Simultaneous mixed-integer disjunctive optimization for synthesis of petroleum refinery topology

Processing Alternatives for Naphtha Produced from Atmospheric Distillation Unit

by

Aida Azwana binti Sabidi

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Chemical Engineering)

January 2010

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Simultaneous mixed-integer disjunctive optimization for synthesis of petroleum refinery topology

Processing Alternatives for Naphtha Produced from Atmospheric Distillation Unit

by

Aida Azwana Binti Sabidi

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

S. KHOR

(Khor Cheng Seong)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

January 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

AIDA AZWANA BINTI SABIDI

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to thank everyone whom had given their support and help throughout the whole period of completing this project.

I would also like to acknowledge the continuous help and support received from my supervisor Mr. Khor Cheng Seong throughout the whole period of completing this final year project. his guidance has really been the main sources of motivation and has driven me in completing this project successfully.

Finally, a bunch of gratitude to fellow colleagues and other lecturers for the their assistance and ideas in completing this project.

AIDA AZWANA BINTI SABIDI

ABSTRACT

In this work, we propose a logic-based modeling technique within a mixed-integer disjunctive superstructure optimization framework on the topological optimization problem for determining the optimal petroleum refinery configuration. We are interested to investigate the use of logic cuts that are linear inequality/equality constraints to the conceptual process synthesis problem of the design of a refinery configuration.

The logic cuts are employed in two ways using 0–1 variables: (1) to enforce certain design specifications based on past design experience, engineering knowledge, and heuristics; and (2) to enforce certain structural specifications on the interconnections of the process units. The overall modeling framework conventionally gives rise to a mixed-integer optimization framework, in this case, a mixed-integer linear programming model (because of the linearity of the constraints). But in this work, we elect to adopt a disjunctive programming framework, specifically generalized disjunctive programming (GDP) proposed by Grossmann and co-workers (Grossmann, I. E. (2002). Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization & Engineering*, 3, 227.) The proposed GDP-based modeling technique is illustrated on a case study to determine the optimal processing route of naphtha in a refinery using the GAMS/LogMIP platform, which yields practically-acceptable solution. The use of LogMIP obviates the need to reformulate the logic propositions and the overall disjunctive problem into algebraic representations, hence reducing the time involved in the typically time-consuming problem formulation. LogMIP typically leads to less computational time and number of iterations in its computational effort because the associated GDP formulation involves less equations and variables compared to MILP. From the computational experiments, it is found that logical constraints of design specifications and structural specifications potentially play an important role to determine the optimal selection of process units and streams. Hence, in general, the GDP formulation can be improved by adding or eliminating constraits that can accelerate or slow-down the problem solution respectively.

TABLE OF CONTENT

CHAPTER 1: INTRODUCTION

CHAPTER 2: LITERATURE REVIEW

CHAPTER 3: METHODOLOGY

3.2 Model Formulation

CHAPTER 4: RESULTS AND DISCUSSION

APPENDICES . . . .

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Optimization is the main objective of process design. Selecting the best among a set of possible solutions requires good engineering judgement to analyze the process with respect to desired objectives. It is important to identify between the objectives of realizing the largest production, the greatest profit, the minimum cost and the least energy usage. In order to find the best solution within the given constraints and flexibilities, a trade-off usually exists between capital and operating costs.

In process synthesis, there are two major approaches in order to determine the optimal configuration of a flowsheet condition and its operation conditions. The first approach can be solved in sequential form, by decomposition, fixing some elements in the flowsheet and then by using heuristic rules to determine changes in the flowsheets that may leed to an improved solution.

The second strategy can be applied to solve a process synthesis problem is based on simultaneous optimization using mathematical programming (Grossmann, 1996). This strategy requires to postulate a superstructure that includes equipment that can be potentially selected in the final flowsheet, as well as their interconnection. The equations of the equipment and their connectivity and constraints for the operating conditions are incorporated in an optimization problem where an objective function is specified such as cost minimization or profit maximization. This approach requires the use of discrete variables to represent the choices of equipment, with which the model becomes a mixed integer linear or non-linear program (MILP or MINLP). The advantage of mathematical programming strategies for process synthesis is that it can perform simultaneous optimization of the configuration and operating condition. Te drawback is that global optimality conditions cannot be guaranteed for nonlinear models unless specific method for global optimization are used.(Grossmann and Yeomand, 1998).

Optimization models provide a means of reducing the number of alternatives which need to be simulated in detail, i.e., screening them. These models search the space of possible design variable values and identify an optimal design and/or operating policy for a given system design objective and set of constraints. The sensitivity of the optimal solution to changes in the model parameters can be readily determined and tradeoffs between several conflicting objectives can also be calculated with most optimization models. These models are usually extensions of simulation models and include as unknowns the design or operating variables (decision variables) of each alternative. These models include relationships which describe the state variables and costs or benefits of each alternative as a function of the decision variables. Constraints are also included in the models to restrict the values of the design or state variables. Optimization models are generally used for preliminary evaluation or screening of alternatives and to identify important data needs prior to extensive data collection and simulation modeling activities.

1.2 Problem Statement

We consider the following process synthesis problem of superstructure optimization for the topology design of a refinery. Assume we are given the following data: (a) fixed production amounts of desired products; (b) available process units and the ranges of their capacities; (c) cost of crude oil and cost structures for the process units;. Thus, we wish to determine the optimal topology or configuration of the refinery in terms of: (a) the selection and sequencing of the process units and materials streams, and (b) the optimum operating level of the stream flowrates. One of the challenges in process synthesis problems is the effective and efficient consideration of essential qualitative design information within a formal mathematical modeling framework. In this work, we propose the extensive use of logic cuts as a means of stipulating these very useful but sometimes tacit design knowledge within an automated optimization-based computational framework. The inclusion of these logic cuts obviates the necessity for post-optimization analysis of a problem solution, in which the latter, in the first place itself, could end up with a suboptimal solution due to the non-enforcement of these qualitative design information

1.3 Objective and Scope of Study

The expected objectives to be achieved in this work are as follows:

- To develop a superstructure representation for a refinery network topology with a suitable level of detail and abstraction by considering the processing alternatives for naphtha.
- To formulate an optimization model based on the superstructure representation by adopting the generalized disjunctive programming (GDP) framework that incorporates both continuous and discrete decisions. The model formulation includes: (a) constant-yield-based

linear material balances, and (b) logical constraints enforcing the design specifications and structural specifications, in which the latter stipulates the interconnectivity relationships among the units and the streams, for the selection and sequencing of the alternative routes;

- To solve GDP by using Logical Mixed Integer Programming (LogMIP)
- To obtain the optimal flow rates for selected streams which will minimize the total cost.

1.4 The relevancy of the project

The most important of the applicability of a mathematical modeling in real life situation, is weather it can be used to solve problems of industry-relevant sizes. It provides systematic framework for modeling and simultaneous optimization and automated capibiites for synthesis problem (Grossmann and Daichendt, 1996). A good mathematical modeling is greatly depend on the solution time of the algorithm. When considering MILP or nonlinear problems, the solution time can be often reduced significantly by appropriate modeling (Kallarath, 2004).

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The optimization approach in process synthesis consists of representation of alternatives and mathematical modeling. Then, followed by detail explaination with regards to refining process.

2.2 Superstructure Representation of Alternatives

A superstructure consists of all possible process design alternatives of interest by incorporating the different process units and their feasible interconnections. Hence, each alternative can be a feasible or optimal process flowsheet. There are three types of superstructure representation which are: State-Task Network (STN), State-Equipment Network (SEN) and Resource-Task Network (RTN). In this research, STN representation is chosen. STN assumes that processing tasks produce and consume states. The states and tasks are defined first, leaving unknown equipment assignment to a second stage. Feedstock must be connected to product and vice-versa. Each intermediate state and task must be on at least one such path. Some tasks are conditional; others must be present in all design alternatives. There is no need to distinguish one from the other at the level of representation, but only at the level of model. One or more of these operations (temperature, momentum, mass or energy transfer) may be performed in one task if technically feasible.

2.3 Mathematical Modeling

2.3.1 Generalized Disjunctive Programming (GDP) Modeling

Turkay and Grossmann (1996) have shown that generalized disjunctive programming model is very useful in modeling nonlinear discrete continuous optimization problems. They have shown benefits for the modeling and solution of flowsheet synthesis problems in which discrete decisions for the selection process units have to be made among the several alternatives. In order to used GDP (Raman & Grossmann, 1994) to model STN or SEN representations, it is necessary to identify the conditional constraints from those that must hold for all synthesis alternatives. The conditional constraints will be represented with disjunctions and assigned a Boolean variable that represents its existence (if the Boolean variable takes a value of 'true'). In general, mixers and splitters can be considered conditional tasks. However, if the equations that are applied to the mixer and splitter are only mass and energy balances, these constraint do not involve any type of discrete decision or discrete assignment for them to be valid.

The major strengths of GDP optimization model are as follows (Vecchietti, Lee and Grossmann, 2003):

i. it allows a symbolic or quantitative representations of discrete and continuous optimization problems;

ii. enables a systematic transformation of abstract disjunctive logic propositions into algebraic constraints that can be directly incorporated in conventional concrete mathematical programming models;

iii. it reduces the combinatorial effort involved in problem modeling by reducing the number of discrete variables especially the binary 0-1 variables;

iv.

v. it improves handling of nonlinearities;

vi. in solving design problems of process networks with restricted unit sizes, the GDP approach potentially does not require the duplication of equation models for each potential unit (Turkay and Grossmann, 1994);

vii. no binary 0-1 variables are explicitly included in a GDP formulation;

viii. avoids the use of big-M logical constraints, which yield a poor relaxation and prevent zero flow (Yeomans and Grossmann, 1999).

The generalized disjunctive programming model is given as follows:

$$
\begin{aligned}
&\min Z = \sum_i c_i + f(x) \\
&s.t. \\
&g(x) \leq 0 \\
&\begin{bmatrix} Y_i \\ h_i(x) \leq 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in D \qquad \text{(P)} \\
&\Omega(Y) = True \\
&x \in R^n, c_i \geq 0, \mathrm{Y} \in \{\text{True, False}\}^m
\end{aligned}
$$

The nonlinear model (P) involves the following three types of variable: x and c_i are the continuous variables (the former correspond to flows, pressures, temperatures, alike the latter are used to exclusively represent fixed charges); Boolean variables, Yi, that are associated with the existence of units and are used to indicate whether a given disjunction i is true or false. In generalized disjunctive nonlinear optimization model (P), the objective function and the first set of constraints may involve linear and nonlinear functions. The first set of constraints represents global inequalities that hold irrespective of the discrete choices. The set of disjunctions, D, apply for the processing units. If a process unit exists (Yi= True), then the equations and constraints describing that unit are enforced and a fixed charge is applied; otherwise ($\neg\ Yi$ = False) a subset of continuous variables and the fixed charge are set to zero.

In general, at least three approaches are available to solve GDP:

i. Reformulation of the disjunctions in GDP into MILP via big-M reformulations; The GDP problem (GDP) can be reformulated as the following MINLP problem (BM) by replacing the Boolean variables Yjk by binary variables yjk and using the big-M constraints. The logic constraints Ω(Y) are converted into linear inequalities (Williams, 1999) that leads to the following big-M MINLP:

$$
\begin{aligned}
\min\ & Z = \sum_{k \in K} \sum_{j \in J_k} \gamma_{jk} y_{jk} + f(x) \\
\text{s.t.}\ & r(x) \leq 0 \\
& g_{jk}(x) \leq M_{jk}\left(1 - y_{jk}\right), \quad j \in J_k,\ k \in K \\
& \sum_{j \in J_k} y_{jk} = 1, \quad k \in K \\
& Ay \leq a \\
& x \geq 0 \\
& y_{jk} \in \{0,1\}, \quad j \in J_k,\ k \in K
\end{aligned}
$$

ii. Reformulation of the disjunctions in GDP into MILP via convex hull formulation, which provides tighter relaxation compared to the first approach as according to Turkay and Grossmann (1996);

A disjunction of the form:

$$
\begin{bmatrix} Z_k \\ A_1 x \leq b_1 \end{bmatrix} \vee \begin{bmatrix} \neg Z_k \\ A_2 x \leq b_2 \end{bmatrix}
$$

is transformed into the following constraints by means of the convex hull formulation of disjunctions (Turkay and Grossmann, 1996b):

$$\begin{aligned}
&x = x_1 + x_2 \\
&A_1 x_1 \le b_1 y_1 \\
&A_2 x_2 \le b_2 y_2 \\
&L y_1 \le x_1 \le U y_1 \\
&L y_2 \le x_2 \le U y_2 \\
&y_1 + y_2 = 1
\end{aligned}$$

- where A_1 and A_2 are coefficient matrices for two different linear sets of constraints; b_1 and b_2 are the right-hand sides of the constraint sets; x_1 and x_2 are variable vectors; and y_1 and y_2 are binary variables.
- It can be seen that every variable inside the disjunction term results in three variables in the MILP problem (the original variable itself plus two disaggregated variables), as well as the inclusion of bounding constraints.
- For a number of synthesis problems, this increase in the number of variables in the master problem is justified because the MILP relaxation becomes tighter (Turkay and Grossmann, 1996b).

iii. Solution of GDP using GAMS/LOGMIP solver.

2.3.2 Relation Between MILP Modelling and Logical Inference

In order to obtain an equivalent mathematical representation for any propositional logic expression, one must first consider basic logical operators to determine how each can be transformed into an equivalent representation in the form of an equation or inequality. These transformations are then used to convert general logical expressions into an equivalent mathematical representation.

The basic unit of propositional logic expression, which can correspond to a state or to an action, is called a literal which is a single variable that can assume either of two values, true or false. Associated with each literal P, there is another literal NOT P $(\neg P)$ such that either P or $(\neg P)$ is always true. A clause is a set of literals separated by OR operators and is also called a disjunction. A proposition is any logical expression and consists of a set of clauses Pi, i=1 are related by the logical operators OR, AND, IMPLICATION, as stated in Raman and Grossmann (1991).

To each proposition P_i, a binary variable y_i is assigned. Then the negation or complement of P_i, $(\neg P_i)$ is given by $1 - y_i$. The logical value of true corresponds to the binary value of 1 and false corresponds to the binary value of 0. The basic operators used in propositional logic and the representation of their relationships are shown in Table 1. The procedure to convert a logical expression into its corresponding conjunctive normal form was formalized by Clocksin & Mellish.

Table 2.1: Representation of logical relations with linear inequalities (Raman and Grossmann (1994)

Logical operator	**Example of use for process networks**	**Logic proposition**	**Logical Boolean expression**	**Representation as algebraic integer linear inequality/equality constraint**
Logical OR	For selection of at least one process unit (or more than one unit or all of the units) in consideration	---	$P_1 \vee P_2 \vee \cdots \vee P_r$	$y_1 + y_2 + \cdots + y_r \geq 1$
Logical AND	For selection of all process units in consideration	---	$P_1 \wedge P_2 \wedge \cdots \wedge P_r$	$y_1 \geq 1; y_2 \geq 1; \cdots; y_r \geq 1$
Implication	Select unit 1 only if unit 2 is selected (e.g., select FCC only if the upstream HDS is selected)	P_2 only if P_1 $P_1 \Rightarrow P_2$ is logically equivalent to $\neg P_1 \vee P_2$	$\neg P_1 \vee P_2$	$(1-y_1)+y_2 \geq 1$ $y_1 - y_2 \leq 0$ or $y_1 \leq y_2$
Equivalence	Selecting a unit implies the selection of another unit or other units	P_1 if and only if P_2 $(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$ which can also be written as: $P_1 \Leftrightarrow P_2$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$(1-y_1)+y_2 \geq 1$ $-y_1 + y_2 \geq 0$ $y_1 - y_2 \leq 0$ $y_1 \leq y_2$ and $(1-y_2)+y_1 \geq 1$ $-y_2 + y_1 \geq 0$ $y_2 - y_1 \leq 0$ $y_2 \leq y_1$ or $y_1 = y_2$
Exclusive OR (EOR)	For selection of only one process unit (or material stream)	Exactly one of the variables is true	$P_1 \veebar P_2 \veebar \cdots \veebar P_r$ or this can equivalently be written as: P_1 EOR P_2 EOR ...EOR P_r	$y_1 + y_2 + \cdots + y_r = 1$
Classification	For selection of any process unit.	$Q = \{P_1, P_2, \ldots, P_r\}$ Q is true if any of the variables inside the brackets are true	---	$y_q = y_1 + y_2 + \cdots + y_r$
"Combination" of Equivalence and OR	Selection of at least one process unit (or more than one unit or all of the units) implies the selection of another unit or other units	---	$(P_1 \vee P_2) \Leftrightarrow P_3$	$p_3 \geq p_1$ $p_3 \geq p_2$ $p_1 + p_2 \geq p_3$
"Combination" of Equivalence and EOR	Selection of only one process unit (or material stream)	---	$(P_1 \veebar P_2) \Leftrightarrow P_3$	$p_1 + p_2 = p_3$

2.4 Refining Process

Figure 2.1 shows the processing sequence in modern refinery, indicating major process flows between operations (Agilent Technologies). The crude oil is heated in a furnace and send to an atmospheric distillation tower, where it is then separated into butanes and lighter wet gas, unstablized full range gasoline, heavy naphtha, kerosene, heavy oil gas and topped crude. The topped crude is sent to the vacuum tower and separated into a vacuum gas oil overhead stream and reduced crude bottoms. The reduced crude bottoms from the vacuum tower is thermally cracked in a delayed coker to produce wet gas, coker gasoline and coke.

The atmospheric and vacuum crude unit gas oils and coker gas oil are used as feedstocks for the catalytic cracking or hydrocraking units. These units crack the heavy molecules into compounds boiling in the gasoline and distillate fuel ranges. The products from the hydrocracker are saturated. The unsaturated catalytic cracker products are saturated and improved in quality by catalytic reforming or hydrotreating. The gasoline streams from the crude tower, coker and cracking units are fed to the catalytic reformer to imprive octane numbers. The products from the catalytic reformer are blend with gasoline for sale.

The wet gas streams for the crude tower, coker, and cracking units are fractioned in the vapor recovery section into fuel gas, liquefied petroleum gas (LPG), unsaturated hydrocarbons, normal butane and isobutene. Meanwhile, the fuel gas is burned in refinery furnaces and the normal butane is blended into gasoline or LPG. The unsaturated hydrocarbons and isobutane are sent to the alkylation unit. The middle distillates from the crude unit, coker and crakin gunits are blended into diesel, jet fuels and furnace oils. The heavy vacuum gas oil and reduced crude form paraffin are processed into lubricating oils

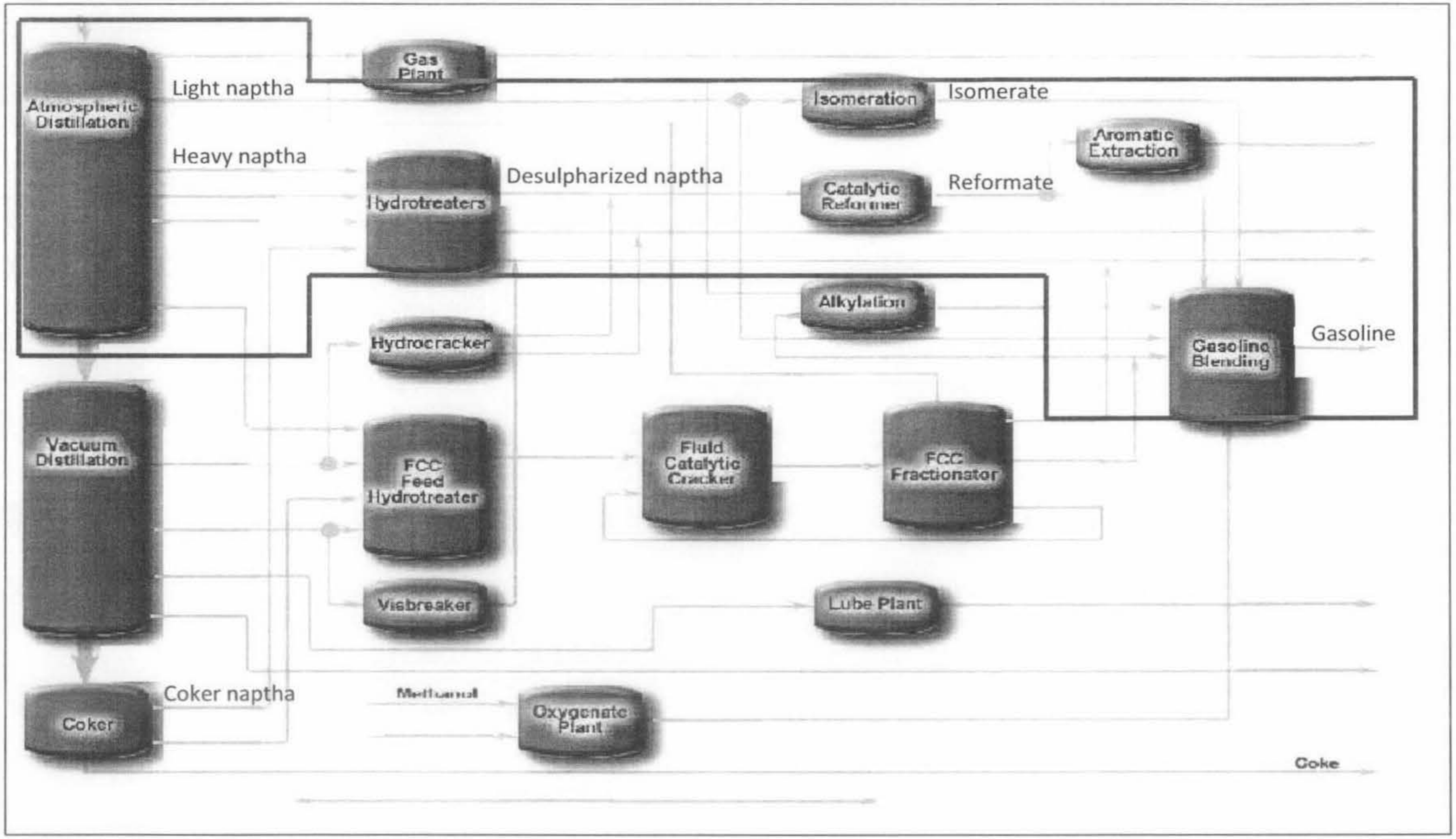

Figure 2.1: Refinery flow diagram (Agilent Technologies, 2010)

CHAPTER 3

METHODOLOGY

There are three basic elements required in development of algorithmic methods for process synthesis which are:

Step 1: Problem representation

Step 2: Modelling/Optimization model formulation

Step 3: Solution strategies (to search for the optimal flowsheet or design)

3.1 Superstructure Representation

Superstructure representation for the naphtha produced from the atmospheric distillation unit (ADU) with shows the optimized refinery topology is presented in this chapter. Figure 3.1 depicts the state–task network (STN) superstructure representation while Table 3.1 shows the Legend for modified state–task network (STN) superstructure representation in Figure 3.1 In developing the superstructure representation, integer binary 0–1 variables are employed as structural variables to represent discrete decisions involved in the selection of the alternative:

1. process units or tasks, as represented by binary variable y_i, and
2. flow rates of the material streams or states, as represented by continuous variable f_i

Table 3.1: Legend for the STN superstructure representation in Figure 3.1

CR	Crude oil	HDT	Hydrotreater
ADU	Atmospheric distillation unit	LPG	Liquefied petroleum gas
LSRN	Light straight run naphtha	H2	Hydrogen
HSRN	Heavy straight run naphtha	ISO	Isomerization unit
NAP	Naphtha	SRU	Sulfur recovery unit
MIX	Mixer	REF	Reformer
SPLT	Splitter	S	Sulfur
VIS	Visbreaker	FG	Fuel gas
COK	Coker	BLND	Blending
FCC	Fluidized catalytic cracker	FGH	Fuel gas header
HCR	Hydrocracker	GSLN	Gasoline
PCHN	Purchased naphtha	TG	Tail gas

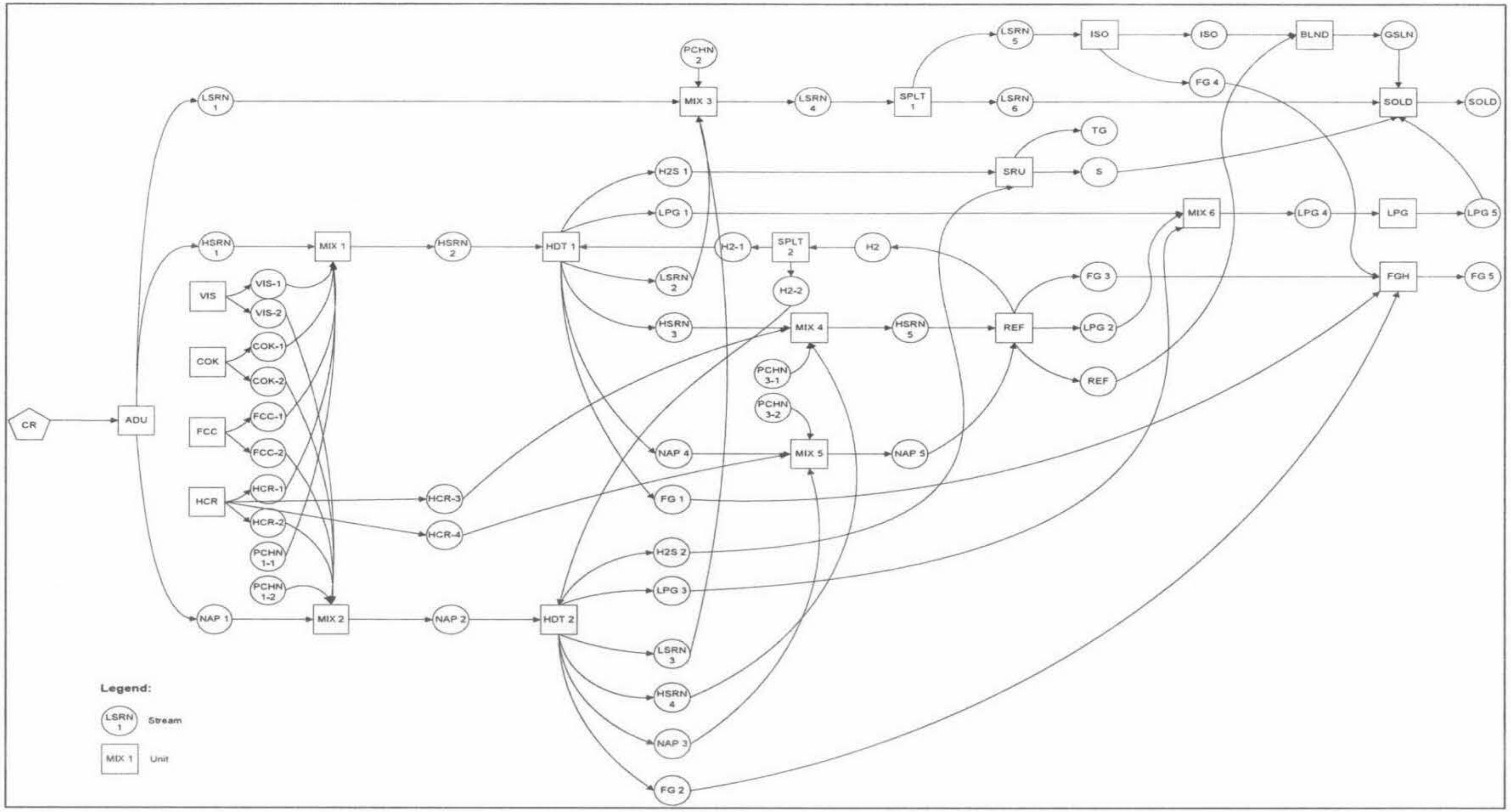

Figure 3.1: State–task network (STN) superstructure representation for the naphtha produced from the ADU.

3.1.1Detailed description of Superstructure

The first processing step in petroleum refining is crude distillation, in which crude oil (CR) is distilled into oil fractions with respect to its boiling points. Naphtha constitutes the lighter fractions that are obtained from this process. Depending on the distillation column design as well as the refinery economics, the atmospheric distillation unit (ADU) can produce: (a) light straight run naphtha (LSRN–1) *and* heavy straight run naphtha (HSRN–1), *or* (b) an undifferentiated class of naphtha, typically referred to as "wild naphtha" (NAP–1), for which, the 0–1 structural variables of z_i are used to represent these three possible states of the naphtha produced from the ADU.

In the first case, LSRN–1 is mixed with purchased naphtha (PCHN–2) and LSRN–2 from the hydrotreater HDT–1 in a mixer (MIX–3). The output from MIX–3, i.e., LSRN–4, can undergo two processes: (a) it is used as a feedstock for the isomerization unit (ISO), and (b) it is sold as a final product. Isomerization yields isomerate (ISO), which is one of the blending components for gasoline (GSLN). Meanwhile, HSRN–1 is mixed with naphtha from the cracking of heavier fractions in MIX–1 before being sent to HDT–1 to be desulfurized. HDT–1 produces hydrogen sulfide gas (H2S–1), liquefied petroleum gas (LPG–1), desulfurized naphtha (LSRN–2, HSRN–3, and NAP–4), and fuel gas (FG–1). H2S–1 is sent to the sulfur recovery unit (SRU) where sulfur (S) is extracted and finally sold. All LPG (LPG–1, LPG–2 and LPG–3) are sent to MIX–6 and subsequently to the LPG recovery unit (LPG), from which treated LPG (LPG–5) is sold. Similar to the outputs from ADU, the desulfurized naphtha from HDT–1 can be classified as light (LSRN–2) *and* heavy (HSRN–3) *or* wild (NAP–4). LSRN–2 is mixed with LSRN–1 and PCHN–2 in MIX–3, as previously stated. On the other hand, HSRN–3 is sent to a mixer (MIX–4), possibly with purchased naphtha (PCHN–3-2) *and/or* naphtha from the hydrocracker (HCR–3). The output of MIX–4 (HSRN–5) is the feedstock for the reformer (REF). FG–1 goes to the fuel gas header (FGH) which supplies fuel gas (FG–5) to the entire refinery. In the case that NAP–4 is produced from HDT–1, it will also be mixed with purchased naphtha (PCHN-3-1) *and/or* naphtha from the hydrocraker (HCR–4) in MIX–5, whose output of NAP–5 is sent to the reformer. The products from the reformer are hydrogen gas (H2), fuel gas (FG–3), liquefied petroleum gas (LPG–2), and reformate (REFs). H2 is a feed to the HDTs while reformate is used as a gasoline blending component. FG–3 is sent to the FGH.

In the second case involving the further treatment and conversion of the wild naphtha NAP–1 exiting the ADU, the processing route is similar to the first case in that NAP–1 will be mixed with naphtha from the cracking processes in MIX–2 before being hydrotreated in HDT–2. The products from HDT–2 are H2S–2, LPG–3, desulfurized naphtha of LSRN–3, HSRN–4, and NAP–3, and FG–2. Each product has the exact same route as the products from HDT–1. Other than distillation, naphtha is also produced from the cracking of distillation bottoms in the visbreaker (VIS), coker (COK), catalytic cracker (FCC), hydrocracker (HCR). VIS has the lowest severity while COK has the highest. Hence, VIS is not used for processing heavy crude. On the other hand COK and HCR are used for processing heavy crude but not light crude. The FCC technology can be used for both types of crude. The FCC technology can be used for both high and low severity processing modes.

Table 3.2: Crude processing modes

Light crude processing	Heavy crude processing
FCC, VIS	FCC, COK, HCR

3.2 Model formulation

3.2.1 Material Balance for the naphtha processing network structure

The material balance on the process units can be in two forms which are:

a. the overall input-output mass flow rates

b. the component mass balances

Table 3.3: Material balances in terms of mass flow rates around the process units

ADU	$0.4176 f_{\text{CR}} = f_{\text{NAP}1} + f_{\text{LSRN}1} + f_{\text{HSRN}1}$
HDT 1	$1.9821\left(f_{\text{HSRN}2} + f_{\text{H2\_1}}\right) = f_{\text{FG}1} + f_{\text{H2S}1} + f_{\text{LPG}1} + f_{\text{LSRN}2} + f_{\text{HSRN}3} + f_{\text{NAP}4}$
HDT 2	$1.9821\left(f_{\text{NAP}2} + f_{\text{H2\_2}}\right) = f_{\text{FG}2} + f_{\text{H2S}2} + f_{\text{LPG}3} + f_{\text{LSRN}3} + f_{\text{HSRN}4} + f_{\text{NAP}3}$
ISO	$f_{\text{LSRN}5} = f_{\text{ISO}} + f_{\text{FG}4}$
SRU	$f_{\text{H2S}1} + f_{\text{H2S}2} = f_{\text{S}} + f_{\text{TG}}$
REF	$f_{\text{HSRN}5} + f_{\text{NAP}5} = f_{\text{H2}} + f_{\text{FG}3} + f_{\text{LPG}2} + f_{\text{REF}}$
SOLD	$f_{\text{LSRN}6} + f_{\text{S}} + f_{\text{GSLN}} + f_{\text{LPG5}} = f_{\text{SOLD}}$
BLND	$f_{\text{ISO}} + f_{\text{REF}} = f_{\text{GSLN}}$
LPG	$f_{\text{LPG}4} = f_{\text{LPG}5}$
FGH	$f_{\text{FG}1} + f_{\text{FG}2} + f_{\text{FG}3} + f_{\text{FG}4} = f_{\text{FG}5}$
SPLT 1	$f_{\text{LSRN}4} = f_{\text{LSRN}5} + f_{\text{LSRN}6}$
SPLT 2	$f_{\text{H2}} = f_{\text{H2\_1}} + f_{\text{H2\_2}}$
MIX 1	$f_{\text{HSRN}1} + f_{\text{VIS\_1}} + f_{\text{COK\_1}} + f_{\text{FCC\_1}} + f_{\text{HCR\_1}} + f_{\text{PCHN}1\text{\_1}} = f_{\text{HSRN}2}$
MIX 2	$f_{\text{NAP}1} + f_{\text{VIS\_2}} + f_{\text{COK\_2}} + f_{\text{FCC\_2}} + f_{\text{HCR\_2}} + f_{\text{PCHN}1\text{\_2}} = f_{\text{NAP}2}$
MIX 3	$f_{\text{LSRN}1} + f_{\text{LSRN}2} + f_{\text{LSRN}3} + f_{\text{PCHN}2} = f_{\text{LSRN}4}$
MIX 4	$f_{\text{HSRN}3} + f_{\text{HSRN}4} + f_{\text{PCHN}3\text{\_1}} + f_{\text{HCR\_3}} = f_{\text{HSRN}5}$
MIX 5	$f_{\text{NAP}3} + f_{\text{NAP}4} + f_{\text{PCHN}3\text{\_2}} + f_{\text{HCR\_4}} = f_{\text{NAP}5}$
MIX 6	$f_{\text{LPG}1} + f_{\text{LPG}2} + f_{\text{LPG}3} = f_{\text{LPG}4}$

Table 3.4: Material balances in terms of component mass flowrates around process units (Maples, 2000, p. 96; Parkash, 2003, pp. 37, 116, 225, 144)

ADU	$(0.0555)\ f_{CR} = f_{LSRN\,1}$ $(0.1533)\ f_{CR} = f_{HSRN\,1}$ $(0.2088)\ f_{CR} = f_{NAP\,1}$
HDT 1	$0.0109\ (f_{H2\_1} + f_{HSRN\,2}) = f_{FG\,1}$ $0.0012\ (f_{H2\_1} + f_{HSRN\,2}) = f_{H2S\,1}$ $0.0058\ (f_{H2\_1} + f_{HSRN\,2}) = f_{LPG\,1}$ $0.2610\ (f_{H2\_1} + f_{HSRN\,2}) = f_{LSRN\,2}$ $0.7211\ (f_{H2\_1} + f_{HSRN\,2}) = f_{HSRN\,3}$ $0.9821\ (f_{H2\_1} + f_{HSRN\,2}) = f_{NAP\,4}$
HDT 2	$0.0109\ (f_{H2\_2} + f_{NAP\,2}) = f_{FG\,2}$ $0.0012\ (f_{H2\_2} + f_{NAP\,2}) = f_{H2S\,2}$ $0.0058\ (f_{H2\_2} + f_{NAP\,2}) = f_{LPG\,3}$ $0.2610\ (f_{H2\_2} + f_{NAP\,2}) = f_{LSRN\,3}$ $0.7211\ (f_{H2\_2} + f_{NAP\,2}) = f_{HSRN\,4}$ $0.9821\ (f_{H2\_2} + f_{NAP\,2}) = f_{NAP\,3}$
ISO	$(0.9900)\ f_{LSRN\,5} = f_{ISO}$ $(0.0100)\ f_{LSRN\,5} = f_{FG\,4}$
SRU	$0.8478\ (f_{H2S\,1} + f_{H2S\,2}) = f_{S}$ $0.1522\ (f_{H2S\,1} + f_{H2S\,2}) = f_{TG}$
REF (based on RON = 102)	$0.0320\ (f_{HSRN\,5} + f_{NAP\,5}) = f_{H2}$ $0.0370\ (f_{HSRN\,5} + f_{NAP\,5}) = f_{FG\,3}$ $0.0780\ (f_{HSRN\,5} + f_{NAP\,5}) = f_{LPG\,2}$ $0.8530\ (f_{HSRN\,5} + f_{NAP\,5}) = f_{REF}$
SPLT 1	$(0.9000)\ f_{LSRN\,4} = f_{LSRN\,5}$ $(0.1000)\ f_{LSRN\,4} = f_{LSRN\,6}$

3.2.2 Reformulation of the disjunctions in GDP into MILP via convex hull formulation

According to Turkay and Grossmann (1996), a disjunction is in the form of:

$$\begin{bmatrix} Z_k \\ A_1 x \le b_1 \end{bmatrix} \vee \begin{bmatrix} \neg Z_k \\ A_2 x \le b_2 \end{bmatrix}$$

For example, the direct formulation on the existence of ADU:

$$\text{ADU:} \begin{bmatrix} Y_{\text{ADU}} \\ (0.2088)\ f_{\text{CR}} = f_{\text{NAP1}} \\ (0.0555)\ f_{\text{CR}} = f_{\text{LSRN1}} \\ (0.1533)\ f_{\text{CR}} = f_{\text{HSRN1}} \\ f_{\text{NAP1}} = f_{\text{LSRN1}} + f_{\text{HSRN1}} \\ c_{\text{ADU}} = 228 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\text{ADU}} \\ f_{\text{CR}} = 0 \\ f_{\text{NAP1}} = 0 \\ f_{\text{LSRN1}} = 0 \\ f_{\text{HSRN1}} = 0 \\ c_{\text{ADU}} = 0 \end{bmatrix}$$

By transforming into the following constraints by means of the convex hull formulation of disjunctions (Turkay and Grossmann, 1996b):

$$\begin{aligned} & x = x_1 + x_2 \\ & A_1 x_1 \le b_1 y_1 \\ & A_2 x_2 \le b_2 y_2 \\ & L y_1 \le x_1 \le U y_1 \\ & L y_2 \le x_2 \le U y_2 \\ & y_1 + y_2 = 1 \end{aligned}$$

Therefore, the transformation of the process unit are shown in Appendix A.

3.2.3 Systematic Transformation from Logical Proposition into Mathematical Representation

According to Raman and Grossmann (1994), the three steps procedures to transform each logical proposition are:

1) replace the implication by its equivalent disjunction:

$$P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2 \qquad (1)$$

2) move the negation inward by applying DeMorgan's Theorem:

$$\neg(P_1 \wedge P_2) \Leftrightarrow \neg P_1 \vee \neg P_2 \quad (2) \qquad \neg(P_1 \vee P_2) \Leftrightarrow \neg P_1 \wedge \neg P_2 \quad (3)$$

3) recursively distribute the "OR' over the "AND":

$$(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_2) \wedge (P_1 \vee P_3) \qquad (4)$$

where:

P_i, : proposition, given by binary variable y_i

$\neg P_i$: proposition negation or complement, given by $1 - y_i$

Having converted each logical proposition into its conjunctive normal form representation, $Q_1 \wedge Q_2 \wedge \ldots \wedge Q_S$, it can then be easily expressed as a set of linear equality and inequality constraints.

3.3 Project Activities

The proposed methodology to tackle the process synthesis problem of naphtha produced from the ADU is presented in this chapter. There are four major steps to solve the this problem:

a. Familiarization of the naptha superstructure, environmental indicators using matrices, relation between MILP modeling and logical inference together with familiarization of GDP

b. Formulation or modeling of the superstructure in a mathematical form that involves discrete and continuous variable. In this model, generalized disjunctive programming (GDP) are being modeled

c. Solution of the corresponding mathematical form, i.e the optimization model for which the optimal topology is determined, in which to solve GDP formulation using LOGMIP solver and MILP by using Cplex solver within the GAMS modeling language.

A diagrammatic description of methodology is shown in Figure 3.2. Project milestone and gantt chart are shown in Table 3.5 and Table 3.6.

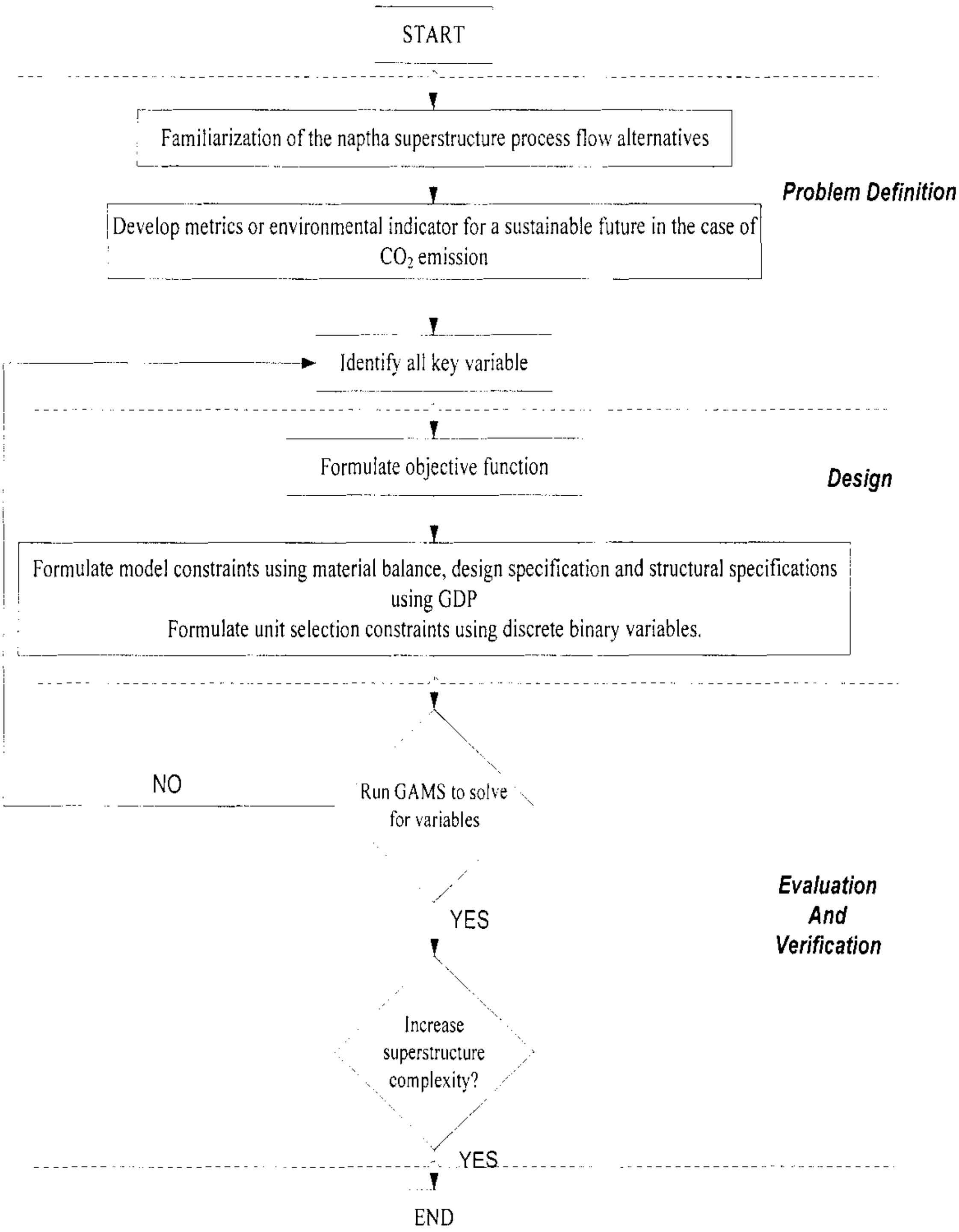

Figure 3.2: Flow chart of the proposed methodology to carry out the thesis research

3.4 Gantt Chart

Table 3.5: Gantt Chart for FYP I

Details/Week	**FYP I**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**10**	**11**	**12**	**13**	**14**
Problem identification	√	√													
Literature review		√	√												
Development of design and structural logical constraints				√	√	√									
Formulating objective function						√	√								
MILP and GDP formulation								√	√		√	√			
Solve MILP optimization model using GAMS												√	√	√	
Submission of interim report and oral presentation														√	√

Table 3.6: Gantt Chart for FYP II

Details/Week	**FYP II**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**10**	**11**	**12**	**13**	**14**
Solve GDP model in GAMS/LOGMIP	√	√	√	√											
Solve MILP optimization model using GAMS				√	√	√									
Result comparison between MILP and GDP							√								
Result verification								√	√						
PreEDX poster presentation											√				
Interim report submission												√	√	√	
Final oral presentation and hardbound submission															√

3.5 Computational Tools

LogMIP 1.0 is a program for solving linear and nonlinear disjunctive programming problems involving binary variables and disjunction definitions for modeling discrete coices. While modeling and solution of these disjunctive optimization problems has not ye reached the stage of maturity and reliability as LP, MIP and NLP modeling, these problems have a rich arean of applications. LogMIP is composed of: (a) a language compiler for the declaration and definition of disjunctions and logic constraints and (b) solvers for linear and non-linear disjunctive models. Those components are linked to GAMS. Both parts are supersets of GAMS language and solvers respectively. LogMIP is not independent of GAMS. Besides the disjunction and logic constraints declaration and definition, LogMIP needs the declaration and definitions of scalars, sets, tables, variables, constraints, equations, etc. made in GAMS language for the specifications and soution of a disjunctive problem.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Computational Experience

LogMIP greatly facilitates the task of posing a discrete model through the use of disjunctions. For instance, the error messages given by the language compiler helps in writing the model when it is not formulated according to its syntactic and semantic rules. The major difficulties faced during writing the model is when defining multiple models to be solved in LogMIP. In this case, the model consists of the possibilities of processing light naphtha and heavy naphtha which are characterized by the API gravity. If the API ≤ 33, therefore it is light naptha and the solution is different from heavy naphtha. Hence, all constraints must be declared for all the multiple models, including constraints that are included in the disjunctions as well as "dummy" constraint. The "dummy" constraints ensures that the binary variables that handle the disjunctions in the GDP formulation are not eliminated from the model in the solution.

Besides that, by writing discrete decisions by means of disjunctions make the model more clear and easy to understand. For example, a discreate variable of selecting or not selecting atmospheric distillation unit (ADU) can be written in terms of disjunction as shown as below:

$$\begin{bmatrix} Y_{\mathrm{ADU}} \\ \underbrace{(0.2088)}_{A} f_{\mathrm{CR}} = \underbrace{f_{\mathrm{NAP1}}}_{b} \\ (0.0555) f_{\mathrm{CR}} = f_{\mathrm{LSRN1}} \\ (0.1533) f_{\mathrm{CR}} = f_{\mathrm{HSRN1}} \\ f_{\mathrm{NAP1}} = f_{\mathrm{LSRN1}} + f_{\mathrm{HSRN1}} \\ c_{\mathrm{ADU}} = 228 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{ADU}} \\ f_{\mathrm{CR}} = 0 \\ f_{\mathrm{NAP1}} = 0 \\ f_{\mathrm{LSRN1}} = 0 \\ f_{\mathrm{HSRN1}} = 0 \\ c_{\mathrm{ADU}} = 0 \end{bmatrix} \ldots (P)$$

The model and computational statistic are reported in Table 4.1. The GDP model is solved by using GAMS/CPLEX solver which will then automatically perform the transformation into mixed-integer programs by using convex-hull reformulation.

Table 4.1: Model and computational statistics (GDP)

Solver	CPLEX
Number of single equations	195
Number of binary variables	22
Number of continuous variables	95
Number of iterations	23
CPU time/resource usage	0.030s

4.2 Base Data

For this research, two design scenarios are considered in the computational experiments which are namely as light crude charge processing (API >33) and heavy crude charge processing (API ≤33). The assumptions made are as follows;

a. Refinery operates 333 days per year
b. Crude charge is fixed to be between 10000000 bbl/day to 50000000 bbl/day
c. Gasoline requirement of at least 700 000 kg/day
d. Total capacity investment = fixed capital investment + working capital
= total equipment base cost + working capital
e. Total operating cost = fixed operating costs + variable operating costs + general expenses
f. Total cost (objective function) = total capital investment + total operating cost

The Nelson-Farrar Refinery Construction Index (NFRCI) (Maples, 2000, p.388; EU-OPEC Rountable on Energy Policies, 2008) are:

a. Jan 1991 : 1241.7
b. Dec 2008 : 2067.2

There are few values that need to be determined by the user. Those values are constant or known as scalar in GAMS language. For example, the user determined the value of API gravity, the crude oil cost and also the cost of purchased naphtha. The cost of utilities per unit is shown in Table 4.2 and Table 4.3 is the base cost and utilities consumption of major unit operations (Maples, 2000, p.386).

Table 4.2: Utilities cost per unit (www.mida.gov.my/2008)

Utilities Cost per unit (RM/kW)	
Electricity	0.1980
Fuel	0.1018
HP Steam	0.0050
Cooling water (CW)	0.8400

Table 4.3: Base cost and utilities consumption of major unit operations (Maples, 2000, p.386)

	Jan '91 (mil RM)	**Dec '08 (mil RM)**	**Electricity (MWh/kg)**	**Fuel (kJ/kg)**	**Steam (kJ/kg)**	**CW (m^3/kg)**
ADU	137	248	0.0039	0.0826	0.0888	0.0000
VIS	86	144	0.0039	0.0660	0.1776	0.0000
COK	166	276	0.0282	0.0991	0.1421	0.0000
FCC	310	515	0.0078	0.0660	0.0710	0.0119
HCR	342	569	0.1402	0.2766	0.0000	0.0000
HDT	58	96	0.0157	0.0248	0.0533	0.0000
REF	162	270	0.0078	0.2477	0.1421	0.0030
ISO	25	42	0.0078	0.0083	0.1279	0.0000
SRU (per tone)	18	30	0.3132	0.0000	2.6636	0.1482

4.3 Computational Results for GDP Model

The results for these model is based on the objective function that have been formulated in GDP. The total feed flowrate from external sources of naphta from visbraker (VIS), coker (COK), catalytic cracker (FCC) and hydrocracker (HCR) varies dependent on light or heavy crude are constant at 200000 kg/d. Table 4.4 shows the results of the objective function which is the total cost of naptha processing for light and heavy crude. Meanwhile, the results on the existance of the process units are shown in Table 4.5 and it s also shown in Figure 4.1 and 4.2 the flowrates of all streams are available in Table 4.6 and Table 4.7.

Table 4.4: Computational result of objective function for GDP

	Heavy Crude	**Light Crude**
CAPEX + OPEX + Raw material (*mil RM*)	2453.060	2464.530

Note: assuming 330 working days

Table 4.5: Computational results on the existance for GDP

Heavy Crude		**Light Crude**	
Process Units	**Binary variables**	**Process Units**	**Binary variables**
ADUu	1	ADUu	1
BLNDu	1	BLNDu	1
COKu	1	COKu	0
FCCu	1	FCCu	1
FGHu	1	FGHu	1
HCRu	1	HCRu	0
HDT1u	1	HDT1u	1
HDT2u	0	HDT2u	0
ISOu	1	ISOu	1
LPGu	1	LPGu	1
MIX1u	1	MIX1u	1
MIX2u	0	MIX2u	0
MIX3u	1	MIX3u	1
MIX4u	0	MIX4u	0
MIX5u	1	MIX5u	1
MIX6u	1	MIX6u	1
REFu	1	REFu	1
SPLT1u	1	SPLT1u	1
SPLT2u	1	SPLT2u	1
SOLDu	1	SOLDu	1
SRUu	1	SRUu	1
VISu	0	VISu	1

Table 4.6: Computational results on the stream flow rates for GDP (heavy crude)

Stream	Flowrate (kg/d)	Stream	Flow rate(kg/d)	Stream	Flow rate (kg/d)
COK_1	2000000.000	HCR_4	0	NAP1	0
COK_2	0	HSRN1	5029419.880	NAP2	0
FCC_1	2000000.000	HSRN2	1.102942E+7	NAP3	0
FCC_2	0	HSRN3	0	NAP4	2.311699E+7
FG1	128283.880	HSRN4	0	NAP5	2.311699E+7
FG2	0	HSRN5	0	PCHN1_1	0
FG3	855328.670	ISO	1622357.000	PCHN1_2	0
FG4	16387.440	LPG1	68261.150	PCHN2	0
FG5	1000000.000	LPG2	1803125.310	PCHN3_1	0
GSLN	2.134115E+7	LPG3	0	PCHN3_2	0
H2	739743.720	LPG4	1871386.460	REF	1.971879E+7
H2_1	739743.720	LPG5	1871386.460	S	11973.480
H2_2	0	LSRN1	1820827.160	SOLD	2.340659E+7
H2S1	2149.520	LSRN2	0	TG	2149.520
H2S2	0	LSRN3	0	VIS_1	0
HCR_1	2000000.000	LSRN4	1820827.160	VIS_2	0
HCR_2	0	LSRN5	1638744.440	CR	3.280770E+7
HCR_3	0	LSRN6	182082.720		

Table 4.7: Computational results on the stream flow rates for GDP (light crude)

Stream	Flowrate (kg/d)	Stream	Flow rate(kg/d)	Stream	Flow rate (kg/d)
COK_1	0	HCR_4	0	NAP1	0
COK_2	0	HSRN1	6958923.390	NAP2	0
FCC_1	2000000.000	HSRN2	1.095892E+7	NAP3	0
FCC_2	0	HSRN3	0	NAP4	2.296923E+7
FG1	127463.930	HSRN4	0	NAP5	2.296923E+7
FG2	0	HSRN5	0	PCHN1_1	0
FG3	849861.690	ISO	2244763.480	PCHN1_2	0
FG4	22674.380	LPG1	67824.850	PCHN2	0
FG5	1000000.000	LPG2	1791600.310	PCHN3_1	0
GSLN	2.183752E+7	LPG3	0	PCHN3_2	0
H2	735015.510	LPG4	1859425.160	REF	1.959276E+7
H2_1	735015.510	LPG5	1859425.160	S	11896.950
H2_2		LSRN1	2519375.400	SOLD	2.396078E+7
H2S1	14032.730	LSRN2	0	TG	2135.780
H2S2	0	LSRN3	0	VIS_1	2000000.000
HCR_1	0	LSRN4	2519375.400	VIS_2	0
HCR_2	0	LSRN5	2267437.860	CR	4.539415E+7
HCR_3	0	LSRN6	251937.540		0

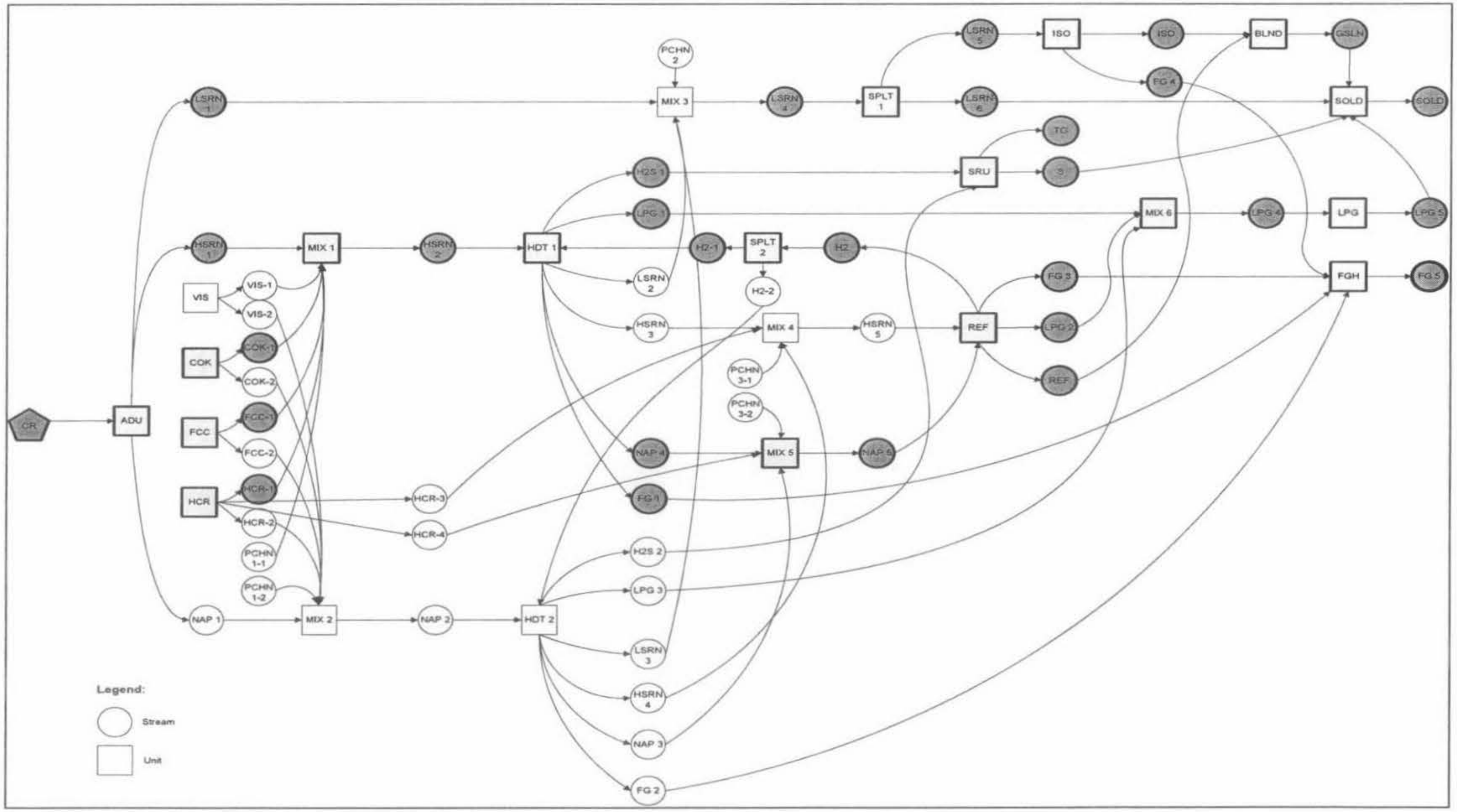

Figure 4.1: Optimal superstructure topology naphtha produced from ADU with heavy crude charge

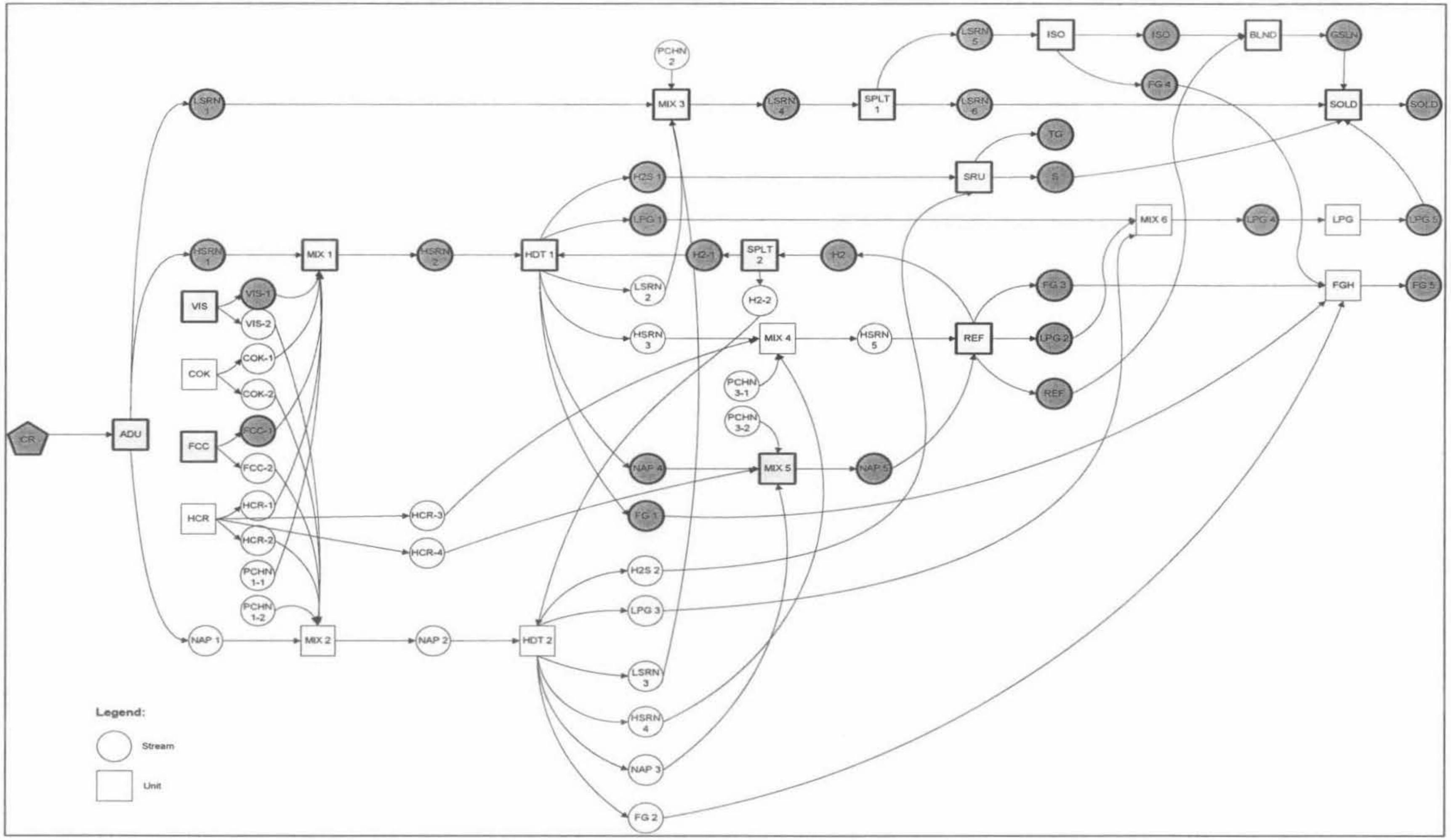

Figure 4.2: Optimal superstructure topology naphtha produced from ADU with light crude charge

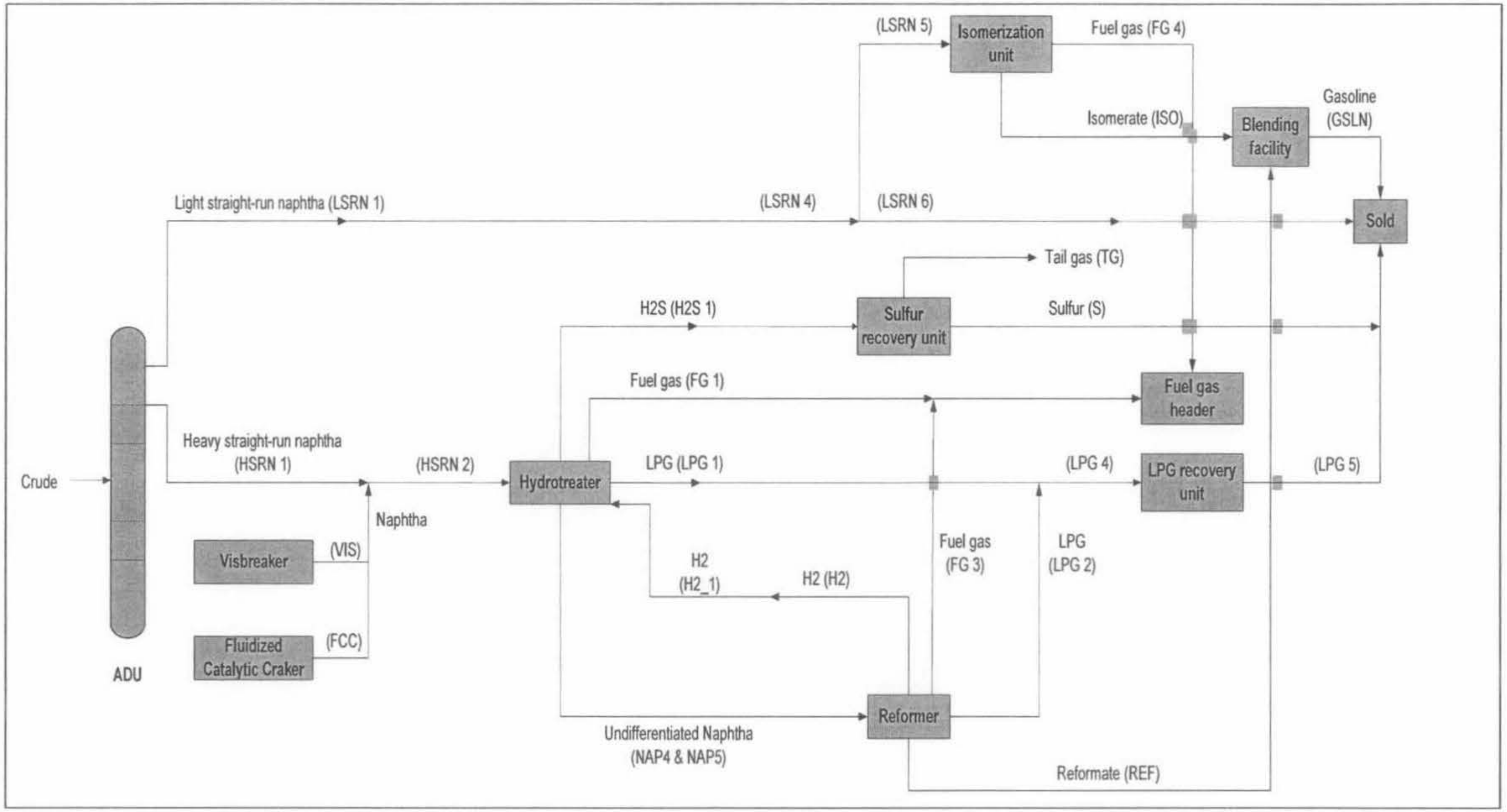

Figure 4.3: Optimal topology naphtha produced from ADU with light crude charge

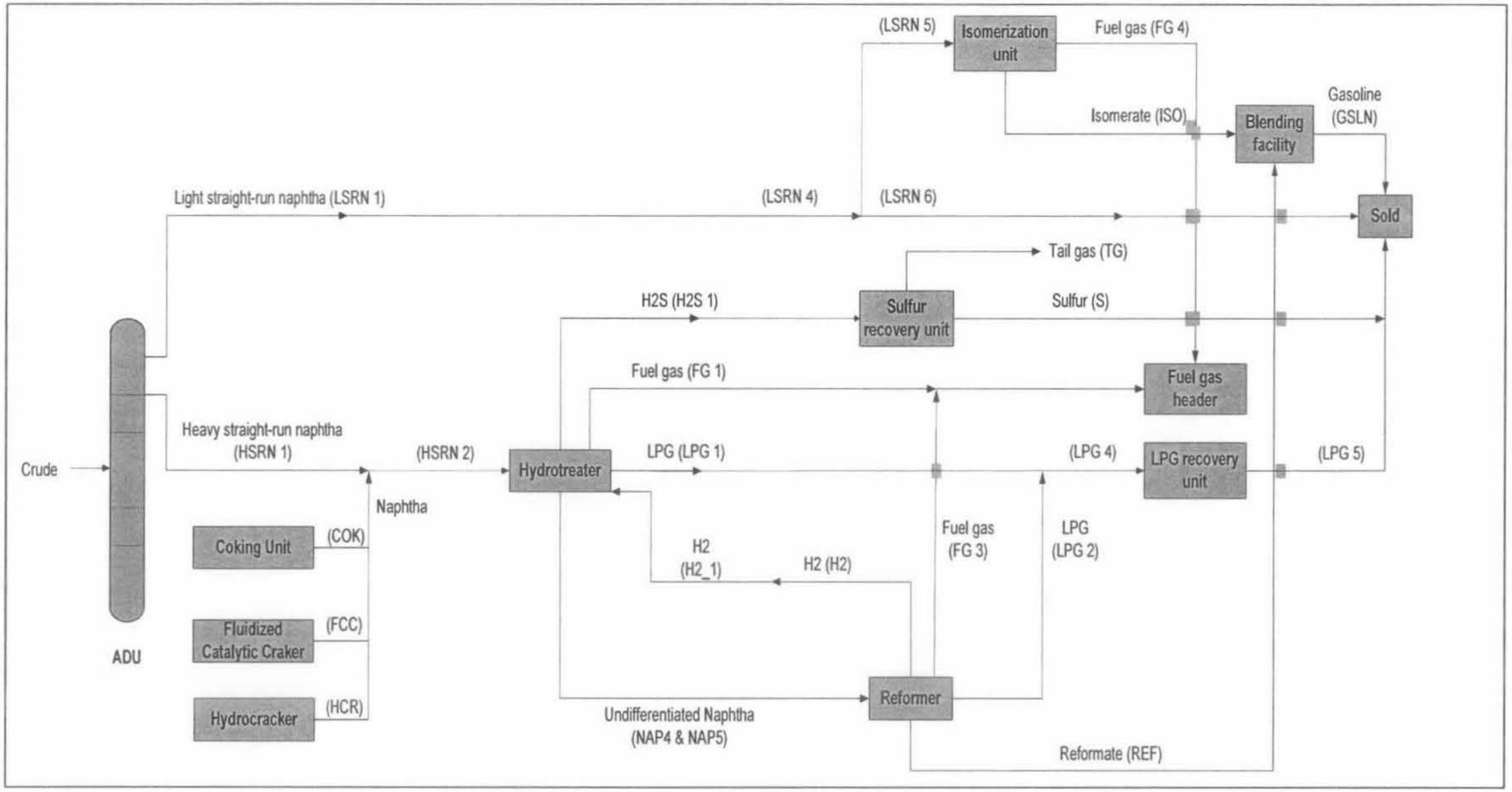

Figure 4.4: Optimal topology naphtha produced from ADU with heavy crude charge

4.4 Discussion on Computational Experience and Numerical Results

Based on the results, the total cost (objective function) of heavy crude is slighty lower than the total cost of light crude charge processing. Heavy crude charge processing requires less amount of crude oil fed into the distillation unit as based on Table 4.6 and Table 4.7. From the optimal solution, both light and heavy crude oil processing are separated into light naphtha and heavy naphtha, which agrees with real life practical features. If the production demand requirements are reduced, there is a possibility of not using the external sources of naphtha such as from the visbreaker (VIS), fluid catalytic cracker (FCC), coker (COK), and hydrocracker (HCR). However, external naphtha sources and also purchased naphtha are required to meet higher production demand requirements.

To ensure that certain design specifications are obeyed especially in terms of selecting the external sources, logical constraints are used. (The constraints are available in the GAMS code in Appendix B.) The use of LogMIP obviates the need to reformulate the logic propositions and the overall disjunctive problem into algebraic representations, hence reducing the time involved in the typically time-consuming problem formulation. LogMIP typically leads to less computational time and number of iterations in its computational effort because the associated GDP formulation involves less equations and variables compared to MILP. From the computational experiments, it is found that logical constraints of design specifications and structural specifications play an important role to determine the optimal selection of process units and streams.

The linear GDP model can be transformed into a mixed-integer linear programming (MILP) algebraic model by using either the reformulation approach of big-M relaxation or convex hull relaxation. The process of transforming and solving within the GAMS platform is done systematically without user intervention.

Although the superior advantages of the convex hull relaxation method is theoretically acknowledged, we have also attempted to apply the big-M reformulation approach but this produces an infeasible solution. This is likely due to unsuitable values of the big-M constants to relate the continuous and discrete variables. It is worth noting that in general, no revision is made on the problem formulation in terms of variables and constraints when applying both the convex hull reformulation and the big-M relaxation methods. It is generally known that the larger the value of M, the poorer the relaxation and hence the result produced would not be accurate. The formulation can be improved by adding or eliminating constraits that can accelerate or slow-down the problem solution respectively.

4.5 Validation of optimal configuration obtained

The optimal naphtha processing configuration obtained the solution given by the model are showed in Figure 4.3 and Figure 4.4 for both light and heavy crude. In the optimal topology, the naphtha outlets of ADU are separated into light naphtha and heavy naphtha streams, as is the typical configuration based on Al-Qahtani and Elkamel (2008) and Favennec (2001). As suggested in Maples (2000, p. 90), the light naptha is sent for gasoline blending or to the isomerization unit while heavy naptha is sent for hydrotreating before it is processed in the reformer. The optimal topology is compared against the process flow diagram of an ADU taken from Maples (2000, p. 91).

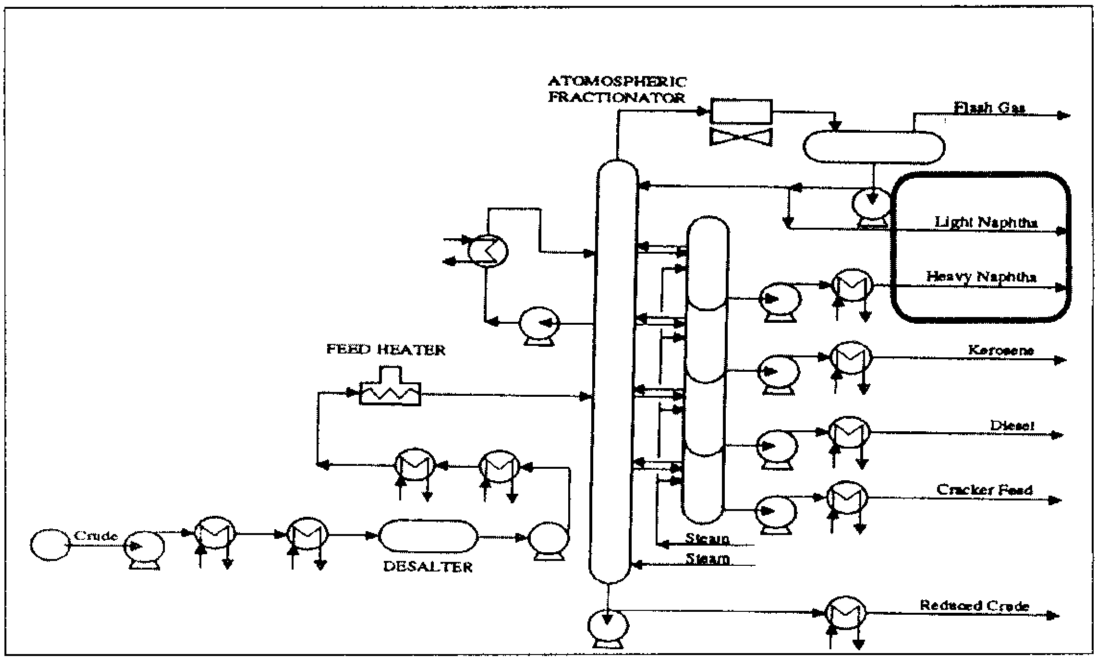

Figure 4.5: Process Flow Diagram of an atmospheric distillation unit from Maples (2000, p.91)

The outlet from the naphtha hydrotreater in the optimal topology generated by our proposed model is an undifferentiated naphtha stream, which is consistent with the configuration reported Parkash (2003, p. 35). The undifferentiated naphtha that has been dehydrodesulfurized is then sent for processing in a catalytic reformer, which is present in our optimal topology. We note that there are also configurations that consider splitting of the naphtha hydrotreater outlet to light and naphtha streams, for instance, in Maples (2000, p. 254).

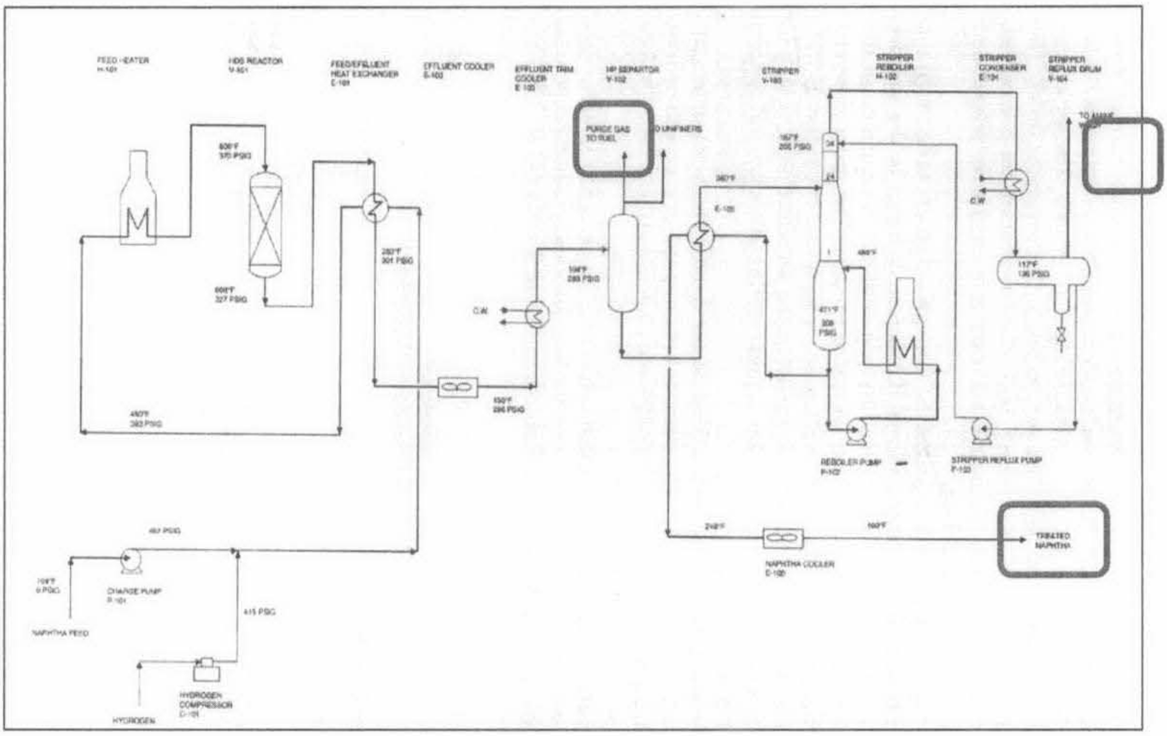

Figure 4.6: Process flow diagram of a naphtha hydrotreater from Parkash (2003) p. 35

It is not uncommon for a light naphtha stream to be sent to an isomerization unit to produce isomerate for gasoline blending, as represented by the stream denoted as LSRN5 in the selected optimal topology of our case (Al-Qahtani and Elkamel, 2008). Besides that, it is also possible for a light naphtha stream to be sold directly, as is the case with the LSRN6 stream in the optimal topology, possibly to a petrochemical plant that consumes lighter petroleum refining feedstocks and has lower light petroleum product availability. Such a petrochemical plant typically uses steam cracking of light naphtha to obtain the main petrochemical building blocks for downstream processing that chiefly produces ethylene and chemical grade propylene (Al-Qahtani et al., 2008).

Hydrocracker, visbraker, coker and catalytic cracker are identified as external sources because the feed for each of the units are from vacuum distillation unit. The external sources required depends on gasoline production demand. Generally, the optimal overall configuration for naphtha processing generated from our model agrees well with the optimal configuration reported by Agilent Technologies as shown in Figure 2.1.

4.6 Model validation

In order to validate the consistency of the model, a few cases have been created. The first case would be the current model where crude rate is need to be determined (variable crude) with fixed production requirement. The total cost for both heavy and light crude is examined. The crude rate for the first case shows the optimum flow rate for both light and heavy crude. For the second case, the crude rate is fixed and is relatively smaller compare to the optimal rate obtained in case 1. The total cost for light crude increase substantially because a very large amount of naptha needed to support the production requirement which directly lead to higher cost. Eventhough the same goes for heavy crude, the total cost is slightly higher compare to case 1 because the less amount of external sources from the hydrocracker (or stream HCRs_4) needed. The third case is examined, in which the crude is still fixed with a larger value, approximately nearer to the value of optimal flow rate for light crude. The total cost for light crude in case 3 is lesser compare to total cost in case 2 due to less purchase naptha is required to support the demand. However, the cost is still substantially greater compare to the optimum cost. And for heavy crude, the cost is higher for both case 1 and case 2 maybe due to the increase in cost of purchasing the raw material itself. Finally, for the fourth case, the crude fixed in greater than the optimal rate for both light and heavy crude. Heavy crude has higher total cost rather than light crude generally due to greater amount of heavy crude required compare to the optimal flow rate. The cost for raw material increase which lead to higher total cost.

Base on the four cases which have been discussed, the model has proves to give a minimum total cost with optimal flow rate of crude. Although the amount of crude has been reduced or increased significantly, it still did not produce a minimum total cost. This validate that the model with the objective function of minimizing the cost and determining the optimal flow rates is consistent and able to operate in different requirement.

Table 4.8: Optimal solution of variable crude oil processing rate for fixed production requirements

	Optimal Crude Rate	**Optimal Total Cost (RM)**
Light Crude Processing	4.539415E+7	2464.53
Heavy Crude Processing	3.280770E+7	2453.06

Note: Light crude processing incurs higher cost than heavy crude processing because this case study mainly considers processing of lighter components in an atmospheric distillation unit, which involve processing of most components of a light crude oil while only a fraction of the components of a heavy crude oil are considered. In other words, processing of most of the components of a heavy crude oil are not considered in this model.

Table 4.9: Case 1. Fixed crude rate lower than optimal solution for both light and heavy crude processing with the same fixed production requirements

	Crude Rate	**Total Cost**	**Cost of purchasing external sources of naptha**	**Remarks**
Light crude processing	3.00E+ 7 (< optimal crude rate)	1 668 695.85 (> optimal total cost)	2 446 145.74 (PCHN1_1s)	Higher cost for light crude processing because: (1)
Heavy crude processing	3.00E+7 (< optimal crude rate)	2458.48 (> optimal total cost)	2.00E+5 (HCR_4)	Although lower crude processing rate should incur less total cost, total cost is still higher due to higher flows in other streams

Table 4.10: Case 2: Fixed crude rate lower and higher than optimal solution for light and heavy crude processing, respectively with the same fixed production requirements

	Crude Rate	**Total Cost**	**External sources of naptha**
Light crude processing with fixed crude rate *lower* than optimal solution	4.00E+7 (< optimal crude rate)	586 316.380	857135.930 (PCHN1_1)
Heavy crude processing with fixed crude rate *higher* than optimal solution	4.00E+7 (> optimal crude rate)	2463.640	-

Table 4.11: Case 3. Fixed crude rate higher than optimal solution for both light and heavy crude processing with the same fixed production requirements

	Crude Rate	**Total Cost**	**External sources of naptha**
Light Crude	4.80E+7 (> optimal crude rate)	2475.220	-
Heavy Crude	4.80E+7 (> optimal crude rate)	2482.270	-

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the objectives of this project have been achieved. Firstly, the development and familiarization of naphtha processing superstructure representation. Then, followed by the formulation of generalized disjunctive programming (GDP) to incorporate both continuous and discrete variables. The continuous variables are the optimum flow rates of the streams and meanwhile the discrete variables are the selection of the process units. The constraints involved in the formulation are the material balances, structural specifications and also design specifications which greatly influence in the decision making of the continuous and discrete variables. The disjunctive formulation had provide an easy and compact representation and visualization of the discrete choices. The GDP formulation then are modeled in GAMS/LogMIP via convex hull relaxation. The optimal configuration obtained is parallel with the real operating refineries. It has been proven that the mathematical model had successfully achieved the minimum total cost (objective function) with optimal flow rate.

5.2 Recommendation

For future work, the model should be more focus on sustainability development of environmental consideration. Thorough studies should be done in terms the emission factors of the process units and how does the formulation of the emission of carbon dioxide, CO_2 to be included in the model. An economic analysis or sensitivity analysis of the model can be analyse in order to cover every aspect of refinery issue. Besides that, the introduction of nonlinearity in the model formulation that takes account the energy balance can lead to a better accuracy to the results. The GDP via big-M can also be done to compare the computational time and other statistical data with convex hull relaxation.

REFERENCES

Gary, J. H and Handwork, G.E. 2001. *Petroleum Refinery: Technology and Economics.* New York : Marcel Dekker

Maple, R.E. 2000. Petroleum Refinery Process Economics. Oklohama: Pennwell

Joseph, K. 2004. *Modeling Languages in Mathematical Optimization*: Kluwer Academic Publisher.

Parkash, S. 2003. *Refining Processes Handbook.* Burlington, Massachussets: Elsvier.

Metin Turkay & Ignacio E. Grossmann.1995. Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Network. *Computers & Chemical Engineering*, 20: 960-978.

Metin Turkay and Ignacio E. Grossmann.1996. Disjunctive Programming Techniques for the Optimization of Process System with Discontinuous Investment Costs – Multiple Size Regions. Integration of logic and heuristic knowledge in MINLP optimization for process synthesis. *Ind. Eng. Chem. Res. Computers*, 35: 2611 – 2623.

Raman, R. and I. E. Grossmann.1991. Relation between MILP modeling and logical inference for chemical process synthesis. *Computers & Chemical Engineering*, 15, 73.

Raman, Ramesh and Ignacio E. Grossmann. 1993. Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Computers & Chemical Engineering* 17, 9: 909–927.

Azapagic, A. and Perdan, S., 1999, Indicators of sustainable development for industry: a general framework, *Trans IChemE, Part B, Proc Safe Env Prot*, 78:243.

Institution of Chemical Engineers, 2002, *Sustainable Development Progress Metrics Recommended for use in the Process Industries* (IChemE, Rugby, UK).

Elkamel, A, Ba-Shammakh, M, Douglas, P. and Croiset, E., 2008, An Optimization Approach for Integrating Planning and CO_2 Emission Reduction on the Petroleum Refinery Industry, *Ind. Eng. Chem. Res*, 47: 760-776.

International Petroleum Industry Environmetal Conservation Association (IPIECA), 2003. *Chap 5: Identification of industry GHG emissions,* 5:1-5

Al-Qahtani, K. and A. Elkamel. Multisite facility network integration design and coordination: an application to the refining industry. *Computers and Chemical Engineering* 32 (2008a) 2189–2202.

Favennec Jean-Pierre (editor). *Refinery Operation and Management.* Volume 5 of Petroleum Refining Series, Chapter 6. Paris, France: Editions Technip, 2001, p. 218.

APPENDICES

APPENDIX A: Reformulation of GDP to MILP via convex hull

$$\begin{bmatrix} Y_{\text{ADU}} \\ \underbrace{(0.2088)}_{A} f_{\text{CR}} = \underbrace{f_{\text{NAP1}}}_{b} \\ (0.0555) f_{\text{CR}} = f_{\text{LSRN1}} \\ (0.1533) f_{\text{CR}} = f_{\text{HSRN1}} \\ f_{\text{NAP1}} = f_{\text{LSRN1}} + f_{\text{HSRN1}} \\ c_{\text{ADU}} = 228 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\text{ADU}} \\ f_{\text{CR}} = 0 \\ f_{\text{NAP1}} = 0 \\ f_{\text{LSRN1}} = 0 \\ f_{\text{HSRN1}} = 0 \\ c_{\text{ADU}} = 0 \end{bmatrix}$$

continuous variables: $f_{\text{CR}}, f_{\text{NAP1}}, f_{\text{LSRN1}}, f_{\text{HSRN1}}$

1. disaggregate continuous variables:

$f_{\text{CR}} = x_1 + x_2$

$f_{\text{NAP1}} = w_1 + w_2$

$f_{\text{LSRN1}} = z_1 + z_2$

$f_{\text{HSRN1}} = v_1 + v_2$

Step 2.

for $\underbrace{(0.2088)}_{A} f_{\text{CR}} = \underbrace{f_{\text{NAP1}}}_{b}$, reformulation: $(0.2088)x_1 = w_1$

for $(0.0555) f_{\text{CR}} = f_{\text{LSRN1}}$, reformulation: $(0.0555)x_1 = z_1$

for $(0.1533) f_{\text{CR}} = f_{\text{HSRN1}}$, reformulation: $(0.1533)x_1 = v_1$

for $f_{\text{NAP1}} = f_{\text{LSRN1}} + f_{\text{HSRN1}}$, reformulation: $w_1 = z_1 + v_1$

for $c_{\text{ADU}} = 228y_1$

$$\begin{bmatrix} Y_{\text{ADU}} \\ \underbrace{(0.2088)}_{A} f_{\text{CR}} = \underbrace{f_{\text{NAP1}}}_{b} \\ (0.0555) f_{\text{CR}} = f_{\text{LSRN1}} \\ (0.1533) f_{\text{CR}} = f_{\text{HSRN1}} \\ f_{\text{NAP1}} = f_{\text{LSRN1}} + f_{\text{HSRN1}} \\ c_{\text{ADU}} = 228 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\text{ADU}} \\ f_{\text{CR}} = 0 \\ f_{\text{NAP1}} = 0 \\ f_{\text{LSRN1}} = 0 \\ f_{\text{HSRN1}} = 0 \\ c_{\text{ADU}} = 0 \end{bmatrix}$$

for $f_{\text{CR}} = 0 : x_2 = 0y_2 \Rightarrow x_2 = 0(1 - y_1) = 0$

for $f_{\text{NAP1}} = 0 : w_2 = 0y_2 = 0$

for $f_{\text{LSRN1}} = 0 : z_2 = 0y_2 = 0$

for $f_{\text{HSRN1}} = 0 : v_2 = 0y_2 = 0$

Step 3: $y_1 + y_2 = 1$

So, from step 1:

$f_{\text{CR}} = x_1$

$f_{\text{NAP1}} = w_1$

$f_{\text{LSRN1}} = z_1$

$f_{\text{HSRN1}} = v_1$

Step 3:

$y_1 + y_2 = 1$

Step 4: upper bound on continuous variables

(where subscript "U" denotes upper bound on the corresponding continuous variable)

$x_1 \le x_1^u y_1$

$w_1 \le w_1^u y_1$

$z_1 \le z_1^u y_1$

$v_1 \le v_1^u y_1$

Finally. the mixed-integer constraints for MIP that equivalent to GDP given by:

$(0.2088) f_{CR} = f_{NAP1}$

$(0.0555) f_{CR} = f_{LSRN1}$

$(0.1533) f_{CR} = f_{HSRN1}$

$f_{NAP1} = f_{LSRN1} + f_{HSRN1}$

$c_{ADU} = 228y_1$

$y_1 + y_2 = 1$

$f_{CR} \le f_{CR}^{U} y_1$

$f_{NAP1} \le f_{NAP1}^{U} y_1$

$f_{NAP1} \le f_{NAP1}^{U} y_1$

$f_{LSRN1} \le f_{LSRN1}^{1} y_1$

$f_{HSRN1} \le f_{HSRN1}^{1} y_1$

APPENDIX B: GDP GAMS Code

```
SETS

I     set of process units (tasks)
/
1   atmospheric distillation unit     ADUu
2   blending unit                     BLNDu
3   coker                             COKu
4   catalytic cracker                 FCCu
5   fuel gas hydrotreater             FGHu
6   hydrocracker                      HCRu
7   hydrotreater 1                    HDT1u
8   hydrotreater 2                    HDT2u
9   isomerization unit                ISOu
10  LPG recovery unit                 LPGu
11  mixer 1                           MIX1u
12  mixer 2                           MIX2u
13  mixer 3                           MIX3u
14  mixer 4                           MIX4u
15  mixer 5                           MIX5u
16  mixer 6                           MIX6u
17  reformate                         REFu
18  spliter 1                         SPLT1u
19  spliter 2                         SPLT2u
20  sulfur recovery unit              SRUu
21  sold unit                         SOLDu
22  visbreaker                        VISu
/

J     set of process streams (states)
/
COK_1s
COK_2s
CRs
FCC_1s
FCC_2s
FG1s
FG2s
FG3s
FG4s
FG5s
GSLNs
H2s
H2_1s
H2_2s
H2S1s
H2S2s
HCR_1s
HCR_2s
HCR_3s
HCR_4s
HSRN1s
HSRN2s
HSRN3s
HSRN4s
HSRN5s
ISOs
LPG1s
LPG2s
LPG3s
LPG4s
LPG5s
LSRN1s
LSRN2s
LSRN3s
LSRN4s
LSRN5s
LSRN6s
NAP1s
NAP2s
NAP3s
NAP4s
NAP5s
PCHN1_1s
PCHN1_2s
PCHN2s
PCHN3_1s
PCHN3_2s
REFs
Ss
SOLDs
TGs
VIS_1s
VIS_2s
/

MIXER(I)
/
11        MIX1u
12        MIX2u
13        MIX3u
14        MIX4u
15        MIX5u
16        MIX6u
/

SPLITTER(I)
/
18        SPLT1u
19        SPLT2u
/


OUTLET_MIXER(MIXER,J)
/
11.HSRN2s
12.NAP2s
13.LSRN4s
14.HSRN5s
15.NAP5s
16.LPG4s
```

```
/

ALIAS(J,J1);

PARAMETERS

CAPEX_M(MIXER)  capital cost of mixers
CAPEX_S(SPLITTER)  capital cost of splitters
;
CAPEX_M(MIXER) = 100;
CAPEX_S(SPLITTER) = 100;

SCALARS
cr_cst          crude oil cost (RM per bbl) /120/
cr_kg_per_bbl    crude oil amount (kg per bbl)
/127.7/
pchn_cst        purchased naphtha cost (RM per
kg) /0.524/
API             API gravity of crude charge /40/
EImax            /100/
;

POSITIVE VARIABLES
F(J)          stream flowrates
*EI(I)  emission flowrates
;

BINARY VARIABLES
Y(I)
;

FREE VARIABLES
TC            total cost of refinery
c(I)          cost of equipment
;

Equations
objfn         min total cost in (mil RM)

*------------------------------------------------------
------------------------------------------------------
--
*MATERIAL BALANCES - Definitions of
equations independent of discrete choices

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5
mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

*HEAVY CRUDE (API<33)
mat_bal20
mat_bal21
mat_bal22
mat_bal23
mat_bal24
mat_bal25
mat_bal26
mat_bal27

*LIGHT CRUDE (API>=33)
mat_bal29
mat_bal30
mat_bal31
mat_bal32
mat_bal33
mat_bal34
mat_bal35
mat_bal36

*MAT_BAL_MIXER

DUMMY

*------------------------------------------------------
------------------------------------------------------
------
*RAW MATERIAL AVAILABILIY
prodreq1
prodreq2
*PRODUCTION REQUIREMENTS
prodreq3
prodreq4
prodreq5
prodreq6
prodreq7
prodreq8
prodreq9

*------------------------------------------------------
------------------------------------------------------
---
*DEFINITION OF DISJUNCTION'S
EQUATIONS
*ADU
*INOUT1_1,
```

INOUT1_2,INOUT1_3,INOUT1_4,INOUT1_5,
INOUT1_6,INOUT1_7
COST1_1

*HDT-1
INOUT2_1,INOUT2_2,INOUT2_3,INOUT2_7,
INOUT2_8,INOUT2_9,INOUT2_10,INOUT2_
11,INOUT2_12,INOUT2_13,INOUT2_14,INO
UT2_15
*INOUT2_4,INOUT2_6, INOUT2_5,
COST2_1

*HDT-2
INOUT3_1,INOUT3_2,INOUT3_3,INOUT3_4,
INOUT3_5,INOUT3_6,INOUT3_7,INOUT3_8,
INOUT3_9,INOUT3_10,INOUT3_11,INOUT3
_12,INOUT3_13,INOUT3_14
COST3_1

*ISO
INOUT4_1,INOUT4_2,INOUT4_3,INOUT4_4,
INOUT4_5
COST4_1

*SRU
INOUT5_1,INOUT5_2,INOUT5_3,INOUT5_4,
INOUT5_5,INOUT5_6
COST5_1

*REF
*INOUT6
INOUT6_1,INOUT6_2,INOUT6_3,INOUT6_4,
INOUT6_5,INOUT6_6,INOUT6_7,INOUT6_8,
INOUT6_9,INOUT6_10
COST6_1

*SOLD
INOUT7_1,INOUT7_2,INOUT7_3,INOUT7_4,
INOUT7_5 ,INOUT7_6
COST7_1

*BLND
INOUT8_1,INOUT8_2,INOUT8_3,INOUT8_4
COST8_1

*LPG
INOUT9_1,INOUT9_2,INOUT9_3
COST9_1

*FG
INOUT10_1,INOUT10_2,INOUT10_3,INOUT
10_4,INOUT10_5,INOUT10_6
COST10_1

*SPLT 1
INOUT11_1,INOUT11_2,INOUT11_3,INOUT
11_4,INOUT11_5
COST11_1

*SPLT 2
INOUT12_1,INOUT12_2,INOUT12_3,INOUT
12_4
COST12_1

*MIX-1
INOUT13_1,INOUT13_2,INOUT13_3,INOUT
13_4,INOUT13_5,INOUT13_6,INOUT13_7,IN
OUT13_8
COST13_1

*MIX-2
INOUT14_1,INOUT14_2,INOUT14_3,INOUT
14_4,INOUT14_5,INOUT14_6,INOUT14_7,IN
OUT14_8
COST14_1

*MIX-3
INOUT15_1,INOUT15_2,INOUT15_3,INOUT
15_4,INOUT15_5,INOUT15_6
COST15_1

*MIX-4
INOUT16_1,INOUT16_2,INOUT16_3,INOUT
16_4,INOUT16_5,INOUT16_6
COST16_1

*MIX-5
INOUT17_1,INOUT17_2,INOUT17_3,INOUT
17_4,INOUT17_5,INOUT17_6
COST17_1

*MIX-6
INOUT18_1,INOUT18_2,INOUT18_3,INOUT
18_4,INOUT18_5
COST18_1

*COK
INOUT19_1, INOUT19_2, INOUT19_3,
INOUT19_4

*FCC
INOUT20_1, INOUT20_2,INOUT20_3,
INOUT20_4

*HCR
INOUT21_1, INOUT21_2,INOUT21_3,
INOUT21_4, INOUT21_5, INOUT21_6,
INOUT21_7, INOUT21_8

```
*VIS
INOUT22_1, INOUT22_2, INOUT22_3,
INOUT22_4
;
*-------------------------------------------------------
-------------------------------------------------------
-----

*MATERIAL BALANCES

mat_bal1..    0.4176*f('CRs') =G=
f('NAP1s')+f('LSRN1s')+f('HSRN1s');
# ADU
mat_bal2..    1.9821*(f('HSRN2s')+f('H2_1s'))
=e=
f('FG1s')+f('H2S1s')+f('LPG1s')+f('LSRN2s')+f(
'HSRN3s')+f('NAP4s');   # HDT1
mat_bal3..    1.9821*(f('NAP2s')+f('H2_2s'))
=e=
f('FG2s')+f('H2S2s')+f('LPG3s')+f('LSRN3s')+f(
'HSRN4s')+f('NAP3s');   # HDT2
mat_bal4..    f('ISOs')+f('FG4s') =e=
f('LSRN5s');
# ISO
mat_bal5..    f('Ss')+f('TGs') =e=
f('H2S1s')+f('H2S2s');
# SRU
mat_bal6..    f('HSRN5s')+f('NAP5s') =e=
f('H2s')+f('FG3s')+f('LPG2s')+f('REFs');
# REF
mat_bal7..    f('SOLDs') =e=
f('LSRN6s')+f('Ss')+f('GSLNs')+f('LPG5s');
# SOLD
mat_bal8..    f('GSLNs') =e=
f('ISOs')+f('REFs');
# BLND
mat_bal9..    f('LPG5s') =e= f('LPG4s');
# LPG
mat_bal10..   f('FG5s') =e=
f('FG1s')+f('FG2s')+f('FG3s')+f('FG4s');
# FGH
mat_bal11..   f('LSRN4s') =e=
f('LSRN5s')+f('LSRN6s');
# SPLT1
mat_bal12..   f('H2s') =e=
f('H2_1s')+f('H2_2s');
# SPLT2
mat_bal13..
f('HSRN1s')+f('VIS_1s')+f('COK_1s')+f('FCC_1
s')+f('HCR_1s')+f('PCHN1_1s') =e=
f('HSRN2s');             # MIX1
mat_bal14..
f('NAP1s')+f('VIS_2s')+f('COK_2s')+f('FCC_2s'
)+f('HCR_2s')+f('PCHN1_2s') =e= f('NAP2s');
# MIX2
```

```
mat_bal15..
f('LSRN1s')+f('LSRN2s')+f('LSRN3s')+f('PCH
N2s') =e= f('LSRN4s');
# MIX3
mat_bal16..
f('HSRN3s')+f('HSRN4s')+f('PCHN3_1s')+f('H
CR_3s') =e= f('HSRN5s');
# MIX4
mat_bal17..
f('NAP3s')+f('NAP4s')+f('PCHN3_2s')+f('HCR_
4s') =e= f('NAP5s');                        #
MIX5
mat_bal18..   f('LPG1s')+f('LPG2s')+f('LPG3s')
=e= f('LPG4s');
# MIX6


*MAT_BAL_MIXER(MIXER,J)$OUTLET_MI
XER(MIXER,J)..
SUM(J1$INLET_MIXER(MIXER,J1), F(J1))
=E= F(J);

*HEAVY CRUDE (API<33)
mat_bal20..   f('COK_1s')=l=2000000;
mat_bal21..   f('COK_2s')=l=2000000;
mat_bal22..   f('HCR_1s')=l=2000000;
mat_bal23..   f('HCR_2s')=l=2000000;
mat_bal24..   f('HCR_3s')=l=2000000;
mat_bal25..   f('HCR_4s')=l=2000000;
mat_bal26..   f('VIS_1s')=e=0;
mat_bal27..   f('VIS_2s')=e=0;

*LIGHT CRUDE (API>=33)
mat_bal29..   f('VIS_1s')=l=2000000;
mat_bal30..   f('VIS_2s')=l=2000000;
mat_bal31..   f('COK_1s')=e=0;
mat_bal32..   f('COK_2s')=e=0;
mat_bal33..   f('HCR_1s')=e=0;
mat_bal34..   f('HCR_2s')=e=0;
mat_bal35..   f('HCR_3s')=e=0;
mat_bal36..   f('HCR_4s')=e=0;

DUMMY..      SUM (I,Y(I))=G=0;

*-------------------------------------------------------
-------------------------------------------------------
-----
*OBJECTIVE FUNCTION

OBJFN..

TC =e=
*tci
0.45*2.4*(228*y('1')+96*y('7')+
96*y('8')+270*y('17')+42*y('9')+30*y('20'))
```

```
+
0.1*0.45*2.4*(228*y('1')+96*y('7')+
96*y('8')+270*y('17')+42*y('9')+30*y('20'))
+
*toc
*ge
0.3*(0.35*0.45*2.4*(228*y('1')+96*y('7')+96*y
('8')+270*y('17')+42*y('9')+30*y('20'))
+
*Electricity used for ADU, VIS, COK, FCC,
HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR,
HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.0248+
(f('NAP5s')+f('HSRN5s'))*0.2477+
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC,
HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)
+
*CW used for ADU, VIS, COK, FCC, HCR,
HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*(
f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('
PCHN3_1s')+f('PCHN3_2s') ) )

+
*foc
0.35*0.45*2.4*(228*y('1')+96*y('7')+96*y('8')+
270*y('17')+42*y('9')+30*y('20'))

+
*voc
*Electricity used for ADU, VIS, COK, FCC,
HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR,
HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.0248+
```

```
(f('NAP5s')+f('HSRN5s'))*0.2477+
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC,
HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)

+
*CW used for ADU, VIS, COK, FCC, HCR,
HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+
f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s')
+f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*(
f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('
PCHN3_1s')+f('PCHN3_2s'))

+ SUM(MIXER,
CAPEX_M(MIXER)*Y(MIXER)) +
SUM(SPLITTER,
CAPEX_S(SPLITTER)*Y(SPLITTER))    #
(capital cost for mixers and splitters)
*+ SUM(J, 10*Z(J))    # (taken to be or
assumed to be piping cost for the selected
stream)

*+ SUM(I,
COST_PER_UNIT_EMISSION(I)*EI(I))
;
```

```
*-----------------------------------------------------
------------------------------------------------------
-----

*BOUND SECTION
F.UP(J) = 1000000000;

*-----------------------------------------------------
------------------------------------------------------
------
*RAW MATERIAL AVAILABILIY
prodreq1..      f('CRs') =g= 10000000;
prodreq2..      f('CRs') =l= 50000000;

*PRODUCTION REQUIREMENTS
prodreq3..      f('GSLNs')=g=7000000;
prodreq4..      f('LPG5s')=g=1000000;
prodreq5..      f('FG5s')=g=1000000;
prodreq6..      f('PCHN3_1s')=l=1000;
prodreq7..      f('PCHN3_2s')=l=1000;
prodreq8..      f('FCC_1s')=l=2000000;
prodreq9..      f('FCC_2s')=l=2000000;

*-----------------------------------------------------
------------------------------------------------------
----------------------
*DEFINITION OF DISJUNCTION'S
EQUATIONS

*ADU
*INOUT1_1..      0.2088*f('CRs') =e=
f('NAP1s');
INOUT1_2..      0.0555*f('CRs') =e=
f('LSRN1s');
INOUT1_3..      0.1533*f('CRs') =e=
f('HSRN1s');
INOUT1_4..      f('NAP1s') =e= 0;
INOUT1_5..      f('LSRN1s') =e= 0;
INOUT1_6..      f('HSRN1s') =e= 0;
INOUT1_7..      f('CRs') =e= 0;
COST1_1..      c('1') =e= 228;

*HDT-1
INOUT2_1..
0.0109*(f('HSRN2s')+f('H2_1s')) =e= f('FG1s');
INOUT2_2..
0.0012*(f('HSRN2s')+f('H2_1s')) =e=
f('H2S1s');
INOUT2_3..
0.0058*(f('HSRN2s')+f('H2_1s')) =e=
f('LPG1s');
*INOUT2_4..
0.2610*(f('HSRN2s')+f('H2_1s')) =e=
f('LSRN2s');
```

```
*INOUT2_5..
0.7211*(f('HSRN2s')+f('H2_1s')) =e=
f('HSRN3s');
*INOUT2_6..
0.9821*(f('HSRN2s')+f('H2_1s')) =e=
f('NAP4s');
INOUT2_7..
2.763*f('LSRN2s')=e=f('HSRN3s');
INOUT2_8..      f('FG1s') =e= 0;
INOUT2_9..      f('H2S1s') =e= 0;
INOUT2_10..      f('LPG1s') =e= 0;
INOUT2_11..     f('LSRN2s') =e= 0;
INOUT2_12..     f('HSRN3s') =e= 0;
INOUT2_13..     f('NAP4s') =e= 0;
INOUT2_14..     f('HSRN2s') =e= 0;
INOUT2_15..     f('H2_1s') =e= 0;
COST2_1..      c('7') =e= 96;

*HDT-2
INOUT3_1..      0.0109*(f('NAP2s')+f('H2_2s'))
=e= f('FG2s');
INOUT3_2..      0.0012*(f('NAP2s')+f('H2_2s'))
=e= f('H2S2s');
INOUT3_3..      0.0058*(f('NAP2s')+f('H2_2s'))
=e= f('LPG3s');
INOUT3_4..      0.2610*(f('NAP2s')+f('H2_2s'))
=e= f('LSRN3s');
INOUT3_5..      0.7211*(f('NAP2s')+f('H2_2s'))
=e= f('HSRN4s');
INOUT3_6..      0.9821*(f('NAP2s')+f('H2_2s'))
=e= f('NAP3s');
INOUT3_7..      f('FG2s') =e= 0;
INOUT3_8..      f('H2S2s') =e= 0;
INOUT3_9..      f('LPG3s') =e= 0;
INOUT3_10..     f('LSRN3s') =e= 0;
INOUT3_11..     f('HSRN4s') =e= 0;
INOUT3_12..     f('NAP3s') =e= 0;
INOUT3_13..     f('NAP2s') =e= 0;
INOUT3_14..     f('H2_2s') =e= 0;
COST3_1..      c('8') =e= 96;

*ISO
INOUT4_1..      0.99*f('LSRN5s') =e=
f('ISOs');
INOUT4_2..      0.01*f('LSRN5s') =e=
f('FG4s');
INOUT4_3..      f('ISOs') =e= 0;
INOUT4_4..      f('FG4s') =e= 0;
INOUT4_5..      f('LSRN5s') =e= 0;
COST4_1..      c('9') =e= 42;

*SRU
INOUT5_1..      0.8478*(f('H2S1s')+f('H2S2s'))
=e= f('Ss');
INOUT5_2..      0.1522*(f('H2S1s')+f('H2S2s'))
=e= f('TGs');
INOUT5_3..      f('Ss') =e= 0;
INOUT5_4..      f('TGs') =e= 0;
INOUT5_5..      f('H2S1s') =e= 0;
INOUT5_6..      f('H2S2s') =e= 0;
COST5_1..      c('20') =e= 30;

*REF

*INOUT6.. f('HSRN5s')+f('NAP5s') =e=
f('H2s')+f('FG3s')+f('LPG2s')+f('REFs');
*$ontext
INOUT6_1..
0.032*(f('HSRN5s')+f('NAP5s')) =e= f('H2s');
INOUT6_2..
0.037*(f('HSRN5s')+f('NAP5s')) =e= f('FG3s');
INOUT6_3..
0.078*(f('HSRN5s')+f('NAP5s')) =e=
f('LPG2s');
INOUT6_4..
0.853*(f('HSRN5s')+f('NAP5s')) =e= f('REFs');
*$offtext

INOUT6_5..      f('H2s') =e= 0;
INOUT6_6..      f('FG3s') =e= 0;
INOUT6_7..      f('LPG2s') =e= 0;
INOUT6_8..      f('REFs') =e= 0;
INOUT6_9..      f('HSRN5s') =e= 0;
INOUT6_10..      f('NAP5s') =e= 0;
COST6_1..      c('17') =e= 270;

*SOLD
INOUT7_1..      f('SOLDs') =e=
f('LSRN6s')+f('Ss')+f('GSLNs')+f('LPG5s');
INOUT7_2..      f('SOLDs') =e= 0;
INOUT7_3..      f('LSRN6s') =e= 0;
INOUT7_4..      f('Ss') =e= 0;
INOUT7_5..      f('GSLNs') =e= 0;
INOUT7_6..      f('LPG5s') =e= 0;
COST7_1..      c('21') =e= 10;

*BLND
INOUT8_1..      f('GSLNs') =e=
f('ISOs')+f('REFs');
INOUT8_2..      f('GSLNs') =e= 0;
INOUT8_3..      f('ISOs') =e= 0;
INOUT8_4..      f('REFs') =e= 0;
COST8_1..      c('2') =e= 10;

*LPG
INOUT9_1..      f('LPG5s') =e= f('LPG4s');
INOUT9_2..      f('LPG5s') =e= 0;
INOUT9_3..      f('LPG4s') =e= 0;
COST9_1..      c('10') =e= 10;
```

```
*FGH
INOUT10_1..     f('FG5s') =e=
f('FG1s')+f('FG2s')+f('FG3s')+f('FG4s');
INOUT10_2..     f('FG5s') =e= 0;
INOUT10_3..     f('FG1s') =e= 0;
INOUT10_4..     f('FG2s') =e= 0;
INOUT10_5..     f('FG3s') =e= 0;
INOUT10_6..     f('FG4s') =e= 0;
COST10_1..     c('5') =e= 10;

*SPLT 1
INOUT11_1..     0.9*f('LSRN4s') =e=
f('LSRN5s');
INOUT11_2..     0.1*f('LSRN4s') =e=
f('LSRN6s');
INOUT11_3..     f('LSRN4s') =e= 0;
INOUT11_4..     f('LSRN5s') =e= 0;
INOUT11_5..     f('LSRN6s') =e= 0;
COST11_1..     c('18') =e= 10;

*SPLT 2
INOUT12_1..     f('H2s') =e=
f('H2_1s')+f('H2_2s');
INOUT12_2..     f('H2s') =e= 0;
INOUT12_3..     f('H2_1s') =e= 0;
INOUT12_4..     f('H2_1s') =e= 0;
COST12_1..     c('19') =e= 10;

*MIX-1
INOUT13_1..
f('HSRN1s')+f('VIS_1s')+f('COK_1s')+f('FCC_1
s')+f('HCR_1s')+f('PCHN1_1s') =e=
f('HSRN2s');
INOUT13_2..     f('HSRN1s') =e= 0;
INOUT13_3..     f('VIS_1s') =e= 0;
INOUT13_4..     f('COK_1s') =e= 0;
INOUT13_5..     f('FCC_1s') =e= 0;
INOUT13_6..     f('HCR_1s') =e= 0;
INOUT13_7..     f('PCHN1_1s') =e= 0;
INOUT13_8..     f('HSRN2s') =e= 0;
COST13_1..     c('11') =e= 10;

*MIX-2
INOUT14_1..
f('NAP1s')+f('VIS_2s')+f('COK_2s')+f('FCC_2s'
)+f('HCR_2s')+f('PCHN1_2s') =e= f('NAP2s');
INOUT14_2..     f('NAP1s') =e= 0;
INOUT14_3..     f('VIS_2s') =e= 0;
INOUT14_4..     f('COK_2s') =e= 0;
INOUT14_5..     f('FCC_2s') =e= 0;
INOUT14_6..     f('HCR_2s') =e= 0;
INOUT14_7..     f('PCHN1_2s') =e= 0;
INOUT14_8..     f('NAP2s') =e= 0;
COST14_1..     c('12') =e= 10;

*MIX-3
INOUT15_1..     f('LSRN4s') =e=
f('LSRN1s')+f('LSRN2s')+f('LSRN3s')+f('PCH
N2s');
INOUT15_2..     f('LSRN4s') =e= 0;
INOUT15_3..     f('LSRN1s') =e= 0;
INOUT15_4..     f('LSRN2s') =e= 0;
INOUT15_5..     f('LSRN3s') =e= 0;
INOUT15_6..     f('PCHN2s') =e= 0;
COST15_1..     c('13') =e= 10;

*MIX-4
INOUT16_1..
f('HSRN3s')+f('HSRN4s')+f('PCHN3_1s')+f('H
CR_3s') =e= f('HSRN5s');
INOUT16_2..     f('HSRN3s') =e= 0;
INOUT16_3..     f('HSRN4s') =e= 0;
INOUT16_4..     f('PCHN3_1s') =e= 0;
INOUT16_5..     f('HCR_3s') =e= 0;
INOUT16_6..     f('HSRN5s') =e= 0;
COST16_1..     c('14') =e= 10;

*MIX-5
INOUT17_1..
f('NAP3s')+f('NAP4s')+f('PCHN3_2s')+f('HCR_
4s') =e= f('NAP5s');
INOUT17_2..     f('NAP3s') =e= 0;
INOUT17_3..     f('NAP4s') =e= 0;
INOUT17_4..     f('PCHN3_2s') =e= 0;
INOUT17_5..     f('HCR_4s') =e= 0;
INOUT17_6..     f('NAP5s') =e= 0;
COST17_1..     c('15') =e= 10;

*MIX-6
INOUT18_1..     f('LPG4s') =e=
f('LPG1s')+f('LPG2s')+f('LPG3s');
INOUT18_2..     f('LPG4s') =e= 0;
INOUT18_3..     f('LPG1s') =e= 0;
INOUT18_4..     f('LPG2s') =e= 0;
INOUT18_5..     f('LPG3s') =e= 0;
COST18_1..     c('16') =e= 10;

*COK
INOUT19_1..     f('COK_1s')=l= 2000000;
INOUT19_2..     f('COK_2s')=l= 2000000;
INOUT19_3..     f('COK_1s')=e= 0;
INOUT19_4..     f('COK_2s')=e= 0;

*FCC
INOUT20_1..     f('FCC_1s')=l= 2000000;
INOUT20_2..     f('FCC_2s')=l= 2000000;
INOUT20_3..     f('FCC_1s')=e= 0;
INOUT20_4..     f('FCC_2s')=e= 0;
```

```
*HCR
INOUT21_1..     f('HCR_1s')=l= 2000000;
INOUT21_2..     f('HCR_2s')=l= 2000000;
INOUT21_3..     f('HCR_3s')=l= 2000000;
INOUT21_4..     f('HCR_4s')=l= 2000000;
INOUT21_5..     f('HCR_1s')=e= 0;
INOUT21_6..     f('HCR_2s')=e= 0;
INOUT21_7..     f('HCR_3s')=e= 0;
INOUT21_8..     f('HCR_4s')=e= 0;

*VIS
INOUT22_1..     f('VIS_1s')=l= 2000000;
INOUT22_2..     f('VIS_2s')=l= 2000000;
INOUT22_3..     f('VIS_1s')=e= 0;
INOUT22_4..     f('VIS_2s')=e= 0;


*-------------------------------------------------------
-------------------------------------------------------
-----
Model naphtha_opt_hvy
/
objfn

*-------------------------------------------------------
-------------------------------------------------------
--
*MATERIAL BALANCES

mat_bal1, mat_bal2, mat_bal3, mat_bal4,
mat_bal5, mat_bal6, mat_bal7, mat_bal8
mat_bal9, mat_bal10, mat_bal11,mat_bal12,
mat_bal13, mat_bal14, mat_bal15
mat_bal16, mat_bal17, mat_bal18

*MAT_BAL_MIXER
mat_bal20, mat_bal21, mat_bal22, mat_bal23,
mat_bal24, mat_bal25, mat_bal26
mat_bal27
*RAW MATERIAL AVAILABILIY
prodreq1, prodreq2
*PRODUCTION REQUIREMENTS
prodreq3, prodreq4, prodreq5, prodreq6,
prodreq7, prodreq8,prodreq9


*-------------------------------------------------------
-------------------------------------------------------
---
*DEFINITION OF DISJUNCTION'S
EQUATIONS
*ADU
*INOUT1_1,
INOUT1_2,INOUT1_3,INOUT1_4,INOUT1_5,
INOUT1_6,INOUT1_7
COST1_1

*HDT-1
INOUT2_1,INOUT2_2,INOUT2_3,INOUT2_7,
INOUT2_8,INOUT2_9,INOUT2_10,INOUT2_
11,INOUT2_12,INOUT2_13,INOUT2_14,INO
UT2_15
*INOUT2_4,INOUT2_6, INOUT2_5,
COST2_1

*HDT-2
INOUT3_1,INOUT3_2,INOUT3_3,INOUT3_4,
INOUT3_5,INOUT3_6,INOUT3_7,INOUT3_8,
INOUT3_9,INOUT3_10,INOUT3_11,INOUT3
_12,INOUT3_13,INOUT3_14
COST3_1

*ISO
INOUT4_1,INOUT4_2,INOUT4_3,INOUT4_4,
INOUT4_5
COST4_1

*SRU
INOUT5_1,INOUT5_2,INOUT5_3,INOUT5_4,
INOUT5_5,INOUT5_6
COST5_1


*REF
*INOUT6
INOUT6_1,INOUT6_2,INOUT6_3,INOUT6_4,
INOUT6_5,INOUT6_6,INOUT6_7,INOUT6_8,
INOUT6_9,INOUT6_10
COST6_1

*SOLD
INOUT7_1,INOUT7_2,INOUT7_3,INOUT7_4,
INOUT7_5 ,INOUT7_6
COST7_1

*BLND
INOUT8_1,INOUT8_2,INOUT8_3,INOUT8_4
COST8_1

*LPG
INOUT9_1,INOUT9_2,INOUT9_3
COST9_1

*FG
INOUT10_1,INOUT10_2,INOUT10_3,INOUT
10_4,INOUT10_5,INOUT10_6
COST10_1

*SPLT 1
```

```
INOUT11_1,INOUT11_2,INOUT11_3,INOUT
11_4,INOUT11_5
COST11_1

*SPLT 2
INOUT12_1,INOUT12_2,INOUT12_3,INOUT
12_4
COST12_1

*MIX-1
INOUT13_1,INOUT13_2,INOUT13_3,INOUT
13_4,INOUT13_5,INOUT13_6,INOUT13_7,IN
OUT13_8
COST13_1

*MIX-2
INOUT14_1,INOUT14_2,INOUT14_3,INOUT
14_4,INOUT14_5,INOUT14_6,INOUT14_7,IN
OUT14_8
COST14_1

*MIX-3
INOUT15_1,INOUT15_2,INOUT15_3,INOUT
15_4,INOUT15_5,INOUT15_6
COST15_1

*MIX-4
INOUT16_1,INOUT16_2,INOUT16_3,INOUT
16_4,INOUT16_5,INOUT16_6
COST16_1

*MIX-5
INOUT17_1,INOUT17_2,INOUT17_3,INOUT
17_4,INOUT17_5,INOUT17_6
COST17_1

*MIX-6
INOUT18_1,INOUT18_2,INOUT18_3,INOUT
18_4,INOUT18_5
COST18_1

*COK
INOUT19_1, INOUT19_2, INOUT19_3,
INOUT19_4

*FCC
INOUT20_1, INOUT20_2,INOUT20_3,
INOUT20_4

*HCR
INOUT21_1, INOUT21_2,INOUT21_3,
INOUT21_4, INOUT21_5, INOUT21_6,
INOUT21_7, INOUT21_8

*VIS
INOUT22_1, INOUT22_2, INOUT22_3,
INOUT22_4

*DUMMY
DUMMY

*EMISSION(I,ENV,POLLUTANT)
*EMISSION_REGULATION
/
*---------------------------------------------------------
-----------------------------------------------------------
------
naphtha_opt_lgt
/
objfn
*---------------------------------------------------------
-----------------------------------------------------------
-----
*MATERIAL BALANCES

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5
mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

*MAT_BAL_MIXER

mat_bal29
mat_bal30
mat_bal31
mat_bal32
mat_bal33
mat_bal34
mat_bal35
mat_bal36
*RAW MATERIAL AVAILABILIY
prodreq1
prodreq2
*PRODUCTION REQUIREMENTS
prodreq3
prodreq4
```

```
prodreq5
prodreq6
prodreq7
prodreq8
prodreq9
*-------------------------------------------------------
-------------------------------------------------------
---
*DEFINITION OF DISJUNCTION'S
EQUATIONS
*ADU
*INOUT1_1,
INOUT1_2,INOUT1_3,INOUT1_4,INOUT1_5,
INOUT1_6,INOUT1_7
COST1_1

*HDT-1
INOUT2_1,INOUT2_2,INOUT2_3,INOUT2_7,
INOUT2_8,INOUT2_9,INOUT2_10,INOUT2_
11,INOUT2_12,INOUT2_13,INOUT2_14,INO
UT2_15
*INOUT2_4,INOUT2_6, INOUT2_5,
COST2_1

*HDT-2
INOUT3_1,INOUT3_2,INOUT3_3,INOUT3_4,
INOUT3_5,INOUT3_6,INOUT3_7,INOUT3_8,
INOUT3_9,INOUT3_10,INOUT3_11,INOUT3
_12,INOUT3_13,INOUT3_14
COST3_1

*ISO
INOUT4_1,INOUT4_2,INOUT4_3,INOUT4_4,
INOUT4_5
COST4_1

*SRU
INOUT5_1,INOUT5_2,INOUT5_3,INOUT5_4,
INOUT5_5,INOUT5_6
COST5_1

*REF
*INOUT6
INOUT6_1,INOUT6_2,INOUT6_3,INOUT6_4,
INOUT6_5,INOUT6_6,INOUT6_7,INOUT6_8,
INOUT6_9,INOUT6_10
COST6_1

*SOLD
INOUT7_1,INOUT7_2,INOUT7_3,INOUT7_4,
INOUT7_5 ,INOUT7_6
COST7_1

*BLND
INOUT8_1,INOUT8_2,INOUT8_3,INOUT8_4
COST8_1

*LPG
INOUT9_1,INOUT9_2,INOUT9_3
COST9_1

*FG
INOUT10_1,INOUT10_2,INOUT10_3,INOUT
10_4,INOUT10_5,INOUT10_6
COST10_1

*SPLT 1
INOUT11_1,INOUT11_2,INOUT11_3,INOUT
11_4,INOUT11_5
COST11_1

*SPLT 2
INOUT12_1,INOUT12_2,INOUT12_3,INOUT
12_4
COST12_1

*MIX-1
INOUT13_1,INOUT13_2,INOUT13_3,INOUT
13_4,INOUT13_5,INOUT13_6,INOUT13_7,IN
OUT13_8
COST13_1

*MIX-2
INOUT14_1,INOUT14_2,INOUT14_3,INOUT
14_4,INOUT14_5,INOUT14_6,INOUT14_7,IN
OUT14_8
COST14_1

*MIX-3
INOUT15_1,INOUT15_2,INOUT15_3,INOUT
15_4,INOUT15_5,INOUT15_6
COST15_1

*MIX-4
INOUT16_1,INOUT16_2,INOUT16_3,INOUT
16_4,INOUT16_5,INOUT16_6
COST16_1

*MIX-5
INOUT17_1,INOUT17_2,INOUT17_3,INOUT
17_4,INOUT17_5,INOUT17_6
COST17_1

*MIX-6
INOUT18_1,INOUT18_2,INOUT18_3,INOUT
18_4,INOUT18_5
COST18_1

*COK
```

```
INOUT19_1, INOUT19_2, INOUT19_3,
INOUT19_4

*FCC
INOUT20_1, INOUT20_2,INOUT20_3,
INOUT20_4

*HCR
INOUT21_1, INOUT21_2,INOUT21_3,
INOUT21_4, INOUT21_5, INOUT21_6,
INOUT21_7, INOUT21_8

*VIS
INOUT22_1, INOUT22_2, INOUT22_3,
INOUT22_4

*DUMMY
DUMMY

*EMISSION(I,ENV,POLLUTANT)
*EMISSION_REGULATION
/

;

OPTION
MIP = LMCHULL
LIMROW = 10000
LIMCOL = 10000;

*-------------------------------------------------------
--------------------------------------------------------
----------------------
*BEGIN DECLARATIONS AND
DEFINITIONS OF DISJUNCTIONS (LOGMIP
SECTION)
$ONECHO>"%lm.info%"

Disjunction
D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,
D13,D14,D15,D16,D17,D18,D19,D20,D21,D22
;

D1 is if Y('1') then
      INOUT1_2;
      INOUT1_3;
      COST1_1;
      else
      INOUT1_4;
      INOUT1_5;
      INOUT1_6;
      INOUT1_7;
      endif;

D2 is if Y('7') then
      INOUT2_1;
      INOUT2_2;
      INOUT2_3;
      INOUT2_7;
      COST2_1;
      else
      INOUT2_8;
      INOUT2_9;
      INOUT2_10;
      INOUT2_11;
      INOUT2_12;
      INOUT2_13;
      INOUT2_14;
      INOUT2_15;
      endif;

D3 is if Y('8') then
      INOUT3_1;
      INOUT3_2;
      INOUT3_3;
      INOUT3_4;
      INOUT3_5;
      INOUT3_6;
      COST3_1;
      else
      INOUT3_7;
      INOUT3_8;
      INOUT3_9;
      INOUT3_10;
      INOUT3_11;
      INOUT3_12;
      INOUT3_13;
      INOUT3_14;
      endif;

D4 is if Y('9') then
      INOUT4_1;
      INOUT4_2;
      COST4_1;
      else
      INOUT4_3;
      INOUT4_4;
      INOUT4_5;
      endif;

D5 is if Y('20') then
      INOUT5_1;
      INOUT5_2;
      COST5_1;
      else
      INOUT5_3;
      INOUT5_4;
      INOUT5_5;
      INOUT5_6;
```

```
        endif;

D6 is if Y('17') then
        INOUT6_1;
        INOUT6_2;
        INOUT6_3;
        INOUT6_4;
        COST6_1;
        else
        INOUT6_5;
        INOUT6_6;
        INOUT6_7;
        INOUT6_8;
        INOUT6_9;
        INOUT6_10;
        endif;

D7 is if Y('21') then
        INOUT7_1;
        COST7_1;
        else
        INOUT7_2;
        INOUT7_3;
        INOUT7_4;
        INOUT7_5;
        INOUT7_6;
        endif;

D8 is if Y('2') then
        INOUT8_1;
        COST8_1;
        else
        INOUT8_2;
        INOUT8_3;
        INOUT8_4;
        endif;

D9 is if Y('10') then
        INOUT9_1;
        COST9_1;
        else
        INOUT9_2;
        INOUT9_3;
        endif;

D10 is if Y('5') then
        INOUT10_1;
        COST10_1;
        else
        INOUT10_2;
        INOUT10_3;
        INOUT10_4;
        INOUT10_5;
        INOUT10_6;
        endif;

D11 is if Y('18') then
        INOUT11_1;
        INOUT11_2;
        COST11_1;
        else
        INOUT11_3;
        INOUT11_4;
        INOUT11_5;
        endif;

D12 is if Y('19') then
        INOUT12_1;
        COST12_1;
        else
        INOUT12_2;
        INOUT12_3;
        INOUT12_4;
        endif;

D13 is if Y('11') then
        INOUT13_1;
        COST13_1;
        else
        INOUT13_2;
        INOUT13_3;
        INOUT13_4;
        INOUT13_5;
        INOUT13_6;
        INOUT13_7;
        INOUT13_8;
        endif;

D14 is if Y('12') then
        INOUT14_1;
        COST14_1;
        else
        INOUT14_2;
        INOUT14_3;
        INOUT14_4;
        INOUT14_5;
        INOUT14_6;
        INOUT14_7;
        INOUT14_8;
        endif;

D15 is if Y('13') then
        INOUT15_1;
        COST15_1;
        else
        INOUT15_2;
        INOUT15_3;
        INOUT15_4;
        INOUT15_5;
        INOUT15_6;
```

```
     endif;

D16 is if Y('14') then
     INOUT16_1;
     COST16_1;
     else
     INOUT16_2;
     INOUT16_3;
     INOUT16_4;
     INOUT16_5;
     INOUT16_6;
     endif;

D17 is if Y('15') then
     INOUT17_1;
     COST17_1;
     else
     INOUT17_2;
     INOUT17_3;
     INOUT17_4;
     INOUT17_5;
     INOUT17_6;
     endif;

D18 is if Y('16') then
     INOUT18_1;
     COST18_1;
     else
     INOUT18_2;
     INOUT18_3;
     INOUT18_4;
     INOUT18_5;
     endif;

D19 is if Y('3') then
     INOUT19_1;
     INOUT19_2;
     else
     INOUT19_3;
     INOUT19_4;
     endif;

D20 is if Y('4') then
     INOUT20_1;
     INOUT20_2;
     else
     INOUT20_3;
     INOUT20_4;
     endif;

D21 is if Y('6') then
     INOUT21_1;
     INOUT21_2;
     INOUT21_3;
     INOUT21_4;
     else
     INOUT21_5;
     INOUT21_6;
     INOUT21_7;
     INOUT21_8;
     endif;

D22 is if Y('22') then
     INOUT22_1;
     INOUT22_2;
     else
     INOUT22_3;
     INOUT22_4;
     endif;

Y('13') and Y('11') -> NOT Y('12');
Y('22') -> Y('11') or Y('12');
Y('6') -> Y('11') or Y('12') or Y('14') or Y('15');
Y('3') -> Y('11') or Y('12');
Y('4') -> Y('11') or Y('12');

$OFFECHO
*END LOGMIP SECTION

if ((API lt 33),
Solve
naphtha_opt_hvy using mip minimizing tc;

Display
TC.l,c.l,f.l,y.l;
;

else
Solve
naphtha_opt_lgt using mip minimizing tc;
;

Display
TC.l,c.l,f.l,y.l
;)
```