

**Optimization of
Upstream Offshore Oilfield Production Planning under Uncertainty
and
Downstream Crude Oil Scheduling at Refinery Front-End**

by

Tan Yin Keong

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

JANUARY 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

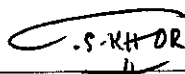
**Optimization of
Upstream Offshore Oilfield Production Planning under Uncertainty
and
Downstream Crude Oil Scheduling at Refinery Front-End**

by

Tan Yin Keong

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,



(KHOR CHENG SEONG)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

January 2009

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.



(TAN YIN KEONG)

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge the people who had given their support and help throughout the completion of this project.

First of all, I would like acknowledge the endless help and support received from my supervisor, Mr. Khor Cheng Seong. His assistance and guidance have been the main source of motivation to work on this project.

Last but not least, I would like to thank all my fellow colleagues for their assistance and ideas in the completion of this project.

ABSTRACT

In this work, we have attempted to solve two problems concerning the planning and scheduling of crude oil operations: first, on the upstream production planning of crude oil from offshore sources and second, on the scheduling of downstream processing of crude oil at the refinery front-end.

The first part is on the offshore oilfield infrastructures planning under both exogenous uncertainty and endogeneous decision-dependent uncertainty. A model representative of the oilfield that is able to select the best routes to obtain the desired objective function is considered. The methodology used is by firstly developing a deterministic model and modeling it with GAMS, followed by a stochastic one. The results obtained show a high accuracy representation in which the uncertainties in both the exogenous and endogeneous uncertainties in planning are accounted for. The stochastic model is a more thorough representation of the problem because it considers all the uncertainties along with the associated probabilities. Having validated the model formulation and solution obtained with results for standard problems reported in the literature, we believe that the model can be a tool to assist upper-level management in preliminary decision-making on an optimal plan for crude oil production from an offshore operation.

The second part is on the scheduling of crude oil operations at a refinery front-end. A technique for obtaining globally optimal schedules for the flow of crude is developed. A continuous time model based on transfer events is used to represent the scheduling problem and this model is a nonconvex MINLP model which presents multiple local optima. We implement a branch-and-contract algorithm that aims at reducing the size of the search region. In order to obtain a global optimum solution of the problem, an outer-approximation algorithm is proposed, whereby lower and upper bounds on the global optimum are generated, which are converged to a specified tolerance. The solution obtained from the LB-MILP model, i.e., the decision variables (binary variables), was used to obtain a feasible solution for model UB-NLP. This solution is the upper bound solution. The application of the proposed algorithm shows significant reduction in the computational effort involved in solving

the problem. Slack variables are introduced to overcome the integer infeasibility problem. The optimization model is developed using GAMS and an optimal solution is found with no logical constraints conflicts or error.

The main contribution on this work in the first part is to conduct an extensive study on the implementation of the model formulation in Iyer et al. (1998). As well, in the second part, we are focused on investigating effective implementation strategies of the model formulation and solution strategy in Karuppiah et al. (2008) using our choice of the modeling platform GAMS and the best numerical solvers that are available. Hence, most of the exposition on the model formulation and solution algorithms are taken directly from the original papers so as to provide the readers with the most accurate information possible.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF FIGURES	xi
LIST OF TABLES	xii

**SECTION A: AN OPTIMIZATION-BASED COMPUTATIONAL
FRAMEWORK FOR UPSTREAM OFFSHORE OILFIELD PLANNING
UNDER UNCERTAINTY 1**

CHAPTER A1 INTRODUCTION	2
A1.1 INTRODUCTION TO STOCHASTIC PROGRAMMING	2
A1.2 PROBLEM STATEMENT	3
A1.2.1 Model Assumptions	3
A1.3 RESEARCH OBJECTIVES	4

CHAPTER A2 LITERATURE REVIEW	6
A2.1 PREVIOUS WORKS	6
A2.1.1 Location-Allocation Problem	6
A2.1.2 Production Planning and Scheduling	6

CHAPTER A3 MODEL FORMULATION	9
A3.1 DETERMINISTIC MODEL FORMULATION FOR OILFIELD PRODUCTION PLANNING	9
A3.1.1 Nomenclature	9
A3.1.2 Model Equations	11
A3.2 STOCHASTIC MODEL FORMULATION FOR OILFIELD PRODUCTION PLANNING	24
A3.2.1. Exogeneous uncertainty – price	24
A3.2.2. Endogeneous uncertainty – reserves	24
A3.2.3. Scenario Generation	32

CHAPTER A4 COMPUTATIONAL RESULTS AND DISCUSSIONS.	33
A4.1 COMPUTATIONAL RESULTS	33
A4.2 DISCUSSIONS	35
CHAPTER A5 CONCLUSIONS AND RECOMMENDATIONS .	37
A5.1 CONCLUSIONS	37
A5.2 RECOMMENDATIONS	37
SECTION B: AN OPTIMIZATION-BASED COMPUTATIONAL FRAMEWORK FOR DOWNSTREAM CRUDE OIL SCHEDULING AT REFINERY FRONT-END	38
CHAPTER B1 INTRODUCTION	39
B1.1 BACKGROUND OF STUDY	39
B1.2 PROBLEM STATEMENT	40
B1.2.1 Model Information	40
B1.2.2 Model Constraints	41
B1.2.3 Model Assumptions	41
B1.3 RESEARCH OBJECTIVES	42
CHAPTER B2 LITERATURE REVIEW	43
B2.1 PREVIOUS WORKS	43
CHAPTER B3 MODEL FORMULATION AND SOLUTION STRATEGY	46
B3.1 FORMULATION	46
B3.1.1 Nomenclature	46
B3.1.2 Model Equations	49
B3.2 SOLUTION STRATEGY	62
B3.2.1 Preprocessing	62
B3.2.2 Lower Bounding Problem	62
B3.2.3 Upper Bounding Sub-problem	69
B3.2.4 Outer-approximation Algorithm	70

B3.3	UPDATING LAGRANGE MULTIPLIERS	. . .	71
B3.4	BRANCH CONTRACTION ALGORITHM	. . .	73
B3.5	SLACK VARIABLES	76
B3.6	MODEL DATA FOR NUMERICAL EXAMPLE	. . .	78
CHAPTER B4 COMPUTATIONAL RESULTS AND DISCUSSIONS ON			
	NUMERICAL RESULTS	81
B4.1	COMPUTATIONAL RESULTS	81
	B4.1.1 Bound Contraction Results	82
	B4.1.2 Overall Results	82
B4.2	DISCUSSION	86
CHAPTER B5 CONCLUSIONS AND RECOMMENDATIONS			
B5.1	CONCLUSIONS	88
B5.2	RECOMMENDATIONS	88
	REFERENCES	90
	APPENDICES	I
	APPENDIX I	I
	(a) Gantt Chart (1 st Semester)	I
	(b) Gantt Chart (2 nd Semester)	II
	APPENDIX II: COMPUTER CODE FOR GAMS INPUT FILE ON		
	DETERMINISTIC MODEL	III
	APPENDIX III: GAMS OUTPUT FILE ON DETERMINISTIC MODEL		VIII
	APPENDIX IV: COMPUTER CODE FOR GAMS INPUT FILE ON		
	STOCHASTIC MODEL	IX
	APPENDIX V: GAMS OUTPUT FILE ON STOCHASTIC MODEL		XIV
	APPENDIX VI: COMPUTER CODE FOR GAMS INPUT FILE ON BRANCH		
	CONTRACT ALGORITHM MODEL	XVII
	APPENDIX VII: GAMS OUTPUT FILE ON BRANCH AND CONTRACT		
	ALGORITHM MODEL	LVI
	APPENDIX VIII: COMPUTER CODE FOR GAMS INPUT FILE ON CRUDE		
	OIL SCHEDULING WITH SLACK VARIABLES MODEL	LVII

**APPENDIX IX: GAMS OUTPUT FILE ON CRUDE OIL SCHEDULING
WITH SLACK VARIABLES MODEL LXXXIII**

LIST OF FIGURES

FIGURE A1.1: Superstructure of an oil field	5
FIGURE A3.1: Reservoir performance characteristics	15
FIGURE A3.2: Linear reservoir model for an oil well platform	26
FIGURE A3.3: Nine scenarios arising from uncertainty in size and initial deliverability of well	27
FIGURE A3.4: Nine scenarios with their probability values	28
FIGURE B1.1: Superstructure representation of the crude oil scheduling operations at refinery front-end	40
FIGURE B3.1: Crude tank representation	50
FIGURE B3.2: Spatial decomposition of network structure.	65
FIGURE B3.3: Proposed outer-approximation algorithm	70
FIGURE B4.1: Connections in a refinery network	78
FIGURE B4.2: Flow of crude oil at refinery front-end for model LB-MILP	83
FIGURE B4.3: Optimal network structure for crude oil flow at refinery front-end	83
FIGURE B4.4: Gantt chart of the optimal crude oil schedule	84

LIST OF TABLES

TABLE A3.1: Data for production planning computation	23
TABLE A3.2: Oil price scenarios	24
TABLE A3.3: Size, initial deliverability and probability for each scenario	28
TABLE A3.4: Joint probability distribution calculations for scenario generation	32
TABLE A4.1: Attributes of computing facilities for computational experiments	33
TABLE A4.2: Computational statistics for deterministic and stochastic models	33
TABLE A4.3: Deterministic model results	34
TABLE A4.4: Stochastic model results	34
TABLE A4.5: Computational results for the deterministic and stochastic models	35
TABLE B3.1: Data for crude oil scheduling computation	79
TABLE B3.2: Lagrange multiplier information	80
TABLE B4.1: Attributes of computing facilities for computational experiments	81
TABLE B4.2: Computational statistics for all models	81
TABLE B4.3: Computational results for all models	85

SECTION A
AN OPTIMIZATION-BASED COMPUTATIONAL
FRAMEWORK
FOR UPSTREAM OFFSHORE OILFIELD PLANNING UNDER
UNCERTAINTY

CHAPTER A1

INTRODUCTION

A1.1 INTRODUCTION TO STOCHASTIC PROGRAMMING

Stochastic programming is a framework for modeling optimization problems that involve uncertainty. Whereas deterministic optimization problems are formulated with known parameters, real world problems almost invariably include some unknown parameters. When the parameters are known only within certain bounds, one approach to tackling such problems is called robust optimization (Beasley, 2002). Here the goal is to find a solution which is feasible for all such data and optimal in some sense. Stochastic programming models are similar in style but take advantage of the fact that probability distributions governing the data are known or can be estimated.

The most widely applied and studied stochastic programming models are two-stage linear programs. Here the decision maker takes some action in the first stage, after which a random event occurs affecting the outcome of the first-stage decision. A recourse decision can then be made in the second stage that compensates for any bad effects that might have been experienced as a result of the first-stage decision. The optimal policy from such a model is a single first-stage policy and a collection of recourse decisions (a decision rule) defining which second-stage action should be taken in response to each random outcome.

In summary, stochastic programming, as the name implies, is mathematical (i.e. linear, integer, mixed-integer, nonlinear) programming but with a stochastic element present in the data.

In deterministic mathematical programming the data (coefficients) are known numbers. In stochastic programming these numbers are unknown; instead we may

have a probability distribution present. Stochastic programming therefore deals with situations where we have uncertainty present.

A1.2 PROBLEM STATEMENT

The design of an oil production system is a complex task that requires the consideration of many factors (Heever and Grossmann, 2000). We need to develop and implement a systematic and automated approach to efficiently and rigorously integrate the elaborate interactions between the design decision variables selecting the best among a set of possible solutions, using sound engineering judgment.

An oilfield layout consists of a number of fields, each containing one or more reservoirs, and each reservoir contains one or more well sites. After the decision has been made to produce oil from a given well site, it is drilled from a well platform (WP) using drilling rigs. A network of pipelines connects the wells to the WPs and the WPs to the production platforms (PP). The location of production platforms (PP), well platforms (WP), and allocation of well (W) to them is a complex optimization problem, as the costs of drilling are affected by the lengths of pipe required (Iyer et al., 1998).

A1.2.1 Model Assumptions

The major assumptions in the model are as follows:

- the productivity index (PI) of each well is constant across the planning horizon;
- operating conditions are constant during a given time period;
- each reservoir contains a single homogeneous mixture at the same pressure;
- multiple wells in a reservoir produce independently on the basis of each well's PI and the reservoir's average pressure;
- pressure drop in pipes and wells is linear in the oil flow rate and gas flow rate (in reality, flow rate is proportional to the square root of pressure drop);
- the well oil flow rate is proportional to the pressure difference between the well head and well bore;

- each well has a single completion in a single reservoir. This implies that a drilled well is capped after extracting oil from the well. Therefore, it is not reused to drill from this well to other well locations in the reservoir;
- the location of potential wells and platforms as well as their potential interconnections to wells.

Further, we consider that each well may be allocated to two or more WPs. However, this will clearly increase the computational effort significantly. A more practical option might be to consider allocating each well to up to two WPs. In this case, the two different allocations are treated as two choices of identical wells, of which only one can be selected (Heever and Grossmann, 2000).

Note that wells from a reservoir may be potentially allocated to more than one well platform and one WP may produce wells from more than one reservoir. However, once a selection is made, each well is assigned to a single WP; Iyer et al., 1998).

A1.3 RESEARCH OBJECTIVES

The questions that we are interested to answer in this research project relates to the planning decisions of oilfield production in the specified time periods, in which the decision variables include:

- whether or not to install a production platform or well platform/which well platforms should be built and when?;
- whether or not to drill each well (at a specified given location and allocation);
- the production profile (of oil) for each well;
- when and where should the drilling rigs be moved?;
- how much is the investment for WP, PP, and drilling rigs?

The superstructure is as shown in Figure A1.1 below:

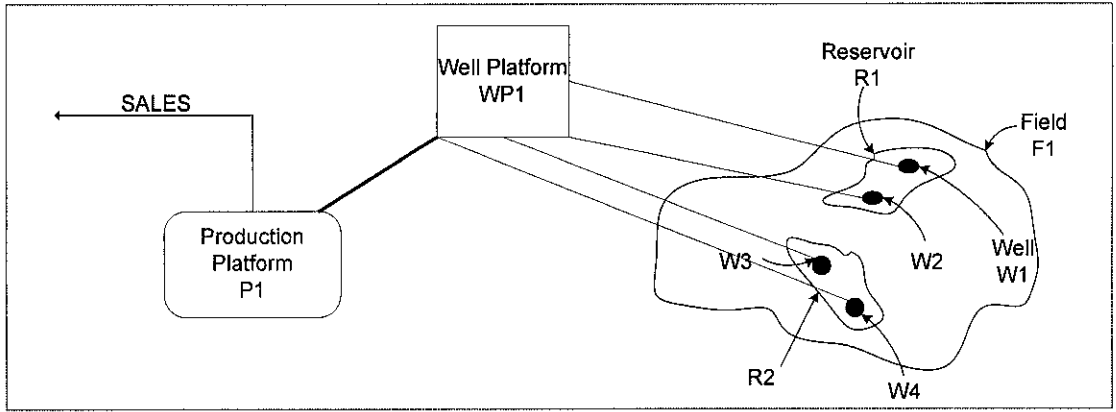


FIGURE A1.1: Superstructure of an oil field

CHAPTER A2

LITERATURE REVIEW

A2.1 PREVIOUS WORK

Previous work in the optimization of offshore field development may be classified into two broad categories, which are location-allocation problems and production planning and scheduling problems.

A2.1.1 Location-Allocation Problem

One of the earliest works by Devine and Lesso (1972) solves the problem of determining the continuous two-dimensional location of production platforms and the allocation of wells to well platforms. The trade-offs associated with the costs involve the capacity of the platforms and the cost of piping from wells to the platforms. They proposed an iterative two-stage algorithm that first fixes the allocation of wells to the platforms and then determines the location of platforms. In the next stage, the location determined in the first stage is fixed and the allocation problem is solved to get a new set of well allocation variables. The problem, however, does not include the scheduling of well drilling or production planning of wells.

Garcia-Diaz et al. (1996) presented a network representation for the same problem and proposed a Lagrangean relaxation solution method.

A2.1.2 Production Planning and Scheduling

Lee and Aranofsky (1958) formulated a production planning problem that expressed the performance of reservoirs linearly as a function of time. Sullivan (1982) used a nonlinear reservoir performance equation and approximated it by using piecewise

linear interpolation using integer variables. Both these works did not include the scheduling of drilling of wells and surface pressure constraints.

Aranofsky and Williams (1962) proposed an LP model for scheduling of well drilling, assuming a preset production profile. In addition, the model had the drawback of fractional solutions for the number of wells drilled in a time period. Bohannon (1970) proposed a mixed-integer linear programming (MILP) model for development of multireservoir systems, assuming a predetermined linear decline of production rate with cumulative oil produced. Frair and Devine (1973) proposed a model that simultaneously included the location-allocation of wells, scheduling of facility operation, and production rates for different time periods. However, the model did not include the reservoir performance equations and assumed a linear production decline curve for each reservoir.

Costa (1975) and Dogru (1975) proposed models for optimal platform location and scheduling of well drilling, but they did not include any production planning. Harding et al. (1996) tested sequential quadratic programming, simulated annealing, and a genetic algorithm for planning the multiple field development. They assumed a predetermined target production rate and prescheduled the capital investment within a fixed time window.

Iyer and Grossmann (1998) proposed multiperiod mixed-integer linear programming (MILP) model formulation for the planning and scheduling of investment and operation in offshore oil field facilities. The formulation employs a general objective function that optimizes a selected economic indicator (e.g., net present value). For a given planning horizon, the decision variables in the model are the choice of reservoirs to develop, selection from among candidate well sites, the well drilling and platform installation schedule, among others. A sequential decomposition strategy using aggregation of time periods and wells, followed by successive disaggregation, is proposed. However, they did not include the uncertainty in reserves.

Van den Heever and Grossmann (2000) proposed a multiperiod mixed-integer nonlinear programming (MINLP) model for offshore oilfield infrastructure planning

where nonlinear reservoir behavior is incorporated directly into the formulation. Discrete decisions include the selection of production platforms, well platforms and wells to be installed/drilled, and the drilling schedule for the wells over the planning horizon. Continuous decisions include the capacities of the platforms and the production profile for each well in each time period. For the solution of this model, an iterative aggregation/disaggregation algorithm is proposed in which logic-based methods, a bilevel decomposition technique, the use of convex envelopes, and aggregation of time periods are integrated.

CHAPTER A3

MODEL FORMULATION

A3.1 DETERMINISTIC MODEL FORMULATION FOR OILFIELD PRODUCTION PLANNING

The model is configured hierarchically in terms of sets to represent the physical configuration in a general form. The following sets, indices, variable, and parameters are defined.

A3.1.1 Nomenclature

(a) Sets and Indices

PP	set of production platforms
p	production platform $p \in \text{PP}$
WP(p)	set of well platforms connected to platform p
n	well platform $n \in \text{WP}(p)$
F	set of fields
f	field $f \in F$
$R(f)$	set of reservoirs associated with field f
r	reservoir $r \in R(f)$
$W_{\text{WP}}(n)$	set of wells connected to well platform n
$W_R(r)$	set of wells connected to reservoir r
$W_{\text{WP},R}(r,n)$	set of wells associated with reservoir r and well platform n
w	well $w \in W_{(\cdot)}(\cdot)$
t	time periods
τ	aggregated time periods
D	set of drilling rigs

k	drilling rigs $k \in D$
$J(r)$	linear interpolation pieces used for reservoir r
j	index of piece used in piecewise linear interpolation $j \in J$

(b) Continuous Variables

x_t	oil/gas flow in period t
\hat{x}_t	cumulative oil/gas flow amount up to period t
l_t	oil flow (mass) in period t
g_t	gas flow (volumetric) in period t
ϕ_t	gas-to-oil ratio in period t
v_t	pressure in period t
δ_t	pressure drop at choke in period t
d_t	design variable in period t
e_t	design expansion variable in period t
$Z_{k,t}$	number of times drilling rig k is moved in period t
$Z_{k,t}^p$	number of times drilling rig k is moved in period t
$\lambda_{j,t}$	interpolation variable in period t

(c) Binary Variables

z_t	= 1 if facility (well or platforms) is drilled/installed in period t
$zd_{k,t}$	= 1 if drilling rig k is located on facility in period t
$zf_{k,t}$	= 1 if the k th rig is located on well platform n first in period t
$zl_{k,t}$	= 1 if the k th rig is located on well platform n last in period t
$s_{k,t}$	= 1 (a slack) when only the n th platform is used for drilling using the k th rig in period t
$ZT_{k,t}$	= 1 if drilling rig k is moved after period t
$y_{j,t}$	= 1 if piece j used for linear interpolation in period t = otherwise

(d) Parameters

ρ	productivity index of well
P_{\max}	maximum pressure drop from well bore to well head
GOR_{\max}	maximum GOR (gas-to-oil ratio)
t_a	number of periods in an aggregated time period τ
T_a	number of aggregated time periods, $\tau = 1 \dots T_a$
M_w	maximum number of wells drilled by a rig in a time period
Δt	length of time period t
Ω^u	upper bound parameter (defined by the respective equation)
α	pressure drop coefficient for oil flow rate
β	pressure drop coefficient for GOR
c_{1t}	discounted revenue price coefficient for oil sales
c_{2t}	discounted fixed cost coefficient for capital investment
c_{3t}	discounted variable cost coefficient for capital investment
c_{4t}	discounted cost coefficient for moving rigs

(e) Superscripts

(w,n,p)	variables associated with well $w \in W$, with well platform n and production platform p
(n,p)	variables associated with well platform n and production platform p
(p)	variables associated with production platform p
(r)	variables associated with reservoir r

(f) Other Notations

$(.)$	formulation with fixed value of variable
$\langle(.)\rangle$	aggregation with respect to well variables and parameter

A3.1.2 Model Equations

(a) Material Balances

The sum of flow of oil/gas from all wells $w \in W_{WP}(n)$ associated with a well platform $n \in WP(p)$ is the total flow of oil/gas at the well platform.

$$\sum_{w \in W_{WP}(n)} x_{w,n,p,t} = x_{n,p,t} \quad \forall n \in WP(p), p \in PP, t \in T \quad (\text{A1})$$

Similarly, flows from well platforms are added to determine the flow at the production platforms.

$$\sum_{n \in WP(p)} x_{n,p,t} = x_{p,t} \quad \forall p \in PP \quad (\text{A2})$$

The total flow at period t is the sum of flows from all production platforms.

$$\sum_{p \in PP} x_t^p = x_t^{total} \quad (\text{A3})$$

for $t = 1 \dots T$

(b) Pressure Balances at Well Platforms and Production Platforms

The pressure at the well platform $n \in WP(p)$ is the pressure at the wells $w \in W_{WP}(n)$ associated with well platform n minus the pressure drop in the corresponding pipe. In reality, the flowrate of oil/gas is equals to square root of the pressure drop. Here, pressure drop is expressed as a linear function of the oil and gas flow rate for simplicity. Here, δ is defined as the pressure drop across the pressure chokes on the lines.

$$v_{n,p,t} = v_{w,n,p,t} - \alpha x_{w,n,p,t} - \beta g_{w,n,p,t} - \delta_{w,n,p,t} \quad \forall w \in W_{WP}(n), n \in WP(p), p \in PP \quad (\text{A4})$$

The pressure at the production platform $p \in PP$ is the pressure at the well platform $n \in WP(p)$ associated with production platform p minus the pressure drop in the corresponding pipe.

$$v_p = v_{n,p} - \alpha x_{n,p} - \beta g_{n,p} - \delta_{n,p}, \quad \forall n \in WP(p), p \in PP \quad (\text{A5})$$

for $t = 1 \dots T$

(c) Flow Constraints in Wells

The oil/gas flow in period t is related to the maximum flow of oil ($l_{w,n,p,t}$) and the maximum gas flow ($g_{w,n,p,t}$):

$$x_{w,n,p,t} = l_{w,n,p,t} + g_{w,n,p,t} \quad (\text{A6})$$

The maximum flow of oil is related to the PI of the well (ρ) and the allowable pressure drop (P_{\max}):

$$l_{w,n,p,t} \leq \rho_{w,n,p,t} P_{\max} \quad (\text{A7})$$

The maximum gas flow is limited by the maximum flow of oil ($l_{w,n,p,t}$) and maximum allowable GOR (GOR_{\max}):

$$g_{w,n,p,t} \leq l_{w,n,p,t} \text{GOR}_{\max} \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP} \quad (\text{A8})$$

for $t = 1 \dots T$

(d) Cumulative Flow Amount from Wells up to a Certain Period, θ

The cumulative amount of oil from a well is calculated from the sum of the amount of oil from all periods up to time period, θ . Thus,

$$\hat{x}_{w,n,p,\theta} = \sum_{t=1}^{\theta-1} x_{w,n,p,t} \Delta t \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP} \quad (\text{A9})$$

for $\theta = 1, \dots, T$

For instance, when $\theta = 1$: $\hat{x}_{w,n,p,\theta} = 0$;

$$\text{when } \theta = 5: \hat{x}_{w,n,p,\theta} = \sum_{t=1}^4 x_{w,n,p,t} \Delta t$$

The cumulative oil amount from a reservoir is obtained by summing the cumulative amounts from all wells associated with the reservoir. We define a new set as follows:

$$W_{F,R}(f,r) \{ (w,n,p) \mid w \in W_{WP}(n) \cap W_R(r), n \in WP(p), p \in PP \} \quad \forall r \in R(f), f \in F$$

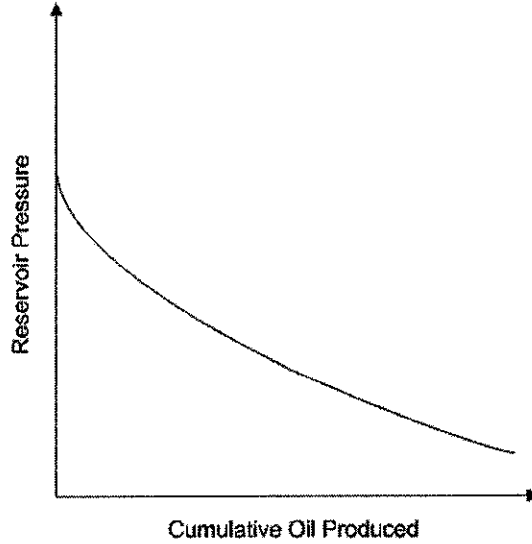
Thus, the cumulative amount of oil from a well

$$\hat{x}_{r,f,\theta} = \sum_{(w,n,p) \in W_{F,R}(f,r)} \hat{x}_{w,n,p,\theta} \quad \forall r \in R, f \in F \quad (\text{A10})$$

for $\theta = 1, \dots, T$

(e) Piecewise Linear Interpolation at Well and Reservoir Level

The pressure and GOR in a reservoir is a nonlinear function of the cumulative oil produced, as can be seen in Figure A3.1 below:



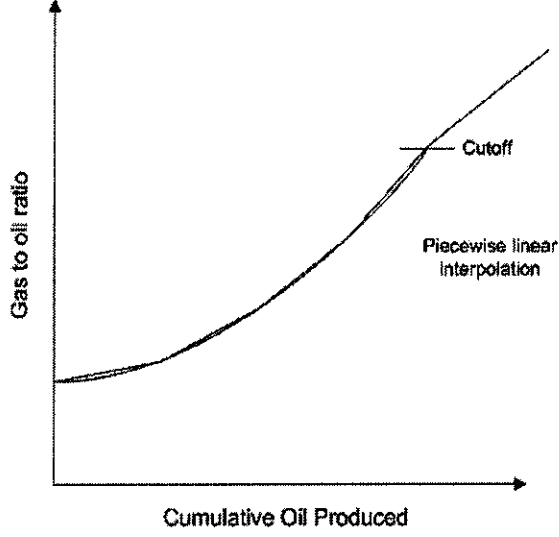


FIGURE A3.1: Reservoir performance characteristics

The actual relation is taken to be a function of the initial pressure, the oil's PVT properties, and the initial oil volume. The pressure and GOR are then calculated using piecewise linear interpolation.

The pressure in a reservoir $r \in R(f)$ is as shown below:

$$v_{r,f,t} = h_1(\hat{x}_{r,f,t}, y_{r,f,j,t}, \lambda_{r,f,j,t}, \vartheta_{r,f,t}^p) \quad \forall r \in R(f), f \in F, t = 1, L, T \quad (\text{A11})$$

The oil and gas flow rates from individual wells in a reservoir are also calculated using piecewise linear interpolation:

$$x_t^{w,\pi,p} = h_1(\hat{x}_t^{w,\pi,p}, y_{j,t}^{w,\pi,p}, \lambda_{j,t}^{w,\pi,p}, \vartheta_t^{w,\pi,p}) \quad (\text{A12})$$

The GOR in period t is as shown below:

$$\phi_{w,n,p,t} = h_1(\hat{x}_{w,n,p,t}, y_{w,n,p,j,t}, \lambda_{w,n,p,j,t}, \vartheta_{w,n,p,t}^p) \quad (\text{A13})$$

$$\forall w \in W_{WP}(n), n \in WP(p), p \in PP, t = 1, L, T$$

It is vital to model the reservoir performance for us to capture the dynamic conditions in the reservoir. The reservoir pressure and GOR is a function of the cumulative amount of oil removed (see Figure A3.1). This performance curve is nonlinear. In this work, it will be approximated using piecewise linear interpolation. The choice of the linear approximation piece used for interpolation is made using binary variables $y_{r,f,j,t}$.

The value of the interpolated variable as a convex combination of $\theta_{r,f,j,t}^o$ and $\theta_{r,f,j-1,t}^o$ are determined by:

$$v_{r,f,t} = \sum_{j \in J(r)} \lambda_{r,f,j,t} \theta_{r,f,j,t}^o \quad (\text{A14})$$

The constraint on the interpolation variable $\lambda_{r,f,j,t}$ is as shown below:

$$\sum_{j \in J(r)} \lambda_{r,f,j,t} = 1 \quad (\text{A15})$$

From the following equation, $\lambda_{r,f,j,t}$ can be non-zero for only two consecutive j . Thus, the corresponding j th piece is used for linear interpolation as all other $\lambda_{r,f,j,t} = 0$.

$$\lambda_{r,f,j,t} \leq y_{r,f,j,t} + y_{r,f,j-1,t} \quad (\text{A16})$$

The following equation allows only one index j for which $y_{r,f,j,t} = 1$. Note that the variables y may be treated as special ordered sets (SOS) variables over the set j .

$$\sum_j y_{r,f,j,t} = 1 \quad (\text{A17})$$

(f) Logical Constraints for Installation and Flow from Facilities

The wells may be drilled only once in one of the time periods.

$$\sum_{t=1}^T z_{w,n,p,t} \leq 1 \quad \forall w \in W_{WP}(n), n \in WP(p), p \in PP \quad (\text{A18})$$

The well platforms may be installed only in one of the time periods.

$$\sum_{t=1}^T z_{n,p,t} \leq 1 \quad \forall n \in WP(p), p \in PP \quad (\text{A19})$$

The production platforms may be installed only in one of the time periods.

$$\sum_{t=1}^T z_{p,t} \leq 1 \quad \forall p \in PP \quad (\text{A20})$$

for $t = 1 \dots T$

Note that the variables z_t may be treated as special ordered sets (SOS) by introducing a dummy period t_d such that

$$\sum_{t=1}^T z_t + z_{t_d} = 1 \quad (\text{A21})$$

By treating the investment binaries as SOS variables, one can potentially reduce the number of nodes enumerated in a branch and bound tree.

The flow of oil and gas from a facility can be nonzero only after it is installed.

For the flow from a well w to a well platform n :

$$x_{w,n,p,\theta} \leq \Omega^\theta \sum_{t=1}^{\theta} z_{w,n,p,t} \quad \forall w \in W_{WP}(n), n \in WP(p), p \in PP \quad (\text{A22})$$

For the flow from a well platform n to a production platform p :

$$x_{n,p,\theta} \leq \Omega^u \sum_{t=1}^{\theta} z_{n,p,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A23})$$

For the flow from a production platform p to shore for sales:

$$x_{p,\theta} \leq \Omega^u \sum_{t=1}^{\theta} z_{p,t} \quad \forall p \in \text{PP} \quad (\text{A24})$$

for $t = 1 \dots T$

Note that the upper bound Ω^u is defined accordingly based on the variables in the equation. In eq 22, Ω^u is equal to the maximum well oil flow rate $x_{w,n,p,\theta}$. The well platform associated with a well must be installed before drilling that well.

$$z_{w,n,p,\theta} \leq \sum_{t=1}^{\theta} z_{n,p,t} \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP} \quad (\text{A25})$$

Similarly, production platforms must be installed before the associated well platforms.

$$z_{n,p,\theta} \leq \sum_{t=1}^{\theta} z_{p,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A26})$$

for $t = 1 \dots T$

(g) Design of Facility

The design capacity of a facility is determined by the maximum flow for all periods $t = 1, \dots, T$. The maximum flow at well platform is:

$$x_{n,p,t} \leq d_{n,p,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A27})$$

Similarly, for production platform:

$$x_{p,t} \leq d_{p,t} \quad \forall p \in \text{PP} \quad (\text{A28})$$

To model the variable design cost using linear terms, a design *expansion* variable is used, which is equal to the design capacity for the time period when the facility is installed. The design variable for well platform is:

$$d_{n,p,t} = d_{n,p,t-1} + e_{n,p,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A29})$$

Similarly, for production platform:

$$d_{p,t} = d_{p,t-1} + e_{p,t} \quad \forall p \in \text{PP} \quad (\text{A30})$$

The expansion variable for well platform is:

$$e_{n,p,t} \leq \Omega^u z_{n,p,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A31})$$

Similarly, for production platform:

$$e_{p,t} \leq \Omega^u z_{p,t} \quad \forall p \in \text{PP} \quad (\text{A32})$$

Note that the expansion variable e may be nonzero only in one time period (from eqs 19 and 31 for well platforms, and from eqs 20 and 32 for production platforms), which represents the period when the facility is installed.

(h) Number of Wells Drilled in Time Period t

The maximum number of wells that can be drilled in a time period (M_W) is dependent on the length of the time period (t), the time taken by a drilling rig to drill a well, and the number of available drilling rigs.

$$\sum_{w \in \mathcal{W}_{\text{WP}}(n)} z_{w,n,p,t} \leq h_2(Z_{p,k,t}, ZT_{n,p,k,t}, M_W) \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A33})$$

The following equations are used to model the resource constraints associated with drilling rigs. The main issues involved in the resource constraints are summarized below. Wells are drilled from the associated well platforms using drilling rigs.

However, there are only a limited number of drilling rigs (usually less than the number of well platforms) available in any time period for drilling a well. Besides the availability of the rig, the physical location of the rig on the well platform also determines which wells may be drilled in any time period. Thus, the scheduling of the movement of rigs between well platforms is essential to ensure its availability. The movement of drilling rigs across well platforms also leads to a fixed charge and a loss of time in the availability of the rig.

This loss of time is accounted for in eq 32, where the number of wells drilled in a time period is a function of the number of moves of the rigs across well platforms.

If any well associated with well platform n is drilled in period t , then $\sum_k z d_{n,p,k,t}$ must be 1 in that time period, implying at least one drilling rig is located on that well platform:

$$z_{w,n,p,t} \leq \sum_{k \in D} z d_{n,p,k,t} \quad \forall w \in W_{WP}(n), n \in WP(p), p \in PP \quad (\text{A34})$$

If in any period, if drilling rig k is located first on well platform n , then $z^f = 1$.

The sum of z^f and z^l must be less than 1 when the rig is located on more than one well platform in the same time period. However, if all wells are drilled from the same well platform in a time period, then both z^f and z^l will be 1. To model both possibilities, a slack binary variable $s_{p,k,t}$ is used.

The k th rig is located on the n th platform either first or last in that time period t (when $s_{p,k,t} = 0$). The location of the rigs on the well platforms is modeled by the following equations:

$$zf_{n,p,k,t} + zl_{n,p,k,t} \leq 1 + s_{p,k,t} \quad \forall n \in W_{WP}(n), n \in WP(p), p \in PP \quad (A35)$$

The slack variable takes a non-zero value if all wells are drilled from a single n th platform in that time period. Thus, $s_{p,k,t}$ takes a value of 1 (a slack) when only the n th platform is used for drilling using the k th rig in period t . If both zf and zl are 1, then the slack is non-zero ($= 1$):

$$\left(\sum_{n \in WP(p)} zd_{n,p,k,t} - 1 \right) \leq \Omega^u (1 - s_{p,k,t}) \quad \forall n \in WP(p), p \in PP \quad (A36)$$

The binary variables zf and zl are non-zero only if a well is drilled from well platform n . Thus, zf and zl are zero when the k th drilling rig is not used to drill from platform n :

$$zf_{n,p,k,t} \leq zd_{n,p,k,t} \quad \forall n \in WP(p), p \in PP \quad (A37)$$

$$zl_{n,p,k,t} \leq zd_{n,p,k,t} \quad \forall n \in WP(p), p \in PP \quad (A38)$$

Any move of rig k from period $t - 1$ to t is calculated from the value of zf_t and zl_{t-1} . The minimum number of moves required within a time period is then calculated from the number of well platforms on which rig k is located in that period.

The movement of the k th rig from platform n across time periods based on whether the rig is located first in period t on n and last in period $t - 1$ is determined by:

$$zf_{n,p,k,t} - zl_{n,p,k,t-1} \leq ZT_{n,p,k,t} \quad \forall n \in WP(p), p \in PP$$

$$\left(zf_{n,p,k,t}, zl_{n,p,k,t-1}, ZT_{n,p,k,t} \right) = (0, 0, 0), (0, 0, 1), (1, 0, 1), (0, 1, 0), (0, 1, 1), (1, 1, 0), (1, 1, 1) \quad (A39)$$

The movement of the rigs within a time period from the number of well platforms used for drilling in that period is determined by:

$$\left(\sum_n zd_{n,p,k,t} - 1\right) \leq Z_{p,k,t} \quad \forall n \in \text{WP}(p), p \in \text{PP} \quad (\text{A40})$$

$$zd_{w1,p,k,t}, zd_{w2,p,k,t} = (0,0), (1,0), (0,1), (1,1)$$

(i) Drilling Rig Scheduling Constraints

The number of times a drilling rig is moved depends on the allocation of drilling rigs to well platforms for drilling wells.

$$Z_{p,k,t} = h_3(z_{w,n,p,t}) \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP} \quad (\text{A41})$$

(j) Flow Profile Constraints

An additional operational constraint is that the flows should be a non-increasing function of time in order to ensure a smooth flow profile.

$$x_{w,n,p,t} \geq x_{w,n,p,t+1} - \Omega^n \left(1 - \sum_{t'=1}^t z_{w,n,p,t'}\right) \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP} \quad (\text{A42})$$

(k) Objective Function

$$\begin{aligned} \max \Psi = & \sum_{1^t} c_{1,t} x_{\text{total},t} - \sum_{1^t} \sum_{p \in \text{PP}} c_{p,2,t} z_{p,t} + c_{p,3,t} e_{p,t} \\ & \text{income from sales} \quad \text{investment costs of the production platform} \\ & - \sum_{1^t} \sum_{p \in \text{PP}} \sum_{n \in \text{WP}(p)} c_{n,p,2,t} z_{n,p,t} + c_{n,p,3,t} e_{n,p,t} \\ & \text{investment costs of the well platform} \\ & - \sum_{1^t} \sum_{p \in \text{PP}} \sum_{n \in \text{WP}(p)} \sum_{w \in W_{\text{WP}}(p)} c_{w,n,p,2,t} z_{w,n,p,t} \\ & \text{drilling costs of wells} \\ & - \left(\sum_{1^t} c_{4,t} \sum_{k \in D} \sum_{p \in \text{PP}} z_{p,k,t} + \sum_{1^t} c_{4,t} \sum_{k \in D} \sum_{p \in \text{PP}} \sum_{n \in \text{WP}(p)} ZT_{n,p,k,t} \right) \\ & \text{costs for moving the drilling rigs} \end{aligned} \quad (\text{A43})$$

In the equation above, the first sum corresponds to the income from sales, the second and third terms are the investment costs of the production and well platforms, respectively, the fourth term represents the drilling costs of wells, and the last term is the sum of costs for moving the drilling rigs.

For our computation, we consider an oil field with 4 wells (W1, W2, W3, W4), 1 well platform (N1), and 1 production platforms (P1). We consider 6 time periods for our computation (T1 to T6). The data for computation is referred to Iyer et al. (1998) paper, and it is as shown in Table A3.1 below:

TABLE A3.1: Data for production planning computation

Description	Unit	Symbol	Facility	T1	T2	T3	T4	T5	T6
Productivity index of well	NA	ρ	W1	47.6	TBD	TBD	TBD	TBD	TBD
			W2	83.8	TBD	TBD	TBD	TBD	TBD
			W3	158.4	TBD	TBD	TBD	TBD	TBD
			W4	210.2	TBD	TBD	TBD	TBD	TBD
Discounted revenue price coefficient for oil sales	\$/barrel	$c_{1,t}$	N/A	22	22	22	22	22	22
Discounted fixed cost coefficient for capital investment	million \$	$c_{p,2,t}$	P1	70	69	68	67	66	65
		$c_{n,p,2,t}$	N1	11	10	9	8	7	6
		$c_{w,n,p,2,t}$	W1	6.69	6.69	6.69	6.69	6.69	6.69
			W2	6.34	6.34	6.34	6.34	6.34	6.34
			W3	5.76	5.76	5.76	5.76	5.76	5.76
			W4	5.83	5.83	5.83	5.83	5.83	5.83
Discounted variable cost coefficient for capital investment	million \$	$c_{p,3,t}$	P1	0.47	0.46	0.45	0.44	0.43	0.42
		$c_{n,p,3,t}$	N1	0.19	0.18	0.17	0.16	0.15	0.14
Discounted cost coefficient for moving rigs	million \$	$c_{4,t}$	N/A	50	49	48	47	46	45

*TBD = to be determined (Using piecewise linear interpolation)

A3.2 STOCHASTIC MODEL FORMULATION FOR OILFIELD PRODUCTION PLANNING

A3.2.1. Exogeneous uncertainty – price

Price is a type of exogeneous uncertainty since it is beyond our control and depends on actual market developments and its associated economics. The price of oil will not be affected by our operations.

A mixed integer programming model for optimal development of an oil field under uncertain future oil prices was developed earlier (Jonsbraten, 1998). A finite set of oil price scenarios with associated probabilities was given, and the scenario and policy aggregation technique developed by Rockafellar and Wets was used to solve the problem.

A near replica of earlier work will be attempted here, by taking into account the exogenous uncertainty of price. A set of scenarios are generated, namely 1, 2, and 3. These scenarios represent the price scenarios, each with their own probability values.

The prices of oil for each scenario with their probabilities are shown in Table A3.2 below:

TABLE A3.2: Oil price scenarios

Scenario, S	Oil price, \$/barrel	Probability
1	17	0.4
2	19	0.3
3	22	0.3

The values for the oil prices are not in accordance to the current price (oil in year 2009 is around USD50 per barrel). This is because the values used in this work are taken from Jonsbraten (1998).

The formulation for the objective function has to take into account the oil price for different scenarios, with their own values of probability, by adding another subscript to discounted revenue price coefficient for oil sales, c_{1s} .

A3.2.2. Endogeneous uncertainty – reserves

Reserves are considered to be a type of endogeneous uncertainty. This is due to the fact that the accuracy of our predictions of reserves values can only be assessed after we have begun our operations.

Optimal investment and operational planning of gas field developments under uncertainty in gas reserves has been considered by Goel, 2004. A novel stochastic programming model that incorporates the decision – dependence of the scenario tree is presented. In the paper, they take into account the uncertainty of size and initial deliverability in the fields of the gas fields. The problem has been formulated as a multistage stochastic program.

The problem of optimal investment and operational planning for development of gas fields under uncertainty in gas reserves have been attempted (Goel, 2006). This takes the same type of uncertainty as the Goel, 2004 paper, which is size and initial deliverabilities of the gas fields. A Lagrangean duality based branch and bound algorithm to solve the stochastic programming model was presented.

The same approach will be taken in our consideration of endogeneous uncertainty of reserves, which are the size and initial deliverabilities. The difference here is that we are attempting on an oil well platform, whereas previous works were on gas fields.

(a) Model equations

The oil reserves of a field are characterized by the “size” and “deliverability” of the field or well platform. The size of a well platform refers to the total amount of oil that can be recovered from the wells connected to it, while the deliverability of a well platform at any time is the maximum rate of oil production that can be obtained from the wells connected to it. The deliverability of a well platform is highest (initial deliverability) when no oil has been recovered from the wells connected to it, and decreases with increase in cumulative production from the wells. When the cumulative production from the wells equals the size of the well platform, the deliverability reduces to zero and hence, no more oil can be produced. In this paper,

we assume that the deliverability of a well decreases linearly with the increase in cumulative production from the well platform. Thus, we assume a linear reservoir model, as shown in Figure A3.2 below:

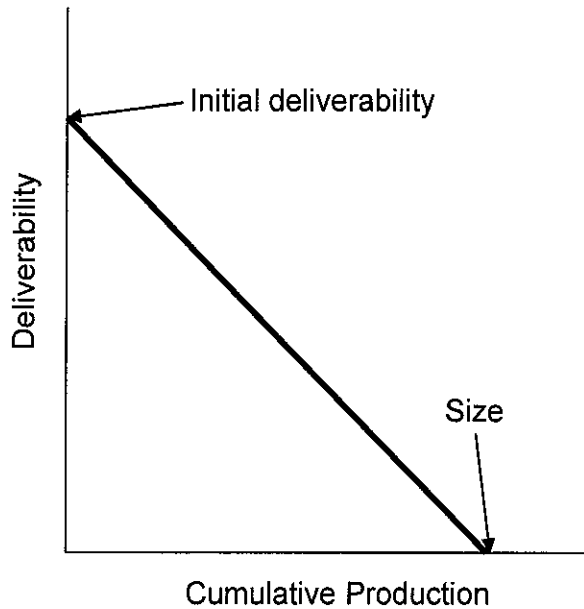


FIGURE A3.2: Linear reservoir model for an oil well platform

Note that in reality, reservoir behavior is characterized by a complex system of partial differential equations. In the planning process, this behavior is frequently approximated by simplified algebraic models. For the sake of simplicity, we have used a linear model to approximate the reservoir behavior.

We assume discrete probability distributions for the sizes and initial deliverabilities of all wells/well platforms. Let $\theta_{\text{Size},w,n,p,t,s}$ and $\theta_{\text{ID},w,n,p,t,s}$ represent the set of possible realizations in the distributions for the size and initial deliverability, respectively, of well w . the overall set of possibilities is represented by a set of “scenarios”, where each scenario is one possible combination of values for the sizes and initial deliverabilities of all fields, and has a given probability. We assume that the set of scenarios consists of every possible combination of realizations for the sizes and initial deliverabilities of various wells. Figure A3.3 shows the nine possible reservoir models for a well with three realizations each for its size and its initial deliverability.

For a problem where decisions have to be made only for this field, each of the nine reservoir models shown in Figure A3.3 corresponds to a scenario.

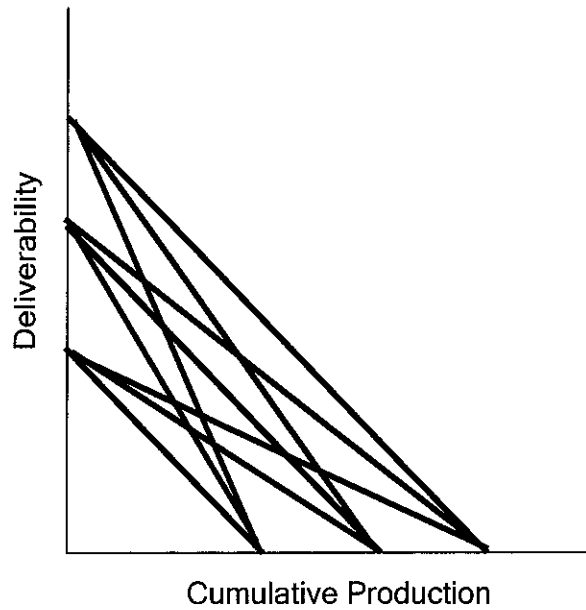


FIGURE A3.3: Nine scenarios arising from uncertainty in size and initial deliverability of well

The scenarios with their respective probability values (in brackets) are shown in Figure A3.4 below:

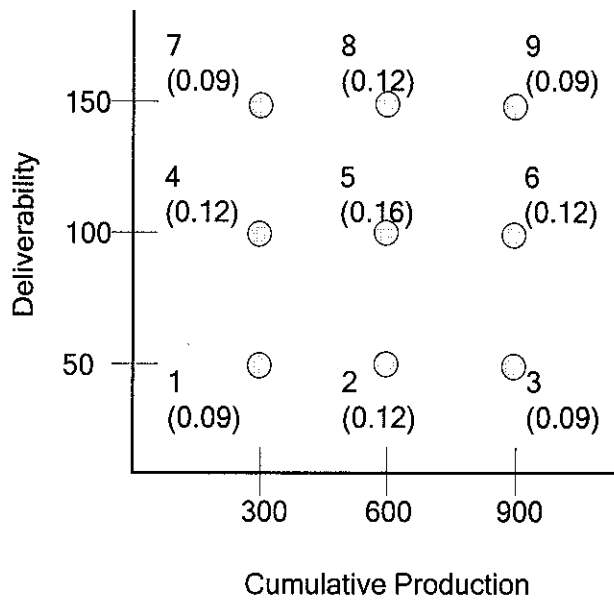


FIGURE A3.4: Nine scenarios with their probability values

The scenarios, along with their values for size, initial deliverability, and probability, is shown in Table A3.3 below:

TABLE A3.3: Size, initial deliverability and probability for each scenario

Scenario	Size	Initial deliverability	Probability
1	300	50	0.09
2	600	50	0.12
3	900	50	0.09
4	300	100	0.12
5	600	100	0.16
6	900	100	0.12
7	300	150	0.09
8	600	150	0.12
9	900	150	0.09

To model the scenarios, a new equation is introduced that takes relates the deliverability of oil from a well, i.e. the maximum oil flow from a well, $x_{D,w,n,p,t,s}$, the cumulative oil flow up to period t , $\hat{x}_{w,n,p,t,s}$, the size of the well, $\theta_{Size,w,n,p,t,s}$, and the initial deliverability of the well, $\theta_{ID,w,n,p,t,s}$. This equation is derived from Figure

A3.2, where deliverability represents oil flow from a well. The steps are as shown below:

- Slope of the line, $m = -\frac{\theta_{ID,w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}}$
- Put it in the form $y = mx + c$:

$$x_{D,w,n,p,t,s} = -\frac{\theta_{ID,w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}} \hat{x}_{w,n,p,t,s} + c \quad (1)$$

- Taking the coordinate of $\hat{x}_{w,n,p,t,s} = 0$, and $x_{D,w,n,p,t,s} = \text{initial deliverability} = \theta_{ID,w,n,p,t,s}$, substituting into (1), and solving:

$$\theta_{ID,w,n,p,t,s} = -\frac{\theta_{ID,w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}} (0) + c$$

$$c = \theta_{ID,w,n,p,t,s}$$

- Substituting values of m and c into (1), and rearranging:

$$x_{D,w,n,p,t,s} = -\frac{\theta_{ID,w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}} \hat{x}_{w,n,p,t,s} + \theta_{ID,w,n,p,t,s}$$

$$x_{D,w,n,p,t,s} + \frac{\theta_{ID,w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}} \hat{x}_{w,n,p,t,s} = \theta_{ID,w,n,p,t,s}$$

- Dividing both LHS and RHS by $\theta_{ID,w,n,p,t,s}$:

$$\frac{x_{D,w,n,p,t,s}}{\theta_{ID,w,n,p,t,s}} + \frac{\hat{x}_{w,n,p,t,s}}{\theta_{Size,w,n,p,t,s}} = 1 \quad (A44)$$

Deliverability is the maximum oil flow from a well, so the oil flow from a well must not be greater than the deliverability of the well at that time and scenario:

$$x_{w,n,p,t,s} \leq x_{D,w,n,p,t,s} \quad (\text{A45})$$

The cumulative oil production from a well at most can be equal to the size of the well:

$$\hat{x}_{w,n,p,t,s} \leq \theta_{\text{Size},w,n,p,t,s} \quad (\text{A46})$$

The objective function, Ψ , has to account for all the scenarios. It is to maximize the expected profit, which is the probability weighted average of the profits over all scenarios:

$$\max \psi = \sum_s \psi_s p_s \quad (\text{A47})$$

(b) Non-anticipativity constraints

Decisions for different scenarios are linked to each other by non-anticipativity constraints. These constraints ensure that decisions at any time are based only on information available at that time, and not on foresight. Scenarios s, s' are said to be indistinguishable at some time t if s, s' are identical in realizations for all parameters in which uncertainty has been resolved up till that time. The non-anticipativity rule states that if two scenarios are indistinguishable at some time, then decisions at that time should be the same in the two scenarios.

The decision variables involved in our model are $z_{w,n,p,t,s}, z_{n,p,t,s}, z_{p,t,s}, z_{d_{n,p,k,t,s}}, ZI_{w,n,p,t,s}, x_{w,n,p,t,s}, e_{n,p,t,s}$ and $e_{n,p,t,s}$. The non-anticipativity constraints are as shown below:

$$z_{w,n,p,t,s} = z_{w,n,p,t,s'}, \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A48})$$

$$z_{n,p,t,s} = z_{n,p,t,s'}, \quad \forall n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A49})$$

$$z_{p,t,s} = z_{p,t,s'} \quad \forall p \in \text{PP}, s < s' \quad (\text{A50})$$

$$zd_{n,p,k,t,s} = zd_{n,p,k,t,s'} \quad \forall n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A51})$$

$$ZT_{w,n,p,t,s} = ZT_{w,n,p,t,s'} \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A52})$$

$$x_{w,n,p,t,s} = x_{w,n,p,t,s'} \quad \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A53})$$

$$e_{n,p,t,s} = e_{n,p,t,s'} \quad \forall n \in \text{WP}(p), p \in \text{PP}, s < s' \quad (\text{A54})$$

$$e_{p,t,s} = e_{p,t,s'} \quad \forall p \in \text{PP}, s < s' \quad (\text{A55})$$

Note that non-anticipativity constraints are imposed only on the decision variables. All other variables are “state variable” that can be eliminated from the model. Hence, non-anticipativity is not imposed on those variables. Also, to avoid duplication of non-anticipativity constraints for a pair of scenarios, constraints (A48) – (A55) are applied for (s, s') only if $s < s'$.

The domains for the variables are specified as below:

$$\begin{aligned} z_{w,n,p,t,s} &\in \{0,1\} & \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s; \\ z_{n,p,t,s} &\in \{0,1\} & \forall n \in \text{WP}(p), p \in \text{PP}, s; \\ z_{p,t,s} &\in \{0,1\} & \forall p \in \text{PP}, s; \\ zd_{n,p,k,t,s} &\in \{0,1\} & \forall n \in \text{WP}(p), p \in \text{PP}, s; \\ ZT_{w,n,p,t,s} &\in \{0,1\} & \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s; \\ x_{w,n,p,t,s} &\geq 0 & \forall w \in W_{\text{WP}}(n), n \in \text{WP}(p), p \in \text{PP}, s; \\ e_{n,p,t,s} &\geq 0 & \forall n \in \text{WP}(p), p \in \text{PP}, s; \\ e_{p,t,s} &\geq 0 & \forall p \in \text{PP}, s \end{aligned}$$

A3.2.3 Scenario Generation

Given two random variables X and Y, the joint distribution of X and Y defines the probability of events defined in terms of both X and Y. The sum of all joint probabilities must be equal to 1. If discrete random variables X and Y are independent of each other, the joint distribution, i.e. the joint probability, is the probability of event Y occurring at the same time event X occurs, are formulated as below:

$$P(X = x \text{ and } Y = y) = P(X = x).P(Y = y) \quad (\text{A55})$$

Since we have two separate uncertainties for our model, i.e., price and reserves, they need to be linked to each other to generate practical scenarios. Joint probability distribution will be used to link between the scenarios, since they are discrete random variables independent of each other. The calculation for joint probability is shown in Table A3.4 below:

TABLE A3.4: Joint probability distribution calculations for scenario generation

Size	Reserves		Price (\$/barrel)			Total Probability
			17	19	22	
	Initial Deliverability	Probability	0.4	0.3	0.3	1
300	50	0.09	0.036	0.027	0.027	0.09
600	50	0.12	0.048	0.036	0.036	0.12
900	50	0.09	0.036	0.027	0.027	0.09
300	100	0.12	0.048	0.036	0.036	0.12
600	100	0.16	0.064	0.048	0.048	0.16
900	100	0.12	0.048	0.036	0.036	0.12
300	150	0.09	0.036	0.027	0.027	0.09
600	150	0.12	0.048	0.036	0.036	0.12
900	150	0.09	0.036	0.027	0.027	0.09
Total Probability		1	0.4	0.3	0.3	1

CHAPTER A4

COMPUTATIONAL EXPERIMENTS AND DISCUSSIONS ON NUMERICAL RESULTS

A4.1 COMPUTATIONAL RESULTS

The formulated logical constraints are coded on GAMS. Please refer to Appendix II and IV for the GAMS input file code and Appendix III and V for the output file containing the result generated by GAMS/CPLEX 10.

Numerical studies and computational experiments of the proposed model in this paper is implemented on GAMS 22.3 for Windows XP platform. The model is solved using the branch-and-cut algorithms available in GAMS/CPLEX 10 on the computing facilities with attributes listed in Table A4.1 below:

TABLE A4.1: Attributes of computing facilities for computational experiments

Computer Type	Laptop (Lenovo)
Processor Type	Intel Centrino Duo
Processor Speed	1.73 GHz
RAM	512 MB

The computational statistics including the CPU times on the mentioned machine are as reported in Table A4.2 below:

TABLE A4.2: Computational statistics for the deterministic and stochastic models

Solver	Model	Number of single continuous variables	Number of discrete (binary 0–1) variables	Number of single equations	Resource usage/ CPU time (s)	Number of iterations
CPLEX 10	Deterministic	386	114	446	0.468	67
	Stochastic	3619	1026	5125	0.500	245

The result of the deterministic model is shown in Table A4.3 below:

TABLE A4.3: Deterministic model results

Well/Time	T1	T2	T3	T4	T5	T6
W1	0	0	0	0	0	1
W2	1	0	0	0	0	0
W3	0	0	1	0	0	0
W4	0	1	0	0	0	0

The result of the stochastic model is shown in Table A4.4 below:

TABLE A4.4: Stochastic model results

Time		T1				T2				T3			
Well		W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Scenario	S1	0	1	0	0	0	0	1	0	1	0	0	0
	S2	0	1	0	0	0	0	1	0	1	0	0	0
	S3	0	1	0	0	0	0	1	0	1	0	0	0
	S4	0	1	0	0	0	0	1	0	1	0	0	0
	S5	0	1	0	0	0	0	1	0	1	0	0	0
	S6	0	1	0	0	0	0	1	0	1	0	0	0
	S7	0	1	0	0	0	0	1	0	1	0	0	0
	S8	0	1	0	0	0	0	1	0	1	0	0	0
	S9	0	1	0	0	0	0	1	0	1	0	0	0

Time		T4				T5				T6			
Well		W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Scenario	S1	0	0	0	0	0	0	0	1	0	0	0	0
	S2	0	0	0	0	0	0	0	1	0	0	0	0
	S3	0	0	0	0	0	0	0	1	0	0	0	0
	S4	0	0	0	0	0	0	0	1	0	0	0	0
	S5	0	0	0	0	0	0	0	1	0	0	0	0
	S6	0	0	0	0	0	0	0	1	0	0	0	0
	S7	0	0	0	0	0	0	0	1	0	0	0	0
	S8	0	0	0	0	0	0	0	1	0	0	0	0
	S9	0	0	0	0	0	0	0	1	0	0	0	0

A summary of the computational results obtained from the deterministic and stochastic models are shown in Table A4.5 below:

TABLE A4.5: Computational results for the deterministic and stochastic models

No.	Description	Symbol	Unit	Deterministic Model	Stochastic Model
1	Objective function (Profit)	Z	Million dollars	331.08	145.58
2	Wells drilled	Z1	-	W2 (T1) W4 (T2) W3 (T3) W1 (T6)	W2 (T1) W3 (T2) W1 (T3) W4 (T5)
3	Well platforms installed	Z2	-	N1 (in T1, T2, T3, T6)	N1 (in T1, T2, T3, T5)
4	Production platforms installed	Z3	-	P1 (in T1, T2, T3, T6)	P1 (in T1, T2, T3, T5)
5	Oil flow for wells	X1	Million barrel	5MMbbls per well	5MMbbls per well
6	Oil flow for well platforms	X2	Million barrel	5MMbbls from N1 (in T1, T2, T3, T6)	5MMbbls from N1 (in T1, T2, T3, T5)
7	Oil flow for production platforms	X3	Million barrel	5MMbbls from P1 (in T1, T2, T3, T6)	5MMbbls from P1 (in T1, T2, T3, T5)

A4.2 DISCUSSION

From the results obtained, it can be seen not only the costs alone are taken into account to choose which well to be drilled. Other factors include the productivity index of the well, the size, and the initial deliverability of the wells. In order to maximize the profit, which is the objective function of the model, it will select the facilities with lower cost, and also taking into account the productivity of oil from the wells. As can be seen from the deterministic model, well 2 (W1) is chosen to be drilled at time period 1 (T1). This can be attributed to the fact that it has higher productivity index compared to W1, and also a lower cost of drilling. Besides that, it is chosen to be drilled at T1 because the profit for sales of oil at T1 is the highest.

Also, it can be seen that, the oil flows have non-zero values only for wells, well and productions platforms that are chosen to be installed. This is in accordance to the fact that, oil can only flow from wells, well or productions platforms that are either drilled or installed, enforced by the constraints in Chapter A3.

An observation on the results of the stochastic model shows that the non-anticipativity constraints are enforced, since all the scenarios are the same for the

same time period T , as can be seen from Figure A4.2. This non-anticipativity constraint is an important part of our model because it ensures that decisions at any time are based only on information available at that time, and not on foresight. Our results are in accordance to the non-anticipativity rule, which states that if two scenarios are indistinguishable at some time, then decisions at that time should be the same in the two scenarios, as the case with our results.

The objective function obtained from the stochastic model is 145.58 million dollars profit, lower than the value obtained from the deterministic model (331.08 million dollars). In this case, the stochastic model is a better representation of the problem. This is because, it takes into account the various scenarios associated with uncertainties in price, size and initial deliverability of the well. Each of the scenarios has their own unique probability values. The representation of the problem is more thorough since the objective function has to account for all the scenarios by taking the weighted average of the profits over all scenarios.

CHAPTER A5

CONCLUSIONS AND RECOMMENDATIONS

A5.1 CONCLUSIONS

An optimization model on the infrastructures planning for an offshore oil field was developed with the objective function of maximizing profit, subject to constraints of mass balance, pressure balance, and flow constraints, among others. Two models were proposed, namely the deterministic model and the stochastic model. The deterministic model does not possess uncertain parameters, whereas the stochastic model takes into account the uncertainties in price, size and initial deliverabilities. Since this is a maximization problem, the model selected the facility that returns the highest profit. Thus, the objective function has been achieved. The mathematical model was developed using GAMS and an optimal solution was found with no logical constraints conflicts or error. Although the stochastic model obtained an objective value lower than the deterministic model, it is a more representative model as it takes into account the uncertainties along with their associated probabilities. The model can be used as a basic tool for upper-management to decide on refinery technologies to be applied in a grass-root refinery.

A5.2. RECOMMENDATIONS

It is recommended that real data are taken for the data input into the model formulation. This will enable a better representation of the model, and also make the manner of comparison easier, because real data is being used. By comparing between the values in reality and what is modeled will give us a clearer picture on the accuracy of the model.

SECTION B
AN OPTIMIZATION-BASED COMPUTATIONAL
FRAMEWORK
FOR DOWNSTREAM CRUDE OIL SCHEDULING AT
REFINERY FRONT-END

CHAPTER B1

INTRODUCTION

B1.1 BACKGROUND OF STUDY

Scheduling and planning of the flow of crude oil is an important issue in petroleum refineries. This is due to the potential realization of large savings in cost and improved feeds. Linear programming (LP) models have previously been used for scheduling and planning problems because of their ease of modeling, and also since they are relatively easy to solve. However, it is usually difficult to model refinery operations, since they involve units operating in both batch and continuous modes with multiple grades of crude oil and products. Furthermore, detailed scheduling models usually require a continuous time representation that normally includes nonlinear (MINLP) models. These models provide additional flexibility to the problem by allowing the modeling of discrete decisions and constraints.

Two major approaches for modeling scheduling problems are discrete time formulations and continuous time formulations. In discrete time models, it is relatively easy to model the material balances and the flow constraints. However, the number of time intervals required for an accurate representation of the system is usually large. Continuous time models, on the other hand, are smaller in comparison, and they allow for a complete utilization of the time domain, although it is difficult to synchronize the material balances and time sequencing constraints in such a representation. Furthermore, it is often not trivial to determine a priori the number of time points or events that are needed.

In this work, the continuous time formulation method is adopted for short-term scheduling of crude oil at the front-end of a refinery and to solve the corresponding MINLP to global optimality. This scheduling problem involves crude oil unloading from a crude supply source to the crude storage tanks, transfer of crude oil from

these tanks to the charging tanks, and charging the crude distillation units (CDUs) continuously over a time horizon with crude-mixes from the charging tank.

B1.2 PROBLEM STATEMENT

The front-end of a refinery is a network consisting of crude oil supply streams, storage tanks, charging tanks, and crude distillation units (CDUs), which structure is shown in Figure B1.1 the crude supply streams are connected to the storage tanks, which are connected to the charging tanks, which in turn, are connected to the CDUs. The crude supply streams, which are vessels that carry crude, deliver crude oil to the storage tanks (intermediate tanks), which transfer the crude to the charging tanks. Different qualities of crude get blended into various crude mixtures inside the charging tanks, which are then charged directly to the distillation units.

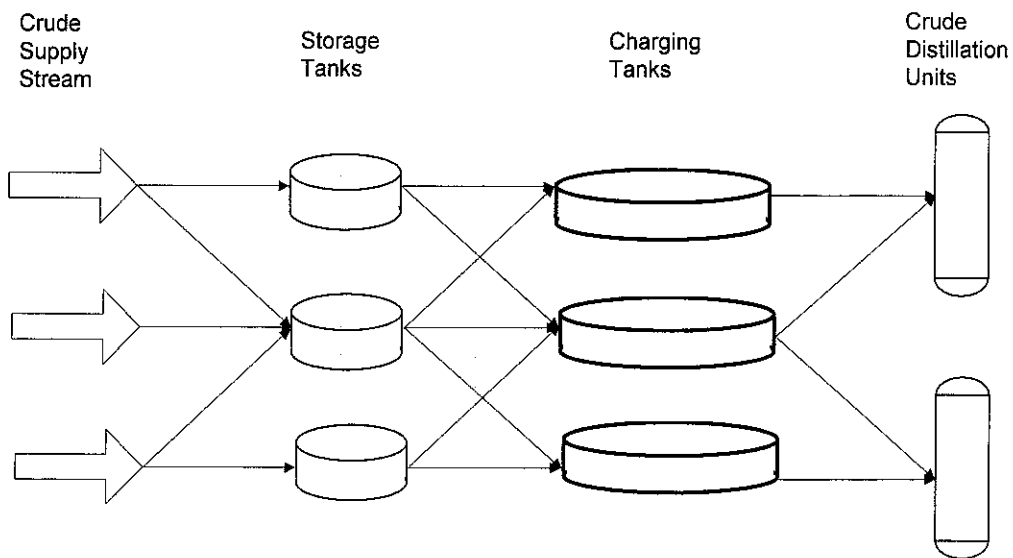


FIGURE B1.1: Superstructure representation of the crude oil scheduling operations at refinery front-end

B1.2.1 Model Information

The major assumptions in the model are as follows: The following information is given:

- the maximum and minimum inventory levels for a tank (capacity limitations);

- the initial total and component inventories in a tank;
- upper and lower bounds on the fraction of key components in the crude inside a tank (crude quality limitations);
- times of arrival of crude oil in the crude supply streams;
- amount of crude arriving in the crude supply streams;
- fractions of various components in the crude supply streams;
- demand of crude-mix to be charged from a charging tank;
- bounds on the flowrates of the streams in the network;
- time horizon for scheduling;
- cost coefficients for calculating the various costs involved;

B1.2.2 Model Constraints

Given the following operating constraints in the network:

- simultaneous inputs into and outputs from a tank cannot be allowed. This is done to allow settling of the crude-mix in a tank;
- each distillation unit may be charged by at most one charging tank over a period of time. This is another operational norm followed in many refineries;
- each charging tank may charge at most one distillation unit at a point of time;
- each charging tank has to discharge a specified amount of crude-mix to the various distillation units within the given time horizon;
- all the distillation units have to be operated continuously throughout the entire time horizon;

B1.2.3 Model Assumptions

We model the optimization of the network as a nonconvex MINLP problem. Certain assumptions are made prior to modeling the system:

- perfect mixing takes place in each tank;
- negligible change in specific gravities on mixing;
- the crude flows into and from a tank need not be continuous;
- changeover times for CDU charging are neglected;
- continuous time model;

B1.3 RESEARCH OBJECTIVES

The questions that we are interested to answer in this research project relates to the optimum values of the following items in the system in order to minimize the total operating cost of the network:

- the total inventory levels and the component inventory levels in the tanks at various instances of time;
- the total flow volume and the component flow volume from one unit to another in a certain time interval;
- start and end times of the flows in each stream present in the network;

CHAPTER B2

LITERATURE REVIEW

B2.1 PREVIOUS WORK

Karuppiah, et al (2008) presented an outer-approximation algorithm to obtain the global optimum of a nonconvex mixed-integer nonlinear programming (MINLP) model that is used to represent the scheduling of crude oil movement at the front-end of a petroleum refinery. The model relies on a continuous time representation making use of transfer events. They proposed an algorithm which focuses on effectively solving a mixed-integer linear programming (MILP) relaxation of the nonconvex MINLP to obtain a rigorous lower bound (LB) on the global optimum. Cutting planes derived by spatially decomposing the network are added to the MILP relaxation of the original nonconvex MINLP in order to reduce the solution time for the MILP relaxation. The solution of this relaxation is used as a heuristic to obtain a feasible solution to the MINLP which serves as an upper bound (UB). The lower and upper bounds are made to converge to within a specified tolerance in the proposed outer-approximation algorithm. As result of applying the proposed technique to test examples, significant savings are realized in the computational effort required to obtain provably global optimal solutions.

The paper by Lee et al (1996) addressed the problem of inventory management of a refinery that imports several types of crude oil which are delivered by different vessels. This problem involves optimal operation of crude oil unloading, is transfer from storage tanks to charging tanks, and the charging schedule for each crude oil distillation unit. A mixed-integer optimization model is developed which relies on time discretization. The problem involves bilinear equations due to mixing operations. The LP-based branch and bound method is applied to solve the model, and several techniques, such as priority branching and bounding, and special ordered sets are implemented to reduce the computation time.

Reddy et al (2004) presented a mixed-integer linear programming (MILP)-based solution approach for optimizing crude oil unloading, storage, and processing operations in a multi-CDU refinery receiving crude from multiparcel Very Large Crude Carriers (VLCCs) Through a high-volume, single-bouy mooring (SBM) pipeline and/or single-parcel tankers through multiple jetties. Their primarily discrete-time model allows multiple sequential crude transfers to occur within a time slot. As a result, an interesting approach to this problem is presented.

Rocha et al (2008) described how mathematical programming is being used to solve the Petroleum Allocation problem and they show the effectiveness of a Local Branching method to solve real industrial problems. They proposed a MILP formulation of the problem that relies on a time/space discretization network. An algorithm was implemented based on a heuristic to find a feasible solution and on a local search procedure by optimization to improve it. As a result, solutions are found for most of the case studies within 10% of optimality in less than 5 hours.

A generalized model is proposed by Furman et al (2007) for the continuous time scheduling problem of fluid transfer in tanks. Their model generally and more robustly handles the synchronization of time events with material balances. A novel method for representing the flow to and from a tank is developed with the potential for significant reduction in the number of necessary time events required for continuous time scheduling formulations. An efficient MINLP formulation is developed based on continuous representation of time domain under the assumption of no simultaneous input and output flow to a tank for fluid streams comprised of multiple components.

Karuppiah and Grossmann (2007) presented a global optimization algorithm for solving a class of large-scale nonconvex optimization models that have a decomposable structure. Such models are very expensive to solve to global optimality. They are frequently encountered in two-stage stochastic programming problems, engineering design, and also in planning and scheduling. They proposed a specialized deterministic branch-and-cut algorithm to solve these models to global optimality, wherein bounds on the global optimum are obtained by solving convex relaxations of these models with certain cuts added to them in order to tighten the

relaxations. Numerical examples are presented to illustrate the effectiveness of the proposed method compared to available commercial global optimization solvers that are based on branch and bound methods.

CHAPTER B3

MODEL FORMULATION AND SOLUTION STRATEGY

B3.1 FORMULATION

The mathematical model for the scheduling problem has largely been taken from Furman et al (2007). This is a continuous time model for scheduling for which a number of transfer events are postulated for the transfer of material between the units in the network over a given time horizon.

B3.1.1 Nomenclature

(a) Indices

<i>A</i>	tank input source
<i>B</i>	crude tank
<i>C</i>	tank output destination
<i>D</i>	distillation unit
<i>G</i>	charging tank
<i>I</i>	component
<i>M</i>	source unit of split pipeline
<i>K</i>	destination unit of split pipeline
<i>U</i>	crude supply stream
<i>Y</i>	storage tank
<i>L</i>	transfer event
<i>PP</i>	set of production platforms

PP = set of production platforms allocated to n

(b) Sets

A	tank input sources
A_b	inputs to tank b
A_s	inputs to storage tank y
B	tanks
B_{Q1}	tanks belonging to sub-structure Q1
B_{Q2}	tanks belonging to sub-structure Q2
C	tank output destinations
C_b	outputs from tank b
C_y	outputs from storage tank y
C_g	outputs from charging tank g
D	distillation units
D_{Q1}	distillation units present in sub-structure Q1
D_{Q2}	distillation units present in sub-structure Q2
D_g	distillation units that can be charged by charging tank g
$G(B)$	charging tanks
G_d	charging tanks that charge distillation unit d
I	components
M	source units of the split pipelines
K_m	destination units of split pipelines with source m
U	crude supply streams
U_{Q1}	crude supply streams present in sub-structure Q1
U_{Q2}	crude supply streams present in sub-structure Q2
$Y(B)$	storage tanks
Y_u	storage tanks connected to crude supply stream u
L	transfer events

(c) Parameters

C_{inv_b}	inventory maintenance cost for tank b
C_{sea}	waiting cost for crude supply streams
C_{set}	changeover cost for charged oil switch
C_{unload}	unloading cost for crude supply streams
DM_g	demand of crude-mix to be charged from charging tank g
$f_{i,b}^L$	lower bound on fraction of component i inside tank b
$f_{i,b}^U$	upper bound on fraction of component i inside tank b
$f_{i,u}^{supply}$	fraction of component i in crude supply stream u
$F_{a,b}^L$	lower bound on flowrate from a to b
$F_{a,b}^U$	upper bound on flowrate from a to b
H	time horizon for scheduling
$I_b^{init-tot}$	initial total inventory of tank b
$I_{i,b}^{init}$	initial inventory of component i in tank b
I_b^L	lower bound on total inventory in a tank b
I_b^U	upper bound on total inventory in a tank b
ND	number of distillation units in the network
ND _{Q1}	number of distillation units in sub-structure Q1
ND _{Q2}	number of distillation units in sub-structure Q2
NE	number of transfer events
$T_u^{arrival}$	arrival time of crude in crude supply stream u
V_u^{supply}	total volume of crude oil arriving in crude supply stream u
$V_{a,b}^L$	lower bound on flow volume from a to b
$V_{b,c}^L$	lower bound on flow volume from b to c
$V_{a,b}^U$	upper bound on flow volume from a to b
$V_{b,c}^U$	upper bound on flow volume from b to c
$\lambda_{m,k,l}^{T1}$	Lagrange multiplier
$\lambda_{m,k,l}^{T2}$	Lagrange multiplier

$\lambda_{m,k,l}^{\text{Tot}}$	Lagrange multiplier
$\lambda_{m,k,l}^{\text{VC}}$	Lagrange multiplier
$\lambda_{m,k,l}^{\text{w}}$	Lagrange multiplier

(d) Continuous Variables

$I_{\text{tot},b,l}$	total inventory of tank b at the end of transfer event l
$I_{i,b,l}$	inventory of component i in tank b at the end of transfer event l
$T_{\text{start},a,b,l}$	starting time of a transfer from a to b in transfer event l
$T_{\text{start},b,c,l}$	starting time of a transfer from b to c in transfer event l
$T_{\text{end},a,b,l}$	ending time of a transfer from a to b in transfer event l
$T_{\text{end},b,c,l}$	ending time of a transfer from b to c in transfer event l
$T_{\text{end},u}$	overall ending time of crude transfer from crude supply stream u
$T_{\text{start},u}$	initial starting time of crude transfer from crude supply stream u
$V_{\text{tot},a,b,l}$	total flow from a to b in transfer event l
$V_{\text{tot},b,c,l}$	total flow from b to c in transfer event l
$V_{\text{tot},i,a,b,l}$	flow of component i from a to b in transfer event l
$V_{\text{tot},i,b,c,l}$	flow of component i from b to c in transfer event l

(e) Binary variables

$w_{a,b,l}$	= 1 if there is a flow from a to b in transfer event l
$w_{b,c,l}$	= 1 if there is a flow from b to c in transfer event l
	= 0 otherwise

B3.1.2 Model Equations

(a) Tank Constraints

The majority of the constraints in the model pertain to the crude tanks in the network.

The representation of a crude tank $b \in B$ with crude input source a and a crude output destination c is shown in Figure B3.1

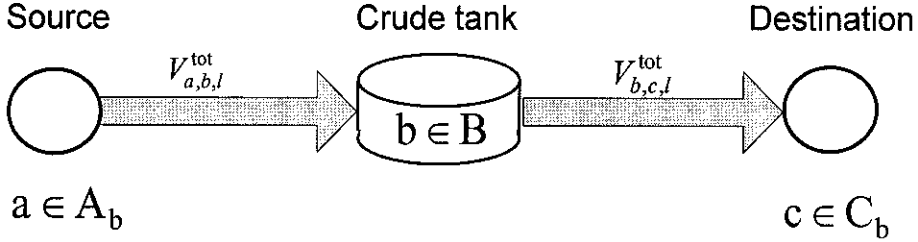


FIGURE B3.1: Crude tank representation

(i) *Big-M logical constraints for flow transfers:*

$$V_{\text{tot},a,b,l} \leq V_{a,b}^U w_{a,b,l} \quad \forall a \in A_b, \forall b \in B, \forall l \in L \quad (\text{B1})$$

$$V_{\text{tot},b,c,l} \leq V_{b,c}^U w_{b,c,l} \quad \forall c \in C_b, \forall b \in B, \forall l \in L \quad (\text{B2})$$

These constraints force the total flow in a stream $V_{\text{tot},a,b,l}$ from a source tank a to any destination b in a particular transfer event l to zero if the binary variable, $w_{a,b,l}$, which enforces the existence of flow in that stream in transfer event l , takes a value of zero. Note that the first subscript in the total flow variable denotes the source from where the flow is taking place, while the second subscript denotes the destination to where the flow is going. The third and final subscript denotes the transfer event when the particular flow occurs.

The binary variable, $w_{a,b,l}$, represents the existence of flow between source a and tank b in transfer event l . The same is true for binary variable, $w_{b,c,l}$, which takes on a value of 1 or 0, respectively, depending on whether or not there is flow between tank b and a destination unit c in transfer event l .

(ii) Duration constraints

Duration constraints for flows: (i) between source and tank and (ii) between tank and destination

$$F_{a,b}^U (T_{\text{end},a,b,l} - T_{\text{start},a,b,l}) + F_{a,b}^U H (1 - w_{a,b,l}) \geq V_{\text{tot},a,b,l} \quad \forall a \in A_b, \forall b \in B, \forall l \in L \quad \text{(B3.i)}$$

$$F_{b,c}^U (T_{\text{end},a,b,l} - T_{\text{start},a,b,l}) + F_{b,c}^U H (1 - w_{b,c,l}) \geq V_{\text{tot},b,c,l} \quad \forall c \in C_b, \forall b \in B, \forall l \in L \quad \text{(B3.ii)}$$

For a flow between source a and tank b , the timing variables $T_{\text{start},a,b,l}$ and $T_{\text{end},a,b,l}$ correspond to the start and end times of flow in a stream from a to b in transfer event l .

The timing variables $T_{\text{start},b,c,l}$ and $T_{\text{end},b,c,l}$ are similarly defined for a flow between tank b and a destination c in transfer event l . H is the overall time horizon of operation. These constraints are relaxed, i.e., they are redundant when $w_{a,b,l} = 0$. Thus, the timing variables can take on any value if there is no flow in a certain transfer event.

The above is expressed through big- M logical constraints, which state that if there is a flow in a stream in the network in transfer event l , then the product of the upper bound on the flowrate of the crude stream with the duration of flow in the transfer event gives an upper bound on the total flow volume in that transfer event.

$$w_{a,b,l} = 0:$$

$$F_{a,b}^U (T_{2,a,b,l} - T_{1,a,b,l}) + F_{a,b}^U H \geq V_{\text{tot},a,b,l} \Rightarrow \text{the constraint is relaxed}$$

$$w_{a,b,l} = 1:$$

$$F_{a,b}^U (T_{2,a,b,l} - T_{1,a,b,l}) \geq V_{\text{tot},a,b,l}$$

Similarly, as given in equations (B4.i) and (B4.ii), if there is a flow in transfer event l into or from a tank, the lower bound on the volume of a flow is obtained by multiplying the fluid flowrate lower bound with the duration of flow:

$$F_{a,y}^L (T_{\text{end},a,y,l} - T_{\text{start},a,y,l}) - F_{a,s}^L H(1 - w_{a,y,l}) \leq V_{\text{tot},a,y,l} \quad \forall a \in A_b, \forall y \in Y, \forall l \in L \quad (\text{B4.i})$$

$$F_{y,c}^L (T_{\text{end},y,c,l} - T_{\text{start},y,c,l}) - F_{y,c}^L H(1 - w_{y,c,l}) \leq V_{\text{tot},y,c,l} \quad \forall a \in C_y, \forall y \in Y, \forall l \in L \quad (\text{B4.ii})$$

$$F_{g,c}^L (T_{\text{end},g,c,l} - T_{\text{start},g,c,l}) \leq V_{\text{tot},g,c,l} \quad \forall c \in C_g, \forall g \in G, \forall l \in L \quad (\text{B4.iii})$$

We should note that for the charging tanks, the start and end times have to coincide if there is no flow in a particular time event (B4.iii). This enforces the continuity of operation of the CDUs under the condition that only one charging tank can charge a CDU in a certain transfer event.

(iii) Simple sequencing constraints

A flow into or from a tank b in transfer event l has to take place before the same flow in event $(l + 1)$. Equations (B5) – (B10) correspond to this necessary condition.

For flows between source and tank and between tank and destination:

$$T_{\text{start},a,b,(l+1)} \geq T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \quad \forall a \in A_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B5})$$

$$T_{\text{start},a,b,(l+1)} \geq T_{\text{start},a,b,l} \quad \forall a \in A_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B6})$$

$$T_{\text{end},a,b,(l+1)} \geq T_{\text{end},a,b,l} \quad \forall a \in A_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B7})$$

For flows into or from a tank:

$$T_{\text{start},b,c,(l+1)} \geq T_{\text{end},b,c,l} - H(1 - w_{b,c,l}) \quad \forall c \in C_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B8})$$

$$T_{\text{start},b,c,(l+1)} \geq T_{\text{start},b,c,l} \quad \forall c \in C_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B9})$$

$$T_{\text{end},b,c,(l+1)} \geq T_{\text{end},b,c,l} \quad \forall c \in C_b, \forall b \in B, \forall l \in L, l < |L| \quad (\text{B10})$$

If no flow exists between a and b in transfer event l (that is, $w_{a,b,l} = 0$), then the big-M inequality (B5) is relaxed. This is because the constraint simply becomes redundant (as $T_{\text{start},a,b,(l+1)}$ is definitely greater than $T_{\text{end},a,b,l}$ and so subtracting with H would make the RHS value even smaller compared to the LHS)

$$\begin{aligned} T_{\text{start},a,b,(l+1)} &\geq T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \\ w_{a,b,l} &= 0 : \\ T_{\text{start},a,b,(l+1)} &\geq T_{\text{end},a,b,l} - H \end{aligned}$$

Similarly, if there is no flow between b and c in transfer event l (that is, $w_{b,c,l} = 0$), then the big-M inequality (B8) is relaxed. Essentially, it means that if there is no flow in a stream in a transfer event l , then the values taken by the variables pertaining to the start and end times of flow in transfer event l are meaningless and do not affect the flow times in the next transfer event when there is flow.

(iv) Input and output restraints for the entire horizon

A set of constraints has to enforce the condition that any inputs or outputs of the current transfer event t must occur after the inputs and outputs of the preceding transfer event. The inclusion of these time constraints, which are expressed as big-M constraints, enforces the material balances to be calculated properly across all tanks in the same transfer event.

$$T_{\text{start},a,b,(l+1)} \geq T_{\text{end},a',b,l} - H(1 - w_{a',b,l}) \quad \forall a, a' \in A_b, a \neq a', \forall b \in B, \forall l \in L, l < |L| \quad \text{(B11.i)}$$

$$T_{\text{start},a,b,(l+1)} \geq T_{\text{end},b,c,l} - H(1 - w_{b,c,l}) \quad \forall a \in A_b, \forall c \in C_b, \forall b \in B, \forall l \in L, l < |L| \quad \text{(B11.ii)}$$

$$T_{\text{start},b,c,(l+1)} \geq T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \quad \forall a \in A_b, \forall c \in C_b, \forall b \in B, \forall l \in L, l < |L| \quad \text{(B11.iii)}$$

$$T_{\text{start},b,c,(l+1)} \geq T_{\text{end},b,c',l} - H(1 - w_{b,c',l}) \quad \forall c, c' \in C_b, c \neq c', \forall b \in B, \forall l \in L, l < |L| \quad \text{(B11.iv)}$$

Also, since all inputs into a tank b are required to finish before any output starts from that tank b in any transfer event, we need the following constraint:

$$T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \leq T_{\text{start},b,c,l} + H(1 - w_{b,c,l}) \quad \forall a \in A_b, \forall c \in C_b, \forall b \in B, \forall l \in L \quad (\mathbf{B12})$$

What do we want to enforce here through this constraint?

When $w_{b,c,l} = 1$ (i.e., output from tank b starts),
must have $w_{a,b,l} = 0$

We do not want the condition:

$$T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \leq T_{\text{start},b,c,l} + H(1 - w_{b,c,l}) \quad \forall a \in A_b, \forall c \in C_b, \forall b \in B, \forall l \in L$$

$$w_{a,b,l} = 1 (\Rightarrow \text{must have } w_{b,c,l} = 0)$$

$$T_{\text{end},a,b,l} \leq T_{\text{start},b,c,l} + H(1 - w_{b,c,l})$$

If $w_{b,c,l} = 1$ also, which it is not supposed to be, then obtain constraint:

$$T_{\text{end},a,b,l} \leq T_{\text{start},b,c,l}$$

this is redundant: $T_{\text{end},a,b,l} \leq T_{\text{start},b,c,l} + H$, which means that when $w_{b,c,l} = 1$, we do not need the condition $w_{a,b,l} = 0$ because it naturally holds

$$T_{\text{end},a,b,l} - H(1 - w_{a,b,l}) \leq T_{\text{start},b,c,l} + H(1 - w_{b,c,l}) \quad \forall a \in A_b, \forall c \in C_b, \forall b \in B, \forall l \in L$$

$w_{a,b,l} = 1 \Rightarrow w_{b,c,l} = 0$ and the following constraint is enforced :

$$T_{\text{end},a,b,l} \leq T_{\text{start},b,c,l} + H$$

$w_{a,b,l} = 0 \Rightarrow w_{b,c,l} = 1$ and the following constraint is enforced :

$$T_{\text{end},a,b,l} - H \leq T_{\text{start},b,c,l}$$

This helps in upholding material balances in the transfer event l and prevents the situation in which output could occur before any input into a tank.

(v) Mass Balances

For each tank $b \in B$ in the network, we have an overall inventory balance (B13 and B14), individual inventory balances (B15) for each component $i \in I$ and the total flow balances (B17 and B18).

The inventory balances imply that the inventory in tank b at the end of a transfer event l is equal to the inventory at the end of transfer event $(l - 1)$ plus the volume flow into the tank from any input source a in transfer event l , minus the flow to any output destination c in the transfer event l . The variables $I_{\text{tot},b,l}$ and $I_{i,b,l}$ correspond to the total inventory and the individual component inventory in a tank b at the end of transfer event l , respectively.

$$I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} = I_{\text{tot},b,l} + \sum_{c \in C_b} V_{\text{tot},b,c,l} \quad \forall b \in B, \forall l \in L \quad (\text{B13})$$

$$I_{\text{tot},b,0} = I_b^{\text{init-tot}} \quad \forall b \in B \quad (\text{B14})$$

$$I_{i,b,(l-1)} + \sum_{a \in A_b} V_{i,a,b,l} = I_{i,b,l} + \sum_{c \in C_b} V_{i,b,c,l} \quad \forall i \in I, \forall b \in B, \forall l \in L \quad (\text{B15})$$

The volume flow balances imply that the total flow into or out from a tank equals the sum of the individual component flows. $V_{\text{tot},a,b,l}$ stands for the total volume flow from any source a to tank b in transfer event l , while $V_{\text{tot},b,c,l}$ represents the flow from tank b to a destination c to which this tank is connected. $V_{i,a,b,l}$ and $V_{i,b,c,l}$ are the respective component flows.

$$I_{i,b,0} = I_{i,b}^{\text{init}} \quad \forall i \in I, \forall b \in B \quad (\text{B16})$$

$$V_{\text{tot},a,b,l} = \sum_{i \in I} V_{i,a,b,l} \quad \forall a \in A_b, \forall b \in B, \forall l \in L \quad (\text{B17})$$

$$V_{\text{tot},b,c,l} = \sum_{i \in I} V_{i,b,c,l} \quad \forall c \in C_b, \forall b \in B, \forall l \in L \quad (\text{B18})$$

(vi) Component Balances

On assuming perfect mixing in a tank, the fraction of a component i in the output flow from a tank should be equal to the fraction of that component present inside the tank. In the same time event, any input flows must occur before any output flows. The final inventory of the tank from the previous time event is used, rather than the final inventory of the tank for a particular time event, to avoid the numerical irregularities that can result when the tank is completely emptied. This constraint is formulated as follows, with bilinear terms, which give rise to be the nonconvexities of the model:

$$\left(I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} \right) V_{i,b,c,l} = \left(I_{i,b,(l-1)} + \sum_{a \in A_b} V_{i,a,b,l} \right) V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L \quad (\text{B19})$$

(vii) Inventory Bounds

Because both the input and output of material may occur in the same time event, simple bounds on the total tank inventory variable $I_{\text{tot},b,l}$ are not sufficient. The following constraint must hold in order to ensure that the total inventory in any transfer event does not exceed the upper bound of the inventory since both inputs and outputs can occur in the same transfer event.

$$I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} \leq I_b^U \quad \forall b \in B, \forall l \in L \quad (\text{B20})$$

The sum $\left(I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} \right)$ is the total inventory in a tank b in transfer event l before any output flow starts to occur from the tank in the same transfer event.

(viii) Bounds on component fractions inside a tank

The fraction of a component in the crude inside any tank should lie between given bounds. This is enforced by the following constraints:

$$f_{i,b}^L I_{\text{tot},b,l} \leq I_{i,b,l} \leq f_{i,b}^U I_{\text{tot},b,l} \quad \forall i \in I, \forall b \in B, \forall l \in L \quad (\text{B21})$$

$$f_{i,b}^L V_{\text{tot},b,c,l} \leq V_{i,b,c,l} \leq f_{i,b}^U V_{\text{tot},b,c,l} \quad \forall i \in I, \forall b \in B, \forall c \in C_b, \forall l \in L \quad (\text{B22})$$

(ix) Crude-mix demand constraints

Each charging tank $g \in G$ must charge a specified amount of crude-mix over the entire scheduling horizon. This volume of crude mix is distributed to the different CDUs in the network.

$$\sum_{d \in D_g} \sum_l V_{\text{tot},g,d,l} = \text{DM}_g \quad \forall g \in G \quad (\text{B23})$$

(x) Bound strengthening cuts

The following constraints are added to the model in an attempt to tighten the relaxation of the MINLP model so as to accelerate the convergence to compute the optimal solution. These are derived using a reformulation and linearization technique given in Sherali and Alameddine (1992). In this we take Eq. (B19) and expand it to get the following equation:

$$I_{\text{tot},b,(l-1)} V_{i,b,c,l} + \sum_{a \in A_b} V_{\text{tot},a,b,l} V_{i,b,c,l} = I_{i,b,(l-1)} V_{\text{tot},b,c,l} + \sum_{a \in A_b} V_{i,a,b,l} V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

$$\sum_{i \in I} I_{\text{tot},b,(l-1)} V_{i,b,c,l} = I_{\text{tot},b,(l-1)} V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall l \in L$$

$$\sum_{i \in I} V_{\text{tot},a,b,l} V_{i,b,c,l} = V_{\text{tot},a,b,l} V_{\text{tot},b,c,l} \quad \forall a \in A_b, \forall b \in B, \forall c \in C_b, \forall l \in L$$

$$\sum_{i \in I} I_{i,b,(l-1)} V_{\text{tot},b,c,l} = I_{\text{tot},b,(l-1)} V_{\text{tot},b,c,l}, \quad \forall b \in B, \forall c \in C_b, \forall l \in L$$

$$\sum_{i \in I} V_{i,a,b,l} V_{\text{tot},b,c,l} = V_{\text{tot},a,b,l} V_{\text{tot},b,c,l} \quad \forall a \in A_b, \forall b \in B, \forall c \in C_b, \forall l \in L$$

(B24)

(b) Distillation Unit Constraints

Each distillation unit $d \in D$ is modeled with the following set of constraints:

(i) Allocation constraints

The condition that each distillation unit can be charged by at most one charging tank in a transfer event is enforced by the following equation:

$$\sum_{g \in G_d} w_{g,d,l} \leq 1 \quad \forall d \in D, \forall l \in L \quad (\text{B25})$$

The condition that at most one CDU can be charged by a single charging tank in a transfer event is enforced by equation (26):

$$\sum_{d \in D_g} w_{g,d,l} \leq 1 \quad \forall g \in G, \forall l \in L \quad (\text{B26})$$

(ii) Continuous operation constraints

Each crude distillation unit must be operated continuously and the total time of operation of each CDU must be equal to the time horizon H . To ensure that the crude distillation unit (CDU) has a continuous feed within certain specifications and meeting demand:

$$\sum_l \sum_{g \in G_d} [T_{\text{end},g,d,l} - T_{\text{start},g,d,l}] \quad \forall d \in D \quad (\text{B27})$$

This equation ensures that a CDU has a crude oil flow into it for the entire duration of the time horizon without a gap in time.

Because of the continuity required in the duration of operation, and the requirement that only one charging tank can charge a CDU over a period of time, for a CDU which is charged in transfer event l , the next charge (in transfer event $(l+1)$) will start

at the ending time of the current transfer event l . This is enforced by equations (28) and (29).

$$T_{\text{start},g,d,(l+1)} \geq T_{\text{end},g',d,l} - H(1 - w_{g',d,l}) \quad \forall g, g' \in G_d, g \neq g', \forall d \in D, \forall l \in L, l < |L| \quad (\text{B28})$$

$$T_{\text{start},g,d,(l+1)} \leq T_{\text{end},g',d,l} + H(1 - w_{g',d,l}) \quad \forall g, g' \in G_d, g \neq g', \forall d \in D, \forall l \in L, l < |L| \quad (\text{B29})$$

(c) Crude Supply Stream Constraints

The crude supply streams have to follow certain mass balance and timing constraints:

(i) Timing constraints

All the flows from a crude supply stream u to storage tank y in any transfer event must start after a particular time ($T_{\text{start},u}$) and end before a certain time ($T_{\text{end},u}$):

$$T_{\text{start},u} \leq T_{\text{start},u,y,l} + H(1 - w_{u,y,l}) \quad \forall u \in U, \forall y \in Y_u, \forall l \in L \quad (\text{B30a})$$

$$T_{\text{end},u} \geq T_{\text{end},u,y,l} - H(1 - w_{u,y,l}) \quad \forall u \in U, \forall y \in Y_u, \forall l \in L \quad (\text{B30b})$$

It is to be noted that the flow from a crude supply stream can be split such that one or more storage tanks are simultaneously fed by a single crude supply stream. Also, two or more supply streams can feed the same storage tank at the same time.

(ii) Overall mass balances

The total amount of crude oil arriving in a crude supply stream u (given by V_u^{supply}), must be completely transferred to the storage tanks over the set of all transfer events in the horizon.

$$\sum_{l \in L} \sum_{y \in Y_u} V_{\text{tot},u,y,l} = V_u^{\text{supply}} \quad \forall u \in U \quad (\text{B31})$$

(iii) Component balances

The component flow from a crude supply stream u to a tank y (storage tank) in a transfer event l is equal to the product of the total flow from that crude supply stream to the tank and the fraction of the component in the crude supply stream, which is known.

$$V_{i,u,y,l} = f_{i,u}^{\text{supply}} V_{\text{tot},u,y,l} \quad \forall i \in I, \forall y \in Y_u, \forall u \in U, \forall l \in L \quad (\text{B32})$$

(d) Variable Bounds

All the continuous variables must lie between specified bounds and the discrete variables can be either 0 or 1.

$$\begin{aligned} 0 &\leq I_{i,b,l} \leq I_b^U && \forall i \in I, \forall b \in B, \forall l \in L \\ I_b^L &\leq I_{\text{tot},b,l} \leq I_b^U && \forall b \in B, \forall l \in L \\ 0 &\leq V_{i,a,b,l} \leq V_{a,b}^U && \forall i \in I, \forall a \in A_b, \forall b \in B, \forall l \in L \\ 0 &\leq V_{i,b,c,l} \leq V_{b,c}^U && \forall i \in I, \forall c \in C_b, \forall b \in B, \forall l \in L \\ 0 &\leq V_{\text{tot},a,b,l} \leq V_{a,b}^U && \forall a \in A_b, \forall b \in B, \forall l \in L \\ 0 &\leq V_{\text{tot},a,b,l} \leq V_{b,c}^U && \forall c \in C_b, \forall b \in B, \forall l \in L \\ 0 &\leq T_{\text{start},a,b,l} \leq H && \forall a \in A_b, \forall b \in B, \forall l \in L \\ 0 &\leq T_{\text{end},a,b,l} \leq H && \forall a \in A_b, \forall b \in B, \forall l \in L \\ 0 &\leq T_{\text{start},b,c,l} \leq H && \forall c \in C_b, \forall b \in B, \forall l \in L \\ 0 &\leq T_{\text{end},b,c,l} \leq H && \forall c \in C_b, \forall b \in B, \forall l \in L \\ T_u^{\text{arrival}} &\leq T_{\text{start},u} \leq H && \forall u \in U \\ T_u^{\text{arrival}} &\leq T_{\text{end},u} \leq H && \forall u \in U \\ w_{a,b,l}, w_{b,c,l} &\in \{0,1\} \end{aligned} \quad (\text{B33})$$

(e) Objective Function

$$\begin{aligned}
\min z = & C_{sea} \sum_{1 \leq u \leq 4} (T_{start,u} - T_u^{arrival}) + C_{unload} \sum_{1 \leq u \leq 4} (T_{start,u} - T_u^{arrival}) \\
& \text{waiting cost for a crude supply stream} \quad \text{unloading cost of crude for a crude supply stream} \\
& + \frac{H \times \sum_{b \in B} C_{inv,b} \times \left[\left(\sum_{b \in B} \sum_l I_{tot,b,l} \right) + \left(\sum_{b \in B} \sum_l \sum_{a \in A_b} V_{tot,a,b,l} \right) + \left(\sum_{b \in B} \sum_{l \in [L]} I_{tot,b,l} \right) + \left(\sum_{b \in B} 2I_b^{init-tot} \right) \right]}{(2 \times NE + 1)} \\
& \text{total inventory maintenance cost of all tanks in the system} \\
& + C_{set} \left(\sum_{1 \leq d \in D} \sum_{g \in G_d} \sum_l w_{g,d,l} - ND \right) \\
& \text{setup cost of charging the CDUs with different crude-mixes} \\
\max \Psi = & \sum_{1 \leq t \leq 4} c_{1,t} x_{total,t} - \sum_{1 \leq t \leq 4} \sum_{p \in PP} c_{p,2,t} z_{p,t} + c_{p,3,t} e_{p,t} \\
& \text{income from sales} \quad \text{investment costs of the production platform} \\
& - \sum_{1 \leq t \leq 4} \sum_{p \in PP} \sum_{n \in WP(p)} c_{n,p,2,t} z_{n,p,t} + c_{n,p,3,t} e_{n,p,t} \\
& \text{investment costs of the well platform} \\
& - \sum_{1 \leq t \leq 4} \sum_{p \in PP} \sum_{n \in WP(p)} \sum_{w \in WP(p)} c_{w,n,p,2,t} z_{w,n,p,t} \\
& \text{drilling costs of wells} \\
& - \left(\sum_{1 \leq t \leq 4} c_{4,t} \sum_{k \in D} \sum_{p \in PP} z_{p,k,t} + \sum_{1 \leq t \leq 4} c_{4,t} \sum_{k \in D} \sum_{p \in PP} \sum_{n \in WP(p)} ZT_{n,p,k,t} \right) \\
& \text{costs for moving the drilling rigs}
\end{aligned} \tag{B434}$$

In the equation above, the first sum is a waiting cost for a crude supply stream while the second term represents the unloading cost of crude for a crude supply stream. The total inventory maintenance cost of all the tanks in the system is given by the third term. This term represents the cost due to an approximate average inventory level in each of the tanks in a transfer event, that is computed by considering the inventory levels at the boundaries of the transfer event, and at the middle point between the end of inputs into a particular tank and start of outputs from that tank in the transfer event. That final term corresponds to the setup cost of charging the ‘ND’ CDUs with different crude-mixes.

Equations (B1) – (B23) and (B25) – (B34) comprise the MINLP model (P) which is to be optimized. The equation that involves bilinearities and is responsible for the nonconvexity of the model is equation (B19).

B3.2 SOLUTION STRATEGY

Large-scale MINLPs such as problem (P) require specialized solution algorithms. Here, a specialized outer-approximation algorithm is proposed for solving the nonconvex model (P) to global optimality within a specified tolerance. In the proposed technique, lower and upper bounds are generated on the global optimum of (P) over a search region, which are then converged in the proposed algorithm.

B3.2.1 PREPROCESSING

The bounds of the variables in the model are determined by physical inspection of the network structure and using the numerical data given for the tanks, crude supply streams and the distillation units. Also, in this step, the original nonconvex MINLP may be locally optimized to obtain an initial overall upper bound (OUB) for the objective function.

B3.2.2 LOWER BOUNDING PROBLEM

The cross-product constraint described in Eq. (B19) is used to properly calculate component fractions and has nonconvex nonlinearity in the form of bilinear terms. A tight convex relaxation for Eq. (B19) may be stated as a set of linear constraints known as McCormick estimators. This set of constraints could be added to a formulation to generally tighten it. They are also useful as a replacement for Eq. (19) to relax the problem and reduce it from the exact nonconvex MINLP to a convex MILP relaxation. New artificial variables would be needed to replace the bilinear terms in Eq. (B19).

The expanded form of Eq. (B19) is shown here:

$$\left(I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} \right) V_{i,b,c,l} = \left(I_{i,b,(l-1)} + \sum_{a \in A_b} V_{i,a,b,l} \right) V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

$$I_{\text{tot},b,(l-1)} V_{i,b,c,l} + \sum_{a \in A_b} V_{\text{tot},a,b,l} V_{i,b,c,l} = I_{i,b,(l-1)} V_{\text{tot},b,c,l} + \sum_{a \in A_b} V_{i,a,b,l} V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

(B19)(expanded)

These new variables are defined as follows:

$$I_{i,b,c,l}^{VJ} = I_{\text{tot},b,(l-1)} V_{i,b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

$$I_{i,b,c,l}^{VT} = I_{i,b,(l-1)} V_{\text{tot},b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

$$V_{i,a,b,c,l}^{VJ} = V_{\text{tot},a,b,l} V_{i,b,c,l} \quad \forall a \in A_b, \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

$$V_{i,a,b,c,l}^{VT} = V_{i,a,b,l} V_{\text{tot},b,c,l} \quad \forall a \in A_b, \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

Replacing the bilinear terms in constraint (B19) with the new variables, constraint (B19) can be converted to constraint (B35):

$$I_{i,b,c,l}^{VJ} + \sum_{a \in A_b} V_{i,a,b,c,l}^{VJ} = I_{i,b,c,l}^{VT} + \sum_{a \in A_b} V_{i,a,b,c,l}^{VT} \quad \forall i \in I, \forall l \in L, \forall c \in C_b, \forall b \in B \quad \text{(B35)}$$

As shown by McCormick, considering a bilinear term $z = xy$, with the upper and lower bounds on x and y ,

$$x^L \leq x \leq x^U$$

$$y^L \leq y \leq y^U$$

the valid overestimator and underestimator for z take the following form :

$$z \geq x^L y + x y^L - x^L y^L$$

$$z \geq x^U y + x y^U - x^U y^U$$

$$z \leq x^U y + x y^L - x^U y^L$$

$$z \leq x^L y + x y^U - x^L y^U$$

Similarly, the convex envelope constraints for bilinear terms $I_{i,b,c,l}^{VJ}$, $I_{i,b,c,l}^{VT}$, $V_{i,a,b,c,l}^{VJ}$, and $V_{i,a,b,c,l}^{VT}$ can be written as in constraints (36) – (39) as follows:

$$\begin{aligned}
I_{i,b,c,l}^{VJ} &\geq I_b^L V_{i,b,c,l} + V_{b,c}^L I_{\text{tot},b,(l-1)} - I_b^L V_{b,c}^L && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VJ} &\geq I_b^U V_{i,b,c,l} + V_{b,c}^U I_{\text{tot},b,(l-1)} - I_b^U V_{b,c}^U && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VJ} &\leq I_b^L V_{i,b,c,l} + V_{b,c}^U I_{\text{tot},b,(l-1)} - I_b^L V_{b,c}^U && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VJ} &\leq I_b^U V_{i,b,c,l} + V_{b,c}^L I_{\text{tot},b,(l-1)} - I_b^U V_{b,c}^L && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B
\end{aligned} \tag{B36}$$

$$\begin{aligned}
I_{i,b,c,l}^{VT} &\geq I_b^L V_{\text{tot},b,c,l} + V_{b,c}^L I_{i,b,(l-1)} - I_b^L V_{b,c}^L && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VT} &\geq I_b^U V_{\text{tot},b,c,l} + V_{b,c}^U I_{i,b,(l-1)} - I_b^U V_{b,c}^U && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VT} &\leq I_b^L V_{\text{tot},b,c,l} + V_{b,c}^U I_{i,b,(l-1)} - I_b^L V_{b,c}^U && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \\
I_{i,b,c,l}^{VT} &\leq I_b^U V_{\text{tot},b,c,l} + V_{b,c}^L I_{i,b,(l-1)} - I_b^U V_{b,c}^L && \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B
\end{aligned} \tag{B37}$$

$$\begin{aligned}
V_{i,a,b,c,l}^{VJ} &\geq V_{a,b}^L V_{i,b,c,l} + V_{b,c}^L V_{\text{tot},a,b,l} - V_{a,b}^L V_{b,c}^L && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VJ} &\geq V_{a,b}^U V_{i,b,c,l} + V_{b,c}^U V_{\text{tot},a,b,l} - V_{a,b}^U V_{b,c}^U && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VJ} &\leq V_{a,b}^L V_{i,b,c,l} + V_{b,c}^U V_{\text{tot},a,b,l} - V_{a,b}^L V_{b,c}^U && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VJ} &\leq V_{a,b}^U V_{i,b,c,l} + V_{b,c}^L V_{\text{tot},a,b,l} - V_{a,b}^U V_{b,c}^L && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B
\end{aligned} \tag{B38}$$

$$\begin{aligned}
V_{i,a,b,c,l}^{VT} &\geq V_{a,b}^L V_{\text{tot},b,c,l} + V_{b,c}^L V_{i,a,b,l} - V_{a,b}^L V_{b,c}^L && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VT} &\geq V_{a,b}^U V_{\text{tot},b,c,l} + V_{b,c}^U V_{i,a,b,l} - V_{a,b}^U V_{b,c}^U && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VT} &\leq V_{a,b}^L V_{\text{tot},b,c,l} + V_{b,c}^U V_{i,a,b,l} - V_{a,b}^L V_{b,c}^U && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B \\
V_{i,a,b,c,l}^{VT} &\leq V_{a,b}^U V_{\text{tot},b,c,l} + V_{b,c}^L V_{i,a,b,l} - V_{a,b}^U V_{b,c}^L && \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B
\end{aligned} \tag{B39}$$

A rigorous lower bound on the global optimum of problem (P) can be obtained by solving an MILP relaxation of the original nonconvex MINLP model (P). This relaxation can be constructed by replacing the nonlinear Eq. (B19) with Eq. (B35) and using convex envelopes. Eqs. (B36) – (B39) for the bilinear terms appearing in Eq. (B19).

The relaxed MILP problem (R) consists of Eqs. (B1) – (B18), (B20) – (B23) and (B25) – (B39). The MILP relaxation (R) is often very large in size and therefore needs significant computational effort to solve. To reduce the computational effort in solving this problem, cutting planes are added to model (R) which are derived using a technique similar to that given in Karuppiyah and Grossmann (2007).

The description of the derivation of these cutting planes is as follows. The network is split into separate decoupled structures, as shown in Figure B3.2, following the concept of spatial decomposition. Here the network is split into two decoupled sub-structures, although more sub-structures are possible. The sub-structure to the left of the dotted line in Figure 3.B is called Q1, while the sub-structure on the right is termed Q2. Physically, such a split can be interpreted as cutting the pipelines between some of the units in the network.

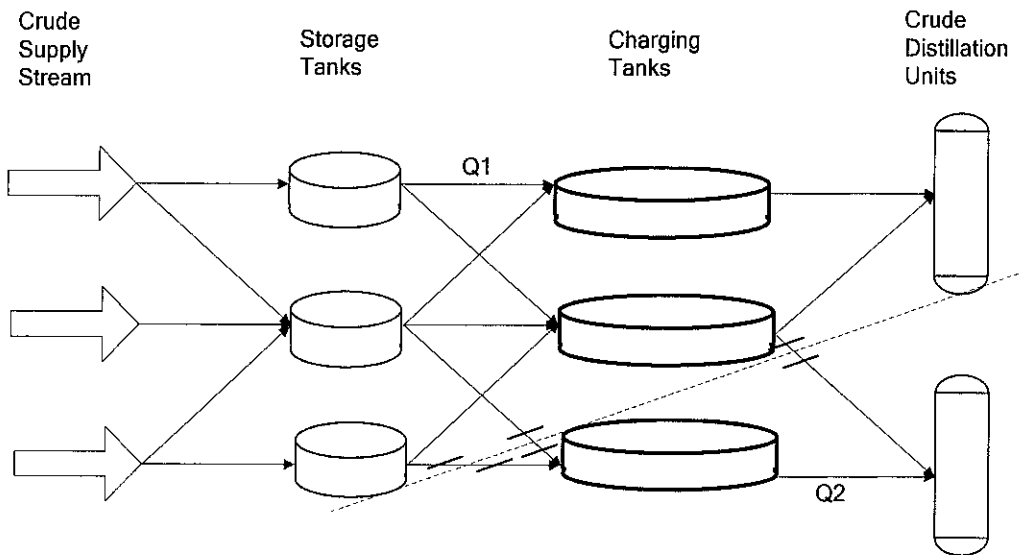


FIGURE B3.2: Spatial decomposition of network structure

The variables pertaining to the flow existence (binary variables $w_{a,b,l}$ and $w_{b,c,l}$), total flow, component flows, and start and end times of flow are duplicated for all the connections in the network that have been split. The results are two sets of duplicate variables

$$\left\{ V_{\text{tot},m,k,l}^1, V_{i,m,k,l}^1(\forall i), T_{\text{start},m,k,l}^1, T_{\text{end},m,k,l}^1, w_{m,k,l}^1 \right\} \forall m \in M, \forall k \in K_m, \forall l \quad \text{and}$$

$$\left\{ V_{\text{tot},m,k,l}^2, V_{i,m,k,l}^2(\forall i), T_{\text{start},m,k,l}^2, T_{\text{end},m,k,l}^2, w_{m,k,l}^2 \right\} \forall m \in M, \forall k \in K_m, \forall l, \quad \text{one set for each}$$

decomposed problem, and replace the variables

$\{V_{\text{tot},m,k,l}, V_{i,m,k,l}(\forall i), T_{\text{start},m,k,l}, T_{\text{end},m,k,l}, w_{m,k,l}\} \forall m \in M, \forall k \in K_m, \forall l$ with these newly created variables in model (P). The subscript m stands for the source of the pipeline that has been split, while the subscript k stands for the destination of a pipe that has been split. The variables $\{V_{\text{tot},m,k,l}, V_{i,m,k,l}(\forall i), T_{\text{start},m,k,l}, T_{\text{end},m,k,l}, w_{m,k,l}\} \forall m \in M, \forall k \in K_m, \forall l$ are said to be the linking variables since they link the different sub-structures. The remaining variables in model (P) are called non-linking variables since they are separate for both sub-structures Q1 and Q2. Due to the introduction of the duplicate variables, the equations involving the split pipelines get duplicated and are written in terms of the variables $\{V_{\text{tot},m,k,l}^1, V_{i,m,k,l}^1(\forall i), T_{\text{start},m,k,l}^1, T_{\text{end},m,k,l}^1, w_{m,k,l}^1\} \forall m \in M, \forall k \in K_m, \forall l$ (equations corresponding to Q1) and $\{V_{\text{tot},m,k,l}^2, V_{i,m,k,l}^2(\forall i), T_{\text{start},m,k,l}^2, T_{\text{end},m,k,l}^2, w_{m,k,l}^2\} \forall m \in M, \forall k \in K_m, \forall l$ (equations corresponding to Q2). Further, since these newly formed variables are duplicates of the variables present in the original model, they are related by the following equality constraints which are added to model (P):

$$V_{\text{tot},m,k,l}^1 - V_{\text{tot},m,k,l}^2 = 0 \quad \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B40})$$

$$V_{i,m,k,l}^1 - V_{i,m,k,l}^2 = 0 \quad \forall i \in I, \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B41})$$

$$T_{\text{start},m,k,l}^1 - T_{\text{start},m,k,l}^2 = 0 \quad \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B42})$$

$$T_{\text{end},m,k,l}^1 - T_{\text{end},m,k,l}^2 = 0 \quad \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B43})$$

$$w_{m,k,l}^1 - w_{m,k,l}^2 = 0 \quad \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B44})$$

Equations (40) – (44) are then dualized, that is, they are multiplied by the fixed values of Lagrange

multipliers $\lambda_{m,k,l}^{V_{\text{tot}}}, \lambda_{i,m,k,l}^{V_C}(\forall i), \lambda_{m,k,l}^{T_{\text{start}}}, \lambda_{m,k,l}^{T_{\text{end}}}, \lambda_{m,k,l}^w \forall m \in M, \forall k \in K_m, \forall l$,

respectively, and transferred to the objective function. The initial values of the

Lagrange multipliers are chosen arbitrarily and updated using the method given in Section 3.3. Dualizing Eqs. (40) – (44) yields a Lagrangean relaxation of the original problem, which is denoted by (LRP), and is decomposable into smaller sub-problems corresponding to Q1 and Q2, which are easier to solve.

The model (LRP) is decomposed into two smaller sub-problems (LQ1) and (LQ2) such that model (LQ1) includes equations and variables pertaining to structure Q1, while model (LQ2) includes equations and variables corresponding to the structure Q2. The bounds of all the non-linking variables in both the sub-problems are the same as in the original full space problem (P). For the case of the duplicate variables, their bounds are the same as the bounds of the corresponding linking variables in the original problem. The two models (LQ1) and (LQ2) are as follows:

$$\begin{aligned}
\min z^{\text{LQ1}} = & C_{\text{sea}} \sum_{u \in U_{Q1}} (T_{\text{start},u} - T_u^{\text{arrival}}) + C_{\text{unload}} \sum_{u \in U_{Q1}} (T_{\text{start},u} - T_u^{\text{arrival}}) \\
& + \frac{H \times \sum_{b \in B_{Q1}} C_{\text{inv}_b} \times \left[\left(\sum_{b \in B_{Q1}} \sum_l I_{\text{tot},b,l} \right) + \left(\sum_{b \in B_{Q1}} \sum_l \sum_{a \in A_b} V_{\text{tot},a,b,l} \right) + \left(\sum_{b \in B_{Q1}} \sum_{l \in |L|} I_{\text{tot},b,l} \right) + \left(\sum_{b \in B_{Q1}} 2I_b^{\text{init-tot}} \right) \right]}{(2 \times \text{NE} + 1)} \\
& + C_{\text{set}} \left(\sum_{d \in D_{Q1}} \sum_{g \in G_d} \sum_l w_{g,d,l} - \text{ND}_{Q1} \right) + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Vtot}} V_{\text{tot},m,k,l}^1 + \sum_l \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{i,m,k,l}^{\text{VC}} V_{i,m,k,l}^1 \\
& + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Tstart}} T_{\text{start},m,k,l}^1 + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Tend}} T_{\text{end},m,k,l}^1 + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^w W_{m,k,l}^1
\end{aligned}$$

s.t. constraints corresponding to units and connections in Q1 (LQ1)

$$\begin{aligned}
\min z^{\text{LQ2}} = & C_{\text{sea}} \sum_{u \in U_{Q2}} (T_{\text{start},u} - T_u^{\text{arrival}}) + C_{\text{unload}} \sum_{u \in U_{Q2}} (T_{\text{start},u} - T_u^{\text{arrival}}) \\
& + \frac{H \times \sum_{b \in B_{Q2}} C_{\text{inv}_b} \times \left[\left(\sum_{b \in B_{Q2}} \sum_l I_{\text{tot},b,l} \right) + \left(\sum_{b \in B_{Q2}} \sum_l \sum_{a \in A_b} V_{\text{tot},a,b,l} \right) + \left(\sum_{b \in B_{Q2}} \sum_{l \in |L|} I_{\text{tot},b,l} \right) + \left(\sum_{b \in B_{Q2}} 2I_b^{\text{init-tot}} \right) \right]}{(2 \times \text{NE} + 1)} \\
& + C_{\text{set}} \left(\sum_{d \in D_{Q2}} \sum_{g \in G_d} \sum_l w_{g,d,l} - \text{ND}_{Q2} \right) + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Vtot}} V_{\text{tot},m,k,l}^2 + \sum_l \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{i,m,k,l}^{\text{VC}} V_{i,m,k,l}^2 \\
& + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Tstart}} T_{\text{start},m,k,l}^2 + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{\text{Tend}} T_{\text{end},m,k,l}^2 + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^w W_{m,k,l}^2
\end{aligned}$$

s.t. constraints corresponding to units and connections in Q2 (LQ2)

The MILP relaxations of models (LQ1) and (LQ2), termed (LQ1-R) and (LQ2-R), respectively, are constructed by replacing the nonlinear terms in these models by convex envelopes. Models (LQ1-R) and (LQ2-R) are solved to obtain their optimal objective values z_{Q1}^* and z_{Q2}^* , respectively. Using these solutions, the following valid linear cuts are generated in the full space of the original problem, which are given by Eqs. (45) and (46):

$$\begin{aligned}
z_{Q1}^* \leq & C_{sea} \sum_{u \in U_{Q1}} (T_{start,u} - T_u^{arrival}) + C_{unload} \sum_{u \in U_{Q1}} (T_{start,u} - T_u^{arrival}) \\
& + \frac{H \times \sum_{b \in B_{Q1}} C_{inv_b} \times \left[\left(\sum_{b \in B_{Q1}} \sum_l I_{tot,b,l} \right) + \left(\sum_{b \in B_{Q1}} \sum_l \sum_{a \in A_b} V_{tot,a,b,l} \right) + \left(\sum_{b \in B_{Q1}} \sum_{l \in |L|} I_{tot,b,l} \right) + \left(\sum_{b \in B_{Q1}} 2I_b^{init-tot} \right) \right]}{(2 \times NE + 1)} \\
& + C_{set} \left(\sum_{d \in D_{Q1}} \sum_{g \in G_d} \sum_l w_{g,d,l} - ND_{Q1} \right) + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{V_{tot}} V_{tot,m,k,l} + \sum_i \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{i,m,k,l}^{VC} V_{i,m,k,l} \\
& + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{T_{start}} T_{start,m,k,l} + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{T_{end}} T_{end,m,k,l} + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^w W_{m,k,l}
\end{aligned} \tag{B45}$$

$$\begin{aligned}
z_{Q2}^* \leq & C_{sea} \sum_{u \in U_{Q2}} (T_{start,u} - T_u^{arrival}) + C_{unload} \sum_{u \in U_{Q2}} (T_{start,u} - T_u^{arrival}) \\
& + \frac{H \times \sum_{b \in B_{Q2}} C_{inv_b} \times \left[\left(\sum_{b \in B_{Q2}} \sum_l I_{tot,b,l} \right) + \left(\sum_{b \in B_{Q2}} \sum_l \sum_{a \in A_b} V_{tot,a,b,l} \right) + \left(\sum_{b \in B_{Q2}} \sum_{l \in |L|} I_{tot,b,l} \right) + \left(\sum_{b \in B_{Q2}} 2I_b^{init-tot} \right) \right]}{(2 \times NE + 1)} \\
& + C_{set} \left(\sum_{d \in D_{Q2}} \sum_{g \in G_d} \sum_l w_{g,d,l} - ND_{Q1} \right) + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{V_{tot}} V_{tot,m,k,l} + \sum_i \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{i,m,k,l}^{VC} V_{i,m,k,l} \\
& + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{T_{start}} T_{start,m,k,l} + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^{T_{end}} T_{end,m,k,l} + \sum_{m \in M} \sum_{k \in K_m} \sum_l \lambda_{m,k,l}^w W_{m,k,l}
\end{aligned} \tag{B46}$$

Theoretical properties of such cuts are given in Karupiah and Grossmann (2007). Namely, the cuts are valid when added to the original problem, and the inclusion of the cuts into the relaxation (R) produces a lower bound at least as strong as the lower bound obtained from Lagrangean decomposition and the one obtained by solving (R) without any cuts. The Lagrange multipliers used in these cuts can be updated using a

procedure given in Section 3.3, and additional cuts can be derived as described above. This procedure of updating the multipliers and adding cuts can be performed for any number of times. It is important to note that the performance of these cuts in reducing the solution time of the relaxation strongly depends on the values of the Lagrange multipliers. The cuts (Eqs. (45) and/or (46)) are then added to (R) which is the MILP relaxation of model (P) to get a modified MILP model (RP) . On solving (RP) , a valid lower bound on the solution of (P) is obtained.

B3.2.3 Upper Bounding Subproblem

The binary variables in problem (P) are fixed to the optimal values of the corresponding binary variables obtained from the solution of (RP) and obtain a nonconvex NLP (P-NLP), which is solved to global optimality with any standard method. To solve (P-NLP), the optimal values of the continuous variables obtained from the solution of (RP) are used as an initial point for obtaining an upper bound with the NLP solver. Solving model (P-NLP) yields an upper bound on the solution of (P) . In case the model (P-NLP) is found to be infeasible, an upper bound can still be obtained heuristically by first finding a sub-optimal integer solution to (RP) , and then using this integer solution to fix the binaries in (P) and re-solving the model (P-NLP) so obtained.

B3.2.4 Outer-Approximation Algorithm

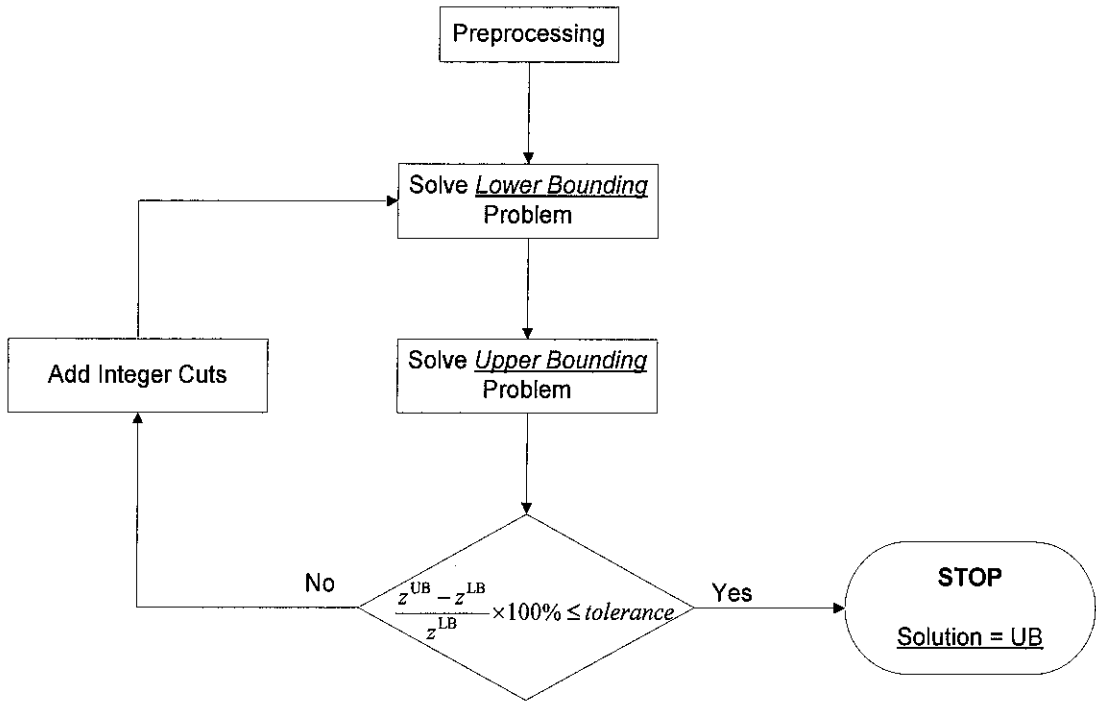


FIGURE B3.3: Proposed outer-approximation algorithm

The proposed outer-approximation algorithm is shown in flowchart in Figure B3.3. The algorithm is along the lines of the techniques proposed by Duran and Grossmann (1986), Kesavan et al. (2004), and Wei et al. (2005) and is outlined as follows:

(a) Preprocessing

The bounds of the variables in the model are determined by physical inspection of the network structure and using the numerical data given for the tanks, crude supply streams and the distillation units. Also, in this step, the original nonconvex MINLP may be locally optimized to obtain an initial overall upper bound (OUB) for the objective function.

(b) Lower bound generation

Generate a valid lower bound on the solution of the nonconvex MINLP following the technique outlined in Section B3.2.1.

(c) Upper bound generation

Generate an upper bound using the method given in Section B3.2.2 and update the OUB if the current upper bound is found to be better than the existing OUB.

(d) Integer cuts

Using the integer solution obtained from solving (RP) , add an integer cut to model (RP) to exclude this particular combination of binary variables. It is important to note that if the model $(P-NLP)$ is not globally optimized in step (c), adding these integer cuts to the relaxation in the next iteration could potentially cut off the global optimum.

(e) Termination

Iterate between solving models (RP) and $(P-NLP)$ until the optimality gap between the lower and upper bounds is less than a specified tolerance. The optimality gap is similar to that in Rocha et al. (2008). In this case, our tolerance is 5%. Convergence to the global optimum is not guaranteed if a local NLP solver is used in step (c) above.

B3.3 UPDATING LAGRANGE MULTIPLIERS

Fisher (1981) proposed a sub-gradient method to update Lagrange multipliers, to be used in solving a Lagrangean relaxation of a given MILP, starting from an initial arbitrary value of the multipliers. This technique is tailored to suit our problem in order to obtain updated values of Lagrange multipliers starting with random initial values. These Lagrange multipliers starting with random initial values are then used to derive cuts to be added to the MILP. A sequence of the multipliers is generated as follows:

$$\left[\lambda_{m,k,l}^{V_{\text{tot}}}\right]^{o+1} = \left[\lambda_{m,k,l}^{V_{\text{tot}}}\right]^o + ts^o \left[\left(V_{\text{tot},m,k,l}^1\right)^{*o} - \left(V_{\text{tot},m,k,l}^2\right)^{*o} \right] \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B46})$$

$$\left[\lambda_{i,m,k,l}^{VC}\right]^{o+1} = \left[\lambda_{i,m,k,l}^{VC}\right]^o + ts^o \left[\left(V_{i,m,k,l}^1\right)^{*o} - \left(V_{i,m,k,l}^2\right)^{*o} \right] \forall i \in I, \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B47})$$

$$\left[\lambda_{m,k,l}^{T_{\text{start}}}\right]^{o+1} = \left[\lambda_{m,k,l}^{T_{\text{start}}}\right]^o + ts^o \left[\left(T_{\text{start},m,k,l}^1\right)^{*o} - \left(T_{\text{start},m,k,l}^2\right)^{*o} \right] \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B48})$$

$$\left[\lambda_{m,k,l}^{T_{\text{end}}}\right]^{o+1} = \left[\lambda_{m,k,l}^{T_{\text{end}}}\right]^o + ts^o \left[\left(T_{\text{end},m,k,l}^1\right)^{*o} - \left(T_{\text{end},m,k,l}^2\right)^{*o} \right] \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B49})$$

$$\left[\lambda_{i,m,k,l}^w\right]^{o+1} = \left[\lambda_{i,m,k,l}^w\right]^o + ts^o \left[\left(w_{m,k,l}^1\right)^{*o} - \left(w_{m,k,l}^2\right)^{*o} \right] \forall m \in M, \forall k \in K_m, \forall l \quad (\text{B50})$$

Scalar step size, ts^o , is calculated at every iteration o :

$$ts^o = \frac{\alpha^o \left(z^U - z^L \left(\lambda^o \right) \right)}{\sum_{m \in M} \sum_{k \in K_m} \sum_l \left[\left(\left(V_{\text{tot},m,k,l}^1 \right)^{*o} - \left(V_{\text{tot},m,k,l}^2 \right)^{*o} \right) + \sum_i \left(\left(V_{i,m,k,l}^1 \right)^{*o} - \left(V_{i,m,k,l}^2 \right)^{*o} \right) \right.} \\ \left. + \left(\left(T_{\text{start},m,k,l}^1 \right)^{*o} - \left(T_{\text{start},m,k,l}^2 \right)^{*o} \right) + \left(\left(T_{\text{end},m,k,l}^1 \right)^{*o} - \left(T_{\text{end},m,k,l}^2 \right)^{*o} \right) + \left(\left(w_{m,k,l}^1 \right) - \left(w_{m,k,l}^2 \right) \right) \right] \quad (\text{B51})$$

Optimal values of the duplicates variables, obtained from the solution of the sub-problems (LQ1-R) and (LQ2-R), at the o th iteration, are represented by

$\left\{ \left(V_{\text{tot},m,k,l}^1 \right)^{*o}, \left(V_{i,m,k,l}^1 \right)^{*o} \forall i, \left(T_{\text{start},m,k,l}^1 \right)^{*o}, \left(T_{\text{end},m,k,l}^1 \right)^{*o}, \left(w_{m,k,l}^1 \right)^{*o} \right\}$ and

$\left\{ \left(V_{\text{tot},m,k,l}^2 \right)^{*o}, \left(V_{i,m,k,l}^2 \right)^{*o} \forall i, \left(T_{\text{start},m,k,l}^2 \right)^{*o}, \left(T_{\text{end},m,k,l}^2 \right)^{*o}, \left(w_{m,k,l}^2 \right)^{*o} \right\}$, respectively. α^o is a scalar

chosen between 0 and 2, $z^L \left(\lambda^o \right)$ is the sum of the objectives of the sub-models (LQ1-R) and (LQ2-R), when the multipliers are set to λ^o , and z^U is the value of the best found feasible solution of (R). The multiplier update is carried out only once in this work.

B3.4 BOUNDS CONTRACTION ALGORITHM

The branch-and-contract algorithm proposed by Zamora and Grossmann (1999) aims at reducing the size of the search region by eliminating portions of the domain in which the objective function takes only values above a known upper bound. The solution of contraction subproblems at selected branch-and-bound nodes is performed within a finite contraction operation that helps reducing the total number of nodes in the branch and bound solution tree.

A summary of the steps involved in performing the branch and contract algorithm is as follows:

- I. Define set of nonconvex or complicating variables as the subset of variables that appear in the nonconvex functions or terms in the model. The complicating variables are characterized by the index set CV .
- I. Specify minimum value, SP_{min} , for a successful contraction step. Specify the parameter ϵ_x , which determines the ϵ_x -closeness property. Specify the maximum number of contraction steps to be performed, NC_{max} . specify the maximum number of unsuccessful contraction steps, NUC_{max} . specify the index set $BLUE_0 \in CV$ that determines the subset of complicating variables over which contraction is to be performed (contraction variables). Initialize the control sets $BLUE := BLUE_0$, and $RED := \emptyset$. Specify the maximum fraction of contraction variables, F_{CV} , allowed in the RED set. Initialize counters $NC = 0$, $NUC = 0$.
- II. For $i \in BLUE$, compute the lower focal distance, $\Delta_i^{f,L}(\Omega)$, and upper focal distance, $\Delta_i^{f,U}(\Omega)$, using the formula below:

$$\Delta_i^{f,L}(\Omega) = \frac{x_i^b - x_i^L}{x_i^{U,in} - x_i^{L,in}}$$

$$\Delta_i^{f,U}(\Omega) = \frac{x_i^U - x_i^b}{x_i^{U,in} - x_i^{L,in}}$$

Label all these focal distances as unmarked. If the focal point at a branch and bound node, x^b is ε_x -close to x_i^L , i.e. $x_i^b \leq x_i^L(1 + \varepsilon_x)$, mark $\Delta_i^{f,L}(\Omega)$. If x^b is ε_x -close to x_i^U , i.e. $x_i^b \geq x_i^U(1 - \varepsilon_x)$, then mark $\Delta_i^{f,U}(\Omega)$.

III. Determine

$$\Delta_{\max}^f(\Omega) = \text{Max}_{i \in \text{BLUE}} \left[\Delta_i^{f,L}(\Omega), \Delta_i^{f,U}(\Omega) \right]$$

for $\Delta_i^{fL}(\Omega), \Delta_i^{fU}(\Omega)$, unmarked

Then, select a complicating variable x_i , with $i \in \text{BLUE}$, such that

$$\Delta_i^{f,L}(\Omega) \text{ is unmarked and } \Delta_i^{f,L}(\Omega) = \Delta_{\max}^f(\Omega)$$

or

$$\Delta_i^{f,U}(\Omega) \text{ is unmarked and } \Delta_i^{f,U}(\Omega) = \Delta_{\max}^f(\Omega)$$

IV. Here, we perform a contraction step, i.e. the process of computing and updating a bound, x_i^L or x_i^U , through the solution of the following equation:

$$\text{Min or Max } x_i \\ (x,y,z)$$

subject to

$$\hat{f}(x, y, z) \leq OUB$$

$$(x, y, z) \in M(\Omega) \subset R^n \times R^{n1} \times R^{n2}$$

The inequality $\hat{f}(x, y, z) \leq OUB$ will be called the OUB constraint, and the solution to the problem will be denoted as $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})_{\Omega}$. The optimization direction, i.e. Min or Max, is selected depending upon which of the bounds, x_i^L or x_i^U , is to be contracted.

If $\Delta_i^{f,L}(\Omega)$ is unmarked and $\Delta_i^{f,L}(\Omega) = \Delta_{\max}^f(\Omega)$ perform a contraction step with a Min direction to contract x_i^L . Otherwise, perform a contraction step with a Max direction to contract x_i^U . Set $NC := NC + 1$.

- V. If the contraction subproblem in step V is feasible, then:
Determine SP using the formula below:

$$SP = \begin{cases} \left(\frac{\mathcal{X}_i^0 - x_i^L}{x_i^U - x_i^L} \right) & \text{if the optimization direction = Min} \\ \left(\frac{x_i^U - \mathcal{X}_i^0}{x_i^U - x_i^L} \right) & \text{if the optimization direction = Max} \end{cases}$$

where \mathcal{X}_i^0 is the minimum (maximum) value of the variable x_i , obtained by solving the contraction step with a Min (Max) direction.

If x_i^L was contracted and $SP < SP_{\min}$, then mark $\Delta_i^{f,L}(\Omega)$ and set $NUC := NUC + 1$.

If x_i^L was contracted and $x_i^b < x_i^L$, then set $x_i^b := x_i^L$.

If x_i^U was contracted and $SP < SP_{\min}$, then mark $\Delta_i^{f,U}(\Omega)$ and set $NUC := NUC + 1$.

If x_i^U was contracted and $x_i^b > x_i^U$, then set $x_i^b := x_i^U$.

Compute $\varepsilon(\Omega)$.

- VI. If the contraction subproblem in step V is infeasible or $\varepsilon(\Omega) \leq \varepsilon_0$, then terminate the contraction operation.

Terminate the contraction operation if any of the following conditions is met: (i) $|RED| \geq F_{CV}|BLUE_0|$; (ii) $NC = NC_{max}$; (iii) $NUC = NUC_{max}$. Otherwise, return to step IV.

Bounds contraction will be applied to equation (B19), which is shown below:

$$\left(I_{tot,b,(l-1)} + \sum_{a \in A_b} V_{tot,a,b,l} \right) V_{i,b,c,l} = \left(I_{i,b,(l-1)} + \sum_{a \in A_b} V_{i,a,b,l} \right) V_{tot,b,c,l} \quad \forall b \in B, \forall c \in C_b, \forall i \in I, \forall l \in L$$

This constraint contains bilinear terms (variable multiplied by another variable) that give rise to be the nonconvexities of the model. The complicating variables that appear in equation (B19) and are evaluated individually using the branch and contract algorithm are:

- total inventory of tank b at the end of transfer event l ($I_{tot,b,l}$);
- inventory of component i in tank b at the end of transfer event l ($I_{i,b,l}$);
- total flow from a to b in transfer event l ($V_{tot,a,b,l}$);
- total flow from b to c in transfer event l ($V_{tot,b,c,l}$);
- flow of component i from a to b in transfer event l ($V_{i,a,b,l}$);
- flow of component i from b to c in transfer event l ($V_{i,b,c,l}$);

B3.5 SLACK VARIABLES

Slack variables are used in optimization to turn constraints stated as inequalities into equalities. This is required to turn an inequality into an equality where a linear combination of variables is less than or equal to a given constant in the former.

The model developed was found to have integer infeasibilities, i.e. there is no assignment of integer values to variables that can satisfy all of the constraints. The problematic constraints have been identified, namely equations (B3.i), (B13), (B15), (B17), (B18), (B21), (B25), (B26), (B36.iii), (B36.iv), (B37.i), (B37.iii), (B38.iv), and (B39.i).

In order to overcome the integer infeasibility problem, slack variables, s , are introduced to these equations, as follows:

$$F_{a,b}^U (T_{\text{end},a,b,l} - T_{\text{start},a,b,l}) + F_{a,b}^U H (1 - w_{a,b,l}) \geq V_{\text{tot},a,b,l} + s_{a,b,l}^{3,i} \quad \forall a \in A_b, \forall b \in B, \forall l \in L \quad (\text{B3.i})$$

$$I_{\text{tot},b,(l-1)} + \sum_{a \in A_b} V_{\text{tot},a,b,l} = I_{\text{tot},b,l} + \sum_{c \in C_b} V_{\text{tot},b,c,l} + s_{b,c,l}^{13} \quad \forall b \in B, \forall l \in L \quad (\text{B13})$$

$$I_{i,b,(l-1)} + \sum_{a \in A_b} V_{i,a,b,l} = I_{i,b,l} + \sum_{c \in C_b} V_{i,b,c,l} + s_{i,b,l}^{15} \quad \forall i \in I, \forall b \in B, \forall l \in L \quad (\text{B15})$$

$$V_{\text{tot},a,b,l} = \sum_{i \in I} V_{i,a,b,l} + s_{a,b,l}^{17} \quad \forall a \in A_b, \forall b \in B, \forall l \in L \quad (\text{B17})$$

$$V_{\text{tot},b,c,l} = \sum_{i \in I} V_{i,b,c,l} + s_{b,c,l}^{18} \quad \forall c \in C_b, \forall b \in B, \forall l \in L \quad (\text{B18})$$

$$f_{i,b}^L I_{\text{tot},b,l} \leq I_{i,b,l} + s_{i,b,l}^{27} \quad \forall i \in I, \forall b \in B, \forall l \in L \quad (\text{B21})$$

$$\sum_{g \in G_d} w_{g,d,l} \leq 1 + s_{d,l}^{25} \quad \forall d \in D, \forall l \in L \quad (\text{B25})$$

$$\sum_{d \in D_g} w_{g,d,l} \leq 1 + s_{g,l}^{26} \quad \forall g \in G, \forall l \in L \quad (\text{B26})$$

$$I_{i,b,c,l}^{VI} \leq I_b^L V_{i,b,c,l} + V_{b,c}^U I_{\text{tot},b,(l-1)} - I_b^L V_{b,c}^U + s_{i,b,c,l}^{36,iii} \quad \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \quad (\text{B36.iii})$$

$$I_{i,b,c,l}^{VI} \leq I_b^U V_{i,b,c,l} + V_{b,c}^L I_{\text{tot},b,(l-1)} - I_b^U V_{b,c}^L + s_{i,b,c,l}^{36,iv} \quad \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B \quad (\text{B36.iv})$$

$$I_{i,b,c,l}^{VT} \geq I_b^L V_{\text{tot},b,c,l} + V_{b,c}^L I_{i,b,(l-1)} - I_b^L V_{b,c}^L + S_{i,b,c,l}^{37.i} \quad \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B$$

(B37.i)

$$I_{i,b,c,l}^{VT} \leq I_b^L V_{\text{tot},b,c,l} + V_{b,c}^U I_{i,b,(l-1)} - I_b^L V_{b,c}^U + S_{i,b,c,l}^{37.iii} \quad \forall i, \forall c \in C_b, \forall l \in L, \forall b \in B$$

(B37.iii)

$$V_{i,a,b,c,l}^{VJ} \leq V_{a,b}^U V_{i,b,c,l} + V_{b,c}^L V_{\text{tot},a,b,l} - V_{a,b}^U V_{b,c}^L + S_{i,a,b,c,l}^{38.iv} \quad \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B$$

(B38.iv)

$$V_{i,a,b,c,l}^{VT} \geq V_{a,b}^L V_{\text{tot},b,c,l} + V_{b,c}^L V_{i,a,b,l} - V_{a,b}^L V_{b,c}^L + S_{i,a,b,c,l}^{39.i} \quad \forall i, \forall a \in A_b, \forall c \in C_b, \forall l \in L, \forall b \in B$$

(B39.i)

B3.6 MODEL DATA FOR NUMERICAL EXAMPLE

The values of the parameters are obtained from Lee et al. (1996). The units of some of the parameters are not specified in order to be consistent with these authors.

The problem is a network consisting of 3 crude supply streams, 3 storage tanks, 3 charging tanks, and 2 distillation units, whose structure is shown in Figure B3.4 below.

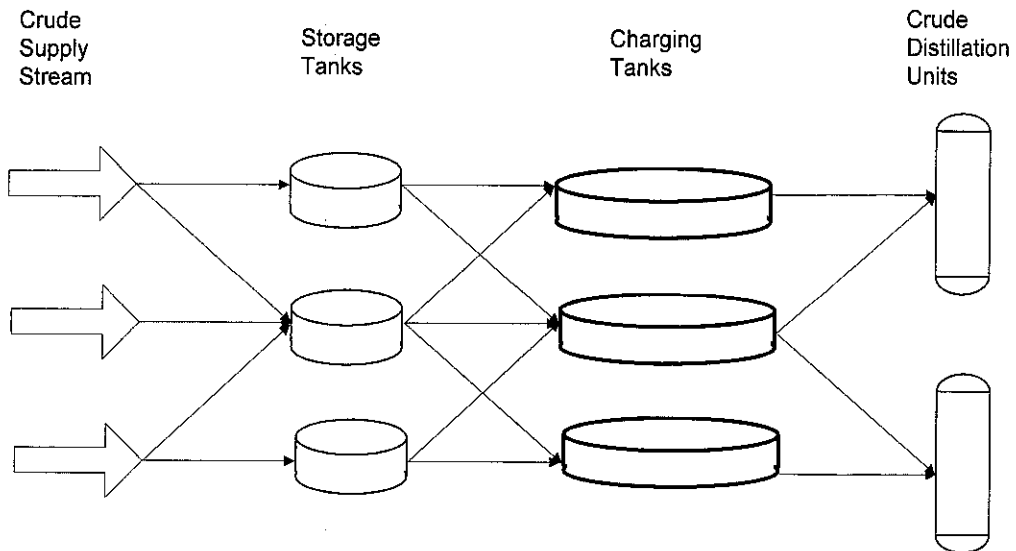


FIGURE B3.4: Connections in a refinery network

The crude oil in this example contains one key component and all the other components are combined into a bulk component, thus effectively making the given crude a two component system. The crude movement has to be scheduled over a time horizon of 12 h.

The values are shown in Table B3.1 below:

TABLE B3.1: Data for crude oil scheduling computation

Scheduling horizon (H): 12 h
Number of crude supply streams: 3

Crude supply stream (u)	Arrival time ($T_u^{arrival}$)	Incoming volume of crude (V_u^{supply})	Fraction of key component ($f_{i,u}^{supply}$)
U1	1	50	0.01
U2	5	50	0.085
U3	9	50	0.06

Number of storage tanks: 3

Storage tank (v)	Capacity (I_b^U)	Initial oil inventory ($I_b^{init-tot}$)	Initial fraction of key component (min-max) [$I_{i,b}^{init}$ ($f_{i,b}^L - f_{i,b}^U$)]
ST1	100	20	0.02 (0.01-0.03)
ST2	100	20	0.05 (0.04-0.06)
ST3	100	20	0.08 (0.07-0.09)

Number of charging tanks: 3

Charging tank (g)	Capacity (I_b^U)	Initial oil inventory ($I_b^{init-tot}$)	Initial fraction of key component (min-max) [$I_{i,b}^{init}$ ($f_{i,b}^L - f_{i,b}^U$)]
CT1	100	30	0.03 (0.025-0.035)
CT2	100	50	0.05 (0.045-0.065)
CT3	1000	30	0.08 (0.075-0.085)

Number of CDUs (ND): 2
Unloading cost for crude supply streams (Cunload): 10
Changeover cost for charged oil switch (Cset): 50
Waiting cost for crude supply streams (Csea): 5
Tank inventory costs (Cinv_b): 0.04 (storage tanks); 0.08 (charging tanks)
Demand of mixed oils by CDUs (DM_g): 50 (oil mix from CT1); 50 (oil mix from CT2); 50 (oil mix from CT3)
Upper bounds on flowrates in the streams ($F_{a,b}^U$): 40

The initial values of Lagrange multipliers are shown in Table B3.2 below:

TABLE B3.2: Lagrange multiplier information

Lagrange multipliers	Initial values
$\lambda_{m,k,l}^{V_{tot}}$	0
$\lambda_{i,m,k,l}^{VC}$	0
$\lambda_{m,k,l}^{T_{start}}$	0
$\lambda_{m,k,l}^{T_{end}}$	0
$\lambda_{m,k,l}^w$	0

The values from Table B3.1 and B3.2 are input into the GAMS model, and allowed to run, to determine the value of the objective function, which is to minimize cost.

CHAPTER B4

COMPUTATIONAL EXPERIMENTS AND DISCUSSIONS ON NUMERICAL RESULTS

B4.1 COMPUTATIONAL RESULTS

The formulated logical constraints are coded on GAMS. Numerical studies and computational experiments of the proposed model in this paper is implemented on GAMS 22.3 for Windows XP platform. The computing facilities along with its attributes are listed in Table B4.1 below:

TABLE B4.1: Attributes of computing facilities for computational experiments

Computer Type	Laptop (Lenovo)
Processor Type	Intel Centrino Duo
Processor Speed	1.73 GHz
RAM	512 MB

The computational statistics including the CPU times on the mentioned machine are as reported in Table 4.2 below:

TABLE B4.2: Computational statistics for all models

Solver	Model	Number of single continuous variables	Number of discrete (binary 0-1) variables	Number of single equations	Resource usage/CPU time (s)	Number of iterations
CPLEX 10	LQ1	1615	155	3423	0.516	207
CPLEX 10	LQ2	1561	161	3405	0.109	173
CPLEX 10	LB - MILP	1963	155	3797	0.14	214
CONOPT	UB - NLP	752	0	2097	0.41	0

The computational result is divided into two parts, which are bound contraction and the overall results.

B4.1.1 Results for Bounds Contraction Procedure

Bounds contraction strategy as discussed in section B3.4 earlier was implemented to deal with the bilinear terms, i.e. the complicating variables, in our model, which are:

- total inventory of tank b at the end of transfer event l ($I_{tot,b,l}$);
- inventory of component i in tank b at the end of transfer event l ($I_{i,b,l}$);
- total flow from a to b in transfer event l ($V_{tot,a,b,l}$);
- total flow from b to c in transfer event l ($V_{tot,b,c,l}$);
- flow of component i from a to b in transfer event l ($V_{i,a,b,l}$);
- flow of component i from b to c in transfer event l ($V_{i,b,c,l}$);

Bounds contraction techniques have been carried out for each of the complicating variables individually, thus resulting in six separate GAMS codes.

The results obtained shows that there are no apparent improvements in the bounds after carrying out the bounds contraction technique, as all of the codes produced only ONE iteration step. An example of the branch contraction model is shown in Appendix VI for total inventory of tank b at the end of transfer event l ($I_{tot,b,l}$).

B4.1.2 Overall Results

After applying the bounds contraction technique, we move on to solve the model using the proposed outer-approximation algorithm by solving four different models, that is, models LQ1, LQ2, LB–MILP, and UB–NLP. The intermediate result for LB – MILP is as shown in Figure B4.1. Then, Figure B4.2 shows the optimal network structure for the crude oil flow for this numerical example as provided by the optimal solution of model UB–NLP.

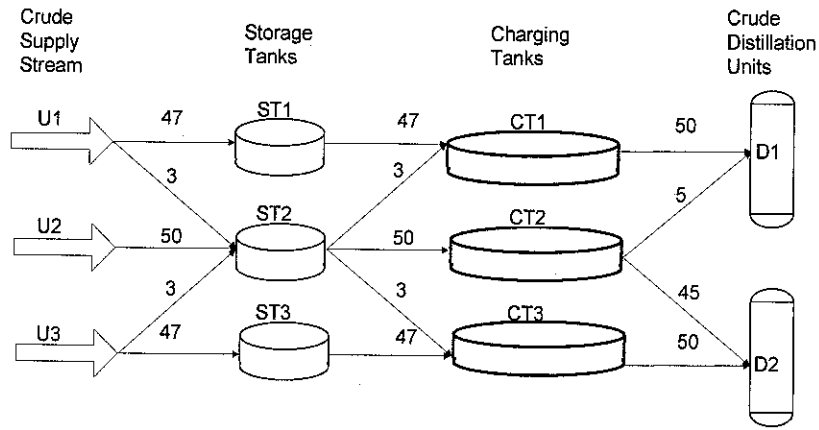


FIGURE B4.1: Flow of crude oil at refinery front-end for model LB-MILP

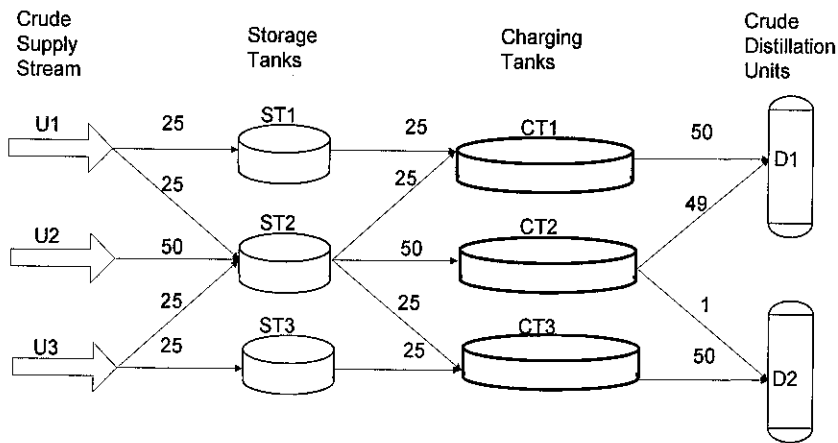


FIGURE B4.2: Optimal network structure for crude oil flow at refinery front-end

The optimal crude schedule is shown in Figure B4.3. The numbers above of the crude transfer line segments in the figure are the actual flow volumes. The inventory profiles of the tanks are not given for this example and for the subsequent numerical examples since the model includes only the times when the crude transfers begin and end, and there is no explicit information in the model pertaining to the start and end times of flow from a tank.

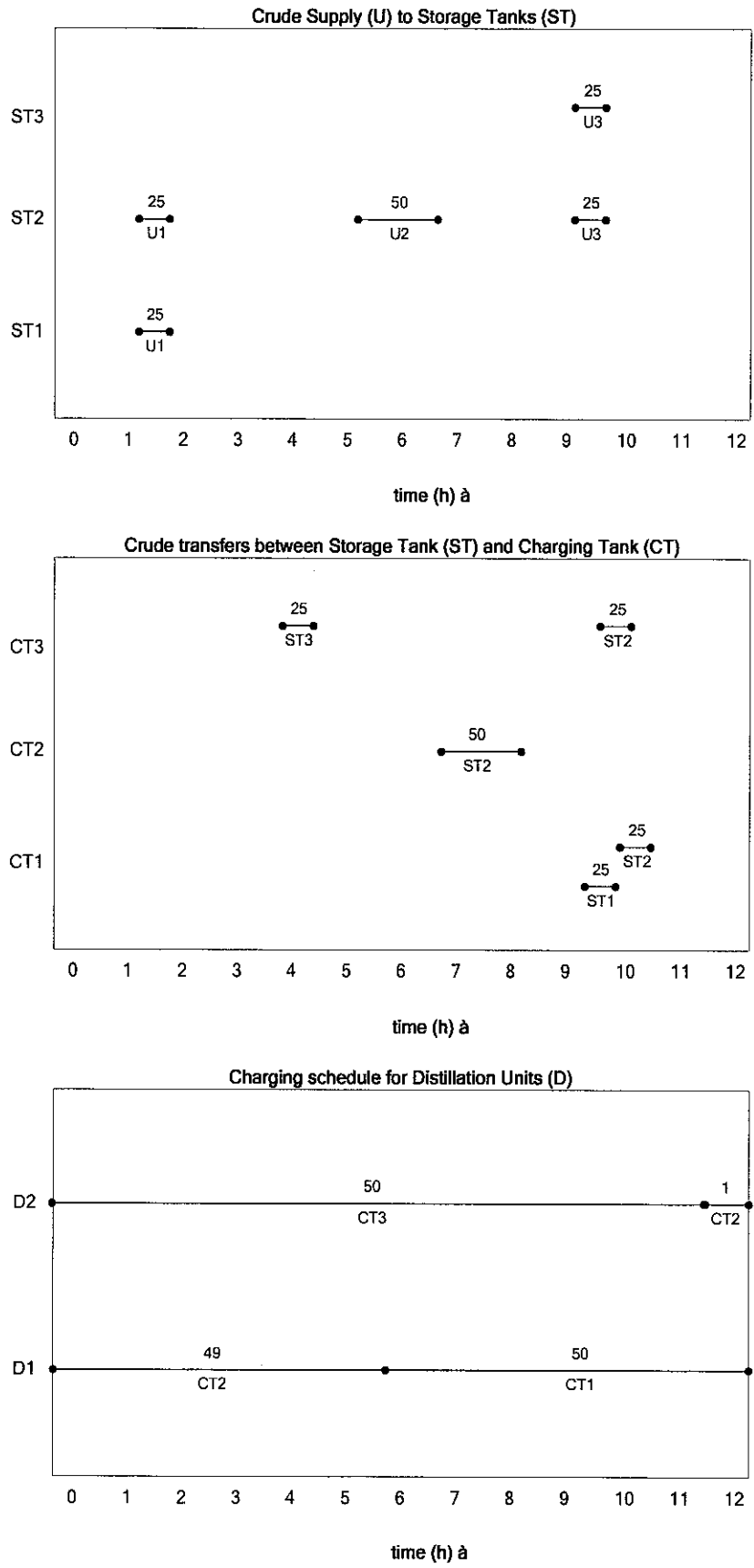


FIGURE B4.3: Gantt chart of the optimal crude oil schedule

A summary of the computational results obtained from all the models are shown in Table B4.3 below:

TABLE B4.3: Computational results for all models

(a) Binary variables (=1 if there is flow from source to destination)

No.	Source	Destination	Binary variables for model			
			LQ1	LQ2	LB - MILP	UB - NLP
1	U1	ST1	1	1	1	1
		ST2	1	1	1	1
		ST3	0	0	0	0
2	U2	ST1	1	1	1	1
		ST2	0	0	0	0
		ST3	0	0	0	0
3	U3	ST1	1	1	1	1
		ST2	1	1	1	1
		ST3	0	0	0	0
4	ST1	CT1	1	1	1	1
		CT2	0	0	0	0
		CT3	0	0	0	0
5	ST2	CT1	1	1	1	1
		CT2	1	1	1	1
		CT3	1	1	1	1
6	ST3	CT1	1	1	1	1
		CT2	0	0	0	0
		CT3	0	0	0	0
7	CT1	D1	1	1	1	1
		D2	0	0	0	0
8	CT2	D1	1	1	1	1
		D2	1	1	1	1
9	CT3	D1	1	1	1	1
		D2	0	0	0	0

(b) Total flow of oil

No.	Source	Destination	Total flow of oil for model			
			LQ1	LQ2	LB - MILP	UB - NLP
1	U1	ST1	47	47	47	25
		ST2	3	3	3	25
		ST3	0	0	0	0
2	U2	ST1	50	50	50	50
		ST2	0	0	0	0
		ST3	0	0	0	0
3	U3	ST1	3	3	3	25
		ST2	47	47	47	25
		ST3	0	0	0	0
4	ST1	CT1	47	47	47	25
		CT2	0	0	0	0
		CT3	0	0	0	0
5	ST2	CT1	3	3	3	25
		CT2	50	50	50	50
		CT3	3	3	3	25
6	ST3	CT1	47	47	47	25
		CT2	0	0	0	0
		CT3	0	0	0	0
7	CT1	D1	50	50	50	50
		D2	0	0	0	0
8	CT2	D1	3	47	5	49
		D2	47	3	45	1
9	CT3	D1	50	50	50	50
		D2	0	0	0	0

(c) Objection function (Cost)

Model	Objective function (Cost)
LQ1	538.6
LQ2	8.0114
LB - MILP	1980.44
UB - NLP	2148.105

B4.3 DISCUSSIONS

From the Gantt chart of the optimal crude oil schedule in Figure B4.3, we can see that the distillation columns are charged continuously. Besides that, the time taken for the transfers between the crude supply streams and the tanks, and for inter-tank transfers are small fractions of the overall scheduling horizon.

From the results obtained, it can be seen that the crude oil flows have non-zero values only for binary variables = 1, i.e. there is flow from source to destination. This is in accordance to the fact that, crude oil can only flow from chosen sets of source to their respective destinations.

Besides that, the results shows a consistent pattern for the total flow of crude oil from source to destination, where the values for one model shows very small difference to another model. For instance, all the flows from source to destination for models LQ1, LQ2 and LB – MILP are identical, except for the flow from CT2 to D1, and CT2 to D2. Although model UB – NLP shows a marked difference in total flow values, the decision variables, i.e. the binary variables that determines whether there is existence of flow from source to destination, are identical for all the models. This means that the decisions of the models are consistent to one another. This is logical since the parameters used are the same for all the models.

Results show that values of slack variables are non-zero for the equations that we have integer infeasibility problems. This shows that there is no assignment of integer values to variables that can satisfy all of the constraints. The use of slack variables managed to overcome the integer infeasibility problem by turning the constraints stated as inequalities into equalities.

The optimal objective function values from models LQ1 and LQ2 are used to generate the valid linear cuts in model LB–MILP by providing the values of z_{Q1}^* and z_{Q2}^* , respectively in constraints **(B45)** and **(B46)**. The binary variables for model UB – NLP are obtained from the solution from LB – MILP. Model UB – NLP uses the original equations, i.e. the solution of UB – NLP will be a global optimum (maximum), whereas model LB – MILP only generates lower bound on the global optimum (maximum), so the objective function from LB – MILP will be less than that of UB – NLP. This is shown from the results obtained, whereby the objective function for UB – NLP (2148.105) is higher than LB – MILP (1980.44).

CHAPTER B5

CONCLUSIONS AND RECOMMENDATIONS

B5.1 CONCLUSIONS

In this work, we have developed a technique for obtaining globally optimal schedules for the flow of crude oil at the front-end of a refinery. A continuous time model based on transfer events is used to represent the scheduling problem and this model is a nonconvex MINLP model which has multiple local optima. We implemented branch and contract algorithm that aims at reducing the size of the search region by eliminating portions of the domain in which the objective function takes only values above a known upper bound. In order to obtain a global optimum solution to the problem, an outer-approximation algorithm is proposed. In this approach, we generated lower and upper bounds on the global optimum, which were converged to a specified tolerance. A rigorous lower bound on the global optimum was obtained by solving a MILP relaxation of the original problem. To reduce the computational effort required in solving this MILP relaxation, cutting planes derived from a spatial decomposition of the network were added to the model LB – MILP. The solution obtained from the LB – MILP model, i.e. the decision variables (binary variables), was used to obtain a feasible solution for model UB – NLP. This solution is the upper bound solution. The application of the proposed algorithm shows significant reduction in the computational effort involved in solving the problem. Slack variables were introduced to overcome the integer infeasibility problem. The mathematical model was developed using GAMS and an optimal solution was found with no logical constraints conflicts or error.

B5.2 RECOMMENDATIONS

It is recommended that penalty should be imposed on the slack variables in the objective function. These penalty terms will give the slack variables' values

significance in the fact that it will be accounted for when determining the optimal objective function value.

REFERENCES

- Furman, K.C., Jia, Z.Y., Ierapetritou, M.G., A robust event – based continuous time formulation for tank transfer scheduling, *Ind. Eng. Chem. Res.* 46 **2007**, 9126 – 9136
- Goel, V, et al. A novel branch and bound algorithm for optimal development of gas fields under uncertainty in reserves. *Computers and Chemical Engineering* 30 **2006**, 1076 – 1092
- Goel, V, Grossmann, I.E. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. *Computers & Chemical Engineering* 24 **2004**, 1409 - 1429
- Higle, J.L., Stochastic programming: Optimization when uncertainty matters, *Tutorials in Operations Research*, **2005**
- Iyer, R.R.; Grossmann, I.E. Optimal planning and scheduling of offshore oilfield infrastructure investment and Operations. *Ind. Eng. Chem. Res.* **1998**, 37, 1380-1397
- JE Beasley, Imperial College, *Operations Research Lecture Notes*, **2002**
- Jonsbraten, T.W., Oil field optimization under price uncertainty. *Journal of the Operational research Society* 49 **1998**, 811-818
- Karuppiah, R., Furman, K.C., Grossmann, I.E., Global optimization of scheduling refinery crude oil operations, *Computers and Chemical Engineering* 32 **2008**, 2745 - 2766
- Karuppiah, R., Grossmann, I.E., A lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures, *Journal of Global Optimization* 41 **2008**, 163 - 186

Kelly, J.D., Mann, J.L., Crude oil blend scheduling optimization: an application with multimillion dollar benefits, *Hydrocarbon Processing*, **June 2003**, 47 - 53

Lee, H., Pinto, J.M., Grossmann, I.E., Park, S.W., Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management, *Ind. Eng. Chem. Res.* 35 **1996**, 1630 – 1641

Sen, S., Higle, J.L., An introductory tutorial on stochastic linear programming models, *Institute for Operations Research and the Management Sciences*, **1999**

Tarhan, B., Grossmann, I.E., A multistage stochastic programming approach with strategies for uncertainty reduction in the synthesis of process networks with uncertain yields. *Computers and Chemical Engineering*, **2007**

van den Heever, S.A.; Grossmann, I.E. An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. *Ins. Eng. Chem. Res.* **2000**, 39, 1955-1971

van den Heever, Susara A., Ignacio E. Grossmann, Sriram Vasantharajan, and Krisanne Edwards. Integrating complex economic objectives with the design and planning of offshore oilfield infrastructures. *Computers & Chemical Engineering* 24 **2000**, 1049–1055.

van den Heever, Susara A., Ignacio E. Grossmann, Sriram Vasantharajan, and Krisanne Edwards. A Lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. *Industrial & Engineering Chemistry Research* 40 **2001**, 13: 2857 – 2875.

Zamora, J.M., Grossmann, I.E., A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms, *Journal of Global Optimization* 14 **1999**, 217 – 249

APPENDICES

APPENDIX I: (A) GANTT CHART (1ST SEMESTER)

No.	Detail	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
2	Preliminary Research Work • Literature Review • Possible process routes														
3	Superstructure construction														
4	Model formulation														
5	Modeling in GAMS														
6	Interim Report Preparation														

APPENDIX II: (B) GANTT CHART (2ND SEMESTER)

No.	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project Work Continue														
2	Submission of Progress Report 1														
3	Project Work Continue														
4	Submission of Progress Report 2														
5	Seminar (compulsory)														
5	Project work continue														
6	Poster Exhibition														
7	Submission of Dissertation (soft bound)														
8	Oral Presentation														
9	Submission of Project Dissertation (Hard														

APPENDIX II: COMPUTER CODE FOR GAMS INPUT FILE ON DETERMINISTIC MODEL

```

$TITLE: OIL PRODUCTION PLANNING
(DETERMINISTIC MODEL)

FILE RESPLAN2 / UPSTREAM2.res /

PUT RESPLAN2;

SETS

T          SET OF TIME /T1*T6/
W          SET OF WELLS /W1*W4/
N          SET OF WELL PLATFORMS
/N1/
P          SET OF PRODUCTION
PLATFORMS /P1/
R          SET OF RESERVOIRS
/R1*R2/
F          SET OF FIELDS /F1/
J          SET OF INDEX OF PIECE
IN PIECEWISE LINEAR INTERPOLATION /J1/
D          SET OF DRILLING RIGS
/D1/
N1(N,P)   SET OF WELL PLATFORMS
N ASSOCIATED WITH PLATFORM P /N1.P1/
W1(W,N)   SET OF WELLS W
ASSOCIATED WITH WELL PLATFORM N
/(W1*W4).N1/
W2(W,R)   SET OF WELLS W
ASSOCIATED WITH RESERVOIR R
/W1.R1,W2.R1,W3.R2,W4.R2/
W3(W,R,N) SET OF WELLS W
ASSOCIATED WITH RESERVOIR R AND WELL
PLATFORM N
/(W1,W2).R1.N1,(W3,W4).R2.N1/
W4(W,R,F) SET OF WELLS W IN
FIELD F ASSOCIATED WITH RESERVOIR R
/W1.R1.F1,W2.R1.F1,W3.R2.F1,W4.R2.F1/
R1(R,F)   SET OF RESERVOIRS
ASSOCIATED WITH FIELD F /R1.F1/
J1(J,R)   SET OF LINEAR
INTERPOLATION PIECES USED FOR
RESERVOIR R /J1.R1/
SP        SET OF SCENARIOS FOR
PRICE UNCERTAINTY /L,M,H/
S         SET OF SCENARIOS FOR
UNCERTAINTY IN SIZE AND INITIAL
DELIVERABILITY OF WELL /S1*S9/
;

```

SCALARS

```

ALPHAN    PRESSURE DROP
COEFFICIENT FOR OIL FLOW RATE FOR WELL
PLATFORM /20/
ALPHAP    PRESSURE DROP
COEFFICIENT FOR OIL FLOW RATE FOR
PRODUCTION PLATFORM /20/
BETAN     PRESSURE DROP
COEFFICIENT FOR GOR FOR PRODUCTION
PLATFORM /18/
BETAP     PRESSURE DROP
COEFFICIENT FOR GOR FOR PRODUCTION
PLATFORM /18/
PMAX      MAXIMUM PRESSURE DROP
FROM WELL BORE TO WELL HEAD /8/
GORMAX    MAXIMUM GAS-TO-OIL
RATIO /11/

```

```

DELTAT    LENGTH OF TIME PERIOD
T (YEAR) /4/
ITERATION
;
ITERATION = 1;

```

PARAMETERS

```

RO(W,N,P) PRODUCTIVITY INDEX
OF WELL W CONNECTED TO WELL PLATFORM N
CONNECTED TO PRODUCTION PLATFORM P IN
TIME T DIMENSIONLESS
/
W1.N1.P1    47.6
W2.N1.P1    83.8
W3.N1.P1    158.4
W4.N1.P1    210.2
/

```

```

C1(T,SP)   DISCOUNTED REVENUE
PRICE COEFFICIENT FOR OIL SALES
/
(T1*T6).L    19
(T1*T6).M    22
(T1*T6).H    25
/

```

```

C2P(P,T)   DISCOUNTED FIXED COST
COEFFICIENT FOR INSTALLATION OF
PRODUCTION PLATFORM IN MILLIONS OF
DOLLARS
/
P1.T1 70
P1.T2 69
P1.T3 68
P1.T4 67
P1.T5 66
P1.T6 65
/

```

```

C2NP(N,P,T) DISCOUNTED FIXED COST
COEFFICIENT FOR INSTALLATION OF WELL
PLATFORM IN MILLIONS OF DOLLARS
/
N1.P1.T1 11
N1.P1.T2 10
N1.P1.T3 9
N1.P1.T4 8
N1.P1.T5 7
N1.P1.T6 6
/

```

```

C2WNP(W,N,P,T) DISCOUNTED FIXED COST
COEFFICIENT FOR DRILLING THE WELL IN
MILLIONS OF DOLLARS
/
W1.N1.P1.(T1*T6) 6.69
W2.N1.P1.(T1*T6) 6.34
W3.N1.P1.(T1*T6) 5.76
W4.N1.P1.(T1*T6) 5.83
/

```

```

C3P(P,T)   DISCOUNTED VARIABLE
COST COEFFICIENT FOR INSTALLATION OF
PRODUCTION PLATFORM IN MILLIONS OF
DOLLARS
/
P1.T1 0.47
P1.T2 0.46
P1.T3 0.45
P1.T4 0.44
P1.T5 0.43
P1.T6 0.42
/

```

```

C3NP(N,P,T) DISCOUNTED VARIABLE
COST COEFFICIENT FOR INSTALLATION OF
WELL PLATFORM IN MILLIONS OF DOLLARS

```

/

N1.P1.T1 0.19
 N1.P1.T2 0.18
 N1.P1.T3 0.17
 N1.P1.T4 0.16
 N1.P1.T5 0.15
 N1.P1.T6 0.14

/

C4(T) DISCOUNTED COST
 COEFFICIENT FOR MOVING RIGS IN
 MILLIONS OF DOLLARS

/

T1 50
 T2 49
 T3 48
 T4 47
 T5 46
 T6 45

/

V5(R,F,J,T) INDEX OF PIECE J USED
 TO CALCULATE PRESSURE IN RESERVOIR R
 IN FIELD F AT TIME T IN BAR
 (PRESPECIFIED FROM RESERVOIR
 PERFORMANCE CURVE)

/

R1.F1.J1.(T1*T6) 10

PROB(SP)

/

L 0.3
 M 0.4
 H 0.3

/

PROBA(S)

/

S1 0.09
 S2 0.12
 S3 0.09
 S4 0.12
 S5 0.16
 S6 0.12
 S7 0.09
 S8 0.12
 S9 0.09

TABLE

UID(W,N,P,T,S) SCENARIOS FOR
 UNCERTAINTY IN INITIAL
 DELIVERABILITIES

	S1	S2
S3 S4 S5 S6 S7		
(W1*W4).N1.P1.(T1*T6)	4	4
4 5 5 5 6		
6 6		

;

TABLE

UIS(W,N,P,T,S) SCENARIOS FOR
 UNCERTAINTY IN SIZE

	S1	S2
S3 S4 S5 S6 S7		
(W1*W4).N1.P1.(T1*T6)	30	60
90 30 60 90 30		
60 90		

;

SCALAR

OMEGA1 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGA2 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGA3 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGA4 UPPER BOUND FOR
 EXPANSION VARIABLE FOR WELL PLATFORM
 /5/
 OMEGA5 UPPER BOUND FOR
 EXPANSION VARIABLE FOR PRODUCTION
 PLATFORM /5/
 OMEGA6 UPPER BOUND FOR
 DRILLING RIG LOCATION /5/
 OMEGA7 UPPER BOUND FOR
 FLOW PROFILE CONSTRAINT /5/

;

ALIAS

(T, THETA, TD, T1, TDASH)

;

FREE VARIABLES

Z TOTAL
 PRODUCTION PROFIT FOR ALL SCENARIOS
 AND PROBABILITIES IN DOLLARS

;

BINARY VARIABLES

Z1(W,N,P,T,S) 1 IF WELL W
 CONNECTED TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IS DRILLED IN
 TIME T
 Z2(N,P,T,S) 1 IF WELL
 PLATFORM N CONNECTED TO PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 Z3(P,T,S) 1 IF PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 Z4(W,N,P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 Z5(N,P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 Z6(P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 YD(R,F,J,T,S) 1 IF PIECE J USED
 FOR LINEAR INTERPOLATION IN PERIOD T
 ZD(N,P,D,T,S) 1 IF DRILLING RIG
 K I LOCATED ON FACILITY IN PERIOD T
 ZF(N,P,D,T,S) 1 IF THE K-TH RIG
 IS LOCATED ON WELL PLATFORM N FIRST IN
 PERIOD T
 ZL(N,P,D,T,S) 1 IF THE K-TH RIG
 IS LOCATED ON WELL PLATFORM N LAST IN
 PERIOD T
 SL(P,D,T,S) 1 WHEN ONLY THE
 N-TH PLATFORM IS USED FOR DRILLING
 USING THE K-TH RIG IN PERIOD T
 ZT(N,P,D,T,S) 1 IF DRILLING RIG
 D IS MOVED AFTER PERIOD T

;

POSITIVE VARIABLES

Z7(P,D,T,S) NUMBER OF TIMES
 DRILLING RIG D IS MOVED IN PERIOD T
 X1(W,N,P,T,S) AMOUNT OF GAS OR
 OIL TO TRANSFER BETWEEN WELL W TO WELL
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME T IN KG
 X2(N,P,T,S) AMOUNT OF GAS OR
 OIL TO TRANSFER BETWEEN WELL PLATFORM

N TO PRODUCTION PLATFORM P IN TIME T
 IN KG
 X3(P,T,S) AMOUNT OF GAS OR
 OIL FLOWING AT PRODUCTION PLATFORM P
 IN TIME T IN KG
 X4(T,S) SUM OF ALL GAS OR
 OIL FLOWING AT PRODUCTION PLATFORM P
 IN TIME T IN KG
 V1(W,N,P,T,S) PRESSURE FOR FLOW
 BETWEEN WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN BAR
 V2(N,P,T,S) PRESSURE FOR FLOW
 BETWEEN WELL PLATFORM N TO PRODUCTION
 PLATFORM P IN TIME T IN BAR
 V3(P,T,S) PRESSURE FOR FLOW
 AT PRODUCTION PLATFORM P IN TIME T IN
 BAR
 GF1(W,N,P,T,S) GAS FLOW
 (VOLUMETRIC) BETWEEN WELL W TO WELL
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME T IN M3
 GF2(N,P,T,S) GAS FLOW
 (VOLUMETRIC) BETWEEN WELL PLATFORM N
 TO PRODUCTION PLATFORM P IN TIME T IN
 M3
 DELTA1(W,N,P,T,S) PRESSURE DROP AT
 CHOKE FOR FLOW BETWEEN WELL W TO WELL
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME T IN BAR
 DELTA2(N,P,T,S) PRESSURE DROP AT
 CHOKE FOR FLOW BETWEEN WELL PLATFORM N
 TO PRODUCTION PLATFORM P IN TIME T IN
 BAR
 LF(W,N,P,T,S) OIL FLOW FROM
 WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN KG
 GF(W,N,P,T,S) GAS FLOW FROM
 WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN M3
 X5(W,N,P,THETA,S) THE CUMULATIVE
 AMOUNT OF OIL TO TRANSFER BETWEEN WELL
 W TO WELL PLATFORM N TO PRODUCTION
 PLATFORM P IN TIME THETA IN KG
 X6(R,F,THETA,S) THE CUMULATIVE
 AMOUNT OF OIL FROM RESERVOIR R IN
 FIELD F IN TIME THETA IN KG
 V4(R,F,T,S) PRESSURE IN
 RESERVOIR R IN FIELD F AT TIME T IN
 BAR
 LAMBDA(R,F,J,T,S) INTERPOLATION
 VARIABLE AT PERIOD T
 X7(N,P,THETA,S) THE CUMULATIVE
 AMOUNT OF OIL TO TRANSFER BETWEEN WELL
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME THETA IN KG
 X8(P,THETA,S) THE CUMULATIVE
 AMOUNT OF OIL TO FLOWING FROM
 PRODUCTION PLATFORM P IN TIME THETA IN
 KG
 DV1(N,P,T,S) DESIGN VARIABLE
 FOR WELL PLATFORM
 DV2(P,T,S) DESIGN VARIABLE
 FOR PRODUCTION PLATFORM
 E1(N,P,T,S) DESIGN EXPANSION
 VARIABLE FOR WELL PLATFORM
 E2(P,T,S) DESIGN EXPANSION
 VARIABLE FOR PRODUCTION PLATFORM
 XD(W,N,P,T,S) DELIVERABILITY OF
 OIL FROM WELL
 ;
 EQUATIONS
 PROFIT OBJECTIVE
 FUNCTION IN MILLION DOLLAR
 XTOTAL1(N,P,T,S) SUM OF ALL GAS OR
 OIL RELATED TO WELL PLATFORM N AND
 PRODUCTION PLATFORM P IN TIME T IN KG
 XTOTAL2(P,T,S) SUM OF ALL GAS OR
 OIL RELATED TO PRODUCTION PLATFORM P
 IN TIME T IN KG
 XTOTAL3(T,S) SUM OF ALL GAS OR
 OIL IN TIME T IN KG
 VTOTAL1(W,N,P,T,S) SUM OF ALL
 PRESSURE RELATED TO WELL PLATFORM N1
 AND PRODUCTION PLATFORM P1 IN TIME T1
 IN BAR
 VTOTAL2(N,P,T,S) SUM OF ALL
 PRESSURE RELATED TO PRODUCTION
 PLATFORM P IN TIME T IN BAR
 XFLOW1(W,N,P,T,S) FLOW OF GAS OR
 OIL FROM WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN KG
 XFLOW2(W,N,P,T,S) FLOW OF OIL FROM
 WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN KG
 XFLOW3(W,N,P,T,S) FLOW OF GAS FROM
 WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN TIME T IN KG
 XCUM1(W,N,P,THETA,S) SUM OF THE AMOUNT
 OF OIL FROM ALL PERIODS UP TO TIME
 PERIOD THETA IN KG
 XCUM2(R,F,THETA,S) SUM OF THE AMOUNT
 OF OIL FROM ALL PERIODS UP TO TIME
 PERIOD THETA IN KG
 PIECE1(R,F,T,S) VALUE OF
 INTERPOLATED VARIABLE FOR EACH
 RESERVOIR R IN FIELD F IN TIME T
 PIECE2(R,F,T,S) CONSTRAINT FOR
 THE INTERPOLATION VARIABLE AT PERIOD T
 PIECE3(R,F,J,T,S) CONSTRAINT FOR
 GAMMA TO BE USED IN LINEAR
 INTERPOLATION
 PIECE4(R,F,T,S) CONSTRAINT FOR
 BINARY VARIABLE Y1=1 IF J USED FOR
 LINEAR INTERPOLATION IN PERIOD T
 INSTALL1(W,N,P,T,S) Z=1 IF WELL W
 CONNECTED TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IS DRILLED IN
 TIME T
 INSTALL2(N,P,T,S) Z=1 IF WELL
 PLATFORM N CONNECTED TO PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 INSTALL3(P,T,S) Z=1 IF PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 DUMMY1(W,N,P,TD,S) TO REDUCE THE
 NUMBER OF NODES ENUMERATED IN A BRANCH
 AND BOUND TREE
 DUMMY2(N,P,TD,S) TO REDUCE THE
 NUMBER OF NODES ENUMERATED IN A BRANCH
 AND BOUND TREE
 DUMMY3(P,TD,S) TO REDUCE THE
 NUMBER OF NODES ENUMERATED IN A BRANCH
 AND BOUND TREE
 XTOT1(W,N,P,THETA,S) THE FLOW OF OIL
 FROM WELL W TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IN PERIOD THETA
 IN KG
 XTOT2(N,P,THETA,S) THE FLOW OF OIL
 FROM WELL PLATFORM N TO PRODUCTION
 PLATFORM P IN PERIOD THETA IN KG
 XTOT3(P,THETA,S) THE FLOW OF OIL
 IN PRODUCTION PLATFORM P IN PERIOD
 THETA IN KG
 ZCON1(W,N,P,THETA,S) THE WELL PLATFORM
 ASSOCIATED WITH A WELL MUST BE
 INSTALLED BEFORE DRILLING THAT WELL
 ZCON2(N,P,THETA,S) THE PRODUCTION
 PLATFORMS MUST BE INSTALLED BEFORE
 ASSOCIATED WELL PLATFORMS

```

DES1(N,P,T,S)          THE FLOW FROM
WELL PLATFORM N TO PRODUCTION PLATFORM
P IN TIME T
DES2(N,P,T,S)          DESIGN VARIABLE
FOR WELL PLATFORM N TO PRODUCTION
PLATFORM P IN TIME T
DES3(N,P,T,S)          DESIGN EXPANSION
VARIABLE FOR WELL PLATFORM N TO
PRODUCTION PLATFORM P IN TIME T
DES4(P,T,S)            THE FLOW FROM
PRODUCTION PLATFORM P IN TIME T
DES5(P,T,S)            DESIGN VARIABLE
FOR PRODUCTION PLATFORM P IN TIME T
DES6(P,T,S)            DESIGN EXPANSION
VARIABLE FOR PRODUCTION PLATFORM P IN
TIME T
DRSC1(W,N,P,T,S)       IF ANY WELL
ASSOCIATED WITH WELL PLATFORM N IS
DRILLED IN PERIOD T _ SUMMATION OF Z (N
P K T) MUST BE 1 IN THAT TIME PERIOD
DRSC2(N,P,D,T,S)       THE K-TH RIG IS
LOCATED ON THE N-TH PLATFORM IN THAT
TIME PERIOD (SEE DRSC3)
DRSC3(P,D,T,S)         THE SLACK
VARIABLE TAKES A NON-ZERO VALUE IF ALL
WELLS ARE DRILLED FROM A SINGLE N-TH
PLATFORM IN THAT TIME PERIOD
DRSC4(N,P,D,T,S)       CONSTRAINT FOR
WELL DRILLED FROM WELL PLATFORM
DRSC5(N,P,D,T,S)       CONSTRAINT FOR
WELL DRILLED FROM WELL PLATFORM
DRSC6(N,P,D,T,S)       MOVEMENT OF RIG
CONSTRAINT
DRSC7(P,D,T,S)         MOVEMENT OF RIG
CONSTRAINT
FLOWPC(W,N,P,T,S)     FLOW PROFILE
CONSTRAINT
ENDO1(W,N,P,T,S)       ENDOGENEOUS
UNCERTAINTY IN SIZE AND INITIAL
DELIVERABILITY OF WELL
ENDO2(W,N,P,T,S)       OIL FLOW
CONSTRAINTS FOR ENDOGENEOUS
UNCERTAINTY
ENDO3(W,N,P,T,S)       CUMULATIVE OIL
CONSTRAINTS FOR ENDOGENEOUS
UNCERTAINTY
NAC1                   NON-
ANTICIPATIVITY CONSTRAINT
NAC2                   NON-
ANTICIPATIVITY CONSTRAINT
NAC3                   NON-
ANTICIPATIVITY CONSTRAINT
NAC4                   NON-
ANTICIPATIVITY CONSTRAINT
NAC5                   NON-
ANTICIPATIVITY CONSTRAINT
NAC6                   NON-
ANTICIPATIVITY CONSTRAINT
NAC7                   NON-
ANTICIPATIVITY CONSTRAINT
;
PROFIT..
Z =E=
SUM((S,SP,T),PROBA(S)*PROB(SP)*C1(T,SP
)*X4(T,S)) - SUM(S,
SUM(T,SUM(P,C2P(P,T)*Z3(P,T,S) +
C3P(P,T)*E2(P,T,S))) -
SUM(T,SUM(P,SUM(N$N1(N,P),(C2NP(N,P,T)
)*Z2(N,P,T,S) +
C3NP(N,P,T)*E1(N,P,T,S)))) -
SUM(T,SUM(P,SUM(N,SUM(W$W1(W,N),C2WNP(
W,N,P,T)*Z1(W,N,P,T,S)))) -
(SUM(T,SUM(P,SUM(D,SUM(N$N1(N,P),ZT(N,
P,D,T,S))))))));
*1.MASS BALANCE
XTOTAL1(N,P,T,S)$N1(N,P)..
X2(N,P,T,S) =E=
SUM(W$W1(W,N),X1(W,N,P,T,S))$N1(N,P);
XTOTAL2(P,T,S)..
SUM(N$N1(N,P),X2(N,P,T,S)) =E=
X3(P,T,S);
XTOTAL3(T,S)..
SUM(P,X3(P,T,S)) =E= X4(T,S);
*2.PRESSURE BALANCE
VTOTAL1(W,N,P,T,S)$ (N1(N,P)$W1(W,N))..
V1(W,N,P,T,S) - ALPHAN*X1(W,N,P,T,S) -
BETAN*GF1(W,N,P,T,S) -
DELTA1(W,N,P,T,S) =E=
V2(N,P,T,S)$ (W1(W,N)$ (N1(N,P)));
VTOTAL2(N,P,T,S)$N1(N,P)..
V2(N,P,T,S) - ALPHAP*X2(N,P,T,S) -
BETAP*GF2(N,P,T,S) - DELTA2(N,P,T,S)
=E= V3(P,T,S)$N1(N,P);
*3.FLOW CONSTRAINTS IN WELLS
XFLOW1(W,N,P,T,S)$ (N1(N,P)$W1(W,N))..
X1(W,N,P,T,S) =E= LF(W,N,P,T,S) +
GF(W,N,P,T,S)$ (W1(W,N)$ (N1(N,P)));
XFLOW2(W,N,P,T,S)$ (N1(N,P)$W1(W,N))..
LF(W,N,P,T,S) =L=
RO(W,N,P)*PMA$ (W1(W,N)$ (N1(N,P)));
XFLOW3(W,N,P,T,S)$ (N1(N,P)$W1(W,N))..
GF(W,N,P,T,S) =L=
LF(W,N,P,T,S)*GORMAX$ (W1(W,N)$ (N1(N,P)
));
*4.CUMULATIVE FLOW AMOUNT FROM WELLS
UP TO PERIOD T
XCUM1(W,N,P,THETA,S)$ (N1(N,P)$W1(W,N))
..
X5(W,N,P,THETA,S) =E=
SUM(T$(ORD(T) LE (ORD(THETA)-1)),
X1(W,N,P,T,S)*DELTAT);
XCUM2(R,F,THETA,S)$R1(R,F)..
X6(R,F,THETA,S) =E=
SUM((W,N,P)$ (W1(W,N)$ (N1(N,P)$ (W4(W,R,
F)))) ,X5(W,N,P,THETA,S));
*5.PIECEWISE LINEAR INTERPOLATION AT
WELL AND RESERVOIR LEVEL
PIECE1(R,F,T,S)..
SUM(J$J1(J,R),LAMBDA(R,F,J,T,S)*V5(R,F
,J,T)) =E= V4(R,F,T,S)$R1(R,F);
PIECE2(R,F,T,S)..
SUM(J$J1(J,R),LAMBDA(R,F,J,T,S)) =E=
1$R1(R,F);
PIECE3(R,F,J,T,S)..
LAMBDA(R,F,J,T,S) =L= YD(R,F,J,T,S) +
YD(R,F,J-1,T,S)$R1(R,F);
PIECE4(R,F,T,S)..
SUM(J,YD(R,F,J,T,S)) =E= 1$R1(R,F);
*6.LOGICAL CONSTRAINTS FOR
INSTALLATION AND FLOW FROM FACILITIES
INSTALL1(W,N,P,T,S)$ (W1(W,N)$ (N1(N,P))
)..
Z1(W,N,P,T,S) =L= 1;
INSTALL2(N,P,T,S)..
Z2(N,P,T,S) =L= 1$N1(N,P);
INSTALL3(P,T,S)..
Z3(P,T,S) =L= 1;
DUMMY1(W,N,P,TD,S)..
SUM(T,Z1(W,N,P,T,S)) + Z4(W,N,P,TD,S)
=E= 1$ (N1(N,P)$W1(W,N));

```

```

DUMMY2 (N,P,TD,S) ..
SUM(T,Z2(N,P,T,S)) + Z5(N,P,TD,S) =E=
1;
DUMMY3 (P,TD,S) ..
SUM(T,Z3(P,T,S)) + Z6(P,TD,S) =E= 1;
XTOT1(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
X1(W,N,P,T,S) =L=
OMEGAU1*Z1(W,N,P,T,S);
XTOT2(N,P,THETA,S)$N1(N,P)..
X2(N,P,THETA,S) =L=
OMEGAU2*SUM(T$(ORD(T)LE
ORD(THETA))),Z2(N,P,T,S));
XTOT3(P,THETA,S)..
X3(P,THETA,S) =L=
OMEGAU3*SUM(T$(ORD(T)LE
ORD(THETA))),Z3(P,T,S));
ZCON1(W,N,P,THETA,S)$(N1(N,P)$W1(W,N))
..
Z1(W,N,P,THETA,S) =L=
SUM(T$(ORD(T)LE
ORD(THETA))),Z2(N,P,T,S));
ZCON2(N,P,THETA,S)$N1(N,P)..
Z2(N,P,THETA,S) =L= SUM(T$(ORD(T)LE
ORD(THETA))),Z3(P,T,S));

*7.DESIGN OF FACILITY

DES1(N,P,T,S)$N1(N,P)..
X2(N,P,T,S) =L= DV1(N,P,T,S);
DES2(N,P,T,S)$N1(N,P)..
DV1(N,P,T,S) =E= DV1(N,P,T-1,S) +
E1(N,P,T,S);
DES3(N,P,T,S)$N1(N,P)..
E1(N,P,T,S) =L= OMEGAU4*Z2(N,P,T,S);
DES4(P,T,S)..
X3(P,T,S) =L= DV2(P,T,S);
DES5(P,T,S)..
DV2(P,T,S) =E= DV2(P,T-1,S) +
E2(P,T,S);
DES6(P,T,S)..
E2(P,T,S) =L= OMEGAU5*Z3(P,T,S);

*8.NUMBER OF WELLS DRILLED IN TIME
PERIOD T

DRSC1(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
Z1(W,N,P,T,S) =L=
SUM(D,ZD(N,P,D,T,S));
DRSC2(N,P,D,T,S)$N1(N,P)..
ZF(N,P,D,T,S) + ZL(N,P,D,T,S) =L= 1 +
SL(P,D,T,S);
DRSC3(P,D,T,S)..
SUM(N$(N1(N,P)),ZD(N,P,D,T,S)) - 1 =L=
OMEGAU6*(1 - SL(P,D,T,S));
DRSC4(N,P,D,T,S)$N1(N,P)..
ZF(N,P,D,T,S) =L= ZD(N,P,D,T,S);
DRSC5(N,P,D,T,S)$N1(N,P)..
ZL(N,P,D,T,S) =L= ZD(N,P,D,T,S);
DRSC6(N,P,D,T,S)$N1(N,P)..
ZF(N,P,D,T,S) - ZL(N,P,D,T-1,S) =L=
ZT(N,P,D,T,S);
DRSC7(P,D,T,S)..
SUM(N$(N1(N,P)),ZD(N,P,D,T,S)) - 1
=L= Z7(P,D,T,S);

*9.DRILLING RIG SCHEDULING CONSTRAINTS

*10.FLOW PROFILE CONSTRAINTS

FLOWPC(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
X1(W,N,P,T,S) =G= X1(W,N,P,T+1,S) -
OMEGAU7*(1 - SUM(TDASH$(ORD(TDASH)LE
ORD(T)),Z1(W,N,P,TDASH,S)));

*11.ENDOGENEOUS UNCERTAINTY IN SIZE
AND INITIAL DELIVERABILITY OF WELL

END01(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
(XD(W,N,P,T,S)/UID(W,N,P,T,S)) +
(X5(W,N,P,T,S)/UIS(W,N,P,T,S)) =E= 1;
END02(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
X1(W,N,P,T,S) =L= XD(W,N,P,T,S);
END03(W,N,P,T,S)$(N1(N,P)$W1(W,N))..
X5(W,N,P,T,S) =L= UIS(W,N,P,T,S);

*12.NONANTICIPATIVITY CONSTRAINTS

NAC1(W,N,P,T,S)$((ORD(S)NE
CARD(S))$(N1(N,P)$W1(W,N)))..
Z1(W,N,P,T,S) =E= Z1(W,N,P,T,S+1);
NAC2(N,P,T,S)$((ORD(S)LT
9)$N1(N,P))..
Z2(N,P,T,S) =E= Z2(N,P,T,S+1);
NAC3(P,T,S)$((ORD(S)LT
9))..
Z3(P,T,S) =E= Z3(P,T,S+1);
NAC4(N,P,D,T,S)$((ORD(S)LT
9)$N1(N,P))..
ZD(N,P,D,T,S) =E= ZD(N,P,D,T,S+1);
NAC5(N,P,D,T,S)$((ORD(S)LT
9))..
ZT(N,P,D,T,S) =E= ZT(N,P,D,T,S+1);
NAC6(N,P,T,S)$((ORD(S)LT
9)$N1(N,P))..
E1(N,P,T,S) =E= E1(N,P,T,S+1);
NAC7(P,T,S)$((ORD(S)LT
9))..
E2(P,T,S) =E= E2(P,T,S+1);

*VARIABLE BOUNDS

X1.UP(W,N,P,T,S) = 1000;

MODEL OILPLAN /ALL/
;

OPTION LIMROW = 100;
OPTION LIMCOL = 100;
OPTION OPTCR = 0.0;

SOLVE OILPLAN USING MIP MAXIMIZING Z;

```


APPENDIX III: GAMS OUTPUT
FILE ON DETERMINISTIC
MODEL

GAMS Rev 149 x86/MS Windows
 04/27/07 17:03:40 Page 6
 : OIL PRODUCTION PLANNING
 (DETERMINISTIC MODEL)
 Execution

----- 286 VARIABLE Z.L
 = 331.080 TOTAL PROFIT IN MILLI

ONS OF DOLLARS

----- 286 VARIABLE Z1.L 1 IF WELL W
 CONNECTED TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IS DRILLED IN
 TIME T

INDEX 1 = W1

T6
 N1.P1 1.000

INDEX 1 = W2

T1
 N1.P1 1.000

INDEX 1 = W3

T3
 N1.P1 1.000

INDEX 1 = W4

T2
 N1.P1 1.000

----- 286 VARIABLE Z2.L 1 IF WELL
 PLATFORM N CONNECTED TO PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T

T1
 N1.P1 1.000

----- 286 VARIABLE Z3.L 1 IF
 PRODUCTION PLATFORM P IS INSTALLED IN
 TIME T

T1
 P1 1.000

----- 286 VARIABLE X1.L AMOUNT OF
 GAS OR OIL TO TRANSFER BETWEEN WELL W
 TO WELL PLATFORM N TO PRODUCTION
 PLATFORM P IN TIME T IN KG

INDEX 1 = W1

T6

N1.P1 5.000

INDEX 1 = W2

T1

N1.P1 5.000

INDEX 1 = W3

T3

N1.P1 5.000

INDEX 1 = W4

T2

N1.P1 5.000

----- 286 VARIABLE X2.L AMOUNT OF
 GAS OR OIL TO TRANSFER BETWEEN WELL,
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME T IN KG

T1 T2
 T3 T6

N1.P1 5.000 5.000
 5.000 5.000

----- 286 VARIABLE X3.L AMOUNT OF
 GAS OR OIL FLOWING AT PRODUCTION
 PLATFORM PIN TIME T IN KG

T1 T2 T3
 T6

P1 5.000 5.000 5.000
 5.000

----- 286 VARIABLE zaa.L
 = 440.000

**** REPORT FILE SUMMARY

RESPLAN C:\Documents and
 Settings\Kippi\My
 Documents\gamsdir\projdir\UPSTREAM.re
 s

EXECUTION TIME = 0.063
 SECONDS 3 Mb WIN226-149 Dec 19,
 2007

USER: CS/IE 635, Spring 2007
 G061206/0001AS-WIN
 Prof. Ferris
 DC2937

License for teaching and
 research at degree granting
 institutions \$TITLE: OIL PRODUCTION
 PLANNING (STOCHASTIC MODEL)

FILE RESPLAN2 / UPSTREAM2.res /

PUT RESPLAN2;

**APPENDIX IV: COMPUTER
CODE FOR GAMS INPUT FILE
ON STOCHASTIC MODEL**

```

SETS

T          SET OF TIME /T1*T6/
W          SET OF WELLS /W1*W4/
N          SET OF WELL PLATFORMS
/N1/
P          SET OF PRODUCTION
PLATFORMS /P1/
R          SET OF RESERVOIRS
/R1*R2/
F          SET OF FIELDS /F1/
J          SET OF INDEX OF PIECE
IN PIECEWISE LINEAR INTERPOLATION /J1/
D          SET OF DRILLING RIGS
/D1/
N1(N,P)   SET OF WELL PLATFORMS
N ASSOCIATED WITH PLATFORM P /N1.P1/
W1(W,N)   SET OF WELLS W
ASSOCIATED WITH WELL PLATFORM N
/(W1*W4).N1/
W2(W,R)   SET OF WELLS W
ASSOCIATED WITH RESERVOIR R
/W1.R1,W2.R1,W3.R2,W4.R2/
W3(W,R,N) SET OF WELLS W
ASSOCIATED WITH RESERVOIR R AND WELL
PLATFORM N
/(W1,W2).R1.N1,(W3,W4).R2.N1/
W4(W,R,F) SET OF WELLS W IN
FIELD F ASSOCIATED WITH RESERVOIR R
/W1.R1.F1,W2.R1.F1,W3.R2.F1,W4.R2.F1/
R1(R,F)   SET OF RESERVOIRS
ASSOCIATED WITH FIELD F /R1.F1/
J1(J,R)   SET OF LINEAR
INTERPOLATION PIECES USED FOR
RESERVOIR R /J1.R1/
SP        SET OF SCENARIOS FOR
PRICE UNCERTAINTY /L,M,H/
S         SET OF SCENARIOS FOR
UNCERTAINTY IN SIZE AND INITIAL
DELIVERABILITY OF WELL /S1*S9/
;

SCALARS

ALPHAN    PRESSURE DROP
COEFFICIENT FOR OIL FLOW RATE FOR WELL
PLATFORM /20/
ALPHAP    PRESSURE DROP
COEFFICIENT FOR OIL FLOW RATE FOR
PRODUCTION PLATFORM /20/
BETAN     PRESSURE DROP
COEFFICIENT FOR GOR FOR PRODUCTION
PLATFORM /18/
BETAP     PRESSURE DROP
COEFFICIENT FOR GOR FOR PRODUCTION
PLATFORM /18/
PMAX      MAXIMUM PRESSURE DROP
FROM WELL BORE TO WELL HEAD /8/
GORMAX    MAXIMUM GAS-TO-OIL
RATIO /11/

DELTAT    LENGTH OF TIME PERIOD
T (YEAR) /4/
ITERATION

```

```

;
ITERATION = 1;

PARAMETERS

RO(W,N,P)      PRODUCTIVITY INDEX
OF WELL W CONNECTED TO WELL PLATFORM N
CONNECTED TO PRODUCTION PLATFORM P IN
TIME T DIMENSIONLESS
/
W1.N1.P1      47.6
W2.N1.P1      83.8
W3.N1.P1      158.4
W4.N1.P1      210.2
/

C1(T,SP)       DISCOUNTED REVENUE
PRICE COEFFICIENT FOR OIL SALES
/
(T1*T6).L     19
(T1*T6).M     22
(T1*T6).H     25
/

C2P(P,T)       DISCOUNTED FIXED COST
COEFFICIENT FOR INSTALLATION OF
PRODUCTION PLATFORM IN MILLIONS OF
DOLLARS
/
P1.T1 70
P1.T2 69
P1.T3 68
P1.T4 67
P1.T5 66
P1.T6 65
/

C2NP(N,P,T)    DISCOUNTED FIXED COST
COEFFICIENT FOR INSTALLATION OF WELL
PLATFORM IN MILLIONS OF DOLLARS
/
N1.P1.T1 11
N1.P1.T2 10
N1.P1.T3 9
N1.P1.T4 8
N1.P1.T5 7
N1.P1.T6 6
/

C2WNP(W,N,P,T) DISCOUNTED FIXED COST
COEFFICIENT FOR DRILLING THE WELL IN
MILLIONS OF DOLLARS
/
W1.N1.P1.(T1*T6) 6.69
W2.N1.P1.(T1*T6) 6.34
W3.N1.P1.(T1*T6) 5.76
W4.N1.P1.(T1*T6) 5.83
/

C3P(P,T)       DISCOUNTED VARIABLE
COST COEFFICIENT FOR INSTALLATION OF
PRODUCTION PLATFORM IN MILLIONS OF
DOLLARS
/
P1.T1 0.47
P1.T2 0.46
P1.T3 0.45
P1.T4 0.44
P1.T5 0.43
P1.T6 0.42
/

C3NP(N,P,T)    DISCOUNTED VARIABLE
COST COEFFICIENT FOR INSTALLATION OF
WELL PLATFORM IN MILLIONS OF DOLLARS
/
N1.P1.T1 0.19
N1.P1.T2 0.18

```

N1.P1.T3 0.17
 N1.P1.T4 0.16
 N1.P1.T5 0.15
 N1.P1.T6 0.14
 /
 C4(T) DISCOUNTED COST
 COEFFICIENT FOR MOVING RIGS IN
 MILLIONS OF DOLLARS
 /
 T1 50
 T2 49
 T3 48
 T4 47
 T5 46
 T6 45
 /
 V5(R,F,J,T) INDEX OF PIECE J USED
 TO CALCULATE PRESSURE IN RESERVOIR R
 IN FIELD F AT TIME T IN BAR
 (PRESPECIFIED FROM RESERVOIR
 PERFORMANCE CURVE)
 /
 R1.F1.J1.(T1*T6) 10
 /

PROB(SP)
 /
 L 0.3
 M 0.4
 H 0.3
 /

PROBA(S)
 /
 S1 0.09
 S2 0.12
 S3 0.09
 S4 0.12
 S5 0.16
 S6 0.12
 S7 0.09
 S8 0.12
 S9 0.09
 /

TABLE

UID(W,N,P,T,S) SCENARIOS FOR
 UNCERTAINTY IN INITIAL
 DELIVERABILITIES

			S1	S2
S3	S4	S5	S6	S7
S8	S9			
(W1*W4)	N1.P1.(T1*T6)		4	4
4	5	5	5	6
6	6			

TABLE

UIS(W,N,P,T,S) SCENARIOS FOR
 UNCERTAINTY IN SIZE

			S1	S2
S3	S4	S5	S6	S7
S8	S9			
(W1*W4)	N1.P1.(T1*T6)		30	60
90	30	60	90	30
60	90			

SCALAR

OMEGAU1 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGAU2 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGAU3 UPPER BOUND FOR OIL
 FLOW RATE IN MILLION BARREL /5/
 OMEGAU4 UPPER BOUND FOR
 EXPANSION VARIABLE FOR WELL PLATFORM
 /5/
 OMEGAU5 UPPER BOUND FOR
 EXPANSION VARIABLE FOR PRODUCTION
 PLATFORM /5/
 OMEGAU6 UPPER BOUND FOR
 DRILLING RIG LOCATION /5/
 OMEGAU7 UPPER BOUND FOR
 FLOW PROFILE CONSTRAINT /5/
 ;

ALIAS

(T, THETA, TD, T1, TDASH)
 ;

FREE VARIABLES

Z TOTAL
 PRODUCTION PROFIT FOR ALL SCENARIOS
 AND PROBABILITIES IN DOLLARS
 ;

BINARY VARIABLES

Z1(W,N,P,T,S) 1 IF WELL W
 CONNECTED TO WELL PLATFORM N TO
 PRODUCTION PLATFORM P IS DRILLED IN
 TIME T
 Z2(N,P,T,S) 1 IF WELL
 PLATFORM N CONNECTED TO PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 Z3(P,T,S) 1 IF PRODUCTION
 PLATFORM P IS INSTALLED IN TIME T
 Z4(W,N,P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 Z5(N,P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 Z6(P,TD,S) DUMMY PERIOD
 VALUES AT PERIOD T
 YD(R,F,J,T,S) 1 IF PIECE J USED
 FOR LINEAR INTERPOLATION IN PERIOD T
 ZD(N,P,D,T,S) 1 IF DRILLING RIG
 K I LOCATED ON FACILITY IN PERIOD T
 ZF(N,P,D,T,S) 1 IF THE K-TH RIG
 IS LOCATED ON WELL PLATFORM N FIRST IN
 PERIOD T
 ZL(N,P,D,T,S) 1 IF THE K-TH RIG
 IS LOCATED ON WELL PLATFORM N LAST IN
 PERIOD T
 SL(P,D,T,S) 1 WHEN ONLY THE
 N-TH PLATFORM IS USED FOR DRILLING
 USING THE K-TH RIG IN PERIOD T
 ZT(N,P,D,T,S) 1 IF DRILLING RIG
 D IS MOVED AFTER PERIOD T
 ;

POSITIVE VARIABLES

Z7(P,D,T,S) NUMBER OF TIMES
 DRILLING RIG D IS MOVED IN PERIOD T
 X1(W,N,P,T,S) AMOUNT OF GAS OR
 OIL TO TRANSFER BETWEEN WELL W TO WELL
 PLATFORM N TO PRODUCTION PLATFORM P IN
 TIME T IN KG
 X2(N,P,T,S) AMOUNT OF GAS OR
 OIL TO TRANSFER BETWEEN WELL PLATFORM
 N TO PRODUCTION PLATFORM P IN TIME T
 IN KG

X3(P,T,S)	AMOUNT OF GAS OR OIL FLOWING AT PRODUCTION PLATFORM P IN TIME T IN KG	XTOTAL2(P,T,S)	SUM OF ALL GAS OR OIL RELATED TO PRODUCTION PLATFORM P IN TIME T IN KG
X4(T,S)	SUM OF ALL GAS OR OIL FLOWING AT PRODUCTION PLATFORM P IN TIME T IN KG	XTOTAL3(T,S)	SUM OF ALL GAS OR OIL IN TIME T IN KG
V1(W,N,P,T,S)	PRESSURE FOR FLOW BETWEEN WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN BAR	VTOTAL1(W,N,P,T,S)	SUM OF ALL PRESSURE RELATED TO WELL PLATFORM N1 AND PRODUCTION PLATFORM P1 IN TIME T1 IN BAR
V2(N,P,T,S)	PRESSURE FOR FLOW BETWEEN WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN BAR	VTOTAL2(N,P,T,S)	SUM OF ALL PRESSURE RELATED TO PRODUCTION PLATFORM P IN TIME T IN BAR
V3(P,T,S)	PRESSURE FOR FLOW AT PRODUCTION PLATFORM P IN TIME T IN BAR	XFLOW1(W,N,P,T,S)	FLOW OF GAS OR OIL FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN KG
GF1(W,N,P,T,S)	GAS FLOW (VOLUMETRIC) BETWEEN WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN M3	XFLOW2(W,N,P,T,S)	FLOW OF OIL FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN KG
GF2(N,P,T,S)	GAS FLOW (VOLUMETRIC) BETWEEN WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN M3	XFLOW3(W,N,P,T,S)	FLOW OF GAS FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN KG
DELTA1(W,N,P,T,S)	PRESSURE DROP AT CHOKE FOR FLOW BETWEEN WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN BAR	XCUM1(W,N,P,THETA,S)	SUM OF THE AMOUNT OF OIL FROM ALL PERIODS UP TO TIME PERIOD THETA IN KG
DELTA2(N,P,T,S)	PRESSURE DROP AT CHOKE FOR FLOW BETWEEN WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN BAR	XCUM2(R,F,THETA,S)	SUM OF THE AMOUNT OF OIL FROM ALL PERIODS UP TO TIME PERIOD THETA IN KG
LF(W,N,P,T,S)	OIL FLOW FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN KG	PIECE1(R,F,T,S)	VALUE OF INTERPOLATED VARIABLE FOR EACH RESERVOIR R IN FIELD F IN TIME T
GF(W,N,P,T,S)	GAS FLOW FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN M3	PIECE2(R,F,T,S)	CONSTRAINT FOR THE INTERPOLATION VARIABLE AT PERIOD T
X5(W,N,P,THETA,S)	THE CUMULATIVE AMOUNT OF OIL TO TRANSFER BETWEEN WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME THETA IN KG	PIECE3(R,F,J,T,S)	CONSTRAINT FOR GAMMA TO BE USED IN LINEAR INTERPOLATION
X6(R,F,THETA,S)	THE CUMULATIVE AMOUNT OF OIL FROM RESERVOIR R IN FIELD F IN TIME THETA IN KG	PIECE4(R,F,T,S)	CONSTRAINT FOR BINARY VARIABLE Y1=1 IF J USED FOR LINEAR INTERPOLATION IN PERIOD T
V4(R,F,T,S)	PRESSURE IN RESERVOIR R IN FIELD F AT TIME T IN BAR	INSTALL1(W,N,P,T,S)	Z=1 IF WELL W CONNECTED TO WELL PLATFORM N TO PRODUCTION PLATFORM P IS DRILLED IN TIME T
LAMBDA(R,F,J,T,S)	INTERPOLATION VARIABLE AT PERIOD T	INSTALL2(N,P,T,S)	Z=1 IF WELL PLATFORM N CONNECTED TO PRODUCTION PLATFORM P IS INSTALLED IN TIME T
X7(N,P,THETA,S)	THE CUMULATIVE AMOUNT OF OIL TO TRANSFER BETWEEN WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME THETA IN KG	INSTALL3(P,T,S)	Z=1 IF PRODUCTION PLATFORM P IS INSTALLED IN TIME T
X8(P,THETA,S)	THE CUMULATIVE AMOUNT OF OIL TO FLOWING FROM PRODUCTION PLATFORM P IN TIME THETA IN KG	DUMMY1(W,N,P,TD,S)	TO REDUCE THE NUMBER OF NODES ENUMERATED IN A BRANCH AND BOUND TREE
DV1(N,P,T,S)	DESIGN VARIABLE FOR WELL PLATFORM	DUMMY2(N,P,TD,S)	TO REDUCE THE NUMBER OF NODES ENUMERATED IN A BRANCH AND BOUND TREE
DV2(P,T,S)	DESIGN VARIABLE FOR PRODUCTION PLATFORM	DUMMY3(P,TD,S)	TO REDUCE THE NUMBER OF NODES ENUMERATED IN A BRANCH AND BOUND TREE
E1(N,P,T,S)	DESIGN EXPANSION VARIABLE FOR WELL PLATFORM	XTOT1(W,N,P,THETA,S)	THE FLOW OF OIL FROM WELL W TO WELL PLATFORM N TO PRODUCTION PLATFORM P IN PERIOD THETA IN KG
E2(P,T,S)	DESIGN EXPANSION VARIABLE FOR PRODUCTION PLATFORM	XTOT2(N,P,THETA,S)	THE FLOW OF OIL FROM WELL PLATFORM N TO PRODUCTION PLATFORM P IN PERIOD THETA IN KG
XD(W,N,P,T,S)	DELIVERABILITY OF OIL FROM WELL	XTOT3(P,THETA,S)	THE FLOW OF OIL IN PRODUCTION PLATFORM P IN PERIOD THETA IN KG
		ZCON1(W,N,P,THETA,S)	THE WELL PLATFORM ASSOCIATED WITH A WELL MUST BE INSTALLED BEFORE DRILLING THAT WELL
		ZCON2(N,P,THETA,S)	THE PRODUCTION PLATFORMS MUST BE INSTALLED BEFORE ASSOCIATED WELL PLATFORMS
		DES1(N,P,T,S)	THE FLOW FROM WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T

;

EQUATIONS

PROFIT	OBJECTIVE FUNCTION IN MILLION DOLLAR
XTOTAL1(N,P,T,S)	SUM OF ALL GAS OR OIL RELATED TO WELL PLATFORM N AND PRODUCTION PLATFORM P IN TIME T IN KG


```

DES2(N,P,T,S)      DESIGN VARIABLE
FOR WELL PLATFORM N TO PRODUCTION
PLATFORM P IN TIME T
DES3(N,P,T,S)      DESIGN EXPANSION
VARIABLE FOR WELL PLATFORM N TO
PRODUCTION PLATFORM P IN TIME T
DES4(P,T,S)        THE FLOW FROM
PRODUCTION PLATFORM P IN TIME T
DES5(P,T,S)        DESIGN VARIABLE
FOR PRODUCTION PLATFORM P IN TIME T
DES6(P,T,S)        DESIGN EXPANSION
VARIABLE FOR PRODUCTION PLATFORM P IN
TIME T
DRSC1(W,N,P,T,S)  IF ANY WELL
ASSOCIATED WITH WELL PLATFORM N IS
DRILLED IN PERIOD T - SUMMATION OF Z(N
P K T) MUST BE 1 IN THAT TIME PERIOD
DRSC2(N,P,D,T,S)  THE K-TH RIG IS
LOCATED ON THE N-TH PLATFORM IN THAT
TIME PERIOD (SEE DRSC3)
DRSC3(P,D,T,S)    THE SLACK
VARIABLE TAKES A NON-ZERO VALUE IF ALL
WELLS ARE DRILLED FROM A SINGLE N-TH
PLATFORM IN THAT TIME PERIOD
DRSC4(N,P,D,T,S)  CONSTRAINT FOR
WELL DRILLED FROM WELL PLATFORM
DRSC5(N,P,D,T,S)  CONSTRAINT FOR
WELL DRILLED FROM WELL PLATFORM
DRSC6(N,P,D,T,S)  MOVEMENT OF RIG
CONSTRAINT
DRSC7(P,D,T,S)    MOVEMENT OF RIG
CONSTRAINT
FLOWPC(W,N,P,T,S) FLOW PROFILE
CONSTRAINT
ENDO1(W,N,P,T,S) ENDOGENEOUS
UNCERTAINTY IN SIZE AND INITIAL
DELIVERABILITY OF WELL
ENDO2(W,N,P,T,S) OIL FLOW
CONSTRAINTS FOR ENDOGENEOUS
UNCERTAINTY
ENDO3(W,N,P,T,S) CUMULATIVE OIL
CONSTRAINTS FOR ENDOGENEOUS
UNCERTAINTY
NAC1              NON-
ANTICIPATIVITY CONSTRAINT
NAC2              NON-
ANTICIPATIVITY CONSTRAINT
NAC3              NON-
ANTICIPATIVITY CONSTRAINT
NAC4              NON-
ANTICIPATIVITY CONSTRAINT
NAC5              NON-
ANTICIPATIVITY CONSTRAINT
NAC6              NON-
ANTICIPATIVITY CONSTRAINT
NAC7              NON-
ANTICIPATIVITY CONSTRAINT
;

PROFIT..
Z =E=
SUM((S,SP,T),PROBA(S)*PROB(SP)*C1(T,SP
)*X4(T,S)) - SUM(S,
SUM(T,SUM(P,C2P(P,T)*Z3(P,T,S) +
C3P(P,T)*E2(P,T,S))) -
SUM(T,SUM(P,SUM(N$N1(N,P),(C2NP(N,P,T)
)*Z2(N,P,T,S) +
C3NP(N,P,T)*E1(N,P,T,S)))))) -
SUM(T,SUM(P,SUM(N,SUM(W$W1(W,N),C2WNP(W,
N,P,T)*Z1(W,N,P,T,S)))))) -
(SUM(T,SUM(P,SUM(D,SUM(N$N1(N,P),ZT(N,
P,D,T,S))))))));

*1.MASS BALANCE

XTOTAL1(N,P,T,S)$N1(N,P)..
X2(N,P,T,S) =E=
SUM(W$W1(W,N),X1(W,N,P,T,S))$N1(N,P);
XTOTAL2(P,T,S)..
SUM(N$N1(N,P),X2(N,P,T,S)) =E=
X3(P,T,S);
XTOTAL3(T,S)..
SUM(P,X3(P,T,S)) =E= X4(T,S);

*2.PRESSURE BALANCE

VTOTAL1(W,N,P,T,S)$N1(N,P)$W1(W,N)..
V1(W,N,P,T,S) - ALPHAN*X1(W,N,P,T,S) -
BETAN*GF1(W,N,P,T,S) -
DELTA1(W,N,P,T,S) =E=
V2(N,P,T,S)$W1(W,N)$N1(N,P));
VTOTAL2(N,P,T,S)$N1(N,P)..
V2(N,P,T,S) - ALPHAP*X2(N,P,T,S) -
BETAP*GF2(N,P,T,S) - DELTA2(N,P,T,S)
=E= V3(P,T,S)$N1(N,P);

*3.FLOW CONSTRAINTS IN WELLS

XFLOW1(W,N,P,T,S)$N1(N,P)$W1(W,N)..
X1(W,N,P,T,S) =E= LF(W,N,P,T,S) +
GF(W,N,P,T,S)$W1(W,N)$N1(N,P));
XFLOW2(W,N,P,T,S)$N1(N,P)$W1(W,N)..
LF(W,N,P,T,S) =L=
RO(W,N,P)*PMA$W1(W,N)$N1(N,P));
XFLOW3(W,N,P,T,S)$N1(N,P)$W1(W,N)..
GF(W,N,P,T,S) =L=
LF(W,N,P,T,S)*GORMA$W1(W,N)$N1(N,P)
));

*4.CUMULATIVE FLOW AMOUNT FROM WELLS
UP TO PERIOD T

XCUM1(W,N,P,THETA,S)$N1(N,P)$W1(W,N)
..
X5(W,N,P,THETA,S) =E=
SUM(T$(ORD(T) LE (ORD(THETA)-1)),
X1(W,N,P,T,S)*DELTAT);
XCUM2(R,F,THETA,S)$R1(R,F)..
X6(R,F,THETA,S) =E=
SUM((W,N,P)$W1(W,N)$N1(N,P)$W4(W,R,
F))),X5(W,N,P,THETA,S));

*5.PIECEWISE LINEAR INTERPOLATION AT
WELL AND RESERVOIR LEVEL

PIECE1(R,F,T,S)..
SUM(J$J1(J,R),LAMBDA(R,F,J,T,S)*V5(R,F
,J,T)) =E= V4(R,F,T,S)$R1(R,F);
PIECE2(R,F,T,S)..
SUM(J$J1(J,R),LAMBDA(R,F,J,T,S)) =E=
1$R1(R,F);
PIECE3(R,F,J,T,S)..
LAMBDA(R,F,J,T,S) =L= YD(R,F,J,T,S) +
YD(R,F,J-1,T,S)$R1(R,F);
PIECE4(R,F,T,S)..
SUM(J,YD(R,F,J,T,S)) =E= 1$R1(R,F);

*6.LOGICAL CONSTRAINTS FOR
INSTALLATION AND FLOW FROM FACILITIES

INSTALL1(W,N,P,T,S)$W1(W,N)$N1(N,P)
)..
Z1(W,N,P,T,S) =L= 1;
INSTALL2(N,P,T,S)..
Z2(N,P,T,S) =L= 1$N1(N,P);
INSTALL3(P,T,S)..
Z3(P,T,S) =L= 1;
DUMMY1(W,N,P,TD,S)..
SUM(T,Z1(W,N,P,T,S)) + Z4(W,N,P,TD,S)
=E= 1$N1(N,P)$W1(W,N);
DUMMY2(N,P,TD,S)..
SUM(T,Z2(N,P,T,S)) + Z5(N,P,TD,S) =E=
1;

```

```

DUMMY3 (P, TD, S) ..
SUM(T, Z3 (P, T, S)) + Z6 (P, TD, S) =E= 1;
XTOT1 (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
X1 (W, N, P, T, S) =L=
OMEGAU1 * Z1 (W, N, P, T, S);
XTOT2 (N, P, THETA, S) $N1 (N, P) ..
X2 (N, P, THETA, S) =L=
OMEGAU2 * SUM(T $ ((ORD (T) LE
ORD (THETA))) , Z2 (N, P, T, S));
XTOT3 (P, THETA, S) ..
X3 (P, THETA, S) =L=
OMEGAU3 * SUM(T $ ((ORD (T) LE
ORD (THETA))) , Z3 (P, T, S));
ZCON1 (W, N, P, THETA, S) $ (N1 (N, P) $W1 (W, N))
.. Z1 (W, N, P, THETA, S) =L=
SUM(T $ (ORD (T) LE
ORD (THETA))) , Z2 (N, P, T, S));
ZCON2 (N, P, THETA, S) $N1 (N, P) ..
Z2 (N, P, THETA, S) =L= SUM(T $ (ORD (T) LE
ORD (THETA))) , Z3 (P, T, S));

*7. DESIGN OF FACILITY

DES1 (N, P, T, S) $N1 (N, P) ..
X2 (N, P, T, S) =L= DV1 (N, P, T, S);
DES2 (N, P, T, S) $N1 (N, P) ..
DV1 (N, P, T, S) =E= DV1 (N, P, T-1, S) +
E1 (N, P, T, S);
DES3 (N, P, T, S) $N1 (N, P) ..
E1 (N, P, T, S) =L= OMEGAU4 * Z2 (N, P, T, S);
DES4 (P, T, S) ..
X3 (P, T, S) =L= DV2 (P, T, S);
DES5 (P, T, S) ..
DV2 (P, T, S) =E= DV2 (P, T-1, S) +
E2 (P, T, S);
DES6 (P, T, S) ..
E2 (P, T, S) =L= OMEGAU5 * Z3 (P, T, S);

*8. NUMBER OF WELLS DRILLED IN TIME
PERIOD T

DRSC1 (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
Z1 (W, N, P, T, S) =L=
SUM(D, ZD (N, P, D, T, S));
DRSC2 (N, P, D, T, S) $N1 (N, P) ..
ZF (N, P, D, T, S) + ZL (N, P, D, T, S) =L= 1 +
SL (P, D, T, S);
DRSC3 (P, D, T, S) ..
SUM(N $ (N1 (N, P))) , ZD (N, P, D, T, S)) - 1 =L=
OMEGAU6 * (1 - SL (P, D, T, S));
DRSC4 (N, P, D, T, S) $N1 (N, P) ..
ZF (N, P, D, T, S) =L= ZD (N, P, D, T, S);
DRSC5 (N, P, D, T, S) $N1 (N, P) ..
ZL (N, P, D, T, S) =L= ZD (N, P, D, T, S);
DRSC6 (N, P, D, T, S) $N1 (N, P) ..
ZF (N, P, D, T, S) - ZL (N, P, D, T-1, S) =L=
ZT (N, P, D, T, S);
DRSC7 (P, D, T, S) ..
SUM(N $ (N1 (N, P))) , ZD (N, P, D, T, S)) - 1
=L= Z7 (P, D, T, S);

*9. DRILLING RIG SCHEDULING CONSTRAINTS

*10. FLOW PROFILE CONSTRAINTS

FLOWPC (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
X1 (W, N, P, T, S) =G= X1 (W, N, P, T+1, S) -
OMEGAU7 * (1 - SUM(TDASH $ (ORD (TDASH) LE
ORD (T))) , Z1 (W, N, P, TDASH, S)));

*11. ENDOGENEOUS UNCERTAINTY IN SIZE
AND INITIAL DELIVERABILITY OF WELL

END01 (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
(XD (W, N, P, T, S) / UID (W, N, P, T, S)) +
(X5 (W, N, P, T, S) / UIS (W, N, P, T, S)) =E= 1;
END02 (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
X1 (W, N, P, T, S) =L= XD (W, N, P, T, S);
END03 (W, N, P, T, S) $ (N1 (N, P) $W1 (W, N)) ..
X5 (W, N, P, T, S) =L= UIS (W, N, P, T, S);

*12. NONANTICIPATIVITY CONSTRAINTS

NAC1 (W, N, P, T, S) $ ((ORD (S) NE
CARD (S)) $ (N1 (N, P) $W1 (W, N))) ..
Z1 (W, N, P, T, S) =E= Z1 (W, N, P, T, S+1);
NAC2 (N, P, T, S) $ ((ORD (S) LT
9) $ (N1 (N, P))) ..
Z2 (N, P, T, S) =E= Z2 (N, P, T, S+1);
NAC3 (P, T, S) $ ((ORD (S) LT 9)) ..
Z3 (P, T, S) =E= Z3 (P, T, S+1);
NAC4 (N, P, D, T, S) $ ((ORD (S) LT
9) $ (N1 (N, P))) ..
ZD (N, P, D, T, S) =E= ZD (N, P, D, T, S+1);
NAC5 (N, P, D, T, S) $ ((ORD (S) LT 9)) ..
ZT (N, P, D, T, S) =E= ZT (N, P, D, T, S+1);
NAC6 (N, P, T, S) $ ((ORD (S) LT
9) $N1 (N, P)) ..
E1 (N, P, T, S) =E= E1 (N, P, T, S+1);
NAC7 (P, T, S) $ ((ORD (S) LT 9)) ..
E2 (P, T, S) =E= E2 (P, T, S+1);

*VARIABLE BOUNDS

X1.UP (W, N, P, T, S) = 1000;

MODEL OILPLAN /ALL/
;

OPTION LIMROW = 100;
OPTION LIMCOL = 100;
OPTION OPTCR = 0.0;

SOLVE OILPLAN USING MIP MAXIMIZING Z;

```

APPENDIX V: GAMS OUTPUT
FILE ON STOCHASTIC MODEL

GAMS Rev 149 x86/MS Windows
 04/27/07 19:20:12 Page 6
 : OIL PRODUCTION PLANNING (STOCHASTIC
 MODEL)
 Execution

---- 374 VARIABLE Z.L
 = 145.580 TOTAL PRODUCTION
 PROFIT FOR ALL SCENARIOS AND
 PROBABILITIES IN

DOLLARS

---- 374 VARIABLE Z1.L 1 IF WELL W
 CONNECTED TO WELL PLATFORM N TO
 PRODUCTIO

N PLATFORM

P IS DRILLED IN TIME T

INDEX 1 = W1 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6
P1.T3	1.000	1.000	
1.000	1.000	1.000	
1.000			

	S7	S8
S9		

P1.T3	1.000	1.000
1.000		

INDEX 1 = W2 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6
P1.T1	1.000	1.000	
1.000	1.000	1.000	
1.000			

	S7	S8
S9		

P1.T1	1.000	1.000
1.000		

INDEX 1 = W3 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6
P1.T2	1.000	1.000	
1.000	1.000	1.000	
1.000			

	S7	S8
S9		

P1.T2	1.000	1.000
1.000		

INDEX 1 = W4 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6

P1.T5	1.000	1.000
1.000	1.000	1.000
1.000		

	S7	S8
S9		

P1.T5	1.000	1.000
1.000		

---- 374 VARIABLE Z2.L 1 IF WELL
 PLATFORM N CONNECTED TO PRODUCTION
 PLATFORM

P IS

INSTALLED IN TIME T

INDEX 1 = N1

	S1	S2	
S3	S4	S5	S6
P1.T1	1.000	1.000	
1.000	1.000	1.000	
1.000			

	S7	S8
S9		

P1.T1	1.000	1.000
1.000		

---- 374 VARIABLE Z3.L 1 IF
 PRODUCTION PLATFORM P IS INSTALLED IN
 TIME T

	S1	S2	
S3	S4	S5	S6
P1.T1	1.000	1.000	
1.000	1.000	1.000	
1.000			

	S7	S8
S9		

P1.T1	1.000	1.000
1.000		

---- 374 VARIABLE X1.L AMOUNT OF
 GAS OR OIL TO TRANSFER BETWEEN WELL W
 TO WELL PLATFORM N TO PRODUCTION
 PLATFORM P IN TIME T IN K

G

INDEX 1 = W1 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6
P1.T3	4.000	4.000	
4.000	5.000	5.000	
5.000			

	S7	S8
S9		

P1.T3	5.000	5.000
5.000		

INDEX 1 = W2 INDEX 2 = N1

	S1	S2	
S3	S4	S5	S6
P1.T1	4.000	4.000	
4.000	5.000	5.000	
5.000			
	S7	S8	
+ S9			
P1.T1	5.000	5.000	
5.000			
INDEX 1 = W3	INDEX 2 = N1		
	S1	S2	
S3	S4	S5	S6
P1.T2	4.000	4.000	
4.000	5.000	5.000	
5.000			
	S7	S8	
+ S9			
P1.T2	5.000	5.000	
5.000			
INDEX 1 = W4	INDEX 2 = N1		
	S1	S2	
S3	S4	S5	S6
P1.T5	4.000	4.000	
4.000	5.000	5.000	
5.000			
	S7	S8	
+ S9			
P1.T5	5.000	5.000	
5.000			

P1.T5 5.000 5.000 5.000

---- 374 VARIABLE X3.L AMOUNT OF GAS OR OIL FLOWING AT PRODUCTION PLATFORM P IN TIME T IN KG

	S1	S2	
S3	S4	S5	S6
P1.T1	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T2	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T3	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T5	4.000	4.000	
4.000	5.000	5.000	
5.000			
	S7	S8	
+ S9			
P1.T1	5.000	5.000	
5.000			
P1.T2	5.000	5.000	
5.000			
P1.T3	5.000	5.000	
5.000			
P1.T5	5.000	5.000	
5.000			

---- 374 VARIABLE X4.L SUM OF ALL GAS OR OIL FLOWING AT PRODUCTION PLATFORM P IN TIME T IN KG

	S1	S2	S3
S4	S5	S6	
T1	4.000	4.000	4.000
5.000	5.000	5.000	
T2	4.000	4.000	4.000
5.000	5.000	5.000	
T3	4.000	4.000	4.000
5.000	5.000	5.000	
T5	4.000	4.000	4.000
5.000	5.000	5.000	
	S7	S8	S9
+ T1			
T1	5.000	5.000	5.000
T2	5.000	5.000	5.000
T3	5.000	5.000	5.000
T5	5.000	5.000	5.000

---- 374 VARIABLE X2.L AMOUNT OF GAS OR OIL TO TRANSFER BETWEEN WELL PLATFORM N TO PRODUCTION PLATFORM P IN TIME T IN KG

INDEX 1 = N1

	S1	S2	
S3	S4	S5	S6
P1.T1	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T2	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T3	4.000	4.000	
4.000	5.000	5.000	
5.000			
P1.T5	4.000	4.000	
4.000	5.000	5.000	
5.000			
	S7	S8	
+ S9			
P1.T1	5.000	5.000	
5.000			
P1.T2	5.000	5.000	
5.000			
P1.T3	5.000	5.000	
5.000			

---- 374 VARIABLE Z7.L NUMBER OF TIMES DRILLING RIG D IS MOVED IN PERIOD T

(ALL 0.000)

**** REPORT FILE SUMMARY

RESPLAN2 C:\Documents and Settings\Kippi\My Documents\gamsdir\projdir\UPSTREAM2.res

EXECUTION TIME = 0.078
SECONDS 3 Mb WIN226-149 Dec 19,
2007

USER: CS/IE 635, Spring 2007
G061206/0001AS-WIN
Prof. Ferris
DC2937
License for teaching and
research at degree granting
institutions

**APPENDIX VI: COMPUTER
CODE FOR GAMS INPUT FILE
ON BRANCH CONTRACT
ALGORITHM MODEL**

\$TITLE: SCHEDULING REFINERY CRUDE OIL OPERATIONS

\$EOLCOM :

\$ONEMPTY

*number of equations listed per block
option limrow = 0;

*number of variables listed per block
option limcol=0;

*do you want solver's solution output printed?
option solprint=on;

*do you want solver's system output printed?
option sysout=off;

*milp relative termination tolerance
option optcr=0;

*milp absolute termination tolerance
option optca=0;

*Decimals to display
option decimals=8;

*File to write results
FILE RESCRU / cos1.res /;

PUT RESCRU;

PUT "-----"
"-----"
PUT / " SUBSYSTEM B.DOWNSTREAM CRUDE
OIL SCHEDULING AT REFINERY FRONT-END "
PUT / "-----"
"-----"

*-----
-----Choosing solvers

option lp=minos5;
option nlp=conopt;
option mip=osl;
option rmip=osl;
option minlp=DICOPT;

SETS

A	SET OF TANK
INPUT SOURCES	/
ST1, ST2, ST3	U1, U2, U3,
	/

B	SET OF CRUDE
TANKS	/
Tank 1	ST1 Storage
Tank 2	ST2 Storage
Tank 3	ST3 Storage

Charging Tank 1	CT1
Charging Tank 2	CT2
Charging Tank 3	CT3 /

H	SET OF TANK
OUTPUT DESTINATIONS	/
CT3, D1, D2	CT1, CT2,
	/

D(H)	SET OF
DISTILLATION UNITS	/
	D1 * D2
	/

G(B)	SET OF
CHARGING TANKS	/
	CT1 * CT3
	/

I	SET OF
COMPONENTS	/
	I1 * I2
	/

M	SOURCE UNIT
FOR SPLIT PIPELINE	/
MCT2	MST2, MST3,
	/

K	DESTINATION
UNIT OF SPLIT PIPELINE	/
	KCT3, KD2
	/

U(A)	SET OF CRUDE
SUPPLY STREAMS	/
	U1 * U3
	/

Y(B)	SET OF
STORAGE TANKS	/
	ST1 * ST3
	/

L	TRANSFER
EVENT	/
	L0 * L2
	/

Q	SET OF SUB-
STRUCTURES	/
	Q1 * Q2
	/

ALIAS

(A , ADASH)
(H , HDASH)
(G , GDASH)
(L , LDASH)

;

* ===== for
MAPPING purposes
===== *

SETS

A1(A,B) SET OF
 INPUTS A TO TANK B
 /
 U1.(ST1,ST2),U2.ST2,U3.(ST2,ST3),
 ST1.(CT1,CT2),ST2.(CT1,CT2,CT3),ST3.(CT2,CT3) /

A2(A,Y) SET OF
 INPUTS A TO STORAGE TANK Y
 /
 U1.(ST1,ST2),U2.ST2,U3.(ST2,ST3)
 /

D3(D,G) SET OF
 DISTILLATION UNITS D THAT CAN BE
 CHARGED BY CHARGING TANK G /
 D1.CT1,
 D1.CT2, D2.CT2, D2.CT3
 /

G1(G,D) SET OF
 CHARGING TANKS G THAT CHARGE
 DISTILLATION UNIT D /
 CT1.D1,
 CT2.D1, CT2.D2, CT3.D2
 /

H1(B,H) SET OF
 OUTPUTS H FROM TANK B
 /
 CT1.D1,CT2.(D1,D2),CT3.D2
 /

H2(Y,H) SET OF
 OUTPUTS H FROM STORAGE TANK Y
 /
 /

H3(G,H) SET OF
 OUTPUTS H FROM CHARGING TANK G
 /
 CT1.D1,CT2.(D1,D2),CT3.D2
 /

H4(Y,H) SET OF
 OUTPUTS H FROM STORAGE TANK Y
 /
 ST1.CT1,ST2.(CT1,CT2,CT3),ST3.CT3 /

Y1(Y,U) SET OF
 STORAGE TANKS Y CONNECTED TO CRUDE
 SUPPLY STREAMS U /
 ST1.U1,
 ST2.U1, ST2.U2, ST2.U3, ST3.U3
 /

* ===== for
 DECOMPOSITION purposes
 ===== *

B1(B) SET OF TANKS
 B BELONGING TO SUB-STRUCTURE Q1 /
 ST1, ST2,
 ST3, CT1, CT2 /

B2(B) SET OF TANKS
 B BELONGING TO SUB-STRUCTURE Q2 /
 CT3
 /

D1(D) SET OF
 DISTILLATION UNITS D PRESENT IN SUB-
 STRUCTURE Q1 /
 D1
 /

D2(D) SET OF
 DISTILLATION UNITS D PRESENT IN SUB-
 STRUCTURE Q2 /
 D2
 /

U1(U) SET OF CRUDE
 SUPPLY STREAMS U PRESENT IN SUB-
 STRUCTURE Q1 /
 U1, U2, U3
 /

U2(U) SET OF CRUDE
 SUPPLY STREAMS U PRESENT IN SUB-
 STRUCTURE Q2 /
 /

K1(M,K) SET OF
 DESTINATION UNIT K OF SPLIT PIPELINES
 WITH SOURCE M /
 MST2.KCT3,
 MST3.KCT3, MCT2.KD2
 /
 ;

*===== BRANCH AND
 CONTRACT ALGORITHM
 =====

SETS
 / CVI / I1, I2
 / CVA / U1, U2, U3, ST1,
 ST2, ST3 /
 CVB / ST1, ST2, ST3,
 CT1, CT2, CT3 /
 CVH / CT1, CT2, CT3, D1,
 D2 /
 CVL / I0 * I2
 /

SBLUEITOT(B,L)
 SREDITOT(B,L)
 SGREENITOT(B,L)
 SBLACKITOT(B,L)

ALIAS (I,J)
 (A,X)
 (B,N)
 (H,C)
 (L,T);

SBLUEITOT(B,L) = YES;
 SREDITOT(B,L) = NO;
 SGREENITOT(B,L) = NO;
 SBLACKITOT(B,L) = NO;

TS1(A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1(A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L
 TS2(B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2(B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L

*===== BRANCH AND
 CONTRACT ALGORITHM
 =====

POSITIVE VARIABLE

ITOTINI(B,L)
 ITOTBEST(B,L)
 ITOTOLD(B,L)
 ITOTNEW(B,L)

CONTADOR

MAXDIS
 MAXPCDIS
 MAXABDIS
 AUXABS
 AUXREL
 INFACIBLE
 VARCONVER
 TIEMPOTOT
 LBEST
 ABSGAP
 RELGAP

TS1Q1(A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1Q1(A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L
 TS2Q1(B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2Q1(B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L

TS1Q2(A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1Q2(A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L

TS2Q2(B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2Q2(B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L

FREE VARIABLE

FREIITOTST1L0
 FREIITOTST1L1
 FREIITOTST1L2
 FREIITOTST2L0
 FREIITOTST2L1
 FREIITOTST2L2
 FREIITOTST3L0
 FREIITOTST3L1
 FREIITOTST3L2
 FREIITOTCT1L0
 FREIITOTCT1L1
 FREIITOTCT1L2
 FREIITOTCT2L0
 FREIITOTCT2L1
 FREIITOTCT2L2
 FREIITOTCT3L0
 FREIITOTCT3L1
 FREIITOTCT3L2

TEO(U) OVERALL
 ENDING TIME OF CRUDE TRANSFER FROM
 CRUDE SUPPLY STREAM U
 TSI(U) INITIAL
 STARTING TIME OF CRUDE TRANSFER FROM
 CRUDE SUPPLY STREAM U

* ===== Flow
 variables
 ===== *

FT1(A,B,L) TOTAL FLOW
 FROM A TO B IN TRANSFER EVENT L
 FT2(B,H,L) TOTAL FLOW
 FROM B TO H IN TRANSFER EVENT L

*=====
 MAIN MODEL
 =====

POSITIVE VARIABLE

ITOT(B,L) TOTAL
 INVENTORY OF TANK B AT THE END OF
 TRANSFER EVENT L
 INVI(I,B,L) INVENTORY OF
 COMPONENT I IN TANK B AT THE END OF
 TRANSFER EVENT L

FL1(I,A,B,L) FLOW OF
 COMPONENT I FROM A TO B IN TRANSFER
 EVENT L
 FL2(I,B,H,L) FLOW OF
 COMPONENT I FROM B TO H IN TRANSFER
 EVENT L

FT1Q1(A,B,L) TOTAL FLOW
 FROM A TO B IN TRANSFER EVENT L
 FT2Q1(B,H,L) TOTAL FLOW
 FROM B TO H IN TRANSFER EVENT L

FL1Q1(I,A,B,L) FLOW OF
COMPONENT I FROM A TO B IN TRANSFER
EVENT L
FL2Q1(I,B,H,L) FLOW OF
COMPONENT I FROM B TO H IN TRANSFER
EVENT L

FT1Q2(A,B,L) TOTAL FLOW
FROM A TO B IN TRANSFER EVENT L
FT2Q2(B,H,L) TOTAL FLOW
FROM B TO H IN TRANSFER EVENT L

FL1Q2(I,A,B,L) FLOW OF
COMPONENT I FROM A TO B IN TRANSFER
EVENT L
FL2Q2(I,B,H,L) FLOW OF
COMPONENT I FROM B TO H IN TRANSFER
EVENT L

* ===== Variables to replace
bilinear terms to solve LBP
===== *

INVVJ(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
INVVT(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

INVVJQ1(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
INVVTQ1(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

INVVJQ2(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
INVVTQ2(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

FLOWVJ(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP
FLOWVT(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP

FLOWVJQ(I,A,B,H,L,Q) FLOW
OF COMPONENT I FOR LBP
FLOWVTQ(I,A,B,H,L,Q) FLOW
OF COMPONENT I FOR LBP

FLOWVJQ1(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP
FLOWVTQ1(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP

FLOWVJQ2(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP
FLOWVTQ2(I,A,B,H,L) FLOW
OF COMPONENT I FOR LBP

* ===== Duplicate
variables declaration
===== *

FLTQ(M,K,L) TOTAL FLOW
VARIABLE
FLTQ1(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
FLTQ2(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

FLQ(I,M,K,L) FLOW
VARIABLE
FLQ1(I,M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
FLQ2(I,M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

TSQ(M,K,L) START TIME
VARIABLE
TSQ1(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
TSQ2(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

TEQ(M,K,L) END TIME
VARIABLE
TEQ1(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
TEQ2(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

WBQ(M,K,L) FLOW
EXISTENCE VARIABLE
WBQ1(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
WBQ2(M,K,L) DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

ONEG1Q1(B,L) NEGATIVE
SLACK VARIABLE

OPOS1Q1(B,L) POSITIVE
SLACK VARIABLE

ONEG2Q1(I,B,L) NEGATIVE
SLACK VARIABLE

OPOS2Q1(I,B,L) POSITIVE
SLACK VARIABLE

ONEG3Q1(B,H,L) NEGATIVE
SLACK VARIABLE

OPOS3Q1(B,H,L) POSITIVE
SLACK VARIABLE

ONEG4Q1(I,B,L) NEGATIVE
SLACK VARIABLE

OPOS4Q1(I,B,L) POSITIVE
SLACK VARIABLE

ONEG5Q1(I,B,H,L) NEGATIVE
SLACK VARIABLE

OPOS5Q1(I,B,H,L) POSITIVE
SLACK VARIABLE

ONEG6Q1(I,B,H,L) NEGATIVE
SLACK VARIABLE

OPOS6Q1(I,B,H,L) POSITIVE
SLACK VARIABLE

ONEG7Q1(I,B,H,L) NEGATIVE
SLACK VARIABLE

OPOS7Q1(I,B,H,L) POSITIVE
SLACK VARIABLE

ONEG1Q2(B,L) NEGATIVE
SLACK VARIABLE

OPOS1Q2(B,L) POSITIVE
SLACK VARIABLE

ONEG2Q2(I,B,L) NEGATIVE
SLACK VARIABLE

OPOS2Q2(I,B,L) POSITIVE
SLACK VARIABLE

ONEG3Q2(B,H,L) NEGATIVE
SLACK VARIABLE

OPOS3Q2(B,H,L) POSITIVE
SLACK VARIABLE

ONEG4Q2(I,B,L) NEGATIVE
SLACK VARIABLE

OPOS4Q2(I,B,L) POSITIVE
SLACK VARIABLE

ONEG5Q2(I,B,H,L) NEGATIVE
SLACK VARIABLE

OPOS5Q2(I,B,H,L) POSITIVE
 SLACK VARIABLE
 ONEG6Q2(I,B,H,L) NEGATIVE
 SLACK VARIABLE
 OPOS6Q2(I,B,H,L) POSITIVE
 SLACK VARIABLE
 ONEG7Q2(I,B,H,L) NEGATIVE
 SLACK VARIABLE
 OPOS7Q2(I,B,H,L) POSITIVE
 SLACK VARIABLE
 ;

FREE VARIABLES

ZZ OBJECTIVE
 FUNCTION OF COST
 ZZUB OBJECTIVE
 FUNCTION FOR UPPER BOUND
 ZZFIXBIN
 ZZQ1 OBJECTIVE
 FUNCTION OF COST FOR STRUCTURE Q1
 ZZQ2 OBJECTIVE
 FUNCTION OF COST FOR STRUCTURE Q2
 ZZU OBJECTIVE
 FUNCTION UPPER BOUND
 * ZZUB OBJECTIVE
 FUNCTION FOR UPPER BOUND
 * ZZOPTQ1 OBJECTIVE
 FUNCTION (OPTIMAL) OF COST FOR
 STRUCTURE Q1
 * ZZOPTQ2 OBJECTIVE
 FUNCTION (OPTIMAL) OF COST FOR
 STRUCTURE Q2

;

BINARY VARIABLES

WB1(A,B,L) EQUAL TO 1
 IF THERE IS A FLOW FROM A TO B IN
 TRANSFER EVENT L
 WB2(B,H,L) EQUAL TO 1
 IF THERE IS A FLOW FROM B TO H IN
 TRANSFER EVENT L

WB1Q1(A,B,L) EQUAL TO 1
 IF THERE IS A FLOW FROM A TO B IN
 TRANSFER EVENT L
 WB2Q1(B,H,L) EQUAL TO 1
 IF THERE IS A FLOW FROM B TO H IN
 TRANSFER EVENT L

WB1Q2(A,B,L) EQUAL TO 1
 IF THERE IS A FLOW FROM A TO B IN
 TRANSFER EVENT L
 WB2Q2(B,H,L) EQUAL TO 1
 IF THERE IS A FLOW FROM B TO H IN
 TRANSFER EVENT L

;

*===== BRANCH AND
 CONTRACT ALGORITHM

PARAMETERS

ZERO;
 ZERO = 0;

PARAMETER

DELINIITOT(B,L)
 DIFABSITOT(B,L)
 DIFABSITOTUP(B,L)
 DIFABSITOTLO(B,L)

DIFRELITOT(B,L)
 DIFRELITOTUP(B,L)
 DIFRELITOTLO(B,L)

SOLOUNO
 INDICAREL

FLAGLOITOT(B,L)
 FLAGUPITOT(B,L)

OPTIMALL
 EPSILONABS
 EPSILONREL
 MINREDABS
 MINREDREL;

OPTIMALL = 0;
 MINREDABS = 1E-5;
 MINREDREL = 20;
 EPSILONABS = 1E-5;
 EPSILONREL = 0;

*=====
 MAIN MODEL
 =====

PARAMETERS

CSEA WAITING COST
 FOR CRUDE SUPPLY STREAMS
 / 5 /

CSET CHANGEOVER
 COST FOR CHARGED OIL SWITCH
 / 50 /

CUNLOAD UNLOADING
 COST FOR CRUDE SUPPLY STREAMS
 / 10 /

THOR TIME HORIZON
 FOR SCHEDULING
 / 12 /

ND NUMBER OF
 DISTILLATION UNITS IN THE NETWORK
 / 2 /

ND1 NUMBER OF
 DISTILLATION UNITS IN SUB-STRUCTURE Q1
 / 1 /

ND2	NUMBER OF		11.CT2
DISTILLATION UNITS IN SUB-STRUCTURE Q2		0.065	
/ 1 /			11.CT3
		0.085	
NE	NUMBER OF	/	
TRANSFER EVENTS			
/ 3 /			
			FRACT(I,U) FRACTION OF
			COMPONENT I IN CRUDE SUPPLY STREAM U
		/	
TSO	SCALAR STEP		11.U1
SIZE FOR UPDATING LAGRANGE MULTIPLIERS		0.01	
/ 1 /			11.U2
		0.085	
ALPHA	SCALAR		11.U3
BETWEEN 0 AND 2		0.06	
/ 0.4 /		/	
ZZ1	OPTIMAL		
OBJECTIVE FOR LQ1-R			INITITOT(B) INITIAL
/ 0 /			TOTAL INVENTORY OF TANK B
		/	
ZZ2	OPTIMAL		ST1 * ST3
OBJECTIVE FOR LQ2-R		20	
/ 1 /			CT1
		30	
ZZLB	SUM OF		CT2
OBJECTIVES ZZQ1 AND ZZQ2		50	
/ 1 /			CT3
* ZZUB	UPPER BOUND	30	
VALUE FOR OBJECTIVE FUNCTION		/	
/ 2 /			
CINV(B)	INVENTORY		INITINVI(I,B) INITIAL
MAINTENANCE COST FOR TANK B			INVENTORY OF COMPONENT I IN TANK B
/		/	
	ST1 * ST3		11.ST1
0.04		0.02	
	CT1 * CT3		11.ST2
0.08		0.05	
/			11.ST3
		0.08	
			11.CT1
DM(G)	DEMAND OF	0.03	
CRUDE-MIX TO BE CHARGED FROM CHARGING			11.CT2
TANK G	/	0.05	
	CT1 * CT3		11.CT3
50		0.08	
/		/	
LB1(I,B)	LOWER BOUND		LB1(B) LOWER BOUND
ON FRACTION OF COMPONENT I INSIDE TANK			ON TOTAL INVENTORY IN A TANK B
B	/	/	
	11.ST1		ST1 * ST3
0.01		1	
	11.ST2		CT1 * CT3
0.04		1	
	11.ST3	/	
0.07			
	11.CT1		UB1(B) UPPER BOUND
0.025			ON TOTAL INVENTORY IN A TANK B
	11.CT2	/	
0.045			ST1 * ST3
	11.CT3	100	
0.075			CT1 * CT3
/		100	
		/	
UB1(I,B)	UPPER BOUND		TARR(U) ARRIVAL TIME
ON FRACTION OF COMPONENT I INSIDE TANK			OF CRUDE IN CRUDE SUPPLY STREAM U
B	/	/	
	11.ST1		U1
0.03		1	
	11.ST2		U2
0.06		5	
	11.ST3		U3
0.09		9	
	11.CT1	/	
0.035			

VCRUDE (U) TOTAL VOLUME
 OF CRUDE OIL ARRIVING IN CRUDE SUPPLY
 SYSTEM U / U1*U3
 50
 /

* ===== value for these
 PARAMETERS are assigned below
 ===== *

LBF1 (A,B) LOWER BOUND
 ON FLOWRATE FROM A TO B
 UBF1 (A,B) UPPER BOUND
 ON FLOWRATE FROM A TO B

LBF2 (B,H) LOWER BOUND
 ON FLOWRATE FROM B TO H
 UBF2 (B,H) UPPER BOUND
 ON FLOWRATE FROM B TO H

LBFV1 (A,B) LOWER BOUND
 ON FLOW VOLUME FROM A TO B
 UBFV1 (A,B) UPPER BOUND
 ON FLOW VOLUME FROM A TO B

LBFV2 (B,H) LOWER BOUND
 ON FLOW VOLUME FROM B TO H
 UBFV2 (B,H) UPPER BOUND
 ON FLOW VOLUME FROM B TO H

* ===== Lagrange
 multipliers
 ===== *

LMTS (M, K, L) LAGRANGE
 MULTIPLIER
 LMTE (M, K, L) LAGRANGE
 MULTIPLIER
 LMFLT (M, K, L) LAGRANGE
 MULTIPLIER
 LMFL (I, M, K, L) LAGRANGE
 MULTIPLIER
 LMWB (M, K, L) LAGRANGE
 MULTIPLIER

* ===== Binary
 variables assignation
 ===== *

WBUB1 (A, B, L) EQUAL TO
 OPTIMAL VALUE OBTAINED FROM THE
 SOLUTION OF CRUDE_LB
 WBUB2 (B, H, L) EQUAL TO
 OPTIMAL VALUE OBTAINED FROM THE
 SOLUTION OF CRUDE_LB

* ===== Duplicate
 variables assignation
 ===== *

DFLTQ (M, K, L) TOTAL FLOW
 VARIABLE
 DFLTQ1 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DFLTQ2 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2
 DFLQ (I, M, K, L) FLOW
 VARIABLE
 DFLQ1 (I, M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1

DFLQ2 (I, M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DTSQ (M, K, L) START TIME
 VARIABLE

DTSQ1 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1

DTSQ2 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DTEQ (M, K, L) END TIME
 VARIABLE

DTEQ1 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1

DTEQ2 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DWBQ (M, K, L) FLOW
 EXISTENCE VARIABLE

DWBQ1 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1

DWBQ2 (M, K, L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

;

* ===== PARAMETER
 value assignment
 ===== *

LBF1 (A, B) = 1 ;
 UBF1 (A, B) = 40 ;

LBF2 (B, H) = 1 ;
 UBF2 (B, H) = 40 ;

LBFV1 (A, B) = 1 ;
 UBFV1 (A, B) = 100 ;

LBFV2 (B, H) = 1 ;
 UBFV2 (B, H) = 100 ;

LMTS (M, K, L) = 0 ;
 LMTE (M, K, L) = 0 ;
 LMFLT (M, K, L) = 0 ;
 LMFL (I, M, K, L) = 0 ;
 LMWB (M, K, L) = 0 ;

*===== BRANCH AND
 CONTRACT ALGORITHM
 =====

EQUATIONS

COTAOF
 ITOTLIBREST1L0
 ITOTLIBREST1L1
 ITOTLIBREST1L2
 ITOTLIBREST2L0
 ITOTLIBREST2L1
 ITOTLIBREST2L2
 ITOTLIBREST3L0
 ITOTLIBREST3L1
 ITOTLIBREST3L2
 ITOTLIBRECT1L0
 ITOTLIBRECT1L1
 ITOTLIBRECT1L2
 ITOTLIBRECT2L0
 ITOTLIBRECT2L1
 ITOTLIBRECT2L2
 ITOTLIBRECT3L0
 ITOTLIBRECT3L1

ITOTLIBRECT3L2;		13	TC13	TANK CONSTRAINT
COTAOF.. ZZ =L= ZZUB;		14	TC14	TANK CONSTRAINT
ITOTLIBREST1L0.. FREEITOTST1L0		15	TC15	TANK CONSTRAINT
=E= ITOT('ST1','L0');		16	TC16	TANK CONSTRAINT
ITOTLIBREST1L1.. FREEITOTST1L1		17	TC17	TANK CONSTRAINT
=E= ITOT('ST1','L1');		18	TC18	TANK CONSTRAINT
ITOTLIBREST1L2.. FREEITOTST1L2		19	TC19	TANK CONSTRAINT
=E= ITOT('ST1','L2');		20	TC20	TANK CONSTRAINT
ITOTLIBREST2L0.. FREEITOTST2L0		21	TC21	TANK CONSTRAINT
=E= ITOT('ST2','L0');		22	TC22	TANK CONSTRAINT
ITOTLIBREST2L1.. FREEITOTST2L1		23	TC23	TANK CONSTRAINT
=E= ITOT('ST2','L1');		24	TC24	TANK CONSTRAINT
ITOTLIBREST2L2.. FREEITOTST2L2		25	TC25	TANK CONSTRAINT
=E= ITOT('ST2','L2');		26	TC26	TANK CONSTRAINT
ITOTLIBREST3L0.. FREEITOTST3L0		27	TC27	TANK CONSTRAINT
=E= ITOT('ST3','L0');		28	TC28	TANK CONSTRAINT
ITOTLIBREST3L1.. FREEITOTST3L1		29	TC29	TANK CONSTRAINT
=E= ITOT('ST3','L1');		30	TC30	TANK CONSTRAINT
ITOTLIBREST3L2.. FREEITOTST3L2		31	TC31	TANK CONSTRAINT
=E= ITOT('ST3','L2');				
			*\$ONTEXT	
		32	TC32	TANK CONSTRAINT
		33	TC33	TANK CONSTRAINT
*=====		34	TC34	TANK CONSTRAINT
MAIN MODEL		35	TC35	TANK CONSTRAINT
=====		36	TC36	TANK CONSTRAINT
EQUATIONS				
				*\$OFFTEXT
FUNCTION OBJFUN OBJECTIVE				
FUNCTION OBJFIXBIN OBJECTIVE				
FUNCTION OBJQ1 OBJECTIVE			DC1	DISTILLATION UNIT
FUNCTION OF COST FOR STRUCTURE Q1			CONSTRAINT 1	
FUNCTION OBJQ2 OBJECTIVE			DC2	DISTILLATION UNIT
FUNCTION OF COST FOR STRUCTURE Q2			CONSTRAINT 2	
			DC3	DISTILLATION UNIT
			CONSTRAINT 3	
			DC4	DISTILLATION UNIT
			CONSTRAINT 4	
TC1 TANK CONSTRAINT 1			DC5	DISTILLATION UNIT
TC2 TANK CONSTRAINT 2			CONSTRAINT 5	
TC3 TANK CONSTRAINT 3				
TC4 TANK CONSTRAINT 4			CC1	CRUDE SUPPLY
TC5 TANK CONSTRAINT 5			STREAM CONSTRAINT 1	
TC6 TANK CONSTRAINT 6			CC2	CRUDE SUPPLY
TC7 TANK CONSTRAINT 7			STREAM CONSTRAINT 2	
TC8 TANK CONSTRAINT 8			CC3	CRUDE SUPPLY
TC9 TANK CONSTRAINT 9			STREAM CONSTRAINT 3	
TC10 TANK CONSTRAINT			CC4	CRUDE SUPPLY
10 TC11 TANK CONSTRAINT			STREAM CONSTRAINT 4	
11 TC12 TANK CONSTRAINT				
12			VB1	VARIABLE BOUND 1

	VB2	VARIABLE BOUND 2		LBP14Q	LOWER BOUNDING
	VB3	VARIABLE BOUND 3	PROBLEM 14		
	VB4	VARIABLE BOUND 4		LBP15Q	LOWER BOUNDING
	VB5	VARIABLE BOUND 5	PROBLEM 15		
	VB6	VARIABLE BOUND 6		LBP16Q	LOWER BOUNDING
	VB7	VARIABLE BOUND 7	PROBLEM 16		
	VB8	VARIABLE BOUND 8		LBP17Q	LOWER BOUNDING
	VB9	VARIABLE BOUND 9	PROBLEM 17		
	VB10	VARIABLE BOUND 10			
	VB11	VARIABLE BOUND 11		LBP1Q1	LOWER BOUNDING
	VB12	VARIABLE BOUND 12	PROBLEM 1		
	VB13	VARIABLE BOUND 13		LBP2Q1	LOWER BOUNDING
	VB14	VARIABLE BOUND 14	PROBLEM 2		
	VB15	VARIABLE BOUND 15		LBP3Q1	LOWER BOUNDING
			PROBLEM 3		
	LBP1	LOWER BOUNDING		LBP4Q1	LOWER BOUNDING
PROBLEM 1			PROBLEM 4		
	LBP2	LOWER BOUNDING		LBP5Q1	LOWER BOUNDING
PROBLEM 2			PROBLEM 5		
	LBP3	LOWER BOUNDING		LBP6Q1	LOWER BOUNDING
PROBLEM 3			PROBLEM 6		
	LBP4	LOWER BOUNDING		LBP7Q1	LOWER BOUNDING
PROBLEM 4			PROBLEM 7		
	LBP5	LOWER BOUNDING		LBP8Q1	LOWER BOUNDING
PROBLEM 5			PROBLEM 8		
	LBP6	LOWER BOUNDING		LBP9Q1	LOWER BOUNDING
PROBLEM 6			PROBLEM 9		
	LBP7	LOWER BOUNDING		LBP10Q1	LOWER BOUNDING
PROBLEM 7			PROBLEM 10		
	LBP8	LOWER BOUNDING		LBP11Q1	LOWER BOUNDING
PROBLEM 8			PROBLEM 11		
	LBP9	LOWER BOUNDING		LBP12Q1	LOWER BOUNDING
PROBLEM 9			PROBLEM 12		
	LBP10	LOWER BOUNDING		LBP13Q1	LOWER BOUNDING
PROBLEM 10			PROBLEM 13		
	LBP11	LOWER BOUNDING		LBP14Q1	LOWER BOUNDING
PROBLEM 11			PROBLEM 14		
	LBP12	LOWER BOUNDING		LBP15Q1	LOWER BOUNDING
PROBLEM 12			PROBLEM 15		
	LBP13	LOWER BOUNDING		LBP16Q1	LOWER BOUNDING
PROBLEM 13			PROBLEM 16		
	LBP14	LOWER BOUNDING		LBP17Q1	LOWER BOUNDING
PROBLEM 14			PROBLEM 17		
	LBP15	LOWER BOUNDING			
PROBLEM 15				LBP1Q2	LOWER BOUNDING
	LBP16	LOWER BOUNDING	PROBLEM 1		
PROBLEM 16				LBP2Q2	LOWER BOUNDING
	LBP17	LOWER BOUNDING	PROBLEM 2		
PROBLEM 17				LBP3Q2	LOWER BOUNDING
			PROBLEM 3		
	LBP18	LOWER BOUNDING		LBP4Q2	LOWER BOUNDING
PROBLEM 18			PROBLEM 4		
	LBP19	LOWER BOUNDING		LBP5Q2	LOWER BOUNDING
PROBLEM 19			PROBLEM 5		
	LBP20	LOWER BOUNDING		LBP6Q2	LOWER BOUNDING
PROBLEM 20			PROBLEM 6		
	LBP21	LOWER BOUNDING		LBP7Q2	LOWER BOUNDING
PROBLEM 21			PROBLEM 7		
	LBP22	LOWER BOUNDING		LBP8Q2	LOWER BOUNDING
PROBLEM 22			PROBLEM 8		
				LBP9Q2	LOWER BOUNDING
			PROBLEM 9		
				LBP10Q2	LOWER BOUNDING
			PROBLEM 10		
				LBP11Q2	LOWER BOUNDING
			PROBLEM 11		
				LBP12Q2	LOWER BOUNDING
			PROBLEM 12		
				LBP13Q2	LOWER BOUNDING
			PROBLEM 13		
				LBP14Q2	LOWER BOUNDING
			PROBLEM 14		
				LBP15Q2	LOWER BOUNDING
			PROBLEM 15		
				LBP16Q2	LOWER BOUNDING
			PROBLEM 16		

* ===== LBP for sub-
structures Q1 and Q2
===== *

	LBP10Q	LOWER BOUNDING			
PROBLEM 10					
	LBP11Q	LOWER BOUNDING			
PROBLEM 11					
	LBP12Q	LOWER BOUNDING			
PROBLEM 12					
	LBP13Q	LOWER BOUNDING			
PROBLEM 13					

LBP17Q2 LOWER BOUNDING
PROBLEM 17

* ===== DUPLICATE
EQUATIONS AND VARIABLES FOR LB1
===== *

```
* ===== Equations to
update Lagrange multiplier
===== *
          ULM1      UPDATING LAGRANGE      1      TC1Q1      TANK CONSTRAINT
MULTIPLIER 1
          ULM2      UPDATING LAGRANGE      2      TC2Q1      TANK CONSTRAINT
MULTIPLIER 2
          ULM3      UPDATING LAGRANGE      3      TC3Q1      TANK CONSTRAINT
MULTIPLIER 3
          ULM4      UPDATING LAGRANGE      4      TC4Q1      TANK CONSTRAINT
MULTIPLIER 4
          ULM5      UPDATING LAGRANGE      5      TC5Q1      TANK CONSTRAINT
MULTIPLIER 5
          ULM6      UPDATING LAGRANGE      6      TC6Q1      TANK CONSTRAINT
MULTIPLIER 6
          ULM6      UPDATING LAGRANGE      7      TC7Q1      TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE      8      TC8Q1      TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE      9      TC9Q1      TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     10     TC10Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     11     TC11Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     12     TC12Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     13     TC13Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     14     TC14Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     15     TC15Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     16     TC16Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     17     TC17Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     18     TC18Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     19     TC19Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     20     TC20Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     21     TC21Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     22     TC22Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     23     TC23Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     24     TC24Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     25     TC25Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     26     TC26Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     27     TC27Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     28     TC28Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     29     TC29Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     30     TC30Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     31     TC31Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     32     TC32Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     33     TC33Q1     TANK CONSTRAINT
          ULM6      UPDATING LAGRANGE     34     TC34Q1     TANK CONSTRAINT

* =====
Equations to generate VLC
===== *
          VLC1      GENERATING VALID      11
LINEAR CUTS
          VLC2      GENERATING VALID      12
LINEAR CUTS

* ===== Equations used to
evaluate UB after setting binary
variables ===== *
          TCUB1     DUPLICATE             16
EQUATION FOR UB EVALUATION
          TCUB2     DUPLICATE             17
EQUATION FOR UB EVALUATION
          TCUB3     DUPLICATE             18
EQUATION FOR UB EVALUATION
          TCUB4     DUPLICATE             19
EQUATION FOR UB EVALUATION
          TCUB5     DUPLICATE             20
EQUATION FOR UB EVALUATION
          TCUB6     DUPLICATE             21
EQUATION FOR UB EVALUATION
          TCUB8     DUPLICATE             22
EQUATION FOR UB EVALUATION
          TCUB11    DUPLICATE             23
EQUATION FOR UB EVALUATION
          TCUB14    DUPLICATE             24
EQUATION FOR UB EVALUATION
          TCUB15    DUPLICATE             25
EQUATION FOR UB EVALUATION
          TCUB16    DUPLICATE             26
EQUATION FOR UB EVALUATION
          TCUB17    DUPLICATE             27
EQUATION FOR UB EVALUATION
          TCUB18    DUPLICATE             28
EQUATION FOR UB EVALUATION
          DCUB1     DUPLICATE             29
EQUATION FOR UB EVALUATION
          DCUB2     DUPLICATE             30
EQUATION FOR UB EVALUATION
          DCUB4     DUPLICATE             31
EQUATION FOR UB EVALUATION
          DCUB5     DUPLICATE             32
EQUATION FOR UB EVALUATION
          CCUB1     DUPLICATE             33
EQUATION FOR UB EVALUATION
          CCUB2     DUPLICATE             34
EQUATION FOR UB EVALUATION
```

35	TC35Q1	TANK CONSTRAINT	6	TC6Q2	TANK CONSTRAINT
	TC36Q1	TANK CONSTRAINT	7	TC7Q2	TANK CONSTRAINT
36	*\$OFFTEXT		8	TC8Q2	TANK CONSTRAINT
	DC1Q1	DISTILLATION	9	TC9Q2	TANK CONSTRAINT
UNIT CONSTRAINT 1	DC2Q1	DISTILLATION	10	TC10Q2	TANK CONSTRAINT
UNIT CONSTRAINT 2	DC3Q1	DISTILLATION	11	TC11Q2	TANK CONSTRAINT
UNIT CONSTRAINT 3	DC4Q1	DISTILLATION	12	TC12Q2	TANK CONSTRAINT
UNIT CONSTRAINT 4	DC5Q1	DISTILLATION	13	TC13Q2	TANK CONSTRAINT
UNIT CONSTRAINT 5			14	TC14Q2	TANK CONSTRAINT
	CC1Q1	CRUDE SUPPLY	15	TC15Q2	TANK CONSTRAINT
STREAM CONSTRAINT 1	CC2Q1	CRUDE SUPPLY	16	TC16Q2	TANK CONSTRAINT
STREAM CONSTRAINT 2	CC3Q1	CRUDE SUPPLY	17	TC17Q2	TANK CONSTRAINT
STREAM CONSTRAINT 3	CC4Q1	CRUDE SUPPLY	18	TC18Q2	TANK CONSTRAINT
STREAM CONSTRAINT 4			19	TC19Q2	TANK CONSTRAINT
	VB1Q1	VARIABLE BOUND	20	TC20Q2	TANK CONSTRAINT
1	VB2Q1	VARIABLE BOUND	21	TC21Q2	TANK CONSTRAINT
2	VB3Q1	VARIABLE BOUND	22	TC22Q2	TANK CONSTRAINT
3	VB4Q1	VARIABLE BOUND	23	TC23Q2	TANK CONSTRAINT
4	VB5Q1	VARIABLE BOUND	24	TC24Q2	TANK CONSTRAINT
5	VB6Q1	VARIABLE BOUND	*	TC25Q2	TANK CONSTRAINT
6	VB7Q1	VARIABLE BOUND	25	TC26Q2	TANK CONSTRAINT
7	VB8Q1	VARIABLE BOUND	26	TC27Q2	TANK CONSTRAINT
8	VB9Q1	VARIABLE BOUND	27	TC28Q2	TANK CONSTRAINT
9	VB10Q1	VARIABLE BOUND	28	TC29Q2	TANK CONSTRAINT
10	VB11Q1	VARIABLE BOUND	29	TC30Q2	TANK CONSTRAINT
11	VB12Q1	VARIABLE BOUND	30	TC31Q2	TANK CONSTRAINT
12	VB13Q1	VARIABLE BOUND	*\$ONTEXT		
13	VB14Q1	VARIABLE BOUND	32	TC32Q2	TANK CONSTRAINT
14	VB15Q1	VARIABLE BOUND	33	TC33Q2	TANK CONSTRAINT
15			34	TC34Q2	TANK CONSTRAINT
			35	TC35Q2	TANK CONSTRAINT
			36	TC36Q2	TANK CONSTRAINT
				*\$OFFTEXT	
	TC1Q2	TANK CONSTRAINT		DC1Q2	DISTILLATION
1	TC2Q2	TANK CONSTRAINT	UNIT CONSTRAINT 1	DC2Q2	DISTILLATION
2	TC3Q2	TANK CONSTRAINT	UNIT CONSTRAINT 2	DC3Q2	DISTILLATION
3	TC4Q2	TANK CONSTRAINT	UNIT CONSTRAINT 3	DC4Q2	DISTILLATION
4	TC5Q2	TANK CONSTRAINT	UNIT CONSTRAINT 4		
5					

```

DC5Q2          DISTILLATION
UNIT CONSTRAINT 5
CSET*(SUM(D,SUM(G$G1(G,D),SUM(L,WB2(G,
D,L)))) - ND)
;

CC1Q2          CRUDE SUPPLY
STREAM CONSTRAINT 1
ZZFIXBIN =E=
CSEA*SUM(U,TSI(U)-TARR(U)) +
CUNLOAD*SUM(U,TEO(U)-TSI(U)) +

CC2Q2          CRUDE SUPPLY
STREAM CONSTRAINT 2
THOR*(SUM(B,CINV(B))*SUM(B,SUM(L,(ITOT
(B,L)))) + SUM(B,CINV(B))*
SUM(B,SUM(L,SUM(A$A1(A,B),FT1(A,B,L)))
) +

CC3Q2          CRUDE SUPPLY
STREAM CONSTRAINT 3
SUM(B,CINV(B))*SUM(B,SUM(L$(ORD(L) LT
CARD(L)),ITOT(B,L))) +
THOR*SUM(B,CINV(B))*SUM(B,INITITOT(B)
) / (2*NE + 1) +

CC4Q2          CRUDE SUPPLY
STREAM CONSTRAINT 4
CSET*(SUM(D,SUM(G$G1(G,D),SUM(L,WBUB2(G,
D,L)))) - ND)
;

VB1Q2          VARIABLE BOUND
1
OBJQ1..
ZZQ1 =E= CSEA*SUM(U1,TSI(U1)-
TARR(U1)) + CUNLOAD*SUM(U1,TEO(U1)-
TSI(U1)) +

VB2Q2          VARIABLE BOUND
2
THOR*(SUM(B1,CINV(B1))*SUM(B1,SUM(L,(I
TOT(B1,L)))) +
SUM(B1,CINV(B1))*SUM(B1,SUM(L,SUM(A,FT
1Q1(A,B1,L))) +

VB3Q2          VARIABLE BOUND
3
SUM(B1,CINV(B1))*SUM(B1,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B1,L))) +
THOR*SUM(B1,CINV(B1))*SUM(B2,INITITOT(B
2))) / (2*NE + 1) +

VB4Q2          VARIABLE BOUND
4
CSET*(SUM(D1,SUM(G$G1(G,D1),SUM(L,WB2Q
1(G,D1,L)))) - ND1) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ1(M,K,L)))) +

VB5Q2          VARIABLE BOUND
5
SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ1(I,M,K,L)))) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ1(M,K,L)))) +

VB6Q2          VARIABLE BOUND
6
SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ1(M,K,L)))) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ1(M,K,L)))) ;

VB7Q2          VARIABLE BOUND
7
OBJQ2..
ZZQ2 =E= CSEA*SUM(U2,TSI(U2)-
TARR(U2)) + CUNLOAD*SUM(U2,TEO(U2)-
TSI(U2)) +

VB8Q2          VARIABLE BOUND
8
THOR*(SUM(B2,CINV(B2))*SUM(B2,SUM(L,(I
TOT(B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L,SUM(A,FT
1(A,B2,L))) +

VB9Q2          VARIABLE BOUND
9
SUM(B2,CINV(B2))*SUM(B2,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B2,L))) +
THOR*SUM(B2,CINV(B2))*SUM(B2,INITITOT(B
2))) / (2*NE + 1) +

VB10Q2         VARIABLE BOUND
10
CSET*(SUM(D2,SUM(G$G1(G,D2),SUM(L,WB2(G,
D2,L)))) - ND2) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ2(M,K,L)))) -

VB11Q2         VARIABLE BOUND
11
SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ2(I,M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ2(M,K,L)))) -

VB12Q2         VARIABLE BOUND
12
* ===== OBJECTIVE
FUNCTION =====
*

OBJFUN..
ZZ =E= CSEA*SUM(U,TSI(U)-
TARR(U)) + CUNLOAD*SUM(U,TEO(U)-
TSI(U)) +

THOR*(SUM(B,CINV(B))*SUM(B,SUM(L,(ITOT
(B,L)))) + SUM(B,CINV(B))*
SUM(B,SUM(L,SUM(A$A1(A,B),FT1(A,B,L)))
) +

SUM(B,CINV(B))*SUM(B,SUM(L$(ORD(L) LT
CARD(L)),ITOT(B,L))) +
THOR*SUM(B,CINV(B))*SUM(B,INITITOT(B)
) / (2*NE + 1) +

```

```

SUM(M, SUM(K$K1(M, K), SUM(L, LMTE(M, K, L) *
TEQ2(M, K, L))) -
SUM(M, SUM(K$K1(M, K), SUM(L, LMWB(M, K, L) *
WBQ2(M, K, L)))));

* ===== 1. TANK
CONSTRAINTS
===== *

**(i) Constraints for Flow Transfers

TC1(A, B, L) $A1(A, B) ..
    FT1(A, B, L) =L=
    UBFV1(A, B) *WB1(A, B, L)
;

TC2(B, H, L) $H1(B, H) ..
    FT2(B, H, L) =L=
    UBFV2(B, H) *WB2(B, H, L)
;

TCUB1(A, B, L) $A1(A, B) ..
    FT1(A, B, L) =L=
    UBFV1(A, B) *WBUB1(A, B, L)
;

TCUB2(B, H, L) $H1(B, H) ..
    FT2(B, H, L) =L=
    UBFV2(B, H) *WBUB2(B, H, L)
;

**(ii) Duration Constraints

TC3(A, B, L) $A1(A, B) ..
    UBF1(A, B) * (TE1(A, B, L) -
    TS1(A, B, L)) + UBF1(A, B) * THOR * (1 -
    WB1(A, B, L)) =G=
    FT1(A, B, L)
;

TC4(B, H, L) $H1(B, H) ..
    UBF2(B, H) * (TE2(B, H, L) -
    TS2(B, H, L)) + UBF2(B, H) * THOR * (1 -
    WB2(B, H, L)) =G=
    FT2(B, H, L)
;

TC5(A, Y, L) $A2(A, Y) ..
    LBF1(A, Y) * (TE1(A, Y, L) -
    TS1(A, Y, L)) - LBF1(A, Y) * THOR * (1 -
    WB1(A, Y, L)) =L=
    FT1(A, Y, L)
;

TC6(Y, H, L) $H2(Y, H) ..
    LBF2(Y, H) * (TE2(Y, H, L) -
    TS2(Y, H, L)) - LBF2(Y, H) * THOR * (1 -
    WB2(Y, H, L)) =L=
    FT2(Y, H, L)
;

TC7(G, H, L) $H3(G, H) ..
    LBF2(G, H) * (TE2(G, H, L) -
    TS2(G, H, L)) =L=
    FT2(G, H, L)
;

TCUB3(A, B, L) $A1(A, B) ..
    UBF1(A, B) * (TE1(A, B, L) -
    TS1(A, B, L)) + UBF1(A, B) * THOR * (1 -
    WBUB1(A, B, L)) =G=
    FT1(A, B, L)
;

TCUB4(B, H, L) $H1(B, H) ..
    UBF2(B, H) * (TE2(B, H, L) -
    TS2(B, H, L)) + UBF2(B, H) * THOR * (1 -
    WBUB2(B, H, L)) =G=
    FT2(B, H, L)
;

TCUB5(A, Y, L) $A2(A, Y) ..
    LBF1(A, Y) * (TE1(A, Y, L) -
    TS1(A, Y, L)) - LBF1(A, Y) * THOR * (1 -
    WBUB1(A, Y, L)) =L=
    FT1(A, Y, L)
;

TCUB6(Y, H, L) $H2(Y, H) ..
    LBF2(Y, H) * (TE2(Y, H, L) -
    TS2(Y, H, L)) - LBF2(Y, H) * THOR * (1 -
    WBUB2(Y, H, L)) =L=
    FT2(Y, H, L)
;

**(iii) Simple Sequencing Constraints

TC8(A, B, L) $ (ORD(L) LT
CARD(L) $A1(A, B)) ..
    TS1(A, B, L+1) =G=
    TE1(A, B, L) - THOR * (1 -
    WB1(A, B, L))
;

TC9(A, B, L) $ (ORD(L) LT
CARD(L) $A1(A, B)) ..
    TS1(A, B, L+1) =G=
    TS1(A, B, L)
;

TC10(A, B, L) $ (ORD(L) LT
CARD(L) $A1(A, B)) ..
    TE1(A, B, L+1) =G=
    TE1(A, B, L)
;

TC11(B, H, L) $ (ORD(L) LT
CARD(L) $H1(B, H)) ..
    TS2(B, H, L+1) =G=
    TE2(B, H, L) - THOR * (1 -
    WB2(B, H, L))
;

TC12(B, H, L) $ (ORD(L) LT
CARD(L) $H1(B, H)) ..
    TS2(B, H, L+1) =G=
    TS2(B, H, L)
;

TC13(B, H, L) $ (ORD(L) LT
CARD(L) $H1(B, H)) ..
    TE2(B, H, L+1) =G=
    TE2(B, H, L)
;

TCUB8(A, B, L) $ (ORD(L) LT
CARD(L) $A1(A, B)) ..
    TS1(A, B, L+1) =G=
    TE1(A, B, L) - THOR * (1 -
    WBUB1(A, B, L))
;

```

```

TCUB11(B,H,L)$(ORD(L) LT
CARD(L)$H1(B,H))..
    TS2(B,H,L+1) =G=
    TE2(B,H,L) - THOR*(1-
WBUB2(B,H,L))
;

```

** (iv) Input and Output Restraints for the Entire Horizon

```

TC14(A,ADASH,B,L)$(ORD(L) LT
CARD(L)$(ORD(A) NE
ORD(ADASH)$A1(A,B)))..
    TS1(A,B,L+1) =G=
    TE1(ADASH,B,L) - THOR*(1-
WB1(ADASH,B,L))
;

```

```

TC15(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H)))..
    TS1(A,B,L+1) =G=
    TE2(B,H,L) - THOR*(1-
WB2(B,H,L))
;

```

```

TC16(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H)))..
    TS2(B,H,L+1) =G=
    TE1(A,B,L) - THOR*(1-
WB1(A,B,L))
;

```

```

TC17(B,H,HDASH,L)$(ORD(L) LT
CARD(L)$(ORD(H) NE
ORD(HDASH)$H1(B,H)))..
    TS2(B,H,L+1) =G=
    TE2(B,HDASH,L) - THOR*(1-
WB2(B,HDASH,L))
;

```

```

TC18(A,B,H,L)$(A1(A,B)$H1(B,H))..
    TE1(A,B,L) - THOR*(1-
WB1(A,B,L)) =L=
    TS2(B,H,L) + THOR*(1-
WB2(B,H,L))
;

```

```

TCUB14(A,ADASH,B,L)$(ORD(L) LT
CARD(L)$(ORD(A) NE
ORD(ADASH)$A1(A,B)))..
    TS1(A,B,L+1) =G=
    TE1(ADASH,B,L) - THOR*(1-
WBUB1(ADASH,B,L))
;

```

```

TCUB15(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H)))..
    TS1(A,B,L+1) =G=
    TE2(B,H,L) - THOR*(1-
WBUB2(B,H,L))
;

```

```

TCUB16(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H)))..
    TS2(B,H,L+1) =G=
    TE1(A,B,L) - THOR*(1-
WBUB1(A,B,L))
;

```

```

TCUB17(B,H,HDASH,L)$(ORD(L) LT
CARD(L)$(ORD(H) NE
ORD(HDASH)$H1(B,H)))..
    TS2(B,H,L+1) =G=

```

```

    TE2(B,HDASH,L) - THOR*(1-
WBUB2(B,HDASH,L))
;

```

```

    TCUB18(A,B,H,L)$(A1(A,B)$H1(B,H))..
    TE1(A,B,L) - THOR*(1-
WBUB1(A,B,L)) =L=
    TS2(B,H,L) + THOR*(1-
WBUB2(B,H,L))
;

```

** (v) Mass Balances

```

TC19(B,L)..
    ITOT(B,L-1) +
SUM(A$A1(A,B),FT1(A,B,L)) =E=
    ITOT(B,L) +
SUM(H$H1(B,H),FT2(B,H,L))
;

```

```

TC20(B)..
    ITOT(B,'L0') =E=
    INITITOT(B)
;

```

```

TC21(I,B,L)..
    INVI(I,B,L-1) +
SUM(A$A1(A,B),FL1(I,A,B,L)) =E=
    INVI(I,B,L)
+SUM(H$H1(B,H),FL2(I,B,H,L))
;

```

```

TC22(I,B)..
    INVI(I,B,'L0') =E=
    INITINVI(I,B)
;

```

```

TC23(A,B,L)$A1(A,B)..
    FT1(A,B,L) =E=
    SUM(I,FL1(I,A,B,L))
;

```

```

TC24(B,H,L)$H1(B,H)..
    FT2(B,H,L) =E=
    SUM(I,FL2(I,B,H,L))
;

```

** (vi) Component Balances (Replaced by LBP)

```

TC25(I,B,H,L)$H1(B,H)..
    FL2(I,B,H,L)*(ITOT(B,L-
1)+SUM(A$A1(A,B),FT1(A,B,L))) =E=
    FT2(B,H,L)*(INVI(I,B,L-
1)+SUM(A$A1(A,B),FL1(I,A,B,L)));

```

```

LBP1(I,B,H,L)$H1(B,H)..
    INVVJ(I,B,H,L) +
SUM(A$A1(A,B),FLOWVJ(I,A,B,H,L)) =E=
    INVVT(I,B,H,L) +
SUM(A$A1(A,B),FLOWVT(I,A,B,H,L))
;

```

```

LBP2(I,B,H,L)$H1(B,H)..
    INVVJ(I,B,H,L) =G=
    LBI1(B)*FL2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H)
;

```

```

LBP3(I,B,H,L)$H1(B,H)..
    INVVJ(I,B,H,L) =G=

```



```

      UBI1(B)*FL2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
UBI1(B)*UBFV2(B,H) ;

      LBP4(I,B,H,L)$SH1(B,H)..
      INVVJ(I,B,H,L) =L=
      LBI1(B)*FL2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*UBFV2(B,H) ;

      LBP5(I,B,H,L)$SH1(B,H)..
      INVVJ(I,B,H,L) =L=
      UBI1(B)*FL2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

      LBP6(I,B,H,L)$SH1(B,H)..
      INVVT(I,B,H,L) =G=
      LBI1(B)*FT2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*LBFV2(B,H) ;

      LBP7(I,B,H,L)$SH1(B,H)..
      INVVT(I,B,H,L) =G=
      UBI1(B)*FT2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*UBFV2(B,H) ;

      LBP8(I,B,H,L)$SH1(B,H)..
      INVVT(I,B,H,L) =L=
      LBI1(B)*FT2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*UBFV2(B,H) ;

      LBP9(I,B,H,L)$SH1(B,H)..
      INVVT(I,B,H,L) =L=
      UBI1(B)*FT2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*LBFV2(B,H) ;

      LBP10(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVJ(I,A,B,H,L) =G=
      LBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

      LBP11(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVJ(I,A,B,H,L) =G=
      UBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

      LBP12(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVJ(I,A,B,H,L) =L=
      LBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

      LBP13(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVJ(I,A,B,H,L) =L=
      UBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

      LBP14(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVT(I,A,B,H,L) =G=
      LBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

      LBP15(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVT(I,A,B,H,L) =G=
      UBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

      LBP16(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVT(I,A,B,H,L) =L=
      LBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

      LBP17(I,A,B,H,L)$ (A1(A,B)$SH1(B,H))..
      FLOWVT(I,A,B,H,L) =L=
      UBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

* ===== for generating
cutting planes using Lagrangean
decomposition ===== *

      LBP10Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVJQ(I,A,B,H,L,Q) =G=
      LBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

*
      LBP10Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVJQ(I,M,N,H,L,Q) =G=
      *LBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

      LBP11Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVJQ(I,A,B,H,L,Q) =G=
      UBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

      LBP12Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVJQ(I,A,B,H,L,Q) =L=
      LBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

      LBP13Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVJQ(I,A,B,H,L,Q) =L=
      UBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

      LBP14Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVTQ(I,A,B,H,L,Q) =G=
      LBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

      LBP15Q(I,A,B,H,L,Q)$ (A1(A,B)$SH1(B,H))..
      FLOWVTQ(I,A,B,H,L,Q) =G=

```

```

UBFV1 (A, B) * FT2 (B, H, L) +
UBFV2 (B, H) * FL1 (I, A, B, L) -
UBFV1 (A, B) * UBFV2 (B, H) ;

LBP16Q (I, A, B, H, L, Q) $ (A1 (A, B) $ H1 (B, H)) .
.
FLOWVTQ (I, A, B, H, L, Q) =L=
LBFV1 (A, B) * FT2 (B, H, L) +
UBFV2 (B, H) * FL1 (I, A, B, L) -
LBFV1 (A, B) * UBFV2 (B, H) ;

LBP17Q (I, A, B, H, L, Q) $ (A1 (A, B) $ H1 (B, H)) .
.
FLOWVTQ (I, A, B, H, L, Q) =L=
UBFV1 (A, B) * FT2 (B, H, L) +
LBFV2 (B, H) * FL1 (I, A, B, L) -
UBFV1 (A, B) * LBFV2 (B, H) ;

* ===== sub-
structure Q1 (w/o Q index)
===== *

LBP1Q1 (I, B, H, L) $ H1 (B, H) . .
INVJQ1 (I, B, H, L) +
SUM (A$A1 (A, B) , FLOWVJQ1 (I, A, B, H, L)) =E=
INVVTQ1 (I, B, H, L) +
SUM (A$A1 (A, B) , FLOWVTQ1 (I, A, B, H, L))
;

LBP2Q1 (I, B, H, L) $ H1 (B, H) . .
INVJQ1 (I, B, H, L) =G=
LBI1 (B) * FL2Q1 (I, B, H, L) +
LBFV2 (B, H) * ITOT (B, L-1) -
LBI1 (B) * LBFV2 (B, H) ;

LBP3Q1 (I, B, H, L) $ H1 (B, H) . .
INVJQ1 (I, B, H, L) =G=
UBI1 (B) * FL2Q1 (I, B, H, L) +
UBFV2 (B, H) * ITOT (B, L-1) -
UBI1 (B) * UBFV2 (B, H) ;

LBP4Q1 (I, B, H, L) $ H1 (B, H) . .
INVJQ1 (I, B, H, L) =L=
LBI1 (B) * FL2Q1 (I, B, H, L) +
UBFV2 (B, H) * ITOT (B, L-1) -
LBI1 (B) * UBFV2 (B, H) ;

LBP5Q1 (I, B, H, L) $ H1 (B, H) . .
INVJQ1 (I, B, H, L) =L=
UBI1 (B) * FL2Q1 (I, B, H, L) +
LBFV2 (B, H) * ITOT (B, L-1) -
LBI1 (B) * LBFV2 (B, H) ;

LBP6Q1 (I, B, H, L) $ H1 (B, H) . .
INVVTQ1 (I, B, H, L) =G=
LBI1 (B) * FT2Q1 (B, H, L) +
LBFV2 (B, H) * INVI (I, B, L-1) -
LBI1 (B) * LBFV2 (B, H) ;

LBP7Q1 (I, B, H, L) $ H1 (B, H) . .
INVVTQ1 (I, B, H, L) =G=
UBI1 (B) * FT2Q1 (B, H, L) +
UBFV2 (B, H) * INVI (I, B, L-1) -
UBI1 (B) * UBFV2 (B, H) ;

LBP8Q1 (I, B, H, L) $ H1 (B, H) . .
INVVTQ1 (I, B, H, L) =L=
LBI1 (B) * FT2Q1 (B, H, L) +
UBFV2 (B, H) * INVI (I, B, L-1) -
LBI1 (B) * UBFV2 (B, H) ;

LBP9Q1 (I, B, H, L) $ H1 (B, H) . .
INVVTQ1 (I, B, H, L) =L=
UBI1 (B) * FT2Q1 (B, H, L) +
LBFV2 (B, H) * INVI (I, B, L-1) -
UBI1 (B) * LBFV2 (B, H) ;

LBP10Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVJQ1 (I, A, B, H, L) =G=
LBFV1 (A, B) * FL2Q1 (I, B, H, L) +
LBFV2 (B, H) * FT1Q1 (A, B, L) -
LBFV1 (A, B) * LBFV2 (B, H) ;

LBP11Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVJQ1 (I, A, B, H, L) =G=
UBFV1 (A, B) * FL2Q1 (I, B, H, L) +
UBFV2 (B, H) * FT1Q1 (A, B, L) -
UBFV1 (A, B) * UBFV2 (B, H) ;

LBP12Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVJQ1 (I, A, B, H, L) =L=
LBFV1 (A, B) * FL2Q1 (I, B, H, L) +
UBFV2 (B, H) * FT1Q1 (A, B, L) -
LBFV1 (A, B) * UBFV2 (B, H) ;

LBP13Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVJQ1 (I, A, B, H, L) =L=
UBFV1 (A, B) * FL2Q1 (I, B, H, L) +
LBFV2 (B, H) * FT1Q1 (A, B, L) -
UBFV1 (A, B) * LBFV2 (B, H) ;

LBP14Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVTQ1 (I, A, B, H, L) =G=
LBFV1 (A, B) * FT2Q1 (B, H, L) +
LBFV2 (B, H) * FL1Q1 (I, A, B, L) -
LBFV1 (A, B) * LBFV2 (B, H) ;

LBP15Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVTQ1 (I, A, B, H, L) =G=
UBFV1 (A, B) * FT2Q1 (B, H, L) +
UBFV2 (B, H) * FL1Q1 (I, A, B, L) -
UBFV1 (A, B) * UBFV2 (B, H) ;

LBP16Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVTQ1 (I, A, B, H, L) =L=
LBFV1 (A, B) * FT2Q1 (B, H, L) +
UBFV2 (B, H) * FL1Q1 (I, A, B, L) -
LBFV1 (A, B) * UBFV2 (B, H) ;

LBP17Q1 (I, A, B, H, L) $ (A1 (A, B) $ H1 (B, H)) . .
FLOWVTQ1 (I, A, B, H, L) =L=
UBFV1 (A, B) * FT2Q1 (B, H, L) +
LBFV2 (B, H) * FL1Q1 (I, A, B, L) -
UBFV1 (A, B) * LBFV2 (B, H) ;

* ===== sub-
structure Q2 (w/o Q index)
===== *

LBP1Q2 (I, B, H, L) $ H1 (B, H) . .
INVJQ2 (I, B, H, L) +
SUM (A$A1 (A, B) , FLOWVJQ2 (I, A, B, H, L)) =E=
INVVTQ2 (I, B, H, L) +
SUM (A$A1 (A, B) , FLOWVTQ2 (I, A, B, H, L))
;

```

```

LBP2Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =G=
    LBI1(B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

LBP3Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =G=
    UBI1(B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP4Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =L=
    LBI1(B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*UBFV2(B,H) ;

LBP5Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =L=
    UBI1(B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

LBP6Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =G=
    LBI1(B)*FT2Q2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*LBFV2(B,H) ;

LBP7Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =G=
    UBI1(B)*FT2Q2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP8Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =L=
    LBI1(B)*FT2Q2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*UBFV2(B,H) ;

LBP9Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =L=
    UBI1(B)*FT2Q2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*LBFV2(B,H) ;

LBP10Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =G=
    LBFV1(A,B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*FT1Q2(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP11Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =G=
    UBFV1(A,B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*FT1Q2(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP12Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =L=
    LBFV1(A,B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*FT1Q2(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP13Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =L=
    UBFV1(A,B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*FT1Q2(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

LBP14Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =G=
    LBFV1(A,B)*FT2Q2(B,H,L) +
LBFV2(B,H)*FL1Q2(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP15Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =G=
    UBFV1(A,B)*FT2Q2(B,H,L) +
UBFV2(B,H)*FL1Q2(I,A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP16Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =L=
    LBFV1(A,B)*FT2Q2(B,H,L) +
UBFV2(B,H)*FL1Q2(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP17Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =L=
    UBFV1(A,B)*FT2Q2(B,H,L) +
LBFV2(B,H)*FL1Q2(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

LBP18(M,K,L)$K1(M,K)..
    FLTQ1(M,K,L) =E=
    FLTQ2(M,K,L) ;

LBP19(I,M,K,L)$K1(M,K)..
    FLQ1(I,M,K,L) =E=
    FLQ1(I,M,K,L) ;

LBP20(M,K,L)$K1(M,K)..
    TSQ1(M,K,L) =E=
    TSQ2(M,K,L) ;

LBP21(M,K,L)$K1(M,K)..
    TEQ1(M,K,L) =E=
    TEQ2(M,K,L) ;

LBP22(M,K,L)$K1(M,K)..
    WBQ1(M,K,L) =E=
    WBQ2(M,K,L) ;

**(vii) Inventory Bounds
TC26(B,L)$ (ORD(L) GT 1)..
    ITOT(B,L-
1)+SUM(A$A1(A,B),FT1(A,B,L)) =L=
    UBI1(B) ;

**(viii) Bounds on Components Fractions
inside a Tank
TC27(I,B,L)..
    LB1(I,B)*ITOT(B,L) =L=
    INVI(I,B,L) ;

```

```

TC28 (I, B, L) ..
    INVI (I, B, L) =L=
    UB1 (I, B) * ITOT (B, L)
;

TC29 (I, B, H, L) $H1 (B, H) ..
    LB1 (I, B) * FT2 (B, H, L) =L=
    FL2 (I, B, H, L)
;

TC30 (I, B, H, L) $H1 (B, H) ..
    FL2 (I, B, H, L) =L=
    UB1 (I, B) * FT2 (B, H, L)
;

** (ix) Crude-mix Demand Constraints

TC31 (G) ..
SUM (D$D3 (D, G), SUM (L, FT2 (G, D, L))) =E=
    DM (G)
;

** (x) Bound Strengthening Cuts

TC32 (I, B, H, L) $H1 (B, H) ..
    ITOT (B, L-1) * FL2 (I, B, H, L) +
SUM (A$A1 (A, B), FT1 (A, B, L) * FL2 (I, B, H, L))
=E=
    INVI (I, B, L-
1) * FT2 (B, H, L) + SUM (A$A1 (A, B), FL1 (I, A, B,
L) * FT2 (B, H, L))
;

TC33 (B, H, L) $H1 (B, H) ..
    SUM (I, ITOT (B, L-
1) * FL2 (I, B, H, L)) =E=
    ITOT (B, L-1) * FT2 (B, H, L)
;

TC34 (A, B, H, L) $ (A1 (A, B) $H1 (B, H)) ..
SUM (I, FT1 (A, B, L) * FL2 (I, B, H, L)) =E=
    FT1 (A, B, L) * FT2 (B, H, L)
;

TC35 (B, H, L) $H1 (B, H) ..
    SUM (I, INVI (I, B, L-
1) * FT2 (B, H, L)) =E=
    ITOT (B, L-1) * FT2 (B, H, L)
;

TC36 (A, B, H, L) $ (A1 (A, B) $H1 (B, H)) ..
SUM (I, FL1 (I, A, B, L) * FT2 (B, H, L)) =E=
    FT1 (A, B, L) * FT2 (B, H, L)
;

* =====
2. DISTILLATION UNIT CONSTRAINTS
===== *

** (i) Allocation Constraints

DC1 (D, L) ..
    SUM (G$G1 (G, D), WB2 (G, D, L)) =L=
    1
;

DC2 (G, L) ..
    SUM (D$D3 (D, G), WB2 (G, D, L)) =L=
    1
;

DCUB1 (D, L) ..
    SUM (G$G1 (G, D), WBUB2 (G, D, L))
=L=
    1
;

DCUB2 (G, L) ..
    SUM (D$D3 (D, G), WBUB2 (G, D, L))
=L=
    1
;

** (ii) Continuous Operation Constraints

DC3 (D) ..
SUM (L, SUM (G$G1 (G, D), TE2 (G, D, L) -
TS2 (G, D, L))) =E=
    THOR
;

DC4 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D))) ..
    TS2 (G, D, L+1) =G=
    TE2 (GDASH, D, L) - THOR * (1-
WB2 (GDASH, D, L))
;

DC5 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D))) ..
    TS2 (G, D, L+1) =L=
    TE2 (GDASH, D, L) + THOR * (1-
WB2 (GDASH, D, L))
;

DCUB4 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D))) ..
    TS2 (G, D, L+1) =G=
    TE2 (GDASH, D, L) - THOR * (1-
WBUB2 (GDASH, D, L))
;

DCUB5 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D))) ..
    TS2 (G, D, L+1) =L=
    TE2 (GDASH, D, L) + THOR * (1-
WBUB2 (GDASH, D, L))
;

*3. CRUDE SUPPLY STREAM CONSTRAINTS

** (i) Timing Constraints

CC1 (U, Y, L) $Y1 (Y, U) ..
    TSI (U) =L=
    TS1 (U, Y, L) + THOR * (1-
WB1 (U, Y, L))
;

CC2 (U, Y, L) $Y1 (Y, U) ..
    TEO (U) =G=
    TE1 (U, Y, L) - THOR * (1-
WB1 (U, Y, L))
;

```

```

CCUB1 (U, Y, L) $Y1 (Y, U) . .
    TSI (U) =L=
    TSI (U, Y, L) + THOR*(1-
WBUB1 (U, Y, L) )
;

CCUB2 (U, Y, L) $Y1 (Y, U) . .
    TEO (U) =G=
    TEL (U, Y, L) - THOR*(1-
WBUB1 (U, Y, L) )
;

**(ii)Overall Mass Balances
CC3 (U) . .
SUM (L, SUM (Y$Y1 (Y, U) , FT1 (U, Y, L) ) ) =E=
    VCRUDE (U)
;

**(iii)Component Balances
CC4 (I, U, Y, L) $Y1 (Y, U) . .
    FL1 (I, U, Y, L) =E=
    FRACT (I, U) *FT1 (U, Y, L)
;

*4.VARIABLE BOUNDS
VB1 (I, B, L) . .
    INVI (I, B, L) =L=
    UBI1 (B)
;
VB2 (B, L) . .
    LBI1 (B) =L=
    ITOT (B, L)
;
VB3 (B, L) . .
    ITOT (B, L) =L=
    UBI1 (B)
;
VB4 (I, A, B, L) $A1 (A, B) . .
    FL1 (I, A, B, L) =L=
    UBFV1 (A, B)
;
VB5 (I, B, H, L) $H1 (B, H) . .
    FL2 (I, B, H, L) =L=
    UBFV2 (B, H)
;
VB6 (A, B, L) $A1 (A, B) . .
    FT1 (A, B, L) =L=
    UBFV1 (A, B)
;
VB7 (B, H, L) $H1 (B, H) . .
    FT2 (B, H, L) =L=
    UBFV2 (B, H)
;
VB8 (A, B, L) $A1 (A, B) . .
    TS1 (A, B, L) =L=
    THOR
;
VB9 (A, B, L) $A1 (A, B) . .
    TE1 (A, B, L) =L=

```

```

THOR
;
VB10 (B, H, L) $H1 (B, H) . .
    TS2 (B, H, L) =L=
    THOR
;
VB11 (B, H, L) $H1 (B, H) . .
    TE2 (B, H, L) =L=
    THOR
;
VB12 (U) . .
    TARR (U) =L=
    TSI (U)
;
VB13 (U) . .
    TSI (U) =L=
    THOR
;
VB14 (U) . .
    TARR (U) =L=
    TEO (U)
;
VB15 (U) . .
    TEO (U) =L=
    THOR
;

* =====
GENERATE VALID LINEAR CUTS
===== *

VLC1 . .
    ZZ1 =L=
    CSEA*SUM (U1, TSI (U1) -TARR (U1) )
+ CUNLOAD*SUM (U1, TEO (U1) -TSI (U1) ) +
THOR*( SUM (B1, CINV (B1) ) *SUM (B1, SUM (L, (I
TOT (B1, L) ) ) ) +
SUM (B1, CINV (B1) ) *SUM (B1, SUM (L, SUM (A, FT
1 (A, B1, L) ) ) ) +
SUM (B1, CINV (B1) ) *SUM (B1, SUM (L$ (ORD (L)
LT CARD (L) ) , ITOT (B1, L) ) ) +
THOR*SUM (B1, CINV (B1) ) *SUM (B2, INITITOT (
B2) ) ) / (2*NE + 1) +
CSET*( SUM (D1, SUM (G$G1 (G, D1) , SUM (L, WB2 (
G, D1, L) ) ) ) - ND1) +
SUM (M, SUM (K$K1 (M, K) , SUM (L, LMFLT (M, K, L)
*FLTQ (M, K, L) ) ) ) +
SUM (I, SUM (M, SUM (K$K1 (M, K) , SUM (L, LMFL (I
, M, K, L) *FLQ (I, M, K, L) ) ) ) ) +
SUM (M, SUM (K$K1 (M, K) , SUM (L, LMFS (M, K, L) *
TSQ (M, K, L) ) ) ) +
SUM (M, SUM (K$K1 (M, K) , SUM (L, LMTE (M, K, L) *
TEQ (M, K, L) ) ) ) +
SUM (M, SUM (K$K1 (M, K) , SUM (L, LMWB (M, K, L) *
WBQ (M, K, L) ) ) )
;
VLC2 . .
    ZZ2 =L=
    CSEA*SUM (U2, TSI (U2) -TARR (U2) )
+ CUNLOAD*SUM (U2, TEO (U2) -TSI (U2) ) +

```

```

THOR*(SUM(B2,CINV(B2))*SUM(B2,SUM(L,(I
TOT(B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L,SUM(A,FT
1(A,B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B2,L))) +
THOR*SUM(B2,CINV(B2))*SUM(B2,INITITOT(
B2)) / (2*NE + 1) +
CSET*(SUM(D2,SUM(G$G1(G,D2),SUM(L,WB2(
G,D2,L)))) - ND2) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ(M,K,L)))) -
SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ(I,M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ(M,K,L))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ(M,K,L))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ(M,K,L)))
;
* ===== FOR DUPLICATING
EQUATIONS AND VARIABLES IN Q1
===== *
TC1Q1(A,B,L)$A1(A,B)..
FT1Q1(A,B,L) =L=
UBFV1(A,B)*WB1Q1(A,B,L)
;
TC2Q1(B,H,L)$H1(B,H)..
FT2Q1(B,H,L) =L=
UBFV2(B,H)*WB2Q1(B,H,L)
;
TC3Q1(A,B,L)$A1(A,B)..
UBF1(A,B)*(TE1Q1(A,B,L) -
TS1Q1(A,B,L)) + UBF1(A,B)*THOR*(1-
WB1Q1(A,B,L)) =G=
FT1Q1(A,B,L)
;
TC4Q1(B,H,L)$H1(B,H)..
UBF2(B,H)*(TE2Q1(B,H,L) -
TS2Q1(B,H,L)) + UBF2(B,H)*THOR*(1-
WB2Q1(B,H,L)) =G=
FT2Q1(B,H,L)
;
TC5Q1(A,Y,L)$A2(A,Y)..
LBF1(A,Y)*(TE1Q1(A,Y,L) -
TS1Q1(A,Y,L)) - LBF1(A,Y)*THOR*(1-
WB1Q1(A,Y,L)) =L=
FT1Q1(A,Y,L)
;
TC6Q1(Y,H,L)$H2(Y,H)..
LBF2(Y,H)*(TE2Q1(Y,H,L) -
TS2Q1(Y,H,L)) - LBF2(Y,H)*THOR*(1-
WB2Q1(Y,H,L)) =L=
FT2Q1(Y,H,L)
;
TC7Q1(G,H,L)$H3(G,H)..
LBF2(G,H)*(TE2Q1(G,H,L) -
TS2Q1(G,H,L)) =L=
FT2Q1(G,H,L)
;
TC8Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TS1Q1(A,B,L+1) =G=
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L))
;
TC9Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TS1Q1(A,B,L+1) =G=
TS1Q1(A,B,L)
;
TC10Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TE1Q1(A,B,L+1) =G=
TE1Q1(A,B,L)
;
TC11Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TS2Q1(B,H,L+1) =G=
TE2Q1(B,H,L) - THOR*(1-
WB2Q1(B,H,L))
;
TC12Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TS2Q1(B,H,L+1) =G=
TS2Q1(B,H,L)
;
TC13Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TE2Q1(B,H,L+1) =G=
TE2Q1(B,H,L)
;
TC14Q1(A,ADASH,B,L)$ (ORD(L) LT
CARD(L)$ (ORD(A) NE
ORD(ADASH)$A1(A,B)))..
TS1Q1(A,B,L+1) =G=
TE1Q1(ADASH,B,L) - THOR*(1-
WB1Q1(ADASH,B,L))
;
TC15Q1(A,B,H,L)$ (ORD(L) LT
CARD(L)$ (A1(A,B)$H1(B,H)))..
TS1Q1(A,B,L+1) =G=
TE2Q1(B,H,L) - THOR*(1-
WB2Q1(B,H,L))
;
TC16Q1(A,B,H,L)$ (ORD(L) LT
CARD(L)$ (A1(A,B)$H1(B,H)))..
TS2Q1(B,H,L+1) =G=
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L))
;
TC17Q1(B,H,HDASH,L)$ (ORD(L) LT
CARD(L)$ (ORD(H) NE
ORD(HDASH)$H1(B,H)))..
TS2Q1(B,H,L+1) =G=
TE2Q1(B,HDASH,L) - THOR*(1-
WB2Q1(B,HDASH,L))
;
TC18Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L)) =L=
TS2Q1(B,H,L) + THOR*(1-
WB2Q1(B,H,L))
;
TC19Q1(B,L)..

```

```

      ITOT(B,L-1) +
SUM(A$A1(A,B),FT1Q1(A,B,L)) =E=
      ITOT(B,L) +
SUM(H$H1(B,H),FT2Q1(B,H,L))
;
*- ONEG1Q1(B,L) + OPOS1Q1(B,L)
;
TC20Q1(B)..
      ITOT(B,'L0') =E=
      INITITOT(B)
;
TC21Q1(I,B,L)..
      INVI(I,B,L-1) +
SUM(A$A1(A,B),FL1Q1(I,A,B,L)) =E=
      INVI(I,B,L)
+SUM(H$H1(B,H),FL2Q1(I,B,H,L)) -
ONEG2Q1(I,B,L) + OPOS2Q1(I,B,L)
;
TC22Q1(I,B)..
      INVI(I,B,'L0') =E=
      INITINVI(I,B)
;
TC23Q1(A,B,L)$A1(A,B)..
      FT1Q1(A,B,L) =E=
      SUM(I,FL1Q1(I,A,B,L))
;
TC24Q1(B,H,L)$H1(B,H)..
      FT2Q1(B,H,L) =E=
      SUM(I,FL2Q1(I,B,H,L))
;
TC26Q1(B,L)$ (ORD(L) GT 1)..
      ITOT(B,L-
1)+SUM(A$A1(A,B),FT1Q1(A,B,L)) =L=
      UB11(B)
;
TC27Q1(I,B,L)..
      LB1(I,B)*ITOT(B,L) =L=
      INVI(I,B,L)
* - ONEG1Q1(I,B,L) + OPOS1Q1(I,B,L)
;
;
TC28Q1(I,B,L)..
      INVI(I,B,L) =L=
      UB1(I,B)*ITOT(B,L)
;
TC29Q1(I,B,H,L)$H1(B,H)..
      LB1(I,B)*FT2Q1(B,H,L) =L=
      FL2Q1(I,B,H,L)
;
TC30Q1(I,B,H,L)$H1(B,H)..
      FL2Q1(I,B,H,L) =L=
      UB1(I,B)*FT2Q1(B,H,L)
;
TC31Q1(G)..
SUM(D$D3(D,G),SUM(L,FT2Q1(G,D,L))) =E=
      DM(G)
;
TC32Q1(I,B,H,L)$H1(B,H)..
      ITOT(B,L-1)*FL2Q1(I,B,H,L) +
SUM(A$A1(A,B),FT1Q1(A,B,L))*FL2Q1(I,B,H
,L)) =E=
      INVI(I,B,L-
1)*FT2Q1(B,H,L)+SUM(A$A1(A,B),FL1Q1(I,
A,B,L))*FT2Q1(B,H,L))
;
      TC33Q1(B,H,L)$H1(B,H)..
      SUM(I,ITOT(B,L-
1)*FL2Q1(I,B,H,L)) =E=
      ITOT(B,L-1)*FT2Q1(B,H,L)
;
      TC34Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
SUM(I,FT1Q1(A,B,L)*FL2Q1(I,B,H,L)) =E=
      FT1Q1(A,B,L)*FT2Q1(B,H,L)
;
      TC35Q1(B,H,L)$H1(B,H)..
      SUM(I,INVI(I,B,L-
1)*FT2Q1(B,H,L)) =E=
      ITOT(B,L-1)*FT2Q1(B,H,L)
;
      TC36Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
SUM(I,FL1Q1(I,A,B,L)*FT2Q1(B,H,L)) =E=
      FT1Q1(A,B,L)*FT2Q1(B,H,L)
;
      DC1Q1(D,L)..
      SUM(G$G1(G,D),WB2Q1(G,D,L))
=L=
      1
;
      DC2Q1(G,L)..
      SUM(D$D3(D,G),WB2Q1(G,D,L))
=L=
      1
;
      DC3Q1(D)..
SUM(L,SUM(G$G1(G,D),TE2Q1(G,D,L)-
TS2Q1(G,D,L))) =E=
      THOR
;
      DC4Q1(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
      TS2Q1(G,D,L+1) =G=
      TE2Q1(GDASH,D,L) - THOR*(1-
WB2Q1(GDASH,D,L))
;
      DC5Q1(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
      TS2Q1(G,D,L+1) =L=
      TE2Q1(GDASH,D,L) + THOR*(1-
WB2Q1(GDASH,D,L))
;
      CC1Q1(U,Y,L)$Y1(Y,U)..
      TSI(U) =L=
      TS1Q1(U,Y,L) + THOR*(1-
WB1Q1(U,Y,L))
;
      CC2Q1(U,Y,L)$Y1(Y,U)..
      TEO(U) =G=
      TE1Q1(U,Y,L) - THOR*(1-
WB1Q1(U,Y,L))
;
      CC3Q1(U)..
SUM(L,SUM(Y$Y1(Y,U),FT1Q1(U,Y,L))) =E=
      VCRUDE(U)
;

```

```

CC4Q1 (I, U, Y, L) $Y1 (Y, U) . .
    FL1Q1 (I, U, Y, L) =E=
    FRACT (I, U) * FT1Q1 (U, Y, L)
;

VB1Q1 (I, B, L) . .
    INVI (I, B, L) =L=
    UBI1 (B)
;

VB2Q1 (B, L) . .
    LBI1 (B) =L=
    ITOT (B, L)
;

VB3Q1 (B, L) . .
    ITOT (B, L) =L=
    UBI1 (B)
;

VB4Q1 (I, A, B, L) $A1 (A, B) . .
    FL1Q1 (I, A, B, L) =L=
    UBFV1 (A, B)
;

VB5Q1 (I, B, H, L) $H1 (B, H) . .
    FL2Q1 (I, B, H, L) =L=
    UBFV2 (B, H)
;

VB6Q1 (A, B, L) $A1 (A, B) . .
    FT1Q1 (A, B, L) =L=
    UBFV1 (A, B)
;

VB7Q1 (B, H, L) $H1 (B, H) . .
    FT2Q1 (B, H, L) =L=
    UBFV2 (B, H)
;

VB8Q1 (A, B, L) $A1 (A, B) . .
    TS1Q1 (A, B, L) =L=
    THOR
;

VB9Q1 (A, B, L) $A1 (A, B) . .
    TE1Q1 (A, B, L) =L=
    THOR
;

VB10Q1 (B, H, L) $H1 (B, H) . .
    TS2Q1 (B, H, L) =L=
    THOR
;

VB11Q1 (B, H, L) $H1 (B, H) . .
    TE2Q1 (B, H, L) =L=
    THOR
;

VB12Q1 (U) . .
    TARR (U) =L=
    TSI (U)
;

VB13Q1 (U) . .
    TSI (U) =L=
    THOR
;

VB14Q1 (U) . .
    TARR (U) =L=
    TEO (U)
;

VB15Q1 (U) . .
    TEO (U) =L=
    THOR
;

* ===== FOR DUPLICATING
EQUATIONS AND VARIABLES IN Q2
===== *

TC1Q2 (A, B, L) $A1 (A, B) . .
    FT1Q2 (A, B, L) =L=
    UBFV1 (A, B) * WB1Q2 (A, B, L)
;

TC2Q2 (B, H, L) $H1 (B, H) . .
    FT2Q2 (B, H, L) =L=
    UBFV2 (B, H) * WB2Q2 (B, H, L)
;

TC3Q2 (A, B, L) $A1 (A, B) . .
    UBF1 (A, B) * (TE1Q2 (A, B, L) -
    TS1Q2 (A, B, L)) + UBF1 (A, B) * THOR * (1 -
    WB1Q2 (A, B, L)) =G=
    FT1Q2 (A, B, L)
;

TC4Q2 (B, H, L) $H1 (B, H) . .
    UBF2 (B, H) * (TE2Q2 (B, H, L) -
    TS2Q2 (B, H, L)) + UBF2 (B, H) * THOR * (1 -
    WB2Q2 (B, H, L)) =G=
    FT2Q2 (B, H, L)
;

TC5Q2 (A, Y, L) $A2 (A, Y) . .
    LBF1 (A, Y) * (TE1Q2 (A, Y, L) -
    TS1Q2 (A, Y, L)) - LBF1 (A, Y) * THOR * (1 -
    WB1Q2 (A, Y, L)) =L=
    FT1Q2 (A, Y, L)
;

TC6Q2 (Y, H, L) $H2 (Y, H) . .
    LBF2 (Y, H) * (TE2Q2 (Y, H, L) -
    TS2Q2 (Y, H, L)) - LBF2 (Y, H) * THOR * (1 -
    WB2Q2 (Y, H, L)) =L=
    FT2Q2 (Y, H, L)
;

TC7Q2 (G, H, L) $H3 (G, H) . .
    LBF2 (G, H) * (TE2Q2 (G, H, L) -
    TS2Q2 (G, H, L)) =L=
    FT2Q2 (G, H, L)
;

TC8Q2 (A, B, L) $ (ORD (L) LT
CARD (L) $A1 (A, B)) . .
    TS1Q2 (A, B, L+1) =G=
    TE1Q2 (A, B, L) - THOR * (1 -
    WB1Q2 (A, B, L))
;

TC9Q2 (A, B, L) $ (ORD (L) LT
CARD (L) $A1 (A, B)) . .
    TS1Q2 (A, B, L+1) =G=
    TS1Q2 (A, B, L)
;

TC10Q2 (A, B, L) $ (ORD (L) LT
CARD (L) $A1 (A, B)) . .
    TE1Q2 (A, B, L+1) =G=
    TE1Q2 (A, B, L)
;

TC11Q2 (B, H, L) $ (ORD (L) LT
CARD (L) $H1 (B, H)) . .
    TS2Q2 (B, H, L+1) =G=

```



```

TE2Q2 (B,H,L) - THOR*(1-
WB2Q2 (B,H,L) )
;

TC12Q2 (B,H,L)$(ORD(L) LT
CARD(L)$H1 (B,H) ) . .
TS2Q2 (B,H,L+1) =G=
TS2Q2 (B,H,L)
;

TC13Q2 (B,H,L)$(ORD(L) LT
CARD(L)$H1 (B,H) ) . .
TE2Q2 (B,H,L+1) =G=
TE2Q2 (B,H,L)
;

TC14Q2 (A,ADASH,B,L)$(ORD(L) LT
CARD(L)$(ORD(A) NE
ORD(ADASH)$A1 (A,B) ) . .
TS1Q2 (A,B,L+1) =G=
TE1Q2 (ADASH,B,L) - THOR*(1-
WB1Q2 (ADASH,B,L) )
;

TC15Q2 (A,B,H,L)$(ORD(L) LT
CARD(L)$(A1 (A,B)$H1 (B,H) ) . .
TS1Q2 (A,B,L+1) =G=
TE2Q2 (B,H,L) - THOR*(1-
WB2Q2 (B,H,L) )
;

TC16Q2 (A,B,H,L)$(ORD(L) LT
CARD(L)$(A1 (A,B)$H1 (B,H) ) . .
TS2Q2 (B,H,L+1) =G=
TE1Q2 (A,B,L) - THOR*(1-
WB1Q2 (A,B,L) )
;

TC17Q2 (B,H,HDASH,L)$(ORD(L) LT
CARD(L)$(ORD(H) NE
ORD(HDASH)$H1 (B,H) ) . .
TS2Q2 (B,H,L+1) =G=
TE2Q2 (B,HDASH,L) - THOR*(1-
WB2Q2 (B,HDASH,L) )
;

TC18Q2 (A,B,H,L)$(A1 (A,B)$H1 (B,H) ) . .
TE1Q2 (A,B,L) - THOR*(1-
WB1Q2 (A,B,L) ) =L=
TS2Q2 (B,H,L) + THOR*(1-
WB2Q2 (B,H,L) )
;

TC19Q2 (B,L) . .
ITOT (B,L-1) +
SUM (A$A1 (A,B) , FT1Q2 (A,B,L) ) =E=
ITOT (B,L) +
SUM (H$H1 (B,H) , FT2Q2 (B,H,L) )
;

TC20Q2 (B) . .
ITOT (B, 'L0' ) =E=
INITITOT (B)
;

TC21Q2 (I,B,L) . .
INVI (I,B,L-1) +
SUM (ASA1 (A,B) , FL1Q2 (I,A,B,L) ) =E=
INVI (I,B,L)
+SUM (H$H1 (B,H) , FL2Q2 (I,B,H,L) )
;

TC22Q2 (I,B) . .
INVI (I,B, 'L0' ) =E=
INITINVI (I,B)
;

TC23Q2 (A,B,L)$A1 (A,B) . .
FT1Q2 (A,B,L) =E=
SUM (I, FL1Q2 (I,A,B,L) )
;

TC24Q2 (B,H,L)$H1 (B,H) . .
FT2Q2 (B,H,L) =E=
SUM (I, FL2Q2 (I,B,H,L) )
;

TC26Q2 (B,L)$(ORD(L) GT 1) . .
ITOT (B,L-
1)+SUM (A$A1 (A,B) , FT1Q2 (A,B,L) ) =L=
UB11 (B)
;

TC27Q2 (I,B,L) . .
LB1 (I,B) *ITOT (B,L) =L=
INVI (I,B,L)
;

TC28Q2 (I,B,L) . .
INVI (I,B,L) =L=
UB1 (I,B) *ITOT (B,L)
;

TC29Q2 (I,B,H,L)$H1 (B,H) . .
LB1 (I,B) *FT2Q2 (B,H,L) =L=
FL2Q2 (I,B,H,L)
;

TC30Q2 (I,B,H,L)$H1 (B,H) . .
FL2Q2 (I,B,H,L) =L=
UB1 (I,B) *FT2Q2 (B,H,L)
;

TC31Q2 (G) . .
SUM (D$D3 (D,G) , SUM (L, FT2Q2 (G,D,L) ) ) =E=
DM (G)
;

TC32Q2 (I,B,H,L)$H1 (B,H) . .
ITOT (B,L-1) *FL2Q2 (I,B,H,L) +
SUM (A$A1 (A,B) , FT1Q2 (A,B,L) *FL2Q2 (I,B,H
,L) ) =E=
INVI (I,B,L-
1) *FT2Q2 (B,H,L) +SUM (A$A1 (A,B) , FL1Q2 (I,
A,B,L) *FT2Q2 (B,H,L) )
;

TC33Q2 (B,H,L)$H1 (B,H) . .
SUM (I, ITOT (B,L-
1) *FL2Q2 (I,B,H,L) ) =E=
ITOT (B,L-1) *FT2Q2 (B,H,L)
;

TC34Q2 (A,B,H,L)$(A1 (A,B)$H1 (B,H) ) . .
SUM (I, FT1Q2 (A,B,L) *FL2Q2 (I,B,H,L) ) =E=
FT1Q2 (A,B,L) *FT2Q2 (B,H,L)
;

TC35Q2 (B,H,L)$H1 (B,H) . .
SUM (I, INVI (I,B,L-
1) *FT2Q2 (B,H,L) ) =E=
ITOT (B,L-1) *FT2Q2 (B,H,L)
;

TC36Q2 (A,B,H,L)$(A1 (A,B)$H1 (B,H) ) . .
SUM (I, FL1Q2 (I,A,B,L) *FT2Q2 (B,H,L) ) =E=
FT1Q2 (A,B,L) *FT2Q2 (B,H,L)
;

DC1Q2 (D,L) . .

```

```

SUM(G$G1(G,D),WB2Q2(G,D,L))
=L=
1
;
DC2Q2(G,L)..
SUM(D$D3(D,G),WB2Q2(G,D,L))
=L=
1
;
DC3Q2(D)..
SUM(L,SUM(G$G1(G,D),TE2Q2(G,D,L))-
TS2Q2(G,D,L)) =E=
THOR
;
DC4Q2(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
TS2Q2(G,D,L+1) =G=
TE2Q2(GDASH,D,L) - THOR*(1-
WB2Q2(GDASH,D,L))
;
DC5Q2(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
TS2Q2(G,D,L+1) =L=
TE2Q2(GDASH,D,L) + THOR*(1-
WB2Q2(GDASH,D,L))
;
CC1Q2(U,Y,L)$Y1(Y,U)..
TSI(U) =L=
TS1Q2(U,Y,L) + THOR*(1-
WB1Q2(U,Y,L))
;
CC2Q2(U,Y,L)$Y1(Y,U)..
TEO(U) =G=
TE1Q2(U,Y,L) - THOR*(1-
WB1Q2(U,Y,L))
;
CC3Q2(U)..
SUM(L,SUM(Y$Y1(Y,U),FT1Q2(U,Y,L))) =E=
VCRUDE(U)
;
CC4Q2(I,U,Y,L)$Y1(Y,U)..
FL1Q2(I,U,Y,L) =E=
FRACT(I,U)*FT1Q2(U,Y,L)
;
VB1Q2(I,B,L)..
INVI(I,B,L) =L=
UBI1(B)
;
VB2Q2(B,L)..
LBI1(B) =L=
ITOT(B,L)
;
VB3Q2(B,L)..
ITOT(B,L) =L=
UBI1(B)
;
VB4Q2(I,A,B,L)$A1(A,B)..
FL1Q2(I,A,B,L) =L=
UBFV1(A,B)
;
VB5Q2(I,B,H,L)$H1(B,H)..
FL2Q2(I,B,H,L) =L=
UBFV2(B,H)
;
VB6Q2(A,B,L)$A1(A,B)..
FT1Q2(A,B,L) =L=
UBFV1(A,B)
;
VB7Q2(B,H,L)$H1(B,H)..
FT2Q2(B,H,L) =L=
UBFV2(B,H)
;
VB8Q2(A,B,L)$A1(A,B)..
TS1Q2(A,B,L) =L=
THOR
;
VB9Q2(A,B,L)$A1(A,B)..
TE1Q2(A,B,L) =L=
THOR
;
VB10Q2(B,H,L)$H1(B,H)..
TS2Q2(B,H,L) =L=
THOR
;
VB11Q2(B,H,L)$H1(B,H)..
TE2Q2(B,H,L) =L=
THOR
;
VB12Q2(U)..
TARR(U) =L=
TSI(U)
;
VB13Q2(U)..
TSI(U) =L=
THOR
;
VB14Q2(U)..
TARR(U) =L=
TEO(U)
;
VB15Q2(U)..
TEO(U) =L=
THOR
;
***** BRANCH AND
CONTRACT ALGORITHM
*****
MODEL ACOTAVARS
/
OBJFUN
TC1
TC2
*TC3
TC4
TC5
TC6
TC7
TC8
TC9
TC10
TC11
TC12
TC13

```

TC14	ITOTLIBRECT3L1
TC15	ITOTLIBRECT3L2
TC16	/
TC17	
TC18	MODEL MR00
*TC19	
TC20	/
*TC21	OBJFUN
TC22	TC1
*TC23	TC2
*TC24	*TC3
	TC4
TC26	TC5
*TC27	TC6
TC28	TC7
TC29	TC8
TC30	TC9
TC31	TC10
*DC1	TC11
*DC2	TC12
DC3	TC13
DC4	TC14
DC5	TC15
CC1	TC16
CC2	TC17
CC3	TC18
CC4	*TC19
VB1	TC20
VB2	*TC21
VB3	TC22
VB4	*TC23
VB5	*TC24
VB6	
VB7	TC26
VB8	*TC27
VB9	TC28
VB10	TC29
VB11	TC30
VB12	TC31
VB13	
VB14	*DC1
VB15	*DC2
LBP1	DC3
LBP2	DC4
LBP3	DC5
*LBP4	CC1
*LBP5	CC2
*LBP6	CC3
LBP7	CC4
*LBP8	VB1
LBP9	VB2
LBP10	VB3
LBP11	VB4
LBP12	VB5
*LBP13	VB6
*LBP14	VB7
LBP15	VB8
LBP16	VB9
LBP17	VB10
COTAOF	VB11
ITOTLIBREST1L0	VB12
ITOTLIBREST1L1	VB13
ITOTLIBREST1L2	VB14
ITOTLIBREST2L0	VB15
ITOTLIBREST2L1	LBP1
ITOTLIBREST2L2	LBP2
ITOTLIBREST3L0	LBP3
ITOTLIBREST3L1	*LBP4
ITOTLIBREST3L2	*LBP5
ITOTLIBRECT1L0	*LBP6
ITOTLIBRECT1L1	LBP7
ITOTLIBRECT1L2	*LBP8
ITOTLIBRECT2L0	LBP9
ITOTLIBRECT2L1	LBP10
ITOTLIBRECT2L2	LBP11
ITOTLIBRECT3L0	LBP12

```

*LBP13
*LBP14
LBP15
LBP16
LBP17
/

MODEL MF00
/
TCUB1
TCUB2
*TCUB3
TCUB4
TCUB5
TCUB6
TC7
TCUB8
TC9
TC10
TCUB11
TC12
TC13
TCUB14
TCUB15
TCUB16
TCUB17
TCUB18
*TC19
TC20
*TC21
TC22
*TC23
*TC24

TC26
*TC27
TC28
TC29
TC30
TC31
*DCUB1
*DCUB2
DC3
DCUB4
DCUB5
CCUB1
CCUB2
CC3
CC4
VB1
VB2
VB3
VB4
VB5
VB6
VB7
VB8
VB9
VB10
VB11
VB12
VB13
VB14
VB15
OBJFIXBIN
/

MODEL MINLPOUB
/
TC1
TC2
TC3
TC4
TC5
TC6

TC7
TC8
TC9
TC10
TC11
TC12
TC13
TC14
TC15
TC16
TC17
TC18
*TC19
TC20
*TC21

TC7
TC8
TC9
TC10
TC11
TC12
TC13
TC14
TC15
TC16
TC17
TC18
TC19
TC20
TC21
TC22
TC23
TC24
TC25
TC26
TC27
TC28
TC29
TC30
TC31
DC1
DC2
DC3
DC4
DC5
CC1
CC2
CC3
CC4
VB1
VB2
VB3
VB4
VB5
VB6
VB7
VB8
VB9
VB10
VB11
VB12
VB13
VB14
VB15
OBJFUN
/

;

MODEL LBMIP
/
TC1
TC2
*TC3
TC4
TC5
TC6
TC7
TC8
TC9
TC10
TC11
TC12
TC13
TC14
TC15
TC16
TC17
TC18
*TC19
TC20
*TC21

```

TC22	TC17Q1
*TC23	TC18Q1
*TC24	TC19Q1
	TC20Q1
TC26	*TC21Q1
*TC27	TC22Q1
TC28	TC23Q1
TC29	TC24Q1
TC30	
TC31	TC26Q1
*DC1	*TC27Q1
*DC2	TC28Q1
DC3	TC29Q1
DC4	TC30Q1
DC5	TC31Q1
CC1	DC1Q1
CC2	DC2Q1
CC3	DC3Q1
CC4	DC4Q1
VB1	DC5Q1
VB2	CC1Q1
VB3	CC2Q1
VB4	CC3Q1
VB5	CC4Q1
VB6	VB1Q1
VB7	VB2Q1
VB8	VB3Q1
VB9	VB4Q1
VB10	VB5Q1
VB11	VB6Q1
VB12	VB7Q1
VB13	VB8Q1
VB14	VB9Q1
VB15	VB10Q1
LBP1	VB11Q1
LBP2	VB12Q1
LBP3	VB13Q1
*LBP4	VB14Q1
*LBP5	VB15Q1
*LBP6	
LBP7	
*LBP8	LBP1Q1
LBP9	LBP2Q1
LBP10	LBP3Q1
LBP11	*LBP4Q1
LBP12	*LBP5Q1
*LBP13	
	*LBP6Q1
*LBP14	LBP7Q1
LBP15	LBP8Q1
LBP16	LBP9Q1
LBP17	LBP10Q1
OBJFUN	LBP11Q1
/	LBP12Q1
;	*LBP13Q1
MODEL CRUDE_LB1	LBP14Q1
/	LBP15Q1
TC1Q1	LBP16Q1
TC2Q1	LBP17Q1
TC3Q1	LBP18
TC4Q1	LBP19
TC5Q1	LBP20
TC6Q1	LBP21
TC7Q1	LBP22
TC8Q1	*OBJF
TC9Q1	OBJQ1
TC10Q1	*OBJQ2
TC11Q1	VLC1
TC12Q1	VLC2
TC13Q1	
TC14Q1	
TC15Q1	
TC16Q1	
	MODEL CRUDE_LB2
	/

TC1Q2	LBP19
TC2Q2	LBP20
TC3Q2	LBP21
TC4Q2	LBP22
TC5Q2	*OBJF
TC6Q2	*OBJQ1
TC7Q2	OBJQ2
TC8Q2	VLC1
TC9Q2	VLC2
TC10Q2	/
TC11Q2	;
TC12Q2	
TC13Q2	
TC14Q2	
TC15Q2	MODEL CRUDE_UB
TC16Q2	TCUB1
TC17Q2	TCUB2
TC18Q2	TCUB3
*TC19Q2	TCUB4
TC20Q2	TCUB5
*TC21Q2	TCUB6
TC22Q2	TC7
TC23Q2	TCUB8
TC24Q2	TC9
	TC10
TC26Q2	TCUB11
*TC27Q2	TC12
TC28Q2	TC13
TC29Q2	TCUB14
TC30Q2	TCUB15
TC31Q2	TCUB16
DC1Q2	TCUB17
DC2Q2	TCUB18
DC3Q2	TC19
DC4Q2	TC20
DC5Q2	TC21
CC1Q2	TC22
CC2Q2	TC23
CC3Q2	TC24
CC4Q2	
VB1Q2	TC26
VB2Q2	TC27
VB3Q2	TC28
VB4Q2	TC29
VB5Q2	TC30
VB6Q2	TC31
VB7Q2	DCUB1
VB8Q2	DCUB2
VB9Q2	DC3
VB10Q2	DCUB4
VB11Q2	DCUB5
VB12Q2	CCUB1
VB13Q2	CCUB2
VB14Q2	CC3
VB15Q2	CC4
	VB1
LBP1Q2	VB2
LBP2Q2	VB3
LBP3Q2	VB4
*LBP4Q2	VB5
*LBP5Q2	VB6
*LBP6Q2	VB7
LBP7Q2	VB8
LBP8Q2	VB9
LBP9Q2	VB10
LBP10Q2	VB11
LBP11Q2	VB12
LBP12Q2	VB13
	VB14
*LBP13Q2	VB15
	OBJFIXBIN
LBP14Q2	/
LBP15Q2	
LBP16Q2	;
LBP17Q2	
LBP18	

```

ZZUB.FX = INF;

***** BOUNDS FOR BRANCH
AND CONTRACT ALGORITHM
*****

ITOT.LO(B,L) = 0;          ITOT.UP(B,L)
= 100;

LOOP( (B,L),
  ITOTINI.LO(B,L) = ITOT.LO(B,L);
  ITOTINI.UP(B,L) = ITOT.UP(B,L);
  DELINI.ITOT(B,L) = ITOTINI.UP(B,L) -
  ITOTINI.LO(B,L);
);

FILE RESCRUBND / CBOUND.RES /;

PUT RESCRUBND;

PUT //

LOOP( (B,L),
  PUT / "ITOT.LO('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.LO(B,L):18:12";"
"          ITOT.UP('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.UP(B,L):18:12";"
);

*-----
*-----
*-----
Solution to model MIP

SOLVE MR00 minimizing ZZ using mip;

*-----
*-----
*-----
creating file with results

PUT RESCRU;

PUT /// "-----"
-----";
PUT / "NEXT THE SOLUTION OF THE
RELAXED PROBLEM (MIP)"
PUT / "-----"
-----";

PUT //
LOOP( (B,L),
  PUT / "ITOT.l('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.l(B,L):18:12";"
);

PUT / "ZZ.l=" ZZ.l:20:12 ";";

*-----

PUT /// "LP solution time : "
MR00.resusd:10:5;
PUT / "Model Status: "
PUT$(MR00.modelstat eq 1) "Optimal.";
PUT$(MR00.modelstat eq 2) "Locally
optimal.";
PUT$(MR00.modelstat eq 3)
"Unbounded.";
PUT$(MR00.modelstat eq 4)
"Infeasible.";

PUT$(MR00.modelstat eq 5) "Locally
infeasible.";
PUT$(MR00.modelstat eq 6)
"Intermediate infeasible.";
PUT$(MR00.modelstat eq 7)
"Intermediate nonoptimal.";
PUT$(MR00.modelstat eq 8) "Integer
solution.";
PUT$(MR00.modelstat eq 9)
"Intermediate non-integer.";
PUT$(MR00.modelstat eq 10) "Integer
infeasible.";
PUT$(MR00.modelstat eq 12) "Error
unknown.";
PUT$(MR00.modelstat eq 13) "Error no
solution.";

*-----
*-----
*-----
*NEXT USED LEVELS ARE REPORTED

PUT //
LOOP( (B,L),
  PUT / "ITOT.LO('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.LO(B,L):18:12";"
"          ITOT.UP('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.UP(B,L):18:12";"
);

*-----
*-----
*-----
*MULTIPLICATION

PUT //
LOOP( (B,L),
  PUT / "ITOT.M('
"B.TL:3, ", ", L.TL:3, "' )=
", ITOT.M(B,L):18:12";"
);

IF(MR00.MODELSTAT EQ 1,
  LOOP( (B,L),
    ITOTBEST.L(B,L) = ITOT.L(B,L);
  );
);

***** TO FIX BINARY
VARIABLES TO PARAMETER VALUES
*****

PUT RESCRU;

PUT //
PUT / "-----"
-----"
PUT / "BINARY VARIABLES OBTAINED FROM
MIP-LB"
PUT / "-----"
-----"

PUT //
LOOP( (A,B) $A1 (A,B),
  LOOP( L,
    WBUB1(A,B,L) = WB1.L(A,B,L)
    PUT "WB1('
"A.TL:3, ", ", B.TL:3, ", ", L.TL:3, " ' )=
", WBUB1(A,B,L):18:12;
    PUT " ";
  );
  PUT /

```

```

);
PUT /
LOOP( (B,H)$H1(B,H),
  LOOP( L,
    WBUB2(B,H,L) = WB2.L(B,H,L)
    PUT "WB2('
"B.TL:3," ", "H.TL:3," ", "L.TL:3," ')=
",WBUB2(B,H,L):18:12;
    PUT " "; "
  );
  PUT /
);

LOOP( (A,B,L)$A1(A,B),
  WB1.FX(A,B,L) = WB1.L(A,B,L);
);

LOOP( (B,H,L)$H1(B,H),
  WB2.FX(B,H,L) = WB2.L(B,H,L);
);

*-----
Solution to NLP non convex model (UB)

SOLVE MF00 MINIMIZING ZZFIXBIN USING
NLP;

*-----
creating file with results

PUT RESCRU;

PUT /// "-----";
PUT / "NEXT THE SOLUTION OF THE
NONCONVEX PROBLEM (UB)"
PUT / "-----";

PUT //
PUT$(MF00.modelstat eq 2 and
ZZFIXBIN.1 lt ZZUB.1)
"*****This solution improves
current upper bound value to ";
ZZUB.fx$(MF00.modelstat eq 2 and
ZZFIXBIN.1 lt ZZUB.1)=ZZFIXBIN.1;
PUT // " ZZUB =
",ZZUB.1:20:12;

PUT //
LOOP( (B,L),
  PUT /"ITOT.L('
"B.TL:3," ", "L.TL:3," ')=
",ITOT.L(B,L):18:12";
);

PUT / "ZZFIXBIN.1=" ZZFIXBIN.1:20:12
";";

PUT /// "Nonconvex NLP solution time :
" MF00.resusd:10:5;
PUT / "Model Status: "
PUT$(MF00.modelstat eq 1) "Optimal.";
PUT$(MF00.modelstat eq 2) "Locally
optimal.";
PUT$(MF00.modelstat eq 3)
"Unbounded.";
PUT$(MF00.modelstat eq 4)
"Infeasible.";
PUT$(MF00.modelstat eq 5) "Locally
infeasible.";
PUT$(MF00.modelstat eq 6)
"Intermediate infeasible.";

PUT$(MF00.modelstat eq 7)
"Intermediate nonoptimal.";
PUT$(MF00.modelstat eq 8) "Integer
solution.";
PUT$(MF00.modelstat eq 9)
"Intermediate non-integer.";
PUT$(MF00.modelstat eq 10) "Integer
infeasible.";
PUT$(MF00.modelstat eq 12) "Error
unknown.";
PUT$(MF00.modelstat eq 13) "Error no
solution.";

*NEXT USED LEVELS ARE REPORTED

PUT //
LOOP( (B,L),
  PUT / "ITOT.LO('
"B.TL:3," ", "L.TL:3," ')=
",ITOT.LO(B,L):18:12";
"
ITOT.UP('
"B.TL:3," ", "L.TL:3," ')=
",ITOT.UP(B,L):18:12";
);

PUT // "-----";
PUT ///

IF(MF00.MODELSTAT EQ 2,
  LOOP((B,L),
    ITOTBEST.L(B,L) = ITOT.L(B,L);
  );
);

IF(MF00.MODELSTAT EQ 10,
  LOOP((B,L),
    ITOTBEST.L(B,L) = ITOT.L(B,L);
  );
);

*DEFINE VARIABLES AND PARAMETERS FOR
FEASIBILITY BASED TESTS

*-----SE ESPECIFICA ARCHIVO
PARA RESULTADOS (SPECIFIC FILES FOR
RESULTS)

PUT RESCRU;

*-----ESPECIFICA EL VALOR DE LA
MEJOR COTA SUPERIOR DISPONIBLE
(SPECIFIC THE VALUE OF THE BEST LEVEL
SUPERIOR AVAILABLE)

LBEST.L=-INF;

*-----NUMERO MAXIMO
DE SUBPROBLEMAS A RESOLVER (NUMBER OF
MAXIMUM SUBPROBLEMS TO SOLVE)
SET ITERATIONS /1*500/;

*-----
INICIALIZA LAS MAXIMAS DISTANCIAS
(INITIALIZES PRINCIPLE DISTANCES)
MAXDIS.L=100000.0;
MAXPCDIS.L=100000.0;

*-----INICIALIZA
LAS REDUCCIONES ALCANZADAS
(INITIALIZES THE REACHED REDUCTIONS)

```



```

ABSGAP.L=100.0;
RELGAP.L=100.0;

AUXABS.L=100.0;
AUXREL.L=100.0;

*-----INDICA QUE EL PROBLEMA
ES FACTIBLE Y TIENE OPTIMO (INDICATES
PROBLEM IS FEASIBLE AND HAS OPTIMUM)
INFACIBLE.L=0;
ACOTAVARS.MODELSTAT=1;

*-----
INICIALIZA TIEMPO TOTAL DE COMPUTO
(INITIALIZES TOTAL TIME OF
CALCULATIONS)
TIEMPOTOT.L=0.00000000000;

*-----INICIALIZA CONTADOR DEL
NUMERO DE SUBPROBLEMAS RESUELTOS
(INITIALIZES ACCOUNTANT FOR NUMBER OF
RESOLUTE SUBPROBS)
CONTADOR.L=0;

FLAGLOITOT(B,L)=1;
FLAGUPITOT(B,L)=1;

LOOP(ITERATIONS $(CARD(SBLUEITOT) NE 0
AND ABSGAP.L GT EPSILONABS AND
RELGAP.L GT EPSILONABS),

  PUT ///
  PUT / "-----"
  -----";
  PUT / "REDUCTION. IT INITIATES
GREATER ITERATION NUMBERS:
"ORD(ITERATIONS):5:0
  PUT / "-----"
  -----";

  ITOT.FX(B,L)$( ITOT.UP(B,L) -
ITOT.LO(B,L) ) LT
EPSILONABS)=ITOTBEST.L(B,L);
  SGREENITOT(B,L)$( ITOTBEST.L(B,L) -
ITOT.LO(B,L) LT EPSILONABS)=YES;
  SBLACKITOT(B,L)$( ITOT.UP(B,L) -
ITOTBEST.L(B,L) LT EPSILONABS)=YES;
  ITOT.FX(B,L)$(SGREENITOT(B,L) AND
SBLACKITOT(B,L))=ITOTBEST.L(B,L);

  LOOP( (B,L),
FLAGLOITOT(B,L)$(SGREENITOT(B,L))=0;
FLAGUPITOT(B,L)$(SBLACKITOT(B,L))=0;

  IF(FLAGLOITOT(B,L) EQ 0 AND
FLAGUPITOT(B,L) EQ 0,
  SBLUEITOT(B,L)=NO;
  SREDITOT(B,L)=YES;
  IF(SGREENITOT(B,L) AND
SBLACKITOT(B,L),
  SREDITOT(B,L)=NO;
  );
  );
);

AUXREL.L=100.0;

*-----
PUT /

PUT /"NUMBER OF ELEMENTS IN SBLUE:
"CARD(SBLUEITOT):5:0 /;
LOOP( (B,L) $(SBLUEITOT(B,L)),
  PUT / B.TL:10,L.TL:10
);

PUT /
PUT /"NUMBER OF ELEMENTS IN SRED:
"CARD(SREDITOT):5:0 /;
LOOP( (B,L) $(SREDITOT(B,L)),
  PUT / B.TL:10,L.TL:10
);

PUT /
PUT /"NUMBER OF ELEMENTS IN SGREEN:
"CARD(SGREENITOT):5:0 /;
LOOP( (B,L) $(SGREENITOT(B,L)),
  PUT / B.TL:10,L.TL:10
);

PUT /
PUT /"NUMBER OF ELEMENTS IN SBLACK:
"CARD(SBLACKITOT):5:0;
LOOP( (B,L) $(SBLACKITOT(B,L)),
  PUT / B.TL:10,L.TL:10
);

PUT /;

*-----ALMACENA COTAS ANTES DE
REDUCIR (STORES LEVELS BEFORE
REDUCING)
LOOP( (B,L),
  ITOTOLD.LO(B,L)=ITOT.LO(B,L);
  ITOTOLD.UP(B,L)=ITOT.UP(B,L);
);

*-----CALCULA DISTANCIA DESDE
LAS COTAS HACIA LA MEJOR SOLUCION
(CALCULATE DISTANCE FROM LEVELS
TOWARDS BEST SOLUTION)

LOOP( (B,L),
  DIFABSITOT(B,L)=ITOT.UP(B,L) -
ITOT.LO(B,L);
  DIFABSITOTUP(B,L)=ITOT.UP(B,L) -
ITOTBEST.L(B,L);
  DIFABSITOTLO(B,L)=ITOTBEST.L(B,L) -
ITOT.LO(B,L);

  DIFRELITOT(B,L)=ZERO;
  DIFRELITOTUP(B,L)=ZERO;
  DIFRELITOTLO(B,L)=ZERO;
  DIFRELITOT(B,L)$(DELINIITOT(B,L) NE
0)=100*DIFABSITOT(B,L)/DELINIITOT(B,L)
;
  DIFRELITOTUP(B,L)$(DELINIITOT(B,L)
NE
0)=100*DIFABSITOTUP(B,L)/DELINIITOT(B,
L);
  DIFRELITOTLO(B,L)$(DELINIITOT(B,L)
NE
0)=100*DIFABSITOTLO(B,L)/DELINIITOT(B,
L);
);

MAXABSDIS.L=0.0;
LOOP(SBLUEITOT,

MAXABSDIS.L=MAX [MAXABSDIS.L,DIFABSITOT
LO(SBLUEITOT),DIFABSITOTUP(SBLUEITOT)]
);

MAXPCDIS.L=0.0;
LOOP(SBLUEITOT,

```

```

      MAXPCDIS.L$(FLAGLOITOT(SBLUEITOT)
EQ
1)=MAX[ MAXPCDIS.L, DIFRELITOTLO(SBLUEIT
OT) ];
      MAXPCDIS.L$(FLAGUPITOT(SBLUEITOT)
EQ
1)=MAX[ MAXPCDIS.L, DIFRELITOTUP(SBLUEIT
OT) ];
    );
*-----

PUT //
LOOP( (N,T),
PUT /"ITOT.LO("N.TL:3,"",T.TL:3,"")=
",ITOT.LO(N,T):18:12";"
"      ITOT.UP("N.TL:3,"",T.TL:3,"")=
",ITOT.UP(N,T):18:12";"
);

PUT //
LOOP(SBLUEITOT(B,L),
PUT
/"DIFABSITOTLO("B.TL:3,"",L.TL:3,"")
= ",DIFABSITOTLO(SBLUEITOT):18:12";"
"
DIFABSITOTUP("B.TL:3,"",L.TL:3,"")=
",DIFABSITOTUP(SBLUEITOT):18:12";" /
"
DIFABSITOT("B.TL:3,"",L.TL:3,"")=
",DIFABSITOT(SBLUEITOT):18:12";"
);
PUT / "DISTANCE ABSOLUTE
PRINCIPLE:"MAXABSDIS.L:18:12;

PUT //
LOOP(SBLUEITOT(B,L),
PUT
/"DIFRELITOTLO("B.TL:3,"",L.TL:3,"")
= ",DIFRELITOTLO(SBLUEITOT):18:12";"
"
DIFRELITOTUP("B.TL:3,"",L.TL:3,"")=
",DIFRELITOTUP(SBLUEITOT):18:12";" /
"
DIFRELITOT("B.TL:3,"",L.TL:3,"")=
",DIFRELITOT(SBLUEITOT):18:12";"
);
PUT / "DISTANCE PERCENTAGE
PRINCIPLE:"MAXPCDIS.L:18:12;
*-----

SOLOUNO=0;
LOOP( (N,T) $(SBLUEITOT(N,T) AND
MAXPCDIS.L GT EPSILONREL AND SOLOUNO
EQ 0 AND
      (MAXPCDIS.L EQ
DIFRELITOTLO(N,T) OR MAXPCDIS.L EQ
DIFRELITOTUP(N,T)) ),
  CONTADOR.L=CONTADOR.L+1;
  PUT // "SUBPROBLEM
NUMBER:"CONTADOR.L:5:0;
  PUT / "LEVELS OF ITOT ARE
UPDATED("N.TL:3,"",T.TL:3,""
      " FROM:" ITOT.LO(N,T):18:12"--
---"ITOT.UP(N,T):18:12;

*===== STORING TANKS
=====

*-----ST1-----
-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L0') AND MAXPCDIS.L
EQ DIFRELITOT('ST1','L0'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTST1LO USING RMIP;
  AUXABS.L=FREEITOTST1LO.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST1LO.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST1','L0'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTST1LO USING RMIP;
  AUXABS.L=ITOT.UP(N,T)-
FREEITOTST1LO.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST1LO.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L1') AND MAXPCDIS.L
EQ DIFRELITOT('ST1','L1'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTST1L1 USING RMIP;
  AUXABS.L=FREEITOTST1L1.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST1L1.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L1') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST1','L1'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTST1L1 USING RMIP;
  AUXABS.L=ITOT.UP(N,T)-
FREEITOTST1L1.L;

```

```

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST1L1.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L2') AND MAXPCDIS.L
EQ DIFRELITOT('ST1','L2'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTST1L2 USING RMIP;
  AUXABS.L=FREEITOTST1L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST1L2.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGLOITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST1','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST1','L2'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTST1L2 USING RMIP;
  AUXABS.L=ITOT.UP(N,T) -
FREEITOTST1L2.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST1L2.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----
*-----ST2-----
----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L0') AND MAXPCDIS.L
EQ DIFRELITOT('ST2','L0'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTST2L0 USING RMIP;
  AUXABS.L=FREEITOTST2L0.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST2L0.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST2','L0'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTST2L0 USING RMIP;
  AUXABS.L=ITOT.UP(N,T) -
FREEITOTST2L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST2L0.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGLOITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L1') AND MAXPCDIS.L
EQ DIFRELITOT('ST2','L1'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTST2L1 USING RMIP;
  AUXABS.L=FREEITOTST2L1.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST2L1.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGLOITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L1') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST2','L1'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTST2L1 USING RMIP;
  AUXABS.L=ITOT.UP(N,T) -
FREEITOTST2L1.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST2L1.L];

  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

```

```

);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L2') AND MAXPCDIS.L
EQ DIFRELITOT('ST2','L2'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTST2L2 USING RMIP;
AUXABS.L=FREEITOTST2L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST2L2.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST2','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST2','L2'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTST2L2 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTST2L2.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST2L2.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);
*-----

*-----ST3-----
-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L0') AND MAXPCDIS.L
EQ DIFRELITOT('ST3','L0'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTST3L0 USING RMIP;
AUXABS.L=FREEITOTST3L0.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST3L0.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST3','L0'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTST3L0 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTST3L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST3L0.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);
*-----

SOLVE ACOTAVARS MAXIMIZING
FREEITOTST3L0 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTST3L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST3L0.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L1') AND MAXPCDIS.L
EQ DIFRELITOT('ST3','L1'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTST3L1 USING RMIP;
AUXABS.L=FREEITOTST3L1.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST3L1.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L1') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST3','L1'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTST3L1 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTST3L1.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST3L1.L];

AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L2') AND MAXPCDIS.L
EQ DIFRELITOT('ST3','L2'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTST3L2 USING RMIP;
AUXABS.L=FREEITOTST3L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
ST3L2.L];

```

```

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('ST3','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('ST3','L2'),
    SOLVE ACOTAVARS MAXIMIZING
FREEITOTST3L2 USING RMIP;
    AUXABS.L=ITOT.UP(N,T) -
FREEITOTST3L2.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
ST3L2.L];

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);
*-----
*==== CHARGING TANKS
=====
*-----CT1-----
----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT1','L0') AND MAXPCDIS.L
EQ DIFRELITOT('CT1','L0'),
    SOLVE ACOTAVARS MINIMIZING
FREEITOTCT1L0 USING RMIP;
    AUXABS.L=FREEITOTCT1L0.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT1L0.L];

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT1','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT1','L0'),
    SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT1L0 USING RMIP;
    AUXABS.L=ITOT.UP(N,T) -
FREEITOTCT1L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT1L0.L];

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT1','L1') AND MAXPCDIS.L
EQ DIFRELITOT('CT1','L1'),
    SOLVE ACOTAVARS MINIMIZING
FREEITOTCT1L1 USING RMIP;
    AUXABS.L=ITOT.UP(N,T) -
FREEITOTCT1L1.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT1L1.L];

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);
*-----
*-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT1','L2') AND MAXPCDIS.L
EQ DIFRELITOT('CT1','L2'),
    SOLVE ACOTAVARS MINIMIZING
FREEITOTCT1L2 USING RMIP;
    AUXABS.L=FREEITOTCT1L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT1L2.L];

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);

```



```

);
IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT1','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT1','L2'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT1L2 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTCT1L2.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT1L2.L];

AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);
*-----
*-----CT2-----
----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT2','L0') AND MAXPCDIS.L
EQ DIFRELITOT('CT2','L0'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTCT2L0 USING RMIP;
AUXABS.L=FREEITOTCT2L0.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT2L0.L];

AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT2','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT2','L0'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT2L0 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTCT2L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT2L0.L];

AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT2','L2') AND MAXPCDIS.L
EQ DIFRELITOT('CT2','L2'),
SOLVE ACOTAVARS MINIMIZING
FREEITOTCT2L2 USING RMIP;
AUXABS.L=FREEITOTCT2L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT2L2.L];

AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T));
AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
FLAGLOITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT2','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT2','L2'),
SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT2L2 USING RMIP;
AUXABS.L=ITOT.UP(N,T)-
FREEITOTCT2L2.L;

```

```

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT2L2.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);
*-----

*-----CT3-----
-----

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT3','L0') AND MAXPCDIS.L
EQ DIFRELITOT('CT3','L0'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTCT3L0 USING RMIP;
  AUXABS.L=FREEITOTCT3L0.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT3L0.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGLOITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT3','L0') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT3','L0'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT3L0 USING RMIP;
  AUXABS.L=ITOT.UP(N,T)-
FREEITOTCT3L0.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT3L0.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGUPITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT3','L2') AND MAXPCDIS.L
EQ DIFRELITOT('CT3','L2'),
  SOLVE ACOTAVARS MINIMIZING
FREEITOTCT3L2 USING RMIP;
  AUXABS.L=FREEITOTCT3L2.L-
ITOT.LO(N,T);

ITOT.LO(N,T)=MAX[ITOT.LO(N,T),FREEITOT
CT3L2.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));
  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) EQ 0)=0.00000000000;
  FLAGLOITOT(N,T)$ (AUXREL.L LT
MINREDREL)=0;
  SOLOUNO=1;
);

IF(SOLOUNO EQ 0 AND
SBLUEITOT('CT3','L2') AND MAXPCDIS.L
EQ DIFRELITOTUP('CT3','L2'),
  SOLVE ACOTAVARS MAXIMIZING
FREEITOTCT3L2 USING RMIP;
  AUXABS.L=ITOT.UP(N,T)-
FREEITOTCT3L2.L;

ITOT.UP(N,T)=MIN[ITOT.UP(N,T),FREEITOT
CT3L2.L];

  AUXREL.L$( (ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T)) NE
0)=100*AUXABS.L/(ITOTOLD.UP(N,T)-
ITOTOLD.LO(N,T));

```



```

    AUXREL.L$( (ITOTOLD.UP(N,T) -
ITOTOLD.LO(N,T)) EQ 0)=0.000000000000;
    FLAGUPITOT(N,T)$(AUXREL.L LT
MINREDREL)=0;
    SOLOUNO=1;
);
*-----

    PUT / "
HASTA:"
        ITOT.LO(N,T):18:12"-----
"ITOT.UP(N,T):18:12;
    PUT // "          RED ABS:"
AUXABS.L:18:12;
    PUT / "          -----RED %:"
AUXREL.L:18:12;
    PUT / "          MODEL STATUS:"
    PUT$(ACOTAVARS.MODELSTAT EQ 1) "
OPTIMAL";

TIEMPOTOT.L=TIEMPOTOT.L+ACOTAVARS.RESU
SD;
    PUT / "          SUBPROBLEM SOL. TIME:
" ACOTAVARS.RESU:10:5;
    PUT / "          ACCUMULATED SOL. TIME:
" TIEMPOTOT.L:10:5;

    IF( ACOTAVARS.MODELSTAT NE 1,
    PUT // "STOP. SUBPROBLEM SOLUCION IS
NOT OPTIMAL.";
    PUT / "ACOTAVARS MODEL STATUS:
"ACOTAVARS.MODELSTAT;
    ABORT "STOP. SUBPROBLEM SOLUCION IS
NOT OPTIMAL."; );

    PUT / "
MULTPLICADOR=" COTAOF.M:18:12;
    LBEST.L$(COTAOF.M NE
0)=MAX[LBEST.L,ZZUB.L-(ITOT.UP(N,T)-
ITOT.LO(N,T))/ABS(COTAOF.M)];
    PUT / "
LBEST.L=" LBEST.L:18:12;
    ABSGAP.L=ZZUB.L-LBEST.L;
    RELGAP.L=ABS[100*ABSGAP.L/ZZUB.L];
    PUT / "          ABSOLUTE GAP (UB-LB) =
" ABSGAP.L:18:12;
    PUT / "          REL GAP 100*(UB-LB)/UB =
" RELGAP.L:18:12;

PUT //
PUT "INICIA REDUCCION POR
FACTIBILIDAD"

PUT //
PUT "TERMINA REDUCCION POR
FACTIBILIDAD"

);
);

PUT ///
PUT / "-----
-----";
PUT / "          RESULTADOS
FINALES DE LA REDUCCION"
PUT / "-----
-----";

*-----CALCULA DISTANCIA DESDE
LAS COTAS HACIA LA MEJOR SOLUCION
LOOP( (B,L),

```

```

DIFABSITOT(B,L)=ITOT.UP(B,L)-
ITOT.LO(B,L);
DIFABSITOTUP(B,L)=ITOT.UP(B,L)-
ITOTBEST.L(B,L);
DIFABSITOTLO(B,L)=ITOTBEST.L(B,L)-
ITOT.LO(B,L);

DIFRELITOT(B,L)=ZERO;
DIFRELITOTUP(B,L)=ZERO;
DIFRELITOTLO(B,L)=ZERO;
DIFRELITOT(B,L)$(DELINIITOT(B,L) NE
0)=100*DIFABSITOT(B,L)/DELINIITOT(B,L)
);
DIFRELITOTUP(B,L)$(DELINIITOT(B,L) NE
0)=100*DIFABSITOTUP(B,L)/DELINIITOT(B,
L);
DIFRELITOTLO(B,L)$(DELINIITOT(B,L) NE
0)=100*DIFABSITOTLO(B,L)/DELINIITOT(B,
L);
);

MAXABSDIS.L=0.0;
LOOP( SBLUEITOT(B,L),
MAXABSDIS.L=MAX[MAXABSDIS.L,DIFABSITOT
LO(SBLUEITOT),DIFABSITOTUP(SBLUEITOT)]
);

MAXPCDIS.L=0.0;
LOOP( SBLUEITOT(B,L),
MAXPCDIS.L=MAX[MAXPCDIS.L,DIFRELITOTLO
(SBLUEITOT)];
MAXPCDIS.L=MAX[MAXPCDIS.L,DIFRELITOTUP
(SBLUEITOT)];
);

*-----SE IMPRIMEN RESULTADOS

PUT //
LOOP( SBLUEITOT(B,L),
PUT /"ITOT.LO('B.TL:3, ",",L.TL:3, "' )=
",ITOT.LO(SBLUEITOT):18:12";"
" ITOT.UP('B.TL:3, ",",L.TL:3, "' )=
",ITOT.UP(SBLUEITOT):18:12";"
);

PUT //
LOOP( SBLUEITOT(B,L),
PUT
/"DAITOTLO('B.TL:3, ",",L.TL:3, "' )=
",DIFABSITOTLO(SBLUEITOT):18:12";"
"
DAITOTUP('B.TL:3, ",",L.TL:3, "' )=
",DIFABSITOTUP(SBLUEITOT):18:12";" /
"
DAITOT('B.TL:3, ",",L.TL:3, "' )=
",DIFABSITOT(SBLUEITOT):18:12";"
);
PUT / "DISTANCIA MAXIMA
ABSOLUTA:"MAXABSDIS.L:18:12;

PUT //
LOOP( SBLUEITOT(B,L),
PUT
/"DRITOTLO('B.TL:3, ",",L.TL:3, "' )=
",DIFRELITOTLO(SBLUEITOT):18:12";"
"
DRITOTUP('B.TL:3, ",",L.TL:3, "' )=
",DIFRELITOTUP(SBLUEITOT):18:12";" /
"
DRITOT('B.TL:3, ",",L.TL:3, "' )=
",DIFRELITOT(SBLUEITOT):18:12";"
);
PUT / "DISTANCIA MAXIMA
PORCENTUAL:"MAXPCDIS.L:18:12;
*-----

```

```

PUT / "-----"
-----";
*-----
Solution to model MIP

SOLVE MR00 minimizing ZZ using mip;
*-----
creating file with results

PUT RESCRU;

PUT /// "-----"
-----";
PUT / "NEXT THE SOLUTION OF THE
RELAXED PROBLEM (MIP)"
PUT / "-----"
-----";

PUT //
LOOP( (B,L),
  PUT /"ITOT.1('
"B.TL:3,"",L.TL:3,"')=
",ITOT.1(B,L):18:12";"
);

PUT / "ZZ.1=" ZZ.1:20:12 ";";

PUT /// "LP solution time : "
MR00.resusd:10:5;
PUT / "Model Status: "
PUT$(MR00.modelstat eq 1) "Optimal.";
PUT$(MR00.modelstat eq 2) "Locally
optimal.";
PUT$(MR00.modelstat eq 3)
"Unbounded.";
PUT$(MR00.modelstat eq 4)
"Infeasible.";
PUT$(MR00.modelstat eq 5) "Locally
infeasible.";
PUT$(MR00.modelstat eq 6)
"Intermediate infeasible.";
PUT$(MR00.modelstat eq 7)
"Intermediate nonoptimal.";
PUT$(MR00.modelstat eq 8) "Integer
solution.";
PUT$(MR00.modelstat eq 9)
"Intermediate non-integer.";
PUT$(MR00.modelstat eq 10) "Integer
infeasible.";
PUT$(MR00.modelstat eq 12) "Error
unknown.";
PUT$(MR00.modelstat eq 13) "Error no
solution.";

*NEXT USED LEVELS ARE REPORTED

PUT //
LOOP( (B,L),
  PUT / "ITOT.LO('
"B.TL:3,"",L.TL:3,"')=
",ITOT.LO(B,L):18:12";"
" ITOT.UP('
"B.TL:3,"",L.TL:3,"')=
",ITOT.UP(B,L):18:12";"
);

*MULTIPLICATION

PUT //
LOOP( (B,L),

```

```

PUT / "ITOT.M('
"B.TL:3,"",L.TL:3,"')=
",ITOT.M(B,L):18:12";"
);

```

```

PUT ///
PUT "MODELSTAT FOR LB:"MR00.MODELSTAT;
PUT / "MODELSTAT FOR
UB:"MF00.MODELSTAT;

```

**APPENDIX VII: GAMS OUTPUT
FILE ON BRANCH AND
CONTRACT ALGORITHM
MODEL**

(FOR DISSERTATION
APPENDIX).gms
Output C:\Documents and
Settings\Kippi\My
Documents\gamsdir\projdir\COS_ITOT
(FOR DISSERTATION
APPENDIX).lst

GAMS Rev 149 x86/MS Windows
04/27/07 19:28:59 Page 7
: SCHEDULING REFINERY CRUDE OIL
OPERATIONS
Model Statistics SOLVE ACOTAVARS
Using RMIP From line 3295

LOOPS
ITERATIONS 1 *
ST1 *
L1

MODEL STATISTICS

BLOCKS OF EQUATIONS	77
SINGLE EQUATIONS	2,015
BLOCKS OF VARIABLES	38
SINGLE VARIABLES	836 1
projected	
NON ZERO ELEMENTS	5,277
DISCRETE VARIABLES	100

GENERATION TIME = 0.047
SECONDS 3 Mb WIN226-149 Dec 19,
2007

**** SOLVE from line 3295 ABORTED,
EXECERROR = 1

**** REPORT FILE SUMMARY

RESCRU C:\Documents and
Settings\Kippi\My
Documents\gamsdir\projdir\cos1.res
RESCRUBND C:\Documents and
Settings\Kippi\My
Documents\gamsdir\projdir\CBOUND.RE
S

EXECUTION TIME = 0.093
SECONDS 3 Mb WIN226-149 Dec 19,
2007

USER: CS/IE 635, Spring 2007
G061206/0001AS-WIN
Prof. Ferris
DC2937

License for teaching and
research at degree granting
institutions

**** FILE SUMMARY

Input C:\Documents and
Settings\Kippi\My
Documents\gamsdir\projdir\COS_ITOT

**APPENDIX VIII: COMPUTER
CODE FOR GAMS INPUT FILE
ON CRUDE OIL SCHEDULING
WITH SLACK VARIABLES**

MODEL

\$TITLE: SCHEDULING REFINERY CRUDE OIL
OPERATIONS

\$EOLCOM !

\$ONEMPTY

*number of equations listed per block
option limrow = 0;

*number of variables listed per block
option limcol=0;

*do you want solver's solution output
printed?
option solprint=on;

*do you want solver's system output
printed?
option sysout=off;

*milp relative termination tolerance
option optcr=0;

*milp absolute termination tolerance
option optca=0;

*Decimals to display
option decimals=8;

*File to write results
FILE RESCRU / cosslavar.res /;

PUT RESCRU;

PUT "-----"
-----"
PUT / " SUBSYSTEM B.DOWNSTREAM CRUDE
OIL SCHEDULING AT REFINERY FRONT-END "
PUT / "-----"
-----"

*-----
-----Choosing solvers

option lp=minos5;
option nlp=conopt;
option nlp=BARON;
option mip=CPLEX;
option rmip=osl;
option minlp=DICOPT;

SETS

A SET OF TANK
INPUT SOURCES /
ST1, ST2, ST3 U1, U2, U3,

B SET OF CRUDE
TANKS /
Tank 1 ST1 Storage
Tank 2 ST2 Storage

Tank 3 ST3 Storage
Charging Tank 1 CT1
Charging Tank 2 CT2
Charging Tank 3 CT3 /
H SET OF TANK
OUTPUT DESTINATIONS /
CT3, D1, D2 CT1, CT2,
/ /
D(H) SET OF
DISTILLATION UNITS /
/ D1 * D2 /
G(B) SET OF
CHARGING TANKS /
/ CT1 * CT3 /
I SET OF
COMPONENTS /
/ I1 * I2 /
M SOURCE UNIT
FOR SPLIT PIPELINE /
MCT2 MST2, MST3,
/ /
K DESTINATION
UNIT OF SPLIT PIPELINE /
/ KCT3, KD2 /
U(A) SET OF CRUDE
SUPPLY STREAMS /
/ U1 * U3 /
Y(B) SET OF
STORAGE TANKS /
/ ST1 * ST3 /
L TRANSFER
EVENT /
/ L0 * L2 /
Q SET OF SUB-
STRUCTURES /
/ Q1 * Q2 /
S SET OF SLACK
VARIABLES /
/ Q1 * Q2 /
ALIAS
(A , ADASH)
(H , HDASH)
(G , GDASH)
(L , LDASH)

* ===== for
MAPPING purposes

===== *

SETS

A1(A,B) SET OF
INPUTS A TO TANK B
/

U1.(ST1,ST2),U2.ST2,U3.(ST2,ST3),
ST1.(CT1,CT2),ST2.(CT1,CT2,CT3),ST3.(C
T2,CT3) /

A2(A,Y) SET OF
INPUTS A TO STORAGE TANK Y
/

U1.(ST1,ST2),U2.ST2,U3.(ST2,ST3)
/

D3(D,G) SET OF
DISTILLATION UNITS D THAT CAN BE
CHARGED BY CHARGING TANK G /
D1.CT1,
D1.CT2, D2.CT2, D2.CT3
/

G1(G,D) SET OF
CHARGING TANKS G THAT CHARGE
DISTILLATION UNIT D /
CT1.D1,
CT2.D1, CT2.D2, CT3.D2
/

H1(B,H) SET OF
OUTPUTS H FROM TANK B
/

ST1.(CT1,CT2),ST2.(CT1,CT2,CT3),ST3.(C
T2,CT3),

CT1.D1,CT2.(D1,D2),CT3.D2
/

H2(Y,H) SET OF
OUTPUTS H FROM STORAGE TANK Y
/

ST1.(CT1,CT2),ST2.(CT1,CT2,CT3),ST3.(C
T2,CT3) /

H3(G,H) SET OF
OUTPUTS H FROM CHARGING TANK G
/

CT1.D1,CT2.(D1,D2),CT3.D2
/

H4(Y,H) SET OF
OUTPUTS H FROM STORAGE TANK Y
/

ST1.CT1,ST2.(CT1,CT2,CT3),ST3.CT3 /

Y1(Y,U) SET OF
STORAGE TANKS Y CONNECTED TO CRUDE
SUPPLY STREAMS U /
ST1.U1,
ST2.U1, ST2.U2, ST2.U3, ST3.U3
/

* ===== for
DECOMPOSITION purposes

===== *

B1(B) SET OF TANKS
B BELONGING TO SUB-STRUCTURE Q1 /
ST1, ST2,
ST3, CT1, CT2 /

B2(B) SET OF TANKS
B BELONGING TO SUB-STRUCTURE Q2 /
CT3
/

D1(D) SET OF
DISTILLATION UNITS D PRESENT IN SUB-
STRUCTURE Q1 /

D1
/

D2(D) SET OF
DISTILLATION UNITS D PRESENT IN SUB-
STRUCTURE Q2 /

D2
/

U1(U) SET OF CRUDE
SUPPLY STREAMS U PRESENT IN SUB-
STRUCTURE Q1 /

U1, U2, U3
/

U2(U) SET OF CRUDE
SUPPLY STREAMS U PRESENT IN SUB-
STRUCTURE Q2 /

/

K1(M,K) SET OF
DESTINATION UNIT K OF SPLIT PIPELINES
WITH SOURCE M /

MST3.KCT3,
MST3.KCT3, MCT2.KD2
/

/

;

***** SLACK
VARIABLES *****

FREE VARIABLE

SLAVARTC3FT1(A,B,L)
SLAVARTC19IPOT(B,L)
SLAVARTC21INVI(I,B,L)
SLAVARTC23FT1(A,B,L)
SLAVARTC24FT2(B,H,L)
SLAVARTC27INVI(I,B,L)
SLAVARDC1WB2(D,L)
SLAVARDC2WB2(G,L)
SLAVARLBP4INVVJ(I,B,H,L)
SLAVARLBP5INVVJ(I,B,H,L)
SLAVARLBP6INVVT(I,B,H,L)
SLAVARLBP8INVVT(I,B,H,L)
SLAVARLBP13FLOWVJ(I,A,B,H,L)
SLAVARLBP14FLOWVT(I,A,B,H,L)
SLAVARDCUB1WBUB2(D,L)
SLAVARDCUB2WBUB2(G,L)

SLAVARTC3FT1Q1 (A,B,L)
 SLAVARTC19ITOTQ1 (B,L)
 SLAVARTC21INVIQ1 (I,B,L)
 SLAVARTC23FT1Q1 (A,B,L)
 SLAVARTC24FT2Q1 (B,H,L)
 SLAVARTC27INVIQ1 (I,B,L)
 SLAVARDC1WB2Q1 (D,L)
 SLAVARDC2WB2Q1 (G,L)
 SLAVARLBP4INVVJQ1 (I,B,H,L)
 SLAVARLBP5INVVJQ1 (I,B,H,L)
 SLAVARLBP6INVVTQ1 (I,B,H,L)
 SLAVARLBP8INVVTQ1 (I,B,H,L)
 SLAVARLBP13FLOWVJQ1 (I,A,B,H,L)
 SLAVARLBP14FLOWVTQ1 (I,A,B,H,L)
 SLAVARDCUB1WBUB2Q1 (D,L)
 SLAVARDCUB2WBUB2Q1 (G,L)

SLAVARTC3FT1Q2 (A,B,L)
 SLAVARTC19ITOTQ2 (B,L)
 SLAVARTC21INVIQ2 (I,B,L)
 SLAVARTC23FT1Q2 (A,B,L)
 SLAVARTC24FT2Q2 (B,H,L)
 SLAVARTC27INVIQ2 (I,B,L)
 SLAVARDC1WB2Q2 (D,L)
 SLAVARDC2WB2Q2 (G,L)
 SLAVARLBP4INVVJQ2 (I,B,H,L)
 SLAVARLBP5INVVJQ2 (I,B,H,L)
 SLAVARLBP6INVVTQ2 (I,B,H,L)
 SLAVARLBP8INVVTQ2 (I,B,H,L)
 SLAVARLBP13FLOWVJQ2 (I,A,B,H,L)
 SLAVARLBP14FLOWVTQ2 (I,A,B,H,L)
 SLAVARDCUB1WBUB2Q2 (D,L)
 SLAVARDCUB2WBUB2Q2 (G,L)

;

*=====

POSITIVE VARIABLE

ITOT (B,L) TOTAL
 INVENTORY OF TANK B AT THE END OF
 TRANSFER EVENT L
 INVI (I,B,L) INVENTORY OF
 COMPONENT I IN TANK B AT THE END OF
 TRANSFER EVENT L
 TS1 (A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1 (A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L
 TS2 (B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2 (B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L
 TS1Q1 (A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1Q1 (A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L

TS2Q1 (B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2Q1 (B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L

TS1Q2 (A,B,L) STARTING
 TIME OF A TRANSFER FROM A TO B IN
 TRANSFER EVENT L
 TE1Q2 (A,B,L) ENDING TIME
 OF A TRANSFER FROM A TO B IN TRANSFER
 EVENT L

TS2Q2 (B,H,L) STARTING
 TIME OF A TRANSFER FROM B TO H IN
 TRANSFER EVENT L
 TE2Q2 (B,H,L) ENDING TIME
 OF A TRANSFER FROM B TO H IN TRANSFER
 EVENT L

TEO (U) OVERALL
 ENDING TIME OF CRUDE TRANSFER FROM
 CRUDE SUPPLY STREAM U

TSI (U) INITIAL
 STARTING TIME OF CRUDE TRANSFER FROM
 CRUDE SUPPLY STREAM U

* =====
 Flow variables
 =====
 *

FT1 (A,B,L) TOTAL FLOW
 FROM A TO B IN TRANSFER EVENT L
 FT2 (B,H,L) TOTAL FLOW
 FROM B TO H IN TRANSFER EVENT L

FL1 (I,A,B,L) FLOW OF
 COMPONENT I FROM A TO B IN TRANSFER
 EVENT L

FL2 (I,B,H,L) FLOW OF
 COMPONENT I FROM B TO H IN TRANSFER
 EVENT L

FT1Q1 (A,B,L) TOTAL FLOW
 FROM A TO B IN TRANSFER EVENT L
 FT2Q1 (B,H,L) TOTAL FLOW
 FROM B TO H IN TRANSFER EVENT L

FL1Q1 (I,A,B,L) FLOW OF
 COMPONENT I FROM A TO B IN TRANSFER
 EVENT L

FL2Q1 (I,B,H,L) FLOW OF
 COMPONENT I FROM B TO H IN TRANSFER
 EVENT L

FT1Q2 (A,B,L) TOTAL FLOW
 FROM A TO B IN TRANSFER EVENT L
 FT2Q2 (B,H,L) TOTAL FLOW
 FROM B TO H IN TRANSFER EVENT L

FL1Q2 (I,A,B,L) FLOW OF
 COMPONENT I FROM A TO B IN TRANSFER
 EVENT L

```

      FL2Q2(I,B,H,L)   FLOW OF
COMPONENT I FROM B TO H IN TRANSFER
EVENT L

* ===== Variables to
replace bilinear terms to solve LBP
===== *

      INVVJ(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
      INVVT(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

      INVVJQ1(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
      INVVTQ1(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

      INVVJQ2(I,B,H,L)
INVENTORY COMPONENT I FOR LBP
      INVVTQ2(I,B,H,L)
INVENTORY COMPONENT I FOR LBP

      FLOWVJ(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP
      FLOWVT(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP

      FLOWVJQ(I,A,B,H,L,Q)   FLOW
OF COMPONENT I FOR LBP
      FLOWVTQ(I,A,B,H,L,Q)   FLOW
OF COMPONENT I FOR LBP

      FLOWVJQ1(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP
      FLOWVTQ1(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP

      FLOWVJQ2(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP
      FLOWVTQ2(I,A,B,H,L)   FLOW
OF COMPONENT I FOR LBP

* ===== Duplicate
variables declaration
===== *

      FLTQ(M,K,L)   TOTAL FLOW
VARIABLE
      FLTQ1(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
      FLTQ2(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

      FLQ(I,M,K,L)   FLOW
VARIABLE
      FLQ1(I,M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
      FLQ2(I,M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

      TSQ(M,K,L)   START TIME
VARIABLE
      TSQ1(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
      TSQ2(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

      TEQ(M,K,L)   END TIME
VARIABLE
      TEQ1(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1

```

```

      TEQ2(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

      WBQ(M,K,L)   FLOW
EXISTENCE VARIABLE
      WBQ1(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q1
      WBQ2(M,K,L)   DUPLICATE
VARIABLE FOR SUB-STRUCTURE Q2

;

FREE VARIABLES

      ZZ   OBJECTIVE
FUNCTION OF COST
      ZZUB   OBJECTIVE
FUNCTION FOR UPPER BOUND
      ZZFIXBIN
      ZZQ1   OBJECTIVE
FUNCTION OF COST FOR STRUCTURE Q1
      ZZQ2   OBJECTIVE
FUNCTION OF COST FOR STRUCTURE Q2
      ZZU   OBJECTIVE
FUNCTION UPPER BOUND

;

BINARY VARIABLES

      WB1(A,B,L)   EQUAL TO 1
IF THERE IS A FLOW FROM A TO B IN
TRANSFER EVENT L
      WB2(B,H,L)   EQUAL TO 1
IF THERE IS A FLOW FROM B TO H IN
TRANSFER EVENT L

      WB1Q1(A,B,L)   EQUAL TO 1
IF THERE IS A FLOW FROM A TO B IN
TRANSFER EVENT L
      WB2Q1(B,H,L)   EQUAL TO 1
IF THERE IS A FLOW FROM B TO H IN
TRANSFER EVENT L

      WB1Q2(A,B,L)   EQUAL TO 1
IF THERE IS A FLOW FROM A TO B IN
TRANSFER EVENT L
      WB2Q2(B,H,L)   EQUAL TO 1
IF THERE IS A FLOW FROM B TO H IN
TRANSFER EVENT L

;

*=====
MAIN MODEL
=====

PARAMETERS

      CSEA   WAITING COST
FOR CRUDE SUPPLY STREAMS
/ 5 /

```

CSET	CHANGEOVER			I1.CT3
COST FOR CHARGED OIL SWITCH		0.075		
/ 50 /		/		
CUNLOAD	UNLOADING			
COST FOR CRUDE SUPPLY STREAMS			UB1(I,B)	UPPER BOUND
/ 10 /			ON FRACTION OF COMPONENT I	INSIDE TANK
			B	/
THOR	TIME HORIZON			I1.ST1
FOR SCHEDULING		0.03		
/ 12 /				I1.ST2
		0.06		
ND	NUMBER OF			I1.ST3
DISTILLATION UNITS IN THE NETWORK		0.09		
/ 2 /				I1.CT1
ND1	NUMBER OF	0.035		
DISTILLATION UNITS IN SUB-STRUCTURE Q1				I1.CT2
/ 1 /		0.065		
ND2	NUMBER OF			I1.CT3
DISTILLATION UNITS IN SUB-STRUCTURE Q2		0.085		
/ 1 /		/		
NE	NUMBER OF		FRACT(I,U)	FRACTION OF
TRANSFER EVENTS			COMPONENT I IN CRUDE SUPPLY STREAM U	
/ 3 /			/	
				I1.U1
TSO	SCALAR STEP	0.01		
SIZE FOR UPDATING LAGRANGE MULTIPLIERS				I1.U2
/ 1 /		0.085		
				I1.U3
ALPHA	SCALAR	0.06		
BETWEEN 0 AND 2		/		
/ 0.4 /				
ZZ1	OPTIMAL		INITITOT(B)	INITIAL
OBJECTIVE FOR LQ1-R			TOTAL INVENTORY OF TANK B	
/ 0 /			/	
				ST1 * ST3
ZZ2	OPTIMAL	20		
OBJECTIVE FOR LQ2-R				CT1
/ 1 /		30		
ZZLB	SUM OF			CT2
OBJECTIVES ZZQ1 AND ZZQ2		50		
/ 1 /				CT3
		30		
		/		
CINV(B)	INVENTORY			
MAINTENANCE COST FOR TANK B			INITINVI(I,B)	INITIAL
/			INVENTORY OF COMPONENT I IN TANK B	
			/	
0.04	ST1 * ST3			I1.ST1
0.08	CT1 * CT3	0.02		I1.ST2
/				
		0.05		I1.ST3
		0.08		I1.CT1
DM(G)	DEMAND OF			
CRUDE-MIX TO BE CHARGED FROM CHARGING		0.03		I1.CT2
TANK G				
/				I1.CT3
	CT1 * CT3	0.05		
50				
/		0.08		
		/		
LB1(I,B)	LOWER BOUND			
ON FRACTION OF COMPONENT I INSIDE TANK			LBI1(B)	LOWER BOUND
B			ON TOTAL INVENTORY IN A TANK B	
/			/	
0.01	I1.ST1			ST1 * ST3
0.04	I1.ST2	1		CT1 * CT3
0.07	I1.ST3	1		
		/		
0.025	I1.CT1			
0.045	I1.CT2		UBI1(B)	UPPER BOUND
			ON TOTAL INVENTORY IN A TANK B	
			/	
				ST1 * ST3
		100		

100 CT1 * CT3
 /
 TARR(U) ARRIVAL TIME
 OF CRUDE IN CRUDE SUPPLY STREAM U
 /
 1 U1
 5 U2
 9 U3
 /

VCRUDE(U) TOTAL VOLUME
 OF CRUDE OIL ARRIVING IN CRUDE SUPPLY
 SYSTEM U /
 50 U1*U3
 /

* ===== value for these
 PARAMETERS are assigned below
 ===== *

LBF1(A,B) LOWER BOUND
 ON FLOWRATE FROM A TO B
 UBF1(A,B) UPPER BOUND
 ON FLOWRATE FROM A TO B

 LBF2(B,H) LOWER BOUND
 ON FLOWRATE FROM B TO H
 UBF2(B,H) UPPER BOUND
 ON FLOWRATE FROM B TO H

 LBFV1(A,B) LOWER BOUND
 ON FLOW VOLUME FROM A TO B
 UBFV1(A,B) UPPER BOUND
 ON FLOW VOLUME FROM A TO B
 LBFV2(B,H) LOWER BOUND
 ON FLOW VOLUME FROM B TO H
 UBFV2(B,H) UPPER BOUND
 ON FLOW VOLUME FROM B TO H

* ===== Lagrange
 multipliers
 ===== *

LMTS(M,K,L) LAGRANGE
 MULTIPLIER
 LMTE(M,K,L) LAGRANGE
 MULTIPLIER
 LMFLT(M,K,L) LAGRANGE
 MULTIPLIER
 LMFL(I,M,K,L) LAGRANGE
 MULTIPLIER
 LMWB(M,K,L) LAGRANGE
 MULTIPLIER

* ===== Binary
 variables assignation
 ===== *

WBUB1(A,B,L) EQUAL TO
 OPTIMAL VALUE OBTAINED FROM THE
 SOLUTION OF CRUDE_LB
 WBUB2(B,H,L) EQUAL TO
 OPTIMAL VALUE OBTAINED FROM THE
 SOLUTION OF CRUDE_LB

* ===== Duplicate
 variables assignation
 ===== *

DFLTQ(M,K,L) TOTAL FLOW
 VARIABLE
 DFLTQ1(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DFLTQ2(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

 DFLQ(I,M,K,L) FLOW
 VARIABLE
 DFLQ1(I,M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DFLQ2(I,M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DTSQ(M,K,L) START TIME
 VARIABLE
 DTSQ1(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DTSQ2(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DTEQ(M,K,L) END TIME
 VARIABLE
 DTEQ1(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DTEQ2(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

DWBQ(M,K,L) FLOW
 EXISTENCE VARIABLE
 DWBQ1(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q1
 DWBQ2(M,K,L) DUPLICATE
 VARIABLE FOR SUB-STRUCTURE Q2

* ===== PARAMETER
 value assignment
 ===== *

LBF1(A,B) = 1 ;
 UBF1(A,B) = 40 ;

 LBF2(B,H) = 1 ;
 UBF2(B,H) = 40 ;

 LBFV1(A,B) = 1 ;
 UBFV1(A,B) = 100 ;

 LBFV2(B,H) = 1 ;
 UBFV2(B,H) = 100 ;

 LMTS(M,K,L) = 0 ;
 LMTE(M,K,L) = 0 ;
 LMFLT(M,K,L) = 0 ;
 LMFL(I,M,K,L) = 0 ;
 LMWB(M,K,L) = 0 ;

*===== MAIN MODEL
 =====

EQUATIONS

FUNCTION	OBJFUN	OBJECTIVE		
FUNCTION	OBJFIXBIN	OBJECTIVE	DC1	DISTILLATION UNIT
			CONSTRAINT 1	
FUNCTION OF COST FOR STRUCTURE Q1	OBJQ1	OBJECTIVE	DC2	DISTILLATION UNIT
			CONSTRAINT 2	
FUNCTION OF COST FOR STRUCTURE Q2	OBJQ2	OBJECTIVE	DC3	DISTILLATION UNIT
			CONSTRAINT 3	
			DC4	DISTILLATION UNIT
			CONSTRAINT 4	
			DC5	DISTILLATION UNIT
			CONSTRAINT 5	
	TC1	TANK CONSTRAINT 1		
	TC2	TANK CONSTRAINT 2		
	TC3	TANK CONSTRAINT 3		
	TC4	TANK CONSTRAINT 4		
	TC5	TANK CONSTRAINT 5	CC1	CRUDE SUPPLY
	TC6	TANK CONSTRAINT 6	STREAM CONSTRAINT 1	
	TC7	TANK CONSTRAINT 7	CC2	CRUDE SUPPLY
	TC8	TANK CONSTRAINT 8	STREAM CONSTRAINT 2	
	TC9	TANK CONSTRAINT 9	CC3	CRUDE SUPPLY
10	TC10	TANK CONSTRAINT	STREAM CONSTRAINT 3	
			CC4	CRUDE SUPPLY
11	TC11	TANK CONSTRAINT	STREAM CONSTRAINT 4	
12	TC12	TANK CONSTRAINT		
			VB1	VARIABLE BOUND 1
			VB2	VARIABLE BOUND 2
13	TC13	TANK CONSTRAINT	VB3	VARIABLE BOUND 3
			VB4	VARIABLE BOUND 4
14	TC14	TANK CONSTRAINT	VB5	VARIABLE BOUND 5
			VB6	VARIABLE BOUND 6
15	TC15	TANK CONSTRAINT	VB7	VARIABLE BOUND 7
			VB8	VARIABLE BOUND 8
16	TC16	TANK CONSTRAINT	VB9	VARIABLE BOUND 9
			VB10	VARIABLE BOUND 10
17	TC17	TANK CONSTRAINT	VB11	VARIABLE BOUND 11
			VB12	VARIABLE BOUND 12
18	TC18	TANK CONSTRAINT	VB13	VARIABLE BOUND 13
			VB14	VARIABLE BOUND 14
19	TC19	TANK CONSTRAINT	VB15	VARIABLE BOUND 15
20	TC20	TANK CONSTRAINT		
			LBP1	LOWER BOUNDING
21	TC21	TANK CONSTRAINT	PROBLEM 1	
			LBP2	LOWER BOUNDING
22	TC22	TANK CONSTRAINT	PROBLEM 2	
			LBP3	LOWER BOUNDING
23	TC23	TANK CONSTRAINT	PROBLEM 3	
			LBP4	LOWER BOUNDING
24	TC24	TANK CONSTRAINT	PROBLEM 4	
			LBP5	LOWER BOUNDING
25	TC25	TANK CONSTRAINT	PROBLEM 5	
			LBP6	LOWER BOUNDING
26	TC26	TANK CONSTRAINT	PROBLEM 6	
			LBP7	LOWER BOUNDING
27	TC27	TANK CONSTRAINT	PROBLEM 7	
			LBP8	LOWER BOUNDING
28	TC28	TANK CONSTRAINT	PROBLEM 8	
			LBP9	LOWER BOUNDING
29	TC29	TANK CONSTRAINT	PROBLEM 9	
			LBP10	LOWER BOUNDING
30	TC30	TANK CONSTRAINT	PROBLEM 10	
			LBP11	LOWER BOUNDING
31	TC31	TANK CONSTRAINT	PROBLEM 11	
			LBP12	LOWER BOUNDING
*\$ONTEXT			PROBLEM 12	
32	TC32	TANK CONSTRAINT	LBP13	LOWER BOUNDING
			PROBLEM 13	
33	TC33	TANK CONSTRAINT	LBP14	LOWER BOUNDING
			PROBLEM 14	
34	TC34	TANK CONSTRAINT	LBP15	LOWER BOUNDING
			PROBLEM 15	
35	TC35	TANK CONSTRAINT	LBP16	LOWER BOUNDING
			PROBLEM 16	
36	TC36	TANK CONSTRAINT	LBP17	LOWER BOUNDING
			PROBLEM 17	
*\$OFFTEXT				

LBP18 LOWER BOUNDING
 PROBLEM 18
 LBP19 LOWER BOUNDING
 PROBLEM 19
 LBP20 LOWER BOUNDING
 PROBLEM 20
 LBP21 LOWER BOUNDING
 PROBLEM 21
 LBP22 LOWER BOUNDING
 PROBLEM 22

* ===== LBP for sub-
 structures Q1 and Q2
 ===== *

LBP10Q LOWER BOUNDING
 PROBLEM 10
 LBP11Q LOWER BOUNDING
 PROBLEM 11
 LBP12Q LOWER BOUNDING
 PROBLEM 12
 LBP13Q LOWER BOUNDING
 PROBLEM 13
 LBP14Q LOWER BOUNDING
 PROBLEM 14
 LBP15Q LOWER BOUNDING
 PROBLEM 15
 LBP16Q LOWER BOUNDING
 PROBLEM 16
 LBP17Q LOWER BOUNDING
 PROBLEM 17

LBP1Q1 LOWER BOUNDING
 PROBLEM 1
 LBP2Q1 LOWER BOUNDING
 PROBLEM 2
 LBP3Q1 LOWER BOUNDING
 PROBLEM 3
 LBP4Q1 LOWER BOUNDING
 PROBLEM 4
 LBP5Q1 LOWER BOUNDING
 PROBLEM 5
 LBP6Q1 LOWER BOUNDING
 PROBLEM 6
 LBP7Q1 LOWER BOUNDING
 PROBLEM 7
 LBP8Q1 LOWER BOUNDING
 PROBLEM 8
 LBP9Q1 LOWER BOUNDING
 PROBLEM 9
 LBP10Q1 LOWER BOUNDING
 PROBLEM 10
 LBP11Q1 LOWER BOUNDING
 PROBLEM 11
 LBP12Q1 LOWER BOUNDING
 PROBLEM 12
 LBP13Q1 LOWER BOUNDING
 PROBLEM 13
 LBP14Q1 LOWER BOUNDING
 PROBLEM 14
 LBP15Q1 LOWER BOUNDING
 PROBLEM 15
 LBP16Q1 LOWER BOUNDING
 PROBLEM 16
 LBP17Q1 LOWER BOUNDING
 PROBLEM 17

LBP1Q2 LOWER BOUNDING
 PROBLEM 1
 LBP2Q2 LOWER BOUNDING
 PROBLEM 2
 LBP3Q2 LOWER BOUNDING
 PROBLEM 3
 LBP4Q2 LOWER BOUNDING
 PROBLEM 4

LBP5Q2 LOWER BOUNDING
 PROBLEM 5
 LBP6Q2 LOWER BOUNDING
 PROBLEM 6
 LBP7Q2 LOWER BOUNDING
 PROBLEM 7
 LBP8Q2 LOWER BOUNDING
 PROBLEM 8
 LBP9Q2 LOWER BOUNDING
 PROBLEM 9
 LBP10Q2 LOWER BOUNDING
 PROBLEM 10
 LBP11Q2 LOWER BOUNDING
 PROBLEM 11
 LBP12Q2 LOWER BOUNDING
 PROBLEM 12
 LBP13Q2 LOWER BOUNDING
 PROBLEM 13
 LBP14Q2 LOWER BOUNDING
 PROBLEM 14
 LBP15Q2 LOWER BOUNDING
 PROBLEM 15
 LBP16Q2 LOWER BOUNDING
 PROBLEM 16
 LBP17Q2 LOWER BOUNDING
 PROBLEM 17

* ===== Equations to
 update Lagrange multiplier
 ===== *

ULM1 UPDATING LAGRANGE
 MULTIPLIER 1
 ULM2 UPDATING LAGRANGE
 MULTIPLIER 2
 ULM3 UPDATING LAGRANGE
 MULTIPLIER 3
 ULM4 UPDATING LAGRANGE
 MULTIPLIER 4
 ULM5 UPDATING LAGRANGE
 MULTIPLIER 5
 ULM6 UPDATING LAGRANGE
 MULTIPLIER 6

* =====
 Equations to generate VLC
 ===== *

VLC1 GENERATING VALID
 LINEAR CUTS
 VLC2 GENERATING VALID
 LINEAR CUTS

* ===== Equations used to
 evaluate UB after setting binary
 variables ===== *

TCUB1 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB2 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB3 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB4 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB5 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB6 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB8 DUPLICATE
 EQUATION FOR UB EVALUATION
 TCUB11 DUPLICATE
 EQUATION FOR UB EVALUATION

	TCUB14	DUPLICATE		TC24Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			24		
	TCUB15	DUPLICATE	*	TC25Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			25		
	TCUB16	DUPLICATE		TC26Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			26		
	TCUB17	DUPLICATE		TC27Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			27		
	TCUB18	DUPLICATE		TC28Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			28		
	DCUB1	DUPLICATE		TC29Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			29		
	DCUB2	DUPLICATE		TC30Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			30		
	DCUB4	DUPLICATE		TC31Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			31		
	DCUB5	DUPLICATE			
EQUATION FOR UB EVALUATION			*\$CONTEXT		
				TC32Q1	TANK CONSTRAINT
			32		
	CCUB1	DUPLICATE		TC33Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			33		
	CCUB2	DUPLICATE		TC34Q1	TANK CONSTRAINT
EQUATION FOR UB EVALUATION			34		
				TC35Q1	TANK CONSTRAINT
			35		
* ===== DUPLICATE				TC36Q1	TANK CONSTRAINT
EQUATIONS AND VARIABLES FOR LB1			36		
===== *			*\$OFFTEXT		
	TC1Q1	TANK CONSTRAINT			
1					
	TC2Q1	TANK CONSTRAINT		DC1Q1	DISTILLATION
2				UNIT CONSTRAINT 1	
	TC3Q1	TANK CONSTRAINT		DC2Q1	DISTILLATION
3				UNIT CONSTRAINT 2	
	TC4Q1	TANK CONSTRAINT		DC3Q1	DISTILLATION
4				UNIT CONSTRAINT 3	
	TC5Q1	TANK CONSTRAINT		DC4Q1	DISTILLATION
5				UNIT CONSTRAINT 4	
	TC6Q1	TANK CONSTRAINT		DC5Q1	DISTILLATION
6				UNIT CONSTRAINT 5	
	TC7Q1	TANK CONSTRAINT			
7					
	TC8Q1	TANK CONSTRAINT		CC1Q1	CRUDE SUPPLY
8				STREAM CONSTRAINT 1	
	TC9Q1	TANK CONSTRAINT		CC2Q1	CRUDE SUPPLY
9				STREAM CONSTRAINT 2	
	TC10Q1	TANK CONSTRAINT		CC3Q1	CRUDE SUPPLY
10				STREAM CONSTRAINT 3	
	TC11Q1	TANK CONSTRAINT		CC4Q1	CRUDE SUPPLY
11				STREAM CONSTRAINT 4	
	TC12Q1	TANK CONSTRAINT			
12					
	TC13Q1	TANK CONSTRAINT		VB1Q1	VARIABLE BOUND
13			1		
	TC14Q1	TANK CONSTRAINT		VB2Q1	VARIABLE BOUND
14			2		
	TC15Q1	TANK CONSTRAINT		VB3Q1	VARIABLE BOUND
15			3		
	TC16Q1	TANK CONSTRAINT		VB4Q1	VARIABLE BOUND
16			4		
	TC17Q1	TANK CONSTRAINT		VB5Q1	VARIABLE BOUND
17			5		
	TC18Q1	TANK CONSTRAINT		VB6Q1	VARIABLE BOUND
18			6		
	TC19Q1	TANK CONSTRAINT		VB7Q1	VARIABLE BOUND
19			7		
	TC20Q1	TANK CONSTRAINT		VB8Q1	VARIABLE BOUND
20			8		
	TC21Q1	TANK CONSTRAINT		VB9Q1	VARIABLE BOUND
21			9		
	TC22Q1	TANK CONSTRAINT		VB10Q1	VARIABLE BOUND
22			10		
	TC23Q1	TANK CONSTRAINT		VB11Q1	VARIABLE BOUND
23			11		

12	VB12Q1	VARIABLE BOUND		
			*\$ONTEXT	
	VB13Q1	VARIABLE BOUND		TC32Q2 TANK CONSTRAINT
13			32	
	VB14Q1	VARIABLE BOUND		TC33Q2 TANK CONSTRAINT
14			33	
	VB15Q1	VARIABLE BOUND		TC34Q2 TANK CONSTRAINT
15			34	
			35	TC35Q2 TANK CONSTRAINT
			36	TC36Q2 TANK CONSTRAINT
	* ===== DPLICATE			
	EQUATIONS AND VARIABLES FOR LB			
	===== *			
			*\$OFFTEXT	
	TC1Q2	TANK CONSTRAINT		
1				DC1Q2 DISTILLATION
	TC2Q2	TANK CONSTRAINT		UNIT CONSTRAINT 1
2				DC2Q2 DISTILLATION
	TC3Q2	TANK CONSTRAINT		UNIT CONSTRAINT 2
3				DC3Q2 DISTILLATION
	TC4Q2	TANK CONSTRAINT		UNIT CONSTRAINT 3
4				DC4Q2 DISTILLATION
	TC5Q2	TANK CONSTRAINT		UNIT CONSTRAINT 4
5				DC5Q2 DISTILLATION
	TC6Q2	TANK CONSTRAINT		UNIT CONSTRAINT 5
6				
	TC7Q2	TANK CONSTRAINT		
7				CC1Q2 CRUDE SUPPLY
	TC8Q2	TANK CONSTRAINT		STREAM CONSTRAINT 1
8				CC2Q2 CRUDE SUPPLY
	TC9Q2	TANK CONSTRAINT		STREAM CONSTRAINT 2
9				CC3Q2 CRUDE SUPPLY
	TC10Q2	TANK CONSTRAINT		STREAM CONSTRAINT 3
10				CC4Q2 CRUDE SUPPLY
	TC11Q2	TANK CONSTRAINT		STREAM CONSTRAINT 4
11				
	TC12Q2	TANK CONSTRAINT		
12				VB1Q2 VARIABLE BOUND
	TC13Q2	TANK CONSTRAINT	1	
13				VB2Q2 VARIABLE BOUND
	TC14Q2	TANK CONSTRAINT	2	
14				VB3Q2 VARIABLE BOUND
	TC15Q2	TANK CONSTRAINT	3	
15				VB4Q2 VARIABLE BOUND
	TC16Q2	TANK CONSTRAINT	4	
16				VB5Q2 VARIABLE BOUND
	TC17Q2	TANK CONSTRAINT	5	
17				VB6Q2 VARIABLE BOUND
	TC18Q2	TANK CONSTRAINT	6	
18				VB7Q2 VARIABLE BOUND
	TC19Q2	TANK CONSTRAINT	7	
19				VB8Q2 VARIABLE BOUND
	TC20Q2	TANK CONSTRAINT	8	
20				VB9Q2 VARIABLE BOUND
	TC21Q2	TANK CONSTRAINT	9	
21				VB10Q2 VARIABLE BOUND
	TC22Q2	TANK CONSTRAINT	10	
22				VB11Q2 VARIABLE BOUND
	TC23Q2	TANK CONSTRAINT	11	
23				VB12Q2 VARIABLE BOUND
	TC24Q2	TANK CONSTRAINT	12	
24				VB13Q2 VARIABLE BOUND
*	TC25Q2	TANK CONSTRAINT	13	
25				VB14Q2 VARIABLE BOUND
	TC26Q2	TANK CONSTRAINT	14	
26				VB15Q2 VARIABLE BOUND
	TC27Q2	TANK CONSTRAINT	15	
27				
	TC28Q2	TANK CONSTRAINT	;	
28				
	TC29Q2	TANK CONSTRAINT	*	
29				
	TC30Q2	TANK CONSTRAINT	-----	
30			*	EQUATION
	TC31Q2	TANK CONSTRAINT		
31			*	DEFINITION

```

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

OBJFUN..
      ZZ =E= CSEA*SUM(U,TSI(U)-
TARR(U)) + CUNLOAD*SUM(U,TEO(U)-
TSI(U)) +

THOR*(SUM(B,CINV(B))*SUM(B,SUM(L,(ITOT
(B,L)))) + SUM(B,CINV(B))*
SUM(B,SUM(L,SUM(A$A1(A,B),FT1(A,B,L)))
) +

SUM(B,CINV(B))*SUM(B,SUM(L$(ORD(L) LT
CARD(L)),ITOT(B,L))) +
THOR*SUM(B,CINV(B))*SUM(B,INITITOT(B)
) / (2*NE + 1) +

CSET*(SUM(D,SUM(G$G1(G,D)),SUM(L,WB2(G,
D,L)))) - ND)
$ontext
*           + 10*SUM((A,B,L),
SLAVARTC3FT1(A,B,L) +
SLAVARTC23FT1(A,B,L)
          + 10*SUM((B,L),
SLAVARTC19ITOT(B,L)
          + 10*SUM((I,B,L),
SLAVARTC21INVI(I,B,L) +
SLAVARTC27INVI(I,B,L) )
*           + 10*SUM((B,H,L),
SLAVARTC24FT2(B,H,L) )
          + 10*SUM((D,L),
SLAVARDC1WB2(D,L) )
          + 10*SUM((G,L),
SLAVARDC2WB2(G,L) )
*           + 10*SUM((I,B,H,L),
SLAVARLBP4INVVJ(I,B,H,L) )
*           + 10*SUM((I,B,H,L),
SLAVARLBP5INVVJ(I,B,H,L) )
*           + 10*SUM((I,B,H,L),
SLAVARLBP8INVVT(I,B,H,L) )
*           + 10*SUM(
(I,A,B,H,L),
SLAVARLBP13FLOWVJ(I,A,B,H,L) +
SLAVARLBP14FLOWVT(I,A,B,H,L) )
*           + 10*SUM(S,
SLACK(S))
$offtext
;

OBJFIXBIN..
      ZZFIXBIN =E=
CSEA*SUM(U,TSI(U)-TARR(U)) +
CUNLOAD*SUM(U,TEO(U)-TSI(U)) +

THOR*(SUM(B,CINV(B))*SUM(B,SUM(L,(ITOT
(B,L)))) + SUM(B,CINV(B))*
SUM(B,SUM(L,SUM(A$A1(A,B),FT1(A,B,L)))
) +

SUM(B,CINV(B))*SUM(B,SUM(L$(ORD(L) LT
CARD(L)),ITOT(B,L))) +
THOR*SUM(B,CINV(B))*SUM(B,INITITOT(B)
) / (2*NE + 1) +

CSET*(SUM(D,SUM(G$G1(G,D)),SUM(L,WBUB2(
G,D,L)))) - ND)
;

```

```

OBJQ1..
      ZZQ1 =E= CSEA*SUM(U1,TSI(U1)-
TARR(U1)) + CUNLOAD*SUM(U1,TEO(U1)-
TSI(U1)) +

THOR*(SUM(B1,CINV(B1))*SUM(B1,SUM(L,(I
TOT(B1,L)))) +
SUM(B1,CINV(B1))*SUM(B1,SUM(L,SUM(A,FT
1Q1(A,B1,L)))) +

SUM(B1,CINV(B1))*SUM(B1,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B1,L))) +
THOR*SUM(B1,CINV(B1))*SUM(B2,INITITOT(
B2))) / (2*NE + 1) +

CSET*(SUM(D1,SUM(G$G1(G,D1)),SUM(L,WB2Q
1(G,D1,L)))) - ND1) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ1(M,K,L)))) +

SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ1(I,M,K,L)))) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ1(M,K,L)))) +

SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ1(M,K,L)))) +
SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ1(M,K,L)))));

OBJQ2..
      ZZQ2 =E= CSEA*SUM(U2,TSI(U2)-
TARR(U2)) + CUNLOAD*SUM(U2,TEO(U2)-
TSI(U2)) +

THOR*(SUM(B2,CINV(B2))*SUM(B2,SUM(L,(I
TOT(B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L,SUM(A,FT
1(A,B2,L)))) +

SUM(B2,CINV(B2))*SUM(B2,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B2,L))) +
THOR*SUM(B2,CINV(B2))*SUM(B2,INITITOT(
B2))) / (2*NE + 1) +

CSET*(SUM(D2,SUM(G$G1(G,D2)),SUM(L,WB2(
G,D2,L)))) - ND2) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ2(M,K,L)))) -

SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ2(I,M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ2(M,K,L)))) -

SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ2(M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ2(M,K,L)))));

```

```

* ===== 1.TANK
CONSTRAINTS
=====
== *

```

```

**(i)Constraints for Flow Transfers

```

```

TC1(A,B,L)$A1(A,B)..
      FT1(A,B,L) =L=
UBFV1(A,B)*WB1(A,B,L)
;

```

```

TC2 (B, H, L) $H1 (B, H) . .
    FT2 (B, H, L) =L=
    UBFV2 (B, H) *WB2 (B, H, L)
;

TCUB1 (A, B, L) $A1 (A, B) . .
    FT1 (A, B, L) =L=
    UBFV1 (A, B) *WBUB1 (A, B, L)
;

TCUB2 (B, H, L) $H1 (B, H) . .
    FT2 (B, H, L) =L=
    UBFV2 (B, H) *WBUB2 (B, H, L)
;

** (ii) Duration Constraints

TC3 (A, B, L) $A1 (A, B) . .
    UBF1 (A, B) * (TE1 (A, B, L) -
    TS1 (A, B, L)) + UBF1 (A, B) * THOR * (1 -
    WB1 (A, B, L)) =G=
    FT1 (A, B, L) +
    SLAVARTC3 FT1 (A, B, L)
;

TC4 (B, H, L) $H1 (B, H) . .
    UBF2 (B, H) * (TE2 (B, H, L) -
    TS2 (B, H, L)) + UBF2 (B, H) * THOR * (1 -
    WB2 (B, H, L)) =G=
    FT2 (B, H, L)
;

TC5 (A, Y, L) $A2 (A, Y) . .
    LBF1 (A, Y) * (TE1 (A, Y, L) -
    TS1 (A, Y, L)) - LBF1 (A, Y) * THOR * (1 -
    WB1 (A, Y, L)) =L=
    FT1 (A, Y, L)
;

TC6 (Y, H, L) $H2 (Y, H) . .
    LBF2 (Y, H) * (TE2 (Y, H, L) -
    TS2 (Y, H, L)) - LBF2 (Y, H) * THOR * (1 -
    WB2 (Y, H, L)) =L=
    FT2 (Y, H, L)
;

TC7 (G, H, L) $H3 (G, H) . .
    LBF2 (G, H) * (TE2 (G, H, L) -
    TS2 (G, H, L)) =L=
    FT2 (G, H, L)
;

TCUB3 (A, B, L) $A1 (A, B) . .
    UBF1 (A, B) * (TE1 (A, B, L) -
    TS1 (A, B, L)) + UBF1 (A, B) * THOR * (1 -
    WBUB1 (A, B, L)) =G=
    FT1 (A, B, L)
;

TCUB4 (B, H, L) $H1 (B, H) . .
    UBF2 (B, H) * (TE2 (B, H, L) -
    TS2 (B, H, L)) + UBF2 (B, H) * THOR * (1 -
    WBUB2 (B, H, L)) =G=
    FT2 (B, H, L)
;

TCUB5 (A, Y, L) $A2 (A, Y) . .
    LBF1 (A, Y) * (TE1 (A, Y, L) -
    TS1 (A, Y, L)) - LBF1 (A, Y) * THOR * (1 -
    WBUB1 (A, Y, L)) =L=
    FT1 (A, Y, L)
;

TCUB6 (Y, H, L) $H2 (Y, H) . .
    LBF2 (Y, H) * (TE2 (Y, H, L) -
    TS2 (Y, H, L)) - LBF2 (Y, H) * THOR * (1 -
    WBUB2 (Y, H, L)) =L=
    FT2 (Y, H, L)
;

** (iii) Simple Sequencing Constraints

TC8 (A, B, L) $ (ORD (L) LT
    CARD (L) $A1 (A, B)) . .
    TS1 (A, B, L+1) =G=
    TE1 (A, B, L) - THOR * (1 -
    WB1 (A, B, L))
;

TC9 (A, B, L) $ (ORD (L) LT
    CARD (L) $A1 (A, B)) . .
    TS1 (A, B, L+1) =G=
    TS1 (A, B, L)
;

TC10 (A, B, L) $ (ORD (L) LT
    CARD (L) $A1 (A, B)) . .
    TE1 (A, B, L+1) =G=
    TE1 (A, B, L)
;

TC11 (B, H, L) $ (ORD (L) LT
    CARD (L) $H1 (B, H)) . .
    TS2 (B, H, L+1) =G=
    TE2 (B, H, L) - THOR * (1 -
    WB2 (B, H, L))
;

TC12 (B, H, L) $ (ORD (L) LT
    CARD (L) $H1 (B, H)) . .
    TS2 (B, H, L+1) =G=
    TS2 (B, H, L)
;

TC13 (B, H, L) $ (ORD (L) LT
    CARD (L) $H1 (B, H)) . .
    TE2 (B, H, L+1) =G=
    TE2 (B, H, L)
;

TCUB8 (A, B, L) $ (ORD (L) LT
    CARD (L) $A1 (A, B)) . .
    TS1 (A, B, L+1) =G=
    TE1 (A, B, L) - THOR * (1 -
    WBUB1 (A, B, L))
;

TCUB11 (B, H, L) $ (ORD (L) LT
    CARD (L) $H1 (B, H)) . .
    TS2 (B, H, L+1) =G=
    TE2 (B, H, L) - THOR * (1 -
    WBUB2 (B, H, L))
;

** (iv) Input and Output Restraints for
the Entire Horizon

TC14 (A, ADASH, B, L) $ (ORD (L) LT
    CARD (L) $ (ORD (A) NE
    ORD (ADASH) $A1 (A, B))) . .
    TS1 (A, B, L+1) =G=
    TE1 (ADASH, B, L) - THOR * (1 -
    WB1 (ADASH, B, L))
;

```

```

TC15(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H))..
      TS1(A,B,L+1) =G=
      TE2(B,H,L) - THOR*(1-
WB2(B,H,L))
;

TC16(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H))..
      TS2(B,H,L+1) =G=
      TE1(A,B,L) - THOR*(1-
WB1(A,B,L))
;

TC17(B,H,HDASH,L)$(ORD(L) LT
CARD(L)$(ORD(H) NE
ORD(HDASH)$H1(B,H))..
      TS2(B,H,L+1) =G=
      TE2(B,HDASH,L) - THOR*(1-
WB2(B,HDASH,L))
;

TC18(A,B,H,L)$(A1(A,B)$H1(B,H))..
      TE1(A,B,L) - THOR*(1-
WB1(A,B,L)) =L=
      TS2(B,H,L) + THOR*(1-
WB2(B,H,L))
;

TCUB14(A,ADASH,B,L)$(ORD(L) LT
CARD(L)$(ORD(A) NE
ORD(ADASH)$A1(A,B))..
      TS1(A,B,L+1) =G=
      TE1(ADASH,B,L) - THOR*(1-
WBUB1(ADASH,B,L))
;

TCUB15(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H))..
      TS1(A,B,L+1) =G=
      TE2(B,H,L) - THOR*(1-
WBUB2(B,H,L))
;

TCUB16(A,B,H,L)$(ORD(L) LT
CARD(L)$(A1(A,B)$H1(B,H))..
      TS2(B,H,L+1) =G=
      TE1(A,B,L) - THOR*(1-
WBUB1(A,B,L))
;

TCUB17(B,H,HDASH,L)$(ORD(L) LT
CARD(L)$(ORD(H) NE
ORD(HDASH)$H1(B,H))..
      TS2(B,H,L+1) =G=
      TE2(B,HDASH,L) - THOR*(1-
WBUB2(B,HDASH,L))
;

TCUB18(A,B,H,L)$(A1(A,B)$H1(B,H))..
      TE1(A,B,L) - THOR*(1-
WBUB1(A,B,L)) =L=
      TS2(B,H,L) + THOR*(1-
WBUB2(B,H,L))
;

** (v) Mass Balances

TC19(B,L)..
      ITOT(B,L-1) +
SUM(A$A1(A,B),FT1(A,B,L)) =E=
      ITOT(B,L) +
SUM(H$H1(B,H),FT2(B,H,L)) +
SLAVARTC19ITOT(B,L) ;

TC20(B)..
      ITOT(B,'L0') =E=
      INITITOT(B)
;

TC21(I,B,L)..
      INVI(I,B,L-1) +
SUM(A$A1(A,B),FL1(I,A,B,L)) =E=
      INVI(I,B,L)
+SUM(H$H1(B,H),FL2(I,B,H,L)) +
SLAVARTC21INVI(I,B,L) ;

TC22(I,B)..
      INVI(I,B,'L0') =E=
      INITINVI(I,B)
;

TC23(A,B,L)$A1(A,B)..
      FT1(A,B,L) =E=
      SUM(I,FL1(I,A,B,L)) +
SLAVARTC23FT1(A,B,L)
;

TC24(B,H,L)$H1(B,H)..
      FT2(B,H,L) =E=
      SUM(I,FL2(I,B,H,L)) +
SLAVARTC24FT2(B,H,L)
;

** (vi) Component Balances (Replaced by
LBP)

TC25(I,B,H,L)$H1(B,H)..
      FL2(I,B,H,L)*(ITOT(B,L-
1)+SUM(A$A1(A,B),FT1(A,B,L))) =E=
      FT2(B,H,L)*(INVI(I,B,L-
1)+SUM(A$A1(A,B),FL1(I,A,B,L)));

LBP1(I,B,H,L)$H1(B,H)..
      INVVJ(I,B,H,L) +
SUM(A$A1(A,B),FLOWVJ(I,A,B,H,L)) =E=
      INVVT(I,B,H,L) +
SUM(A$A1(A,B),FLOWVT(I,A,B,H,L))
;

LBP2(I,B,H,L)$H1(B,H)..
      INVVJ(I,B,H,L) =G=
      LBI1(B)*FL2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

LBP3(I,B,H,L)$H1(B,H)..
      INVVJ(I,B,H,L) =G=
      UBI1(B)*FL2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP4(I,B,H,L)$H1(B,H)..
      INVVJ(I,B,H,L) =L=
      LBI1(B)*FL2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP4INVVJ(I,B,H,L) ;

LBP5(I,B,H,L)$H1(B,H)..
      INVVJ(I,B,H,L) =L=
      UBI1(B)*FL2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP5INVVJ(I,B,H,L) ;

```



```

LBP6(I,B,H,L)$H1(B,H)..
    INVVT(I,B,H,L) =G=
    LBI1(B)*FT2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP6INVVT(I,B,H,L) ;

LBP7(I,B,H,L)$H1(B,H)..
    INVVT(I,B,H,L) =G=
    UBI1(B)*FT2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP8(I,B,H,L)$H1(B,H)..
    INVVT(I,B,H,L) =L=
    LBI1(B)*FT2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP8INVVT(I,B,H,L) ;

LBP9(I,B,H,L)$H1(B,H)..
    INVVT(I,B,H,L) =L=
    UBI1(B)*FT2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*LBFV2(B,H) ;

LBP10(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJ(I,A,B,H,L) =G=
    LBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP11(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJ(I,A,B,H,L) =G=
    UBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP12(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJ(I,A,B,H,L) =L=
    LBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP13(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJ(I,A,B,H,L) =L=
    UBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP13FLOWVJ(I,A,B,H,L) ;

LBP14(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVT(I,A,B,H,L) =G=
    LBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP14FLOWVT(I,A,B,H,L) ;

LBP15(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVT(I,A,B,H,L) =G=
    UBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP16(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVT(I,A,B,H,L) =L=
    LBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP17(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVT(I,A,B,H,L) =L=
    UBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

* ===== for generating
cutting planes using Lagrangean
decomposition ===== *

LBP10Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ(I,A,B,H,L,Q) =G=
    LBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

*
LBP10Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    *FLOWVJQ(I,M,N,H,L,Q) =G=
    *LBFV1(A,B)*FL2(I,B,H,L) +
    *LBFV2(B,H)*FT1(A,B,L) -
    *LBFV1(A,B)*LBFV2(B,H) ;

LBP11Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ(I,A,B,H,L,Q) =G=
    UBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP12Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ(I,A,B,H,L,Q) =L=
    LBFV1(A,B)*FL2(I,B,H,L) +
UBFV2(B,H)*FT1(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP13Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ(I,A,B,H,L,Q) =L=
    UBFV1(A,B)*FL2(I,B,H,L) +
LBFV2(B,H)*FT1(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

LBP14Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ(I,A,B,H,L,Q) =G=
    LBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP15Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ(I,A,B,H,L,Q) =G=
    UBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP16Q(I,A,B,H,L,Q)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ(I,A,B,H,L,Q) =L=
    LBFV1(A,B)*FT2(B,H,L) +
UBFV2(B,H)*FL1(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

```

```

LBP17Q(I,A,B,H,L,Q)$(A1(A,B)$H1(B,H)).
      FLOWVTQ(I,A,B,H,L,Q) =L=
      UBFV1(A,B)*FT2(B,H,L) +
LBFV2(B,H)*FL1(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

* ===== sub-
structure Q1 (w/o Q index)
===== *

LBP1Q1(I,B,H,L)$H1(B,H)..
      INVJQ1(I,B,H,L) =G=
SUM(A$A1(A,B),FLOWVJQ1(I,A,B,H,L)) =E=
      INVVTQ1(I,B,H,L) +
SUM(A$A1(A,B),FLOWVTQ1(I,A,B,H,L))
;

LBP2Q1(I,B,H,L)$H1(B,H)..
      INVJQ1(I,B,H,L) =G=
      LBI1(B)*FL2Q1(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

LBP3Q1(I,B,H,L)$H1(B,H)..
      INVJQ1(I,B,H,L) =G=
      UBI1(B)*FL2Q1(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP4Q1(I,B,H,L)$H1(B,H)..
      INVJQ1(I,B,H,L) =L=
      LBI1(B)*FL2Q1(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP4INVJQ1(I,B,H,L) ;

LBP5Q1(I,B,H,L)$H1(B,H)..
      INVJQ1(I,B,H,L) =L=
      UBI1(B)*FL2Q1(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP5INVJQ1(I,B,H,L) ;

LBP6Q1(I,B,H,L)$H1(B,H)..
      INVVTQ1(I,B,H,L) =G=
      LBI1(B)*FT2Q1(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP6INVVTQ1(I,B,H,L) ;

LBP7Q1(I,B,H,L)$H1(B,H)..
      INVVTQ1(I,B,H,L) =G=
      UBI1(B)*FT2Q1(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP8Q1(I,B,H,L)$H1(B,H)..
      INVVTQ1(I,B,H,L) =L=
      LBI1(B)*FT2Q1(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP8INVVTQ1(I,B,H,L) ;

LBP9Q1(I,B,H,L)$H1(B,H)..
      INVVTQ1(I,B,H,L) =L=
      UBI1(B)*FT2Q1(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*LBFV2(B,H) ;

LBP10Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVJQ1(I,A,B,H,L) =G=
      LBFV1(A,B)*FL2Q1(I,B,H,L) +
LBFV2(B,H)*FT1Q1(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP11Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVJQ1(I,A,B,H,L) =G=
      UBFV1(A,B)*FL2Q1(I,B,H,L) +
UBFV2(B,H)*FT1Q1(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP12Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVJQ1(I,A,B,H,L) =L=
      LBFV1(A,B)*FL2Q1(I,B,H,L) +
UBFV2(B,H)*FT1Q1(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP13Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVJQ1(I,A,B,H,L) =L=
      UBFV1(A,B)*FL2Q1(I,B,H,L) +
LBFV2(B,H)*FT1Q1(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP13FLOWVJQ1(I,A,B,H,L) ;

LBP14Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVTQ1(I,A,B,H,L) =G=
      LBFV1(A,B)*FT2Q1(B,H,L) +
LBFV2(B,H)*FL1Q1(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP14FLOWVTQ1(I,A,B,H,L) ;

LBP15Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVTQ1(I,A,B,H,L) =G=
      UBFV1(A,B)*FT2Q1(B,H,L) +
UBFV2(B,H)*FL1Q1(I,A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP16Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVTQ1(I,A,B,H,L) =L=
      LBFV1(A,B)*FT2Q1(B,H,L) +
UBFV2(B,H)*FL1Q1(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP17Q1(I,A,B,H,L)$(A1(A,B)$H1(B,H))..
      FLOWVTQ1(I,A,B,H,L) =L=
      UBFV1(A,B)*FT2Q1(B,H,L) +
LBFV2(B,H)*FL1Q1(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

* ===== sub-
structure Q2 (w/o Q index)
===== *

LBP1Q2(I,B,H,L)$H1(B,H)..
      INVJQ2(I,B,H,L) +
SUM(A$A1(A,B),FLOWVJQ2(I,A,B,H,L)) =E=
      INVVTQ2(I,B,H,L) +
SUM(A$A1(A,B),FLOWVTQ2(I,A,B,H,L))
;

LBP2Q2(I,B,H,L)$H1(B,H)..
      INVJQ2(I,B,H,L) =G=
      LBI1(B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) ;

```

```

LBP3Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =G=
    UBI1(B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP4Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =L=
    LBI1(B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP4INVJQ2(I,B,H,L) ;

LBP5Q2(I,B,H,L)$H1(B,H)..
    INVJQ2(I,B,H,L) =L=
    UBI1(B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*ITOT(B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP5INVJQ2(I,B,H,L) ;

LBP6Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =G=
    LBI1(B)*FT2Q2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*LBFV2(B,H) +
SLAVARLBP6INVVTQ2(I,B,H,L) ;

LBP7Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =G=
    UBI1(B)*FT2Q2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*UBFV2(B,H) ;

LBP8Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =L=
    LBI1(B)*FT2Q2(B,H,L) +
UBFV2(B,H)*INVI(I,B,L-1) -
LBI1(B)*UBFV2(B,H) +
SLAVARLBP8INVVTQ2(I,B,H,L) ;

LBP9Q2(I,B,H,L)$H1(B,H)..
    INVVTQ2(I,B,H,L) =L=
    UBI1(B)*FT2Q2(B,H,L) +
LBFV2(B,H)*INVI(I,B,L-1) -
UBI1(B)*LBFV2(B,H) ;

LBP10Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =G=
    LBFV1(A,B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*FT1Q2(A,B,L) -
LBFV1(A,B)*LBFV2(B,H) ;

LBP11Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =G=
    UBFV1(A,B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*FT1Q2(A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP12Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =L=
    LBFV1(A,B)*FL2Q2(I,B,H,L) +
UBFV2(B,H)*FT1Q2(A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP13Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVJQ2(I,A,B,H,L) =L=
    UBFV1(A,B)*FL2Q2(I,B,H,L) +
LBFV2(B,H)*FT1Q2(A,B,L) -
UBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP13FLOWVJQ2(I,A,B,H,L) ;

LBP14Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =G=
    LBFV1(A,B)*FT2Q2(B,H,L) +
LBFV2(B,H)*FL1Q2(I,A,B,L) -
LBFV1(A,B)*LBFV2(B,H) +
SLAVARLBP14FLOWVTQ2(I,A,B,H,L) ;

LBP15Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =G=
    UBFV1(A,B)*FT2Q2(B,H,L) +
UBFV2(B,H)*FL1Q2(I,A,B,L) -
UBFV1(A,B)*UBFV2(B,H) ;

LBP16Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =L=
    LBFV1(A,B)*FT2Q2(B,H,L) +
UBFV2(B,H)*FL1Q2(I,A,B,L) -
LBFV1(A,B)*UBFV2(B,H) ;

LBP17Q2(I,A,B,H,L)$ (A1(A,B)$H1(B,H))..
    FLOWVTQ2(I,A,B,H,L) =L=
    UBFV1(A,B)*FT2Q2(B,H,L) +
LBFV2(B,H)*FL1Q2(I,A,B,L) -
UBFV1(A,B)*LBFV2(B,H) ;

LBP18(M,K,L)$K1(M,K)..
    FLTQ1(M,K,L) =E=
    FLTQ2(M,K,L) ;

LBP19(I,M,K,L)$K1(M,K)..
    FLQ1(I,M,K,L) =E=
    FLQ1(I,M,K,L) ;

LBP20(M,K,L)$K1(M,K)..
    TSQ1(M,K,L) =E=
    TSQ2(M,K,L) ;

LBP21(M,K,L)$K1(M,K)..
    TEQ1(M,K,L) =E=
    TEQ2(M,K,L) ;

LBP22(M,K,L)$K1(M,K)..
    WBQ1(M,K,L) =E=
    WBQ2(M,K,L) ;

**(vii) Inventory Bounds

TC26(B,L)$ (ORD(L) GT 1)..
    ITOT(B,L-
1)+SUM(ASA1(A,B),FT1(A,B,L)) =L=
    UBI1(B) ;

**(viii) Bounds on Components Fractions
inside a Tank

TC27(I,B,L)..
    LBI(I,B)*ITOT(B,L) =L=
    INVI(I,B,L) +
SLAVARTC27INVI(I,B,L) ;

```

```

TC28(I,B,L)..
    INVI(I,B,L) =L=
    UB1(I,B)*ITOT(B,L)
;

TC29(I,B,H,L)$H1(B,H)..
    LB1(I,B)*FT2(B,H,L) =L=
    FL2(I,B,H,L)
;

TC30(I,B,H,L)$H1(B,H)..
    FL2(I,B,H,L) =L=
    UB1(I,B)*FT2(B,H,L)
;

** (ix) Crude-mix Demand Constraints

TC31(G)..
SUM(D$D3(D,G),SUM(L,FT2(G,D,L))) =E=
    DM(G)
;

** (x) Bound Strengthening Cuts

TC32(I,B,H,L)$H1(B,H)..
    ITOT(B,L-1)*FL2(I,B,H,L) +
SUM(A$A1(A,B),FT1(A,B,L)*FL2(I,B,H,L))
=E=
    INVI(I,B,L-
1)*FT2(B,H,L)+SUM(A$A1(A,B),FL1(I,A,B,
L)*FT2(B,H,L))
;

TC33(B,H,L)$H1(B,H)..
    SUM(I,ITOT(B,L-
1)*FL2(I,B,H,L)) =E=
    ITOT(B,L-1)*FT2(B,H,L)
;

TC34(A,B,H,L)$A1(A,B)$H1(B,H)..
SUM(I,FT1(A,B,L)*FL2(I,B,H,L)) =E=
    FT1(A,B,L)*FT2(B,H,L)
;

TC35(B,H,L)$H1(B,H)..
    SUM(I,INVI(I,B,L-
1)*FT2(B,H,L)) =E=
    ITOT(B,L-1)*FT2(B,H,L)
;

TC36(A,B,H,L)$A1(A,B)$H1(B,H)..
SUM(I,FL1(I,A,B,L)*FT2(B,H,L)) =E=
    FT1(A,B,L)*FT2(B,H,L)
;

1 + SLAVARDC2WB2(G,L)
;

DCUB1(D,L)..
    SUM(G$G1(G,D),WBUB2(G,D,L))
=L=
    1 + SLAVARDCUB1WBUB2(D,L)
;

DCUB2(G,L)..
    SUM(D$D3(D,G),WBUB2(G,D,L))
=L=
    1 + SLAVARDCUB2WBUB2(G,L)
;

** (ii) Continuous Operation Constraints

DC3(D)..
SUM(L,SUM(G$G1(G,D),TE2(G,D,L))-
    TS2(G,D,L)) =E=
    THOR
;

DC4(G,GDASH,D,L)$ORD(L) LT
CARD(L)$ORD(G) NE
ORD(GDASH)$G1(G,D))..
    TS2(G,D,L+1) =G=
    TE2(GDASH,D,L) - THOR*(1-
WB2(GDASH,D,L))
;

DC5(G,GDASH,D,L)$ORD(L) LT
CARD(L)$ORD(G) NE
ORD(GDASH)$G1(G,D))..
    TS2(G,D,L+1) =L=
    TE2(GDASH,D,L) + THOR*(1-
WB2(GDASH,D,L))
;

DCUB4(G,GDASH,D,L)$ORD(L) LT
CARD(L)$ORD(G) NE
ORD(GDASH)$G1(G,D))..
    TS2(G,D,L+1) =G=
    TE2(GDASH,D,L) - THOR*(1-
WBUB2(GDASH,D,L))
;

DCUB5(G,GDASH,D,L)$ORD(L) LT
CARD(L)$ORD(G) NE
ORD(GDASH)$G1(G,D))..
    TS2(G,D,L+1) =L=
    TE2(GDASH,D,L) + THOR*(1-
WBUB2(GDASH,D,L))
;

*3. CRUDE SUPPLY STREAM CONSTRAINTS

** (i) Timing Constraints

CC1(U,Y,L)$Y1(Y,U)..
    TSI(U) =L=
    TS1(U,Y,L) + THOR*(1-
WB1(U,Y,L))
;

CC2(U,Y,L)$Y1(Y,U)..
    TEO(U) =G=
    TE1(U,Y,L) - THOR*(1-
WB1(U,Y,L))
;

* =====
2. DISTILLATION UNIT CONSTRAINTS
===== *

** (i) Allocation Constraints

DC1(D,L)..
    SUM(G$G1(G,D),WB2(G,D,L)) =L=
    1 + SLAVARDC1WB2(D,L)
;

DC2(G,L)..
    SUM(D$D3(D,G),WB2(G,D,L)) =L=

```

```

CCUB1(U,Y,L)$Y1(Y,U)..
    TSI(U) =L=
    TSI(U,Y,L) + THOR*(1-
WBUB1(U,Y,L))
;

CCUB2(U,Y,L)$Y1(Y,U)..
    TEO(U) =G=
    TE1(U,Y,L) - THOR*(1-
WBUB1(U,Y,L))
;

**(ii)Overall Mass Balances

CC3(U)..

SUM(L,SUM(Y$Y1(Y,U),FT1(U,Y,L))) =E=
    VCRUDE(U)
;

**(iii)Component Balances

CC4(I,U,Y,L)$Y1(Y,U)..
    FL1(I,U,Y,L) =E=
    FRACT(I,U)*FT1(U,Y,L)
;

*4.VARIABLE BOUNDS

VB1(I,B,L)..
    INVI(I,B,L) =L=
    UBI1(B)
;

VB2(B,L)..
    LB11(B) =L=
    ITOT(B,L)
;

VB3(B,L)..
    ITOT(B,L) =L=
    UBI1(B)
;

VB4(I,A,B,L)$A1(A,B)..
    FL1(I,A,B,L) =L=
    UBFV1(A,B)
;

VB5(I,B,H,L)$H1(B,H)..
    FL2(I,B,H,L) =L=
    UBFV2(B,H)
;

VB6(A,B,L)$A1(A,B)..
    FT1(A,B,L) =L=
    UBFV1(A,B)
;

VB7(B,H,L)$H1(B,H)..
    FT2(B,H,L) =L=
    UBFV2(B,H)
;

VB8(A,B,L)$A1(A,B)..
    TSI(A,B,L) =L=
    THOR
;

VB9(A,B,L)$A1(A,B)..
    TE1(A,B,L) =L=

```

```

THOR
;

VB10(B,H,L)$H1(B,H)..
    TS2(B,H,L) =L=
    THOR
;

VB11(B,H,L)$H1(B,H)..
    TE2(B,H,L) =L=
    THOR
;

VB12(U)..
    TARR(U) =L=
    TSI(U)
;

VB13(U)..
    TSI(U) =L=
    THOR
;

VB14(U)..
    TARR(U) =L=
    TEO(U)
;

VB15(U)..
    TEO(U) =L=
    THOR
;

* =====
GENERATE VALID LINEAR CUTS
===== *

VLC1..
    ZZQ1.L =L=
    CSEA*SUM(U1,TSI(U1)-TARR(U1))
+ CUNLOAD*SUM(U1,TEO(U1)-TSI(U1)) +
    THOR*(SUM(B1,CINV(B1))*SUM(B1,SUM(L,(I
TOT(B1,L)))) +
    SUM(B1,CINV(B1))*SUM(B1,SUM(L,SUM(A,FT
1(A,B1,L)))) +
    SUM(B1,CINV(B1))*SUM(B1,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B1,L))) +
    THOR*SUM(B1,CINV(B1))*SUM(B2,INITITOT(
B2)))/ (2*NE + 1) +
    CSET*(SUM(D1,SUM(G$G1(G,D1)),SUM(L,WB2(
G,D1,L)))) - ND1) +
    SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ(M,K,L)))) +
    SUM(I,SUM(A,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ(I,M,K,L)))) +
    SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ(M,K,L)))) +
    SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ(M,K,L)))) +
    SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ(M,K,L))))
;

VLC2..
    ZZQ2.L =L=
    CSEA*SUM(U2,TSI(U2)-TARR(U2))
+ CUNLOAD*SUM(U2,TEO(U2)-TSI(U2)) +

```

```

THOR*(SUM(B2,CINV(B2))*SUM(B2,SUM(L,(I
TOT(B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L,SUM(A,FT
1(A,B2,L)))) +
SUM(B2,CINV(B2))*SUM(B2,SUM(L$(ORD(L)
LT CARD(L)),ITOT(B2,L))) +
THOR*SUM(B2,CINV(B2))*SUM(B2,INITITOT(
B2)) / (2*NE + 1) +
CSET*(SUM(D2,SUM(G$G1(G,D2),SUM(L,WB2(
G,D2,L)))) - ND2) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMFLT(M,K,L)
*FLTQ(M,K,L)))) -
SUM(I,SUM(M,SUM(K$K1(M,K),SUM(L,LMFL(I
,M,K,L)*FLQ(I,M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTS(M,K,L)*
TSQ(M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMTE(M,K,L)*
TEQ(M,K,L)))) -
SUM(M,SUM(K$K1(M,K),SUM(L,LMWB(M,K,L)*
WBQ(M,K,L))))
;
* ===== FOR DUPLICATING
EQUATIONS AND VARIABLES IN Q1
===== *
TC1Q1(A,B,L)$A1(A,B)..
FT1Q1(A,B,L) =L=
UBFV1(A,B)*WB1Q1(A,B,L)
;
TC2Q1(B,H,L)$H1(B,H)..
FT2Q1(B,H,L) =L=
UBFV2(B,H)*WB2Q1(B,H,L)
;
TC3Q1(A,B,L)$A1(A,B)..
UBF1(A,B)*(TE1Q1(A,B,L) -
TS1Q1(A,B,L)) + UBF1(A,B)*THOR*(1-
WB1Q1(A,B,L)) =G=
FT1Q1(A,B,L) +
SLAVARTC3FT1Q1(A,B,L)
;
TC4Q1(B,H,L)$H1(B,H)..
UBF2(B,H)*(TE2Q1(B,H,L) -
TS2Q1(B,H,L)) + UBF2(B,H)*THOR*(1-
WB2Q1(B,H,L)) =G=
FT2Q1(B,H,L)
;
TC5Q1(A,Y,L)$A2(A,Y)..
LBF1(A,Y)*(TE1Q1(A,Y,L) -
TS1Q1(A,Y,L)) - LBF1(A,Y)*THOR*(1-
WB1Q1(A,Y,L)) =L=
FT1Q1(A,Y,L)
;
TC6Q1(Y,H,L)$H2(Y,H)..
LBF2(Y,H)*(TE2Q1(Y,H,L) -
TS2Q1(Y,H,L)) - LBF2(Y,H)*THOR*(1-
WB2Q1(Y,H,L)) =L=
FT2Q1(Y,H,L)
;
TC7Q1(G,H,L)$H3(G,H)..
LBF2(G,H)*(TE2Q1(G,H,L) -
TS2Q1(G,H,L)) =L=
FT2Q1(G,H,L)
;
FT2Q1(G,H,L)
;
TC8Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TS1Q1(A,B,L+1) =G=
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L))
;
TC9Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TS1Q1(A,B,L+1) =G=
TS1Q1(A,B,L)
;
TC10Q1(A,B,L)$ (ORD(L) LT
CARD(L)$A1(A,B))..
TE1Q1(A,B,L+1) =G=
TE1Q1(A,B,L)
;
TC11Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TS2Q1(B,H,L+1) =G=
TE2Q1(B,H,L) - THOR*(1-
WB2Q1(B,H,L))
;
TC12Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TS2Q1(B,H,L+1) =G=
TS2Q1(B,H,L)
;
TC13Q1(B,H,L)$ (ORD(L) LT
CARD(L)$H1(B,H))..
TE2Q1(B,H,L+1) =G=
TE2Q1(B,H,L)
;
TC14Q1(A,ADASH,B,L)$ (ORD(L) LT
CARD(L)$ (ORD(A) NE
ORD(ADASH)$A1(A,B)))..
TS1Q1(A,B,L+1) =G=
TE1Q1(ADASH,B,L) - THOR*(1-
WB1Q1(ADASH,B,L))
;
TC15Q1(A,B,H,L)$ (ORD(L) LT
CARD(L)$ (A1(A,B)$H1(B,H)))..
TS1Q1(A,B,L+1) =G=
TE2Q1(B,H,L) - THOR*(1-
WB2Q1(B,H,L))
;
TC16Q1(A,B,H,L)$ (ORD(L) LT
CARD(L)$ (A1(A,B)$H1(B,H)))..
TS2Q1(B,H,L+1) =G=
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L))
;
TC17Q1(B,H,HDASH,L)$ (ORD(L) LT
CARD(L)$ (ORD(H) NE
ORD(HDASH)$H1(B,H)))..
TS2Q1(B,H,L+1) =G=
TE2Q1(B,HDASH,L) - THOR*(1-
WB2Q1(B,HDASH,L))
;
TC18Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
TE1Q1(A,B,L) - THOR*(1-
WB1Q1(A,B,L)) =L=
TS2Q1(B,H,L) + THOR*(1-
WB2Q1(B,H,L))
;

```

```

TC19Q1(B,L)..
    ITOT(B,L-1) +
SUM(A$A1(A,B),FT1Q1(A,B,L)) =E=
    ITOT(B,L) +
SUM(H$H1(B,H),FT2Q1(B,H,L)) +
SLAVARTC19ITOTQ1(B,L)
;
*- ONEG1Q1(B,L) + OPOS1Q1(B,L)
;

TC20Q1(B)..
    ITOT(B,'L0') =E=
    INITITOT(B)
;

TC21Q1(I,B,L)..
    INVI(I,B,L-1) +
SUM(A$A1(A,B),FL1Q1(I,A,B,L)) =E=
    INVI(I,B,L)
+SUM(H$H1(B,H),FL2Q1(I,B,H,L)) +
SLAVARTC21INVIQ1(I,B,L)
;

TC22Q1(I,B)..
    INVI(I,B,'L0') =E=
    INITINVI(I,B)
;

TC23Q1(A,B,L)$A1(A,B)..
    FT1Q1(A,B,L) =E=
    SUM(I,FL1Q1(I,A,B,L)) +
SLAVARTC23FT1Q1(A,B,L)
;

TC24Q1(B,H,L)$H1(B,H)..
    FT2Q1(B,H,L) =E=
    SUM(I,FL2Q1(I,B,H,L)) +
SLAVARTC24FT2Q1(B,H,L)
;

TC26Q1(B,L)$ (ORD(L) GT 1)..
    ITOT(B,L-
1)+SUM(A$A1(A,B),FT1Q1(A,B,L)) =L=
    UB11(B)
;

TC27Q1(I,B,L)..
    LB1(I,B)*ITOT(B,L) =L=
    INVI(I,B,L) +
SLAVARTC27INVIQ1(I,B,L)
;

* - ONEG1Q1(I,B,L) + OPOS1Q1(I,B,L)
;

TC28Q1(I,B,L)..
    INVI(I,B,L) =L=
    UB1(I,B)*ITOT(B,L)
;

TC29Q1(I,B,H,L)$H1(B,H)..
    LB1(I,B)*FT2Q1(B,H,L) =L=
    FL2Q1(I,B,H,L)
;

TC30Q1(I,B,H,L)$H1(B,H)..
    FL2Q1(I,B,H,L) =L=
    UB1(I,B)*FT2Q1(B,H,L)
;

TC31Q1(G)..
SUM(D$D3(D,G),SUM(L,FT2Q1(G,D,L))) =E=
    DM(G)
;

TC32Q1(I,B,H,L)$H1(B,H)..
    ITOT(B,L-1)*FL2Q1(I,B,H,L) +
SUM(A$A1(A,B),FT1Q1(A,B,L))*FL2Q1(I,B,H
,L)) =E=
    INVI(I,B,L-
1)*FT2Q1(B,H,L)+SUM(A$A1(A,B),FL1Q1(I,
A,B,L))*FT2Q1(B,H,L)
;

TC33Q1(B,H,L)$H1(B,H)..
    SUM(I,ITOT(B,L-
1)*FL2Q1(I,B,H,L)) =E=
    ITOT(B,L-1)*FT2Q1(B,H,L)
;

TC34Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
SUM(I,FT1Q1(A,B,L)*FL2Q1(I,B,H,L)) =E=
    FT1Q1(A,B,L)*FT2Q1(B,H,L)
;

TC35Q1(B,H,L)$H1(B,H)..
    SUM(I,INVI(I,B,L-
1)*FT2Q1(B,H,L)) =E=
    ITOT(B,L-1)*FT2Q1(B,H,L)
;

TC36Q1(A,B,H,L)$ (A1(A,B)$H1(B,H))..
SUM(I,FL1Q1(I,A,B,L)*FT2Q1(B,H,L)) =E=
    FT1Q1(A,B,L)*FT2Q1(B,H,L)
;

DC1Q1(D,L)..
    SUM(G$G1(G,D),WB2Q1(G,D,L))
=L=
    1 + SLAVARDC1WB2Q1(D,L)
;

DC2Q1(G,L)..
    SUM(D$D3(D,G),WB2Q1(G,D,L))
=L=
    1 + SLAVARDC2WB2Q2(G,L)
;

DC3Q1(D)..
SUM(L,SUM(G$G1(G,D),TE2Q1(G,D,L) -
TS2Q1(G,D,L))) =E=
    THOR
;

DC4Q1(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
    TS2Q1(G,D,L+1) =G=
    TE2Q1(GDASH,D,L) - THOR*(1-
WB2Q1(GDASH,D,L))
;

DC5Q1(G,GDASH,D,L)$ (ORD(L) LT
CARD(L)$ (ORD(G) NE
ORD(GDASH)$G1(G,D)))..
    TS2Q1(G,D,L+1) =L=
    TE2Q1(GDASH,D,L) + THOR*(1-
WB2Q1(GDASH,D,L))
;

CC1Q1(U,Y,L)$Y1(Y,U)..
    TSI(U) =L=
    TS1Q1(U,Y,L) + THOR*(1-
WB1Q1(U,Y,L))
;

CC2Q1(U,Y,L)$Y1(Y,U)..
    TEO(U) =G=

```

```

      TE1Q1 (U, Y, L) - THOR*(1-
WB1Q1 (U, Y, L))
;
      CC3Q1 (U) ..
SUM (L, SUM (Y$Y1 (Y, U), FT1Q1 (U, Y, L))) =E=
      VCRUDE (U)
;
      CC4Q1 (I, U, Y, L) $Y1 (Y, U) ..
      FL1Q1 (I, U, Y, L) =E=
      FRACT (I, U)*FT1Q1 (U, Y, L)
;
      VB1Q1 (I, B, L) ..
      INVI (I, B, L) =L=
      UBI1 (B)
;
      VB2Q1 (B, L) ..
      LBI1 (B) =L=
      ITOT (B, L)
;
      VB3Q1 (B, L) ..
      ITOT (B, L) =L=
      UBI1 (B)
;
      VB4Q1 (I, A, B, L) $A1 (A, B) ..
      FL1Q1 (I, A, B, L) =L=
      UBFV1 (A, B)
;
      VB5Q1 (I, B, H, L) $H1 (B, H) ..
      FL2Q1 (I, B, H, L) =L=
      UBFV2 (B, H)
;
      VB6Q1 (A, B, L) $A1 (A, B) ..
      FT1Q1 (A, B, L) =L=
      UBFV1 (A, B)
;
      VB7Q1 (B, H, L) $H1 (B, H) ..
      FT2Q1 (B, H, L) =L=
      UBFV2 (B, H)
;
      VB8Q1 (A, B, L) $A1 (A, B) ..
      TS1Q1 (A, B, L) =L=
      THOR
;
      VB9Q1 (A, B, L) $A1 (A, B) ..
      TE1Q1 (A, B, L) =L=
      THOR
;
      VB10Q1 (B, H, L) $H1 (B, H) ..
      TS2Q1 (B, H, L) =L=
      THOR
;
      VB11Q1 (B, H, L) $H1 (B, H) ..
      TE2Q1 (B, H, L) =L=
      THOR
;
      VB12Q1 (U) ..
      TARR (U) =L=
      TSI (U)
;
      VB13Q1 (U) ..
      TSI (U) =L=
      THOR
;
      VB14Q1 (U) ..
      TARR (U) =L=
      TEO (U)
;
      VB15Q1 (U) ..
      TEO (U) =L=
      THOR
;
* ===== FOR DUPLICATING
EQUATIONS AND VARIABLES IN Q2
===== *
      TC1Q2 (A, B, L) $A1 (A, B) ..
      FT1Q2 (A, B, L) =L=
      UBFV1 (A, B)*WB1Q2 (A, B, L)
;
      TC2Q2 (B, H, L) $H1 (B, H) ..
      FT2Q2 (B, H, L) =L=
      UBFV2 (B, H)*WB2Q2 (B, H, L)
;
      TC3Q2 (A, B, L) $A1 (A, B) ..
      UBF1 (A, B)*(TE1Q2 (A, B, L) -
      TS1Q2 (A, B, L)) + UBF1 (A, B)*THOR*(1-
      WB1Q2 (A, B, L)) =G=
      FT1Q2 (A, B, L) +
      SLAVARTC3FT1Q2 (A, B, L)
;
      TC4Q2 (B, H, L) $H1 (B, H) ..
      UBF2 (B, H)*(TE2Q2 (B, H, L) -
      TS2Q2 (B, H, L)) + UBF2 (B, H)*THOR*(1-
      WB2Q2 (B, H, L)) =G=
      FT2Q2 (B, H, L)
;
      TC5Q2 (A, Y, L) $A2 (A, Y) ..
      LBF1 (A, Y)*(TE1Q2 (A, Y, L) -
      TS1Q2 (A, Y, L)) - LBF1 (A, Y)*THOR*(1-
      WB1Q2 (A, Y, L)) =L=
      FT1Q2 (A, Y, L)
;
      TC6Q2 (Y, H, L) $H2 (Y, H) ..
      LBF2 (Y, H)*(TE2Q2 (Y, H, L) -
      TS2Q2 (Y, H, L)) - LBF2 (Y, H)*THOR*(1-
      WB2Q2 (Y, H, L)) =L=
      FT2Q2 (Y, H, L)
;
      TC7Q2 (G, H, L) $H3 (G, H) ..
      LBF2 (G, H)*(TE2Q2 (G, H, L) -
      TS2Q2 (G, H, L)) =L=
      FT2Q2 (G, H, L)
;
      TC8Q2 (A, B, L) $ (ORD (L)) LT
      CARD (L) $A1 (A, B) ..
      TS1Q2 (A, B, L+1) =G=
      TE1Q2 (A, B, L) - THOR*(1-
      WB1Q2 (A, B, L))
;
      TC9Q2 (A, B, L) $ (ORD (L)) LT
      CARD (L) $A1 (A, B) ..
      TS1Q2 (A, B, L+1) =G=
      TS1Q2 (A, B, L)
;

```



```

TC10Q2 (A,B,L) $ (ORD(L) LT
CARD(L) $A1 (A,B) ) . .
    TE1Q2 (A,B,L+1) =G=
    TE1Q2 (A,B,L)
;

TC11Q2 (B,H,L) $ (ORD(L) LT
CARD(L) $H1 (B,H) ) . .
    TS2Q2 (B,H,L+1) =G=
    TE2Q2 (B,H,L) - THOR* (1-
WB2Q2 (B,H,L) )
;

TC12Q2 (B,H,L) $ (ORD(L) LT
CARD(L) $H1 (B,H) ) . .
    TS2Q2 (B,H,L+1) =G=
    TS2Q2 (B,H,L)
;

TC13Q2 (B,H,L) $ (ORD(L) LT
CARD(L) $H1 (B,H) ) . .
    TE2Q2 (B,H,L+1) =G=
    TE2Q2 (B,H,L)
;

TC14Q2 (A,ADASH,B,L) $ (ORD(L) LT
CARD(L) $ (ORD(A) NE
ORD(ADASH) $A1 (A,B) ) ) . .
    TS1Q2 (A,B,L+1) =G=
    TE1Q2 (ADASH,B,L) - THOR* (1-
WB1Q2 (ADASH,B,L) )
;

TC15Q2 (A,B,H,L) $ (ORD(L) LT
CARD(L) $ (A1 (A,B) $H1 (B,H) ) ) . .
    TS1Q2 (A,B,L+1) =G=
    TE2Q2 (B,H,L) - THOR* (1-
WB2Q2 (B,H,L) )
;

TC16Q2 (A,B,H,L) $ (ORD(L) LT
CARD(L) $ (A1 (A,B) $H1 (B,H) ) ) . .
    TS2Q2 (B,H,L+1) =G=
    TE1Q2 (A,B,L) - THOR* (1-
WB1Q2 (A,B,L) )
;

TC17Q2 (B,H,HDASH,L) $ (ORD(L) LT
CARD(L) $ (ORD(H) NE
ORD(HDASH) $H1 (B,H) ) ) . .
    TS2Q2 (B,H,L+1) =G=
    TE2Q2 (B,HDASH,L) - THOR* (1-
WB2Q2 (B,HDASH,L) )
;

TC18Q2 (A,B,H,L) $ (A1 (A,B) $H1 (B,H) ) . .
    TE1Q2 (A,B,L) - THOR* (1-
WB1Q2 (A,B,L) ) =L=
    TS2Q2 (B,H,L) + THOR* (1-
WB2Q2 (B,H,L) )
;

TC19Q2 (B,L) . .
    ITOT (B,L-1) +
SUM (A$A1 (A,B) , FT1Q2 (A,B,L) ) =E=
    ITOT (B,L) +
SUM (H$H1 (B,H) , FT2Q2 (B,H,L) ) +
SLAVARTC19ITOTQ2 (B,L)
;

TC20Q2 (B) . .
    ITOT (B, 'L0' ) =E=
    INITITOT (B)
;

TC21Q2 (I,B,L) . .
    INVI (I,B,L-1) +
SUM (A$A1 (A,B) , FL1Q2 (I,A,B,L) ) =E=
    INVI (I,B,L)
+SUM (H$H1 (B,H) , FL2Q2 (I,B,H,L) ) +
SLAVARTC21INVIQ2 (I,B,L)
;

TC22Q2 (I,B) . .
    INVI (I,B, 'L0' ) =E=
    INITINVI (I,B)
;

TC23Q2 (A,B,L) $A1 (A,B) . .
    FT1Q2 (A,B,L) =E=
    SUM (I, FL1Q2 (I,A,B,L) ) +
SLAVARTC23FT1Q2 (A,B,L)
;

TC24Q2 (B,H,L) $H1 (B,H) . .
    FT2Q2 (B,H,L) =E=
    SUM (I, FL2Q2 (I,B,H,L) ) +
SLAVARTC24FT2Q2 (B,H,L)
;

TC26Q2 (B,L) $ (ORD(L) GT 1) . .
    ITOT (B,L-
1) +SUM (A$A1 (A,B) , FT1Q2 (A,B,L) ) =L=
    UB1 (B)
;

TC27Q2 (I,B,L) . .
    LB1 (I,B) *ITOT (B,L) =L=
    INVI (I,B,L) +
SLAVARTC27INVIQ2 (I,B,L)
;

TC28Q2 (I,B,L) . .
    INVI (I,B,L) =L=
    UB1 (I,B) *ITOT (B,L)
;

TC29Q2 (I,B,H,L) $H1 (B,H) . .
    LB1 (I,B) *FT2Q2 (B,H,L) =L=
    FL2Q2 (I,B,H,L)
;

TC30Q2 (I,B,H,L) $H1 (B,H) . .
    FL2Q2 (I,B,H,L) =L=
    UB1 (I,B) *FT2Q2 (B,H,L)
;

TC31Q2 (G) . .
    SUM (D$D3 (D,G) , SUM (L, FT2Q2 (G,D,L) ) ) =E=
    DM (G)
;

TC32Q2 (I,B,H,L) $H1 (B,H) . .
    ITOT (B,L-1) *FL2Q2 (I,B,H,L) +
SUM (A$A1 (A,B) , FT1Q2 (A,B,L) *FL2Q2 (I,B,H
,L) ) =E=
    INVI (I,B,L-
1) *FT2Q2 (B,H,L) +SUM (A$A1 (A,B) , FL1Q2 (I,
A,B,L) *FT2Q2 (B,H,L) )
;

TC33Q2 (B,H,L) $H1 (B,H) . .
    SUM (I, ITOT (B,L-
1) *FL2Q2 (I,B,H,L) ) =E=
    ITOT (B,L-1) *FT2Q2 (B,H,L)
;

TC34Q2 (A,B,H,L) $ (A1 (A,B) $H1 (B,H) ) . .
    SUM (I, FT1Q2 (A,B,L) *FL2Q2 (I,B,H,L) ) =E=

```

```

      FT1Q2 (A, B, L) * FT2Q2 (B, H, L)
;
TC35Q2 (B, H, L) $H1 (B, H) ..
      SUM (I, INVI (I, B, L-
1) * FT2Q2 (B, H, L) =E=
      ITOT (B, L-1) * FT2Q2 (B, H, L)
;
TC36Q2 (A, B, H, L) $ (A1 (A, B) $H1 (B, H) ) ..
SUM (I, FL1Q2 (I, A, B, L) * FT2Q2 (B, H, L) =E=
      FT1Q2 (A, B, L) * FT2Q2 (B, H, L)
;
DC1Q2 (D, L) ..
      SUM (G$G1 (G, D), WB2Q2 (G, D, L) )
=L=
      1 + SLAVARDC1WB2Q2 (D, L)
;
DC2Q2 (G, L) ..
      SUM (D$D3 (D, G), WB2Q2 (G, D, L) )
=L=
      1 + SLAVARDC2WB2Q2 (G, L)
;
DC3Q2 (D) ..
SUM (L, SUM (G$G1 (G, D), TE2Q2 (G, D, L) -
TS2Q2 (G, D, L) ) ) =E=
      THOR
;
DC4Q2 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D) ) ) ..
      TS2Q2 (G, D, L+1) =G=
      TE2Q2 (GDASH, D, L) - THOR * (1-
WB2Q2 (GDASH, D, L) )
;
DC5Q2 (G, GDASH, D, L) $ (ORD (L) LT
CARD (L) $ (ORD (G) NE
ORD (GDASH) $G1 (G, D) ) ) ..
      TS2Q2 (G, D, L+1) =L=
      TE2Q2 (GDASH, D, L) + THOR * (1-
WB2Q2 (GDASH, D, L) )
;
CC1Q2 (U, Y, L) $Y1 (Y, U) ..
      TSI (U) =L=
      TS1Q2 (U, Y, L) + THOR * (1-
WB1Q2 (U, Y, L) )
;
CC2Q2 (U, Y, L) $Y1 (Y, U) ..
      TEO (U) =G=
      TE1Q2 (U, Y, L) - THOR * (1-
WB1Q2 (U, Y, L) )
;
CC3Q2 (U) ..
SUM (L, SUM (Y$Y1 (Y, U), FT1Q2 (U, Y, L) ) ) =E=
      VCRUDE (U)
;
CC4Q2 (I, U, Y, L) $Y1 (Y, U) ..
      FL1Q2 (I, U, Y, L) =E=
      FRACT (I, U) * FT1Q2 (U, Y, L)
;
VB1Q2 (I, B, L) ..
      INVI (I, B, L) =L=

```

```

      UBI1 (B)
;
VB2Q2 (B, L) ..
      LB11 (B) =L=
      ITOT (B, L)
;
VB3Q2 (B, L) ..
      ITOT (B, L) =L=
      UBI1 (B)
;
VB4Q2 (I, A, B, L) $A1 (A, B) ..
      FL1Q2 (I, A, B, L) =L=
      UBFV1 (A, B)
;
VB5Q2 (I, B, H, L) $H1 (B, H) ..
      FL2Q2 (I, B, H, L) =L=
      UBFV2 (B, H)
;
VB6Q2 (A, B, L) $A1 (A, B) ..
      FT1Q2 (A, B, L) =L=
      UBFV1 (A, B)
;
VB7Q2 (B, H, L) $H1 (B, H) ..
      FT2Q2 (B, H, L) =L=
      UBFV2 (B, H)
;
VB8Q2 (A, B, L) $A1 (A, B) ..
      TS1Q2 (A, B, L) =L=
      THOR
;
VB9Q2 (A, B, L) $A1 (A, B) ..
      TE1Q2 (A, B, L) =L=
      THOR
;
VB10Q2 (B, H, L) $H1 (B, H) ..
      TS2Q2 (B, H, L) =L=
      THOR
;
VB11Q2 (B, H, L) $H1 (B, H) ..
      TE2Q2 (B, H, L) =L=
      THOR
;
VB12Q2 (U) ..
      TARR (U) =L=
      TSI (U)
;
VB13Q2 (U) ..
      TSI (U) =L=
      THOR
;
VB14Q2 (U) ..
      TARR (U) =L=
      TEO (U)
;
VB15Q2 (U) ..
      TEO (U) =L=
      THOR
;
===== LAGRANGEAN
ALGORITHM =====

```

```

$ontext
1. Decompose into Q1 and Q2
2. Solve Q1 to obtain objective value
of ZZQ1
3. Use value of ZZQ1 to generate
linear cut
4. repeat for Q2
5. solve MR00 with the two linear cuts
6. Solve CRUDE_UB

$offtext

MODEL CRUDE_LB1
* to generate linear cut for Q1
/
TC1Q1
TC2Q1
TC3Q1
TC4Q1
TC5Q1
TC6Q1
TC7Q1
TC8Q1
TC9Q1
TC10Q1
TC11Q1
TC12Q1
TC13Q1
TC14Q1
TC15Q1
TC16Q1
TC17Q1
TC18Q1
TC19Q1
TC20Q1
TC21Q1
TC22Q1
TC23Q1
TC24Q1

TC26Q1
TC27Q1
TC28Q1
TC29Q1
TC30Q1
TC31Q1
DC1Q1
DC2Q1
DC3Q1
DC4Q1
DC5Q1
CC1Q1
CC2Q1
CC3Q1
CC4Q1
VB1Q1
VB2Q1
VB3Q1
VB4Q1
VB5Q1
VB6Q1
VB7Q1
VB8Q1
VB9Q1
VB10Q1
VB11Q1
VB12Q1
VB13Q1
VB14Q1
VB15Q1

LBP1Q1

LBP2Q1
LBP3Q1
LBP4Q1
LBP5Q1

LBP6Q1
LBP7Q1
LBP8Q1
LBP9Q1
LBP10Q1
LBP11Q1
LBP12Q1

LBP13Q1

LBP14Q1
LBP15Q1
LBP16Q1
LBP17Q1
LBP18
LBP19
LBP20
LBP21
LBP22
*OBJF
OBJQ1
*OBJQ2
/
;

MODEL CRUDE_LB2
* to generate linear cut for Q2
/
TC1Q2
TC2Q2
TC3Q2
TC4Q2
TC5Q2
TC6Q2
TC7Q2
TC8Q2
TC9Q2
TC10Q2
TC11Q2
TC12Q2
TC13Q2
TC14Q2
TC15Q2
TC16Q2
TC17Q2
TC18Q2
TC19Q2
TC20Q2
TC21Q2
TC22Q2
TC23Q2
TC24Q2

TC26Q2
TC27Q2
TC28Q2
TC29Q2
TC30Q2
TC31Q2
DC1Q2
DC2Q2
DC3Q2
DC4Q2
DC5Q2
CC1Q2
CC2Q2
CC3Q2
CC4Q2
VB1Q2

```

VB2Q2	
VB3Q2	TC26
VB4Q2	TC27
VB5Q2	TC28
VB6Q2	TC29
VB7Q2	TC30
VB8Q2	TC31
VB9Q2	DC1
VB10Q2	DC2
VB11Q2	DC3
VB12Q2	DC4
VB13Q2	DC5
VB14Q2	CC1
VB15Q2	CC2
	CC3
	CC4
LBP1Q2	VB1
LBP2Q2	VB2
LBP3Q2	VB3
LBP4Q2	VB4
LBP5Q2	VB5
LBP6Q2	VB6
LBP7Q2	VB7
LBP8Q2	VB8
LBP9Q2	VB9
LBP10Q2	VB10
LBP11Q2	VB11
LBP12Q2	VB12
	VB13
LBP13Q2	VB14
	VB15
LBP14Q2	LBP1
LBP15Q2	LBP2
LBP16Q2	LBP3
LBP17Q2	LBP4
LBP18	LBP5
LBP19	LBP6
LBP20	LBP7
LBP21	LBP8
LBP22	LBP9
*OBJF	LBP10
*OBJQ1	LBP11
OBJQ2	LBP12
*VLC1	LBP13
*VLC2	LBP14
/	LBP15
;	LBP16
	LBP17
MODEL MR00	VLC1
*MIP with convex envelopes	VLC2
/	/
OBJFUN	;
TC1	
TC2	MODEL MF00
TC3	*NLP to obtain upper bound to original
TC4	MINLP
TC5	/
TC6	TCUB1
TC7	TCUB2
TC8	TCUB3
TC9	TCUB4
TC10	TCUB5
TC11	TCUB6
TC12	TC7
TC13	TCUB8
TC14	TC9
TC15	TC10
TC16	TCUB11
TC17	TC12
TC18	TC13
TC19	TCUB14
TC20	TCUB15
TC21	TCUB16
TC22	TCUB17
TC23	TCUB18
TC24	TC19

```

TC20
TC21
TC22
TC23
TC24
TC25
TC26
TC27
TC28
TC29
TC30
TC31
DCUB1
DCUB2
DC3
DCUB4
DCUB5
CCUB1
CCUB2
CC3
CC4
VB1
VB2
VB3
VB4
VB5
VB6
VB7
VB8
VB9
VB10
VB11
VB12
VB13
VB14
VB15
OBJFIXBIN
/
;

ZZUB.FX = INF;

*to generate linear cuts for Q1
SOLVE CRUDE_LB1 USING MIP MINIMIZING
ZZQ1      ;

*to generate linear cuts for Q2
SOLVE CRUDE_LB2 USING MIP MINIMIZING
ZZQ2      ;

*-----
-----
*-----
Solution to model MIP

SOLVE MR00 minimizing ZZ using mip;

*===== TO FIX BINARY
VARIABLES TO PARAMETER VALUES
=====

PUT RESCRU;

PUT //
PUT / "-----"
-----"
PUT / "BINARY VARIABLES OBTAINED FROM
MIP-LB"
PUT / "-----"
-----"

PUT //

LOOP( (A,B)$A1(A,B),

```

```

LOOP( L,
  WBUB1(A,B,L) = WB1.L(A,B,L)
  PUT "WB1('
    "A.TL:3,"",",B.TL:3,"",",L.TL:3," ")=
    ",WBUB1(A,B,L):18:12;
    PUT "; "
  );
  PUT /
);

PUT /

LOOP( (B,H)$H1(B,H),
  LOOP( L,
    WBUB2(B,H,L) = WB2.L(B,H,L)
    PUT "WB2('
      "B.TL:3,"",",H.TL:3,"",",L.TL:3," ")=
      ",WBUB2(B,H,L):18:12;
      PUT "; "
    );
    PUT /
  );

*-----
Solution to NLP non convex model (UB)

SOLVE MF00 MINIMIZING ZZFIXBIN USING
NLP;

```

APPENDIX IX: GAMS OUTPUT

FILE ON CRUDE OIL

SCHEDULING WITH SLACK

VARIABLES MODEL

GAMS Rev 149 x86/MS Windows
 04/28/07 15:45:03 Page 5
 : SCHEDULING REFINERY CRUDE OIL
 OPERATIONS
 Solution Report SOLVE CRUDE_LB2
 Using MIP From line 2458

GAMS Rev 149 x86/MS Windows
 04/28/07 15:45:03 Page 3
 : SCHEDULING REFINERY CRUDE OIL
 OPERATIONS
 Solution Report SOLVE CRUDE_LB1
 Using MIP From line 2455

S O L V E S U M M A R Y
 MODEL CRUDE_LB2
 OBJECTIVE ZZQ2
 TYPE MIP
 DIRECTION MINIMIZE
 SOLVER CPLEX FROM
 LINE 2458

S O L V E S U M M A R Y
 MODEL CRUDE_LB1
 OBJECTIVE ZZQ1
 TYPE MIP
 DIRECTION MINIMIZE
 SOLVER CPLEX FROM
 LINE 2455

**** SOLVER STATUS 1 NORMAL
 COMPLETION
 **** MODEL STATUS 1 OPTIMAL
 **** OBJECTIVE VALUE
 8.0114

**** SOLVER STATUS 1 NORMAL
 COMPLETION
 **** MODEL STATUS 1 OPTIMAL
 **** OBJECTIVE VALUE
 538.6000
 RESOURCE USAGE, LIMIT 0.125
 1000.000
 ITERATION COUNT, LIMIT 207
 10000

RESOURCE USAGE, LIMIT 0.125
 1000.000
 ITERATION COUNT, LIMIT 173
 10000

ILOG CPLEX Dec 24, 2007 WIN.CP.CP
 22.6 035.037.041.vis For Cplex 11.0
 Cplex 11.0.0, GAMS Link 34
 Cplex licensed for 1 use of lp, qp,
 mip and barrier, with 2 parallel
 threads.

ILOG CPLEX Dec 24, 2007 WIN.CP.CP
 22.6 035.037.041.vis For Cplex 11.0
 Cplex 11.0.0, GAMS Link 34
 Cplex licensed for 1 use of lp, qp,
 mip and barrier, with 2 parallel
 threads.

Proven optimal solution.

Proven optimal solution.

MIP Solution: 538.600000
 (34 iterations, 0 nodes)
 Final Solve: 538.600000
 (173 iterations)

MIP Solution: 8.011429
 (31 iterations, 0 nodes)
 Final Solve: 8.011429
 (142 iterations)

Best possible: 538.600000
 Absolute gap: 0.000000
 Relative gap: 0.000000

Best possible: 8.011429
 Absolute gap: 0.000000
 Relative gap: 0.000000

GAMS Rev 149 x86/MS Windows
 04/28/07 15:45:03 Page 7
 : SCHEDULING REFINERY CRUDE OIL
 OPERATIONS
 Solution Report SOLVE MR00 Using
 MIP From line 2464

GAMS Rev 149 x86/MS Windows
 04/28/07 15:45:03 Page 9
 : SCHEDULING REFINERY CRUDE OIL
 OPERATIONS
 Solution Report SOLVE MF00 Using
 NLP From line 2500

S O L V E S U M M A R Y

MODEL MR00
 OBJECTIVE ZZ
 TYPE MIP
 DIRECTION MINIMIZE
 SOLVER CPLEX FROM
 LINE 2464

**** SOLVER STATUS 1 NORMAL
 COMPLETION
 **** MODEL STATUS 1 OPTIMAL
 **** OBJECTIVE VALUE
 1980.4400

RESOURCE USAGE, LIMIT 0.156
 1000.000
 ITERATION COUNT, LIMIT 214
 10000

ILOG CPLEX Dec 24, 2007 WIN.CP.CP
 22.6 035.037.041.vis For Cplex 11.0
 Cplex 11.0.0, GAMS Link 34
 Cplex licensed for 1 use of lp, qp,
 mip and barrier, with 2 parallel
 threads.

Proven optimal solution.

MIP Solution: 1980.440000
 (34 iterations, 0 nodes)
 Final Solve: 1980.440000
 (180 iterations)

Best possible: 1980.440000
 Absolute gap: 0.000000
 Relative gap: 0.000000

S O L V E S U M M A R Y

MODEL MF00
 OBJECTIVE ZZFIXBIN
 TYPE NLP
 DIRECTION MINIMIZE
 SOLVER BARON FROM
 LINE 2500

**** SOLVER STATUS 1 NORMAL
 COMPLETION
 **** MODEL STATUS 1 OPTIMAL
 **** OBJECTIVE VALUE
 2148.1050

RESOURCE USAGE, LIMIT 1.840
 1000.000
 ITERATION COUNT, LIMIT 0
 10000
 EVALUATION ERRORS 0

GAMS/BARON Dec 24, 2007 WIN.BA.NA
 22.6 011.000.000.vis P3PC

Branch And Reduce Optimization
 Navigator
 Nikolaos Sahinidis and Mohit
 Tawarmalani
 The Optimization Firm, LLC.

Total time elapsed : 000:00:02,
 in seconds: 2.09
 on parsing : 000:00:01,
 in seconds: 1.17
 on preprocessing: 000:00:01,
 in seconds: 0.92
 on navigating : 000:00:00,
 in seconds: 0.00
 on relaxed : 000:00:00,
 in seconds: 0.00
 on local : 000:00:00,
 in seconds: 0.00
 on tightening : 000:00:00,
 in seconds: 0.00
 on marginals : 000:00:00,
 in seconds: 0.00
 on probing : 000:00:00,
 in seconds: 0.00

Total no. of BaR iterations: -
 1
 Best solution found at node: -
 1
 Max. no. of nodes in memory:
 0

Solution = 2148.10499999 best
 solution found during preprocessing
 Best possible = 2148.105
 Absolute gap = 9.99989424599335E-9
 optca = 1E-9
 Relative gap = 0 optcr = 0