment cost of upgrading plant vs small plant capacity
SLU2	Slope equipment cost of upgrading plant vs large plant capacity
smr	Supervision, maintenance and repairs
sp	Selling price

st1	Supplier type 1
st2	Supplier type 2
t	Year
utf	Utility cost factor
VC	Variable cost
VL	Average velocity
WH	Working hours per day
tax	Taxation rate
tcf	Transportation cost factor
u	Upgrading plant

Simbol

b	Biomass
gd	Gasoline + diesel
o	pyrolysis oil

Appendix F Result Of Biofuel Production Supply Chain

Result of biofuel production from MILP algorithm shows in Fig. F1.

```
The best answer of this problem is      317.481919799799100

X (     1) =          1080.4420413079
X (     2) =           898.1639848793
X (     3) =           369.9675152789
X (     4) =           459.8119724456
X (     5) =            28.3551703823
X (     6) =           495.2744328024
X (     7) =           192.3763379507
X (     8) =            17.9380400784
X (     9) =            27.6236620493
X (    10) =           476.6619599216
X (    11) =            14.6405230175
X (    12) =            29.2591072711
X (    13) =            43.0182909892
X (    14) =             3.6000000000
X (    15) =             1.4640523017
X (    16) =             2.9259107271
X (    17) =             4.3018290989
X (    18) =              .3600000000
X (    19) =            12.3570761312
X (    20) =           118.2944766381
X (    21) =           283.6430123619
X (    22) =             7.7978920506
X (    23) =            30.6897677543
X (    24) =             9.5042434475
X (    25) =            22.7889950425
X (    26) =             4.7781201291
X (    27) =              .9871056786
X (    28) =             1.4215569440
X (    29) =             3.4085673760
X (    30) =             9.5679643249
X (    31) =             1.1782868726
X (    32) =             2.5931762811
X (    33) =             6.2178417189
X (    34) =             1.6633094345
X (    35) =            32.8058050215
X (    36) =            13.4478988254
X (    37) =            32.2449757686
X (    38) =            15.5309956723
X (    39) =            40.7360437147
X (    40) =            13.6378392912
X (    41) =            32.6249291922
X (    42) =            16.7065457661
X (    43) =            47.0000000000
X (    44) =             1.0000000000
X (    45) =             2.0000000000
X (    46) =             7.0000000000
```

```
X (    47) =             4.8752386932
X (    48) =              .1249404658
X (    49) =              .2499534235
X (    50) =              .7205500938
X (    51) =           153.4961228752
X (    52) =           216.4713924037
X (    53) =              .0000000000
X (    54) =           459.8119724456
X (    55) =            81.0895198640
X (    56) =           111.2868180867
X (    57) =              .0000000000
X (    58) =            17.9380400784
X (    59) =              .0000000000
X (    60) =            27.6236620493
X (    61) =            81.0656768625
X (    62) =           395.5962830592
X (    63) =              .9993237166
X (    64) =              .0006762834
X (    65) =              .9998707751
X (    66) =              .0001292249
X (    67) =              .9995767801
X (    68) =              .0004232199
```

Figure F1 Result of biofuel production supply chain with MILP.

Table F1 Specification of x to supply chain variables

Mass Balance							
x_1	FSP_{11}	x_2	$FSMP_{11}$	x_3	FPU_{11}	x_4	FPU_{12}
x_5	$FMPU_{11}$	x_6	$FMPU_{12}$	x_7	FUC_{11}	x_8	FUC_{12}
x_9	FUC_{21}	x_{10}	FUC_{22}				
Equipment cost							
x_{11}	EC^{fp}	x_{12}	EC^{u1}	x_{13}	EC^{u2}	x_{14}	EC^{mp}
Depreciation							
x_{15}	DP^{fp}	x_{16}	DP^{u1}	x_{17}	DP^{u2}	x_{18}	DP^{mp}
Product price and Raw material cost							
x_{19}	vp^{fp}	x_{20}	vp^{u1}	x_{21}	vp^{u2}	x_{22}	vp^{mp}
x_{23}	CRM^{fp}	x_{24}	CRM^{u1}	x_{25}	CRM^{u2}	x_{26}	CRM^{mp}
Operating labour and Utility							
x_{27}	OL^{fp}	x_{28}	OL^{u1}	x_{29}	OL^{u2}	x_{30}	OL^{mp}
x_{31}	UT^{fp}	x_{32}	UT^{u1}	x_{33}	UT^{u2}	x_{34}	UT^{mp}
Variable cost and Total production cost							
x_{35}	VC^{fp}	x_{36}	VC^{u1}	x_{37}	VC^{u2}	x_{38}	VC^{mp}
x_{39}	TPC^{fp}	x_{40}	TPC^{u1}	x_{41}	TPC^{u2}	x_{42}	TPC^{mp}
Transportation truck and transportation cost							
x_{43}	TT^{fp}	x_{44}	TT^{u1}	x_{45}	TT^{u2}	x_{46}	TT^{mp}
x_{47}	TC^{fp}	x_{48}	TC^{u1}	x_{49}	TC^{u2}	x_{50}	TC^{mp}
Small and large scale of plant							
x_{51}	$FPU1_{1,1}$	x_{52}	$FPU2_{1,1}$	x_{53}	$FPU1_{1,2}$	x_{54}	$FPU2_{1,2}$
x_{55}	$FUC1_{1,1}$	x_{56}	$FUC2_{1,1}$	x_{57}	$FUC1_{1,2}$	x_{58}	$FUC2_{1,2}$
x_{59}	$FUC1_{2,1}$	x_{60}	$FUC2_{2,1}$	x_{61}	$FUC1_{2,2}$	x_{62}	$FUC2_{2,2}$
Binary for choosing one scale							
x_{63}	$y1^{fp}$	x_{64}	$y2^{fp}$	x_{65}	$y1^{u1}$	x_{66}	$y2^{u1}$
x_{67}	$y1^{u2}$	x_{68}	$y2^{u2}$				

From Microsoft excel evaluation, the total annual gross profit is 317.48 M\$ y^{-1}. The total supply chain net present value for 10 years in plant operating is 2,723.34 M\$. The results of mass flow rates and the transportation trucks show in Table F2 and Fig. F2.

Table F2 Biofuel production result from excel (validated with MILP)

Variable	Value	Variable	Value
FSP_{11}	1,080.44 $Mg\ d^{-1}$	$FMPU_{12}$	495.27 $Mg\ d^{-1}$
$FSMP_{11}$	898.16 $Mg\ d^{-1}$	FUC_{11}	192.37 $m^3\ d^{-1}$
FPU_{11}	369.96 $Mg\ d^{-1}$	FUC_{12}	17.93 $m^3\ d^{-1}$
FPU_{12}	459.81 $Mg\ d^{-1}$	FUC_{21}	27.62 $m^3\ d^{-1}$
$FMPU_{11}$	28.35 $Mg\ d^{-1}$	FUC_{22}	476.66 $m^3\ d^{-1}$
TTfp	47	TTu2	2
TTu1	1	TTmp	7

From Table F2 and Fig. F2, the transportation trucks from supplier 1 to the upgrading plants are greater than other ones because of high mass flow rate of biomass and pyrolysis oil. The transportation trucks from upgrading plant 1 and 2 are small, 1 and 2, because of low volume flow rate of biofuel. Transportation trucks from mobile pyrolyzer to upgrading plant are 7 trucks because of lower mass flow rate than one from supplier 1 to the upgrading plants, and higher mass flow rate than one from the both upgrading plants to the customers.

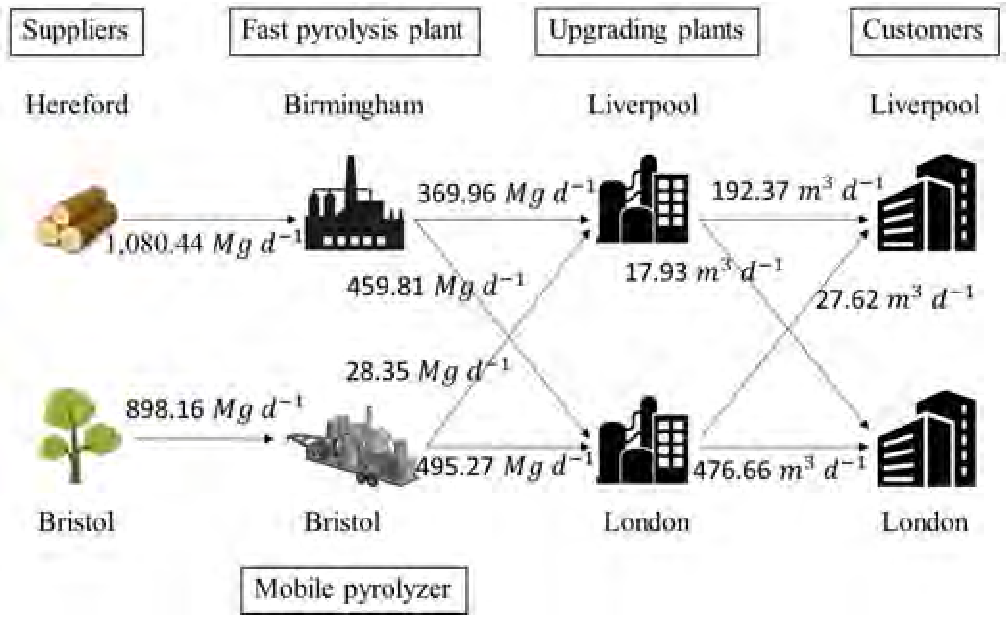


Figure F2 Biofuel production result from excel (validated with MILP).

Result of biofuel production from BIP algorithm shows in Fig. F3.

```
The best answer of this problem is      317.180041490127300

X (    1) =          1079.0482954545
X (    2) =           900.0000000000
X (    3) =           652.8030303030
X (    4) =           175.9060606061
X (    5) =              .0000000000
X (    6) =           524.7000000000
X (    7) =           220.0000000000
X (    8) =           124.6800000000
X (    9) =              .0000000000
X (   10) =           369.9200000000
X (   11) =            17.1734145455
X (   12) =            40.0690240000
X (   13) =            41.2502560000
X (   14) =             3.6000000000
X (   15) =             1.7173414545
X (   16) =             4.0069024000
X (   17) =             4.1250256000
X (   18) =              .3600000000
X (   19) =            12.3411357818
X (   20) =           193.8704362000
X (   21) =           208.0670528000
X (   22) =             7.8138324000
X (   23) =            30.6501786464
X (   24) =            15.5763132420
X (   25) =            16.7169252480
X (   26) =             4.7878875000
X (   27) =              .9858323345
X (   28) =             2.3297610560
X (   29) =             2.5003632640
X (   30) =             9.5875230330
X (   31) =             1.1767669091
X (   32) =             4.2499044000
X (   33) =             4.5611136000
X (   34) =             1.6667095500
X (   35) =            32.7634862733
X (   36) =            22.0394906452
X (   37) =            23.6533839488
X (   38) =            15.5627439313
X (   39) =            40.5324611026
X (   40) =            22.8181608452
X (   41) =            23.8327026688
X (   42) =            16.7081640313
X (   43) =            46.0497689792
X (   44) =             4.1019120000
X (   45) =              .9432960000
X (   46) =             6.8211000000
```

```
X (   47) =             4.7757398456
X (   48) =              .5120459200
X (   49) =              .1180044800
X (   50) =              .7020486000
X (   51) =              .0000000000
X (   52) =           652.8030303030
X (   53) =              .0000000000
X (   54) =           175.9060606061
X (   55) =              .0000000000
X (   56) =           220.0000000000
X (   57) =              .0000000000
X (   58) =           124.6800000000
X (   59) =              .0000000000
X (   60) =              .0000000000
X (   61) =              .0000000000
X (   62) =           369.9200000000
X (   63) =              .0000000000
X (   64) =             1.0000000000
X (   65) =              .0000000000
X (   66) =             1.0000000000
X (   67) =              .0000000000
X (   68) =             1.0000000000
```

Figure F3 Result of biofuel production supply chain with BIP.

From Microsoft excel evaluation, the total annual gross profit is 317.08 M\$ y^{-1}. The total supply chain net present value for 10 years in plant operating is 2,721.15 M\$. The results of mass flow rates and the transportation trucks show in Table F3.

Table F3 Biofuel production result from excel (validated with BIP)

Variable	Value	Variable	Value
FSP_{11}	1,079.05 $Mg\ d^{-1}$	$FMPU_{12}$	524.7 $Mg\ d^{-1}$
$FSMP_{11}$	900.00 $Mg\ d^{-1}$	FUC_{11}	220.0 $m^3\ d^{-1}$
FPU_{11}	700.6 $Mg\ d^{-1}$	FUC_{12}	149.9 $m^3\ d^{-1}$
FPU_{12}	128.10 $Mg\ d^{-1}$	FUC_{21}	0 $m^3\ d^{-1}$
$FMPU_{11}$	0 $Mg\ d^{-1}$	FUC_{22}	344.68 $m^3\ d^{-1}$
$y1^{fp}$	0	$y2^{u1}$	1
$y2^{fp}$	1	$y1^{u2}$	1
$y1^{u1}$	0	$y2^{u2}$	0

From Table F3, the decision variable for large scale of the fast pyrolysis plant is 1 because large amount of mass flow from the fast pyrolysis plant to the both upgrading plant, 828.70 $Mg\ d^{-1}$ of pyrolysis oil or 1,079.05 $Mg\ d^{-1}$ of the throughput biomass. The decision for small scale of the upgrading plant 2 is 1 because mass flow from it to the both customers is 344.68 $m^3\ d^{-1}$ of biofuel or 850 $Mg\ d^{-1}$ of the throughput biomass. So the small scale of the upgrading plant 2 is used. Finally, the decision variable for large scale of the upgrading plant 1 is 1 because mass flow from it to the both customers is 369.9 $m^3\ d^{-1}$ of biofuel or 912.2 $Mg\ d^{-1}$ of the throughput biomass. Moreover, the mass flow rate could be shown as node model, as shows in Fig. F4.

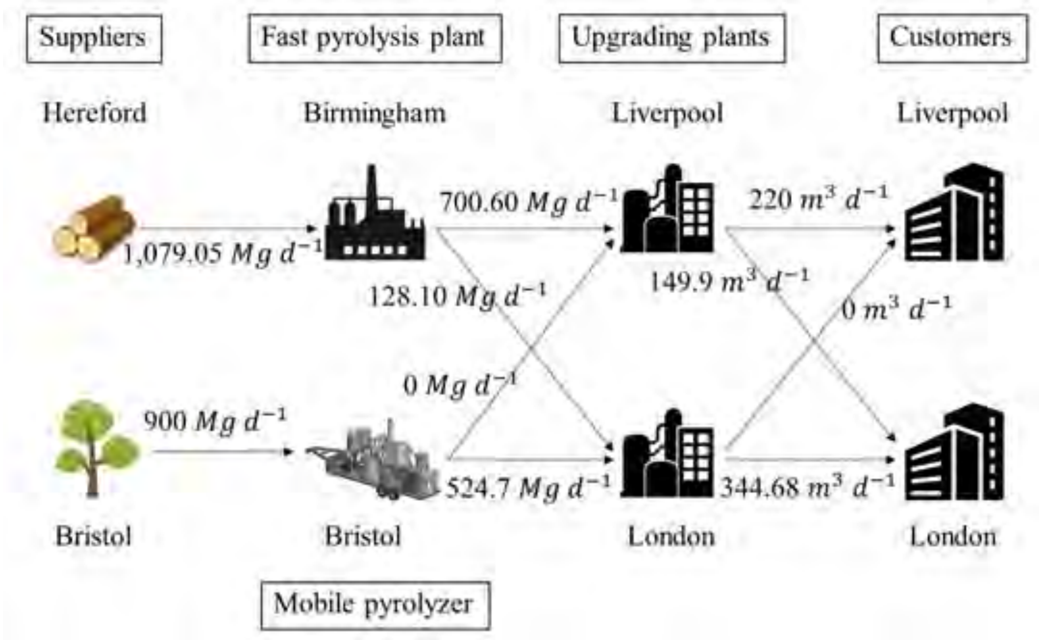


Figure F4 Biofuel production result from excel (validated with BIP).

From Fig. F4, the mass flow rate from mobile pyrolyzer to the upgrading plant 1 is zero because large scale of the fast pyrolysis plant is used. So the mass flow from the fast pyrolysis plant to the upgrading plant 1 is high enough. Moreover, the mass flow rate from the upgrading plant 2 to the customer 1 is zero because mass flow from the small scale of the upgrading plant 2 to customer 2 is used in many amounts. So, all of mass flow rates from the upgrading plant 2 are used for the customer 2.

Result of biofuel production from full MILP algorithm shows in Fig. F5.

```
The best answer of this problem is       316.973115141939700

X (    1) =        1079.6438137908
X (    2) =         899.2155092929
X (    3) =         677.1070930048
X (    4) =         152.0593559866
X (    5) =          27.1831845690
X (    6) =         522.3173805680
X (    7) =         215.1247029120
X (    8) =         156.7405636470
X (    9) =           4.8752970880
X (   10) =         337.8594363530
X (   11) =          17.1796803512
X (   12) =          41.3412944750
X (   13) =          39.9297494669
X (   14) =           3.6000000000
X (   15) =           1.7179680351
X (   16) =           4.1341294475
X (   17) =           3.9929749467
X (   18) =            .3600000000
X (   19) =          12.3479467584
X (   20) =         209.1611971551
X (   21) =         192.7762918449
X (   22) =           7.8070214234
X (   23) =          30.6670942408
X (   24) =          16.8048331082
X (   25) =          15.4884053818
X (   26) =           4.7837141075
X (   27) =            .9863764077
X (   28) =           2.5135117097
X (   29) =           2.3166126103
X (   30) =           9.5791660078
X (   31) =           1.1774163576
X (   32) =           4.5850987367
X (   33) =           4.2259192633
X (   34) =           1.6652567521
X (   35) =          32.7815681857
X (   36) =          23.7777679691
X (   37) =          21.9151066249
X (   38) =          15.5491785669
X (   39) =          40.5420662194
X (   40) =          24.7268813069
X (   41) =          22.1051618379
X (   42) =          16.7247250006
X (   43) =          46.0000000000
X (   44) =           5.0000000000
X (   45) =           1.0000000000
X (   46) =           7.0000000000
```

```
X (   47) =           4.7704980337
X (   48) =            .6241133378
X (   49) =            .1250552131
X (   50) =            .7205464336
X (   51) =            .0000000000
X (   52) =         677.1070930048
X (   53) =            .0000000000
X (   54) =         152.0593559866
X (   55) =            .0000000000
X (   56) =         215.1247029120
X (   57) =            .0000000000
X (   58) =         156.7405636470
X (   59) =           4.8752970879
X (   60) =            .0000000002
X (   61) =         337.8594363530
X (   62) =            .0000000000
X (   63) =            .0000000000
X (   64) =           1.0000000000
X (   65) =            .0000000000
X (   66) =           1.0000000000
X (   67) =           1.0000000000
X (   68) =            .0000000000
```

Figure F5 Result of biofuel production supply chain with full MILP.

From Microsoft excel evaluation, the total annual gross profit is 317.39 M$ y^{-1}. The total supply chain net present value for 10 years in plant operating is 2,723.54 M$. The results of mass flows, transportation trucks and the decision variables show in Table F4.

From Table F4, the transportation trucks from supplier 1 to the both upgrading plants are greater than the others because of high mass flow rate of biomass and pyrolysis oil, 1,910.22 $Mg\,d^{-1}$. The transportation trucks from upgrading plant 1 and 2 are small, 1 and 2, because of low volume flow rate of biofuel. Transportation trucks from mobile pyrolyzer to upgrading plant are 7 trucks because of lower mass flow rate than one from supplier 1 to the upgrading plants, and higher mass flow rate than the one from the both upgrading plants to the customers. Moreover, the results would be shown as node model in Fig. F6.

Table F4 Biofuel production result from excel (validated with full MILP)

Variable	Value	Variable	Value
FSP_{11}	1,080.44 $Mg\,d^{-1}$	$FMPU_{12}$	495.28 $Mg\,d^{-1}$
$FSMP_{11}$	898.16 $Mg\,d^{-1}$	FUC_{11}	192.38 $m^3\,d^{-1}$
FPU_{11}	369.97 $Mg\,d^{-1}$	FUC_{12}	17.94 $m^3\,d^{-1}$
FPU_{12}	459.81 $Mg\,d^{-1}$	FUC_{21}	27.62 $m^3\,d^{-1}$
$FMPU_{11}$	28.35 $Mg\,d^{-1}$	FUC_{22}	476.66 $m^3\,d^{-1}$
Integer variables			
TTfp	47	TTu2	2
TTu1	1	TTmp	7
Binary variables			
$y1^{fp}$	0	$y2^{u1}$	0
$y2^{fp}$	1	$y1^{u2}$	0
$y1^{u1}$	1	$y2^{u2}$	1

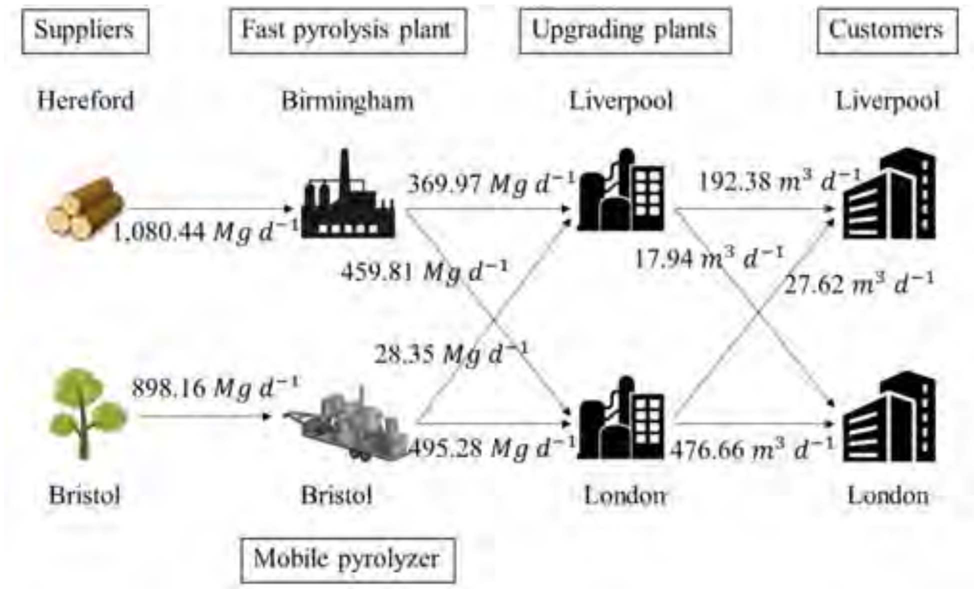


Figure F6 Biofuel production result from excel (validated with full MILP).

From Fig. F6, the mass flow rate from mobile pyrolyzer to the upgrading plant 1 is small because the upgrading plant 1 is small scale. The mass flow rate from the upgrading plant 1 to the customer 2 is little because of small scale of the upgrading plant 1. Most of the flow rates from the upgrading plant 1 are sent to customer 1. Finally, the mass flow rate from the upgrading plant 2 to the customer 1 is very small because of high demand of customer 2. So, all of the mass flow rates from the upgrading plant 2 are used for the customer 2.

Appendix G Additional Complex Supply Chain Case

For more complex case, the complex supply chain is used to test simplex and MILP algorithm. The supply chain consists of 5 plants, 6 warehouse and 2 markets, as shown in Fig. G1.

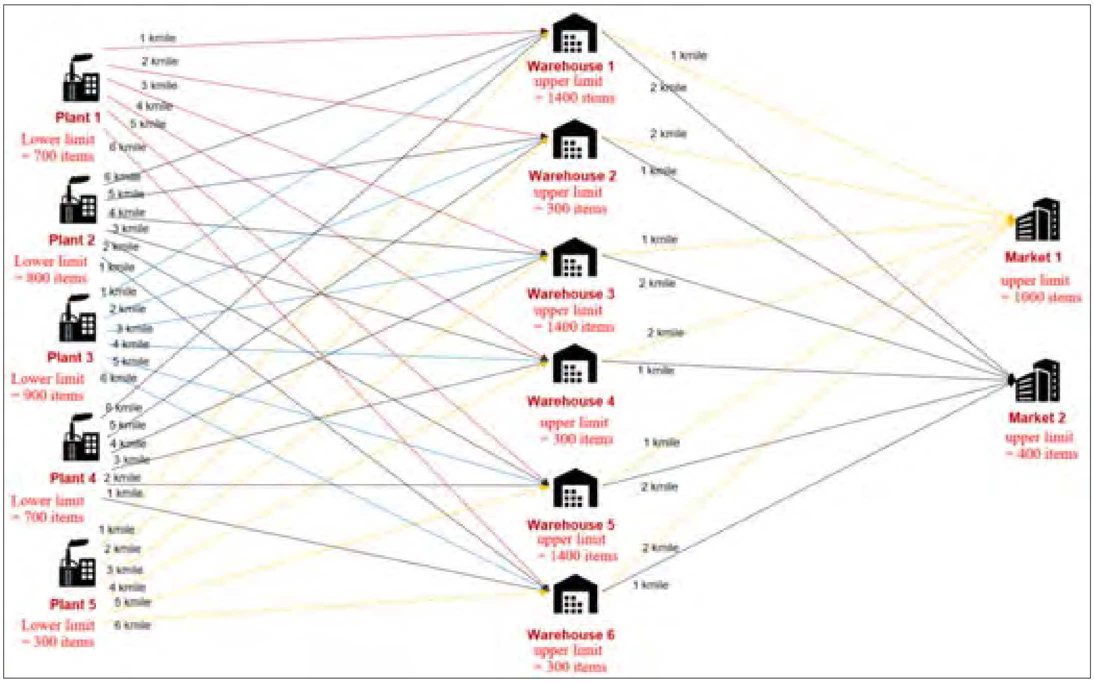


Figure G1 Detail of the supply chain 5-6-2.

For this case, the profit is optimized with the difference of revenue and transportation cost. The mass flow is identified into $x_{i,j}$ *and* $y_{j,k}$, when i is number of plants, j is number of warehouse and k is number of markets, product price is 1,000 $ per item and transportation cost is 90 $ per kmile per item, as shown in Fig. G2.

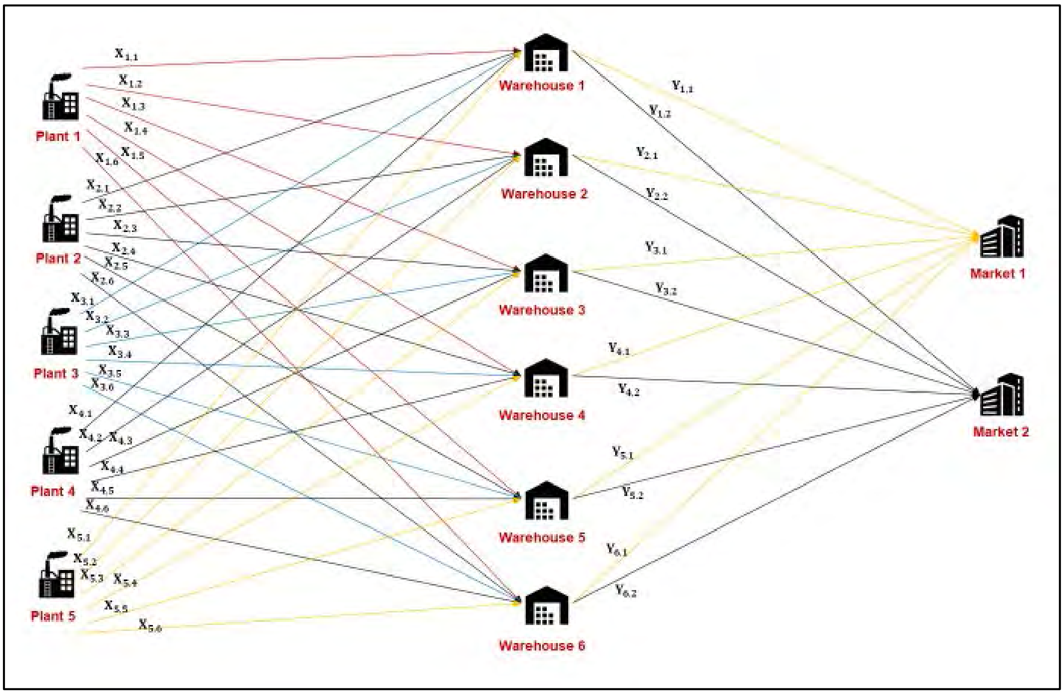


Figure G2 Variables of mass flow in supply chain 5-6-2.

All variables will be used as 'x' variables, which are identified in Fig. G3.

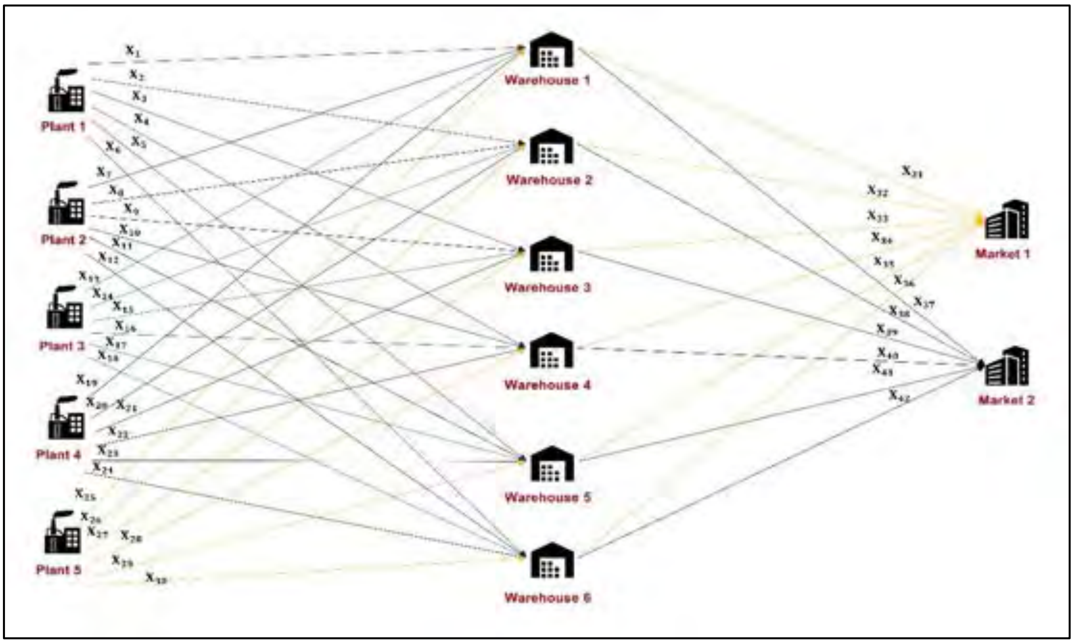


Figure G3 All 'x' variables of mass flow in supply chain 5-6-2.

Product price is x_{43}

Transportation cost (TC) is x_{44}

The objective function

Max Profit = Revenue – Transportation cost

Constraint (supplier limit)

1) $x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} \leq 700$

2) $x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} \leq 800$

3) $x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} \leq 900$

4) $x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,6} \leq 700$

5) $x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,6} \leq 300$

Constraint (Demand limit)

6) $y_{1,1} + y_{2,1} + y_{3,1} + y_{4,1} + y_{5,1} + y_{6,1} \geq 1{,}000$

7) $y_{1,2} + y_{2,2} + y_{3,2} + y_{4,2} + y_{5,2} + y_{6,2} \geq 400$

Constraint (Warehouse limit)

8) $x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} \leq 1{,}400$

9) $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} \leq 300$

10) $x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} \leq 1{,}400$

11) $x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} \leq 300$

12) $x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} \leq 1{,}400$

13) $x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} \leq 300$

Constraint (Demand and Supplier balance)

14) $x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} = y_{1,1} + y_{1,2}$

15) $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} = y_{2,1} + y_{2,2}$

16) $x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} = y_{3,1} + y_{3,2}$

17) $x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} = y_{4,1} + y_{4,2}$

18) $x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} = y_{5,1} + y_{5,2}$

19) $x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} = y_{6,1} + y_{6,2}$

Constraint (Product Price)

20) Product price = 1,000($y_{1,1} + y_{2,1} + y_{3,1} + y_{4,1} + y_{5,1} + y_{6,1} + y_{1,2} + y_{2,2} + y_{3,2} + y_{4,2} + y_{5,2} + y_{6,2}$)

Constraint (Transportation Cost, TC)

21) TC = $90x_{1,1} + 180x_{1,2} + 270x_{1,3} + 360x_{1,4} + 450x_{1,5} + 540x_{1,6} +$
$540x_{2,1} + 450x_{2,2} + 360x_{2,3} + 270x_{2,4} + 180x_{2,5} + 90x_{2,6} +$
$90x_{3,1} + 180x_{3,2} + 270x_{3,3} + 360x_{3,4} + 450x_{3,5} + 540x_{3,6} +$
$540x_{4,1} + 450x_{4,2} + 360x_{4,3} + 270x_{4,4} + 180x_{4,5} + 90x_{4,6} +$
$90x_{5,1} + 180x_{5,2} + 270x_{5,3} + 360x_{5,4} + 450x_{5,5} + 540x_{5,6} +$
$90y_{1,1} + 180y_{2,1} + 90y_{3,1} + 180y_{4,1} + 90y_{5,1} + 180y_{6,1} +$
$180y_{1,2} + 90y_{2,2} + 180y_{3,2} + 90y_{4,2} + 180y_{5,2} + 90y_{6,2}$

The optimum profit is 2,617,000 \$. The mass flow results are shown in Fig. G4. The source code used to solve show in Fig. G5 – G9 and the results shows in Fig. G10.

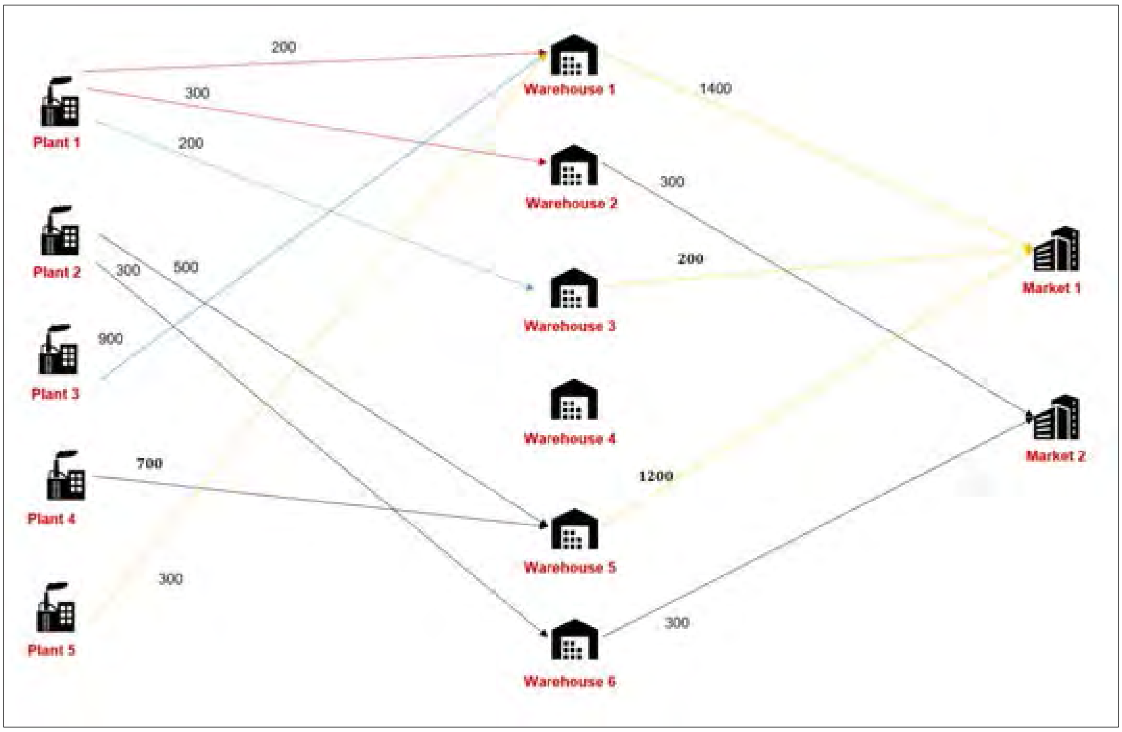


Figure G4 Results of supply chain 5-6-2 by simplex algorithm (Case 1).

Figure G5 Main source code of algorithm.

The main source code is used to input values from notepad by using subroutine of inputting value. The subroutine of inputting value, values on notepad and simplex source code show in Fig. G6 – G8.

Figure G6 Subroutine of inputting values.

Simplex evaluation 5 6 2

```
1
22
44
Objective
2
43,1
44,-1
0
Constraint1
6
1,1
2,1
3,1
4,1
5,1
6,1
1
700
Constraint2
6
7,1
8,1
9,1
10,1
11,1
12,1
1
800
Constraint3
6
13,1
14,1
15,1
16,1
17,1
18,1
1
900
Constraint4
6
19,1
20,1
21,1
22,1
23,1
24,1
1
700
Constraint5
6
25,1
26,1
```

27,1
28,1
29,1
30,1
1
300
Constraint6
6
31,1
32,1
33,1
34,1
35,1
36,1
2
1000
Constraint7
6
37,1
38,1
39,1
40,1
41,1
42,1
2
400
Constraint8
5
1,1
7,1
13,1
19,1
25,1
1
1400
Constraint9
5
2,1
8,1
14,1
20,1
26,1
1
300
Constraint10
5
3,1
9,1
15,1
21,1
27,1
1

```
1400
Constraint11
5
4,1
10,1
16,1
22,1
28,1
1
300
Constraint12
5
5,1
11,1
17,1
23,1
29,1
1
1400
Constraint13
5
6,1
12,1
18,1
24,1
30,1
1
300
Constraint14
7
1,1
7,1
13,1
19,1
25,1
31,-1
37,-1
3
0
Constraint15
7
2,1
8,1
14,1
20,1
26,1
32,-1
38,-1
3
0
Constraint16
7
```

3,1
9,1
15,1
21,1
27,1
33,-1
39,-1
3
0
Constraint17
7
4,1
10,1
16,1
22,1
28,1
34,-1
40,-1
3
0
Constraint18
7
5,1
11,1
17,1
23,1
29,1
35,-1
41,-1
3
0
Constraint19
7
6,1
12,1
18,1
24,1
30,1
36,-1
42,-1
3
0
Constraint20
13
31,-1000
32,-1000
33,-1000
34,-1000
35,-1000
36,-1000
37,-1000
38,-1000

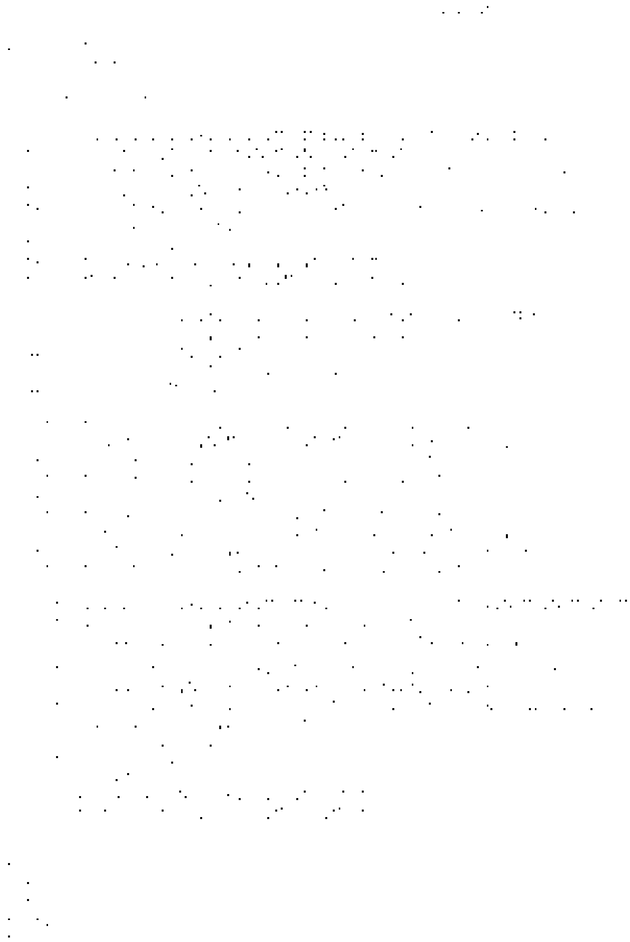
Simplex evaluation-5-6-2

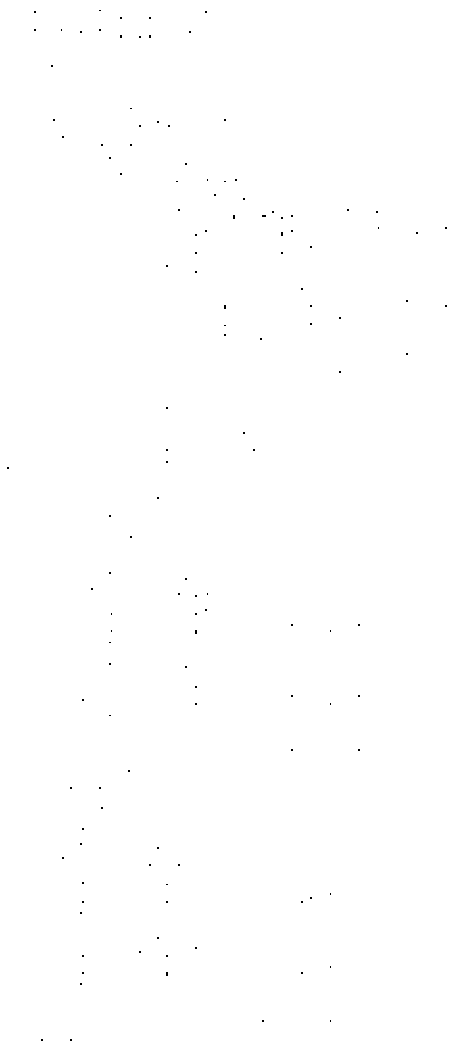
```
39,-1000
40,-1000
41,-1000
42,-1000
43,1
3
0
Constraint21
43
1,-90
2,-180
3, 270
4,-360
5,-450
6,-540
7,-540
8,-450
9,-360
10,-270
11,-180
12,-90
13, 90
14,-180
15,-270
16, 360
17,-450
18, 540
19,-540
20,-450
21,-360
22,-270
23,-180
24,-90
25,-90
26,-180
27,-270
28, 360
29, 450
30,-540
31, 90
32, 180
33,-90
34,-180
35,-90
36,-180
37,-180
38,-90
39,-180
40,-90
41,-180
42, 90
44,1
```

Page 5

Figure G7 The values of problem 5-6-2 on notepad.

In the first line, number 1 means maximization and number 2 means minimization. The second line is number of equations (including objective function and constraints). And the third line is number of variables. For variable values, coefficient is input with its number such as constraint 1 has 6 variables and variable number 1 is 1 until variable number 6 is 1. The equation is less than or equal to 700. Then the simplex subroutine is called to solve the problem. And The result shows in Fig. G9.





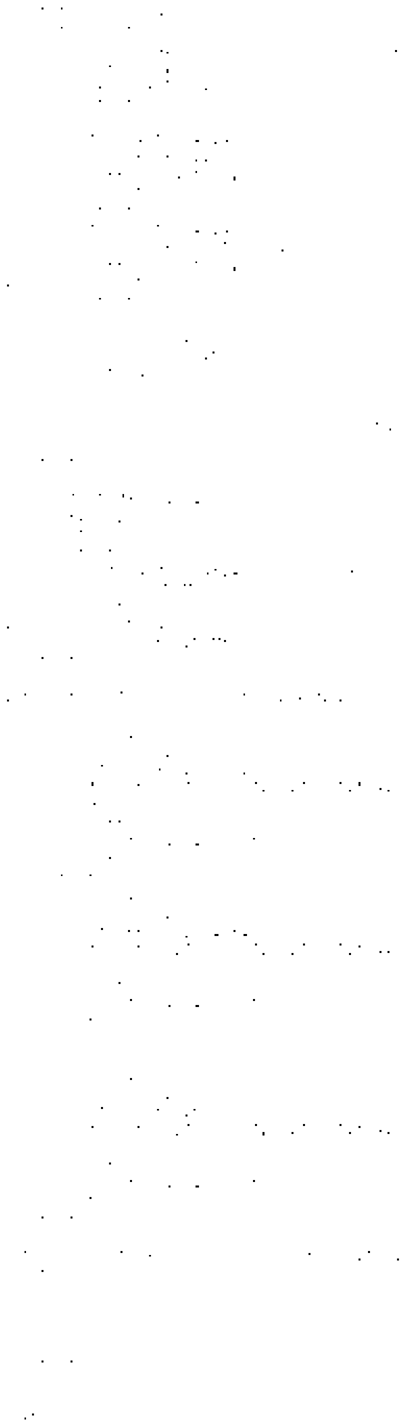


Figure G8 Simplex subroutine.

```
X (    1) =           200.0000000000
X (    2) =           300.0000000000
X (    3) =           200.0000000000
X (    4) =              .0000000000
X (    5) =              .0000000000
X (    6) =              .0000000000
X (    7) =              .0000000000
X (    8) =              .0000000000
X (    9) =              .0000000000
X (   10) =              .0000000000
X (   11) =           500.0000000000
X (   12) =           300.0000000000
X (   13) =           900.0000000000
X (   14) =              .0000000000
X (   15) =              .0000000000
X (   16) =              .0000000000
X (   17) =              .0000000000
X (   18) =              .0000000000
X (   19) =              .0000000000
X (   20) =              .0000000000
X (   21) =              .0000000000
X (   22) =              .0000000000
X (   23) =           700.0000000000
X (   24) =              .0000000000
X (   25) =           300.0000000000
X (   26) =              .0000000000
X (   27) =              .0000000000
X (   28) =              .0000000000
X (   29) =              .0000000000
X (   30) =              .0000000000
X (   31) =          1400.0000000000
X (   32) =              .0000000000
X (   33) =           200.0000000000
X (   34) =              .0000000000
X (   35) =          1200.0000000000
X (   36) =              .0000000000
X (   37) =              .0000000000
X (   38) =           300.0000000000
X (   39) =              .0000000000
X (   40) =              .0000000000
X (   41) =              .0000000000
X (   42) =           300.0000000000
X (   43) =       3400000.0000000000
X (   44) =        783000.0000000000
The maximal value of the problem is      2617000.0000000000
Complete
```

Figure G9 Results of supply chain 5-6-2 from the program.

Moreover, the alternative solutions from program show the source code in Fig. G10.

Figure G10 Subroutine source code for alternative solution.

From this case, 2 other solutions can be obtained from the developed algorithm, as shown in Fig. G11 (case 2) and G12 (case 3). The first most negative variable is used to show result in case 1 and the second most negative variable is used to show result in case 2. Finally, the zero non-basic variable is used to be entering variable and show result in case 3.

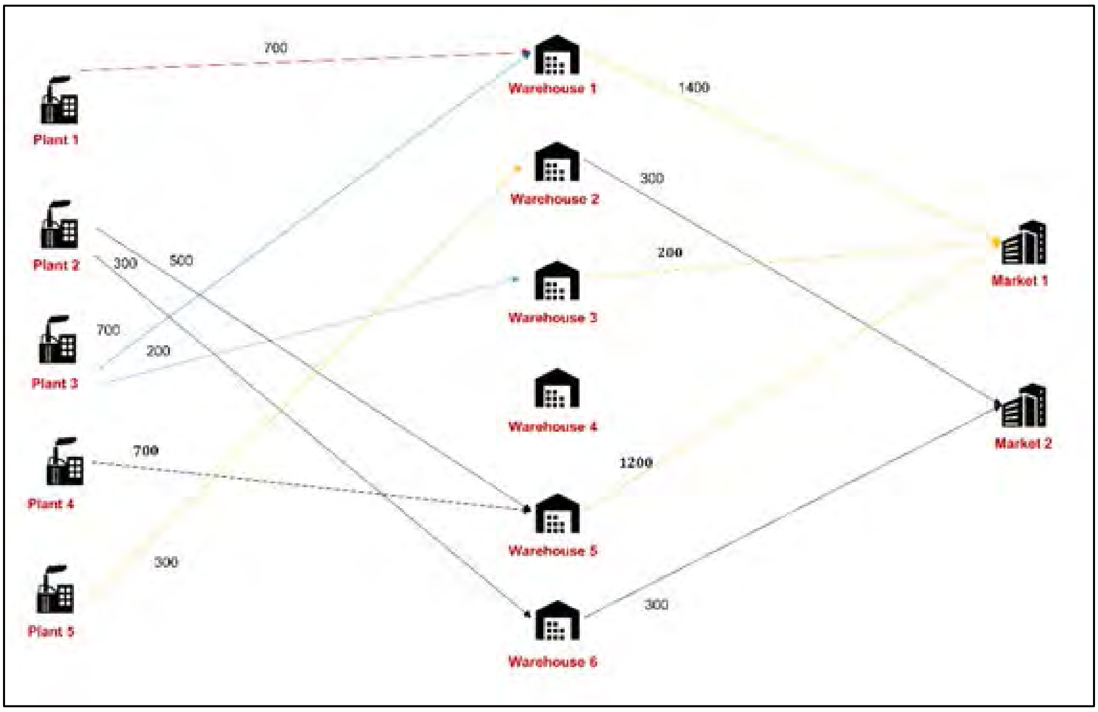


Figure G11 Other results of supply chain 5-6-2 by simplex algorithm (Case 2).

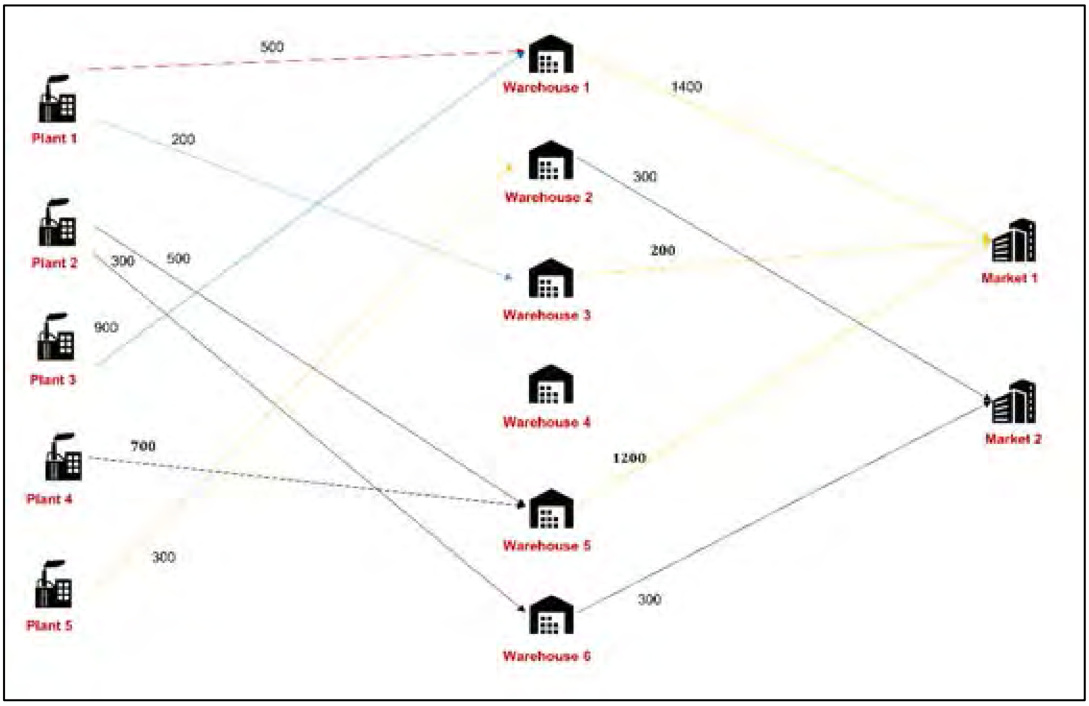


Figure G12 Other results of supply chain 5-6-2 by simplex algorithm (Case 3).

Moreover, this case is solved as MILP case with MILP algorithm. The transportation lines are minimized. Binary variables are identified as $Zx_{i,j}$ and $Zy_{j,k}$, as shown in Fig. G13 and G14.

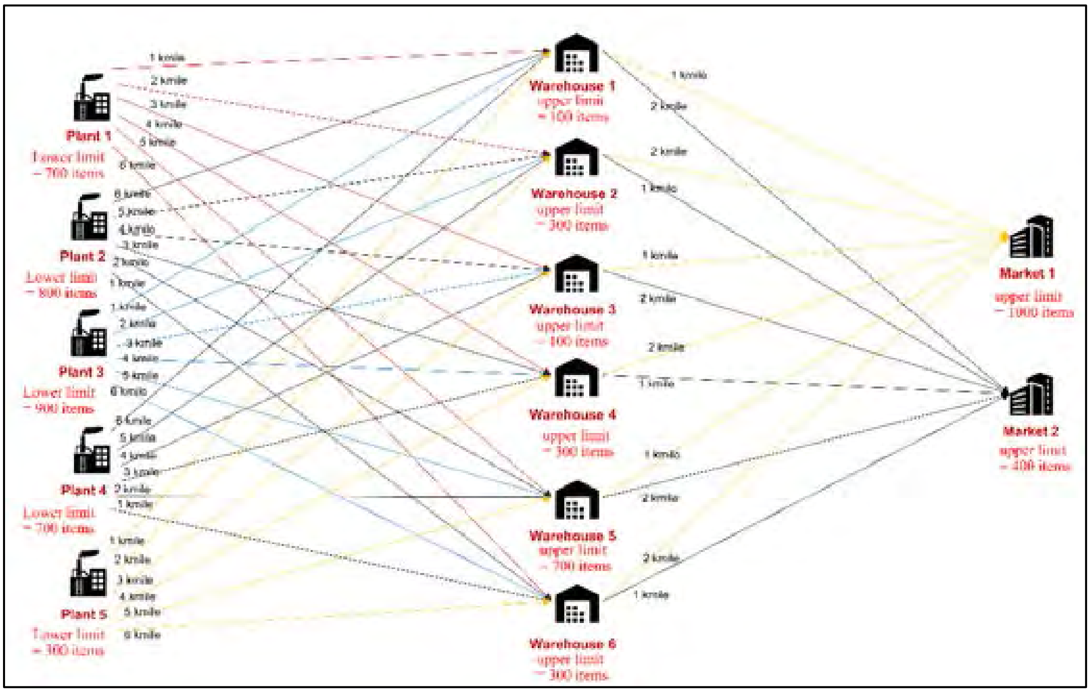


Figure G13 Detail of the supply chain 5-6-2 MILP.

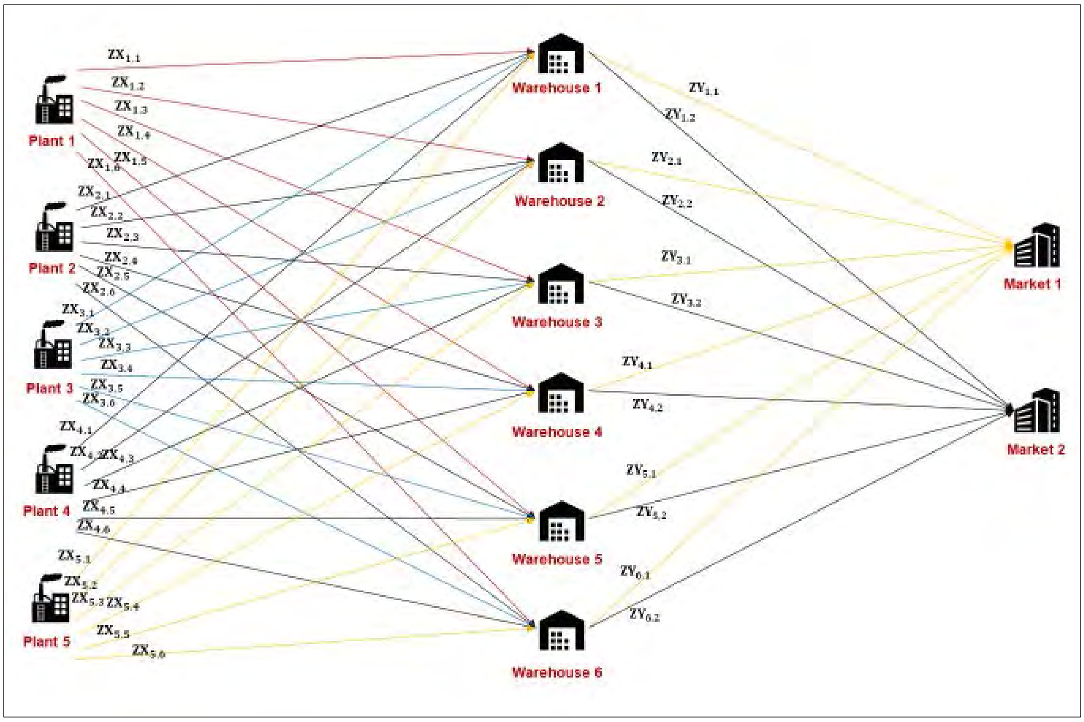


Figure G14 Binary variables in supply chain 5-6-2.

All binary variables will be used as ‘x’ variables, which are identified in Fig. G15.

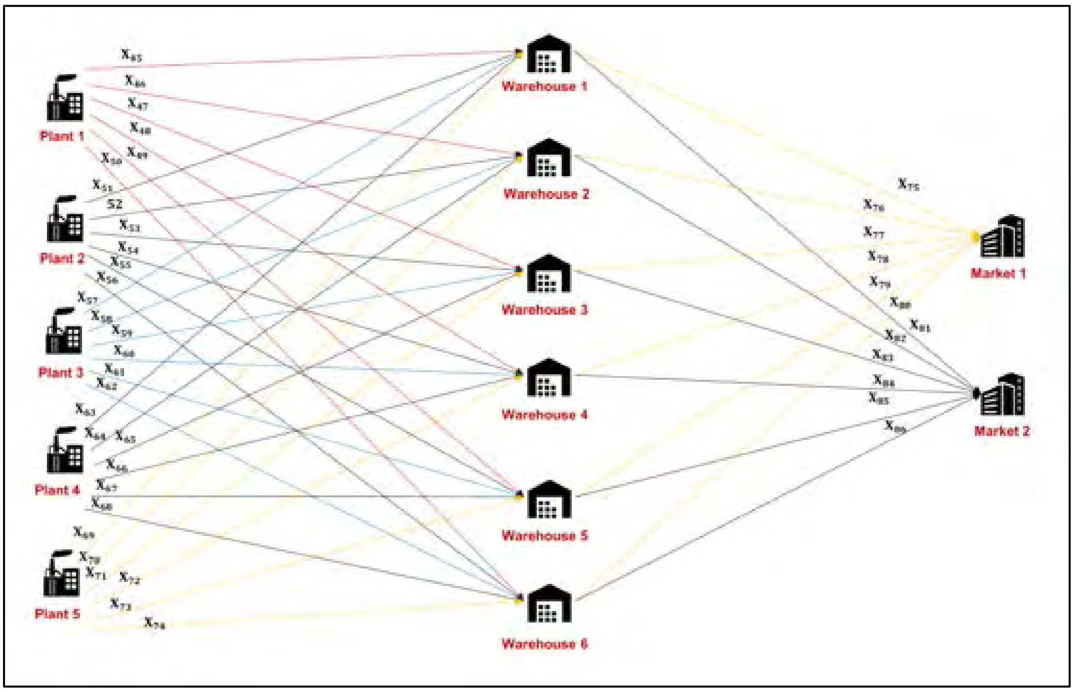


Figure G15 All ‘x’ variables of binary variables in supply chain 5-6-2.

The objective function is changed to summation of transportation lines.

<u>The objective function</u>

Min Transportation line = $Zx_{1,1} + Zx_{1,2} + Zx_{1,3} + Zx_{1,4} + Zx_{1,5} + Zx_{1,6} +$

$$Zx_{2,1} + Zx_{2,2} + Zx_{2,3} + Zx_{2,4} + Zx_{2,5} + Zx_{2,6} +$$

$$Zx_{3,1} + Zx_{3,2} + Zx_{3,3} + Zx_{3,4} + Zx_{3,5} + Zx_{3,6} +$$

$$Zx_{4,1} + Zx_{4,2} + Zx_{4,3} + Zx_{4,4} + Zx_{4,5} + Zx_{4,6} +$$

$$Zx_{5,1} + Zx_{5,2} + Zx_{5,3} + Zx_{5,4} + Zx_{5,5} + Zx_{5,6} +$$

$$Zy_{1,1} + Zy_{2,1} + Zy_{3,1} + Zy_{4,1} + Zy_{5,1} + Zy_{6,1} +$$

$$Zy_{1,2} + Zy_{2,2} + Zy_{3,2} + Zy_{4,2} + Zy_{5,2} + Zy_{6,2}$$

Constraint (supplier limit)

1) $x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} \leq 700$

2) $x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} \leq 800$

3) $x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} \leq 900$

4) $x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,6} \leq 700$

5) $x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,6} \leq 300$

Constraint (Demand limit)

6) $y_{1,1} + y_{2,1} + y_{3,1} + y_{4,1} + y_{5,1} + y_{6,1} \geq 1{,}000$

7) $y_{1,2} + y_{2,2} + y_{3,2} + y_{4,2} + y_{5,2} + y_{6,2} \geq 400$

Constraint (Warehouse limit)

8) $x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} \leq 100$

9) $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} \leq 300$

10) $x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} \leq 100$

11) $x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} \leq 300$

12) $x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} \leq 700$

13) $x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} \leq 300$

Constraint (Demand and Supplier balance)

14) $x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} = y_{1,1} + y_{1,2}$

15) $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} = y_{2,1} + y_{2,2}$

16) $x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} = y_{3,1} + y_{3,2}$

17) $x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} = y_{4,1} + y_{4,2}$

18) $x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} = y_{5,1} + y_{5,2}$

19) $x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} = y_{6,1} + y_{6,2}$

Constraint (Product Price)

20) Product price = 1,000($y_{1,1} + y_{2,1} + y_{3,1} + y_{4,1} + y_{5,1} + y_{6,1} + y_{1,2} + y_{2,2} + y_{3,2} + y_{4,2} + y_{5,2} + y_{6,2}$)

Constraint (Transportation Cost, TC)

21) TC = $90x_{1,1} + 180x_{1,2} + 270x_{1,3} + 360x_{1,4} + 450x_{1,5} + 540x_{1,6} +$
$540x_{2,1} + 450x_{2,2} + 360x_{2,3} + 270x_{2,4} + 180x_{2,5} + 90x_{2,6} +$
$90x_{3,1} + 180x_{3,2} + 270x_{3,3} + 360x_{3,4} + 450x_{3,5} + 540x_{3,6} +$
$540x_{4,1} + 450x_{4,2} + 360x_{4,3} + 270x_{4,4} + 180x_{4,5} + 90x_{4,6} +$
$90x_{5,1} + 180x_{5,2} + 270x_{5,3} + 360x_{5,4} + 450x_{5,5} + 540x_{5,6} +$
$90y_{1,1} + 180y_{2,1} + 90y_{3,1} + 180y_{4,1} + 90y_{5,1} + 180y_{6,1} +$
$180y_{1,2} + 90y_{2,2} + 180y_{3,2} + 90y_{4,2} + 180y_{5,2} + 90y_{6,2}$

And the constraints for binary variables are added. To create the binary variables, the constraint must be in form $x \leq Mz$ and $y \leq Mz$, when x and y are continuous variables, z are binary and M is big value.

22) $x_{1,1} \leq 10{,}000Zx_{1,1}$
23) $x_{1,2} \leq 10{,}000Zx_{1,2}$
24) $x_{1,3} \leq 10{,}000Zx_{1,3}$
25) $x_{1,4} \leq 10{,}000Zx_{1,4}$
26) $x_{1,5} \leq 10{,}000Zx_{1,5}$
27) $x_{1,6} \leq 10{,}000Zx_{1,6}$

28) $x_{2,1} \leq 10{,}000Zx_{2,1}$
29) $x_{2,2} \leq 10{,}000Zx_{2,2}$
30) $x_{2,3} \leq 10{,}000Zx_{2,3}$

31) $x_{2,4} \leq 10{,}000Zx_{2,4}$

32) $x_{2,5} \leq 10{,}000Zx_{2,5}$

33) $x_{2,6} \leq 10{,}000Zx_{2,6}$

34) $x_{3,1} \leq 10{,}000Zx_{3,1}$

35) $x_{3,2} \leq 10{,}000Zx_{3,2}$

36) $x_{3,3} \leq 10{,}000Zx_{3,3}$

37) $x_{3,4} \leq 10{,}000Zx_{3,4}$

38) $x_{3,5} \leq 10{,}000Zx_{3,5}$

39) $x_{3,6} \leq 10{,}000Zx_{3,6}$

40) $x_{4,1} \leq 10{,}000Zx_{4,1}$

41) $x_{4,2} \leq 10{,}000Zx_{4,2}$

42) $x_{4,3} \leq 10{,}000Zx_{4,3}$

43) $x_{4,4} \leq 10{,}000Zx_{4,4}$

44) $x_{4,5} \leq 10{,}000Zx_{4,5}$

45) $x_{4,6} \leq 10{,}000Zx_{4,6}$

46) $x_{5,1} \leq 10{,}000Zx_{5,1}$

47) $x_{5,2} \leq 10{,}000Zx_{5,2}$

48) $x_{5,3} \leq 10{,}000Zx_{5,3}$

49) $x_{5,4} \leq 10{,}000Zx_{5,4}$

50) $x_{5,5} \leq 10{,}000Zx_{5,5}$

51) $x_{5,6} \leq 10{,}000Zx_{5,6}$

52) $y_{1,1} \leq 10{,}000Zy_{1,1}$

53) $y_{2,1} \leq 10{,}000Zy_{2,1}$

54) $y_{3,1} \leq 10{,}000Zy_{3,1}$

55) $y_{4,1} \leq 10{,}000Zy_{4,1}$

56) $y_{5,1} \leq 10{,}000Zy_{5,1}$

57) $y_{6,1} \leq 10{,}000Zy_{6,1}$

58) $y_{1,2} \leq 10{,}000Zy_{1,2}$

59) $y_{2,2} \leq 10{,}000Zy_{2,2}$

60) $y_{3,2} \leq 10{,}000Zy_{3,2}$

61) $y_{4,2} \leq 10{,}000Zy_{4,2}$

62) $y_{5,2} \leq 10{,}000Zy_{5,2}$

63) $y_{6,2} \leq 10{,}000Zy_{6,2}$

From the solutions, the minimum transportation line = 8 and profit is 780,000. The mass flow solution is shown in Fig. G16. The source code to solve the MILP problem shows in Fig. G17 – G20. Finally, the result shows in Fig. G21.

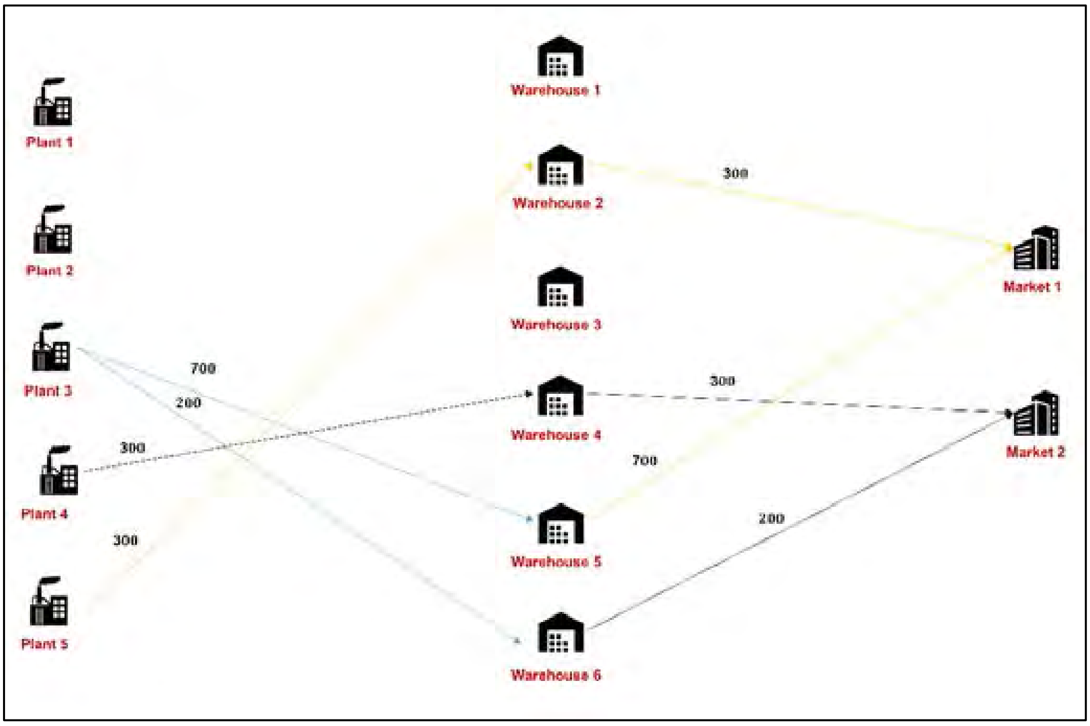


Figure G16 Mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 1).

Figure G17 Main source code for MILP.

The main source code is used to input values from notepad and optimize the solution by branch-and-bound after obtain continuous solution from simplex routine. The subroutine of inputting value, the values of the problem on notepad and simplex subroutine show in Fig. G18 – G20.

Figure G18 Subroutine of inputting value (MILP case).

2
64
86
Objective
42
45,1
46,1
47,1
48,1
49,1
50,1
51,1
52,1
53,1
54,1
55,1
56,1
57,1
58,1
59,1
60,1
61,1
62,1
63,1
64,1
65,1
66,1
67,1
68,1
69,1
70,1
71,1
72,1
73,1
74,1
75,1
76,1
77,1
78,1
79,1
80,1
81,1
82,1
83,1
84,1
85,1
86,1
0
Constraint1
6
1,1
2,1

3,1
4,1
5,1
6,1
1
700
Constraint2
6
7,1
8,1
9,1
10,1
11,1
12,1
1
800
Constraint3
6
13,1
14,1
15,1
16,1
17,1
18,1
1
900
Constraint4
6
19,1
20,1
21,1
22,1
23,1
24,1
1
700
Constraint5
6
25,1
26,1
27,1
28,1
29,1
30,1
1
300
Constraint6
6
31,1
32,1
33,1
34,1

```
35,1
36,1
2
1000
Constraint7
6
37,1
38,1
39,1
40,1
41,1
42,1
2
400
Constraint8
5
1,1
7,1
13,1
19,1
25,1
1
100
Constraint9
5
2,1
8,1
14,1
20,1
26,1
1
300
Constraint10
5
3,1
9,1
15,1
21,1
27,1
1
100
Constraint11
5
4,1
10,1
16,1
22,1
28,1
1
300
Constraint12
5
```

5,1
11,1
17,1
23,1
29,1
1
700
Constraint13
5
6,1
12,1
18,1
24,1
30,1
1
300
Constraint14
7
1,1
7,1
13,1
19,1
25,1
31,-1
37,-1
3
0
Constraint15
7
2,1
8,1
14,1
20,1
26,1
32,-1
38,-1
3
0
Constraint16
7
3,1
9,1
15,1
21,1
27,1
33,-1
39,-1
3
0
Constraint17
7
4,1

```
10,1
16,1
22,1
28,1
34,-1
40,-1
3
0
Constraint18
7
5,1
11,1
17,1
23,1
29,1
35, 1
41, 1
3
0
Constraint19
7
6,1
12,1
18,1
24,1
30,1
36,-1
42,-1
3
0
Constraint20
13
31, 1000
32,-1000
33,-1000
34,-1000
35,-1000
36,-1000
37,-1000
38,-1000
39,-1000
40,-1000
41, 1000
42,-1000
43,1
3
0
Constraint21
43
1, 90
2,-180
3,-270
```

```
4,-360
5,-450
6, 540
7,-540
8,-450
9,-360
10,-270
11,-180
12,-90
13,-90
14,-180
15,-270
16, 360
17,-450
18,-540
19, 540
20, 450
21,-360
22,-270
23, 180
24,-90
25,-90
26,-180
27,-270
28,-360
29,-450
30,-540
31,-90
32,-180
33, 90
34,-180
35,-90
36, 180
37,-180
38,-90
39,-180
40,-90
41,-180
42,-90
44,1
3
0
Constraint22
2
1,1
45, 10000
1
0
Constraint23
2
2,1
46,-10000
```

```
1
0
Constraint24
2
3,1
47,-10000
1
0
Constraint25
2
4,1
48,-10000
1
0
Constraint26
2
5,1
49,-10000
1
0
Constraint27
2
6,1
50,-10000
1
0
Constraint28
2
7,1
51,-10000
1
0
Constraint29
2
8,1
52,-10000
1
0
Constraint30
2
9,1
53,-10000
1
0
Constraint31
2
10,1
54,-10000
1
0
Constraint32
2
```

```
11,1
55,-10000
1
0
Constraint33
2
12,1
56,-10000
1
0
Constraint34
2
13,1
57,-10000
1
0
Constraint35
2
14,1
58,-10000
1
0
Constraint36
2
15,1
59,-10000
1
0
Constraint37
2
16,1
60,-10000
1
0
Constraint38
2
17,1
61,-10000
1
0
Constraint39
2
18,1
62,-10000
1
0
Constraint40
2
19,1
63,-10000
1
0
```

```
Constraint41
2
20,1
64,-10000
1
0
Constraint42
2
21,1
65,-10000
1
0
Constraint43
2
22,1
66,-10000
1
0
Constraint44
2
23,1
67,-10000
1
0
Constraint45
2
24,1
68,-10000
1
0
Constraint46
2
25,1
69,-10000
1
0
Constraint47
2
26,1
70,-10000
1
0
Constraint48
2
27,1
71,-10000
1
0
Constraint49
2
28,1
72,-10000
```

```
1
0
Constraint50
2
29,1
73,-10000
1
0
Constraint51
2
30,1
74,-10000
1
0
Constraint52
2
31,1
75,-10000
1
0
Constraint53
2
32,1
76,-10000
1
0
Constraint54
2
33,1
77,-10000
1
0
Constraint55
2
34,1
78,-10000
1
0
Constraint56
2
35,1
79,-10000
1
0
Constraint57
2
36,1
80,-10000
1
0
Constraint58
2
```

```
37,1
81,-10000
1
0
Constraint59
2
38,1
82,-10000
1
0
Constraint60
2
39,1
83,-10000
1
0
Constraint61
2
40,1
84,-10000
1
0
Constraint62
2
41,1
85,-10000
1
0
Constraint63
2
42,1
86,-10000
1
0

42

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
```

Simplex evaluation-5-6-2 MILP

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86

3

Figure G19 The values of problem 5-6-2 MILP on notepad.

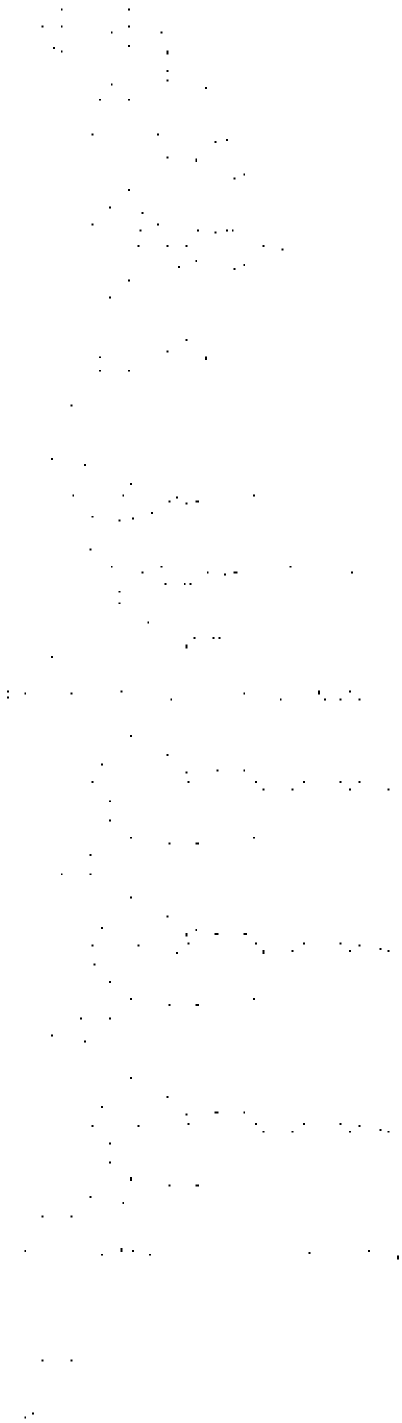


Figure G20 Simplex subroutine (MILP case).

```
All of integer variables are integer
The best answer of this problem is        8.00000023306347

X (   1) =            .0000000000
X (   2) =            .0000000000
X (   3) =            .0000000000
X (   4) =            .0000000000
X (   5) =            .0000000000
X (   6) =            .0000000000
X (   7) =            .0000000000
X (   8) =            .0000000000
X (   9) =            .0000000000
X (  10) =            .0000000000
X (  11) =            .0000000000
X (  12) =            .0000000000
X (  13) =            .0000000000
X (  14) =            .0000000000
X (  15) =            .0000000000
X (  16) =            .0000000000
X (  17) =         700.0000000000
X (  18) =         200.0000000000
X (  19) =            .0000000000
X (  20) =            .0000000000
X (  21) =            .0000000000
X (  22) =         300.0000000000
X (  23) =            .0000000000
X (  24) =            .0000000000
X (  25) =            .0000000000
X (  26) =         300.0000000000
X (  27) =            .0000000000
X (  28) =            .0000000000
X (  29) =            .0000000000
X (  30) =            .0000000000
X (  31) =            .0000000000
X (  32) =         300.0000000000
X (  33) =            .0000000000
X (  34) =            .0000000000
X (  35) =         700.0000000000
X (  36) =            .0000000000
X (  37) =            .0000000000
X (  38) =            .0000000000
X (  39) =            .0000000000
X (  40) =         300.0000000000
X (  41) =            .0000000000
X (  42) =         200.0000000000
X (  43) =     1500000.0000000000
X (  44) =      720000.0000000000
X (  45) =            .0000000000
X (  46) =            .0000000000
X (  47) =            .0000000000
X (  48) =            .0000000000
X (  49) =            .0000000000
X (  50) =            .0000000000
X (  51) =            .0000000000
X (  52) =            .0000000000
X (  53) =            .0000000000
X (  54) =            .0000000000
X (  55) =            .0000000000
X (  56) =            .0000000000
X (  57) =            .0000000000
X (  58) =            .0000000000
X (  59) =            .0000000000
X (  60) =            .0000000000
X (  61) =           1.0000000000
X (  62) =           1.0000000000
X (  63) =            .0000000000
```

```
X (    64) =            .0000000000
X (    65) =            .0000000000
X (    66) =           1.0000000000
X (    67) =            .0000000000
X (    68) =            .0000000000
X (    69) =            .0000000000
X (    70) =           1.0000000000
X (    71) =            .0000000000
X (    72) =            .0000000000
X (    73) =            .0000000000
X (    74) =            .0000000000
X (    75) =            .0000000000
X (    76) =           1.0000000000
X (    77) =            .0000000000
X (    78) =            .0000000000
X (    79) =           1.0000000000
X (    80) =            .0000000000
X (    81) =            .0000000000
X (    82) =            .0000000000
X (    83) =            .0000000000
X (    84) =           1.0000000000
X (    85) =            .0000000000
X (    86) =           1.0000000000
```

Figure G21 Results of supply chain 5-6-2 with binary from the program (Case 1).

Moreover, the alternative solutions from program show the source code in Fig. G22.

Figure G22 Subroutine source code for alternative solution (MILP case).

From this case, 2 other solutions can be obtained from the developed algorithm, as shown in Fig. G23 and G24. The profit of case 2 and case 3 is 780,000. For case 1 and 2, the last most negative variable is used to solve in the program. However, the minimum solution shows in 3 nodes (in branch-and-bound process), which show in case 1, 2 and 3, respectively.

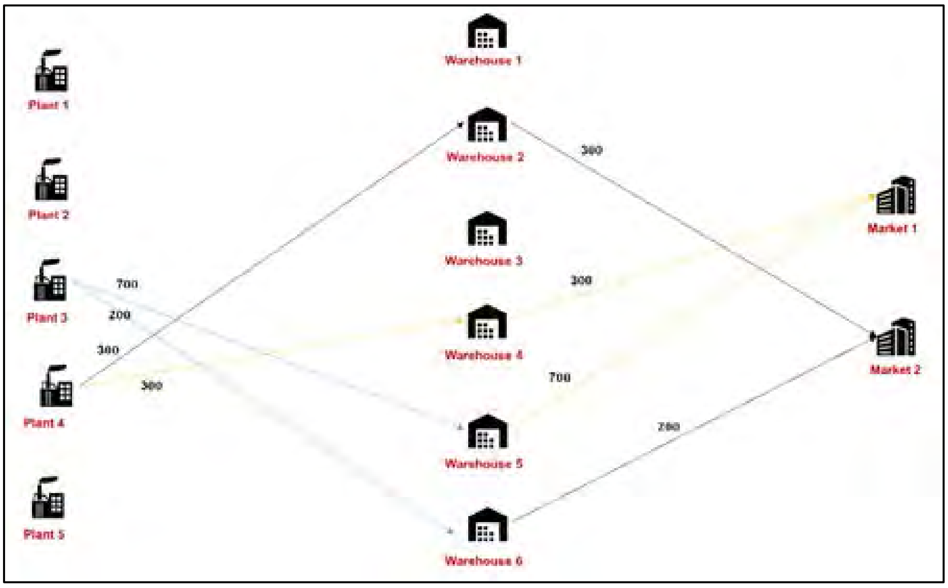


Figure G23 Other mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 2).

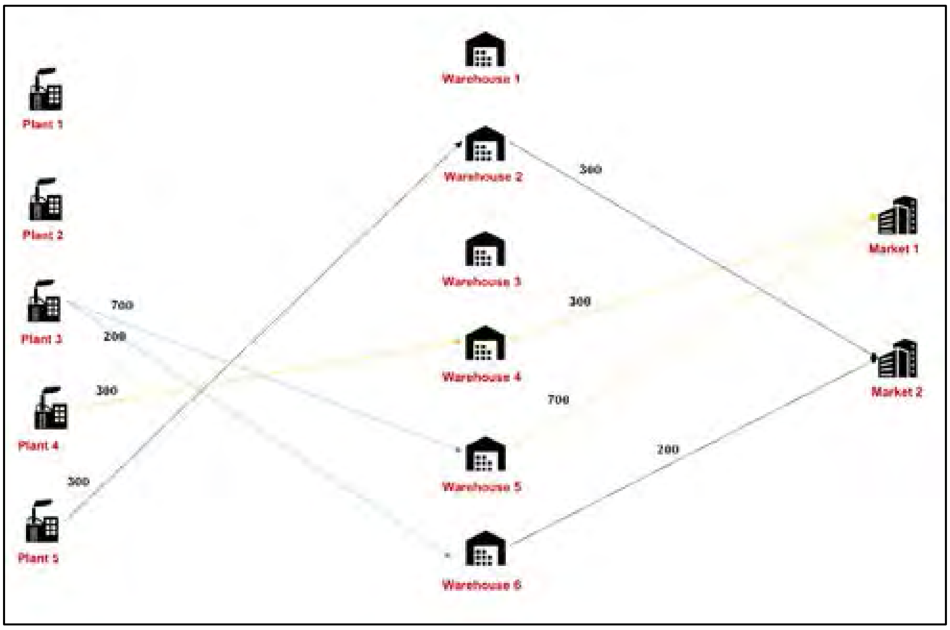


Figure G24 Other mass flow results of supply chain 5-6-2 with binary by MILP algorithm (Case 3).

CIRRICULUM VITAE

Name: Mr. Tittawat Fongchaintuk

Date of Birth: Feb 8, 1995

Nationality: Thai

University Education:

2016-2018 Master Degree of Petrochemical Technology, The Petroleum and Petrochemical College, Chulalongkorn University, Bangkok, Thailand

2012-2015 Bachelor Degree of Chemical Engineering, Faculty of Engineering, Mahanakorn University, Bangkok, Thailand

Work Experience:

2014	Position:	Student Internship
	Company name:	Mitrphol Kalasin

Proceedings:

1. Meeyoo, V., Fongchantuk, T. and Phongprueksathat, N. (2017, Jan 30) Acrylic acid production from lactic acid dehydration over vanadium-phosphorus oxide catalysts. Proceeding of WCCE10, Barcelona, Spain

2. Fongchantuk, T. and Siemanond, K. (2018, June 5) Optimization algorithm study: Mixed Integer Linear Programming. Proceeding of The 24th PPC Symposium on Petroleum, Petrochemicals, and Polymers and The 9th Research Symposium on Petrochemical and Materials Technology, Bangkok, Thailand