



**FINAL PROJECT – TI 141501**

**IMPROVING CRUDE OIL SUPPLY ASSIGNMENT PROCESS  
USING SIMULATED ANNEALING ALGORITHM:  
CASE STUDY IN EAST INDONESIA**

FIKRI IRSYAD MUHAMMAD

NRP 02411440000064

Supervisor

Prof. Ir. Budi Santosa, M.Sc, Ph.D.

NIP. 196905121994021001

DEPARTMENT OF INDUSTRIAL ENGINEERING

Faculty of Industrial Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2018



**APPROVAL SHEETS**

**IMPROVING CRUDE OIL SUPPLY ASSIGNMENT PROCESS  
USING SIMULATED ANNEALING ALGORITHM:  
CASE STUDY IN EAST INDONESIA**

**FINAL PROJECT**

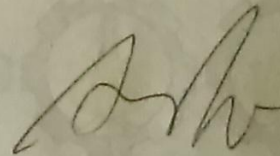
Submitted as a Requirement for Bachelor Degree in  
Department of Industrial Engineering  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya, Indonesia

Written By:

**FIKRI IRSYAD MUHAMMAD**  
**NRP 02411440000064**

Acknowledged and approved by:

Supervisor



**Prof. Ir. Budi Santosa, M.Sc. Ph.D.**  
**NIP 196905121994021001**

**SURABAYA, JULY 2018**





**IMPROVING CRUDE OIL SUPPLY ASSIGNMENT PROCESS  
USING SIMULATED ANNEALING ALGORITHM:  
CASE STUDY IN EAST INDONESIA**

Student Name : Fikri Irsyad Muhammad  
Student ID : 02411440000064  
Supervisor : Prof. Ir. Budi Santosa, M.Sc, Ph.D.

**ABSTRACT**

Oil industry is one of the most profitable industry in the world, with considerably high level of income and margin profit, which imply that any improvement no matter how small could have a bigger impact than expected. For this research a case study is reviewed which focuses on a supply chain department of an oil refinery in East Indonesia. One of the responsibility of said department is to create a crude oil tank assignment plan. The current condition of the job is quite wasteful as the job is still done manually, thus taking very long time to complete. This research is aimed to improve the method of creating crude oil tanks assignment by utilizing an algorithm that will considerably reduce the computation time needed, as well as improving the crude composition target achievement. Simulated annealing algorithm is the chosen method of this research because of its capability to reliably solve a problem with large dataset in reasonable amount of time. From 3 experiments with differing datasets, the results shows that the algorithm is capable of improving the target achievement of crude classification composition up to below 1 % deviation from the target as well as significantly reducing the time needed to do the assignment from several hours into under 1 minutes.

Keywords: Crude Oil Assignment, Crude Oil Classification, Metaheuristics,  
Simulated Annealing

*(This page is intentionally left blank)*

## ACKNOWLEDGEMENT

First of all, the author would like to express sincere gratitude and praise towards Allah SWT, by whose grace, guidance, and blessing has allowed the author to finish this research entitled “Improving Crude Oil Supply Assignment Process using Simulated Annealing Algorithm: Case Study in East Indonesia”

The author also would like to especially express gratitude and appreciation to those who had supported the author during the completion of this research, which includes but not limited to these people, namely:

1. Prof. Ir. Budi Santosa, M.Sc, Ph.D. as the supervisor, under whose guidance, direction, and supervision during the whole process of the research, has allowed the author to complete the research in time.
2. Prof. Dr. Ir. Budisantoso Wirjodirjo, M.Eng, Diesta Iva Maftuhah, S.T., M.T., as well as Effi Latiffianti, S.T., M.Sc., as the reviewers of the research proposal and the final report, whose advice and feedback had helped the author during the completion of this research.
3. Nurhadi Siswanto, S.T., M.S.I.E., Ph.D., as the Head of Industrial Engineering Department who provided the ease of administration during the completion of this research.
4. Author’s own parents, family members, friends and colleagues, as well as everyone else who had provided countless support in any form for the author during the completion of this research.

The author also fully aware of the fact that this report is still far from perfect. Therefore, any suggestions or critics for this report will be fully appreciated. The author also would like to apologize beforehand in case of any mistake or unpleasant word encountered within this report. Hopefully this research report can provide numerous benefit to the readers.

Surabaya, July 23<sup>rd</sup> 2018

Author

*(This page is intentionally left blank)*



## TABLE OF CONTENTS

ABSTRACT .....	i
ACKNOWLEDGEMENT .....	iii
TABLE OF CONTENTS .....	v
LIST OF TABLES .....	vii
LIST OF FIGURES .....	ix
CHAPTER I INTRODUCTION .....	1
1.1 Background .....	1
1.2 Problem Formulation.....	5
1.3 Research Objectives .....	5
1.4 Research Benefits .....	5
1.5 Research Scope and Limitations .....	6
1.5.1 Limitations .....	6
1.5.2 Assumptions.....	6
1.6 Report Structure .....	6
CHAPTER II LITERATURE REVIEW.....	9
2.1 Crude Oil .....	9
2.2 Crude Oil Assignment Process.....	10
2.3 Metaheuristics .....	18
2.4 Simulated Annealing .....	18
2.5 Previous Research .....	19
CHAPTER III RESEARCH METHODOLOGY .....	21
3.1 Mathematical Model and Algorithm Development Step.....	22
3.2 Experiment and Analysis Step .....	30
3.3 Conclusion Step.....	30

CHAPTER IV EXPERIMENT AND ANALYSIS .....	31
4.1 Algorithm Development .....	31
4.1.1 First Step: Initialization .....	31
4.1.2 Second Step: Initial Solution Generation .....	32
4.1.3 Third Step: Objective Function Calculation .....	34
4.1.4 Fourth Step: New Solution Generation .....	35
4.1.5 Fifth Step: Comparing New and Initial Solution.....	42
4.1.6 Sixth Step: Iteration and Temperature Update .....	43
4.1.7 Seventh Step: Stopping Criteria .....	43
4.1.8 Last Step: Algorithm Validation .....	43
4.2 Data Collection .....	45
4.3 Experiment and Result.....	48
4.4 Result Analysis .....	52
4.4.1 Existing Condition Analysis.....	52
4.4.2 Analysis of Experiment Result.....	52
CHAPTER V CONCLUSION .....	55
5.1 Conclusion .....	55
5.2 Suggestion.....	56
REFERENCES .....	57
APPENDIX A .....	59
APPENDIX B.....	61
APPENDIX C.....	67
APPENDIX D .....	91
BIOGRAPHY .....	167

## LIST OF TABLES

Table 2. 1 Crude Oil Fractions .....	9
Table 2. 2 Comparison with Previous Researches .....	20
Table 4. 1 Validation Experiment Result .....	45
Table 4. 2 Crude Oil Shipment Schedule for June 2017 .....	46
Table 4. 3 Crude Oil Shipment Schedule for July 2017 .....	46
Table 4. 4 Crude Class Composition Percentage Target for June and July 2017 .	47
Table 4. 5 Oil Tanks Initial Condition on June 1 <sup>st</sup> 2017 .....	47
Table 4. 6 Oil Tanks Initial Condition on July 1 <sup>st</sup> 2017.....	48
Table 4. 7 Final Composition Recapitulation of Experiment 1 .....	49
Table 4. 8 Target Achievement Deviation for Experiment 1 .....	50
Table 4. 9 Computation Time & BYU Target Achievement for Experiment 1....	50
Table 4. 10 Final Composition Recapitulation of Experiment 2 .....	50
Table 4. 11 Target Achievement Deviation for Experiment 2.....	50
Table 4. 12 Computation Time & BYU Target Achievement for Experiment 2..	51
Table 4. 13 Final Composition Recapitulation of Experiment 3 .....	51
Table 4. 14 Target Achievement Deviation for Experiment 3.....	51
Table 4. 15 Computation Time & BYU Target Achievement for Experiment 3..	51

*(This page is intentionally left blank)*

## LIST OF FIGURES

Figure 2. 1 Illustration of Crude Oil Movement in Refinery Plant.....	10
Figure 2. 2 Crude Classification and Tank Composition Example.....	12
Figure 2. 3 Manual Simulation Spreadsheet (Part 1).....	13
Figure 2. 4 Manual Simulation Spreadsheet (Part 2).....	14
Figure 2. 5 Example of Crude Intake Composition Indicator.....	16
Figure 2. 6 Existing Method Conceptual Model.....	17
Figure 3. 1 Research Methodology Flowchart.....	21
Figure 3. 2 Model Outline of Crude Oil Supply Assignment Process.....	22
Figure 3. 3 Simulated Annealing Algorithm Flowchart.....	29
Figure 3. 4 Illustrations of the Relationships within the Objective Function .....	30
Figure 4. 1 The Initial Solution Function in MATLAB.....	33
Figure 4. 2 Transfer Set Example for Initial Solution.....	33
Figure 4. 3 Medium Load Set Example for Initial Solution .....	34
Figure 4. 4 Result Report from LINGO Experiment .....	43
Figure 4. 5 LINGO Solver Status Example .....	44
Figure 4. 6 LINGO Experiment Solver Status.....	49
Figure 4. 7 Objective Plot for Experiment 1 .....	53
Figure 4. 8 Objective Plot for Experiment 2.....	54

*(This page is intentionally left blank)*

# **CHAPTER I**

## **INTRODUCTION**

This chapter serves as the introduction of the research report. The background of the research, its goals, benefits, scope and limitations, as well as the report structure will be outlined in this chapter.

### **1.1 Background**

Oil industry is one of the most profitable industry in the world, with considerably high level of income and margin profit. This also holds true in Indonesia, as oil and gas sector provide a considerable profit of around 20% of total profit obtained by state-owned companies (Kementerian BUMN, 2016). This fact implies that any improvement made on oil and gas sector would give considerable benefit based on their profit margin. Take the supply chain sector for example, the inbound supply on oil refinery consists of periodic incoming crude oil shipment, which is stored in several storage tanks, which is transferred daily to the refinery. Focusing on the storage tanks, the oil movement going in and out of the tanks itself requires planning to assign instruction such as which tank receive crude shipment, which tank transfer to refinery, the amount to transfer, etc. While the current development shows that refineries are able to determine which types of product to be produced in monthly plans based on factors such as price forecast, expected crude level and composition; this only result in the overall crude composition target for crude oil to be transferred to the refinery. Meanwhile, the planning for the crude oil transfer are left to be done manually, through trial and error simulation. Based on observation done by Shah et al. (2011), this happened due to the lack of an existing computationally efficient technique for automatic scheduling, and the non-existence of commercially available software tools for crude oil short-term scheduling. This open up some questions regarding the accuracy and efficiency of the crude oil assignment process, with opportunity for improvement. Therefore, it is logically sound to conduct research about the crude oil assignment process.

In this research, the author conducted direct observation in an oil refinery located in Indonesia. In this refinery, things regarding crude oil supply are the responsibility of the supply chain and distribution section which reside under the refinery planning and optimization division. In there the previously mentioned case of tank assignment process for crude oil supply (or shortened as Crude Oil Assignment Process) is typically worked by a single staff. This process consist of manually assigning transfer value to various storage tanks on daily basis, monitoring the daily stock of crude oil in each tank, as well as its composition, all while simultaneously looking out for when a crude shipment is arriving so that the received crude oil shipment can be pumped directly into one of the empty storage tanks. This is done on monthly scale, few weeks before the month starts. Whenever any change in the incoming shipment schedule occurs, revision for the assignment plan would need to be done as well.

This tank assignment process main purpose is to provide an assignment guideline for crude oil transfer and load activity. This guideline is not meant as a strict plan that must be followed, but rather as the overall plan to show the general directions for the crude oil movement in the following month. This way the worker in the field can easily do any adjustment that might be needed due to the uncertainty of the crude oil scheduling. Any changes can occur unexpectedly, such as ships arriving late or sometime even earlier than predicted, forcing adjustment to be made for the crude loading plan.

Aside from the previously mentioned main purpose, the assignment process also provide an additional benefit, which is to help guide and plan the crude oil transfer and the crude shipment load in a way such that the overall class composition of the crude oil that will be processed in the following month will be as close as possible with the ideal class composition that will yield the best possible combination of refinery product based on the monthly calculation done by the refinery management. This specific calculation is provided in a document called “Short Term Survey” which is the output of the routine monthly meeting done by the refinery management with the goal of determining the most optimal plan of utilizing resources available to the refinery in the following month, based on various data such as crude stock, production capacity, and market price forecast.



The classifications of crude oil directly affect what kind of final product that will be produced after the refining process. Crude oil generally classified based on its density. Heavier crude tends to contain more Light Gas Oil (LGO) or commonly known as Diesel fuel, while lighter crude tends to contain more Naphta which is the raw material of gasoline, making the lighter type of crude getting more sought in the world market as its price rises. Further explanations on crude oil classifications will be provided in later chapter. Since the management of crude oil class composition could have significant impact further down the production process, this shows that the crude oil tanks assignment process is important enough to be researched further.

The basic concept of crude oil tank assignment process is actually quite simple, to determine in each day which oil tanks are available for crude oil transfer into the refinery, as well as which oil tanks are available to unload crude oil from incoming shipment. The main issue of solving this problem is that there are numerous constraints that is related to the process. This is further complicated with the fact that the oil tanks in this refinery do not separate tanks by crude type and classification as commonly practiced, instead any incoming crude oil shipment could be mixed into any tanks with enough space. This made the additional purpose of the tanks assignment process, which is to control the crude oil classification of the transferred oil significantly harder to achieve, since the worker need to considers the crude level and class composition of each tanks in each days.

Currently the tank assignment process are done through manual simulation, by using a spreadsheet as the tools. This means it takes a considerable amount of time to finish, and also prone to error, leading to multiple revision later on which consumes even more time. The time that it takes to create a single solution set could range into couple hours, taking into account of possible revision later on, the total required time could add up to days. Some of the staff when asked for their opinion agree that the current work method is a bit inconvenient, but admit that currently there are not any better solution. The representative of the refinery itself expressed a desire to improve their work method for this process, in which a method that is capable to automatically design a schedule is more preferable.

One of the possible method to solve this kind of problem is by utilizing metaheuristics approach. Metaheuristics is defined as a method to find solution which combines interaction between local search procedures and other higher strategy to create process which is capable of escaping local optima and searching in solution space to find the global solution (Santosa, 2011). One of the main benefit of metaheuristics approach is that it is capable of generating a solution which is close to the global optima, with considerably minimum time. Therefore, this approach is suited to solve a large problem which requires the result to be generated in shortest time possible, such as the crude oil assignment process.

According to the constructed mathematical model for the crude oil assignment process (which will be provided in detail in further chapter), the solution set for this problem consists of a large number of variables, which is to be expected as one of the content of the assignment result is the value of crude oil transferred in each tanks, for each days, in which each one of these are represented as a variable. Based on this fact, the method of metaheuristics that is considered to be the appropriate method to solve the problem is the Simulated Annealing algorithm, because while it is just as capable of solving similar type of optimization problem as other popular population-based method like Particle Swarm Optimization (PSO), simulated annealing works by evaluating generated solution once at a time, hence more suitable to work with a large solution set.

An Industrial Engineer always seek to improve upon a system, either through a large-scale change or even from a small improvement, such as work simplification. One of the principle of work simplification is to apply and evaluate a new and better work method if an improvement is deemed necessary (Wignjosoebroto, 2003). Other form of work improvement is waste elimination. Doing unnecessary or suboptimal activities in repeat can be considered as wasting time (Puvanasvaran, 2014). This is indeed the case with the problem that occurred in the observed refinery, which further reinforce the need of improvement.

Therefore, the author is proposing to conduct a research to improve the crude oil assignment process in a refinery located in Indonesia, by utilizing simulated annealing algorithm.

## **1.2 Problem Formulation**

Based on the provided background, the current method used to plan the crude oil assignment in the observed refinery is considered wasteful since it is done manually by a single worker which takes considerable amount of time to finish and prone to error. In this regard, improvement in the achievement of crude oil class composition target as well as reduction of the time required to complete the assignment task is desired. Therefore, to improve the crude assignment process both result-wise and time-wise, the proposed solution is to develop an algorithm that is capable to provide better crude oil tank assignment with a significant time reduction. The time freed up then can be allocated for the worker to check and revise the assignment result, or to do other work that may benefit from the additional work time provided.

## **1.3 Research Objectives**

The objectives to fulfill in this research are as follows:

1. To develop an algorithm that is capable of generating more optimal tank assignment plan (minimizing the cumulative crude composition difference from target).
2. To considerably reduce the time needed to create or revise the tank assignment plan.

## **1.4 Research Benefits**

The main benefit of this research is that the time needed to complete the tank assignment will be considerably reduced by utilizing the proposed algorithm, hence reducing the workload of the worker. The nature of algorithm-based calculation will also further reduce the human error factor of the task. This research will also open future possibility of integrating more algorithm-based calculation into other area of crude oil supply chain management. Other benefits includes the possible improvement on crude composition target achievement and the possibility of saving cost by reducing the number of worker in the future.

## **1.5 Research Scope and Limitations**

Here stated the scope and limitations of this research:

### *1.5.1 Limitations*

1. The crude oil storage tanks is considered as the focus of the research. Oil movement unrelated to the storage tanks will be disregarded.
2. The data to be analyzed are primary data obtained from an oil refinery in East Indonesia.
3. The model does not allow flexible daily load allocation, the amount to be loaded from crude oil shipment will be determined beforehand.

### *1.5.2 Assumptions*

1. Crude Oil classifications (L/M/H) are assumed to stay the same during the research.
2. Shipment schedule data are assumed to be fixed with no revision for every month.
3. Aside from specific crude oil type that have special constraint, all type of crude oil in the same classifications are regarded as the same.

## **1.6 Report Structure**

Below provided the structure of this research report.

### CHAPTER I – INTRODUCTION

This chapter consist of the general topic about the research, such as the background of the research, its goals, and its benefits. Background includes a brief description of the problems, the existing condition, plus the motivation behind conducting the research. Problem formulation and research objectives will state the research problem and its objectives. Research benefits refers to benefits that will be obtained by conducting the research activity. Scope and limitations will state the area that this research will cover, as well as any limitations needed. Rounding up this chapter is a brief exposition on the report structure.

## CHAPTER II – LITERATURE REVIEW

Chapter II will cover references and theory related to the problem, including previous researches. References related to the method used to solve the problem will be provided in here as well.

## CHAPTER III – RESEARCH METHODOLOGY

This chapter will provide explanation of each step taken during the course of the research. The mathematical model of the problem is provided here.

## CHAPTER IV – EXPERIMENT AND ANALYSIS

Chapter V will cover the algorithm development, research experiment, as well as the analysis of the result. Comparison analysis will be conducted between experimented methods and the real existing result.

## CHAPTER V – CONCLUSION AND RECOMMENDATION

The final chapter will provide conclusion to the research, answering the objectives stated in the beginning. Recommendation resulted will also be provided in this chapter.

*(This page is intentionally left blank)*

## CHAPTER II

### LITERATURE REVIEW


This chapter will provide references and theory related to the research problems, including the result of previous researches.

#### 2.1 Crude Oil

Crude oil, also known as Petroleum, is derived from remains of animals and plants from million years ago which has been pressured and decomposed underground. It is a mixture of materials, which mainly comprised of hydrocarbon plus various other compounds (Speight, 2014). Crude Oil is used as the main raw material in oil refinery industry. The end product of oil refinery ranges from various type of fuel such as Gasoline, Diesel, and Kerosene (which can be further refined into jet fuel, known as Avtur). Other product including specialized fuel with higher octane number, liquid gases, and usable residues such as asphalt and paraffin.

Crude oil typically consist of many fraction of different types of hydrocarbon chain, which normally would be separated through distillation as the first process in refinery. Different fraction means different type of processing which means different end product as well. The composition of these fraction may vary between different sources of crude oil.

**Table 2. 1 Crude Oil Fractions**

	Boiling Point °C		
<b>Light</b>	< 30	C <sub>4</sub> <sup>-</sup>	Gas & LPG
	30 - 80	C <sub>5</sub> - C <sub>6</sub>	Light Naphta
	80 - 200	C <sub>7</sub> - C <sub>10</sub>	Heavy Naphta (Gasoline)
	200 - 280	C <sub>10</sub> - C <sub>22</sub>	Kerosene
	280 - 330	C <sub>22</sub> <sup>+</sup>	Diesel (LGO)
	330 - 370		Atmospheric Gas Oil
	370 - 540		Vacuum Gas Oil
	<b>Heavy</b>	540 +	

Source: Refinery archives

Comparison between crude oil are done through its density, as different types of hydrocarbon fraction has different density. In the observed refinery, based on its density (weight), the crude oil are classified into three classes, which are Light Crude (L), Medium Crude (M), and Heavy Crude (H). Heavier crude typically contains more LGO (Diesel Fuel), while lighter crude usually has more Naptha which can be refined into gasoline.

Crude oil normally also contain impurities in certain amount. Some are negligible, while some are very detrimental, such as Sulfur (S) which generally found more often in heavier crude oil (Hyne, 2012). This lead to some undesirable effect such as forcing refinery to limit the usage of crude oil from said source to reduce the burden on the chemical equipment in the refinery.

## 2.2 Crude Oil Assignment Process

One of the task of the crude oil supply chain division in the observed refinery is to plan and manage the assignment of incoming crude oil shipment into the available oil storage tanks, as well as its subsequent daily transfer into the refinery for further processing. A simplified diagram of oil movement for inbound supply in a refinery plant is shown in Figure 2.1 as follows:

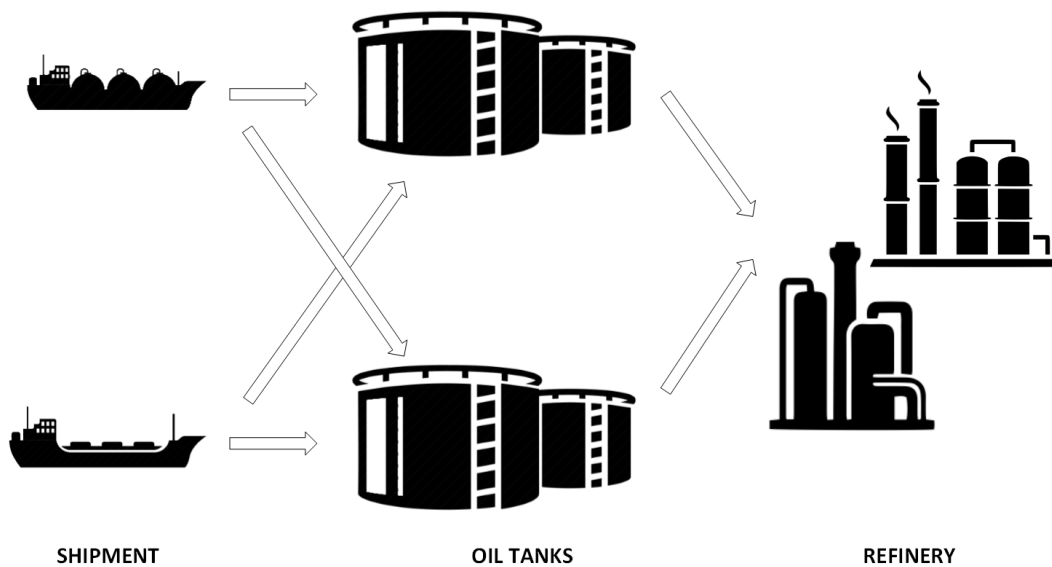


Figure 2. 1 Illustration of Crude Oil Movement in Refinery Plant



Using the oil storage tanks as the focus point of the research, there are two kinds of crude oil movement: incoming crude oil from shipments, and outgoing crude oil toward the refinery. Planning and managing these oil movements are the responsibility of the supply chain division of the observed refinery, specifically the crude supply chain section.

Incoming crude refer to oil received through shipments, both domestic-based crude and imported crude. In the current case study, the incoming crude shipment are received through a system called Single Point Mooring (SPM). The SPM is a system for receiving crude from shipment through a loading buoy anchored offshore in a certain distance from the land to ensure that the offload location is deep enough to receive larger ship that otherwise unable to dock in the shallower sea around the local jetty (Wichers, 2013).

The crude oil shipments received through SPM travel along the underwater pipeline until it reached the mainland in which the pipeline continues to the storage tanks. Once a ship pumped out all their oil cargo, there will be some oil remaining along the pipeline which will stay there without entering the storage. These oil remainder will then be pushed out during the next oil shipment into the storage tanks before the actual shipment enters (Miesner and Leffler, 2006). These small amount of oil will have the same characteristic as the last type of oil received from the last shipment. These oil remainder are known as “Ex Line Fill”, or ELF in short. These ELF oil will be calculated first for each time a crude oil shipment arrives.

Outgoing crude refers to the crude oil transferred into the refinery for further processing. The amount of crude oil to be transferred are determined based on the production target which is decided on each month. The process to determine the target is influenced by various factors including the market price forecast, current oil stock, and incoming shipment. This information is provided in a monthly report called Short Term Survey (STS). The monthly production target will then adjusted to a daily transfer amount which will be uniform throughout the month. The monthly production target itself, aside from the targeted amount of crude oil to be processed, also include the crude composition percentage (Light / Medium / Heavy crude) and certain constraint for specific crude type if needed.

In managing the crude oil movement of the storage tanks, both incoming crude and outgoing crude are closely related. The daily transfer of crude oil will empty the storage tank bit by bit, leaving space to put in the new shipment of crude oil. The crude classification percentage of each storage tanks depends on the composition of crude oil that is stored inside. Example of crude classifications and crude composition in a storage tanks is provided in the following Figure 2.2:

HAN	Handil Mix	L
MAD	Madura	L
SEN	Senipah Condensate	L
BEL	Belida	L
SER	Seria	L
BEK	Bekapai	M
SEP	Sepinggan	M
BON	Bonny Light	M
ESC	Escravos	M
BAL	Belanak	M
JAT	Jatibarang	M
LAN	Langsa	M
QUA	Qua Iboe	M
ARJ	Arjuna	M
AZE	Azeri	H
BYU	Banyu Urip	H

A			
1	AZE	H	206.71
2	ESC	M	101.81
3	HAN	L	6.67
4	SEP	M	49.18
5			
6			
7			
8			
9			
10			
11			
Total			364.37
L	1.8 %		6.67
M	41.4 %		151.00
H	56.7 %		206.71

B			
1	BYU	H	423.08
2	ESC	M	111.21
3	HAN	L	64.33
4	SEP	M	28.70
5	BON	M	91.61
6	BEK	M	5.29
7			
8			
9			
10			
11			
Total			724.23
L	8.9 %		64.33
M	32.7 %		236.82
H	58.4 %		423.08

**Figure 2. 2 Crude Classification and Tank Composition Example**

The method that is currently being used by the crude oil supply chain staff of the observed refinery to plan the crude oil assignment is through manual simulation with spreadsheet as media. Illustration of this method is shown in Figure 2.3 and 2.4 on the following page. As seen in the figures, these display is what would be shown to a staff of the supply chain section of the observed refinery when they first started there. This is a part of the spreadsheet that they use for conducting the manual simulation of the assignment process. Several important information can be found there. The topmost row refer to the date. The few row below shows incoming shipment and the crude they carry. The six box below them refer to each storage tanks, each tank has the topmost row as their stock indicator, row number one for transfer value (colored orange), and the remaining row for shipment loading.

Date	1-Jun-17	2-Jun-17	3-Jun-17	4-Jun-17	5-Jun-17	6-Jun-17	7-Jun-17	8-Jun-17	9-Jun-17	10-Jun-17	
Tanker			GAMKONORA					SC CHAMPION		NORDIC	
Crude 2			HAN 400					BYU 400		BON 950	
Crude 3			SEN 200					HAN 100			
Crude 4											
A	= 364	782	782	782	782	782	782	782	782	782	
	1										
	2	AZE 418									
	3										
	4										
B	= 724	724	699	674	644	614	554	494	429	364	
	1		-25	-25	-30	-30	-60	-60	-65	-65	
	2										
	3										
	4										
C	= 228	263	467	759	759	759	629	499	374	249	
	1	-105				-130	-130	-125	-125	-125	
	2	AZE 140	AZE 204	ELF 92							
	3			HAN 200							
	4										
D	= 775	775	610	445	285	125	125	125	617	625	
	1		-165	-165	-160	-160					
	2							ELF 92	HAN 8		
	3							BYU 400			
	4										
E	= 781	781	781	781	781	781	781	781	781	781	
	1										
	2										
	3										
	4										
F	= 206	121	121	121	321	429	429	429	429	429	
	1	-85									
	2			HAN 200	SEN 108					ELF 92	
	3									BON 100	
	4										
TRANSFER	-190	-190	-190	-190	-190	-190	-190	-190	-190	-190	
CUMULATIVE INT.											
INTAKE (L)	17%	37.29 20%	39.51 10%	41.73 7%	44.40 6%	47.06 5%	94.13 8%	141.19 11%	187.10 12%	233.00 14%	278.91 15%
INTAKE (M)	48%	66.05 35%	212.21 56%	358.37 63%	501.98 66%	645.60 68%	676.11 59%	706.63 53%	738.37 49%	770.10 45%	801.84 42%
INTAKE (H)	35%	86.65 46%	128.28 34%	169.90 30%	213.62 28%	257.34 27%	369.76 32%	482.17 36%	594.53 39%	706.89 41%	819.25 43%

Figure 2. 3 Manual Simulation Spreadsheet (Part 1)

Date	21-Jun-17	22-Jun-17	23-Jun-17	24-Jun-17	25-Jun-17	26-Jun-17	27-Jun-17	28-Jun-17	29-Jun-17	30-Jun-17
Tanker		SUCCESS CHALLENGER				OLYMPIC FLAG			CHEVRON	
Crude 2		HAN 100				BON 950			SEP 50	
Crude 3		BYU 400								
Crude 4										
A	=	782	782	782	782	782	782	782	782	782
	1									
	2									
	3									
	4									
B	=	774	774	644	514	384	254	124	524	782
	1		-130	-130	-130	-130	-130			
	2							BON 400	BON 258	
	3									
	4									
C	=	162	470	470	470	470	470	762	762	762
	1									
	2	BYU 308					ELF 92 BON 200			
	3									
	4									
D	=	165	125	317	625	625	625	625	625	575
	1	-40								-50
	2		ELF 92 HAN 100	BYU 308						
	3									
	4									
E	=	768	768	768	768	768	768	768	678	588
	1							-90	-90	-140
	2									
	3									
	4									
F	=	771	621	561	501	441	381	321	221	121
	1	-150	-60	-60	-60	-60	-60	-100	-100	
	2									ELF 50
	3									
	4									
TRANSFER		-190	-190	-190	-190	-190	-190	-190	-190	-190
CUMULATIVE INT.										
INTAKE (L)	17%	509.89 13%	542.89 13%	575.88 13%	608.88 13%	641.88 14%	674.88 14%	739.06 14%	803.25 15%	831.94 15%
INTAKE (M)	48%	1572.55 39%	1709.27 41%	1845.99 42%	1982.71 43%	2119.42 45%	2256.14 46%	2355.68 46%	2455.22 46%	2557.91 46%
INTAKE (H)	35%	1907.56 48%	1927.84 46%	1948.13 45%	1968.41 43%	1988.69 42%	2008.98 41%	2035.26 40%	2061.54 39%	2120.16 38%

Figure 2. 4 Manual Simulation Spreadsheet (Part 2)

On the bottom of the main section is the indicator for total transfer amount of that day. The section below is the cumulative intake of the crude oil transferred into the refinery, which is accumulated until the end of the month. On the initial column is the crude composition target which can be used to compare with the end result at the end of the month. The column from left to right show progression of each day during the manual simulation. Figure 2.5 illustrate the crude intake composition indicator, which can be used to crosscheck the daily intake crude classification with the corresponding constraints.

To do the manual simulation one need to understand all the rules and constraints of this process. The conceptual model of the existing method can be seen in Figure 2.6. It is basically a set of primarily standardized procedure which is done in repeat for each day of the month. For each day, the first step to be done is to identify which tanks that will be used to transfer the crude to the refinery, as well as determining the amount to be transferred from respective tanks.

One of the most commonly practiced rule of thumb is to use 2 tanks for transfer, let 2 tanks available for loading, and left 2 tanks standby which makes it a balanced division (2 : 2 : 2). Another rule that is in place is to always use a tank for transfer until it is empty, unless situation decides otherwise. This rule is created to simplify the pipeline control for the worker in the field, since opening and closing multiple valves on daily basis is deemed to be too much work. Another additional rule is that a tank cannot be used for transfer within a day after the tank has finished receiving crude shipment to ensure that the water that possibly mixed up within the shipment can separate itself from the stored crude oil. These water will be drained out of the storage tanks when it reached certain amount.

The constraints that affect this particular process are the tanks current capacity, its capacity lower limit, the daily transfer target value, and certain crude type transfer limit. In the currently observed case, represented by Figure 2.3 - 2.4 for the time period of June 2017, the tanks initial capacity are provided for the first day, the capacity lower limit is 120 MB, the daily transfer target value is 190 MB, and the special limit for BYU crude is 25% of total intake.

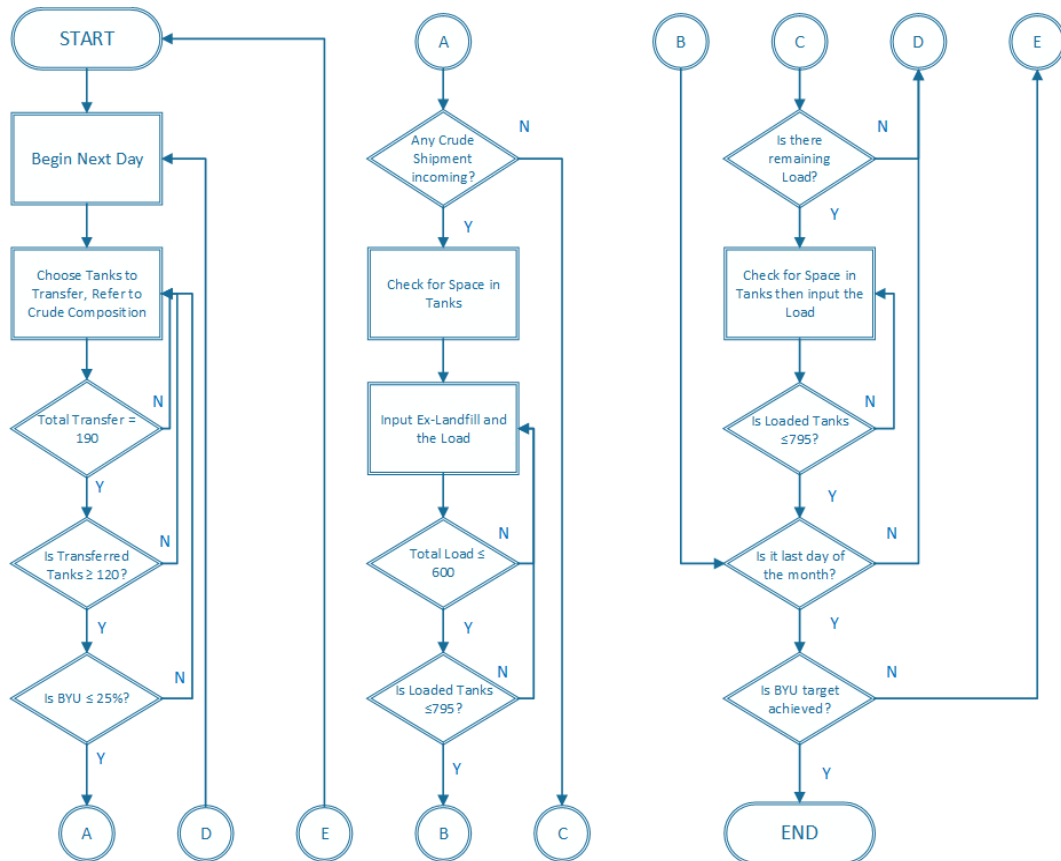
To illustrate these rules in practice, please take a look at the example provided in the previous Figure 2.3, especially the section for 6<sup>th</sup> of June. There it shows from the previous day, that the tank D has been used to transfer until it almost reached the lower capacity limit, while tank B that is being used simultaneously still have spare crude stock to transfer. This means that the B tank will be used along with another tank that is available to use. The decision of which tank to use and the amount of crude to be transferred is left to the staff judgment, as long as it is satisfying all the related constraints. In this observed case, the staff decided to transfer 60 MB of crude from tank B and 130 MB of crude from tank C.

Date			1-Jun-17	2-Jun-17	3-Jun-17	4-Jun-17	5-Jun-17					
Tanker					GAMKONORA							
Crude 2					HAN	400						
Crude 3					SEN	200						
Crude 4												
INTAKE (L)	17%		37.29	20%	2.22	1%	2.22	1%	2.66	1%	2.66	1%
INTAKE (M)	48%		66.05	35%	146.16	77%	146.16	77%	143.61	76%	143.61	76%
INTAKE (H)	35%		86.65	46%	41.62	22%	41.62	22%	43.72	23%	43.72	23%
INTAKE COMP												
1	AZE	H	42.18	22%	10.47	6%	10.47	6%	10.16	5%	10.16	5%
2	ESC	M	1.35	1%	8.00	4%	8.00	4%	8.64	5%	8.64	5%
3	HAN	L	-	-	2.22	1%	2.22	1%	2.66	1%	2.66	1%
4	SEP	M	3.35	2%	0.99	1%	0.99	1%	1.19	1%	1.19	1%
5	BYU	H	44.48	23%	31.15	16%	31.15	16%	33.57	18%	33.57	18%
6	BON	M	38.03	20%	124.77	66%	124.77	66%	121.72	64%	121.72	64%
7	BEK	M	5.75	3%	4.73	2%	4.73	2%	4.63	2%	4.63	2%
8	ARJ	M	2.36	1%	-	-	-	-	-	-	-	-
9	SEN	L	11.60	6%	-	-	-	-	-	-	-	-
10	JATL	L	4.03	2%	-	-	-	-	-	-	-	-
11	KIK	M	5.37	3%	7.67	4%	7.67	4%	7.43	4%	7.43	4%
12	MAD	L	21.66	11%	-	-	-	-	-	-	-	-
13	QUA	M	9.85	5%	-	-	-	-	-	-	-	-
14	SEF	L	-	-	-	-	-	-	-	-	-	-
15	JAT	M	-	-	-	-	-	-	-	-	-	-
16	BEL	L	-	-	-	-	-	-	-	-	-	-
17	LAN	M	-	-	-	-	-	-	-	-	-	-
18												
19												
20												

Figure 2. 5 Example of Crude Intake Composition Indicator

The next process in the manual simulation is to allocate the crude shipment into tanks with space when the scheduled shipment arrive. The related constraints are the tanks upper capacity limit, daily load limit, and certain procedure for a specific shipment. These process are repeated until final day of the month. In the final day, the cumulative class composition of the transferred crude oil is then compared to the initial target. If it is close enough then the result is accepted.

In the case if there is some error found, or when the final crude oil class composition result judged to be not good enough, a revision to the tanks assignment will need to be made. The judging criteria is mostly subjective, with result within 1% difference generally accepted to be good enough.



**Figure 2. 6 Existing Method Conceptual Model**

The major weakness of this method is that it is inaccurate by nature, as it is done by trial and error which subject to the user capability and human error, as well as the fact that most of the time it took a lot of time to get a desirable result. The reason that this method is still in use is that it is simple to prepare.

### **2.3 Metaheuristics**

*Metaheuristics* term is derived from the term *meta* and *heuristics*, which literally translate to “more than heuristics”. Heuristics itself is known as structured method of trial & errors or randomly generating solutions, therefore metaheuristics is implied to be a more proven and systematical method of solving problems through randomly generated solutions.

Talbi in his book (2009) defined metaheuristics as upper level general methodologies (templates) that can be used as guiding strategies in designing underlying heuristics to solve specific optimization problems. Meanwhile, Budi Santosa in his book (2013) defined metaheuristics as a method to find solution which combines interaction between local search procedures and other higher strategy to create process which is capable of escaping local optima and searching in solution space to find the global solution. From those definitions, several key aspect of metaheuristics is defined as follows: Based on heuristics, able to find solution close to global optima, and minimal computation time.

### **2.4 Simulated Annealing**

Simulated Annealing is a method of metaheuristics based on the annealing process used in metallurgic. In simulated annealing optimization problem, the structure of the substance represents a codified solution of the problem, and the temperature is used to determine how and when new solutions are generated and accepted. The algorithm is basically a three steps process: generate solution, evaluate the solution, and accept or reject the solution (Ledesma et al, 2008)

It is known as a search method that is able to avoid local optima, thus giving higher chance to find the most optimal solution. It is capable of generating near-optimal solution for most problems. Based on research previously conducted, it is the most widely used metaheuristics approach. Its popularity is due to its simplicity and ease of implementation (Nikolaev and Jacobson, 2010)

In this research, simulated annealing algorithm is deemed to be suitable for solving the crude oil assignment problem. The reason being that this algorithm is capable of working with large solution set with less load on computer compared to other population-based method such as ACO and PSO.



## 2.5 Previous Research

There exist several researches in the past few years which is related to the crude oil assignment problem, as well as researches that utilize the Simulated Annealing algorithm. The various literature that has been studied are summarized as follows:

Hou et al. (2015) attempted to solve crude oil scheduling problem using genetic algorithm. In their research, they successfully modeled the whole oil movement from shipment until refinery process in a nonlinear model, which is solved using a modified Genetic Algorithm. Still by the same author (Hou and Wu, 2015), they also attempted to model a short-term scheduling problem of crude oil operations, and optimize it using a heuristic approach.

Rocha et al. (2009), modelled the crude oil scheduling problem as a Mixed-Integer Linear Programming, which is solved using a variance of a Local Branching Procedure. Saharidis and Ierapetritou (2009), tried to model the scheduling of loading and unloading of crude oil in a refinery, and solve it using certain problem-specific heuristic method. These various researches mostly modelled the problem with broader scope, which is until the refinery process itself. Nevertheless, these findings prove that metaheuristics approach mostly effective to solve this kind of problem.

Some researches has also shown that Simulated Annealing algorithm is capable of solving similar optimization problem. Ribeiro et al. (2010) provided a simple and robust Simulated Annealing algorithm for scheduling workover rigs on onshore oil fields in Brazil. They did a comparison study with other heuristics method and found that simulated annealing algorithm generally produce better results. Saraiva et al. (2011) tried to solve a generator maintenance scheduling problem using a simulated annealing based approach. They found that simulated annealing is easily implemented and provides good results in numerous optimization problems.

Based on the various example of previous researches, it can be seen that the current research is still related to the examples. Comparison with previous researches are provided in Table 2.2 in the following page:

**Table 2. 2 Comparison with Previous Researches**

Research	Problem Type	Objective	Method
Hou et al. (2015)	Crude oil scheduling	Minimize cost	Genetic Algorithm
Hou and Wu (2015)	Short term scheduling	Minimize mixing and switching	Heuristic Approach
Rocha et al. (2009)	Petroleum allocation problem	Minimize total cost	Local Branching Procedure
Saharidis and Ierapetritou (2009)	Scheduling crude oil loading process	Minimize setup cost	Heuristic Approach
Ribeiro et al. (2010)	Scheduling workover rigs	Minimize production loss	Simulated Annealing
Saraiva et al. (2011)	Maintenance scheduling problem	Minimize operation cost	Simulated Annealing
This Research	Crude oil tanks assignment	Minimize target deviation	Simulated Annealing

Compared to previous researches, this research shows similarity in the general theme which is petroleum industry, as well as research method of metaheuristics. Meanwhile, some differences are noted in form of problem scope and research objectives. Most research are focusing in cost minimization while this research focuses on minimizing target deviation of crude composition plan.

# CHAPTER III

## RESEARCH METHODOLOGY

This chapter will describe each step of the method to conduct the research, as shown in the flowchart below:

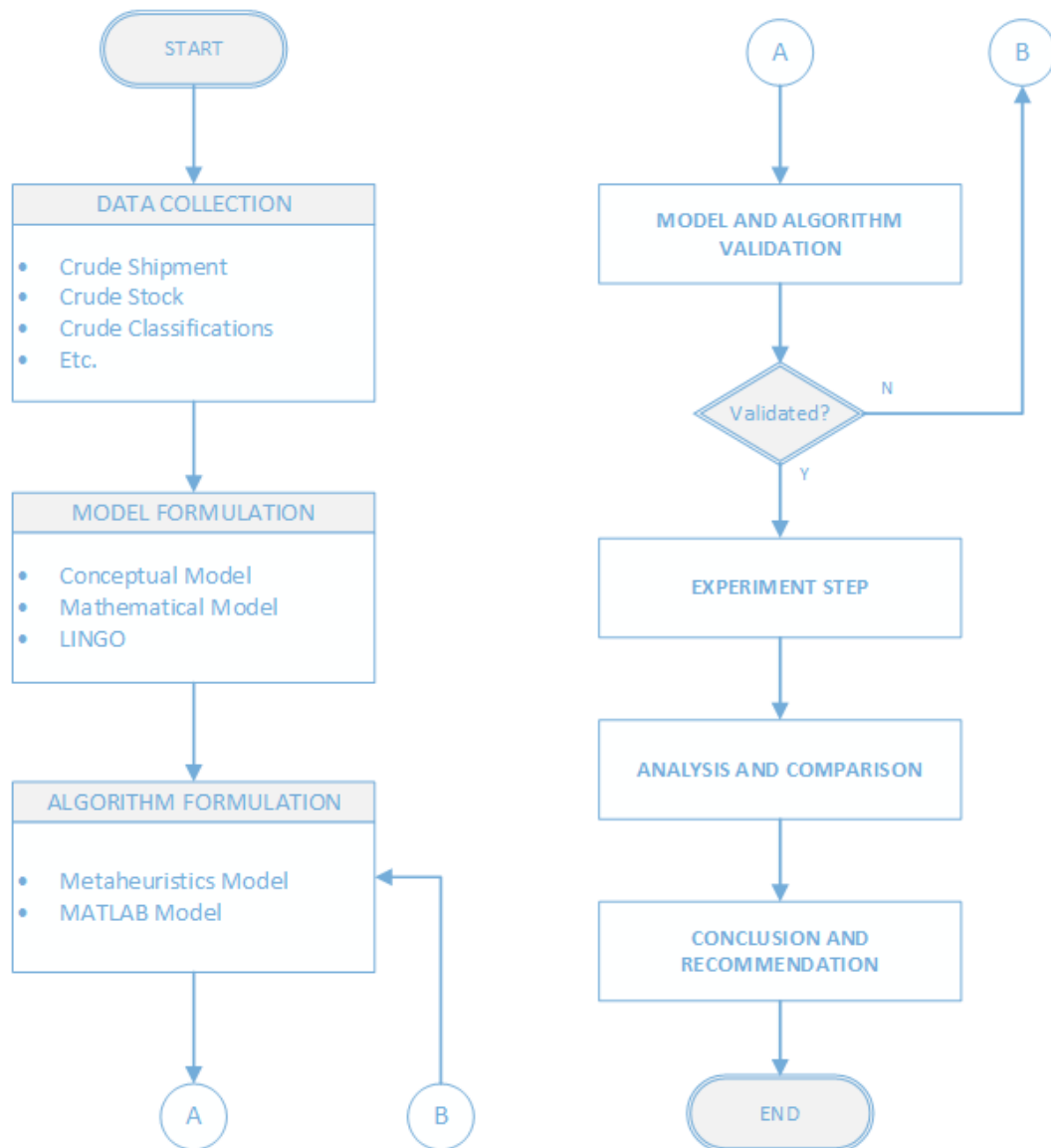


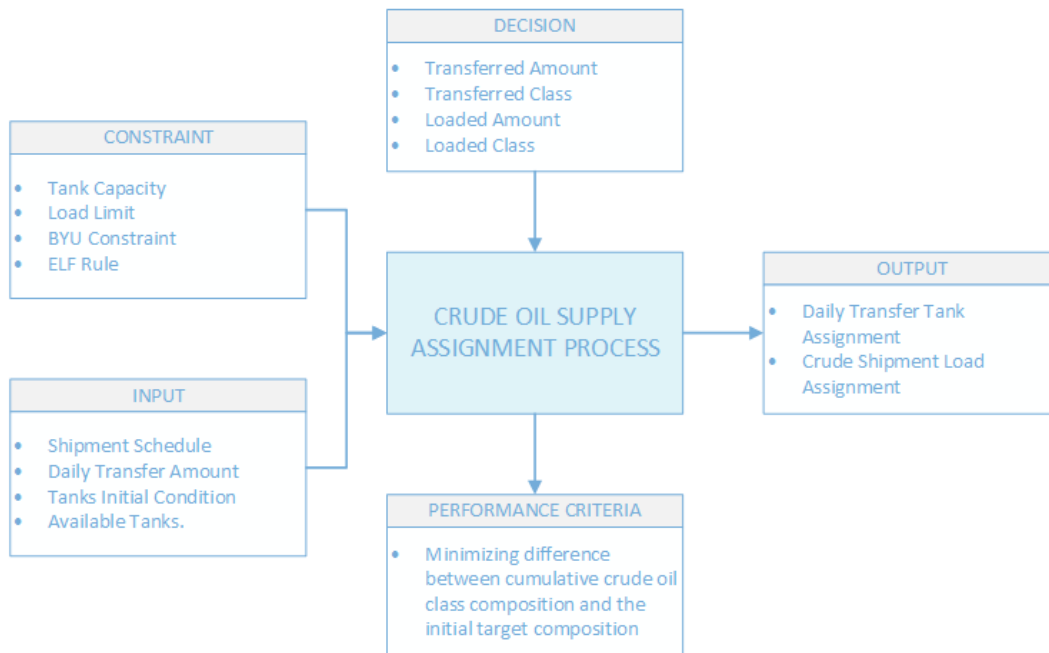
Figure 3. 1 Research Methodology Flowchart

### 3.1 Mathematical Model and Algorithm Development Step

To solve the researched problem, the first step to take is the mathematical model and algorithm development step. The process of developing the mathematical model will be provided in this chapter, while the algorithm development process will be in the next chapter. Based on the existing condition of the researched problem that has been observed (as previously provided in chapter II), the mathematical model for the researched problem can be constructed. The outline for the model is as shown below in Figure 3.2.

The mathematical model is then constructed in Lingo software to help verify whether the model is solvable or not. After constructing the mathematical model, a metaheuristics algorithm need to be constructed to help solve the mathematical model. The metaheuristics method that will be used is the simulated annealing algorithm. The algorithm modelling will be done using MATLAB software.

Before proceeding to the next step, both model need to be verified and validated first. The validation method to be used is through comparison of result with normal calculation. After both model is validated, the experiment can be conducted using determined method.



**Figure 3. 2 Model Outline of Crude Oil Supply Assignment Process**

From the provided outline, it can be seen what kinds of input that are there as well as the kinds of constraint that exist. To achieve the objective, there is a need to determine the optimal value of decision variable such as the transferred crude amount and class as well as the loaded amount and class of the crude shipment, for each tank in each day. From the generated solution the output will give the daily transfer tank assignment as well as the crude shipment load assignment for the month. This result will be used as guidance for the crude oil transfer and load activity in the following month.

The mathematical model of the problem are provided as follows:

Constants (Input Variable):

$H_0$  : Total monthly heavy crude percentage target

$M_0$  : Medium crude percentage target

$L_0$  : Light crude percentage target

$T$  : Total storage tanks available

$n$  : Amount of days in selected month

$IT$  : Daily transfer/intake amount

$HA_{(1j)}$  : Initial amount of heavy crude stock in tank (j)

$MA_{(1j)}$  : Initial amount of medium crude stock in tank (j)

$LA_{(1j)}$  : Initial amount of light crude stock in tank (j)

$HbA_{(1j)}$  : Initial amount of BYU crude stock in tank (j)

$SH_{(i)}$  : Amount of heavy crude oil shipment on day (i)

$SM_{(i)}$  : Amount of medium crude oil shipment on day (i)

$SL_{(i)}$  : Amount of light crude oil shipment on day (i)

$SHb_{(i)}$  : Amount of BYU crude oil shipment on day (i)

Support Variable:

$H$	: Cumulative monthly total heavy crude percentage
$M$	: Cumulative monthly total medium crude percentage
$L$	: Cumulative monthly total light crude percentage
$HA$	: Total amount of transferred heavy crude oil
$MA$	: Total amount of transferred medium crude oil
$LA$	: Total amount of transferred light crude oil
$HbA$	: Total amount of transferred BYU crude oil
$H(i)$	: Total transferred heavy crude percentage on day (i)
$M(i)$	: Total transferred medium crude percentage on day (i)
$L(i)$	: Total transferred light crude percentage on day (i)
$Hb(i)$	: Total transferred BYU crude percentage on day (i)
$T(ij)$	: Current stock of Storage tank (j) on day (i)
$HA(ij)$	: Total amount of heavy crude stock on day (i) in tank (j)
$MA(ij)$	: Total amount of med. crude stock on day (i) in tank (j)
$LA(ij)$	: Total amount of light crude stock on day (i) in tank (j)
$HbA(ij)$	: Total amount of BYU crude stock on day (i) in tank (j)
$HP(ij)$	: Percentage of heavy crude stock on day (i) in tank (j)
$MP(ij)$	: Percentage of medium crude stock on day (i) in tank (j)
$LP(ij)$	: Percentage of light crude stock on day (i) in tank (j)
$HbP(ij)$	: Percentage of BYU crude stock on day (i) in tank (j)

Solution Set (Decision Variable):

- $IT_{(ij)}$  : Transfer/intake amount on day (i) from tank (j)  
 $TS_{(ij)}$  : Transfer status of tank (j) on day (i), 1 if used, else 0  
 $LS_{(ij)}$  : Load status of tank (j) on day (i), 1 if used, else 0  
 $XH_{(ij)}$  : Daily load value of heavy crude oil on day (i) to tank (j)  
 $XM_{(ij)}$  : Daily load value of med. crude oil on day (i) to tank (j)  
 $XL_{(ij)}$  : Daily load value of light crude oil on day (i) to tank (j)  
 $XHb_{(ij)}$  : Daily load value of BYU crude oil on day (i) to tank (j)

Indices Explanation:

- (i) refers to specific days, from 1 to n  
(j) refers to specific tanks, from 1 to T

Objective function:

$$\text{Min } Z = |H_0 - H| + |M_0 - M| + |L_0 - L| \quad (3.1)$$

Subject to:

Cumulative monthly total crude percentage is the total of each class of crude divided by total transferred crude:

$$H = \frac{HA + HbA}{HA + MA + LA + HbA} \quad (3.2)$$

$$M = \frac{MA}{HA + MA + LA + HbA} \quad (3.3)$$

$$L = \frac{LA}{HA + MA + LA + HbA} \quad (3.4)$$

Total percentage of each class transferred crude on day (i) is equal to the sum of the transferred amount of crude oil in each tank multiplied with the percentage of the crude class in that tank divided by the sum of the transferred amount of crude oil in each tank:

$$H_{(i)} = \frac{\sum_{j=1}^T (IT_{(ij)} \times HP_{(ij)})}{\sum_{j=1}^T IT_{(ij)}} ; \forall i \quad (3.5)$$

$$M_{(i)} = \frac{\sum_{j=1}^T (IT_{(ij)} \times MP_{(ij)})}{\sum_{j=1}^T IT_{(ij)}} ; \forall i \quad (3.6)$$

$$L_{(i)} = \frac{\sum_{j=1}^T (IT_{(ij)} \times LP_{(ij)})}{\sum_{j=1}^T IT_{(ij)}} ; \forall i \quad (3.7)$$

$$Hb_{(i)} = \frac{\sum_{j=1}^T (IT_{(ij)} \times HbP_{(ij)})}{\sum_{j=1}^T IT_{(ij)}} ; \forall i \quad (3.8)$$

Percentage of each class of crude stock on day (i) in tank (j) is equal to the total amount of that class of crude stock on day (i) in tank (j) divided by the total crude oil stock of storage tank (j) on day (i):

$$HP_{(ij)} = \frac{HA_{(ij)}}{T_{(ij)}} ; \forall i, \forall j \quad (3.9)$$

$$MP_{(ij)} = \frac{MA_{(ij)}}{T_{(ij)}} ; \forall i, \forall j \quad (3.10)$$

$$LP_{(ij)} = \frac{LA_{(ij)}}{T_{(ij)}} ; \forall i, \forall j \quad (3.11)$$

$$HbP_{(ij)} = \frac{HbA_{(ij)}}{T_{(ij)}} ; \forall i, \forall j \quad (3.12)$$

Total amount of each type of crude stock on day (i) in tank (j) is equal to the amount of crude stock of the previous day, subtracted with the amount transferred the previous day, added with the amount of crude loaded into the tank in the previous day, as well as the amount of the ex-linefill added into the tank in the previous day  $\{1 < i \leq n\}$ :

$$HA_{(ij)} = HA_{(i-1,j)} - (HP_{(i-1,j)} \times IT_{(i-1,j)}) + XH_{(i-1,j)} ; \forall i, \forall j \quad (3.13)$$

$$MA_{(ij)} = MA_{(i-1,j)} - (MP_{(i-1,j)} \times IT_{(i-1,j)}) + XM_{(i-1,j)} ; \forall i, \forall j \quad (3.14)$$

$$LA_{(ij)} = LA_{(i-1,j)} - (LP_{(i-1,j)} \times IT_{(i-1,j)}) + XL_{(i-1,j)} ; \forall i, \forall j \quad (3.15)$$

$$HbA_{(ij)} = HbA_{(i-1,j)} - (HbP_{(i-1,j)} \times IT_{(i-1,j)}) + XHb_{(i-1,j)} ; \forall j \quad (3.16)$$



Current stock of Storage tank (j) on day (i) is equal to the sum of each class of crude oil (L, M, and H) in the tank on said day:

$$T_{(ij)} = HA_{(ij)} + MA_{(ij)} + LA_{(ij)} + HbA_{(ij)}; \forall i, \forall j \quad (3.17)$$

Total amount of transferred crude oil for each class is equal to the sum of each day value of the transferred amount of crude oil in each tank multiplied with the percentage of the crude class in that tank:

$$HA = \sum_{i=1}^n \sum_{j=1}^T (IT_{(ij)} \times HP_{ij}) \quad (3.18)$$

$$MA = \sum_{i=1}^n \sum_{j=1}^T (IT_{(ij)} \times MP_{ij}) \quad (3.19)$$

$$LA = \sum_{i=1}^n \sum_{j=1}^T (IT_{(ij)} \times LP_{ij}) \quad (3.20)$$

$$HbA = \sum_{i=1}^n \sum_{j=1}^T (IT_{(ij)} \times HbP_{ij}) \quad (3.21)$$

Total daily amount of transferred crude from each tank must be equal to the predetermined daily transfer amount:

$$\sum_{j=1}^T IT_{(ij)} = IT; \forall i \quad (3.22)$$

Total amount of crude oil in each tank is between 120 and 795 MB:

$$T_{(ij)} \geq 120; \forall i, \forall j \quad (3.23)$$

$$T_{(ij)} \leq 795; \forall i, \forall j \quad (3.24)$$

BYU transfer percentage max 25% per day, total intake min 1000 MB:

$$Hb_{(i)} \leq 0.25; \forall i \quad (3.25)$$

$$HbA \geq 1000 \quad (3.26)$$

Only 2 tanks can be used to transfer crude oil simultaneously in a day:

$$\sum_{j=1}^T TS_{(ij)} \leq 2; \forall i \quad (3.27)$$

$$TS_{(ij)} \times IT_{(ij)} = IT_{(ij)}; \forall i, \forall j \quad (3.28)$$

Tanks during loading cannot be used for transfer, must wait 1 day after finished loading before a tank may be used for transfer:

$$(XH_{(i-1,j)} + XM_{(i-1,j)} + XL_{(i-1,j)} + XH_{(i-1,j)}) \times IT_{(ij)} = 0; \forall i, \forall j \quad (3.29)$$

Must finish transfer all crude in tank before changing transfer tank:

$$TS_{(ij)} \times TS_{(i-1,j)} \times (T_{(ij)} - 130 - XH_{(i-1,j)} - XM_{(i-1,j)} - XL_{(i-1,j)} - XH_{(i-1,j)}) \\ \geq TS_{(i-1,j)} \times (T_{(ij)} - 130 - XH_{(i-1,j)} - XM_{(i-1,j)} - XL_{(i-1,j)} - XH_{(i-1,j)}); \forall i, \forall j \quad (3.30)$$

Daily load follows each day shipment schedule:

$$\sum_{j=1}^T XH_{(ij)} = SH_{(i)}; \forall i \quad (3.31)$$

$$\sum_{j=1}^T XM_{(ij)} = SM_{(i)}; \forall i \quad (3.32)$$

$$\sum_{j=1}^T XL_{(ij)} = SL_{(i)}; \forall i \quad (3.33)$$

$$\sum_{j=1}^T XHb_{(ij)} = SHb_{(i)}; \forall i \quad (3.34)$$

Only 2 tanks can be used for loading crude oil simultaneously in a day:

$$\sum_{j=1}^T LS_{(ij)} \leq 2; \forall i \quad (3.35)$$

$$LS_{(ij)} \times XH_{(ij)} = XH_{(ij)}; \forall i, \forall j \quad (3.36)$$

$$LS_{(ij)} \times XM_{(ij)} = XM_{(ij)}; \forall i, \forall j \quad (3.37)$$

$$LS_{(ij)} \times XL_{(ij)} = XL_{(ij)}; \forall i, \forall j \quad (3.38)$$

$$LS_{(ij)} \times XHb_{(ij)} = XHb_{(ij)}; \forall i, \forall j \quad (3.39)$$

Constraints for binary, integer, and non-negativity:

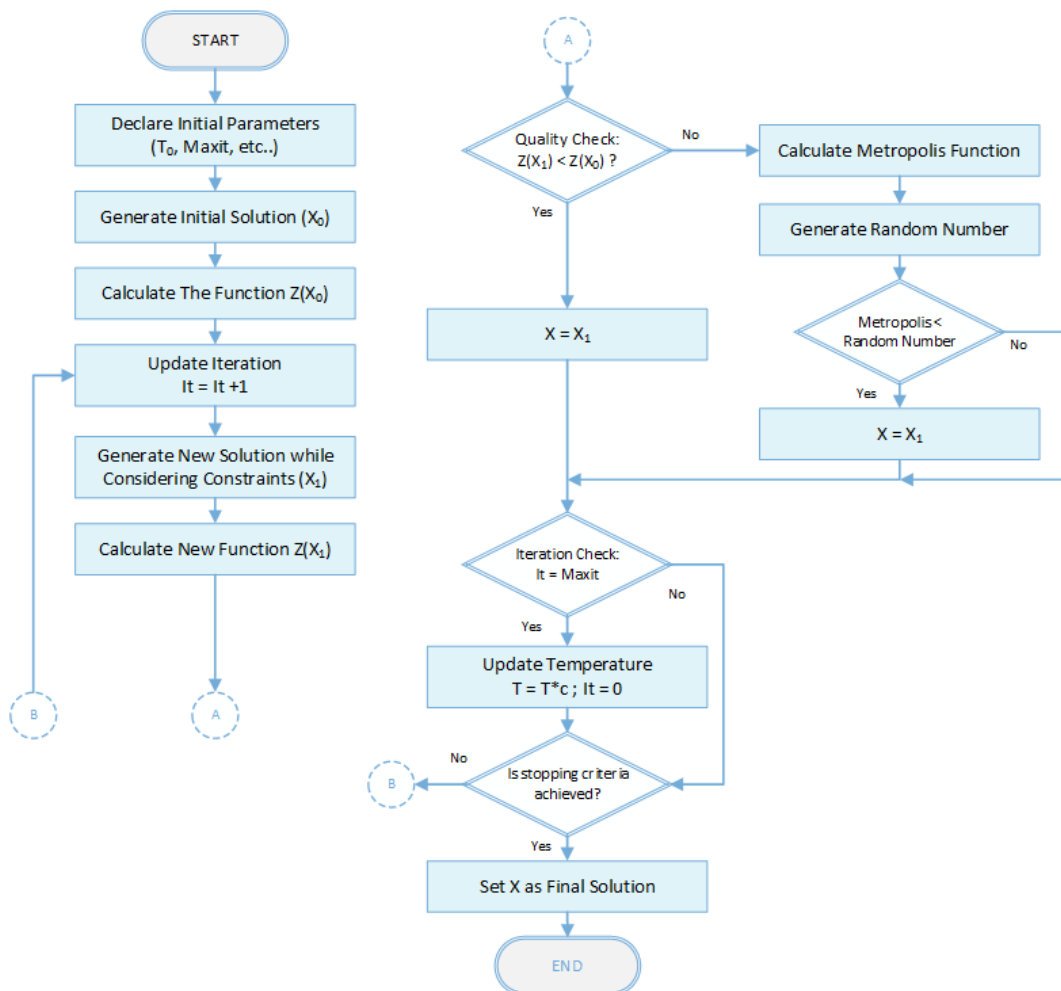
$$IT_{(ij)} = \{IT_{(ij)} \in \mathbb{Z} | IT_{(ij)} \bmod 5 = 0\}; \forall i, \forall j \quad (3.40)$$

$$TS_{(ij)}, LS_{(ij)} = \{0,1\}; \forall i, \forall j \quad (3.41)$$

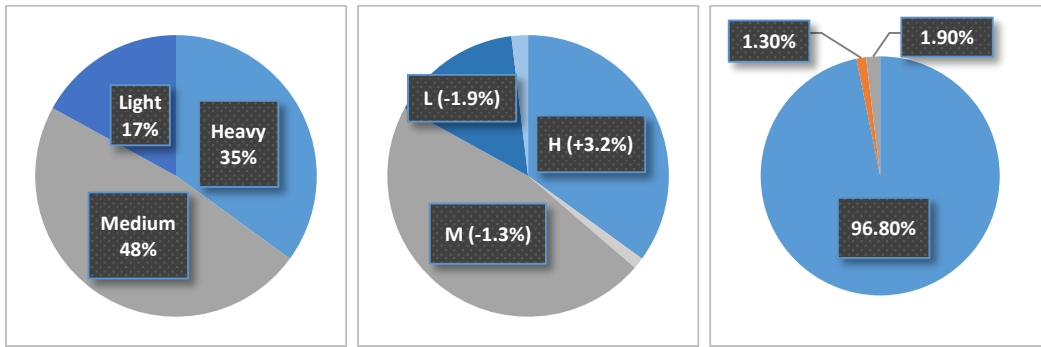
$$XH_{(ij)}, XM_{(ij)}, XL_{(ij)}, XHb_{(ij)} \in \mathbb{Z}; \forall i, \forall j \quad (3.42)$$

$$\text{All variables} \geq 0; \forall i, \forall j \quad (3.43)$$

After developing the mathematical model of the problem, the next step is to develop the simulated annealing algorithm. The workflow of the algorithm is illustrated as flowchart in the following Figure 3.3. Further development of the simulated annealing algorithm is covered in chapter IV.



**Figure 3. 3 Simulated Annealing Algorithm Flowchart**



**Figure 3. 4 Illustrations of the Relationships within the Objective Function**

About the objective function of this mathematical model, the cumulative percentage of each class of crude oil refer to the overall amount of transferred crude oil in the month which satisfy the equation (3.44) and (3.45) below. These equation are not included within the model because it is implied already. Equation (3.45) works in any combination, which imply that the total difference from target is equal to half of the total of crude oil deviation from target as shown on Figure 3.4.

$$H + M + L = 1 \quad (3.44)$$

*in case*  $|H_0 - H| > |M_0 - M| > |L_0 - L|$

$$\textit{then } |H_0 - H| = |M_0 - M| + |L_0 - L| \quad (3.45)$$

### 3.2 Experiment and Analysis Step

This step consists of experiment sub-step and analysis sub-step. In experiment sub-step, an attempt to solve the model will be done using an algorithm. The experiment will run the model in several iterations using available data.

After the experiment is concluded, the result will be analyzed and compared to the real life data from the observed refinery, to show how much the process actually improved. The experiment and analysis step is covered in chapter IV.

### 3.3 Conclusion Step

For the final step, the research conclusion will be drawn from the analysis result, which will show whether the research objectives is fulfilled or not. Recommendations for the observed refinery, the reader, or future researchers will be provided as well. The conclusion step is covered in chapter V.

## **CHAPTER IV**

### **EXPERIMENT AND ANALYSIS**

This chapter will provide the collected data, algorithm development, the experiment, as well as its analysis.

#### **4.1 Algorithm Development**

The algorithm development will be done using MATLAB software. The mathematical model previously created will be the core of this algorithm. Here are the steps of the algorithm development:

##### *4.1.1 First Step: Initialization*

First step is to initialize the necessary parameter of the simulated annealing algorithm. Simulated annealing algorithm is based on the annealing process in the furnace. It works by testing new solution set for each iteration as the temperature cooling down. While the algorithm naturally accept any better result found as the best solution, it also gives opportunity for worse result to be accepted. This probability gets lower as the temperature keep decreasing, which means in the beginning when the temperature is high, the algorithm will have a large option of path to take while it converges later on as the temperature decrease. Based on the explanation given, the crucial parameters includes the initial temperature ( $T$ ), the temperature reduction coefficient, and the maximum number of iteration.

To determine the parameters, for the initial temperature the value should be neither too high nor too low. The reasoning given is that very low temperature in relation to the function changes will return a very small value of Boltzman probability which in turn makes it harder for the function to escape local optima, while higher temperature will result in more time needed to converge.

Other input parameter of the algorithm is related to the mathematical model. These parameters are the shipment schedule, oil tanks initial composition, and the cumulative composition target. The value of these additional parameter will be provided in the data collection section of this chapter.

#### *4.1.2 Second Step: Initial Solution Generation*

In simulated annealing algorithm, the solution set is generated randomly. The initial solution set is comprised of various variable, including the transfer amount of crude oil from each tank on each day, as well as the load amount and classification of crude oil shipment for each tank on each day. In total there are 7 set of variables with specific boundary which conforms to numerous constraints. All of the variable in the solution set is an integer, with some of them specifically set as binary variable.

The variable in the solution set are interconnected to each other, which makes randomly generating solution set that satisfy all constraints very difficult. Therefore, to generate an initial solution, a simple heuristics method is applied. The method to be used is a modified form of Northwest Corner Method, a classic heuristics method which focuses in generating an initial solution set in least time possible while disregarding the objective function (Taha, 2007). This method is more favored for this research due to the high amount of constraint as well as the need to complete the calculation in least time possible.

The variables of solution set are dependent to the number of tanks available and the number of days in the month. The value of variables in any given day are directly related to the value of variables in the previous day. The value of the variables in the solution set represents the activity of crude oil shipment loading and crude oil transfer into refinery. To generate the solution, instead of the normal northwest corner method which blindly select the first variable slot as the starting point, the initial solution will be generated by following a simple logic, the stock of oil in each tank are ranked daily from 1 to 6, the two tanks with highest capacity will be used for transfer until it hits the capacity lower limit, while the 2 tanks with lowest capacity will be used for loading if there is shipment scheduled. The daily stock of tanks will be re-ranked whenever the transfer tanks gets empty or the loading tank gets filled. To simplify the calculation, the amount to be transferred or loaded will be divided evenly between the two chosen tanks. Since this generated solution is nowhere near final, some constraints are omitted to make it easier to generate an initial solution.

Here provided an example of the generated initial solution in MATLAB:

```
>> [ Z0, TransferSet0, HLoadSet0, MLoadSet0, LLoadSet0, HbLoadSet0 ] =
InitSolution( Target, Schedule, Stock, Tank, Day, TransferLimit )

Z0 =

0.1708
```

**Figure 4. 1 The Initial Solution Function in MATLAB**

TransferSet0 =					
0	0	0	95	95	0
0	95	0	0	95	0
0	95	0	95	0	0
95	0	95	0	0	0
0	95	0	95	0	0
0	0	95	0	95	0
0	0	0	95	0	95
95	95	0	0	0	0
0	0	0	95	0	95
0	0	0	95	0	95
95	0	0	95	0	0
0	0	95	0	95	0
0	0	95	0	95	0
95	0	95	0	0	0
0	0	0	95	95	0
0	0	0	95	95	0
0	95	0	0	0	95
95	0	0	95	0	0
0	0	95	0	95	0
0	95	0	0	0	95
0	95	0	0	0	95
95	0	0	95	0	0
95	0	0	95	0	0
0	0	95	0	95	0
0	95	0	0	0	95
0	0	95	0	95	0
95	0	0	95	0	0
0	0	95	0	95	0

**Figure 4. 2 Transfer Set Example for Initial Solution**

MLoadSet0 =					
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
50	50	0	0	0	0
0	0	275	0	275	0
104	0	0	104	0	0
0	0	0	0	0	0
0	0	0	0	0	0
121	0	121	0	0	0
0	0	0	300	300	0
54	0	54	0	0	0
0	75	0	0	0	75
46	0	0	46	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	100	0	0	0	100
0	0	200	0	200	0
129	0	0	129	0	0
0	25	0	0	0	25
0	0	0	0	0	0

**Figure 4. 3 Medium Load Set Example for Initial Solution**

Notice that the objective value for the initial set is still very high at 17%.

#### 4.1.3 Third Step: Objective Function Calculation

The value of objective function for each solution set will be calculated after the determination of the decision variable. The objective function is calculated based on the mathematical model, which if translated into MATLAB code will be as follows:

$$Z = \text{abs}(H0 - H) + \text{abs}(M0 - M) + \text{abs}(L0 - L)$$



#### 4.1.4 Fourth Step: New Solution Generation

In generating a new solution in simulated annealing algorithm, the most common way is by utilizing a method known as neighborhood search, in which a new solution will be generated by altering the initial solution into a value close to it. This process normally will be repeated until maximum number of the iteration is reached. For this research however, altering an initial set of solution proved to be difficult, since the variables are closely related to each other. Therefore, to generate a brand new solution is more favorable for this problem.

The selected method for generating the new solution will be different to the method of generating the initial solution. The tank for transfer and loading in each day will be determined based on several conditions. Based on the mathematical model, the selection of tank as well as the amount of transfer or load value is fully related to various constraints. The solution set consist of 5 array of variable, one for the transfer set and the rest is for the load set of each crude oil classifications. Note that the BYU type crude oil is represented by a separate variable from the other Heavy class crude due to the numerous amount of constraint specific to this type of crude oil, except for the total cumulative count of transferred heavy crude oil that include the BYU type crude oil into consideration.

The input for the new solution function are mainly the initial input for the simulated annealing algorithm, which includes the crude composition target, the shipment schedule of the month, initial crude stock at the beginning of the month, as well as parameters such as the number of tanks, number of days, and the daily transfer limit. The initial tank selection for the current best solution and the current temperature of the main algorithm also included for comparison. Below provided the input command of the new solution function:

```
[ tx1, tx2, TransferSet, HLoadSet, MLoadSet, LLoadSet, HbLoadSet,
TotalH, TotalM, TotalL, TotalHb, Z ] = NewSolution(
Target, Schedule, Stock, Tank, Day, TransferLimit, tb1, tb2, Tx)
```

It can be seen that the output of this function includes the new solution set variables, the objective function result, the total amount of transferred crude, as well as the initial tank choice for future comparison.

Technically, the whole process of generating the new solution set is a cycle of several process which goes like this: Determine two tanks for transfer -> Determine transfer value for both tanks -> Determine one or two tanks for loading -> Determine load value for selected tanks -> Recalculate support variables -> Back to beginning for next day variables. This cycle is repeated until the end of the month is reached, in which the constraint are checked and the objective function is calculated. This whole process might need to be repeated in case of error encountered or there are constraints that are not satisfied. The process itself are developed in a way so that the possibility of error is greatly reduced. In case that an invalid solution set is generated, the function will generate a new solution until a valid one appears.

Before the whole process which previously described as a cycle is started, the first step of the new solution generation is the selection of the initial transfer tanks. Due to the fact that the solution generation utilized random number, the generated solution set would generally have no relation to each other, which discourage the usage of simulated annealing algorithm. Therefore, to give some degree of connection to the previously generated solution set, the initial tank selection generated by this function will be compared with the initial tank selection from the current best solution in the main algorithm. The comparison method will use metropolis criteria just like the simulated annealing algorithm in general. The temperature will use the current temperature of the main algorithm. This way in the beginning the initial tank selection will hardly be affected, allowing wider selection choice of tank combination. However, as the main algorithm runs longer and the temperature keep decreasing, the initial tank selection will converge toward the current best solution. This idea works for this function since the transfer value amount is basically decided the moment the tank is selected without any random factor, while the main random number usage is for the selection of tanks. This shows that the initial tank selection is very influential to the end result.

The code for the initial tank selection basically the same as the normal code which is part of the cycle, except the fact that the code is specific for day 1, plus the additional comparison using metropolis criteria. In the initial part of the function there are also some initial calculation and various code line for variable initiation.

Here provided example of codes related to the initial part of the function (initial tank selection comparison using metropolis criteria):

```
comp0=sort([tb1 tb2]);
comp1=sort([T1(1) T2(1)]);
if abs(comp1(1)-comp0(1))+abs(comp1(2)-comp0(2))>0
    deltaf=abs(comp1(1)-comp0(1))+abs(comp1(2)-comp0(2));
    PE=exp(-deltaf/T); %boltzmann probability
    if rand>PE
        T1(1)=tb1;
        T2(1)=tb2;
    end
end
```

Aside from the initial tank selection, the other part of the function generally work the same way for each day calculation. Here provided the explanation of each process of new solution generation:

*Determine tanks for transfer:*

In the transfer process, to make it easier to know which tank is available for transfer, an array that represent the availability of each tank for transfer is utilized. This array is initialized daily by doing a net stock check of each tank. Tank that still have some net stock available for transfer will be assigned the value of one, while tank that is essentially empty will be assigned the value of zero. After this initial check, there are several other constraints that affect the transfer tank availability, such as the rule that a tank cannot be used for transfer one day after it is used for loading (refer to equation 3.29), and that a tank selected for transfer must be used for transfer continuously until it is empty before another tank can be selected (refer to equation 3.30). Related code example is provided as follows:

```
if TotalStock(i,j) >= 125
    AvT(i,j)=1;
end
if (HLoadSet(i-1,j)+MLoadSet(i-1,j)+LLoadSet(i-1,j)
+HbLoadSet(i-1,j)) > 0
    AvT(i,j)=0;
end
if TotalStock(i,j) - (HLoadSet(i-1,j)+MLoadSet(i-1,j)
+LLoadSet(i-1,j)+HbLoadSet(i-1,j)) >= 130
    AvT(i,j)=1;
    T1(i)=j;
end
```

As the tank selection for transfer process always comes in pair, each possible combination of tanks need to be checked with various condition to see whether a combination is allowed to be selected. The first check is to see whether both tank in the combination is available individually. Next is to check whether the currently selected transfer tank need to be replaced. After that, combination of tank that will yield result with BYU crude percentage above the limit will be set as not available to be selected. Finally, the last check that also serves as the transfer value determination, as it will consider both tanks net stock as well as the BYU percentage to simulate the transfer process until one of the tank is empty. This is done for all possible tank combination to ensure that the accepted transfer value will satisfy every related constraint. Related code example is provided as follows:

```

if dt(i,a) + dt(i,b) >= 38
    CT(a,b,i)=AvT(i,a)*AvT(i,b);
else
    CT(a,b,i)=0;
end
if T1(i)~=0 && T2(i)==0
    if a ~= T1(i) && b ~= T1(i)
        CT(a,b,i)=0;
    end
elseif T1(i)==0 && T2(i)~=0
    if a ~= T2(i) && b ~= T2(i)
        CT(a,b,i)=0;
    end
end
if BYUPercentage(i,a)>0.25 && BYUPercentage(i,b)>0.25
    CT(a,b,i)=0;
end

```

For the selection, the function will list all tank combination that is available for transfer, and randomly select one of them. The tank combination will be stored in the variable  $T1(i)$  and  $T2(i)$ . Related code example is provided as follows:

```

AT(i)=sum(sum(CT(:, :, i), 2));
CTs(i, :)=reshape(CT(:, :, i), 1, 36);
[~, idT(i, 1:36)]=sort(CTs(i, :), 'descend');
rt(i, 1:AT(i))=randperm(AT(i));
if mod(idT(i, rt(i, 1)), 6) == 0
    T1(i)=idT(i, rt(i, 1))/6;
    T2(i)=6;
else
    T1(i)=ceil(idT(i, rt(i, 1))/6);
    T2(i)=mod(idT(i, rt(i, 1)), 6);
end

```

*Determine transfer value for both tanks:*

The transfer value for each tank is already calculated during the previous final check, in which the transfer value of one of the tank is stored in a variable, so it just need to be recalled again. This value refer either to the tank with less BYU percentage or less net stock between the two, depending on the switch. Based on the selected tank combination, the checked variables also include the number of days left before the tank becomes empty, as well as the transfer value for the last day in case the value changes. Related code example is provided as follows:

```
if dt(i,T1(i)) <= dt(i,T2(i))
    TransferSet(i,T1(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
    TransferSet(i,T2(i))=
    TransferLimit*(38-tvl(T1(i),T2(i),i))/38;
else
    TransferSet(i,T2(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
    TransferSet(i,T1(i))=
    TransferLimit*(38-tvl(T1(i),T2(i),i))/38;
end
```

*Determine tanks for loading:*

Just like the transfer process, an array that represent the availability of each tank for loading is utilized. In each day, this array is initialized by doing space check on each tank. Tank that already full will be assigned the value of zero, while tank that has any amount of space is assigned the value of one. After that, the tank previously selected for transfer is checked. If the transfer tank is not empty yet, it cannot be used for loading, hence the availability status of said tank is overwritten into the value of zero. This is the final state of the load availability status for individual tank. Related code example is provided as follows:

```
if TotalStock(i,j) <= 794
    AvL(i,j)=1;
end
if (TotalStock(i,T1(i))-TransferSet(i,T1(i))) > 130
    AvL(i,T1(i))=0;
end
if (TotalStock(i,T2(i))-TransferSet(i,T2(i))) > 130
    AvL(i,T2(i))=0;
end
```

Before the tank for loading is selected, another set of capacity check would need to be done first. As opposed to the transfer process that will always use two tanks, the loading process would use only one tank whenever possible. In the case that single loading is not applicable, the loading can be separated into two tanks instead. This set of check is recorded as single load capability status. The check is done by comparing the available space in each tank compared to the total load in each day. Related code example is provided as follows:

```
for j=1:Tank
    if dl(i,j) >= totalSchedule(i)
        SL(i,j)=AvL(i,j);
    else
        SL(i,j)=0;
    end
end
```

The tank selection process can be started after all the checking is done. If there are tanks that is capable of single loading, the loading tank will be selected from those. Meanwhile, in case there are no tank capable of single loading, the tank will be selected based on combination which is able to contain all of that day load. The output will be stored in the variable  $L1(i)$  and  $L2(i)$ , with  $L2(i)$  set to zero when single loading tank is chosen.

*Determine load value for selected tanks:*

In the case that single loading tank is chosen, the load value for each class of crude oil is simply the scheduled load amount of that day, which will be put into the selected single loading tank. In the case that a tank combination is needed, the load value will be determined systematically from the first class of crude until the last one by considering both tanks capacity. Related code is provided as follows:

```
if L2(i) == 0
    HLoadSet(i,L1(i))=Schedule(1,i);
    (et cetera)
elseif L2(i) ~= 0
    if Schedule(1,i) ~=0
        RH(i,1:Schedule(1,i))=randperm(Schedule(1,i));
        HLoadSet(i,L1(i))=RH(i,1);
        HLoadSet(i,L2(i))=Schedule(1,i)-RH(i,1);
    end
end
```

*Recalculate support variables:*

This step basically a process to fill out support variables based on the generated solution set of the day, which will be used as consideration to determine the next day solution set. The processes typically do not have any kind of decision switch, only straightforward calculation. These variables include the likes of the amount of next day stock, the net stock available for transfer, the stock percentage of each crude classifications, as well as the BYU transfer percentage which is calculated for constraint purposes. Related code example is provided as follows:

```
HeavyPercentage(i,j)=HeavyStock(i,j)/TotalStock(i,j);
MediumPercentage(i,j)=MediumStock(i,j)/TotalStock(i,j);
LightPercentage(i,j)=LightStock(i,j)/TotalStock(i,j);
BYUPercentage(i,j)=BYUStock(i,j)/TotalStock(i,j);
dt(i,j)=floor((TotalStock(i,j)-120)/5);
HeavyStock(i+1,j)=HeavyStock(i,j)-
(HeavyPercentage(i,j)*TransferSet(i,j))+HLoadSet(i,j);
MediumStock(i+1,j)=MediumStock(i,j)-
(MediumPercentage(i,j)*TransferSet(i,j))+MLoadSet(i,j);
LightStock(i+1,j)=LightStock(i,j)-
(LightPercentage(i,j)*TransferSet(i,j))+LLoadSet(i,j);
TotalStock(i+1,j) = HeavyStock(i+1,j)+MediumStock(i+1,j)
+LightStock(i+1,j)+BYUStock(i+1,j);
BYUTrans(i,j)=BYUPercentage(i,j)*TransferSet(i,j);
BYUTransferPercentage(i)=
sum(BYUTrans(i,:))/sum(TransferSet(i,:));
```

After the whole process is completed the final calculation is conducted, which includes the objective function result as well as the constraint check of the generated solution set. Related code example is provided as follows:

```
TotalH=TotalH+TransferSet(i,j)*HeavyPercentage(i,j);
TotalM=TotalM+TransferSet(i,j)*MediumPercentage(i,j);
TotalL=TotalL+TransferSet(i,j)*LightPercentage(i,j);
TotalHb=TotalHb+TransferSet(i,j)*BYUPercentage(i,j);

H=(TotalH+TotalHb)/(TotalH+TotalM+TotalL+TotalHb);
M=(TotalM)/(TotalH+TotalM+TotalL+TotalHb);
L=(TotalL)/(TotalH+TotalM+TotalL+TotalHb);

Z= abs(Target(1)-H)+abs(Target(2)-M)+abs(Target(3)-L);

if BYUTransferPercentage(i) > 0.25
    Z=1;
end
```

The complete code for this function is provided in the appendix at the end of this report, alongside the main code for the algorithm.

#### 4.1.5 Fifth Step: Comparing New and Initial Solution

The objective function value of the newly generated solution will be compared with the objective function value of the initial solution set. The difference between this value will be represented as  $\Delta f$ . The calculation is as follows:

$$\Delta f = Z_{\text{new}} - Z$$

If the result is negative, the new solution is better and accepted as the new initial solution. However if the result is positive, it doesn't mean the new solution which is worse is discarded directly. The metropolis calculation will be done instead. A random generated number will be compared with the Boltzman probability, if the random number is lower than the Boltzman probability, then the new solution is accepted as the new initial solution, if it is higher, the new solution is discarded. Here provided an example of metropolis calculation:

$$\Delta f = 58$$

$$T = 3700$$

$$k=1$$

$$\text{rand}=0.534$$

$$P(E) = e^{-\Delta f/kT}$$

$$P(E) = e^{-58/3700}$$

$$P(E) = 0.984$$

Because the random generated number (0.534) is lower than the Boltzman probability value (0.984), the new solution is then accepted as the new initial solution. In case the random number is higher, the algorithm will simply keep the current best solution and continue to generate new solution set in the next iteration.

For this algorithm, since the mathematical model includes a global constraint that can be evaluated after the solution set is complete (refer to equation (3.26) in chapter III), the constraint will be evaluated here instead of the new equation function after taking consideration of computation time. Therefore, aside from having better objective value, the new solution set must satisfy this constraint as well before accepted automatically as the new best solution. Other cases will be evaluated using the metropolis criteria as usual.



#### 4.1.6 Sixth Step: Iteration and Temperature Update

Iteration update is done each time a new solution set is evaluated, after the maximum number of the iteration is reached, the temperature is updated by multiplying it with the reducing coefficient. After a temperature update, the current iteration number is set to 1 again.

#### 4.1.7 Seventh Step: Stopping Criteria

The stopping criteria of the algorithm is determined to be when the temperature is dropped low enough that the effect to the objective function value from further reducing the temperature is negligible ( $T < 10^{-4}$ ).

#### 4.1.8 Last Step: Algorithm Validation

Algorithm validation is done to check for any error in the algorithm coding. To validate the algorithm, a simple test data set is used as experiment. Each function of the algorithm is then checked whether the result is correct or not.

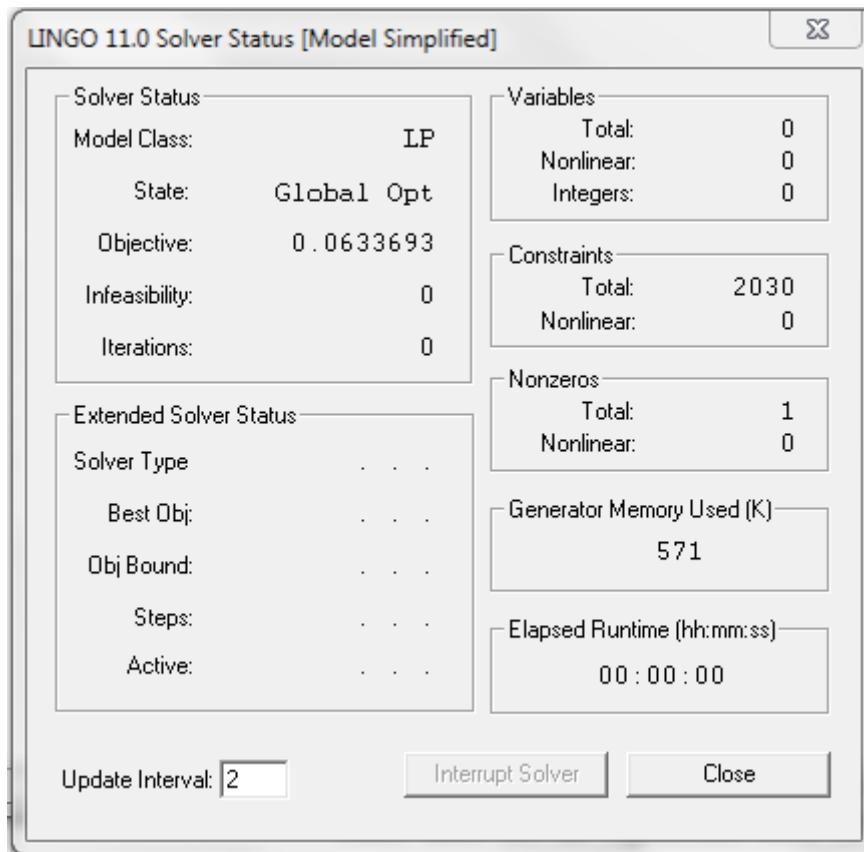
The validation for the mathematical model also provided here. LINGO software is used to validate the model. By conducting an experiment with known dataset, the result can be compared with normal calculation to check the validity of the experiment result.

```
Local optimal solution found.
Objective value:                0.6336934E-01
Objective bound:                0.000000
Infeasibilities:                0.000000
Extended solver steps:          1
Total solver iterations:        13639371

Model Title: Crude Oil Tanks Assignment Problem
```

**Figure 4. 4 Result Report from LINGO Experiment**

The validation process for the mathematical model is done by inputting the original solution set of June 2017 assignment using the original input data such as the shipment schedule and storage tank initial condition.



**Figure 4. 5 LINGO Solver Status Example**

In the original June 2017 assignment, the solution set satisfy all constraints with the final objective function value of 0.063. By using the completed set from the original data, it can be seen that the result that shown by the model is exactly the same without any error with all constraints satisfied. The computation time is instant since there are no more open variable left. From this experiment result it is understood that the mathematical model is valid.

For the simulated annealing algorithm, the validation method is done by checking whether the solution set generated by the algorithm satisfy all constraint of the mathematical model, regardless of the objective function result. The experiment will be done by running the algorithm function which is used for generating new solution set. The function command line is provided as follows:

```
[ tx1, tx2, TransferSet, HLoadSet, MLoadSet, LLoadSet,
HbLoadSet, TotalH, TotalM, TotalL, TotalHb, Z ] = NewSolution(
Target, Schedule, Stock, Tank, Day, TransferLimit, tb1, tb2, Tx)
```

The validation experiment will be done using the input data for June 2017 assignment, with parameters of Tx equal to 10, tb1 and tb2 of 3 and 6 respectively. The experiment will be replicated five times, with result as provided in the following Table 4.1:

**Table 4. 1 Validation Experiment Result**

Exp.	Total H	Total M	Total L	Total BYU	Objective Function	Error
1	1784.60	2623.90	818.10	473.40	0.0923	No
2	1713.30	2556.90	743.00	686.80	0.1421	No
3	1630.00	2696.90	731.40	641.70	0.0971	No
4	1730.60	2610.30	750.20	608.90	0.1209	No
5	1703.30	2558.50	894.70	543.50	0.0833	No

The experiment shows that from five replication, the function is capable of generating a solution set which satisfy all constraints without error. Granted that the value of the objective function is not minimized enough, however the result is valid and acceptable for the validation experiment.

By the nature of the function for the new solution generation, it is possible for a solution set that do not conform to the constraints to be generated. However, this will be detected by a constraint check within the function, in which this solution set will be tagged as invalid. The function will then redo the solution generation until a valid one appear, therefore proving that any solution set generated by the algorithm will conform to the mathematical model constraints. Since there are no significant error in the result of the algorithm, the simulated annealing algorithm is considered valid and applicable to solve problem with larger dataset.

## **4.2 Data Collection**

The oil refinery uses several oil tanks to store crude oil. Each of these tanks are large in size, capable of storing up to 800 thousand barrels (MB) of crude oil regardless of crude classifications. These stored crude oil are transferred daily bit by bit into the refinery for further processing. To replenish the crude oil stock, the refinery schedule crude oil shipment on monthly basis. Here are the shipment schedule of crude oil for the month of June and July 2017.

**Table 4. 2 Crude Oil Shipment Schedule for June 2017**

NO	DATE	KAPAL	CRUDE	QTY	CLASS
1	3	SHIP A	HANDIL	200,000	L
			SENIPAH	200,000	L
			HANDIL	200,000	L
2	8	SHIP B	BANYU URIP	400,000	H
			HANDIL	100,000	L
3	10	SHIP C	BONNY LIGHT	950,000	M
4	15	SHIP D	ESCRAVOS	950,000	M
5	18	PIPELINE 1	SEPINGGAN	100,000	M
6	19	SHIP E	BANYU URIP	400,000	H
			MADURA	140,000	L
7	22	SHIP F	BANYU URIP	400,000	H
			HANDIL	100,000	L
8	26	SHIP G	BONNY LIGHT	950,000	M
9	29	PIPELINE 2	SEPINGGAN	100,000	M

**Table 4. 3 Crude Oil Shipment Schedule for July 2017**

NO	DATE	KAPAL	CRUDE	QTY	CLASS
1	1	SHIP H	BEKAPAI	200,000	M
			SENIPAH	200,000	L
			HANDIL	100,000	L
2	4	SHIP I	BELANAK	100,000	M
			BELIDA	185,000	L
			LANGSA	90,000	M
3	7	SHIP J	AZERI	950,000	H
4	10	SHIP K	SERIA	600,000	L
5	13	SHIP L	MADURA	150,000	L
			JATIBARANG	200,000	M
6	15	PIPELINE 3	SEPINGGAN	150,000	M
7	16	SHIP M	BANYU URIP	400,000	H
			BEKAPAI	100,000	M
8	21	SHIP N	QUA IBOE	950,000	M
9	25	SHIP O	BONNY LIGHT	950,000	M
10	28	PIPELINE 4	SEPINGGAN	50,000	M
11	30	SHIP P	BANYU URIP	400,000	H
			HANDIL	100,000	L

One of the objective of this research is to help create crude oil tanks assignment with result that is close to the initial cumulative class composition target. The cumulative class composition target are month based. In Table 4.4 below the class composition target for June and July 2017 are provided. As it can be seen, the target for both of these month are identical.

**Table 4. 4 Crude Class Composition Percentage Target for June and July 2017**

	Jun-17	Jul-17
Light	17%	17%
Medium	48%	48%
Heavy	35%	35%

To properly assign which oil tanks will be used to transfer crude oil into the refinery while considering the crude oil class composition, the initial data of crude oil level and its class composition in each tanks are required. Here are the report of each tank condition on 00:00 of 1<sup>st</sup> day in June and July 2017.

**Table 4. 5 Oil Tanks Initial Condition on June 1<sup>st</sup> 2017**

A				B				C			
1	AZE	H	206.71	1	BYU	H	423.08	1	ARJ	M	5.11
2	ESC	M	101.81	2	ESC	M	111.21	2	SEN	L	25.17
3	HAN	L	6.67	3	HAN	L	64.33	3	AZE	H	21.47
4	SEP	M	49.18	4	SEP	M	28.70	4	SEP	M	6.23
5				5	BON	M	91.61	5	JATL	L	8.75
6				6	BEK	M	5.29	6	BEK	M	12.47
7				7				7	KIK	M	11.66
8				8				8	BYU	H	7.46
9				9				9	MAD	L	47.00
10				10				10	BON	M	82.53

D				E				F			
1	BYU	H	77.68	1	QUA	M	74.68	1	SEP	M	1.16
2	ESC	M	19.55	2	BYU	H	30.54	2	ESC	M	3.27
3	BEK	M	21.35	3	AZE	H	646.48	3	AZE	H	78.07
4	KIK	M	36.00	4	SEN	L	29.09	4	QUA	M	23.81
5	BON	M	571.05	5				5	BYU	H	99.24
6	AZE	H	49.18	6				6			

**Table 4. 6 Oil Tanks Initial Condition on July 1<sup>st</sup> 2017**

A				B				C			
1	AZE	H	624.71	1	BYU	H	11.64	1	ARJ	M	0.07
2	ESC	M	101.81	2	ESC	M	92.59	2	SEN	L	0.37
3	HAN	L	6.67	3	HAN	L	1.77	3	AZE	H	12.16
4	SEP	M	49.18	4	SEP	M	0.79	4	SEP	M	0.09
5				5	BON	M	675.28	5	JATL	L	0.13
6				6	BEK	M	0.15	6	BEK	M	0.18
7				7				7	KIK	M	0.17
8				8				8	BYU	H	400.11
9				9				9	MAD	L	40.69
10				10				10	BON	M	302.44
11				11				11	HAN	L	5.44

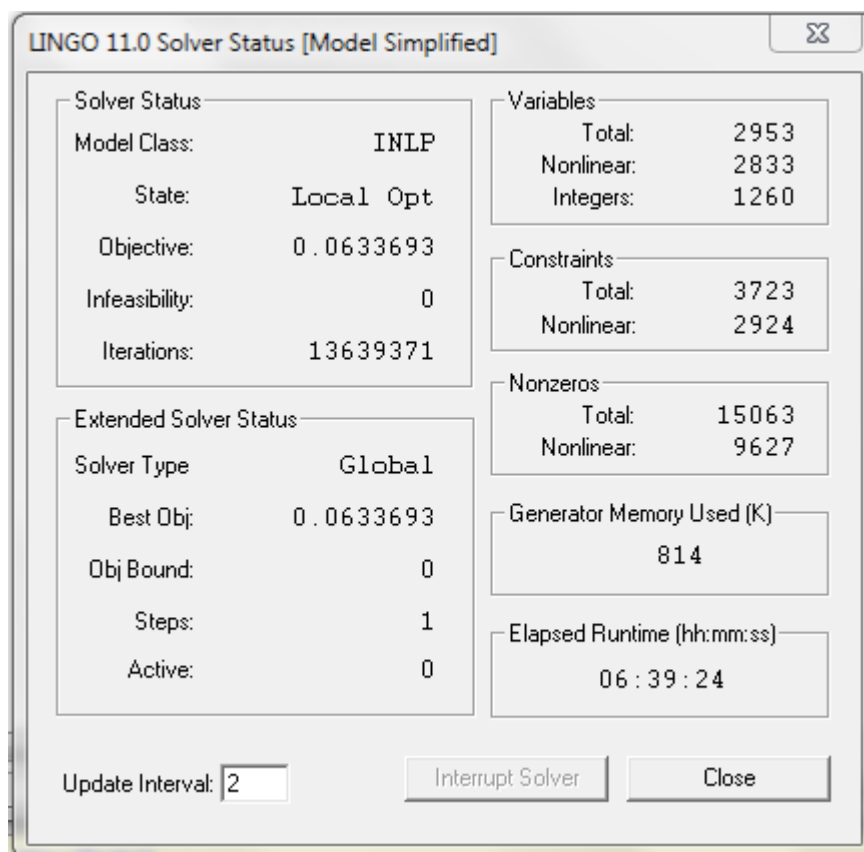
D				E				F			
1	BYU	H	409.06	1	QUA	M	4.74	1	SEP	M	0.11
2	ESC	M	0.53	2	BYU	H	1.94	2	ESC	M	0.30
3	BEK	M	0.58	3	AZE	H	41.07	3	AZE	H	7.16
4	KIK	M	0.98	4	SEN	L	1.85	4	QUA	M	2.18
5	BON	M	15.58	5	ESC	M	154.59	5	BYU	H	9.11
6	AZE	H	1.34	6	SEP	M	59.15	6	HAN	L	45.68
7	SEN	L	15.58	7	MAD	L	39.44	7	SEN	L	16.90
8	HAN	L	86.15	8				8	BON	M	89.11

### 4.3 Experiment and Result

The experiment will be based on the existing condition which derived from the collected data. There will be three experiment in total, first one will be based on original existing data of June 2017. The next one will be based on the original existing data of July 2017. Finally, the last one will use the best result of the June 2017 experiment as input for the new July experiment. The experiment will be done using MATLAB software. The software will be used to run the simulated annealing algorithm to solve the researched problem.

The experiments will be conducted using parameters as follows: Initial Temperature (T) of 10, Temperature Reduction Coefficient (cr) of 0.5, and 25 as the maximum number of iteration. Each experiment will be done in five replication each to find the best possible solution.

In the first experiment, an attempt to solve the model using LINGO solver is also done with the result as shown in Figure 4.6 below.



**Figure 4. 6 LINGO Experiment Solver Status**

The recapitulation of the result from all three experiment are provided in the following Table 4.7 – Table 4.15:

**Table 4. 7 Final Composition Recapitulation of Experiment 1**

	Total Transferred Crude			Percentage		
	H	M	L	H %	M %	L %
Target	1995.00	2736.00	969.00	35.0%	48.0%	17.0%
Original Data	2175.61	2664.04	860.35	38.2%	46.7%	15.1%
Replication 1	2047.19	2796.20	856.61	35.9%	49.1%	15.0%
Replication 2	2166.85	2700.90	832.25	38.0%	47.4%	14.6%
Replication 3	2027.40	2775.06	897.54	35.6%	48.7%	15.7%
Replication 4	2103.05	2752.30	844.65	36.9%	48.3%	14.8%
Replication 5	1981.25	2801.60	917.15	34.8%	49.2%	16.1%

**Table 4. 8 Target Achievement Deviation for Experiment 1**

	Total			Percentage			Deviation	
	H	M	L	H %	M %	L %	Total	%
Original Data	180.61	71.96	108.65	3.2%	1.3%	1.9%	361.22	6.3%
Replication 1	52.19	60.2	112.39	0.9%	1.1%	2.0%	224.78	3.9%
Replication 2	171.85	35.1	136.75	3.0%	0.6%	2.4%	343.7	6.0%
Replication 3	32.4	39.06	71.46	0.6%	0.7%	1.3%	142.92	2.5%
Replication 4	108.05	16.3	124.35	1.9%	0.3%	2.2%	248.7	4.4%
Replication 5	13.75	65.6	51.85	0.2%	1.2%	0.9%	131.2	2.3%

**Table 4. 9 Computation Time & BYU Target Achievement for Experiment 1**

Target	Time Target		BYU Target	
	< 1 Minutes	-	> 1000	-
Original Data	~3 Hours	Not Achieved	1006.15	Achieved
Replication 1	7.25 Seconds	Achieved	632.89	Not Achieved
Replication 2	8.00 Seconds	Achieved	641.58	Not Achieved
Replication 3	6.22 Seconds	Achieved	859.19	Not Achieved
Replication 4	8.20 Seconds	Achieved	645.10	Not Achieved
Replication 5	6.36 Seconds	Achieved	827.42	Not Achieved

**Table 4. 10 Final Composition Recapitulation of Experiment 2**

	Total Transferred Crude			Percentage		
	H	M	L	H %	M %	L %
Target	2061.50	2827.20	1001.30	35.0%	48.0%	17.0%
Original Data	1999.44	2802.34	1088.22	33.9%	47.6%	18.5%
Replication 1	2020.70	2826.90	1042.40	34.3%	48.0%	17.7%
Replication 2	1984.90	2837.50	1067.60	33.7%	48.2%	18.1%
Replication 3	2051.20	2813.20	1025.60	34.8%	47.8%	17.4%
Replication 4	2144.04	2833.50	912.46	36.4%	48.1%	15.5%
Replication 5	2043.70	2863.63	982.67	34.7%	48.6%	16.7%

**Table 4. 11 Target Achievement Deviation for Experiment 2**

	Total			Percentage			Deviation	
	H	M	L	H %	M %	L %	Total	%
Original Data	62.06	24.86	86.92	1.1%	0.4%	1.5%	173.84	3.0%
Replication 1	40.8	0.3	41.1	0.7%	0.0%	0.7%	82.2	1.4%
Replication 2	76.6	10.3	66.3	1.3%	0.2%	1.1%	153.2	2.6%
Replication 3	10.3	14	24.3	0.2%	0.2%	0.4%	48.6	0.8%
Replication 4	82.54	6.3	88.84	1.4%	0.1%	1.5%	177.68	3.0%
Replication 5	17.8	36.43	18.63	0.3%	0.6%	0.3%	72.86	1.2%



**Table 4. 12 Computation Time & BYU Target Achievement for Experiment 2**

Target	Time Target		BYU Target	
	< 1 Minutes	-	> 1000	-
Original Data	~3 Hours	Not Achieved	1070.40	Achieved
Replication 1	5.80 Seconds	Achieved	600.36	Not Achieved
Replication 2	6.17 Seconds	Achieved	617.26	Not Achieved
Replication 3	5.25 Seconds	Achieved	636.37	Not Achieved
Replication 4	5.64 Seconds	Achieved	660.39	Not Achieved
Replication 5	5.49 Seconds	Achieved	715.86	Not Achieved

**Table 4. 13 Final Composition Recapitulation of Experiment 3**

	Total Transferred Crude			Percentage		
	H	M	L	H %	M %	L %
Target	2061.50	2827.20	1001.30	35.0%	48.0%	17.0%
Original Data	1999.44	2802.34	1088.22	33.9%	47.6%	18.5%
Replication 1	2059.38	2839.20	991.42	35.0%	48.2%	16.8%
Replication 2	2044.50	2785.60	1059.90	34.7%	47.3%	18.0%
Replication 3	2132.20	2786.40	971.40	36.2%	47.3%	16.5%
Replication 4	2031.20	2897.63	961.17	34.5%	49.2%	16.3%
Replication 5	2128.70	2808.28	953.02	36.1%	47.7%	16.2%

**Table 4. 14 Target Achievement Deviation for Experiment 3**

	Total			Percentage			Deviation	
	H	M	L	H %	M %	L %	Total	%
Original Data	62.06	24.86	86.92	1.1%	0.4%	1.5%	173.84	3.0%
Replication 1	2.12	12	9.88	0.0%	0.2%	0.2%	24	0.4%
Replication 2	17	41.6	58.6	0.3%	0.7%	1.0%	117.2	2.0%
Replication 3	70.7	40.8	29.9	1.2%	0.7%	0.5%	141.4	2.4%
Replication 4	30.3	70.43	40.13	0.5%	1.2%	0.7%	140.86	2.4%
Replication 5	67.2	18.92	48.28	1.1%	0.3%	0.8%	134.4	2.3%

**Table 4. 15 Computation Time & BYU Target Achievement for Experiment 3**

Target	Time Target		BYU Target	
	< 1 Minutes	-	> 1000	-
Original Data	~3 Hours	Not Achieved	1070.40	Achieved
Replication 1	9.07 Seconds	Achieved	638.68	Not Achieved
Replication 2	8.40 Seconds	Achieved	726.07	Not Achieved
Replication 3	8.73 Seconds	Achieved	643.12	Not Achieved
Replication 4	8.36 Seconds	Achieved	786.55	Not Achieved
Replication 5	8.50 Seconds	Achieved	636.64	Not Achieved

## 4.4 Result Analysis

Here provided the analysis of the experiment results:

### 4.4.1 Existing Condition Analysis

For the existing condition, the achieved result of the manual simulation method which is currently being used by the refinery worker is acceptable. The final assignment result tends to be close enough with the monthly target. However, to reach this point the amount of revision needed is very high, and consumed a lot of time. Depending on the worker skills, it could take from multiple hours to several days. Including human error factor, this condition is not ideal for the company as the high variation of time needed to complete the task shows that this task is not standardized enough. If the time needed to complete this task can be reduced considerably, the additional free time can be utilized by the worker to do other work.

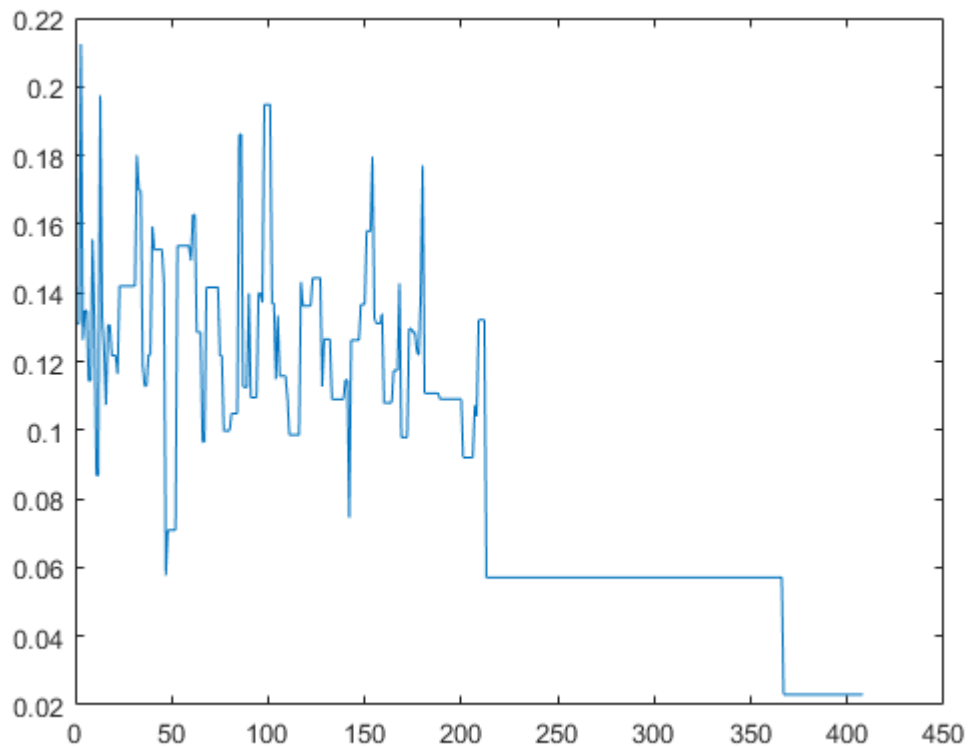
### 4.4.2 Analysis of Experiment Result

In the first experiment for June 2017 data, aside from the algorithm result, a comparison with exact calculation method using LINGO software is provided. It can be seen that due to the complexity of the problem, the global solver method of LINGO software is unable to find the global optima in reasonable amount of time. With calculation time of 6 hour and 39 minutes, the LINGO software yields a result of 0.0633 which is not a significant improvement toward the result from simulated annealing algorithm. Due to the large amount of computation time needed, the comparison of LINGO software are provided in this experiment only.

The first experiment using simulated annealing algorithm yield the best result of 2.3 % deviation. Compared with the real final assignment result for June 2017, there is an improvement of 4% less deviation. However, it can be seen that from all five replications, the algorithm is currently unable to generate solution that satisfy the BYU target of 1000 MB, with the best result for experiment 1 is only 859.19 MB from the third replication.

The total computation time for the first experiment in average is 7.2 seconds, which is considered as a significant improvement from the initial time of around several hours.

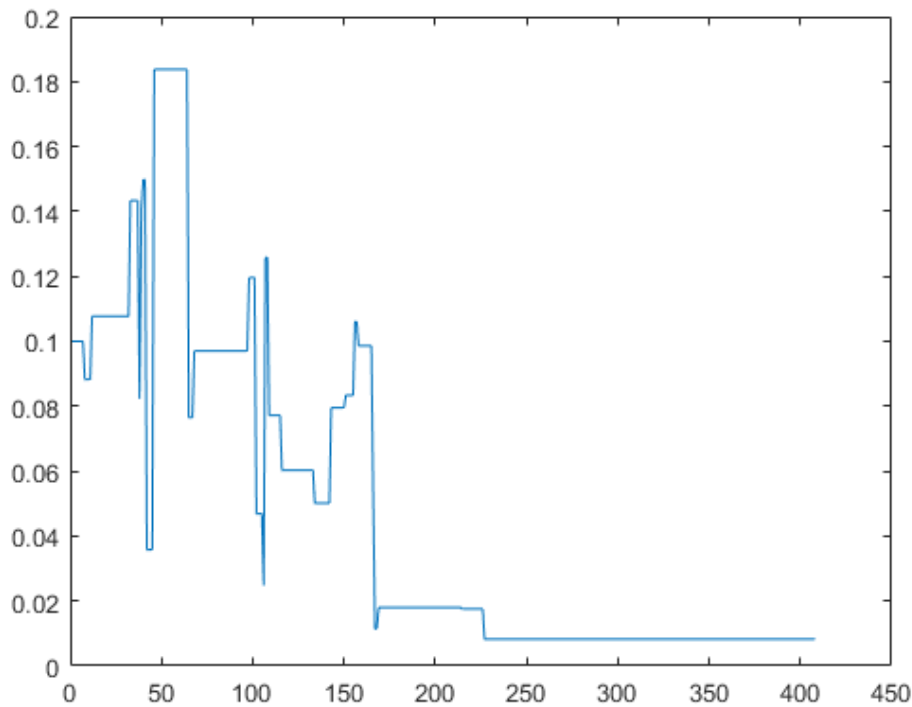
From the objective plot of experiment 1 shown on Figure 4.7 below, it can be seen that the result undergone fluctuative change at the beginning, before finally reaching the lowest point nearing the end of the iteration. This shows that the simulated annealing algorithm is working as intended.



**Figure 4. 7 Objective Plot for Experiment 1**

In the next experiment for July 2017 data, the simulated annealing algorithm yield the best result of 0.8 % deviation from the target. Compared with the real final assignment result for July 2017, there is an improvement of 2.2% less deviation from the target. This shows that the algorithm is capable of returning a near optimal result in a very short amount of time. However, as it is the case with the first experiment, the BYU target is unable to be achieved, with the best result of 715.86 MB from the last replication.

The total computation time for the second experiment in average is 5.67 seconds, which is considered as a significant improvement from the initial time of around several hours.



**Figure 4. 8 Objective Plot for Experiment 2**

The last experiment is conducted for July 2017 data with initial stock data taken from the first experiment best result. Using simulated annealing algorithm yield the best result of 0.4 % deviation from the target. Compared with the real final assignment result for July 2017, there is an improvement of 2.6% less deviation from the target. However, as it is the case with the first experiment, the BYU target is unable to be achieved, with the best result of 786.55 MB from the 4<sup>th</sup> replication.

The total computation time for the second experiment in average is 8.61 seconds, which is considered as a significant improvement from the initial time of around several hours.

The fact that the algorithm is still unable to generate a solution that satisfy the cumulative BYU target shows that the logic of solution generation in the algorithm could still be improved. Despite that, the generated result can still be useful as a starting point for future adjustment. With these results, overall it can be said that the simulated annealing algorithm is very capable of returning a near optimal result in a very short amount of times.

## **CHAPTER V**

### **CONCLUSION**

The conclusion of this research and the suggestion for future research from the author will be provided in this chapter.

#### **5.1 Conclusion**

Based on the experiment result, it can be concluded that:

1. The experiment showed that the simulated annealing algorithm is capable of improving the tank assignment plan from the initial result which was done using manual method. The target achievement result is improved in all of the experiment done, with the deviation from the target is decreased by 4%, 2.2%, and 2.6% respectively. Overall the target deviation can be reduced up to below 1% for each experiment.
2. The algorithm is capable of significantly reducing the time needed to do the assignment from the initial time of around several hours into within several seconds. The average computation time for each experiments are 7.2 seconds, 5.67 seconds, and 8.61 seconds respectively. Overall the time needed to generate the near optimal solution can be reduced up to under one minute for each assignment.
3. It is understood that the composition result from the simulated annealing algorithm developed for this is not the most optimal, with experiment results showed that not a single one of the generated solution set managed to satisfy the BYU crude cumulative target. Therefore, it is advised to treat the output of this algorithm as a starting point for the worker to do further adjustment in later revision. Using this method will give better overall result with less difficulty and less time needed compared with the initial method.

## **5.2 Suggestion**

Here are some suggestion for further research:

1. In future research, other type of metaheuristics algorithm can be considered in comparison to the simulated annealing algorithm.
2. Instead of focusing in reducing computation time, future research can choose to focus in getting the most optimal result, which does not necessarily need metaheuristics algorithm.
3. Future research may consider other related factor that are not considered in this research, such as scheduling optimization as well as the cost aspect of oil transfer.

## REFERENCES

- Hou, Y., Wu, N.Q., Zhou, M.C., & Li, Z.W. (2015), "Pareto-Optimization for Scheduling of Crude Oil Operations in Refinery via Genetic Algorithm", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 3, pp. 517-530.
- Hou, Yan & Wu, Nai Qi. (2015), "Modeling and Optimization for Short-Term Scheduling of Crude Oil Operations in Refinery", *Proceedings of 2015 IEEE 12th International Conference on Networking, Sensing and Control*, IEEE, Taipei, Taiwan.
- Hyne, Norman J. (2012), *Nontechnical Guide to Petroleum Geology, Exploration, Drilling, and Production*. 3<sup>rd</sup> Edition. Oklahoma: PennWell Books.
- Kementerian BUMN Indonesia. (2016), *Laporan Kinerja Kementerian BUMN 2016*, Kementerian BUMN Indonesia.
- Ledesma, S., Aviña, G., & Sanchez, R. (2008), "Practical Considerations for Simulated Annealing Implementation", *Simulated Annealing*, Ed: Tan, Cher Ming, InTech.
- Miesner, Thomas O. and Leffler, William L. (2006), *Oil & Gas Pipelines in Nontechnical Language*, Oklahoma: PennWell Books.
- Nikolaev, Alexander G. and Jacobson, Sheldon H. (2010), "Simulated Annealing", *Handbook of Metaheuristics*, 2<sup>nd</sup> Edition, Gendreau, Michel (Ed.) Springer.
- Puvanasvaran, A/LA. Perumal (2014), *Lean Manufacturing Tools and Techniques*, 1<sup>st</sup> Edition. Melaka: Penerbit UTeM.
- Ribeiro, G.M., Mauri, G.R., & Lorena, L.A.N. (2010), "A Simple and Robust Simulated Annealing Algorithm for Scheduling Workover Rigs on Onshore Oil Fields", *Computers & Industrial Engineering*, Vol. 60, pp. 519–526.
- Rocha, R., Grossmann, I.E., & de Aragão, M.V.S.P. (2009), "Petroleum Allocation at Petrobras: Mathematical Model and A Solution Algorithm", *Computers & Chemical Engineering*, Vol. 33, No. 12, pp. 2123-2133.

- Saharidis, Georgios K.D. and Ierapetritou, Marianthi G. (2009), “Scheduling of Loading and Unloading of Crude Oil in a Refinery with Optimal Mixture Preparation”, *Industrial & Engineering Chemistry Research*, Vol. 48, No. 5, pp. 2624–2633.
- Santosa, B. & Willy, P. (2011), *Metoda Metaheuristik Konsep dan Implementasi*. 1st Edition. Surabaya: Prima Printing.
- Saraiva, J.T., Pereira, M.L., Mendes, V.T., & Sousa, J.C. (2011), “A Simulated Annealing based approach to solve the generator maintenance scheduling problem”, *Electric Power Systems Research*, Vol. 81, pp. 1283–1291.
- Shah, N.K., Li, Z. & Ierapetritou, Marianthi G. (2011), “Petroleum refining operations: Key issues, advances, and opportunities”, *Industrial & Engineering Chemistry Research*, Vol. 50, No. 3, pp. 1161–1170.
- Speight, James G. (2014), *The Chemistry and Technology of Petroleum*. 5<sup>th</sup> Edition. Florida: CRC Press.
- Taha, Hamdy A. (2007), *Operations Research: An Introduction*. 8<sup>th</sup> Edition. New Jersey: Pearson Prentice Hall.
- Talbi, El-Ghazali. (2009), *Metaheuristics: From Design to Implementation*. New Jersey: John Wiley & Sons.
- Wichers, Johan. (2013), *Guide to Single Point Moorings*, WMooring Inc., CreateSpace Independent Publishing Platform.
- Wignjosoebroto, Sritomo (2003), *Pengantar Teknik & Manajemen Industri*. 1<sup>st</sup> Edition. Surabaya: Guna Widya.



## APPENDIX A

### INPUT DATA

Here provided the input data for this research experiments which derived from the original collected data provided in chapter IV:

#### Input Data for Experiment 1 (June 2017):

Shipment schedule:

	1-Jun	2-Jun	3-Jun	4-Jun	5-Jun	6-Jun	7-Jun	8-Jun	9-Jun	10-Jun
H	558	204	92	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	100
L	0	0	200	200	108	0	0	92	8	92
BYU	0	0	0	0	0	0	0	400	0	0

	11-Jun	12-Jun	13-Jun	14-Jun	15-Jun	16-Jun	17-Jun	18-Jun	19-Jun	20-Jun
H	0	0	0	0	0	0	0	0	0	0
M	550	208	0	0	242	600	108	150	92	0
L	0	0	0	0	0	0	0	0	100	40
BYU	0	0	0	0	0	0	0	0	0	0

	21-Jun	22-Jun	23-Jun	24-Jun	25-Jun	26-Jun	27-Jun	28-Jun	29-Jun	30-Jun
H	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	200	400	258	50	0
L	0	100	0	0	0	0	0	0	0	0
BYU	308	92	308	0	0	92	0	0	0	0

Crude oil initial stock:

	A	B	C	D	E	F
H	207	0	22	49	646	78
M	151	237	118	648	75	28
L	6	64	81	0	29	0
BYU	0	423	7	78	31	100
Total	364	724	228	775	781	206

Class composition target:

Light	Medium	Heavy
17%	48%	35%

**Input Data for Experiment 2 (July 2017):**

Shipment schedule:

	1-Jul	2-Jul	3-Jul	4-Jul	5-Jul	6-Jul	7-Jul	8-Jul	9-Jul	10-Jul
H	0	0	0	0	0	0	150	600	108	92
M	92	0	108	192	90	0	0	0	0	0
L	200	100	0	0	0	93	92	0	0	200
BYU	0	0	0	0	0	0	0	0	0	0

	11-Jul	12-Jul	13-Jul	14-Jul	15-Jul	16-Jul	17-Jul	18-Jul	19-Jul	20-Jul
H	0	0	0	0	0	0	0	0	0	0
M	0	0	0	108	150	192	100	0	0	0
L	308	0	242	0	0	0	0	0	0	0
BYU	0	0	0	0	0	0	0	308	0	0

	21-Jul	22-Jul	23-Jul	24-Jul	25-Jul	26-Jul	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul
H	0	0	0	0	0	0	0	0	0	0	0
M	150	600	108	0	242	600	108	50	0	92	0
L	0	0	0	0	0	0	0	0	0	0	8
BYU	92	0	0	0	0	0	0	0	0	400	0

Crude oil initial stock:

	A	B	C	D	E	F
H	625	0	12	1	41	7
M	151	769	303	18	219	92
L	6	2	47	102	41	63
BYU	0	11	400	409	2	9
Total	782	782	762	530	303	171

Class composition target:

Light	Medium	Heavy
17%	48%	35%

**Input Data for Experiment 3 (July 2017 using Experiment 1 best result):**

Crude oil initial stock:

	A	B	C	D	E	F
H	2	0	20	5	646	30
M	70	304	149	152	75	663
L	35	8	5	55	29	70
BYU	15	104	616	228	31	18
Total	122	416	790	440	781	781





```

!--(3.5 - 3.8)--;
@FOR(LINKD(i):
[Daily_Heavy_Percentage] HeavyTransferPercentage(i) =
@SUM(LINKDT(i,j):
HeavyPercentage(i,j)*TransferAmount(i,j))
/@SUM(LINKDT(i,j):TransferAmount(i,j)));

@FOR(LINKD(i):
[Daily_Medium_Percentage] MediumTransferPercentage(i) =
@SUM(LINKDT(i,j):
MediumPercentage(i,j)*TransferAmount(i,j))
/@SUM(LINKDT(i,j):TransferAmount(i,j)));

@FOR(LINKD(i):
[Daily_Light_Percentage] LightTransferPercentage(i) =
@SUM(LINKDT(i,j):
LightPercentage(i,j)*TransferAmount(i,j))
/@SUM(LINKDT(i,j):TransferAmount(i,j)));

@FOR(LINKD(i):
[Daily_BYU_Percentage] BYUTransferPercentage(i) =
@SUM(LINKDT(i,j):
BYUPercentage(i,j)*TransferAmount(i,j))
/@SUM(LINKDT(i,j):TransferAmount(i,j)));

!--(3.9 - 3.12)--;
@FOR(LINKDT(i,j): [Heavy_Stock_Percentage] HeavyPercentage(i,j) =
HeavyStock(i,j)/TotalStock(i,j));

@FOR(LINKDT(i,j): [Medium_Stock_Percentage] MediumPercentage(i,j)
= MediumStock(i,j)/TotalStock(i,j));

@FOR(LINKDT(i,j): [Light_Stock_Percentage] LightPercentage(i,j) =
LightStock(i,j)/TotalStock(i,j));

@FOR(LINKDT(i,j): [BYU_Stock_Percentage] BYUPercentage(i,j) =
BYUStock(i,j)/TotalStock(i,j));

!--(3.13 - 3.16)--;
@FOR(LINKDT(i,j)|i #NE# 1: [Heavy_Stock] HeavyStock(i,j) =
HeavyStock(i-1,j) - (HeavyPercentage(i-1,j)*TransferAmount(i-1,j))
+ LoadH(i-1,j));

@FOR(LINKDT(i,j)|i #NE# 1: [Medium_Stock] MediumStock(i,j) =
MediumStock(i-1,j) - (MediumPercentage(i-1,j)*TransferAmount(i-
1,j)) + LoadM(i-1,j));

@FOR(LINKDT(i,j)|i #NE# 1: [Light_Stock] LightStock(i,j) =
LightStock(i-1,j) - (LightPercentage(i-1,j)*TransferAmount(i-1,j))
+ LoadL(i-1,j));

@FOR(LINKDT(i,j)|i #NE# 1: [BYU_Stock] BYUStock(i,j) =
BYUStock(i-1,j) - (BYUPercentage(i-1,j)*TransferAmount(i-1,j)) +
LoadHb(i-1,j));

!--(3.17)--;
@FOR(LINKDT(i,j): [Total_Stock] TotalStock(i,j) =
HeavyStock(i,j)+MediumStock(i,j)+LightStock(i,j)+BYUStock(i,j));

```

```

!--(3.18 - 3.21)--;
[Total_Heavy_Transfer] TotalHeavy =
@SUM(LINKDT(i,j):HeavyPercentage(i,j)*TransferAmount(i,j));

[Total_Medium_Transfer] TotalMedium =
@SUM(LINKDT(i,j):MediumPercentage(i,j)*TransferAmount(i,j));

[Total_Light_Transfer] TotalLight =
@SUM(LINKDT(i,j):LightPercentage(i,j)*TransferAmount(i,j));

[Total_BYU_Transfer] TotalBYU =
@SUM(LINKDT(i,j):BYUPercentage(i,j)*TransferAmount(i,j));

!-----Main Constraints-----;

!--(3.22)--;
@FOR(LINKD(i): [Transfer_Amount_Equation]
@SUM(LINKDT(i,j):TransferAmount(i,j)) = IT);

!--(3.23 - 3.24)--;
@FOR(LINKDT(i,j): [Tank_Lower_Limit] TotalStock(i,j)>= 120);
@FOR(LINKDT(i,j): [Tank_Upper_Limit] TotalStock(i,j)<= 795);

!--(3.25 - 3.26)--;
@FOR(LINKD(i):
[BYU_Daily_Limit] BYUTransferPercentage(i) <= 0.25);

[BYU_Total_Target] TotalBYU >= 1000;

!--(3.27 - 3.28)--;
@FOR(LINKD(i): [Transfer_Limit]
@SUM(LINKDT(i,j): TransferStatus(i,j)) <= 2);

@FOR(LINKDT(i,j): [Transfer_Slot_Equation]
TransferStatus(i,j)*TransferAmount(i,j) = TransferAmount(i,j));

!--(3.29)--;
@FOR(LINKDT(i,j)|i #NE# 1: [Afterload_Wait] (LoadH(i-1,j)+LoadM(i-
1,j)+LoadL(i-1,j)+LoadHb(i-1,j))*TransferAmount(i,j) = 0);

!--(3.30)--;
@FOR(LINKDT(i,j)|i #NE# 1: [Transfer_Until_Empty]
TransferStatus(i,j)*TransferStatus(i-1,j)*(TotalStock(i,j)-130-
LoadH(i-1,j)-LoadM(i-1,j)-LoadL(i-1,j)-LoadHb(i-1,j)) >=
TransferStatus(i-1,j)*(TotalStock(i,j)-130-LoadH(i-1,j)-LoadM(i-
1,j)-LoadL(i-1,j)-LoadHb(i-1,j)));

!--(3.31 - 3.34)--;
@FOR(LINKD(i): [Daily_Heavy_Load_Equation]
@SUM(LINKDT(i,j):LoadH(i,j)) = ScheduleHeavy(i));
@FOR(LINKD(i): [Daily_Medium_Load_Equation]
@SUM(LINKDT(i,j):LoadM(i,j)) = ScheduleMedium(i));
@FOR(LINKD(i): [Daily_Light_Load_Equation]
@SUM(LINKDT(i,j):LoadL(i,j)) = ScheduleLight(i));
@FOR(LINKD(i): [Daily_BYU_Load_Equation]
@SUM(LINKDT(i,j):LoadHb(i,j)) = ScheduleBYU(i));

```

```

!--(3.35 - 3.39)--;
@FOR(LINKD(i):
[Load_Limit] @SUM(LINKDT(i,j): LoadStatus(i,j)) <= 2);

@FOR(LINKDT(i,j): [Heavy_Load_Slot_Equation]
LoadStatus(i,j)*LoadH(i,j) = LoadH(i,j));

@FOR(LINKDT(i,j): [Medium_Load_Slot_Equation]
LoadStatus(i,j)*LoadM(i,j) = LoadM(i,j));

@FOR(LINKDT(i,j): [Light_Load_Slot_Equation]
LoadStatus(i,j)*LoadL(i,j) = LoadL(i,j));

@FOR(LINKDT(i,j): [BYU_Load_Slot_Equation]
LoadStatus(i,j)*LoadHb(i,j) = LoadHb(i,j));

!-----Binary and Integer-----;

!--(3.40)--;
@FOR( LINKDT: @GIN( TransferAmount));
@FOR( LINKDT: @BND( 0, TransferAmount, IT));
@FOR( LINKDT: @MOD(TransferAmount,5) = 0);

!--(3.41)--;
@FOR( LINKDT: @BIN( TransferStatus));
@FOR( LINKDT: @BIN( LoadStatus));

!--(3.42)--;
@FOR( LINKDT: @GIN( LoadH));
@FOR( LINKDT: @GIN( LoadM));
@FOR( LINKDT: @GIN( LoadL));
@FOR( LINKDT: @GIN( LoadHb));
@FOR( LINKDT: @BND( 0, LoadH, 600));
@FOR( LINKDT: @BND( 0, LoadM, 600));
@FOR( LINKDT: @BND( 0, LoadL, 600));
@FOR( LINKDT: @BND( 0, LoadHb, 600));

!non-negativity constraint unneeded for LINGO;

END

```

*(This page is intentionally left blank)*



## APPENDIX C

### MATLAB FUNCTION

Here provided the full code of MATLAB function of the Simulated Annealing algorithm which developed in chapter IV:

#### *Simulated Annealing Function:*

```
function [ besttransferset, bestHloadset, bestMloadset,
bestLloadset, bestHbloadset, bestTotalH, bestTotalM, bestTotalL,
bestTotalHb, ObjectiveValue, computationtime ]
= SAlgorithm( T, cr, maxit, Target, Schedule, Stock, Tank, Day,
TransferLimit )

computationtime = cputime;
objective=[];
it=1;
Tx=T;

%Initial solution
[ Z0, TotalH0, TotalM0, TotalL0, TransferSet0, HLoadSet0,
MLoadSet0, LLoadSet0, HbLoadSet0, TotalHb0, tx1, tx2 ] =
InitSolution( Target, Schedule, Stock, Tank, Day, TransferLimit );

%Set initial solution as current best solution
currentbest=Z0;
besttransferset=TransferSet0;
bestHloadset=HLoadSet0;
bestMloadset=MLoadSet0;
bestLloadset=LLoadSet0;
bestHbloadset=HbLoadSet0;
bestTotalH=TotalH0;
bestTotalM=TotalM0;
bestTotalL=TotalL0;
bestTotalHb=TotalHb0;

%Store the choice of transfer tank for future comparison
tb1=tx1;
tb2=tx2;

%-Simulated Annealing Algorithm-
while Tx>=10^-4

    %Generate new solution
    [ tx1, tx2, TransferSet, HLoadSet, MLoadSet, LLoadSet,
      HbLoadSet, TotalH, TotalM, TotalL, TotalHb, Z ]
    = NewSolution( Target, Schedule, Stock, Tank, Day,
      TransferLimit, tb1, tb2, Tx);
```

```

%Compare with current best solution
if Z<currentbest && TotalHb >= 1000 %Accept if better
    currentbest=Z;
    besttransferset=TransferSet;
    bestHloadset=HLoadSet;
    bestMloadset=MLoadSet;
    bestLloadset=LLoadSet;
    bestHbloadset=HbLoadSet;
    bestTotalH=TotalH;
    bestTotalM=TotalM;
    bestTotalL=TotalL;
    bestTotalHb=TotalHb;
    tb1=tx1;
    tb2=tx2;
elseif Z == 1
    %Reject invalid solution
elseif TotalHb >= 600 %Compare using metropolis criteria
    deltaf=Z-currentbest;
    PE=exp(-deltaf/Tx); %Boltzmann probability
    if rand<PE
        currentbest=Z;
        besttransferset=TransferSet;
        bestHloadset=HLoadSet;
        bestMloadset=MLoadSet;
        bestLloadset=LLoadSet;
        bestHbloadset=HbLoadSet;
        bestTotalH=TotalH;
        bestTotalM=TotalM;
        bestTotalL=TotalL;
        bestTotalHb=TotalHb;
        tb1=tx1;
        tb2=tx2;
    end
end
objective=[objective;currentbest]; %Compile each iteration Z
it=it+1; %Iteration update
%Max iteration check
if it == maxit
    Tx=cr*Tx; %Reduce temperature if reaches max iteration
    it=1; %Iteration reset
end
end

ObjectiveValue=currentbest; %Save the best result
computationtime=cputime-computationtime; %Calculate computation
time
plot(objective);
end

```

### ***Function for generating Initial Solution:***

```
function [ Z0, TotalH0, TotalM0, TotalL0, TransferSet0, HLoadSet0,
MLoadSet0, LLoadSet0, HbLoadSet0, TotalHb0, tx1, tx2 ] =
InitSolution( Target, Schedule, Stock, Tank, Day, TransferLimit )

%Preallocate support variables
HeavyStock=zeros (Day, Tank) ;
MediumStock=zeros (Day, Tank) ;
LightStock=zeros (Day, Tank) ;
BYUStock=zeros (Day, Tank) ;
TotalStock=zeros (Day, Tank) ;

BYUTransferPercentage=zeros (Day) ;

HeavyPercentage=zeros (Day, Tank) ;
MediumPercentage=zeros (Day, Tank) ;
LightPercentage=zeros (Day, Tank) ;
BYUPercentage=zeros (Day, Tank) ;

TotalH=0;
TotalM=0;
TotalL=0;
TotalHb0=0;

%Preallocate solution set
TransferSet0=zeros (Day, Tank) ;
HLoadSet0=zeros (Day, Tank) ;
MLoadSet0=zeros (Day, Tank) ;
LLoadSet0=zeros (Day, Tank) ;
HbLoadSet0=zeros (Day, Tank) ;

%Initial Stock
for j=1:Tank
    HeavyStock (1, j)=Stock (1, j) ;
    MediumStock (1, j)=Stock (2, j) ;
    LightStock (1, j)=Stock (3, j) ;
    BYUStock (1, j)=Stock (4, j) ;
end

%Precalculate day 1 total stock
TotalStock (1, :) =
HeavyStock (1, :)+MediumStock (1, :)+LightStock (1, :)+BYUStock (1, :);

%Determine tanks for transfer by sorting the capacity
resort=1;
[~,idx]=sort (TotalStock (1, :), 'descend');
itx=sort ([idx (6) idx (5)]);

%Store initial transfer tanks choice for future comparison
tx1=itx (1);
tx2=itx (2);
```

```

%-----Main calculation-----
for i=1:Day
    for j=1:Tank
        %Calculate current total stock
        TotalStock(i,j) =
            HeavyStock(i,j)+MediumStock(i,j)
            +LightStock(i,j)+BYUStock(i,j);

        %Check if need re-sorting
        if TotalStock(i,idx(6))<= 220
            || TotalStock(i,idx(5))<= 220
                resort=1;
        elseif TotalStock(i,idx(1))>= 495
            || TotalStock(i,idx(2))>= 495
                resort=1;
        end

        %Calculate stock percentages
        HeavyPercentage(i,j)=HeavyStock(i,j)/TotalStock(i,j);
        MediumPercentage(i,j)=MediumStock(i,j)/TotalStock(i,j);
        LightPercentage(i,j)=LightStock(i,j)/TotalStock(i,j);
        BYUPercentage(i,j)=BYUStock(i,j)/TotalStock(i,j);
    End

    %Re-sort if needed
    if resort==1
        [~,idx]=sort(TotalStock(i,:));
        resort=0;
    end

    %Determine transfer value
    TransferSet0(i,idx(6))=TransferLimit/2;
    TransferSet0(i,idx(5))=TransferLimit/2;

    %Determine load value
    HLoadSet0(i,idx(1))=Schedule(1,i)/2+(mod(Schedule(1,i),2)/2);
    HLoadSet0(i,idx(2))=Schedule(1,i)/2-(mod(Schedule(1,i),2)/2);
    MLoadSet0(i,idx(1))=Schedule(2,i)/2+(mod(Schedule(2,i),2)/2);
    MLoadSet0(i,idx(2))=Schedule(2,i)/2-(mod(Schedule(2,i),2)/2);
    LLoadSet0(i,idx(1))=Schedule(3,i)/2+(mod(Schedule(3,i),2)/2);
    LLoadSet0(i,idx(2))=Schedule(3,i)/2-(mod(Schedule(3,i),2)/2);
    HbLoadSet0(i,idx(1))=Schedule(4,i)/2+(mod(Schedule(4,i),2)/2);
    HbLoadSet0(i,idx(2))=Schedule(4,i)/2-(mod(Schedule(4,i),2)/2);

    %Calculate next day stock
    for j=1:Tank
        HeavyStock(i+1,j)=HeavyStock(i,j)-
            (HeavyPercentage(i,j)*TransferSet0(i,j))+HLoadSet0(i,j);
        MediumStock(i+1,j)=MediumStock(i,j)-
            (MediumPercentage(i,j)*TransferSet0(i,j))+MLoadSet0(i,j);
        LightStock(i+1,j)=LightStock(i,j)-
            (LightPercentage(i,j)*TransferSet0(i,j))+LLoadSet0(i,j);
        BYUStock(i+1,j)=BYUStock(i,j)-
            (BYUPercentage(i,j)*TransferSet0(i,j))+HbLoadSet0(i,j);
    end
end

```

```

%-----

%BYU transfer percentage
up=zeros (Day);
down=zeros (Day);
for i=1:Day
    up(i)=0;
    down(i)=0;
    for j=1:Tank
        up(i)=up(i)+(BYUPercentage(i,j)*TransferSet0(i,j));
        down(i)=down(i)+TransferSet0(i,j);
    end
    BYUTransferPercentage(i)= up(i)/down(i);
end

%-----ending calculation-----

%Total transferred crude per class
for i=1:Day
    for j=1:Tank
        TotalH0=TotalH0+TransferSet0(i,j)*HeavyPercentage(i,j);
        TotalM0=TotalM0+TransferSet0(i,j)*MediumPercentage(i,j);
        TotalL0=TotalL0+TransferSet0(i,j)*LightPercentage(i,j);
        TotalHb0=TotalHb0+TransferSet0(i,j)*BYUPercentage(i,j);
    end
end

%Total H/M/L Percentage
H=(TotalH0+TotalHb0)/(TotalH0+TotalM0+TotalL0+TotalHb0);
M=(TotalM0)/(TotalH0+TotalM0+TotalL0+TotalHb0);
L=(TotalL0)/(TotalH0+TotalM0+TotalL0+TotalHb0);

%Objective Function
Z0= abs(Target(1)-H)+abs(Target(2)-M)+abs(Target(3)-L);

%BYU Constraint
for i=1:Day
    if BYUTransferPercentage(i) >0.25
        Z0=1;
        break;
    end
end

end

```

### ***Function for generating New Solution:***

```
function [ tx1, tx2, TransferSet, HLoadSet, MLoadSet, LLoadSet,
HbLoadSet, TotalH, TotalM, TotalL, TotalHb, Z ] = NewSolution(
Target, Schedule, Stock, Tank, Day, TransferLimit, tb1, tb2, Tx)

Z=1; %Preallocate Z value for looping

while Z==1 %New solution generation

    %Preallocate support variables
    HeavyStock=zeros (Day, Tank) ;
    MediumStock=zeros (Day, Tank) ;
    LightStock=zeros (Day, Tank) ;
    BYUStock=zeros (Day, Tank) ;
    TotalStock=zeros (Day, Tank) ;

    BYUTransferPercentage=zeros (Day, 1) ;
    BYUTrans=zeros (Day, Tank) ; %Daily amount of BYU transfer

    HeavyPercentage=zeros (Day, Tank) ;
    MediumPercentage=zeros (Day, Tank) ;
    LightPercentage=zeros (Day, Tank) ;
    BYUPercentage=zeros (Day, Tank) ;

    totalSchedule=zeros (Day, 1) ;

    TotalH=0;
    TotalM=0;
    TotalL=0;
    TotalHb=0;

    %Preallocate transfer-related variables
    AvT=zeros (Day, Tank) ; %Transfer availability for each tank
    AT=zeros (Day, 1) ; %Total available tanks for transfer each day

    CT=zeros (Tank, Tank, Day) ; %Transfer combination availability
    CTs=zeros (Day, 36) ; %Transform transfer combination array

    T1=zeros (Day, 1) ; %Selected transfer tank number 1
    T2=zeros (Day, 1) ; %Selected transfer tank number 2
    dt=zeros (Day, Tank) ; %Available stock for transfer in each tank

    BYR=zeros (Day, Tank) ; %Rating of BYU percentage

    left=zeros (Tank, Tank, Day) ; %Day left of transfer combination
    sw=zeros (Tank, Tank, Day) ; %Switch between lower stock and BYU
    tvl=zeros (Tank, Tank, Day) ; %Lower transfer value of combination
    ld=zeros (Tank, Tank, Day) ; %Switch for last day transfer value
    tvll=zeros (Tank, Tank, Day) ; %Lower transfer value of last day

    rt=zeros (Day, 36) ; %Randomly permuted transfer tank choice
    idT=zeros (Day, 36) ; %Index for tank available for transfer
```

```

%Preallocate load-related variables
AvL=zeros(Day,Tank); %Load availability for each tank
AL=zeros(Day,1); %Total available tanks for load on each day

SL=zeros(Day,Tank); %Single load capability of each tank

CL=zeros(Tank,Tank,Day); %Load combination availability
CLs=zeros(Day,36); %Transform load combination array

L1=zeros(Day,1); %Selected load tank number 1
L2=zeros(Day,1); %Selected load tank number 2
dl=zeros(Day,Tank); %Available space for loading in each tank

ral=zeros(Day,36); %Randomly permuted load tank choice
idL=zeros(Day,36); %Index for tank available for loading
RH=zeros(Day,600); %Randomly permuted load value for heavy
RM=zeros(Day,600); %Randomly permuted load value for medium
RL=zeros(Day,600); %Randomly permuted load value for light
RHb=zeros(Day,600); %Randomly permuted load value for BYU

%Preallocate solution set
TransferSet=zeros(Day,Tank);
HLoadSet=zeros(Day,Tank);
MLoadSet=zeros(Day,Tank);
LLoadSet=zeros(Day,Tank);
HbLoadSet=zeros(Day,Tank);

T=Tx*100; %Set initial temperature for inner SA

%Initial Stock
for j=1:Tank
    HeavyStock(1,j)=Stock(1,j);
    MediumStock(1,j)=Stock(2,j);
    LightStock(1,j)=Stock(3,j);
    BYUStock(1,j)=Stock(4,j);
end

%Precalculate day 1 total stock
TotalStock(1,:) = HeavyStock(1,:)+MediumStock(1,:)
    +LightStock(1,:)+BYUStock(1,:);

%Determine available tanks for initial transfer
for j=1:Tank
    if TotalStock(1,j) >= 125
        AvT(1,j)=1;
    end
end

%Calculate initial net stock of each tanks
for j=1:Tank
    dt(1,j)=floor((TotalStock(1,j)-120)/5);
end

%Calculate transfer combination availability

```

```

for a=1:Tank
    for b=1:Tank
        if a==b
            CT(a,b,1)=0;
        else
            if dt(1,a) + dt(1,b) >= 38
                CT(a,b,1)=AvT(1,a)*AvT(1,b);
            else
                CT(a,b,1)=0;
            end
        end
    end
end

%Calculate initial BYU percentage
for j=1:Tank
    BYUPercentage(1,j)=BYUStock(1,j)/TotalStock(1,j);
end

%Update combination availability based on BYU percentage
for a=1:Tank
    for b=1:Tank
        if BYUPercentage(1,a)>0.25 && BYUPercentage(1,b)>0.25
            CT(a,b,1)=0;
        end
    end
end

%Rate each tank from initial BYU percentage
for j=1:Tank
    if BYUPercentage(1,j) < 0.17
        BYR(1,j)=1;
    elseif BYUPercentage(1,j) > 0.25
        BYR(1,j)=3;
    else
        BYR(1,j)=2;
    end
end

%Check possible final day composition of each tank combination
for a=1:Tank
    for b=1:Tank
        if BYUPercentage(1,a) <= BYUPercentage(1,b)
            low=a;
            high=b;
        else
            low=b;
            high=a;
        end
        if dt(1,low) + dt(1,high) >= 76 %Set boundary
            if BYR(1,low) == BYR(1,high)
                if dt(1,low) >= 38 && dt(1,high) >= 38
                    cs=37;
                    cf=0;
                    ds=1;
                    df=38;
                elseif dt(1,low) < 38 && dt(1,high) >= 38

```



```

        cs=dt(1,low);
        cf=0;
        ds=(38-dt(1,low));
        df=38;
elseif dt(1,low) >= 38 && dt(1,high) < 38
    cs=37;
    cf=(37-dt(1,high));
    ds=1;
    df=dt(1,high)+1;
end
else
    if BYR(1,low) + BYR(1,high) == 3
        if dt(1,low) >= 38 && dt(1,high) >= 38
            llim=ceil(((0.17-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            cs=(38-llim);
            cf=0;
            ds=llim;
            df=38;
        elseif dt(1,low) < 38 && dt(1,high) >= 38
            llim=ceil(((0.17-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            if dt(1,low) <= (38-llim)
                cs=dt(1,low);
                ds=(38-dt(1,low));
            else
                cs=(38-llim);
                ds=llim;
            end
            cf=0;
            df=38;
        elseif dt(1,low) >= 38 && dt(1,high) < 38
            llim=ceil(((0.17-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            cs=(38-llim);
            cf=(37-dt(1,high));
            ds=llim;
            df=dt(1,high)+1;
        end
    elseif BYR(1,low) + BYR(1,high) == 4
        if dt(1,low) >= 38 && dt(1,high) >= 38
            llim=ceil(((0.17-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            ulim=floor(((0.25-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            cs=(38-llim);
            cf=(37-ulim);
            ds=llim;
            df=ulim+1;
        elseif dt(1,low) < 38 && dt(1,high) >= 38
            llim=ceil(((0.17-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            ulim=floor(((0.25-BYUPercentage(1,low))
/ (BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            if dt(1,low) < (38-llim)
                cs=dt(1,low);
                ds=(38-dt(1,low));
            else

```

```

        cs=(38-llim);
        ds=llim;
    end
    cf=(37-ulim);
    df=ulim+1;
    elseif dt(1,low) >= 38 && dt(1,high) < 38
        llim=ceil(((0.17-BYUPercentage(1,low))
)/(BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
        ulim=floor(((0.25-BYUPercentage(1,low)
)/(BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
        if dt(1,high) <= ulim
            cf=(37-dt(1,high));
            df=dt(1,high)+1;
        else
            cf=(37-ulim);
            df=ulim+1;
        end
        cs=(38-llim);
        ds=llim;
    end
    elseif BYR(1,low) + BYR(1,high) == 5
        if dt(1,low) >= 38 && dt(1,high) >= 38
            ulim=floor(((0.25-BYUPercentage(1,low)
)/(BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            cs=37;
            cf=(37-ulim);
            ds=1;
            df=ulim+1;
        elseif dt(1,low) < 38 && dt(1,high) >= 38
            ulim=floor(((0.25-BYUPercentage(1,low)
)/(BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            cs=dt(1,low);
            cf=(37-ulim);
            ds=(38-dt(1,low));
            df=ulim+1;
        elseif dt(1,low) >= 38 && dt(1,high) < 38
            ulim=floor(((0.25-BYUPercentage(1,low)
)/(BYUPercentage(1,high)-BYUPercentage(1,low)))*38);
            if dt(1,high) <= ulim
                cf=(37-dt(1,high));
                df=dt(1,high)+1;
            else
                cf=(37-ulim);
                df=ulim+1;
            end
            cs=37;
            ds=1;
        end
    end
    end
    sim=1; %Do the following 'while' calculation
else %Directly test the BYU transfer percentage result
    if dt(1,a) < 38 && dt(1,b) < 38
        if dt(1,a) <= dt(1,b)
            test=((BYUPercentage(1,a)*dt(1,a)) +
(BYUPercentage(1,b)*(38-dt(1,a))))/38;
        else

```

```

        test=((BYUPercentage(1,b)*dt(1,b)) +
(BYUPercentage(1,a)*(38-dt(1,b))))/38;
    end
    elseif dt(1,a) < 38 && dt(1,b) >= 38
        test=((BYUPercentage(1,a)*dt(1,a)) +
(BYUPercentage(1,b)*(38-dt(1,a))))/38;
    elseif dt(1,a) >= 38 && dt(1,b) < 38
        test=((BYUPercentage(1,b)*dt(1,b)) +
(BYUPercentage(1,a)*(38-dt(1,b))))/38;
    end
    if test > 0.25
        CT(a,b,1)=0;
    else
        left(a,b,1)=1;
        sw(a,b,1)=1;
        if dt(1,a) <= dt(1,b)
            tvl(a,b,1)=dt(1,a);
        else
            tvl(a,b,1)=dt(1,b);
        end
        ld(a,b,1)=0;
    end
    sim=0; %Skip the following 'while' calculation
    cs=0;
    ds=0;
    cf=0;
    df=0;
end
check=0;
c=cs;
d=ds;
while check == 0 && c > cf && d < df && sim == 1
    dayleft=floor(dt(1,low)/c);
    if floor(dt(1,high)/d) >= dayleft
        lastday=mod(dt(1,low),c);
        if lastday == 0
            check=1;
            left(a,b,1)=dayleft;
            sw(a,b,1)=2;
            tvl(a,b,1)=c;
            ld(a,b,1)=0;
        else
            if dt(1,high) - (dayleft*d) < lastday
                if lastday >= 38 -
                    (dt(1,high) - (dayleft*d))
                    test= ((BYUPercentage(1,high)
                        *(dt(1,high) - (dayleft*d)))
                    +(BYUPercentage(1,low)
                        *(38-(dt(1,high) -
                            (dayleft*d)))))/38;
                    if test <= 0.25
                        check=1;
                        left(a,b,1)=dayleft;
                        sw(a,b,1)=2;
                        tvl(a,b,1)=c;
                        ld(a,b,1)=1;
                        tvll(a,b,1)=dt(1,high)
                            - (dayleft*d);
                    end
                end
            end
        end
    end
end

```

```

        end
    end
else
    if dt(1,high) - (dayleft*d)
        >= (38-lastday)
        test=((BYUPercentage(1,low)
            *lastday) +
            (BYUPercentage(1,high)
            *(38-lastday)))/38;
        if test <= 0.25
            check=1;
            left(a,b,1)=dayleft;
            sw(a,b,1)=2;
            tvl(a,b,1)=c;
            ld(a,b,1)=1;
            tvll(a,b,1)=lastday;
        end
    end
end
end
end
end
    c=c-1;
    d=d+1;
end
if sim == 1
    CT(a,b,1)=CT(a,b,1)*check;
end
end
end

%Randomly choose tank for transfer from available choice
AT(1)=sum(sum(CT(:, :, 1), 2));
CTs(1, :)=reshape(CT(:, :, 1), 1, 36);
[~, idT(1, 1:36)]=sort(CTs(1, :), 'descend');
rt(1, 1:AT(1))=randperm(AT(1));
if mod(idT(1, rt(1, 1)), 6) == 0
    T1(1)=idT(1, rt(1, 1))/6;
    T2(1)=6;
else
    T1(1)=ceil(idT(1, rt(1, 1))/6);
    T2(1)=mod(idT(1, rt(1, 1)), 6);
end

%Determine available tanks for initial load
for j=1:Tank
    if TotalStock(1, j) <= 495
        AvL(1, j)=1;
    end
end

%Compare initial transfer tanks with current best solution
comp0=sort([tb1 tb2]);
comp1=sort([T1(1) T2(1)]);
if abs(comp1(1)-comp0(1))+abs(comp1(2)-comp0(2))>0
    deltaf=abs(comp1(1)-comp0(1))+abs(comp1(2)-comp0(2));
    PE=exp(-deltaf/T); %boltzmann probability
    if rand>PE

```

```

        T1(1)=tb1;
        T2(1)=tb2;
    end
end

%Store initial transfer tanks choice for future comparison
tx1=T1(1);
tx2=T2(1);

%Calculate total scheduled load per day based on shipment
schedule
for i=1:Day
    totalSchedule(i)=sum(Schedule(:,i));
end

%-----Main calculation-----
for i=1:Day
    for j=1:Tank
        %Refresh this day solution set
        TransferSet(i,j)=0;
        HLoadSet(i,j)=0;
        MLoadSet(i,j)=0;
        LLoadSet(i,j)=0;
        HbLoadSet(i,j)=0;
        %Check transfer and load availability
        if i>1
            if TotalStock(i,j) >= 125
                AvT(i,j)=1;
            end
            if TotalStock(i,j) <= 794
                AvL(i,j)=1;
            end
            %--Constraint (3.29)--
            if (HLoadSet(i-1,j)+MLoadSet(i-1,j)+LLoadSet(i-
1,j)+HbLoadSet(i-1,j)) > 0
                AvT(i,j)=0;
            end
            %--Constraint (3.30)--
            if T1(i-1)==j
                if TotalStock(i,j) - (HLoadSet(i-
1,j)+MLoadSet(i-1,j)+LLoadSet(i-1,j)+HbLoadSet(i-1,j)) >= 130
                    AvT(i,j)=1;
                    T1(i)=j;
                else
                    AvT(i,j)=0;
                    T1(i)=0;
                end
            end
            if T2(i-1)==j
                if TotalStock(i,j) - (HLoadSet(i-
1,j)+MLoadSet(i-1,j)+LLoadSet(i-1,j)+HbLoadSet(i-1,j)) >= 130
                    AvT(i,j)=1;
                    T2(i)=j;
                else
                    AvT(i,j)=0;
                    T2(i)=0;
                end
            end
        end
    end
end

```

```

        end
    end

    %Calculate stock percentages
    HeavyPercentage(i,j)=HeavyStock(i,j)/TotalStock(i,j);
    MediumPercentage(i,j)=
    MediumStock(i,j)/TotalStock(i,j);
    LightPercentage(i,j)=LightStock(i,j)/TotalStock(i,j);
    BYUPercentage(i,j)=BYUStock(i,j)/TotalStock(i,j);

    %Calculate stock available for transfer
    dt(i,j)=floor((TotalStock(i,j)-120)/5);
end

%Calculate transfer combination availability
for a=1:Tank
    for b=1:Tank
        if a==b
            CT(a,b,i)=0;
        else
            if dt(i,a) + dt(i,b) >= 38
                CT(a,b,i)=AvT(i,a)*AvT(i,b);
            else
                CT(a,b,i)=0;
            end
        end
    end
end

%Update transfer combination availability
if T1(i)~=0 && T2(i)==0
    for a=1:Tank
        for b=1:Tank
            if a ~= T1(i) && b ~= T1(i)
                CT(a,b,i)=0;
            end
        end
    end
elseif T1(i)==0 && T2(i)~=0
    for a=1:Tank
        for b=1:Tank
            if a ~= T2(i) && b ~= T2(i)
                CT(a,b,i)=0;
            end
        end
    end
elseif T1(i)~=0 && T2(i)~=0
    for a=1:Tank
        for b=1:Tank
            if a ~= T1(i) || b ~= T2(i)
                CT(a,b,i)=0;
            end
        end
    end
end
end

```

```

        %Update combination availability based on BYU percentage
        for a=1:Tank
            for b=1:Tank
                if BYUPercentage(i,a)>0.25 &&
BYUPercentage(i,b)>0.25
                    CT(a,b,i)=0;
                end
            end
        end
        %Update transfer combination availability based on
calculation
        for a=1:Tank
            for b=1:Tank
                if BYUPercentage(i,a) <= BYUPercentage(i,b)
                    low=a;
                    high=b;
                else
                    low=b;
                    high=a;
                end
                if dt(i,low) + dt(i,high) >= 76 %Set boundary
                    if BYR(i,low) == BYR(i,high)
                        if dt(i,low) >= 38 && dt(i,high) >= 38
                            cs=37;
                            cf=0;
                            ds=1;
                            df=38;
                        elseif dt(i,low) < 38 && dt(i,high) >= 38
                            cs=dt(i,low);
                            cf=0;
                            ds=(38-dt(i,low));
                            df=38;
                        elseif dt(i,low) >= 38 && dt(i,high) < 38
                            cs=37;
                            cf=(37-dt(i,high));
                            ds=1;
                            df=dt(i,high)+1;
                        end
                    else
                        if BYR(i,low) + BYR(i,high) == 3
                            if dt(i,low) >= 38 && dt(i,high) >= 38
                                llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
                                cs=(38-llim);
                                cf=0;
                                ds=llim;
                                df=38;
                            elseif dt(i,low) < 38 && dt(i,high) >=
38
                                llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
                                if dt(i,low) <= (38-llim)
                                    cs=dt(i,low);
                                    ds=(38-dt(i,low));
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

else
    cs=(38-llim);
    ds=llim;
end
cf=0;
df=38;
elseif dt(i,low) >= 38 && dt(i,high) <
38
    llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
    cs=(38-llim);
    cf=(37-dt(i,high));
    ds=llim;
    df=dt(i,high)+1;
end
elseif BYR(i,low) + BYR(i,high) == 4
    if dt(i,low) >= 38 && dt(i,high) >= 38
        llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        cs=(38-llim);
        cf=(37-ulim);
        ds=llim;
        df=ulim+1;
    elseif dt(i,low) < 38 && dt(i,high) >=
38
        llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        if dt(i,low) < (38-llim)
            cs=dt(i,low);
            ds=(38-dt(i,low));
        else
            cs=(38-llim);
            ds=llim;
        end
        cf=(37-ulim);
        df=ulim+1;
    elseif dt(i,low) >= 38 && dt(i,high) <
38
        llim=ceil(((0.17-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        if dt(i,high) <= ulim
            cf=(37-dt(i,high));
            df=dt(i,high)+1;
        else
            cf=(37-ulim);

```



```

        df=ulim+1;
    end
    cs=(38-llim);
    ds=llim;
end
elseif BYR(i,low) + BYR(i,high) == 5
    if dt(i,low) >= 38 && dt(i,high) >= 38
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        cs=37;
        cf=(37-ulim);
        ds=1;
        df=ulim+1;
    elseif dt(i,low) < 38 && dt(i,high) >=
38
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        cs=dt(i,low);
        cf=(37-ulim);
        ds=(38-dt(i,low));
        df=ulim+1;
    elseif dt(i,low) >= 38 && dt(i,high) <
38
        ulim=floor(((0.25-
BYUPercentage(i,low))/(BYUPercentage(i,high)-
BYUPercentage(i,low)))*38);
        if dt(i,high) <= ulim
            cf=(37-dt(i,high));
            df=dt(i,high)+1;
        else
            cf=(37-ulim);
            df=ulim+1;
        end
        cs=37;
        ds=1;
    end
end
end
sim=1; %Do the following 'while' calculation
else %Directly test the BYU transfer percentage
    if dt(i,a) < 38 && dt(i,b) < 38
        if dt(i,a) <= dt(i,b)
            test=((BYUPercentage(i,a)*dt(i,a)) +
(BYUPercentage(i,b)*(38-dt(i,a))))/38;
        else
            test=((BYUPercentage(i,b)*dt(i,b)) +
(BYUPercentage(i,a)*(38-dt(i,b))))/38;
        end
    elseif dt(i,a) < 38 && dt(i,b) >= 38
        test=((BYUPercentage(i,a)*dt(i,a)) +
(BYUPercentage(i,b)*(38-dt(i,a))))/38;
    elseif dt(i,a) >= 38 && dt(i,b) < 38
        test=((BYUPercentage(i,b)*dt(i,b)) +
(BYUPercentage(i,a)*(38-dt(i,b))))/38;
    end
    if test > 0.25

```

```

        CT(a,b,i)=0;
    else
        left(a,b,i)=1;
        sw(a,b,i)=1;
        if dt(i,a) <= dt(i,b)
            tvl(a,b,i)=dt(i,a);
        else
            tvl(a,b,i)=dt(i,b);
        end
        ld(a,b,i)=0;
    end
    sim=0; %Skip the following 'while' calculation
    cs=0;
    ds=0;
    cf=0;
    df=0;
end
check=0;
c=cs;
d=ds;
while check == 0 && c > cf && d < df && sim == 1
    dayleft=floor(dt(i,low)/c);
    if floor(dt(i,high)/d) >= dayleft
        lastday=mod(dt(i,low),c);
        if lastday == 0
            check=1;
            left(a,b,i)=dayleft;
            sw(a,b,i)=2;
            tvl(a,b,i)=c;
            ld(a,b,i)=0;
        else
            if dt(i,high) - (dayleft*d) < lastday
                if lastday >= 38 - (dt(i,high) -
(dayleft*d))
                    test=((BYUPercentage(i,high) * (dt(i,high) -
(dayleft*d)))+(BYUPercentage(i,low) * (38-(dt(i,high) -
(dayleft*d)))))/38;
                    if test <= 0.25
                        check=1;
                        left(a,b,i)=dayleft;
                        sw(a,b,i)=2;
                        tvl(a,b,i)=c;
                        ld(a,b,i)=1;
                        tvll(a,b,i)=dt(i,high) -
(dayleft*d);
                    end
                end
            else
                if dt(i,high) - (dayleft*d) >=
(38-lastday)
                    test=((BYUPercentage(i,low) * lastday) + (BYUPercentage(i,high) * (38-
lastday)))/38;
                    if test <= 0.25
                        check=1;
                        left(a,b,i)=dayleft;
                        sw(a,b,i)=2;

```

```

                                                    tvl(a,b,i)=c;
                                                    ld(a,b,i)=1;
                                                    tvll(a,b,i)=lastday;
                                                end
                                            end
                                        end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    c=c-1;
    d=d+1;
end
if sim == 1
    CT(a,b,i)=CT(a,b,i)*check;
end
end
end
%Change transfer tanks if needed
AT(i)=sum(sum(CT(:,:,i),2));
if AT(i) == 0
    break; %In case no possible transfer combination
end
CTs(i,:)=reshape(CT(:,:,i),1,36);
[~,idT(i,1:36)]=sort(CTs(i,:), 'descend');
rt(i,1:AT(i))=randperm(AT(i));
if mod(idT(i,rt(i,1)),6) == 0
    T1(i)=idT(i,rt(i,1))/6;
    T2(i)=6;
else
    T1(i)=ceil(idT(i,rt(i,1))/6);
    T2(i)=mod(idT(i,rt(i,1)),6);
end
%Determine transfer value based on the final check
if left(T1(i),T2(i),i) > 0
    if sw(T1(i),T2(i),i) == 1
        if dt(i,T1(i)) <= dt(i,T2(i))
TransferSet(i,T1(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
                TransferSet(i,T2(i))=TransferLimit*(38-
tvl(T1(i),T2(i),i))/38;
                else
TransferSet(i,T2(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
                TransferSet(i,T1(i))=TransferLimit*(38-
tvl(T1(i),T2(i),i))/38;
                end
                elseif sw(T1(i),T2(i),i) == 2
                    if BYUPercentage(i,T1(i)) <=
BYUPercentage(i,T2(i))
TransferSet(i,T1(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
                TransferSet(i,T2(i))=TransferLimit*(38-
tvl(T1(i),T2(i),i))/38;
                else
TransferSet(i,T2(i))=TransferLimit*tvl(T1(i),T2(i),i)/38;
                TransferSet(i,T1(i))=TransferLimit*(38-
tvl(T1(i),T2(i),i))/38;
                end

```

```

        end
    elseif left(T1(i),T2(i),i) == 0
        if ld(T1(i),T2(i),i) == 1
            if BYUPercentage(i,T1(i)) <=
BYUPercentage(i,T2(i))

TransferSet(i,T1(i))=TransferLimit*tvll(T1(i),T2(i),i)/38;
                TransferSet(i,T2(i))=TransferLimit*(38-
tvll(T1(i),T2(i),i))/38;
                    else

TransferSet(i,T2(i))=TransferLimit*tvll(T1(i),T2(i),i)/38;
                TransferSet(i,T1(i))=TransferLimit*(38-
tvll(T1(i),T2(i),i))/38;
                    end
                end
            end
        end
        %Continue check load availability
        if (TotalStock(i,T1(i))-TransferSet(i,T1(i))) > 130
            AvL(i,T1(i))=0;
        end
        if (TotalStock(i,T2(i))-TransferSet(i,T2(i))) > 130
            AvL(i,T2(i))=0;
        end
        end
        %Reset random variables
        ral(i,:)=zeros;
        idL(i,:)=zeros;
        RH(i,:)=zeros;
        RM(i,:)=zeros;
        RL(i,:)=zeros;
        RHb(i,:)=zeros;
        L1(i)=zeros;
        L2(i)=zeros;
        %Calculate space available for load
        for j=1:Tank
            dl(i,j)=floor(795 - TotalStock(i,j) +
TransferSet(i,j));
        end
        %Check if any tank is capable of single loading
        for j=1:Tank
            if dl(i,j) >= totalSchedule(i)
                SL(i,j)=AvL(i,j);
            else
                SL(i,j)=0;
            end
        end
        end
        %Randomly choose tank for load from available choice
        if sum(SL(i,:)) == 0
            %Check combination load availability
            for a=1:Tank
                for b=1:Tank
                    if a == b
                        CL(a,b,i)=0;
                    else
                        if dl(i,a) + dl(i,b) >= totalSchedule(i)
                            CL(a,b,i)=AvL(i,a)*AvL(i,b);
                        else
                            CL(a,b,i)=0;
                        end
                    end
                end
            end
        end
    end
end

```

```

        end
    end
end
end
end
%Determine tank load combination
AL(i)=sum(sum(CL(:, :, i), 2));
if AL(i) == 0
    break; %In case no possible load combination
end
CLs(i, :)=reshape(CL(:, :, i), 1, 36);
[~, idL(i, 1:36)]=sort(CLs(i, :), 'descend');
ral(i, 1:AL(i))=randperm(AL(i));
if mod(idL(i, ral(i, 1)), 6) == 0
    L1(i)=idL(i, ral(i, 1))/6;
    L2(i)=6;
else
    L1(i)=ceil(idL(i, ral(i, 1))/6);
    L2(i)=mod(idL(i, ral(i, 1)), 6);
end
else
%Determine single tank load choice
AL(i)=sum(SL(i, :));
[~, idL(i, 1:6)]=sort(SL(i, :), 'descend');
ral(i, 1:AL(i))=randperm(AL(i));
L1(i)=idL(i, ral(i, 1));
end
%Randomly determine load value within boundary
if L2(i) == 0
    %Load value selection for 1 tanks
    HLoadSet(i, L1(i))=Schedule(1, i);
    MLoadSet(i, L1(i))=Schedule(2, i);
    LLoadSet(i, L1(i))=Schedule(3, i);
    HbLoadSet(i, L1(i))=Schedule(4, i);
elseif L2(i) ~= 0
    %Load value selection for 2 tanks
    if Schedule(1, i) ~= 0
        if dl(i, L1(i)) >= Schedule(1, i) && dl(i, L2(i)) >=
Schedule(1, i)
            RH(i, 1:Schedule(1, i))=randperm(Schedule(1, i));
            HLoadSet(i, L1(i))=RH(i, 1);
            HLoadSet(i, L2(i))=Schedule(1, i)-RH(i, 1);
        elseif dl(i, L1(i)) < Schedule(1, i) && dl(i, L2(i))
>= Schedule(1, i)
            RH(i, 1:dl(i, L1(i)))=randperm(dl(i, L1(i)));
            HLoadSet(i, L1(i))=RH(i, 1);
            HLoadSet(i, L2(i))=Schedule(1, i)-RH(i, 1);
        elseif dl(i, L1(i)) >= Schedule(1, i) && dl(i, L2(i))
< Schedule(1, i)
            RH(i, 1:dl(i, L2(i)))=randperm(dl(i, L2(i)));
            HLoadSet(i, L2(i))=RH(i, 1);
            HLoadSet(i, L1(i))=Schedule(1, i)-RH(i, 1);
        elseif dl(i, L1(i)) < Schedule(1, i) && dl(i, L2(i))
< Schedule(1, i)
            if dl(i, L1(i)) < dl(i, L2(i))
                HLoadSet(i, L1(i))=dl(i, L1(i));
                HLoadSet(i, L2(i))=Schedule(1, i)-
dl(i, L1(i));
            elseif dl(i, L1(i)) >= dl(i, L2(i))

```

```

                                HLoadSet(i,L2(i))=dl(i,L2(i));
                                HLoadSet(i,L1(i))=Schedule(1,i)-
dl(i,L2(i));
                                end
                                end
                                dl(i,L1(i))=dl(i,L1(i))-HLoadSet(i,L1(i));
                                dl(i,L2(i))=dl(i,L2(i))-HLoadSet(i,L2(i));
                                end
                                if Schedule(2,i) ~=0
                                if dl(i,L1(i)) >= Schedule(2,i) && dl(i,L2(i)) >=
Schedule(2,i)
                                RM(i,1:Schedule(2,i))=randperm(Schedule(2,i));
                                MLoadSet(i,L1(i))=RM(i,1);
                                MLoadSet(i,L2(i))=Schedule(2,i)-RM(i,1);
                                elseif dl(i,L1(i)) < Schedule(2,i) && dl(i,L2(i))
>= Schedule(2,i)
                                RM(i,1:dl(i,L1(i)))=randperm(dl(i,L1(i)));
                                MLoadSet(i,L1(i))=RM(i,1);
                                MLoadSet(i,L2(i))=Schedule(2,i)-RM(i,1);
                                elseif dl(i,L1(i)) >= Schedule(2,i) && dl(i,L2(i))
< Schedule(2,i)
                                RM(i,1:dl(i,L2(i)))=randperm(dl(i,L2(i)));
                                MLoadSet(i,L2(i))=RM(i,1);
                                MLoadSet(i,L1(i))=Schedule(2,i)-RM(i,1);
                                elseif dl(i,L1(i)) < Schedule(2,i) && dl(i,L2(i))
< Schedule(2,i)
                                if dl(i,L1(i)) < dl(i,L2(i))
                                MLoadSet(i,L1(i))=dl(i,L1(i));
                                MLoadSet(i,L2(i))=Schedule(2,i)-
dl(i,L1(i));
                                elseif dl(i,L1(i)) >= dl(i,L2(i))
                                MLoadSet(i,L2(i))=dl(i,L2(i));
                                MLoadSet(i,L1(i))=Schedule(2,i)-
dl(i,L2(i));
                                end
                                end
                                end
                                if Schedule(3,i) ~=0
                                if dl(i,L1(i)) >= Schedule(3,i) && dl(i,L2(i)) >=
Schedule(3,i)
                                RL(i,1:Schedule(3,i))=randperm(Schedule(3,i));
                                LLoadSet(i,L1(i))=RL(i,1);
                                LLoadSet(i,L2(i))=Schedule(3,i)-RL(i,1);
                                elseif dl(i,L1(i)) < Schedule(3,i) && dl(i,L2(i))
>= Schedule(3,i)
                                RL(i,1:dl(i,L1(i)))=randperm(dl(i,L1(i)));
                                LLoadSet(i,L1(i))=RL(i,1);
                                LLoadSet(i,L2(i))=Schedule(3,i)-RL(i,1);
                                elseif dl(i,L1(i)) >= Schedule(3,i) && dl(i,L2(i))
< Schedule(3,i)
                                RL(i,1:dl(i,L2(i)))=randperm(dl(i,L2(i)));
                                LLoadSet(i,L2(i))=RL(i,1);
                                LLoadSet(i,L1(i))=Schedule(3,i)-RL(i,1);
                                elseif dl(i,L1(i)) < Schedule(3,i) && dl(i,L2(i))
< Schedule(3,i)
                                if dl(i,L1(i)) < dl(i,L2(i))

```

```

                                LLoadSet (i, L1 (i))=dl (i, L1 (i));
                                LLoadSet (i, L2 (i))=Schedule (3, i) -
dl (i, L1 (i));
                                elseif dl (i, L1 (i)) >= dl (i, L2 (i))
                                LLoadSet (i, L2 (i))=dl (i, L2 (i));
                                LLoadSet (i, L1 (i))=Schedule (3, i) -
dl (i, L2 (i));
                                end
                                end
                                dl (i, L1 (i))=dl (i, L1 (i)) - LLoadSet (i, L1 (i));
                                dl (i, L2 (i))=dl (i, L2 (i)) - LLoadSet (i, L2 (i));
                                end
                                if Schedule (4, i) ~=0
                                if dl (i, L1 (i)) >= Schedule (4, i) && dl (i, L2 (i)) >=
Schedule (4, i)
RHb (i, 1:Schedule (4, i))=randperm (Schedule (4, i));
                                HbLoadSet (i, L1 (i))=RHb (i, 1);
                                HbLoadSet (i, L2 (i))=Schedule (4, i) - RHb (i, 1);
                                elseif dl (i, L1 (i)) < Schedule (4, i) && dl (i, L2 (i))
>= Schedule (4, i)
                                RHb (i, 1:dl (i, L1 (i)))=randperm (dl (i, L1 (i)));
                                HbLoadSet (i, L1 (i))=RHb (i, 1);
                                HbLoadSet (i, L2 (i))=Schedule (4, i) - RHb (i, 1);
                                elseif dl (i, L1 (i)) >= Schedule (4, i) && dl (i, L2 (i))
< Schedule (4, i)
                                RHb (i, 1:dl (i, L2 (i)))=randperm (dl (i, L2 (i)));
                                HbLoadSet (i, L2 (i))=RHb (i, 1);
                                HbLoadSet (i, L1 (i))=Schedule (4, i) - RHb (i, 1);
                                elseif dl (i, L1 (i)) < Schedule (4, i) && dl (i, L2 (i))
< Schedule (4, i)
                                if dl (i, L1 (i)) < dl (i, L2 (i))
                                HbLoadSet (i, L1 (i))=dl (i, L1 (i));
                                HbLoadSet (i, L2 (i))=Schedule (4, i) -
dl (i, L1 (i));
                                elseif dl (i, L1 (i)) >= dl (i, L2 (i))
                                HbLoadSet (i, L2 (i))=dl (i, L2 (i));
                                HbLoadSet (i, L1 (i))=Schedule (4, i) -
dl (i, L2 (i));
                                end
                                end
                                end
                                end
                                %Calculate next day stock
                                if i<Day
                                for j=1:Tank
                                HeavyStock (i+1, j)=HeavyStock (i, j) -
(HeavyPercentage (i, j)*TransferSet (i, j))+HLoadSet (i, j);
                                MediumStock (i+1, j)=MediumStock (i, j) -
(MediumPercentage (i, j)*TransferSet (i, j))+MLoadSet (i, j);
                                LightStock (i+1, j)=LightStock (i, j) -
(LightPercentage (i, j)*TransferSet (i, j))+LLoadSet (i, j);
                                BYUStock (i+1, j)=BYUStock (i, j) -
(BYUPercentage (i, j)*TransferSet (i, j))+HbLoadSet (i, j);
                                TotalStock (i+1, j) =
HeavyStock (i+1, j)+MediumStock (i+1, j)+LightStock (i+1, j)+BYUStock (i+
1, j);
                                end

```

```

end
%Calculate BYU transfer percentage
for j=1:Tank
    BYUTrans(i,j)=BYUPercentage(i,j)*TransferSet(i,j);
end
BYUTransferPercentage(i)=
sum(BYUTrans(i,:))/sum(TransferSet(i,:));
end

%-----ending calculation-----

%Total transferred crude per class
for i=1:Day
    for j=1:Tank
        TotalH=TotalH+TransferSet(i,j)*HeavyPercentage(i,j);
        TotalM=TotalM+TransferSet(i,j)*MediumPercentage(i,j);
        TotalL=TotalL+TransferSet(i,j)*LightPercentage(i,j);
        TotalHb=TotalHb+TransferSet(i,j)*BYUPercentage(i,j);
    end
end
%Total H/M/L Percentage
H=(TotalH+TotalHb)/(TotalH+TotalM+TotalL+TotalHb);
M=(TotalM)/(TotalH+TotalM+TotalL+TotalHb);
L=(TotalL)/(TotalH+TotalM+TotalL+TotalHb);
%Objective Function
Z=abs(Target(1)-H)+abs(Target(2)-M)+abs(Target(3)-L);
%Check BYU Constraint
for i=1:Day
    if BYUTransferPercentage(i) > 0.25
        Z=1;
    end
end
%Check Capacity Constraint
for i=1:Day
    if TotalStock(i,1) > 795 || TotalStock(i,2) > 795 ||
TotalStock(i,3) > 795 || TotalStock(i,4) > 795 || TotalStock(i,5)
> 795 || TotalStock(i,6) > 795 || (TotalStock(i,1)-
TransferSet(i,1)) < 120 || (TotalStock(i,2)-TransferSet(i,2)) <
120 || (TotalStock(i,3)-TransferSet(i,3)) < 120 ||
(TotalStock(i,4)-TransferSet(i,4)) < 120 || (TotalStock(i,5)-
TransferSet(i,5)) < 120 || (TotalStock(i,6)-TransferSet(i,6)) <
120
        Z=1;
    end
end
%Set Z to 1 and TotalHb to 0 when break occurs
for i=1:Day
    if AT(i) == 0 || AL(i) == 0
        Z=1;
        TotalHb=0;
        break;
    end
end
end
end

```



## APPENDIX D

### EXPERIMENT RESULT

Here provided the full value of important decision variables (Transfer Value and Load Value) from all replications in each experiment done in chapter IV:

#### Experiment 1:

##### *Replication 1:*

Transfer Set:

	A	B	C	D	E	F
1	105	0	0	0	0	85
2	85	0	105	0	0	0
3	50	0	0	0	0	140
4	0	0	0	5	0	185
5	0	0	0	5	0	185
6	0	0	0	145	0	45
7	0	10	0	180	0	0
8	0	5	0	185	0	0
9	0	55	0	135	0	0
10	0	25	165	0	0	0
11	0	25	165	0	0	0
12	0	25	165	0	0	0
13	0	80	110	0	0	0
14	155	35	0	0	0	0
15	155	35	0	0	0	0
16	160	30	0	0	0	0
17	110	80	0	0	0	0
18	0	40	150	0	0	0
19	0	40	150	0	0	0
20	0	35	155	0	0	0
21	0	80	110	0	0	0
22	0	0	0	185	0	5
23	0	0	0	185	0	5
24	0	0	0	185	0	5
25	0	0	0	115	0	75
26	185	0	0	0	0	5
27	185	0	0	0	0	5
28	120	0	0	0	0	70
29	0	0	0	0	180	10
30	0	0	0	0	185	5

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	558
2	0	0	204	0	0	0
3	0	0	92	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	100	0	0
11	379	0	0	0	0	171
12	0	0	0	208	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	242	0	0	0
16	0	0	326	274	0	0
17	108	0	0	0	0	0
18	150	0	0	0	0	0
19	92	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	200	0	0	0	0
27	0	0	0	400	0	0
28	0	0	0	258	0	0
29	0	0	50	0	0	0
30	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	200	0	0	0
4	200	0	0	0	0	0
5	0	0	108	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	92
9	0	0	0	0	0	8
10	0	0	0	92	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	100	0	0	0	0	0
20	40	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	100	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	400
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	308	0	0	0
22	0	0	92	0	0	0
23	0	308	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	92	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

*Replication 2:*

Transfer Set:

	A	B	C	D	E	F
1	185	0	0	0	5	0
2	55	0	0	0	135	0
3	0	0	0	5	185	0
4	0	0	0	5	185	0
5	0	0	0	40	150	0
6	0	35	0	155	0	0
7	0	35	0	155	0	0
8	0	30	0	160	0	0
9	0	55	0	135	0	0
10	0	30	160	0	0	0
11	0	30	160	0	0	0
12	0	25	165	0	0	0
13	0	75	115	0	0	0
14	0	35	0	155	0	0
15	0	30	0	160	0	0
16	0	30	0	160	0	0
17	0	75	0	115	0	0
18	0	25	0	0	0	165
19	0	20	0	0	0	170
20	0	20	0	0	0	170
21	0	50	0	0	0	140
22	0	0	0	185	5	0
23	0	0	0	185	5	0
24	0	0	0	120	70	0
25	0	0	185	0	5	0
26	0	0	185	0	5	0
27	0	0	185	0	5	0
28	0	0	105	0	85	0
29	185	0	0	0	5	0
30	185	0	0	0	5	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	558
2	204	0	0	0	0	0
3	0	0	92	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	100	0	0
11	359	0	0	191	0	0
12	0	0	0	208	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	242	0	0	0
16	0	0	419	0	181	0
17	0	0	0	108	0	0
18	0	0	0	150	0	0
19	0	0	0	92	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	200	0	0	0	0
27	0	0	0	400	0	0
28	0	0	0	258	0	0
29	0	0	0	0	0	50
30	0	0	0	0	0	0



Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	200	0	0	0
4	0	0	200	0	0	0
5	108	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	92	0
9	0	0	0	0	0	8
10	0	0	0	92	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	100	0	0
20	0	0	0	40	0	0
21	0	0	0	0	0	0
22	0	100	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	400	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	308
22	0	92	0	0	0	0
23	0	0	0	0	0	308
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	92	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

*Replication 3:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	105	85	0	0
2	0	45	0	145	0	0
3	0	50	0	140	0	0
4	0	45	0	145	0	0
5	0	55	0	135	0	0
6	0	25	0	0	0	165
7	0	20	0	0	0	170
8	0	35	0	0	0	155
9	0	40	150	0	0	0
10	0	40	150	0	0	0
11	0	40	150	0	0	0
12	0	80	110	0	0	0
13	175	15	0	0	0	0
14	175	15	0	0	0	0
15	175	15	0	0	0	0
16	110	80	0	0	0	0
17	0	0	175	15	0	0
18	0	0	175	15	0	0
19	0	0	100	90	0	0
20	165	0	0	25	0	0
21	170	0	0	20	0	0
22	170	0	0	20	0	0
23	100	0	0	90	0	0
24	0	0	0	25	0	165
25	0	0	0	20	0	170
26	0	0	0	20	0	170
27	0	0	0	85	0	105
28	0	155	0	35	0	0
29	0	160	0	30	0	0
30	0	135	0	55	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	558	0	0	0
2	0	0	0	0	0	204
3	92	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	100
11	0	0	0	178	0	372
12	0	0	208	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	242	0	0	0
16	600	0	0	0	0	0
17	0	108	0	0	0	0
18	0	150	0	0	0	0
19	0	92	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	200	0	0	0	0	0
27	0	0	0	0	0	400
28	0	0	0	0	0	258
29	0	0	50	0	0	0
30	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	200	0	0	0	0	0
4	0	0	0	0	0	200
5	108	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	92	0	0
9	0	0	0	0	0	8
10	0	0	0	0	0	92
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	100	0	0	0	0
20	0	0	0	0	0	40
21	0	0	0	0	0	0
22	0	0	100	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	400	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	308	0	0	0
22	0	0	92	0	0	0
23	308	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	92	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

*Replication 4:*

Transfer Set:

	A	B	C	D	E	F
1	185	0	0	5	0	0
2	55	0	0	135	0	0
3	0	0	0	5	185	0
4	0	0	0	5	185	0
5	0	0	0	5	185	0
6	0	0	0	85	105	0
7	0	50	0	140	0	0
8	0	50	0	140	0	0
9	0	55	0	135	0	0
10	0	45	0	0	0	145
11	0	45	0	0	0	145
12	0	45	0	0	0	145
13	0	45	0	0	0	145
14	160	30	0	0	0	0
15	160	30	0	0	0	0
16	165	25	0	0	0	0
17	110	80	0	0	0	0
18	0	10	0	0	0	180
19	0	10	0	0	0	180
20	0	5	0	0	0	185
21	0	75	0	0	0	115
22	0	0	0	185	5	0
23	0	0	0	185	5	0
24	0	0	0	185	5	0
25	0	0	0	115	75	0
26	185	0	0	0	5	0
27	185	0	0	0	5	0
28	120	0	0	0	70	0
29	0	0	185	0	5	0
30	0	0	185	0	5	0



Heavy Load Value:

	A	B	C	D	E	F
1	0	0	558	0	0	0
2	0	0	0	0	0	204
3	0	0	0	0	0	92
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	100	0	0	0	0	0
11	0	0	0	550	0	0
12	84	0	0	124	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	242
16	0	0	0	0	182	418
17	108	0	0	0	0	0
18	150	0	0	0	0	0
19	92	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	200	0	0
27	0	0	0	228	0	172
28	258	0	0	0	0	0
29	0	0	0	50	0	0
30	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	200
4	200	0	0	0	0	0
5	108	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	92	0
9	8	0	0	0	0	0
10	92	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	100	0	0	0	0	0
20	40	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	100
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	400	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	308
22	0	0	0	0	0	92
23	0	308	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	92	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

*Replication 5:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	105	85	0	0
2	0	45	0	145	0	0
3	0	50	0	140	0	0
4	0	45	0	145	0	0
5	0	55	0	135	0	0
6	165	25	0	0	0	0
7	170	20	0	0	0	0
8	110	80	0	0	0	0
9	0	5	185	0	0	0
10	0	5	185	0	0	0
11	0	5	185	0	0	0
12	0	80	110	0	0	0
13	155	35	0	0	0	0
14	155	35	0	0	0	0
15	155	35	0	0	0	0
16	110	80	0	0	0	0
17	0	0	185	0	0	5
18	0	0	185	0	0	5
19	0	0	80	0	0	110
20	0	0	0	35	0	155
21	0	0	0	30	0	160
22	0	0	0	50	0	140
23	0	170	0	20	0	0
24	0	170	0	20	0	0
25	0	175	0	15	0	0
26	0	125	0	65	0	0
27	170	0	0	20	0	0
28	170	0	0	20	0	0
29	170	0	0	20	0	0
30	130	0	0	60	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	558	0	0	0
2	204	0	0	0	0	0
3	0	0	0	0	0	92
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	100	0	0	0	0	0
11	380	0	0	170	0	0
12	0	0	208	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	242	0	0	0
16	0	600	0	0	0	0
17	108	0	0	0	0	0
18	150	0	0	0	0	0
19	92	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	200	0	0	0	0
27	0	0	0	0	0	400
28	0	0	0	0	0	258
29	0	0	50	0	0	0
30	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	200
4	0	0	0	0	0	200
5	0	0	108	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	92	0	0
9	0	0	0	8	0	0
10	92	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	100	0	0	0	0	0
20	0	40	0	0	0	0
21	0	0	0	0	0	0
22	100	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0



BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	400	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	308	0	0	0
22	92	0	0	0	0	0
23	0	0	308	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	92	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

**Experiment 2:***Replication 1:*

Transfer Set:

	A	B	C	D	E	F
1	0	110	80	0	0	0
2	0	185	5	0	0	0
3	0	185	5	0	0	0
4	0	180	10	0	0	0
5	0	0	35	0	155	0
6	0	0	35	0	155	0
7	0	0	30	0	160	0
8	0	0	85	0	105	0
9	185	0	5	0	0	0
10	185	0	5	0	0	0
11	185	0	5	0	0	0
12	105	0	85	0	0	0
13	0	0	5	0	0	185
14	0	0	5	0	0	185
15	0	0	5	0	0	185
16	0	0	75	0	0	115
17	0	0	25	0	165	0
18	0	0	25	0	165	0
19	0	0	25	0	165	0
20	0	0	90	0	100	0
21	0	175	0	15	0	0
22	0	175	0	15	0	0
23	0	180	0	10	0	0
24	0	110	0	80	0	0
25	0	0	155	35	0	0
26	0	0	155	35	0	0
27	0	0	160	30	0	0
28	0	0	130	60	0	0
29	0	185	0	5	0	0
30	0	185	0	5	0	0
31	0	155	0	35	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	150	0	0	0	0
8	0	0	0	0	600	0
9	0	108	0	0	0	0
10	0	92	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	92	0
2	0	0	0	0	0	0
3	0	0	0	0	0	108
4	0	0	0	0	0	192
5	0	0	0	0	0	90
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	108	0	0	0	0	0
15	150	0	0	0	0	0
16	0	0	0	0	0	192
17	0	0	0	0	0	100
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	150	0
22	0	0	600	0	0	0
23	108	0	0	0	0	0
24	0	0	0	0	0	0
25	0	242	0	0	0	0
26	0	173	0	0	427	0
27	0	108	0	0	0	0
28	0	0	50	0	0	0
29	0	0	0	0	0	0
30	0	0	92	0	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	200	0
2	0	0	0	0	100	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	93
7	0	92	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	200	0	0	0	0
11	0	0	0	167	0	141
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	8

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	308
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	92	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	400	0	0	0
31	0	0	0	0	0	0

*Replication 2:*

Transfer Set:

	A	B	C	D	E	F
1	0	10	0	0	180	0
2	185	5	0	0	0	0
3	185	5	0	0	0	0
4	185	5	0	0	0	0
5	105	85	0	0	0	0
6	0	185	0	5	0	0
7	0	185	0	5	0	0
8	0	180	0	10	0	0
9	0	0	0	10	180	0
10	0	0	0	85	105	0
11	0	165	0	25	0	0
12	0	165	0	25	0	0
13	0	165	0	25	0	0
14	0	105	0	85	0	0
15	0	0	0	35	0	155
16	0	0	0	30	0	160
17	0	0	0	30	0	160
18	0	0	0	80	0	110
19	0	0	0	30	160	0
20	0	0	0	25	165	0
21	0	0	0	25	165	0
22	0	0	0	85	105	0
23	0	150	40	0	0	0
24	0	150	40	0	0	0
25	0	100	90	0	0	0
26	0	0	35	155	0	0
27	0	0	30	160	0	0
28	0	0	30	160	0	0
29	0	0	65	125	0	0
30	0	165	25	0	0	0
31	0	165	25	0	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	150
8	0	600	0	0	0	0
9	108	0	0	0	0	0
10	0	0	0	0	92	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0



Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	92
2	0	0	0	0	0	0
3	0	0	0	108	0	0
4	0	0	0	0	192	0
5	90	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	108	0	0	0	0
15	150	0	0	0	0	0
16	0	192	0	0	0	0
17	0	100	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	150
22	0	0	0	600	0	0
23	0	0	0	0	108	0
24	0	0	0	0	0	0
25	0	0	0	0	242	0
26	0	600	0	0	0	0
27	0	0	0	0	0	108
28	0	0	0	0	50	0
29	0	0	0	0	0	0
30	0	0	0	92	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	200
2	0	0	0	100	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	93	0
7	0	0	0	0	0	92
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	200	0
11	0	0	0	0	308	0
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	8	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	308
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	92
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	400	0	0
31	0	0	0	0	0	0

*Replication 3:*

Transfer Set:

	A	B	C	D	E	F
1	110	0	80	0	0	0
2	185	0	5	0	0	0
3	185	0	5	0	0	0
4	180	0	10	0	0	0
5	0	0	35	0	155	0
6	0	0	35	0	155	0
7	0	0	30	0	160	0
8	0	0	85	0	105	0
9	0	0	5	0	0	185
10	0	0	85	0	0	105
11	0	0	25	0	165	0
12	0	0	25	0	165	0
13	0	0	20	0	170	0
14	0	0	90	0	100	0
15	0	180	10	0	0	0
16	0	185	5	0	0	0
17	0	185	5	0	0	0
18	0	105	85	0	0	0
19	185	0	0	0	5	0
20	185	0	0	0	5	0
21	185	0	0	0	5	0
22	60	0	0	0	130	0
23	0	45	0	0	145	0
24	0	40	0	0	150	0
25	0	80	0	0	110	0
26	0	35	155	0	0	0
27	0	35	155	0	0	0
28	0	30	160	0	0	0
29	0	60	130	0	0	0
30	0	30	0	0	0	160
31	0	30	0	0	0	160

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	150
8	0	0	0	0	600	0
9	0	0	0	108	0	0
10	0	0	0	0	0	92
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	92	0
2	0	0	0	0	0	0
3	0	0	0	108	0	0
4	192	0	0	0	0	0
5	90	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	108	0
15	0	0	0	0	150	0
16	0	0	0	0	192	0
17	0	0	0	0	100	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	150	0	0	0	0
22	0	0	600	0	0	0
23	108	0	0	0	0	0
24	0	0	0	0	0	0
25	242	0	0	0	0	0
26	0	0	0	0	600	0
27	108	0	0	0	0	0
28	0	0	0	0	50	0
29	0	0	0	0	0	0
30	0	0	92	0	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	200	0
2	0	0	0	0	100	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	93	0	0	0	0	0
7	0	0	0	0	0	92
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	200
11	0	0	0	0	0	308
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	8	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	308	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	92	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	400	0	0	0
31	0	0	0	0	0	0



*Replication 4:*

Transfer Set:

	A	B	C	D	E	F
1	0	110	80	0	0	0
2	0	185	5	0	0	0
3	0	185	5	0	0	0
4	0	180	10	0	0	0
5	185	0	5	0	0	0
6	185	0	5	0	0	0
7	185	0	5	0	0	0
8	105	0	85	0	0	0
9	0	160	30	0	0	0
10	0	160	30	0	0	0
11	0	105	85	0	0	0
12	165	0	25	0	0	0
13	165	0	25	0	0	0
14	170	0	20	0	0	0
15	100	0	90	0	0	0
16	0	0	20	0	0	170
17	0	0	15	0	0	175
18	0	0	15	0	0	175
19	0	0	85	0	0	105
20	0	0	0	10	180	0
21	0	0	0	10	180	0
22	0	0	0	10	180	0
23	0	0	0	55	135	0
24	0	0	155	35	0	0
25	0	0	160	30	0	0
26	0	0	160	30	0	0
27	0	0	125	65	0	0
28	0	0	0	5	185	0
29	0	0	0	5	185	0
30	0	0	0	35	155	0
31	0	185	0	5	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	150	0	0	0	0
8	600	0	0	0	0	0
9	0	0	0	0	108	0
10	0	0	0	0	92	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	92
2	0	0	0	0	0	0
3	0	0	0	108	0	0
4	0	0	0	0	0	192
5	0	90	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	108	0	0
15	150	0	0	0	0	0
16	192	0	0	0	0	0
17	100	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	150
22	0	0	600	0	0	0
23	0	0	0	0	0	108
24	0	0	0	0	0	0
25	0	0	0	0	242	0
26	0	0	0	0	285	315
27	0	108	0	0	0	0
28	0	0	50	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	92	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	200
2	0	0	0	0	0	100
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	93	0	0	0	0
7	0	92	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	200	0
11	0	308	0	0	0	0
12	0	0	0	0	0	0
13	0	242	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	8	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	216	0	0	0	92	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	92
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	400	0
31	0	0	0	0	0	0

*Replication 5:*

Transfer Set:

	A	B	C	D	E	F
1	185	5	0	0	0	0
2	185	5	0	0	0	0
3	185	5	0	0	0	0
4	105	85	0	0	0	0
5	0	145	0	45	0	0
6	0	145	0	45	0	0
7	0	150	0	40	0	0
8	0	120	0	70	0	0
9	165	0	0	25	0	0
10	165	0	0	25	0	0
11	165	0	0	25	0	0
12	120	0	0	70	0	0
13	0	0	0	5	185	0
14	0	0	0	5	185	0
15	0	0	0	5	185	0
16	0	0	0	70	120	0
17	185	0	0	5	0	0
18	185	0	0	5	0	0
19	125	0	0	65	0	0
20	0	0	10	0	0	180
21	0	0	10	0	0	180
22	0	0	5	0	0	185
23	0	0	85	0	0	105
24	0	0	35	155	0	0
25	0	0	30	160	0	0
26	0	0	35	155	0	0
27	0	0	60	130	0	0
28	155	0	35	0	0	0
29	155	0	35	0	0	0
30	155	0	35	0	0	0
31	135	0	55	0	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	150	0	0	0	0	0
8	0	0	0	0	0	600
9	0	108	0	0	0	0
10	0	92	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	0	92	0
2	0	0	0	0	0	0
3	0	0	0	0	108	0
4	192	0	0	0	0	0
5	90	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	108	0	0	0	0	0
15	150	0	0	0	0	0
16	0	0	0	0	192	0
17	0	0	0	0	100	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	150	0	0	0	0	0
22	0	0	0	600	0	0
23	108	0	0	0	0	0
24	0	0	0	0	0	0
25	242	0	0	0	0	0
26	0	0	0	0	0	600
27	0	0	0	108	0	0
28	0	0	0	0	0	50
29	0	0	0	0	0	0
30	0	0	0	92	0	0
31	0	0	0	0	0	0



Light Load Value:

	A	B	C	D	E	F
1	0	0	0	0	200	0
2	0	0	0	100	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	93	0	0	0	0	0
7	92	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	200	0	0	0	0
11	0	216	0	0	92	0
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	8	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	308	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	92	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	400	0	0
31	0	0	0	0	0	0

**Experiment 3:**

*Replication 1:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	0	0	5	185
2	0	0	0	0	5	185
3	0	0	0	0	5	185
4	0	0	0	0	85	105
5	0	0	0	25	165	0
6	0	0	0	25	165	0
7	0	0	0	25	165	0
8	0	0	0	125	65	0
9	185	0	0	5	0	0
10	185	0	0	5	0	0
11	155	0	0	35	0	0
12	0	0	0	10	180	0
13	0	0	0	10	180	0
14	0	0	0	10	180	0
15	0	0	0	130	60	0
16	185	0	0	5	0	0
17	185	0	0	5	0	0
18	80	0	0	110	0	0
19	100	90	0	0	0	0
20	0	165	25	0	0	0
21	0	170	20	0	0	0
22	0	160	30	0	0	0
23	0	0	35	0	155	0
24	0	0	35	0	155	0
25	0	0	60	0	130	0
26	0	155	35	0	0	0
27	0	155	35	0	0	0
28	0	155	35	0	0	0
29	0	135	55	0	0	0
30	0	0	5	0	185	0
31	0	0	5	0	185	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	150	0	0	0	0	0
8	0	0	0	0	600	0
9	0	0	0	0	0	108
10	0	0	0	0	0	92
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	92	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	108	0	0
4	192	0	0	0	0	0
5	90	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	108
15	0	0	0	0	150	0
16	0	0	0	0	192	0
17	0	0	0	0	100	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	150	0	0	0	0	0
22	0	600	0	0	0	0
23	108	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	242	0
26	0	0	0	364	236	0
27	108	0	0	0	0	0
28	0	0	0	0	50	0
29	0	0	0	0	0	0
30	0	92	0	0	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	200	0	0	0	0
2	0	0	0	100	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	93
7	92	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	200
11	308	0	0	0	0	0
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	8	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	308	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	92	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	400	0	0	0	0
31	0	0	0	0	0	0

*Replication 2:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	0	0	5	185
2	0	0	0	0	5	185
3	0	0	0	0	5	185
4	0	0	0	0	85	105
5	0	0	0	20	170	0
6	0	0	0	20	170	0
7	0	0	0	15	175	0
8	0	0	0	145	45	0
9	185	0	0	5	0	0
10	185	0	0	5	0	0
11	65	0	0	125	0	0
12	0	170	0	20	0	0
13	0	170	0	20	0	0
14	0	175	0	15	0	0
15	0	75	0	115	0	0
16	185	0	0	5	0	0
17	185	0	0	5	0	0
18	95	0	0	95	0	0
19	85	105	0	0	0	0
20	0	185	5	0	0	0
21	0	155	35	0	0	0
22	0	0	35	0	155	0
23	0	0	35	0	155	0
24	0	0	35	0	155	0
25	0	0	55	0	135	0
26	0	155	35	0	0	0
27	0	155	35	0	0	0
28	0	160	30	0	0	0
29	0	130	60	0	0	0
30	175	0	15	0	0	0
31	175	0	15	0	0	0



Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	150	0	0	0	0	0
8	0	0	0	0	600	0
9	0	0	0	0	0	108
10	0	0	0	0	0	92
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	92	0	0
2	0	0	0	0	0	0
3	0	108	0	0	0	0
4	192	0	0	0	0	0
5	0	90	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	108
15	0	150	0	0	0	0
16	0	192	0	0	0	0
17	0	100	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	150	0	0	0	0	0
22	0	600	0	0	0	0
23	0	0	0	108	0	0
24	0	0	0	0	0	0
25	0	0	0	242	0	0
26	0	0	0	0	600	0
27	108	0	0	0	0	0
28	0	0	0	0	0	50
29	0	0	0	0	0	0
30	0	92	0	0	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	200	0	0
2	0	100	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	93
7	92	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	200
11	308	0	0	0	0	0
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	8	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	308	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	92	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	400	0	0	0	0
31	0	0	0	0	0	0

*Replication 3:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	0	0	5	185
2	0	0	0	0	5	185
3	0	0	0	0	5	185
4	0	0	0	0	85	105
5	0	5	0	0	185	0
6	0	5	0	0	185	0
7	0	5	0	0	185	0
8	0	165	0	25	0	0
9	0	165	0	25	0	0
10	0	165	0	25	0	0
11	0	75	0	115	0	0
12	185	0	0	5	0	0
13	185	0	0	5	0	0
14	185	0	0	5	0	0
15	85	0	0	105	0	0
16	0	185	0	5	0	0
17	0	165	0	25	0	0
18	0	0	0	5	185	0
19	0	0	0	5	185	0
20	0	0	0	5	185	0
21	0	0	0	140	50	0
22	185	0	0	5	0	0
23	80	0	0	110	0	0
24	80	110	0	0	0	0
25	0	35	0	0	155	0
26	0	35	0	0	155	0
27	0	30	0	0	160	0
28	0	85	0	0	105	0
29	0	45	0	145	0	0
30	0	40	0	150	0	0
31	0	100	0	90	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	150	0	0	0	0	0
8	0	0	0	0	600	0
9	0	0	0	0	0	108
10	92	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	92	0	0	0	0
2	0	0	0	0	0	0
3	108	0	0	0	0	0
4	0	0	0	192	0	0
5	0	0	0	0	0	90
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	108	0	0	0	0
15	150	0	0	0	0	0
16	192	0	0	0	0	0
17	0	100	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	150	0
22	0	264	0	0	336	0
23	0	0	0	108	0	0
24	0	0	0	0	0	0
25	242	0	0	0	0	0
26	432	0	0	168	0	0
27	0	0	0	108	0	0
28	0	0	0	0	50	0
29	0	0	0	0	0	0
30	0	0	0	0	92	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	200	0	0	0	0
2	0	0	0	100	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	93
7	92	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	200	0	0	0	0	0
11	0	0	0	0	0	308
12	0	0	0	0	0	0
13	0	242	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	8	0	0



BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	308	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	92	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	400	0
31	0	0	0	0	0	0

*Replication 4:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	0	0	5	185
2	0	0	0	0	5	185
3	0	0	0	0	5	185
4	0	0	0	0	85	105
5	0	0	50	0	140	0
6	0	0	50	0	140	0
7	0	0	50	0	140	0
8	0	0	50	0	140	0
9	165	0	25	0	0	0
10	165	0	25	0	0	0
11	170	0	20	0	0	0
12	135	0	55	0	0	0
13	0	0	35	0	0	155
14	0	0	35	0	0	155
15	0	0	30	0	0	160
16	0	0	60	0	0	130
17	130	0	60	0	0	0
18	130	0	60	0	0	0
19	130	0	60	0	0	0
20	0	185	0	5	0	0
21	0	185	0	5	0	0
22	0	185	0	5	0	0
23	0	120	0	70	0	0
24	185	0	0	5	0	0
25	185	0	0	5	0	0
26	185	0	0	5	0	0
27	45	0	0	145	0	0
28	0	185	0	5	0	0
29	0	185	0	5	0	0
30	0	185	0	5	0	0
31	0	100	0	90	0	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	150	0	0	0	0	0
8	0	0	0	0	0	600
9	0	0	0	0	108	0
10	0	0	0	0	92	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

Medium Load Value:

	A	B	C	D	E	F
1	0	92	0	0	0	0
2	0	0	0	0	0	0
3	108	0	0	0	0	0
4	0	0	0	192	0	0
5	90	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	108	0	0
15	150	0	0	0	0	0
16	0	0	0	0	0	192
17	0	0	0	0	0	100
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	150	0	0	0
22	600	0	0	0	0	0
23	0	108	0	0	0	0
24	0	0	0	0	0	0
25	0	0	242	0	0	0
26	0	548	0	0	52	0
27	0	0	108	0	0	0
28	0	0	50	0	0	0
29	0	0	0	0	0	0
30	92	0	0	0	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	200	0	0	0	0
2	100	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	93	0	0	0	0	0
7	92	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	200	0
11	0	87	0	0	221	0
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	8	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	308
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	92	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	400	0	0	0	0	0
31	0	0	0	0	0	0

*Replication 5:*

Transfer Set:

	A	B	C	D	E	F
1	0	0	0	0	5	185
2	0	0	0	0	5	185
3	0	0	0	0	5	185
4	0	0	0	0	85	105
5	0	0	0	20	170	0
6	0	0	0	20	170	0
7	0	0	0	15	175	0
8	0	0	0	145	45	0
9	0	0	0	5	0	185
10	0	0	0	40	0	150
11	0	0	0	5	185	0
12	0	0	0	5	185	0
13	0	0	0	5	185	0
14	0	0	0	145	45	0
15	0	185	0	5	0	0
16	0	185	0	5	0	0
17	0	120	0	70	0	0
18	0	0	0	5	185	0
19	0	0	0	5	185	0
20	0	0	0	110	80	0
21	0	30	0	0	0	160
22	0	30	0	0	0	160
23	0	25	0	0	0	165
24	0	75	0	0	0	115
25	0	30	0	160	0	0
26	0	30	0	160	0	0
27	0	30	0	160	0	0
28	0	65	0	125	0	0
29	0	15	0	0	175	0
30	0	15	0	0	175	0
31	0	15	0	0	175	0

Heavy Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	150
8	0	0	0	0	600	0
9	108	0	0	0	0	0
10	0	0	0	0	0	92
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0



Medium Load Value:

	A	B	C	D	E	F
1	0	0	0	92	0	0
2	0	0	0	0	0	0
3	0	108	0	0	0	0
4	192	0	0	0	0	0
5	0	90	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	108	0
15	0	0	0	0	150	0
16	0	0	0	0	192	0
17	0	100	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	150	0
22	0	0	0	600	0	0
23	0	0	0	0	108	0
24	0	0	0	0	0	0
25	0	0	0	0	0	242
26	0	0	0	0	323	277
27	0	0	0	0	0	108
28	0	0	0	50	0	0
29	0	0	0	0	0	0
30	0	0	0	92	0	0
31	0	0	0	0	0	0

Light Load Value:

	A	B	C	D	E	F
1	0	0	0	200	0	0
2	100	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	93
7	0	0	0	0	0	92
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	200
11	0	0	0	0	0	308
12	0	0	0	0	0	0
13	242	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	8	0	0	0	0	0

BYU Load Value:

	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	308	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	92	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	400	0	0
31	0	0	0	0	0	0

*(This page is intentionally left blank)*

## BIOGRAPHY



Fikri Irsyad Muhammad was born on September 29<sup>th</sup> 1996 in Surabaya. The author went through formal education in SDN Kaliasin 1 Surabaya (2002-2008), SMP Negeri 2 Surabaya (2008-2011), and SMA Negeri 5 Surabaya (2011-2014). After finishing high school, the author continued with undergraduate study in Industrial Engineering Department of Institut Teknologi Sepuluh Nopember (ITS) Surabaya from 2014 until 2018.

During the 5<sup>th</sup> semester of undergraduate study, the author participated in a student mobility program in Universiti Teknikal Malaysia Melaka (UTeM) for one semester from September 2016 until January 2017. During this program the author participated in a Lean Six-Sigma training course held by Infineon Melaka and received the Six-Sigma Yellow Belt certification. The author also completed an on-the-job training as part of the undergraduate study in form of an internship in PT Pertamina (Persero), specifically in Supply Chain & Distribution (SC&D) Section of Refinery Planning & Optimization (RP&O) Function – Refinery Unit V Balikpapan in July 2017 until September 2017.

Any critics or suggestion for this research will be fully appreciated by the author. For further discussion the author can be reached on email address [fikriirsyad@gmail.com](mailto:fikriirsyad@gmail.com).