OPTIMIZATION OF A MULTI-LEVEL STEAM DISTRIBUTION SYSTEM BY MIXED INTEGER NON-LINEAR PROGRAMMING

By

Roland Saunion

Submitted in fulfilment of the requirements for the degree of Master of Science in Engineering in the school of Chemical Engineering, University of Natal, Durban.

Durban
October 2001

Abstract

The objective of this project is to optimize the SAPREF oil refinery steam distribution in which imbalances between the various levels presently require the venting of steam from the lowest level. The overall steam balance shows that the problem originates from an excess of high-pressure (HP) steam production for too few medium pressure steam users and turbines.

We proposed to solve this problem by considering the replacement of selected steam turbines with electrical drives. Given a set of demands of electricity, mechanical power and steam at various pressure levels, the objective is to recommend configuration changes to minimize overall cost. This is not a trivial problem, as steam not passed down through turbines to lower levels can create a shortage there, so a combination of replacements is required.

The variables of the problem are both decision variables on every steam turbine and continuous variables, such as flows and enthalpies. These decision variables are integer variables, 0 or 1 for every steam turbine. Depending on whether it is kept on steam use or replaced with an electrical drive, these variables are as follows:

$\varepsilon = 0$: keep the existing steam turbine

$\varepsilon = 1$: switch it to an electrical drive.

A complete and realistic model of this utility section must be constructed in order to represent the actual distribution accurately. This model will include an objective function to minimize, some equality and inequality constraints, and some cost functions. If we want this model to be accurate, we shall have to deal with nonlinearities to avoid simplifications, and these non-linearities could lead to infeasabilities or sub-optimal solutions. So we are facing a typical MINLP (Mixed Integer Non-Linear Programming) problem to find optimal configuration changes which will maximize the return on investment, meeting the electrical, mechanical and steam demands of the refinery. In order to solve this difficult optimization problem we shall use the user-friendly package GAMS (General Algebraic Modeling System).

Preface

SAPREF initiated the project to identify the root cause for venting approximately 400 tons/day of low-pressure steam. This venting is considered a waste of both energy and condensate quality water as well as an environmental problem. Firstly this project was given to two chemical engineering students, Judisha Chetty (University of Natal Durban) and Thoba Majola (University of the Witwatersrand), as part of their vacation work in December 1999 and January 2000. They performed refinery wide steam balances and identified where the venting problem lay. However, it appeared that removing steam turbines would change the overall refinery steam and energy balance and thus more investigation was required to optimize the steam distribution. Professor Michael Mulholland suggested tackling this challenging issue with a French student graduated form the ENSIGC (National School of Chemical Engineering) in Toulouse, France. The research study was funded by SAPREF and part of the living expenses of the student jointly by the NRF (National Research Foundation, South Africa) and the CNRS (National Center of Scientific Research, France).

Most of the work has been done in the postgraduate offices of the School of Chemical Engineering at the University of Natal, Durban. Investigations were also conducted at the SAPREF refinery under the supervision of Neeshlin Govender and with the co-operation of many other refinery staff and engineers.

The following courses were completed with the corresponding credits and results achieved:

DNC4DC1	Process Dynamics and Control	(16.0)	81%
DNC5RT1	Real Time Process Data Analysis	(16.0)	75%

I hereby declare that this dissertation is my own work, expect where otherwise stated, and that is has not been submitted for a degree to any other university or institution.

....
R. Saunion

2001-11-09
Date

As the candidate's supervisor I have / have not approved this dissertation for submission

Signed : M. Mulholland Name : M. MULHOLLAND Date : 2001.11.09

Acknowledgements

I would like to thank the following people for their contribution towards this research study:

My supervisor in the School of Chemical Engineering, **Professor Michael Mulholland**, for his all round guidance and assistance. He suggested the project and my coming to South Africa. He helped me in many ways and made my integration and stay here very enjoyable and enriching.

The **National Research Foundation of South Africa** and the **Centre National de Recherche Scientifique of France** for their financial assistance and co-operation for this project.

The **University of Natal** and the **Ecole Nationale Superieure d'Ingenieurs de Genie Chimique** (France) for having established an agreement allowing my coming here as a postgraduate student.

Professeur M.V Le Lann in France for having suggested my study in Durban and made this exchange and financial assistance possible.

The **SAPREF refinery** for having funded and conceiving this project and for entrusting it with the University of Natal.

Neeshlin Govender for his supervision of the project and his knowledge of the steam distribution at SAPREF.

All the **people working at SAPREF** for their availability and precious help during our investigations there.

My **colleagues in the School of Chemical Engineering**, University of Natal, for their understanding and for having integrated me into a fun work atmosphere.

My **family** for its great support and my girlfriend **Charlotte** for her true love and for having joined me here and made our time in South Africa unforgettable.

Table of contents

List of Figures

Chapter 3:

Chapter 4:

Chapter 5:

Chapter 6:

Chapter 7:

List of Tables

Chapter 5:

Chapter 6:

Chapter 7:

Chapter 8:

List of Symbols

Lowercase letters

a	Electricity cost (R/kWh)
f(x,y)	Objective function depending on x and y
$g_j(x,y)$	Constraint function depending on x and y
h_0	Enthalpy of water at 100°C and 1 bar: 400 kJ/kg
h^0_i	Enthalpy of outlet steam of turbine i (kJ/kg)
h_w	Enthalpy of water delivered to the desuperheaters (kJ/kg)
k	Iteration step
w_i	Flow of water delivered to the desuperheater i (tons/day)
x	Continuous variables
y	Integer variables
y^k	Fixed integer variables

Uppercase letters

$Celec_i$	Annual cost of running electrical drive i of size We_i kW (Rands/year)
CRp_i	Capital cost for replacement of steam turbine *i*, producing W_i kW (Rands)
Csteam	Cost of HP steam production (Rands/ton)
D	Project lifetime (years)
Fe_j	Steam imbalance on header j (tons/day)
FH	Flow of HP steam produced (tons/day)
Fi	Flow of steam used by unit i (tons/day)
FL	Flow of LP steam produced by local LP feeders (tons/day)
FM	Flow of MP steam produced by local MP feeders (tons/day)
Ga	Equation as seen by the algorithm
Gs	Equation scale factor
Gu	Equation as seen by the user
H_{FH}	Enthalpy of HP steam produced (kJ/kg)
H_{FL}	Enthalpy of steam from LP local feeders (kJ/kg)
H_{FM}	Enthalpy of steam from MP local feeders (kJ/kg)
H_j	Enthalpy of steam on header j (kJ/kg)
I	Investment (Rands)
LDA	Flow of steam let down from the AMP header to the MP header (tons/day)
LDAL	Flow of steam let down from the AMP header to the LP header (tons/day)
LDH_i	Flow of steam through HP turbine bypasses (tons/day)
LDHM	Flow of steam let down from the HP header to the MP header (tons/day)
LDM	Flow of steam let down from the MP header to the LP header (tons/day)
$Melec_i$	Annual cost of maintaining electrical drive i of size We_i kW (Rands/year)
Mst_i	Annual cost of maintaining turbine i, producing W_i kW (Rands/year)
NLP1	NLP subproblem to P1 for fixed yk
NLPinf	Feasibility NLP subproblem to P1
NS	Net Savings per annum (Rands/year)
P	Payback period (years)
P1	Most basic algebraic form of an MINLP problem
ROI	Return on Investment (% per annum)
Uj	Flow of steam used by users on the header j (tons/day)
Va	Variable as seen by the algorithm
Vent	Flow of LP steam vented to atmosphere (tons/day)
Vs	Variable scale factor
Vs_i	Salvage value (or net realisable value) for turbine *i* of size W_i kW (Rands/year)
Vu	Variable as seen by the user
$Werr_j$	Power requirement for error term on header j (kW)
W_n	Power produced by turbine n (kW)
Wu_j	Power demand by steam users on header j (kW)

X^{w}_{i}	Ratio of water to steam supplied to desuperheater i
X^{w}_{LDHM}	Ratio of water to steam supplied to LDHM desuperheater
Z	Objective function
Z^{*}_{U}	Best upper bound
Z^{k}_{L}	Lower bound to P1
Z^{k}_{U}	Upper bound to P1
Z_{m}	Cost function of operation m (Rands/year)

Greek letters

η_{elec}	Electrical efficiency of electrical drives
ΔH_{n}	Enthalpy drop across turbine n (kJ/kg)
$\in_{i}$	Decision variables on steam turbine i

Acronyms

AMP	Assured-Medium-Pressure
AP/OA/ER	Augmented Penalty/ Outer-Approximation/ Equality-Relaxation
BB	Branch and Bound
BP	British Petroleum
CONOPT	NLP solver
DICOPT	Discrete Continuous OPTimizer
ECP	Extended Cutting Plane
GAMS	General Algebraic Modeling System
GBD	Generalized Benders' Decomposition
HP	High-Pressure
IP	Integer Programming
LP	Low-Pressure
LP	Linear Programming
M/D	Modeling/ Decomposition
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MINOS	NLP solver
MIP	Mixed Integer Programming
MP	Medium-Pressure
NLP	Non-Linear Programming
OA	Outer-Approximation
OA/ER	Equality-Relaxation/ Outer-Approximation
OSL	IBM Optimization Subroutine Library
Pross	SAPREF plant database access system
QP	Quadratic Programming
RMINLP	Relaxed Mixed Integer Non-Linear Programming
SAPREF	South African Petroleum REFinery
SRF	Standard Refinery Fuel
SRS	Standard Refinery Steam
VSD	Variable Speed Drive

Chapter 1

Introduction

1.1 SAPREF oil refinery

The SAPREF refinery in Durban, South Africa, is a complex refinery, the largest in South Africa, having a nameplate distillation capacity of 8250 ktonnes per annum (165 tbpd). The refinery is jointly owned by Shell SA and BP SA. The SAPREF site has a fuels refinery as well as a base oil refinery, the Samco lubricating oil refinery, which is also jointly owned. The base oil refinery has a capacity of 155 thousand tonnes per year. This is possibly the largest base oils refinery in Africa and it contributes substantially to the production capacity of the South African lubricants industry. Sapref is able to augment its fuels and lubricants and asphalt production, it also contributes some propylene feedstock for the chemicals sector as well as producing aliphatic hyrocarbon solvents, industrial processing oils and sulphur.
The SAPREF refinery originally opened in 1963 with an integrated unit and associated storage facilities. In the years following, a bitumen high vacuum unit, blowing unit and blending facilities, first crude distillation unit and lube oil plant were added. Other expansions in the 1960's and 1970's include the installation of a catalytic cracker, alkylation unit, second distillation unit, visbreaker and hydrocarbon solvents plant.

Work on upgrading the refinery commenced in 1991 at a total cost of $150 million. Capacity was increased by more than 30% and provided facilities for producing unleaded gasoline and low-sulphur diesel as well as decreasing energy consumption and environmental emissions. Named the Fox Project (Future Options for Expansion), it included the commissioning of a third crude distillation unit, a fourth hydrodesulphuriser with dedicated fractionation facilities, and a new high-vacuum distillation unit for the production of lube oil distillates. The SAPREF expansion used the latest refining technology from Shell with engineering design by Process Industry Engineering (PIE), a joint venture between Badger NV and EMS of South Africa. The installation contractor was Fluor Daniel. Currently Sapref is developing a master plan for future expansions into the next century.
In October 1996, Sapref announced a two year project to upgrade the refinery and centralise process control systems. The project was estimated to cost R 200 million ($US 48 million). Shareholders are still considering whether to take advantage of the upgrade project to also build new expansion capacity to take advantage of the demand growth in the local markets.

1.2 Steam distribution optimization

As part of these improvements, the steam distribution had to be considered for optimization.

Indeed SAPREF is producing its own steam, a source of power and heating and which may be involved in process reactions too.

Steam applications include producing mechanical power and serving as a stripping, fluidizing, agitating, atomizing, ejector-motive and direct-heating stream. Its quantities, pressure levels and degrees of superheat are set by such process needs and define the utility section of the refinery.

Utility plants supply the required energy demands to industrial process plants, namely, electrical, mechanical and steam demands. Electrical demands arise from external and internal electrical utility plant devices. Mechanical demands come from the power required to drive process units such as compressors, pumps, blowers, etc, and from the power to drive utility pumps and air fans. Steam demands arise from the heat that is required from the heat exchange network and from the reaction system.

The equipment that can be typically used in a utility plant includes different types of boilers and steam turbines, electric motors, electric generators driven by steam turbines, headers at different pressures to collect and distribute steam and condensate, and other auxiliary units such as deaerators, pressure-letdown stations, condensers, and utility pumps. A number of feasible arrangements of these units can provide the specified utility demands.

The SAPREF steam distribution network is a very complex one, as can be seen in Appendix A. It has four different headers, each one at a constant pressure going from the high-pressure header (50 bars) to the low-pressure one (approximately 3-5 bars). It has local steam feeders, steam users and different types of steam turbines (back-pressure, condensing, etc) for the various headers.

It was a well-established fact that the SAPREF refinery was "long" in low-pressure (LP) steam, meaning an excessive low-pressure steam production leading to venting of this excess of steam to atmosphere. Venting is forced because the LP steam pressure (3-5 bars) is too low to allow for excess LP steam to be exported into the refinery steam grid. This vent of unused LP steam has to be avoided for both economical and environmental concerns.

1.3 Objectives

The objective of this project is so to optimize the SAPREF steam distribution to avoid wasting money by producing unused steam, which has to be vented. The problem of excessive LP steam is not only a local problem on the LP steam header. The whole distribution is involved as the LP steam is mainly produced by medium-pressure (MP) and assured-medium-pressure (AMP) steam back-pressure turbines. The remaining LP steam produced is via the MP and AMP letdown-pressure stations.

The overall steam balance shows that the actual problem comes from an excess of high-pressure (HP) steam production for too few MP steam users and turbines. So the problem has to be tackled from the root, the HP steam header.

It is proposed to solve this problem by considering the replacement of steam turbines with electrical drives. So given a set of demands of electricity, mechanical power and steam at various pressure levels, the objective is to design a utility plant at minimum cost by determining the equipment configuration and its corresponding operating conditions. The variables of the problem are both decision variables on every steam turbine and continuous variables such as flows and enthalpies. These decision variables are integer variables, 0 or 1 for each steam turbine.

Depending on whether it is kept on steam use or replaced with an electrical drive, these variables are as follows: $\varepsilon = 0$: keep existing steam turbine

$\varepsilon = 1$: switch it to electrical drive.

A complete and realistic model of the utility section must be done in order to represent the actual distribution accurately. This model will include an objective function to minimize, some equality and inequality constraints, and some cost functions. If we want this model to be accurate enough, we will have to deal with nonlinearities to avoid simplifications, which could lead to infeasabilities or sub-optimal solutions.

So we are facing a typical MINLP (Mixed Integer Non-Linear Programming) problem to find the optimal configuration at minimum cost meeting the electrical, mechanical and steam demands of the refinery. In order to solve this kind of very difficult optimization problem we will use the user-friendly optimization package GAMS (General Algebraic Modeling System).

1.4 Thesis layout

Chapter 1 introduces the reader to the SAPREF refinery, its steam distribution network and what the objectives of this distribution optimization are. Chapter 2 presents an overall look at the history of optimization techniques over the last decades from a literature review on articles and books dealing with the topic, focusing on programming methods. The chapter following this overview goes into details on the complex theory of the type of problem we are facing, MINLP problems, and presents the most common methods to tackle these problems. A complete description of the SAPREF utility section to optimize its components is then done in Chapter 4. This utility section is then rigorously modeled in Chapter 5. Both the physical and financial sides are considered as well as the objective function to minimize. All this forms the model to program in the GAMS language, which is presented and applied to formulate our optimization program in Chapter 6. Then in Chapter 7 a complete view of results achieved as well as a sensitivity analysis are discussed. Eventually some conclusions and recommendations are drawn in the last, eighth, chapter.

Chapter 2

A review of optimization techniques

In this chapter we will see the previous and latest work that has been done, especially in the last 20 years, on optimization techniques and their implementation in actual process synthesis problems. Even if the availability of specific articles on MINLP problems appears to be limited, we can have a good overall understanding of optimization techniques by a small team of well-recognized international researchers dedicated to this field, who have written most of the papers on the topic.

2.1 History of optimization

Reviews on earlier developments in the area of process synthesis can be found in Hendry et al.(1973), Hlavacek (1978), and Nishida et al. (1981). In the late sixties, work began to develop a systematic approach to process synthesis based mostly on the use of decomposition and heuristic rules (Rudd and Watson, 1968; Rudd, 1968; Masso and Rudd, 1969). Algorithmic methods for selecting the optimal configuration from a given superstructure (problem formulation) also began to be developed through the use of direct search methods for continuous variables (Umeda et al., 1972; Ichikawa and Fan, 1973) as well as branch and bound search methods (Lee et al., 1970).

In terms of process flowsheets, the first computer-aided process synthesizer for generating initial structures, AIDES (Adaptive Initial DEsign Synthesizer), was developed by Rudd and his students (Siirola et al., 1971; Siirola and Rudd, 1971; Powers, 1972); using a high level representation of tasks, it relied on the use of heuristics and linear programming, which were coordinated through a means-ends-analysis search. The second computer-aided process synthesizer to be developed was BALTAZAR (Mahalec and Motard, 1977a,b). It also relied on heuristics and linear programming, and used a tree search within the framework of theorem proving. Neither AIDES nor BALTAZAR incorporated equipment costs directly, but they employed heuristics as indicators of economic performance.

Grossmann (1996) defines the current state of flowsheet synthesis as represented by two different approaches:

(1) hierarchical decomposition (Douglas, 1985, 1988, 1990) and its computer implementation PIP (Process Invention Procedure), (Kirkwood et al., 1988) and
(2) mathematical programming (Grossmann, 1985, 1990a,b) and its implementation in PROSYN (PROcess SYNthesizer), (Kravanja and Grossmann, 1990). It has been pointed out (Rippin, 1990) that these two approaches are concerned with different aspects of design and can be regarded as complementary.

However some others methods, based on a "thermodynamic" approach can be found in the literature.

2.2 Optimization techniques

2.2.1 The hierarchical decomposition techniques

The hierarchical design strategy is used for creating and screening processing plant alternatives based on heuristics (Kovac and Glavic, 1995). Grossmann (1996) describes this technique as follows:
It breaks the synthesis procedure into five discrete decision levels: (1) batch versus continuous, (2) input/output structure flowsheet, (3) recycle structure and reactor considerations, (4) separation systems, and (5) heat exchanger network. At each decision level beyond the first, the economic potential of the project is evaluated and a decision is made whether or not further work on the project is justified. This method utilizes heuristics, short-cut design procedures, and physical insight to develop an initial base-case design. This approach is motivated by Douglas' claim (1985) that only 1% of all designs are implemented in practice. This same Douglas (1985) proposed a hierarchical procedure where heuristics rules were used to guide the search direction, overcome the need to examine all possible structures and find a small near-optimal arrangement. However, based on heuristics, this approach cannot rigorously produce an optimal design and high accuracy is not expected (Nelson and Douglas, 1990).

2.2.2 The thermodynamic methods

The pinch analysis is one of them: heat integration of process streams is used to minimize utility consumption (Linnhoff et al., 1982). This technique is limited to process streams with fixed values of flow rates and temperatures, and it often leads to nonoptimal flow rates and temperatures in the superstructure (Lang et al., 1988).

Chou and Shih (1987) proposed a systematic, thermodynamically oriented method for design and synthesis of plant utility systems which relies on the idea that heat requirements are satisfied in preference to power requirements. This method gives a good overall thermal efficiency. The procedure simplifies the calculations and provides a better understanding of the problem's characteristics.

Marechal and Kalitventzeff (1997) use a method based on the Carnot factor to identify and to estimate the mechanical power that can be recovered using the exergy available in the process. This method called the Integrated Combined Heat and Power (ICHP) approach allows estimation of the combined power production potential of the system and identification of the optimal pressure levels of the steam network. This method is based on the representation of steam as constant temperature utility streams. Marechal and Kalitventzeff (1999) extended this method to present an optimisation model to target the optimal flowrates in a steam network and in the related utilities, using a superstructure obtained from the definition of the pressure and the enthalpy content of the headers and a linear simulation model of this superstructure. The interest of this extended method it its ability to represent the integration of complex steam networks starting only from the definition of headers and, of course, the hot and cold streams of the site processes.

Mavromatis and Kokossis (1998) presented in their paper another thermodynamic approach to select the steam levels and the configuration of the operating units between these steam levels in a utility network. They use the Total Site Analysis (TSA), developed by Dhole and Linnhoff (1993) and Raissi (1994), which is an extension of Pinch Analysis (Linnhoff et al, 1982 and Linnhoff, 1993) to address the design of integrated processes. By this approach it is then possible to determine the optimum placement of the steam levels and the optimum loads at each steam level that will maximise the heat recovery on the site. Then in order to select the turbine configuration, a new shaftwork-targeting tool is proposed, termed the Turbine Hardware Model (THM), based on the principle of the Willians line which provides a linear relation between the steam flowrate and the power output.

The same authors address in a second article the development and optimization of the utility network designed in their previous paper. They use a three-stage procedure to reduce the size and complexity of the problem. Based on engineering knowledge and analytical insight this method anticipates operational variations and eventually the network most suitable for a particular case is then determined among a number of alternative turbine networks, on the basis of practical considerations and controllability aspects.

The main drawback of these methods is that even if the design with highest thermal efficiency is obtained, it may not be economically attractive because capital costs may be too high and so it may not lead to the optimum solution in terms of costs involved.

2.2.3 The mathematical programming methods

2.2.3.1 General programming methods

As indicated by Grossmann (1985), the previous two approaches have been used quite extensively with some important successes despite their obvious limitations, such as the fact of not being able to assert the quality of the solution, the assumption on dominance of energy costs, and the restricted application to specific subproblems.

Algorithmic methods, on the other hand, offer a more general and systematic approach since they explicitly account for the economic trade-offs and interactions in the synthesis of arbitrary processing systems. Furthermore, because of their nature these methods can accommodate the other two approaches and are better suited for automatic synthesis of systems.

Nevertheless, Chou and Shih (1987) pointed out that although a suitable formulation of the mathematical representation of a system can easily solve the problem, the usefulness of these approaches is detracted by the heavy dependence on mathematical calculations. In addition, an inappropriate objective function may hinder the desired real solution, and the final unique numerical optimal solution of LP and MILP methods is generally not convincing enough in engineering practice.

However, we will see in this section that the recent improvements in both mathematical techniques and computers capabilities make the algorithmic method the more appropriate to optimization problems.

One of the first methods based on algorithmic approach by Gaines and Gaddy (1976) developed a modular process simulator PROPS (Process Optimization System) capable of optimization of complex systems by using an adaptive random search algorithm.

Then the first papers using mathematical optimization approaches were based on Linear Programming (LP) such as the ones in Nishio and Johnson (1979) and Petroulas and Reklaitis (1984) who used a dynamic programming method to optimise the steam header conditions as continuous variables and a linear programming method for the optimum allocation of drivers with the common objective of minimizing the real work loss.

A non-linear programming strategy was applied by Colmenares and Sieder (1989) for the design of utility systems integrated with the chemical process. It was based on the temperature interval method and the development of a superstructure of Rankine cycles.

Dolan, Cummings and Le Van (1989) applied the simulated annealing multivariable optimization technique to network design. This technique is based on the Monte Carlo method used in statistical studies of condensed systems and follows by drawing an analogy between energy minimization in physical systems and costs minimization in design applications.

Papoulias and Grossmann (1983) introduced the Mixed Integer Linear Programming (MILP) approach. This approach consists in formulating an MILP model to select among all the alternative units included in a proposed utility plant superstructure by minimizing linear capital costs summed with fixed charges and operating costs.

The MILP formulation is derived from the original MINLP (Mixed Integer Non Linear Programming) formulation by fixing operating conditions such as pressures and temperatures, which render the energy balances to be linear.

2.2.3.2 Mixed Integer Non Linear Programming methods

In this section, we will have a more specific review on articles dealing with MINLP problems. Note that all the algorithms mentioned here will be discussed in detail in the next section.

More specifically, the mathematical programming approach utilizes optimization techniques to select the configuration and parameters of the processing system (Grossmann, 1985, 1990a,b). A superstructure containing alternative processing units and interconnections is modeled as discrete, binary variables (0-1) to depict the existence (1) or nonexistence (0) of that unit. To explicitly handle the nonlinearities in process models, some algorithms for MINLP have been developed and implemented in a computer-aided process synthesizer.

Generalized Benders' Decomposition (GBD):
Geoffrion (1972) generalized the idea by Benders to exploit the structure of mathematical programming problems with complicating variables (variables which, when temporarily fixed, render the remaining optimization problem considerably more tractable). Indeed, fixing the values of the complicating variables reduces the given problem to an ordinary linear problem parameterized by the value of the complicating variables vector. Geoffrion's approach, called the Generalized Benders Decomposition (GBD), is the generalization of the Benders' approach to a broader class of programs in which the parameterized subproblem need no longer be a linear program.

Outer-Approximation (OA):
The Outer-Approximation (OA) algorithm developed by Duran and Grossmann (1986) is now widely used to solve the MINLP problem where binary variables appear in linear functions and continuous variables appear in both linear and nonlinear convex functions. Briefly, this algorithm partitions the problem into two parts: (1) an NLP subproblem, where the continuous variables for a single flowsheet configuration are initially optimized and then the remaining alternative substructures are suboptimized for the given flows; (2) linearization of the nonlinear equations to obtain an MILP master problem, which then determines a new optimal flowsheet configuration (i.e. new set of binary variables) for the next NLP subproblem. Duran and Grossmann applied this algorithm as well as a method for the simultaneous optimization and heat integration in process flowsheets to the optimization of several example problems.

Equality-Relaxation Outer Approximation (OA/ER):
Kocis and Grossmann (1987) presented an Equality-Relaxation variant to the Outer-Approximation algorithm (OA/ER) for solving MINLP problems. The proposed algorithm has the important capability of being able to explicitly handle nonlinear equations within MINLP formulations that have linear integer variables and linear/nonlinear continuous variables. Note that the OA algorithm could only handle linear equality and linear/nonlinear inequality constraints. The basic idea behind this algorithm is to relax the nonlinear equations to inequalities. The proposed algorithm has been implemented in the computer package DICOPT (Discrete Continuous OPTimizer) for the automated solution of MINLP problems.

DICOPT and GAMS:

The same authors in 1988 used the same implementation of the OA/ER algorithm, DICOPT, coupling with a powerful modelling language GAMS (General Algebraic Modelling System, Kendrick and Meeraus, 1985) to solve large scale MINLP problems. They showed that the improved efficiency of NLP (MINOS, Murtagh and Saunders, 1985) and MILP solvers (MPSX, IBM, 1979) and the increased computing power render DICOPT a very efficient tool for solving this type of problems.

Modeling/ decomposition (M/D):

Kocis and Grossmann again (1989) pointed out that modeling can have a great impact in the quality of solutions that are obtained. They presented a modeling/decomposition (M/D) scheme that exploits special features in structural flowsheet optimization problems to enhance the performance of the OA/ER algorithm. Some features of the proposed procedure are that it avoids zero flows or replaces as many nonlinearities as possible (for instance the model should be as linear as possible and it is preferable to place the nonlinearities in the objective function rather than in the constraints). It reduces the computational effort required to solve large-scale problems and increases the likelihood of converging to the global optimum.

Prosyn:

Kravanja and Grossmann (1990) implemented this M/D strategy in the computer package PROSYN and proposed systematic procedures for deriving the decomposition for arbitrary superstructures, and for handling heat exchangers network costs in simultaneous optimization and heat integration. The associated MINLP problems can be initialized more easily, the decomposition of the superstructure reduces the computational effort for solving the NLP subproblems, and the effects of nonconvexities involved can be reduced.

Approaches for nonconvex problems:

Viswanathan and Grossmann (1990) presented an improved Outer-Approximation algorithm for MINLP problems where convexity conditions may not hold. The proposed algorithm has as main features that it starts with the solution of the NLP relaxation problem, and that it features an MILP master problem with an augmented penalty function that allows violations of linearizations of the nonlinear functions. This scheme provides a direct way of handling nonconvexities, which are often present in engineering design and optimization problems. The algorithm, called Augmented Penalty/Outer-Approximation/Equality-Relaxation (AP/OA/ER), has been implemented in DICOPT++ as part of the modelling system GAMS. Once again, although no theoretical guarantee can be given, the proposed method has a high degree of reliability for finding the global optimum in nonconvex problems.

A new global optimization approach is proposed by Floudas, Aggarwal and Ciric (1989) for the solution of nonconvex NLP and MINLP problems. First, the sources of nonconvexities are identified. Then, the variables and constraints are decomposed into two sets (using transformations if needed). This leads to the decomposition of the original nonconvex problem, that has no special structure, into two subproblems, which can be solved for their respective global solutions. Based on the Generalized Benders' decomposition method, this approach was able to determine the global solution of a number of example problems even though the identification of the global optimum cannot be mathematically guaranteed.

Salcedo (1992) proposed another approach to solve nonconvex NLP and MINLP problems, using an adaptive random search. His optimization algorithm (the MSGA algorithm) is an extension of an adaptive random search NLP solver (the SGA algorithm). The main characteristics of the NLP solver are that it employs a variable, parameter-dependent compression vector for the contraction of the search regions and that it incorporates shifting strategies allowing for wrong-way moves to be produced. These features coupled with the random-search concept are responsible for the good robustness in escaping local optima. The MINLP solver does not require any problem decomposition, nor identification or elimination of nonconvexities, prior to its application. Even though this algorithm works efficiently for small and medium scale problems, it may be preferable to use more efficient methods such as the generalized Benders' decomposition or the OA/ER for larger problems.

Ryoo and Sahindis (1994) presented another algorithm for finding global solutions of nonconvex NLP and MINLP problems based on the solution of a sequence of convex underestimating subproblems generated by evolutionary subdivision of the search region. The key component of this new algorithm, called branch and reduce algorithm, is to reduce the ranges of variables based on optimality and feasibility tests. The former use known feasible solutions and perturbation results to exclude inferior parts of the search region from consideration, while the latter analyze constraints to obtain valid inequalities. Furthermore, the algorithm integrates these devices with an efficient local search heuristic. Computational results demonstrate that the algorithm is typically faster, requires less storage and produces more accurate results than several other current branch and bound based approaches.

Kravanja and Grossmann (1994) improved the PROSYN package to a new version, more user-friendly and able to handle nonconvexities. Indeed in the previous version the user was required to provide a complex model representation for the superstructure and complex logic relations of the M/D strategy. In the new version logic has been automated, library models for process units and interconnection nodes have been built in, and data for basic physical properties have been added. Furthermore, the latest version of the OA algorithm, the AP/OA/ER has been implemented in order to handle nonconvexities. All solutions of the problems prove that significant savings can be obtained if topology and parameter optimization of different systems are performed simultaneously and integrally as a total system.

Extended Cutting Plane (ECP):

Westerlund, Petterson and Grossmann (1994) introduced the Extended Cutting Plane (ECP) method combined with a general (integer) branch and bound method for solving optimal pump configurations as a MINLP problem. Different problem formulations are also given to improve the solution. By using the standard cutting plane method (Kelley, 1960), only the continuous optimal solution is solved. In order to obtain the MI solution, the linearized LP problem can, however, be solved as an MILP problem by a branch and bound method (Beale, 1977; Gupta, 1980) in each iteration. However since the problem is non-convex Kelleys cutting plane algorithm cannot be applied directly and an extended procedure where linearized constraints are both removed, replaced and added in each iteration is used instead. With this extension the algorithm showed good convergence properties also for non-convex problems. Concerning the formulation of the problem, it is mentioned that generally, bilinear expressions and nonlinear functions multiplied by binary variables should be avoided. Some examples using both DICOPT++ and the ECP method showed the significant improvements achieved when using this simple rule. No conclusions were drawn as whether it is better to use DICOPT++ or the ECP method.

Binary Separable Programming (BSP):

Westerlund and Petterson (1995) improved their own approach on this nonconvex problem of pump configurations by presenting a new method using binary separable programming. The proposed method separates the problem into a two level optimization problem. The lower level problems are convex MINLP problems and can be solved globally with existing MINLP codes such as OA or ECP. The upper level problem contains nonconvex functions and the minimization task has been formulated as a Binary Separable Programming (BSP) problem consisting only of binary variables. The authors suggested that this approach might also be applicable for other structural optimization problems but it has to be noted that their method is successful here because in this specific case the separable programming problem can be formulated as a zero-one programming problem and this might not be the case for any other problems. This BSP formulation can than be solved with a zero-one or an integer programming algorithm.

Quadratic Outer Approximation:

Fletcher and Leyffer (1994) generalized the OA algorithm by Duran and Grossmann (1986) to nonlinear functions involving integer variables, allowing a much wider class of problem to be tackled. Their problem formulation allows the integer variables to occur nonlinearly and a new and simpler proof of termination is given. The occurrence of infeasible solutions to NLP subproblems is treated in a rigorous way which is generally applicable to many different methods for solving this kind of problem. The practical performance of the resulting algorithm has proved to be similar to that of the Duran and Grossmann algorithm. However an example is provided which shows that the algorithm can be very inefficient. This behavior leads the authors to investigate a new quadratic outer approximation algorithm that takes second order information into account. Eventually, an alternative approach is considered to the difficulties caused by infeasibility in outer approximation, in which exact penalty functions are used to solve the NLP subproblems.

Comparison of three methods:

An interesting survey by Skrifvars, Harjunkoski, Westerlund, Kravanja and Porn (1996) compared different MINLP methods applied on certain chemical engineering problems. They used the three following methods for each problem: The Extended Cutting Plane (ECP) (Westerlund and Petterson, 1995), the Generalized Benders' Decomposition (GBD) (Geoffrion, 1972) and the Outer Approximation (OA) (Duran and Grossmann, 1986). It emerges from this survey that the ECP efficiency as compared to the other tested methods increases as the proportion of discrete variables increases in smaller problems. In large problems ECP needs less time but more iterations than OA so no clear, general conclusions can be drawn. More generally, the number of iterations does not indicate the actual effort and CPU time needs to be investigated as well.

Branch and bound algorithms:

SMIN & GMIN branch and bound:
Two new branch and bound algorithms for the global optimization of MINLP problems with twice-differentiable functions in the continuous variables are presented by Adjiman, Androulakis and Floudas (1997). Their theoretical foundations provide guarantees that the global optimum solution is reached provided that condition of twice differentiability, even with nonconvex functions, is respected. In the Special Structure Mixed Integer Nonlinear αBB (SMIN-αBB), valid lower bounds are obtained by constructing and solving a convex MINLP in which the binary variables participate in linear or mixed-bilinear terms. In the General Structure Mixed Integer Nonlinear αBB (GMIN-αBB), a continuous relaxation is solved to global optimality or rigorously underestimated and more complex binary or mixed terms can therefore be handled. Both approaches use recent developments for the valid underestimation of general functions with continuous second-order derivatives to build the convex lower bounding MINLPs as well as GBD or OA algorithms to solve the underestimating problem to global optimality, provided that the binary variables participate separably and linearly. The potential application of these global optimization algorithms for a broad class of problems has been shown on few problems and realistic examples.

Reformulation/Spatial branch and bound:
Smith and Pantelides (1997) tackled as well the problem of nonconvexity in MINLP problems by presenting a modified version of their own previous algorithm (1996), the reformulation/spatial branch and bound algorithm. This new algorithm is composed of four steps: Bounds tightening, objective function lower bound generation, objective function upper bound generation, and branching. The proposed method has been implemented within the gPROMS process-modelling environment (Barton and Pantelides, 1994). It has been possible to locate the global optimum of a number of test problems from the field of engineering design in reasonable computational time.

Applications to industrial problems:

Pahor and Kravanja (1995) describe a more general procedure for simultaneous solution and MINLP synthesis of process problems represented in an equation oriented modelling environment by differential-algebraic systems of equations (DAE) using an orthogonal collocation on finite elements (OCFE) to discretize differential equations, then embedded into MINLP problems. The approximated modelling usually raises the robustness of the optimization, but approximations in the model can also lead to an incorrect optimal topology. The proposed approach is important for solving MINLP problems when analytical solutions of differential equations cannot be found or they are too complicated or inadequate to be used. The modelling technique has been successfully applied to the synthesis of the PFR reactor network problem using the computer package PROSYN.

Diaz and Bandoni (1995) showed that the application of a MINLP algorithm integrated to a process simulator is a possible and promising strategy for the structural and continuous optimization of entire large-scale real chemical plants in operation. In their example, the application of MINLP techniques has given an increase of about US$300,000/yr in the gross profit over NLP techniques. They found the OA algorithm, as originally proposed, perfectly adequate to be linked with a rigorous process simulator, where the equations representing the mathematical model are implicitly solved inside the simulator. However, this procedure must be implemented with care since complex, nonlinear simulation models may lead to nonconvexities and thus suboptimal solutions. Even though global optimality cannot be assured, the "best solution found" for an industrial problem may represent an important improvement in the profit of the operation, making the effort of an optimization project worthwhile.

Bruno, Fernandez, Castells and Grossmann (1998) presented a rigorous MINLP model for the optimal synthesis and operation of utility plants that satisfy given electrical, mechanical and heating demands of industrial processes. The objective here is to develop a nonlinear model accurate enough for its implementation to actual industrial problems. In this model, nonlinear equations are extensively used for the cost of equipment and for the plant performance in terms of enthalpies, entropies and efficiencies. The proposed approach allows for the simultaneous optimization of the configuration, and selection of flowrates, enthalpies and steam turbine efficiencies. The model has been implemented in the computer package STEAM, that automatically generates the model and interfaces with GAMS. DICOPT++ has been used with MINOS as the NLP solver and OSL as the MILP for the master problem. Modest computational effort is required even with relatively large examples involving up to one hundred integer variables and several hundred continuous variables and constraints. The comparison with a simplified MILP model showed that variables such as temperatures and steam turbine efficiencies, which are treated as fixed parameters in the MILP model, can lead to infeasible or suboptimal solutions when compared to the solutions obtained from the MINLP model.

Note that the subject of this paper is similar to our problem except on certain assumptions in the modelisation like the choice of decision variables and the use of a computer package to generate the model, as we shall see in a following section.

2.2.3.3 Multiperiod optimization and disjunctive programming

We will briefly review in this section the different techniques developed in the recent years, that deal with these two aspects of the optimization field. We will especially be concerned about murliperiod optimization for the purpose of our project.

2.2.3.3.1 Multiperiod optimization problems

Multiperiod optimization problems for design and planning in the chemical industry are problems in which constraints are specified over several time periods or a set of scenarios (e.g., costs or demands vary from period to period due to market or seasonal changes).

From the middle of the eighties, there has been an increased interest in the development of systematic methods for the design of flexible chemical plants. The motivation for this comes from the fact that in practise flexibility is usually introduced by applying empirical overdesign, a practise that does not guarentee optimality or even feasibility over the desirable range of conditions. A major class of flexibility problems, the multiperiod design problem, involves designing plants which are capable of operating under various specified conditions in a sequence of different time periods. Another class deals with uncertainty involved in some design parameters, which is the problem of optimal design under uncertainty. As has been shown by Grossmann and Sargent (1978) and Halemane and Grossmann (1983), this problem requires the iterative solution of multiperiod designs which involve mixed integer nonlinear programming (MINLP), where the number of decision variables and constraints can become rather large. Typical examples of multiperiod plants are refineries that process different types of crudes and batch plants that produce a variety of different specialty chemicals. Other situations in which multiperiod operation schemes appear are plants that have seasonal demands. One characteristic of the multiperiod design problem is that it involves two distinct classes of variables, the design and the state variables. The design variables, representing equipment elements such as reactor and vessel volumes or exchanger areas, or equipment existence such as steam turbines or heat exchangers, are the same for all periods of operation. The state variables, representing operating conditions such as temperatures, flow rates, or concentrations, can be different for different periods of operation. The goal in multiperiod design is to.decide on the values of the design and state variables so that the plants will be feasible to operate at all specified conditions while being optimal over the specified time horizon. The biggest problem in solving a multiperiod model lies in the fact that as the number of periods increases there is a disproportionate increase in the computational complexity in terms of both solution time requirements and robustness. This behavior prohibits the solution of industrially relevant multiperiod problems using standard general-purpose optimization methods.

Outer-Approximation based decomposition method:

Varvarezos, Grossmann and Biegler (1992) developed an efficient optimization method for convex NLP and MINLP multiperiod design optimization problems. They proposed an Outer-Approximation based decomposition method for solving these problems. Decomposition strategies aim to avoid failures and computational expenses of the standard optimization techniques. The basic idea behind the NLP decomposition strategy is based on a projection restriction procedure by which the problem at the level of design variables is reduced into one in which variables are eliminated by using the active constraints at each time period. Motivated from the ideas of the cutting plane method for convex NLP's (Kelley, 1960) and the outer-approximation method for MINLP (Duran and Grossmann, 1986), the method also involves an alternative sequence of primal and master problems. The main difference lies in the formulation of the master problem, which is similar in nature to the OA algorithm for MINLP. Here, the information from the primal problem is used to approximate the feasible region by accumulating linear approximations of the constraints of the original problem at the optimal points, in each iteration of the primal problem. The computational performance on some examples shows that substantial savings in computation time, and increased robustness can be achieved with the proposed method.

Two-stage approach for large MILP problems:

Iyer and Grossmann (1996) used a different method for determining the optimal planning over the planning horizon with startup and shutdown costs for utility systems. A two-stage approach is proposed that requires the solution of MILP subproblems coupled with a shortest path algorithm, resulting in orders of magnitude reduction in computation time as compared to a direct MILP solution using branch and bound enumeration. Links between periods because of startups and shutdowns make the number of discrete variables increase with the number of periods and can make the solution time increase exponentially. To avoid this computational problem, the new method first removes the linking constraints and solves for each period independently, then takes into account the effects of linking constraints on the objective function in the second stage in which the global optimum solution of the problem is determined with the shortest path algorithm. The proposed algorithm is faster than OSL (full space MILP) in GAMS and its computational requirements are linear with respect to the number of periods and global solution of the MILP is guaranteed. Unfortunately, this method is not applicable with nonlinear programming problems.

Bilevel Decomposition for MILP:

Iyer and Grossmann (1998) again presented a bilevel decomposition algorithm for long-range planning of process networks. To reduce the computational cost in the multiperiod MILP model, they proposed a decomposition algorithm that solves a master problem in the reduced space of binary variables to determine a selection of processes and an upper bound to the net present value. A planning model is then solved for the selected processes to determine the expansion policy and a lower bound to the objective function. The proposed decomposition algorithm, which produces the same optimal solution as a full space MILP method, however requires less computational effort than a full space method as it involves the solution of smaller subproblems. Indeed for very small problems, the method is no faster than the full space branch and bound method, but when it comes to medium and large-sized problems the proposed method proved to be faster. For large problems, the method gives a 10% improvement in objective function values compared to the suboptimal solution obtained by the full space method. Once again this method can only be applied to linear programming problems.

Reduced Hessian Successive Quadratic Programming (rSQP):

A more recent approach by Bhatia and Biegler (1999) developed an efficient algorithm for solving multiperiod design problems (MPD) using interior point method within a reduced Hessian successive quadratic programming (rSQP) framework. The limiting factor in solving MPD problems is a disproportionate increase in computational resources and decrease in solution robustness, with an increase in the number of periods. However, efficient decomposition strategies exist that exploit the block bordered diagonal structure of these problems and provide a linear increase in computational resources with the growth in periods. This was proposed in the (MPD-SQP) algorithm for general nonlinear MPD problems by Varvarezos, Biegler and Grossmann, 1994. On the other hand, the MPD-SQP algorithm employs an active set strategy for solving the quadratic programming (QP) subproblem and this is combinatorial in the number of active constraints. Also, it needs to address a potentially different structure with each update of the active set. This consideration usually leads MPD-SQP to adopt an early termination for the QP problem, and this often requires additional SQP iterations. In order to solve the QP completely at each iteration, interior point methods have advantages as the number of updates is independent of the number of active constraints and we deal with a fixed structure throughout the solution procedure. Incorporating these concepts leads to the MPD-rISQP algorithm. The efficiency of this algorithm is proved on example problems with up to 16000 variables.

Papalexandri, Pistikopoulos, Kalitventzeff, Dumont, Urmann and Gorschluter (1996) presented a model and an optimization framework to determine a steam production network operation and energy management schedules for variable demand levels and for uncertain process parameters. First a mathematical model is developed to describe the operation of the network, based on network measurements and employing data reconciliation techniques. Three different cases are studied:
One without any switch cost, a second one with costs assigned to start-up and/or switch of operation of equipment and then a third under process uncertainties (e.g., pump efficiencies). Generalized Benders Decomposition (GBD) is employed for the solution of the MINLP multiperiod model, as applied using the modelling language GAMS (Brooke et al., 1988). These case studies showed how such an integrated tool can be exploited in effective energy management and production scheduling.

Papalexandri, Pistikopoulos and Kalitventzeff (1997) presented an application to industrial problems of the modelling and optimization aspects in energy management and plant operation with variable energy demands. Continuous and discrete operating decisions are explicitly considered and costed accordingly, addressing realistically the trade-off between flexible operation and the corresponding cost. Several cases studies with different assumptions made with regards to shutdown and startup costs and uncertainties are presented. The MINLP case is solved using the generalized Benders' decomposition (GBD) and nonlinearities and uncertainties are set on the turbine efficiencies. This paper presents a similar method as the one mentioned previously by the same authors but the method is applied to larger problems and especially one very close to ours which determines a steam turbine network configuration but with different assumptions in the model regarding nonlinearities due to turbine efficiencies. Once again the examples studied show the effectiveness of the method in energy management and production scheduling.

2.2.3.3.2 Disjunctive optimization problems

Disjunctive programming is an optimization problem solution technique for an objective function and constraints expressed in logic form disjunctions (OR operators), and propositional logic.

Symbolic integration within branch and bound method for MILP:
Raman and Grossmann (1993) considered logic relations between potential units in a superstructure through symbolic integration within the numerically based branch and bound scheme. The objective of this integration is to reduce the number of nodes that must be enumerated by using the logic to decide on the branching of variables, and to determine by symbolic inference whether additional variables can be fixed at each node. Two different strategies for performing the integration are proposed that use the disjunctive and conjunctive normal form representations of the logic, respectively DNF and CNF. The DNF form produces the most savings. A procedure to systematically generate the logic for process flowsheet superstructures is addressed. As for the comparison with the case when all logic constraints are included as inequalities, there is a trade-off between solving a tighter but larger MILP problem vs solving a problem with fewer constraints. This trade-off can in part be resolved with a hybrid scheme that has been proposed in which only violated inequalities at the LP relaxation stage are included in the symbolic branch and bound. This method has been implemented in OSL2 in GAMS.

Generalized Disjunctive Programming (GDP) for MILP:
Raman and Grossmann (1994) again presented a modelling framework and computational technique for discrete optimization problems that relies on logic representation in which mixed-integer logic is represented through disjunctions, and integer logic through propositions. Modelling may have a great impact on the efficiency for solving these problems. The greatest factor that commonly influences the solution efficiency is the gap between the continuous relaxation and the optimal integer solution. Logic representation offers an alternative framework. The proposed method generalized the solution by Raman and Grossmann (1993) in the previous article, to mixed-integer logic constraints because it was only done for logic relations involving Boolean variables. That method can thus handle disjunctions with inequalities but is restricted to the case of linear equations and inequalities and symbolic logic relations that can be expressed in propositional logic. Also, the method is implemented with an LP based branch and bound scheme in which the use of violated inequalities from logic relations is considered as discussed in Raman and Grossmann (1993). Further, the potential efficiency of this approach has been demonstrated on several problems in jobshop scheduling and process networks.

Convex hull formulation:

Turkay and Grossmann (1996) addressed the optimization of process models that involve a discontinuous investment cost function with fixed charges, and which are defined over several regions (the cost function depends on in which region the design variables lie). Indeed for real problems, one cannot always use a single correlation assuming a continuous linear or nonlinear function over the entire range of the design variable. These discontinuous cost functions are then naturally expressed by disjunctions. It was shown that the conventional approaches used to tackle this kind of problem such as the direct NLP approach, the smooth approximation and big-M model, can produce failure and not lead to an optimal solution. So two alternative modelling techniques based on disjunctive programming have been proposed: convex hull formulation and disjunctive branch and bound, based mainly on linear underestimators. The convex hull formulation is a natural representation of discontinuities in realistic cost functions and theoretically yields the tightest relaxation amongst the alternative modeling techniques examined in the paper. The disjunctive branch and bound algorithm performs reasonably well for small problems but the convex hull formulation tends to be computationally more efficient (less expensive) and produces reliable robust results.

Generalized Disjunctive programming (GDP) for MINLP:

Turkay and Grossmann (1998) addressed the same topic of optimization of process systems with complex investment cost functions in a new paper. The discontinuities with respect to these variables are modeled with disjunctions that are converted into tight mixed-integer constraints with the convex hull formulation for each disjunction. It is shown again that the convex hull formulation outperforms the big M formulation. The new thing is that in order to address the structural optimization of process flowsheets, they proposed a generalized disjunctive programming algorithm (GDP) in which the complex investment cost functions are formulated as embedded disjunctions. The GDP algorithm consists of MINLP subproblems for the optimization of fixed flowsheet structures and MILP master problems to predict new flowsheets to be optimized. This algorithm is rigorous for handling discontinuities in complex cost functions. It has been tested on large-scale problems and proved to be efficient and robust for structural flowsheet optimization.

Hybrid disjunctive bilevel decomposition algorithm:

Finally Van den Heever and Grossmann (1998) proposed a hybrid disjunctive bilevel decomposition algorithm motivated by the use of Generalized Disjunctive Programming (Turkay and Grossmann, 1998) and a bilevel decomposition technique (Iyer and Grossmann, 1998). Because MINLP problems are known to be NP-complete (Garey and Johnson, 1978) meaning they require exponential solution times in the worst case, there was a clear need for developing more efficient algorithms and models. This difficulty becomes particularly acute for multiperiod MINLP problems due to the large increase in the number of variables and constraints with each additional period. The basic idea behind this hybrid algorithm is that an outer loop iterates between the design problem (DP) and the operation and expansion planning (OEP), similar to the algorithm of Iyer and Grossmann, while both (DP) and (OEP) are solved through inner loops using the disjunctive OA algorithm. In this work, the operation and expansion planning are incorporated into one subproblem, (OEP), whereas Iyer and Grossmann considered these planning decisions separately. The proposed method is applicable to nonlinear problems with guarantees of an optimal solution if the problem is convex. Results from a numerical example show that the proposed method shows a significant decrease in total solution time compared to DICOPT++.

2.2.4 Combined methods

As mentioned earlier, thermodynamic methods of process synthesis are very useful for the design of complex and energy intensive processes, but they cannot be used simultaneously with material balances. Algorithmic methods are simultaneous, but they are difficult to solve for complex and energy intensive processes because the number of variables increases with the number of combinations. Because of their nature, these algorithmic methods can accommodate the other two approaches (heuristic and thermodynamic). Thus we can approach the optimal design for complex and intensive processes by combining thermodynamic and algorithmic methods.

Kovac and Glavic (1995) address in their paper this combined approach. In the first step they eliminate unpromising structures and they include new, potentially good ones by studying an Extended Grand Composite Curve. Then in a second step they can optimize the superstructure obtained by MINLP. Analyzing the problem with thermodynamic method allows obtaining a simplified superstructure that can be solved with the MINLP. In future, they would like to study energy recovery and material flow optimization simultaneously. In a retrofit case studied, they have targeted energy saving using rigorous models and fixed flow rates to find two promising structures, and then by using DICOPT in GAMS they determined the best alternative and its parameters.

Hostrup, Gani, Kravanja, Sorsak and Grossmann (2001) present an integrated approach, based on thermodynamic insights and structural optimization, together with a simulation engine, to the solution of process synthesis, design and analysis. The thermodynamic insights based technique gives a non-optimal flowsheet used as a starting point by the structural optimization based technique. The latest makes use of Modeling/Decomposition (M/D) technique as well as a logic based Outer-Approximation algorithm to solve the Generalized Disjunctive Programming (GDP) model obtained. They use MIPSYN (a successor of PROSYN) as a disjunctive MINLP computer environment on several cases studied. This integration also made it possible to extend the application range of both techniques. For the thermodynamic based technique, it is possible now to compare the generated flowsheets against the optimal while for the structural optimization technique it is now possible to formulate a well defined and concise mathematical problem. There are, however, certain limitations that still need to be addressed in future. The need for process models for the two techniques to be the same (meaning using realistic and not simple models for the flowsheet optimization step) is one of them. Finally, as full formulation of disjunctions gives the tightest mixed-integer constraints, its implementation within MIPSYN is currently being investigated for the robust selection of process alternatives as well as for the robust optimization of processes with complex cost functions defined over wide ranges of equipment sizes and operational conditions.

2.3 Conclusion of this survey

From this literature survey we were able to get an overall view of optimization techniques developed in the last years and their applications to current industrial problems. Even though some work has still to be done in future as for instance on nonconvexities or disjunctions to ensure that the global optimum is achieved, algorithmic methods coupled with recent computer capabilities provide a powerful and robust tool for flowsheet optimization.

In the next section of this thesis we will study more closely algorithmic methods that we will consider for our project, as the ones implemented in GAMS.

Chapter 3

MINLP problems

In this section we will go into more detail on how a MINLP problem is generally represented and then on how optimization techniques mentioned earlier actually find the optimum solution. Nevertheless we won't go into too much detail as far as the mathematical solution is concerned, this being a somewhat specialised subject. Note that some of the work presented in this section is based on papers about MINLP problems, especially by Grossmann (1996). So the purpose will be to give a background on these optimization techniques in order to understand their general ideas and operations.

3.1 Representation of MINLP

The most basic form of an MINLP problem when represented in algebraic form is as follows:

$$\min Z = f(x,y)$$

Subject to: $g_j(x,y) \leq 0 \quad j \in J$ (P1)

$$x \in X, y \in Y$$

where f and g are convex, differentiable functions, and x and y are the continuous and discrete variables, respectively. The set X is commonly assumed to be a compact set, with the continuous variables usually associated with processing parameters such as flow rates, pressures, temperatures and equipment sizing characteristics. X is usually constrained by known lower and upper bounds on the continuous variables. The discrete set Y is, in most applications, restricted to 0-1 binary variables associated with units in the superstructure. These binary variables represent the potential existence of its associated unit in the final configuration (e.g., a value of 1 will exclude it while a value of 0 will include it). The constraint functions g_j represent performance relationships for the superstructure, such as energy and material balance equations, physical constraints or design specifications. The objective function Z is typically a cost function involving both investment and operating costs. In most applications of interest the objective and constraint functions f and g are linear in y (e.g., fixed cost charges and logic constraints).

Methods that have addressed the solution of problem (P1) include the branch and bound method (BB) (Gupta and Ravindran, 1985; Nabar and Schrage, 1991; Borchers and Mitchell, 1992), the Generalized Benders' decomposition (GBD) method (Geoffrion, 1972), the Outer-Approximation (OA) method (Duran and Grossmann, 1986) and the Extended Cutting Plane (ECP) method (Westerlund and Petterson, 1992).

There is one basic NLP subproblem and one MILP subproblem that can be considered for problem (P1):

(a) NLP subproblem for fixed y^k:

$$\begin{aligned} & \min Z^k_U = f(x,y^k) \\ \text{s.t.} \quad & g_j(x,y^k) \le 0 \quad j \in J \qquad \text{(NLP1)} \\ & x \in X \end{aligned}$$

which clearly yields an upper bound Z^k_U to (P1) because it gives a possible solution to it, provided (NLP1) has a feasible solution (which may not always be the case).

(b) MILP master problem

The new predicted values y^K (or (y^K,x^K)) are obtained from a MILP master problem that is based on the K points, (x^k,y^k), k=1,...K generated at the K previous steps. Nonlinearities have been relaxed as follows:

$$\text{Min } Z^K_L = \alpha \qquad \text{(MILP)}$$

$$\text{s.t.} \quad \left.\begin{aligned} & \alpha \ge f(x^k,y^k) + \nabla f(x^k,y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ & g_j(x^k,y^k) + \nabla g_j(x^k,y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \end{aligned}\right\} k = 1,\dots,K$$

$$j \in J^k$$

$$x \in X, \quad y \in Y, \quad \alpha \in R^1$$

Where Z^K_L yields a valid lower bound to problem (P1).

The different methods classified below will refer to these subproblems.

3.2 Algorithmic methods

3.2.1 Branch and Bound

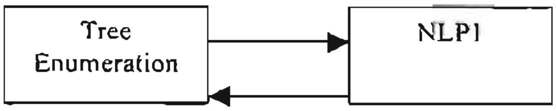


Figure 3.1: Branch and Bound

The Branch and Bound method (Gupta, 1980; Nabar and Achrage, 1991; Borchers and Mitchell, 1992) is the most widely used and probably the most successful method for solving both Integer (IP) and Mixed Integer Linear (MILP) Programming. The basic idea is to solve the problem as if it contains only continuous variables, then if the optimal solution contains non-integer values for some integer variables, it uses partitioning methods for these integer variables to divide up all of the possible solutions into subsets. This method avoids exhaustive searches using bounding methods. Thus when solving MINLP problem (P1), it solves the continuous NLP relaxation where integers are considered as real within boundaries. So it is attractive only if the NLP subproblems are relatively inexpensive to solve, or only a few of them need to be solved. This could happen either when the dimensionality of the discrete variables is low or when the continuous NLP relaxation of (P1) is tight.

Let us see on a simple IP problem how the search is done:

Maximize	$Z = 3x_1 + 2x_2$
Subject to	$x_1 \leq 2, \quad x_2 \leq 2$
	$x_1 + x_2 \leq 3.5$
	x_1 and x_2 integer variables

Step 1: Solve the problem as an LP problem by ignoring the integer restrictions:

- The LP optimal solution is at x_1=2 and x_2=1.5 and the maximum value of z is 9 (this also is the upper bound solution).
- This is not a feasible solution for the IP problem as x_2 has taken a fractional value.

Step 2: Partition x_2, to examine other integer values of x_2 that are larger or smaller than 1.5:

- Create a first new problem, LP_2, by adding the constraint $x_2 \leq 1$ to the original problem.
- Create a second new problem, LP_3, by adding the constraint $x_2 \geq 2$ to the original problem.

Step 3: Solve LP_2 and LP_3:

- The optimal solution to LP_2 is x_1=2 and x_2=1 with Z=8.

- The optimal solution to LP_3 is x_1=1.5 and x_2=2 with Z=8.5. However this not feasible as x_1 has taken a non-integer value. Note that because of the two constraints $x_2 \leq 2$ and $x_2 \geq 2$, x_2 has to be equal to 2.

Step 4: Partition x_1, to see if LP_3 has an integer solution. So we create two new problems, LP_4 and LP_5, by respectively adding the following constraints to LP_3: $x_1 \leq 1$ and $x_1 \geq 2$.

Step 5: Solve LP_4 and LP_5:

- Optimal solution to LP_4 is at x_1=1 and x_2=2 with Z= 7. This is a feasible solution to the original problem but not better than LP_2 (Z=8).
- The only possible answer for LP_5 is x_1=2 and x_2=2 (again because $x_1 \leq 2$ and $x_1 \geq 2$) but it's not a feasible solution as it violates $x_1 + x_2 \leq 3.5$. Thus there is no feasible solution for LP_5.

So the optimal solution to the original problem is Z=8 with x_1=2 and x_2=1. The sequence of LP problems is represented below:

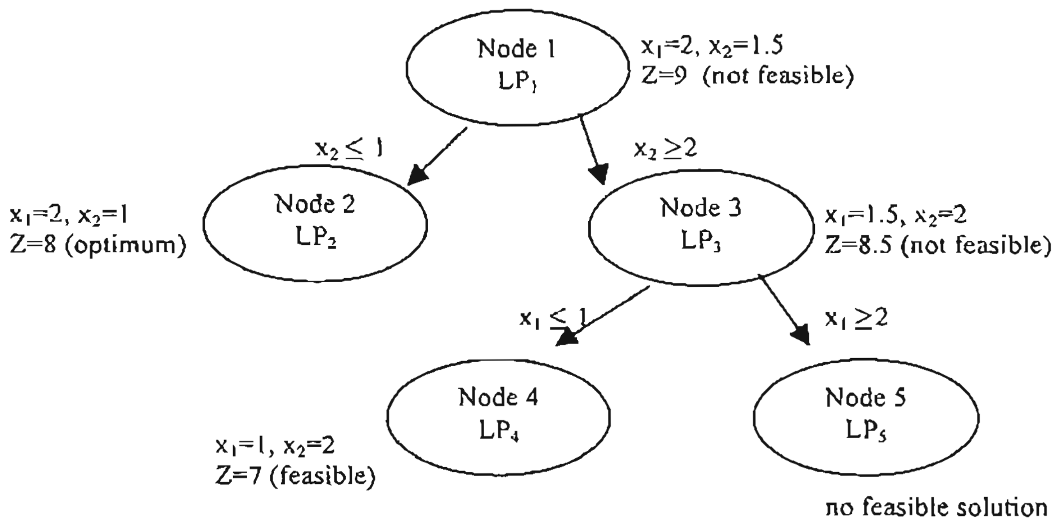


Figure 3.2: Branch and bound diagram

Note that this small example is a MILP problem. The procedure is the same for a MINLP problem with NLP problems solved at nodes instead of LP problems.

3.2.2 Outer Approximation

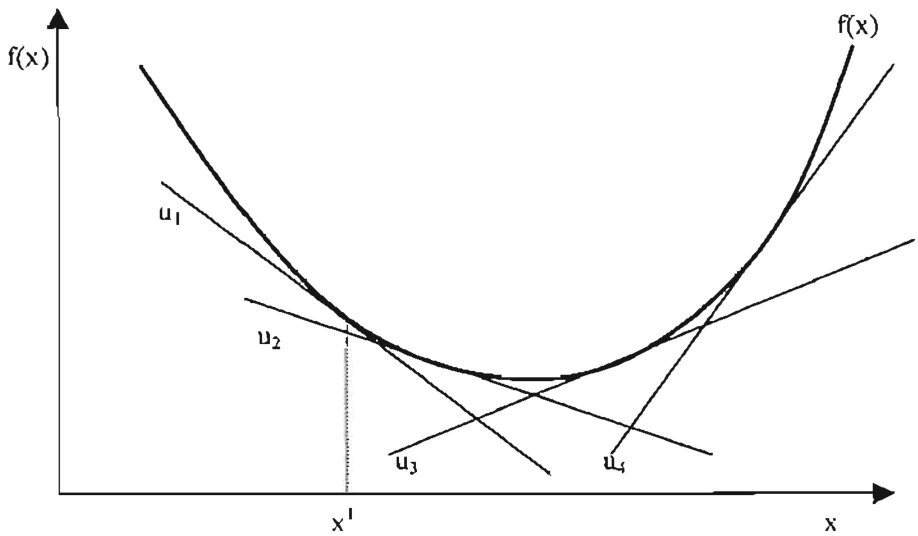


Figure 3.3: Outer approximation (at four points) of a convex function in R^1

The main idea in the proposed algorithm (Duran and Grossmann, 1986) is as follows. Because of the linearity of the discrete variables, the continuous and discrete feasible spaces of program (P1) can be independently characterized. In other words, the proposed algorithm consists then of solving an alternating finite sequence of nonlinear programming (NLP) subproblems, where binary variables are fixed (NLP1 subproblems), and relaxed versions of mixed-integer linear master programs (MILP). The NLP subproblems are solved for fixed binary variable values and they involve the optimization of the continuous parameters. The NLP subproblems provide upper bounds to the objective function while the MILP subproblems (which have only continuous variables) provide a lower bound and new configurations of the processing scheme (new sets of binary variables). That outer-approximation will define the master program in the procedure as the equivalent mixed-integer linear programming (MILP) representation of the original MINLP program (P1). Because of the potentially many continuous points required for outer-approximation, a strategy based on relaxation will be implemented to build up increasingly tight relaxations of the master program which will select discrete combinations. The continuous points for outer-approximation will be given by the optimal primal solutions of convex nonlinear programs that represent the projection of problem (P1) onto the discrete space.

Also, if infeasible NLP subproblems are found, the feasibility problem (NLPinf) is solved to provide the point x^k:

$$\begin{aligned} &\min u \\ \text{s.t.}\quad & g_j(x,y^k) \le u \qquad j \in J \qquad\qquad \text{(NLPinf)} \\ & x \in X,\quad u \in R^1 \end{aligned}$$

The OA method generally requires relatively few cycles or major iterations and is promising in applications where NLP problems are expensive to solve. It trivially converges in one iteration if f(x,y) and g(x,y) are linear. It is also important to note that the MILP master problem need not be solved to optimality. The procedure continues until the MILP problem gives a lower bound greater than the current upper bound.

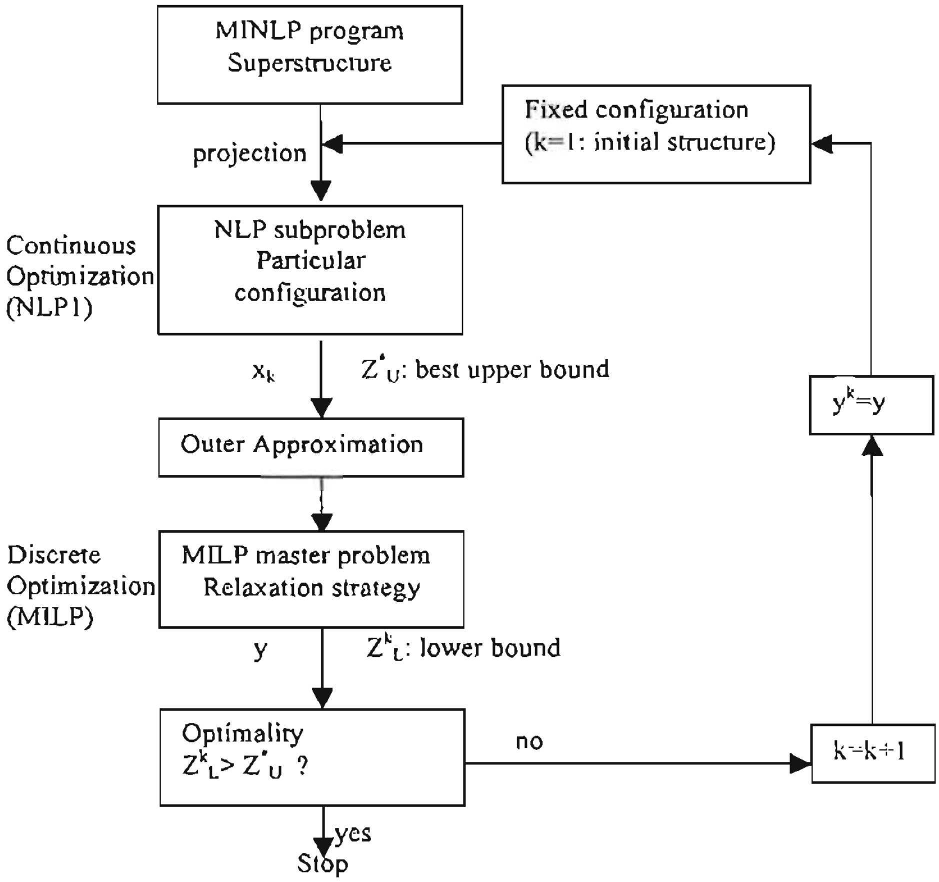


Figure 3.4: Outer Approximation algorithm

As far as the implementation of the method in computer codes is concerned, the solution of a sequence of MILP problems can become the major bottleneck in large-scale applications since the relaxed master programs will grow in size as iterations proceed. Indeed, this MILP problem includes the linear constraints from the original MINLP, as well as the linear approximations to the nonlinear functions derived at each NLP subproblem solution. This strategy thus has the drawback of increasing the size of the MILP problems and so the computational effort, but since these approximations become increasingly constrained, the associated solutions of successive master problems will give a monotonically nondecreasing sequence of lower bounds on the objective function.

Thus, the efficiency of the algorithm can be improved if the following considerations are taken into account. Firstly, integer cuts can be derived and added at each iteration so as to reduce the enumeration effort when solving subsequent relaxed master programs. Secondly, a very desirable improvement would be to keep the size of the master programs as small as possible by using a constraint dropping scheme. Thirdly, the solution of the relaxed master problem could be prematurely terminated as soon as an integer solution is found that lies below the current upper bound.

3.2.3 Generalized Benders' Decomposition

J.F.Benders (1962) devised a clever approach for exploiting the structure of mathematical programming problems with "complicating variables" (variables which, when temporaly fixed, render the remaining optimization problem considerably more tractable). Fixing the values of the complicating variables reduces the given problem to an ordinary linear problem parameterized by the value of the complicating variables vector. The algorithm he proposed for finding the optimal value of this vector employs a cutting-plane approach for building up adequate representations of firstly, the extremal value of the linear program as a function of the parameterizing vector and secondly the set of values of the parameterizing vector for which the linear program is feasible. Benders's approach was generalized by Geoffrion (1972) to a broader class of programs in which the parameterized subproblem need no longer to be a linear program.

So the GBD method is similar to the OA method in that the MINLP problem is split into two subproblems, a NLP one and a MILP one. The difference arises in the definition of the MILP master problem. In the GBD method, only active inequalities are considered unlike in the OA method, so the corresponding relaxed master problem is smaller. As has been shown by Duran and Grossmann (1996), the lower bounds of the OA method are greater than or equal to those of the GBD method. For this reason GBD commonly requires a larger number of cycles or major iterations (but the work per iteration in OA is greater due to the large size of the relaxed master problem). As the number of 0-1 variables increases, this difference becomes more pronounced. Therefore "user-supplied constraints must be added to the master problem to strengthen the bounds" (Sahindis and Grossmann, 1991). Also, Turkay and Grossmann (1994) have proved that performing one Benders iteration on the MILP master of OA is equivalent to a GBD iteration.

3.2.4 Extended Cutting Plane

The ECP method (Westerlund and Petterson, 1992), which is an extension of Kelly's cutting plane algorithm for convex NLP (Kelley, 1960), does not rely on the use of NLP subproblems and algorithms. It relies only on the iterative solution of the MILP master problem by successively adding the most violated constraint at the predicted point (x^k,y^k). Convergence is achieved when the maximum constraint violation lies within the specified tolerance. The optimal objective value of the MILP master problem yields a nondecreasing sequence of lower bounds. Note that since the discrete and continuous variables are converged simultaneously, a large number of iterations may be required. Also, the objective must be defined as a linear function. For these reasons this method is not often used especially for large problems.

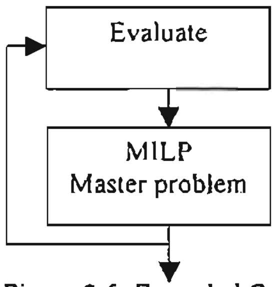


Figure 3.5: Extended Cutting plane

3.3 Extensions of MINLP methods

In this section, we present an overview of some of the major extensions of the methods presented in the previous section.

3.3.1 Quadratic Master Problems

For most problems of interest, problem (P1) is linear in y:

$$f(x,y) = \Phi(x) + c^T y \text{ and } g(x,y) = h(x) + By \tag{3.1}$$

When this is not the case Fletcher and Leyffer (1994) suggest including a quadratic approximation to the MILP master problem. As noted by Ding and Sargent (1992), who developed a master problem similar to the one done by Fletcher and Leyffer (1994), the quadratic approximations can help to reduce the number of major iterations since an improved representation of the continuous space is obtained. This, however, comes at the price of having to solve an MIQP instead of an MILP.

3.3.2 Reducing the dimensionality of the Master Problem in OA

The master problem can be rather large in the OA method. One option is to keep only the last linearization point, but this may lead to nonconvergence even in convex problems. Quesada and Grossmann (1992) proved that a rigorous reduction of dimensionality without greatly sacrificing the strength of the lower bound can be achieved in the case of what they call the "largely" linear MINLP problem. They also showed that linear approximations to the nonlinear objective and constraints can be done with a different formulation of the MILP master problem. Numerical results have shown that the quality of the bounds is not greatly degraded with the above MILP, as might happen if GBD is applied to their "largely" linear MINLP.

3.3.3 Incorporating cuts

One way to expedite the convergence in OA and GBD algorithms when the discrete variables in problem (P1) are 0-1 is to introduce the following integer cut, which has as an objective to make infeasible the choice of the previous 0-1 values generated at the K previous iterations (Duran and Grossmann, 1986):

$$\sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq |B^k| - 1 \qquad k = 1,\ldots, K \tag{3.2}$$

where $B^k = \{i|\ y^k_i=1\}$, $N^k = \{i|\ y^k_i=0\}$, $k = 1,\ldots, K$. This cut becomes very weak as the dimensionality of the 0-1 variables increases. However, it has the useful feature of ensuring that new 0-1 values are generated at each major iteration. In this way, the algorithm will not return to a previous integer point when convergence is achieved. Also, in the case of the GBD method it is sometimes possible to generate multiple cuts from the solution of an NLP subproblem in order to strengthen the lower bound (Magnanti and Wong, 1981).

3.3.4 Handling of equalities: OA/ER

One of the limitations of the OA method is that it can only handle linear equality and nonlinear/linear inequality constraints. So for the case when nonlinear equalities of the form $h(x,y)=0$ are added to (P1), there are two difficulties. First, it is not possible to enforce the linearized equalities at K points. Second, the nonlinear equations may generally introduce nonconvexities. These limitations have motivated Kocis and Grossmann (1987) to develop an Equality Relaxation Outer-Approximation (OA/ER) strategy in which the nonlinear equalities of the form $h(x,y)=0$ are replaced by the inequalities:

$$T^k \nabla h(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \tag{3.3}$$

where the diagonal matrix $T^k = \{t^k_{ii}\}$, and t^k_{ii} = sign (λ^k_i) in which λ^k_i is the optimal lagrange multiplier associated with the equation $h_i(x,y)=0$. The basic idea is to relax the nonlinear equations to inequalities. One yields an equivalent problem with the same optimum, without any more nonlinear equalities (only inequalities) so one can then use a similar reasoning as in the OA algorithm. Note that if these equations relax as the inequalities $h(x,y) \leq 0$ for all y, and $h(x,y) = 0$ is convex, this is a rigorous procedure. Also, note that in the master problem of GBD, no special provision is required to handle equations so that makes the GBD method capable of handling nonlinear equality constraints as well, but OA/ER needs less iterations. The OA/ER algorithm has been implemented in the computer package DICOPT. However, difficulties similar to those in OA arise if the equations do not relax as convex inequalities.

3.3.5 Handling zero flows: M/D

The modeling/decomposition (M/D) strategy proposed by Kocis and Grossmann (1989) is largely motivated by the need to simplify the solution of the NLP and MILP problems. Indeed, they showed that modeling can have a great impact on the quality of solutions that are obtained, as well as on the computational efficiency. The proposed strategy exploits the special structure of flowsheet synthesis problems that are to be solved with the OA/ER algorithm. It reduces the undesirable effect of nonconvexities and eliminates the optimization of "dry units" with zero flows, which are temporarily turned off in the superstructure. The solution of the NLP is simplified by optimizing only the particular flowsheet considered, instead of optimizing the entire superstructure. The MILP solution is simplified by incorporating an approximation to the particular flowsheet only at each iteration. So the proposed procedure reduces the computational effort and increases the likelihood of converging to the global optimum. This strategy has been automated in the flowsheet synthesizer PROSYN by Kravanja and Grossmann (1990).

3.3.6 Handling Nonconvexities: AP/OA/ER

Another limitation of the above techniques is the assumption of convexity for both f(x,y) and g(x,y). So when these two functions are nonconvex, two difficulties arise. Firstly, the NLP subproblems may not have a unique local optimum solution. Secondly, the master problem and its variants do not guarantee a valid lower bound Z^K_L or a valid bounding representation with which the global optimum may be cut off. Two approaches can be used to address this problem: either assume a special structure in the MINLP problem and rely on methods for global optimization (Floudas and Grossmann, 1994); otherwise, apply an heuristic strategy to try to reduce the effect of nonconvexities as much as possible. We will describe only the second approach here, with the objective of reducing the effect of nonconvexities at the level of the MILP master problem. Viswanathan and Grossmann (1990) proposed a combined penalty function and outer-approximation method where they introduce slacks in the MILP master problem to reduce the likelihood of cutting off feasible solutions.

The proposed algorithm starts by solving the NLP relaxation. If an integer solution is not found, a sequence of iterations consisting of NLP subproblems and an MILP master problem is solved. The proposed MILP master problem is based on the OA/ER algorithm and features an exact penalty function that allows violations of linearizations of nonconvex constraints. The search proceeds until no improvement is found in the NLP subproblems since bounding properties of the new MILP master problem cannot be guaranteed. Note that if the functions are convex, the MILP master problem predicts rigorous lower bounds to (P1) since all the slacks are set to zero. No theoretical guarantee can be given but the proposed method has a high degree of reliability for finding the global optimum in nonconvex problems. Clearly, this is an heuristic, but one that works reasonably well in many problems.

It should also be noted that the program DICOPT++ (Viswanathan and Grossmann, 1990), a MINLP solver commercially available (as part of GAMS; Brooke and al., 1988), is based on this specific master problem.

We have seen in this section the main ideas of the specific optimization techniques and their extension to solve MINLP problems. We will move now to the next chapter where we start the work on our specific problem of flowsheet optimization.

Chapter 4

The SAPREF steam distribution

In this section we describe in detail the current SAPREF steam network to be optimized. We present all its components such as back-pressure steam turbines or pressure-letdown stations on every header. This is done in order to get a model that represents as accurately as possible the steam distribution, which will be used for optimization later on.

The latest flowsheet can be found in appendix A. Unfortunately it has only been done in 1989 and since then some units have been removed or added. We investigated at the refinery to update it with changes that have occurred since 1989. The distribution has four different headers, each one at a constant pressure going from the high-pressure header (50 bar) to the low-pressure one (approximately 3-5 bar). The assured medium pressure and the medium pressure headers are respectively at approximately 17 and 16 bars. The steam is supplied all over the plant by these 4 headers. We will consider each of these headers as a single continuous pipeline all over the plant because all the headers at the same pressure over the different sections of the plant are linked to import or export steam from one place to another. From this representation, shown on figure 4.1, several components can be extracted and classified:

4.1 Boilers

4.1.1 High pressure producers

The refinery is producing is own HP steam from 10 boilers. The average production is approximately 6000 tons/day spread in the different sections of the HP header.

4.1.2 Medium and Low pressure producers

Steam from the HP and Assured MP steam is let down to the MP header through turbines or letdown stations. Some local MP steam producers feed the MP line with approximately 700 tons/day.

We have also boilers and processes producing around 500 tons/day of LP steam recovered by the LP header.

4.2 Process steam users

On each of these headers, steam is required as reaction steam in refinery processes, for ejectors, blowers or for indirect heating purposes such as distillation-tower reboilers, process heaters, preheaters, etc. These steam users are represented on the flowsheet by boxes that required a variable flow of steam depending on the current demand.

4.3 Steam turbines

Steam turbines are used to generate electricity, and to satisfy mechanical power demands. Note that we use the same name for the steam turbine and the utility driven by this turbine. We classified all of the steam turbines in two categories:

4.3.1 Back-pressure turbines

These turbines are exhausting steam to the following headers, either the one just after (e.g., from HP to Assured MP) or the one two levels down (e.g., form HP to MP steam)

4.3.1.1 High Pressure (HP) turbines

We have 4 back-pressure turbines driven by HP steam:

G3171, electricity generator

This is an electricity generator on the plant and its function is to supply vital utility drives with power in the event of a power cut. It is capable of running at a maximum capacity of 5 MW of electrical energy. However, due to limitations on the refinery steam demands, G3171 only runs at approximately 0.6 MW. This current low electricity production leaves SAPREF in a vulnerable position in the event of an electricity failure. It exhausts steam to Assured Medium Pressure header.

K301, platformer recycle gas compressor

This compressor consumes nearly half of the refinery HP steam, exhausted to the Assured MP steam header. This results in the refinery operation being determined by the HP steam demand without sufficient MP and LP users, thus resulting in venting. Its use is however vital as a failure would result in the coking of the catalyst.

K3471, instrument air blower

The steam turbine driving this blower and exhausting to the Assured MP steam header uses approximately 400 tons/day of HP steam.

K6101, compressor

This turbine has the distinction of being a condensing turbine with steam extraction to medium pressure. It means that around 600 tons/day of HP steam is condensed while the rest of the HP steam, approximately 1000 tons/day (depending on the current demand) is exhausted to MP steam.

4.3.1.2 Assured Medium Pressure (AMP) turbines

We have 5 back-pressure turbines driven by AMP steam and all of them are exhausting to LP steam.

K3262 is driving a forced draught fan and requires quite low steam consumption around 60 tons/day.
K3271 is also using a small amount of steam around 80 tons/day to drive an air blower.
K3272 uses approximately 50 tons/day driving a boiler blower.
U3200 represents the largest AMP steam demand by requiring around 300 tons/day whereas the unit *U3500* needs only 20 tons/day.
Note that the letter "*K*" refers to a turbine while "*U*" to a whole plant unit.

4.3.1.3 Medium Pressure (MP) turbines

We have on this header 16 turbines, all of them exhausting to LP steam as well, without sufficient LP consumers resulting in LP steam venting. We shall not explain the duty of all of them in detail. They all drive compressors or blowers. The biggest MP steam consumption is 800 tons/day from *K4402* (propane gas compressor).

4.3.2 Condensing turbines

There is actually only one turbine of this kind in the entire network. *K6102* is a condensing turbine using approximately 300 tons/day of HP steam to compress wet gas. The exhausted steam is fully condensed.

4.4 Turbine bypass and desuperheaters

On each of the three HP turbines *K301*, *K3471* and *G3171*, is added a bypass line aiming to control the actual steam flow sent into the turbine. The excess HP steam that is not used by the turbine bypasses the turbine and mixes with the exhaust steam. After the connection point a desuperheater is placed, spraying water into the steam to desuperheat it before it reaches the AMP line.

Note that a fourth desuperheater using the same principle is located after the HP-MP letdown station just before the MP header.

4.5 Letdown stations

These pressure-letdown stations are pressure control valves that exhaust the excess steam not used by users or turbines from one header to the following one (e.g., MP-LP letdown station) or the one two levels down (e.g., HP-MP letdown station).

In this section we list all of the elements of the SAPREF steam distribution. This leads us to the following representation of the network on figure 4.1. We put together all the HP feeders, MP feeders, letdown stations and steam users on every header and represented only one of these components as the sum of all of them. Thus they will be represented with a steam flow, production in case of the feeders and demand in case of the users, equal to the sum of all the steam flows of the same kind of elements.

In the following chapter we shall use this representation to make a mathematical model of the distribution.

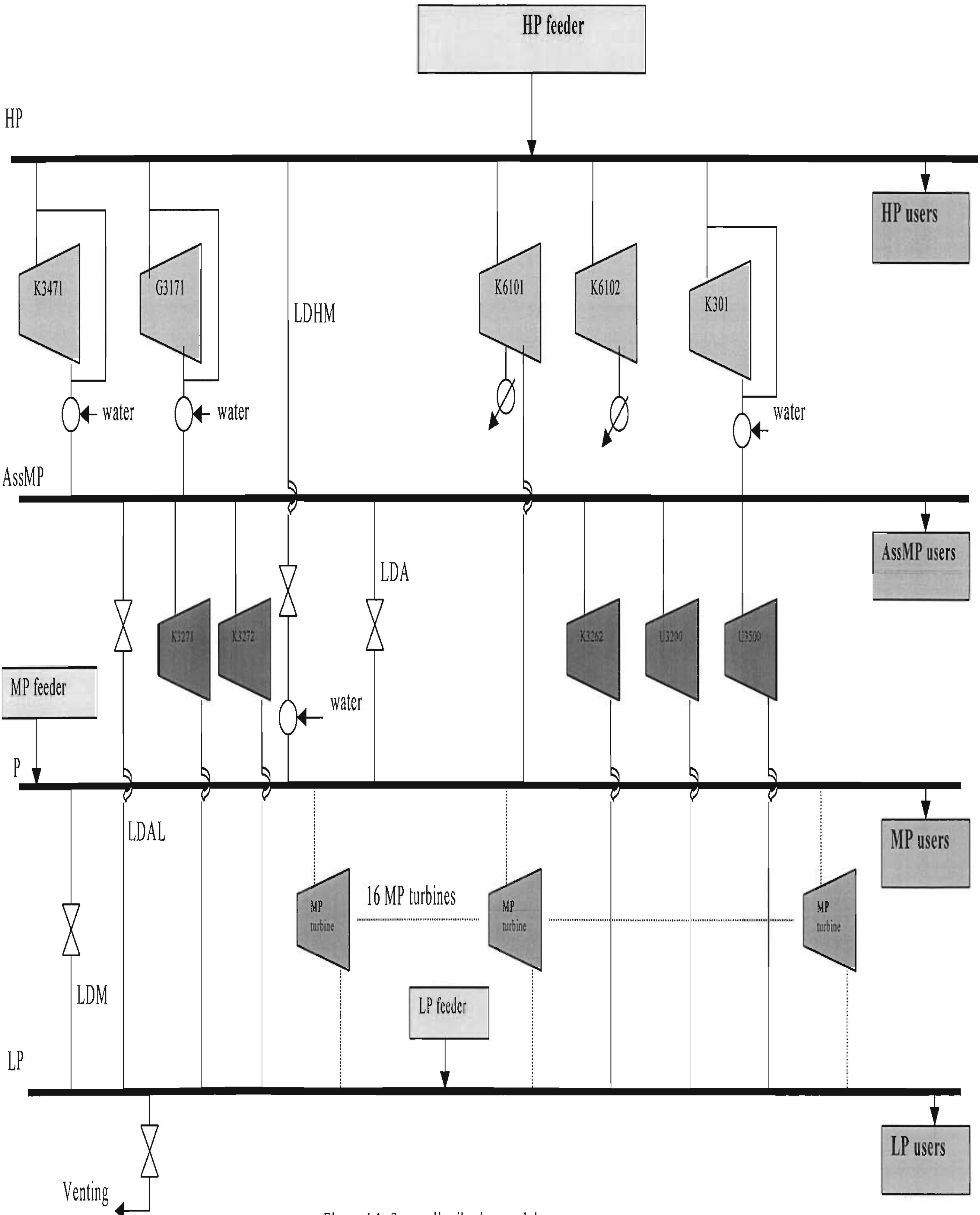


Figure 4.1: Steam distribution model

Chapter 5

Modeling of the steam distribution

This chapter is the modeling part of our project. The purpose is, using the representation of the flowsheet in the previous section, to make an accurate mathematical model of the steam distribution that will be used later for optimization. This model is basically made from the steam mass and energy balances on every header. The last section deals with the cost functions involved.

All of the flows given are in tons/day and mass enthalpies are given in kJ/ton.

5.1 Decision variable ε

At this stage of our work we have to introduce the decision variable ε_i on steam turbine *i*. Indeed the objective of this project is to determine for each steam turbine currently in use whether or not savings can be done by switching it to an electrical drive to meet the power demand. Thus, this will be the role of the optimizer, taking into account constraints and boundaries of the model, to minimize costs involved in the objective function by determining for each turbine whether:

$\varepsilon_i = 0$, meaning that we keep the turbine *i* in use or

$\varepsilon_i = 1$, meaning that steam turbine *i* is replaced by an electrical drive.

So $\varepsilon_i \in \{0,1\}$ is a binary variable that will appear in our model of the steam distribution. For instance if F_i is the flow of steam in tons/day, previously used by turbine *i*, then the new flow F_i^* in tons/day, using ε_i will be:

$$F_i^* = F_i - F_i * \varepsilon_i = (1 - \varepsilon_i) * F_i$$

We can check in the equation above that if turbine *i* is left in the distribution then $\varepsilon_i = 0$, so there is no change and $F_i^* = F$. But if turbine *i* has to be switched off to minimize the objective function, meaning $\varepsilon_i = 1$, then $F_i^* = 0$ because the turbine is no more in use and obviously no steam is required here.

5.2 Generator G3171

This electricity generator G3171 is one of the HP steam turbines of the steam distribution. Its decision variable ε_{G3171} will be, however, fixed to 0 in our model as its function is to supply vital utility drives with power in the event of a power cut and so this generator will obviously not be available for replacement.

5.3 Mass balance

We proceed to the theoretical mass balance on every header using the conservation principle:

$$Accumulation = Inflow + Generation - Consumption - Outflow \tag{5.1}$$

As there is no reaction, nor accumulation in our case, equation (5.1) becomes merely

$$Inflow = Outflow \tag{5.2}$$

5.3.1 HP steam balance

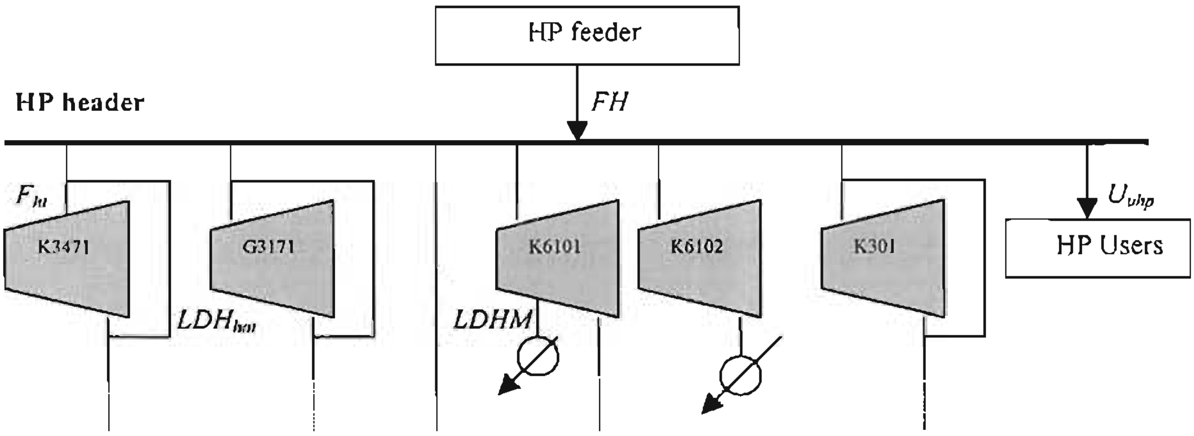


Figure 5.1: HP header mass balance

5.3.1.1 Inflow

The only HP steam produced here is from all of the HP boilers. So we just have:

$$Inflow = FH \tag{5.3}$$

where FH is the total flow of HP steam produced by the ten boilers in tons/day.

5.3.1.2 Outflow

The HP steam is used through turbines, turbine bypasses, letdown stations and users

Steam turbines:

Steam consumption of HP turbines if no turbine is switched over to an electrical drive, HTP_0:

$$HPT_0 = F_{K3471} + F_{G3171} + F_{K6101} + F_{K6102} + F_{K301} = \sum_{ht} F_{ht} \qquad ht \in \text{HT} \tag{5.4}$$

with HT ={ht | HP turbines K3471,G3171,K6101,K6102,K301} and F_{ht} the flow of steam used by turbine *ht*. However if any of these turbines is replaced with an electrical drive by the optimizer then we have to subtract the corresponding steam flows required for the turbine. So the steam consumption of HP turbines taking into account this possibility becomes, HPT:

$$HPT = \sum_{ht} F_{ht} - \sum_{ht} \varepsilon_{ht} F_{ht} = \sum_{ht} \{(1 - \varepsilon_{ht}) * F_{ht}\} \qquad ht \in \text{HT} \tag{5.5}$$

with ε_{ht} the decision variable on HP turbines. So if $\varepsilon_{ht} = 0$, meaning that the turbine is left in the distribution, we see that the corresponding flow F_{ht} appears in the equation (5.5) but if the turbine *ht* is switched off then $\varepsilon_{ht} = 1$ and F_{ht} is removed from the equation.

Turbine bypasses:

The flow of steam trough the HP turbine bypasses, HPB, is as follows:

$$HPB = LDH_{K3471} + LDH_{G3171} + LDH_{K301} = \sum_{hat} LDH_{hat} \qquad hat \in \text{HAT} \tag{5.6}$$

with HAT={hat| HP turbines with bypass, K3471, G3171, K301} and LDH_{hat} the flow of steam letdown through the bypass of turbine *hat*. Note that even if the turbine *hat* is switched over to an electrical drive the bypass is kept in use as a simple letdown station.

Letdown stations:

As mentioned in the previous section we put together all of the letdown stations from the HP header and consider only one with a flow of steam LDHM equal to the sum of all of them. From the HP header we have only two letdown stations exhausting to medium pressure.

Steam users:

Let us call U_{uhp} the steam flow required by the user $uhp \in$ UHP with UHP = {*uhp*| HP steam users}.

So the total steam demand for HP steam users will appear as $\sum_{uhp} U_{uhp}$.

Finally we can express the outflow on the HP header:

$$Outflow = HPT + HPB + LDHM + \sum_{uhp} U_{uhp} \qquad uhp \in \text{UHP} \tag{5.7}$$

After substitution of *HPT* and *HPB* by respectively (5.4) and (5.5), equation (5.6) becomes:

$$Outflow = \sum_{ht} \{(1-\varepsilon_{ht}) * F_{ht}\} + \left(\sum_{hat} LDH_{hat}\right) + LDHM + \sum_{uhp} U_{uhp} \tag{5.8}$$

$ht \in$ HT, $hat \in$ HAT, $uhp \in$ UHP

Eventually, using (5.3) and (5.8) in (5.2) leads to the following HP steam balance:

$$\boxed{FH = \sum_{ht} \{(1-\varepsilon_{ht}) * F_{ht}\} + \left(\sum_{hat} LDH_{hat}\right) + LDHM + \sum_{uhp} U_{uhp}} \tag{5.9}$$

$ht \in$ HT, $hat \in$ HAT, $uhp \in$ UHP

5.3.2 AMP steam balance

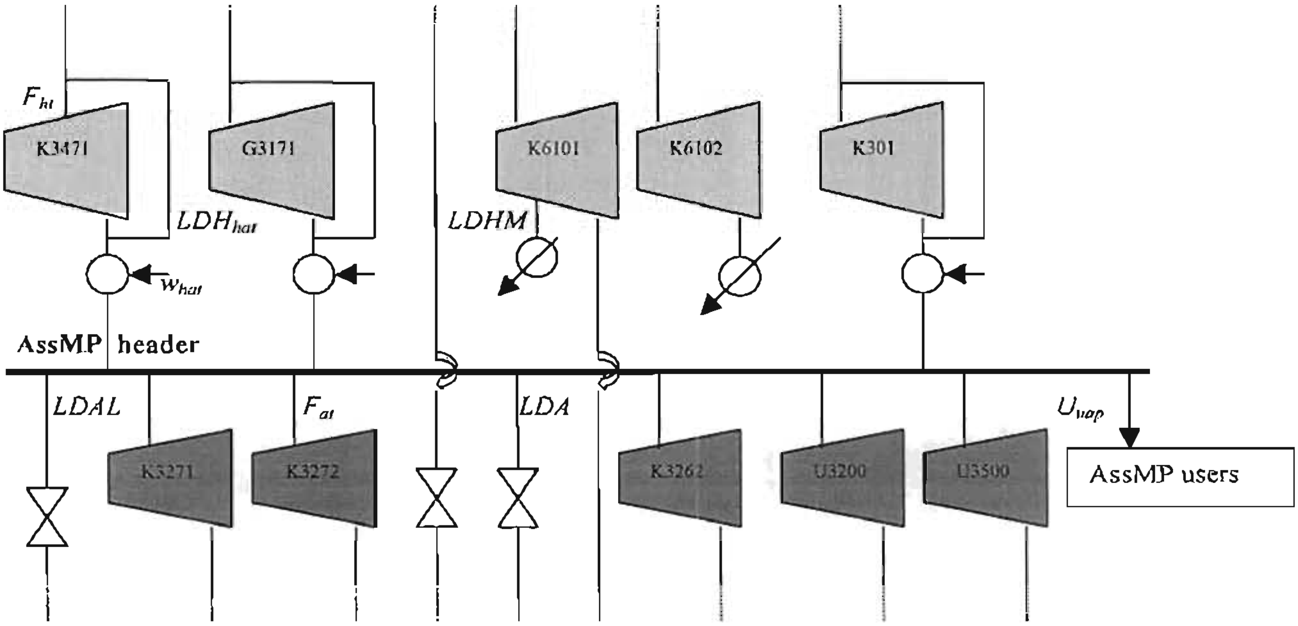


Figure 5.2: AMP header mass balance

Using the same equation (5.2) we proceed to the AMP steam balance

5.3.2.1 Inflows

The only producers of AMP steam are the three *hat* turbines K3471, G3171 and K301 and their corresponding bypasses. Note that K6101 is exhausting to MP steam and K6102 is a condensing turbine. There is no local feeder on this header and no letdown station is letting down HP steam to AMP steam.

Desuperheaters after the "hat" turbines:

At this point we need to do a local mass balance on desuperheaters located just after the *hat* turbines. The amount of steam coming in is equal to the exhausted steam, F_{hat} and the steam letdown through the bypass, LDH_{hat}. Added to this steam is a certain quantity of water w_{hat}, which is sprayed and fully vaporized to desuperheat the steam. Bearing in mind that if the "hat" turbine considered is switched off then we will not have the F_{hat} term. Then the total inflow on the AMP header can be written as follows:

$$Inflow = \sum_{hat}\left(\{(1-\varepsilon_{hat})*F_{hat}\} + w_{hat} + LDH_{hat}\right) \qquad hat \in \text{HAT} \qquad (5.10)$$

5.3.2.2 Outflows

The AMP steam is then used by steam turbines, letdown stations and users:

Steam turbines:

We have 5 steam turbines using AMP steam. Let us define AT = {at | AMP steam turbines, K3262, K3271, K3272, U3200, U3500}. Then once again depending on whether or not these turbines are switched to electrical drives, the total flow of steam required from the AMP header, AMPT, is:

$$AMPT = \sum_{at}\{(1-\varepsilon_{at})*F_{at}\} \qquad at \in \text{AT} \qquad (5.11)$$

Letdown stations:

We have from this header two different kinds of letdown stations. Some of them are letting down to medium pressure so we call the total amount of steam passing through these letdown stations LDA. The others are exhausting to low pressure a total of LDAL tons/day of steam. Note here that we have as well two kinds of letdown stations exhausting to MP steam, one with a desuperheater just after the pressure valve and one without any desuperheater following the valve. As the letdown stations followed by a desuperheater are not currently in use and are only installed for emergency reasons we will not show them in our balance. So LDA is the total steam flow through letdown stations exhausting to MP steam with no desuperheater.

Steam users:

The total AMP steam requirement for users on this level is: $\sum_{uap} U_{uap}$ $\quad uap \in \text{UAP}$

with UAP = {*uap*| AMP steam users} and U_{uap} the steam flow in tons/day required for user *uap*.

So from all that we can write the total AMP steam outflow:

$$Outflow = \sum_{at}\{(1-\varepsilon_{at})*F_{at}\} + \sum_{uap} U_{uap} + LDA + LDAL \qquad at \in \text{AT},\ uap \in \text{UAP} \qquad (5.12)$$

Finally the AMP steam balance is:

$$\boxed{\sum_{hat}\left(\{(1-\varepsilon_{hat})*F_{hat}\}+w_{hat}+LDH_{hat}\right)=\sum_{at}\{(1-\varepsilon_{at})*F_{at}\}+\sum_{uap}U_{uap}+LDA+LDAL}$$

$hat \in$ HAT, $at \in$ AT, $uap \in$ UAP (5.13)

5.3.3 MP steam balance

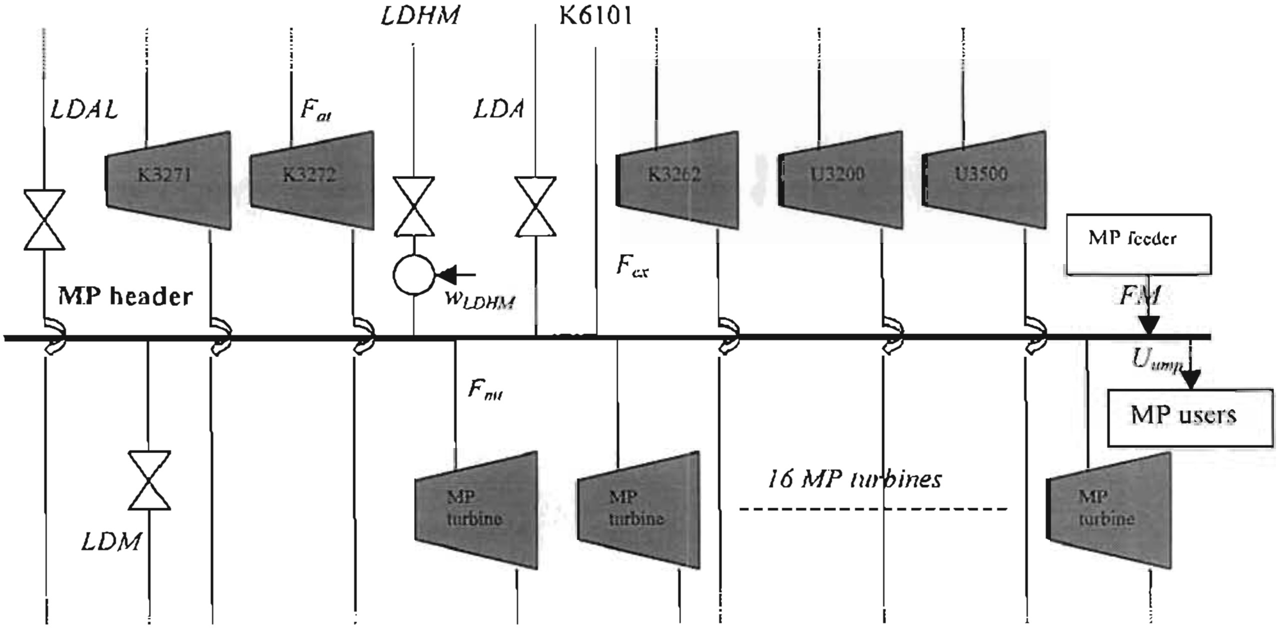


Figure 5.3: MP header mass balance

We proceed to the MP balance using the same technique.

5.3.3.1 Inflows

On the MP header, we have several MP steam producers: Local MP feeder, exhaust steam from K6101, letdown stations.

Local MP feeder:

The MP steam produced by this feeder is called *FM* in tons/day and depends on the plant working conditions.

Steam from K6101:

As we have seen earlier, K6101 is a condensing turbine with steam extraction to medium pressure.

Defining y_{K6101} as the ratio of steam exhaust, F_{ex}, on total steam used by K6101, F_{K6101}, we have:

$$F_{ex} = y_{K6101} * F_{K6101} \tag{5.14}$$

The flow of steam coming from K6101 to the MP header, FMP_{K6101}, can be written depending on the decision variable ε_{K6101}:

$$FMP_{K6101} = (1-\varepsilon_{K6101}) * F_{ex} \tag{5.15}$$

Substituting F_{ex} in (5.15) we obtain:

$$FMP_{K6101} = (1-\varepsilon_{K6101}) * F_{K6101} * y_{K6101} \tag{5.16}$$

Letdown stations:

We have two types of letdown stations exhausting to MP steam:

The first one is the one letting down steam from the HP header with a desuperheater following the pressure valve. So to the LDHM defined earlier we add the quantity of water w_{LDHM} vaporized into the desuperheater. The second type is the steam from the AMP header, LDA.

So putting together these three kinds of MP steam producers we get:

$$Inflow = FM + FMP_{K6101} + LDA + LDHM + w_{LDHM} \tag{5.17}$$

And finally using (5.16) we get:

$$Inflow = FM + (1-\varepsilon_{K6101}) * F_{K6101} * y_{K6101} + LDA + LDHM + w_{LDHM} \tag{5.18}$$

5.3.3.2 Outflows

As far as outflows are concerned, we have all 16 steam turbines, letdown stations to LP steam and MP users.

MP steam turbines:

Let us call MPT the total amount of steam required for the 16 steam turbines and let us define MT as follows: MT={mt | MP steam turbines, K4402,..., P3701C}

Using these notations with ε_{mt} the decision variable on turbine *mt*, we have:

$$MPT = \sum_{mt} \{(1-\varepsilon_{mt}) * F_{mt}\} \qquad mt \in \text{MT} \tag{5.19}$$

Where F_{mt} is the flow of MP steam used by turbine *mt*.

Letdown stations:

We have only one type of letdown station, which is taking steam from the MP header and exhausting to the LP line. So we call *LDM*, the total flow of steam exhausted through these letdown stations.

MP steam users:

Finally, MP steam required for processes or indirect heating is equal to $\sum_{ump} U_{ump}$ $ump \in$ UMP

where UMP = {*ump*| MP steam users} and U_{ump} the MP steam flow in tons/day required for user *ump*.

So from all of these equations we can write the MP steam outflow:

$$Outflow = \sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\} + \sum_{ump} U_{ump} + LDM \qquad mt \in \text{MT},\ ump \in \text{UMP} \tag{5.20}$$

and eventually, replacing (5.18) and (5.20) in (5.2) we get the MP steam balance:

$$\boxed{FM + (1-\varepsilon_{K6101})*F_{K6101}*y_{K6101} + LDA + LDHM + w_{LDHM} = \sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\} + \sum_{ump} U_{ump} + LDM}$$

$$mt \in \text{MT},\ ump \in \text{UMP} \tag{5.21}$$

5.3.4 LP steam balance

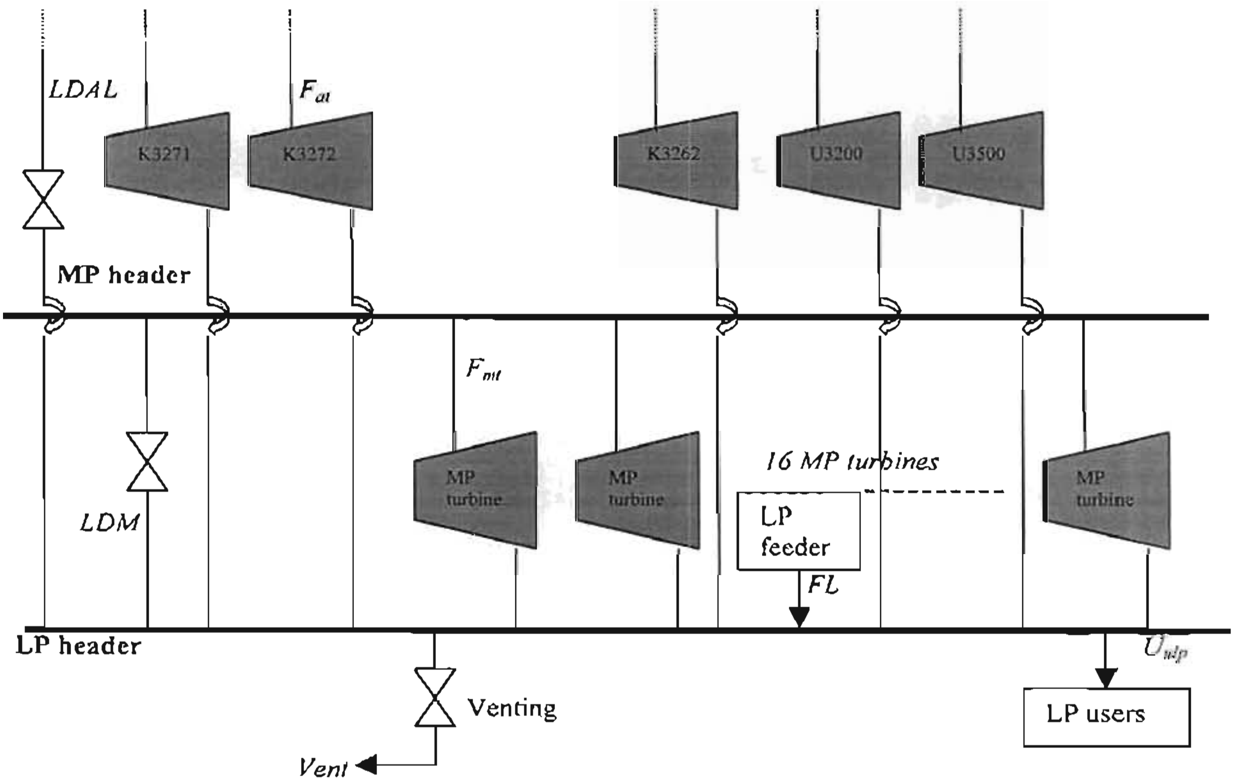


Figure 5.4: LP header mass balance

5.3.4.1 Inflows

On this last level, the steam coming in is from the AMP and MP steam turbines, both exhausting to LP steam, as well as from letdown stations and a local LP steam producer.

LP steam producer:

Some local processes in the distribution send steam to the LP header. We will consider only one local LP steam feeder, producing a flow FL of LP steam in tons/day, depending on the working conditions. This flow represents the sum of all the flows of local feeders providing steam to this header.

AMP steam turbines:

With the previous notations used, the LP steam coming from the outlets of AMP steam turbines can actually be written as equal to $AMPT$, the flow of AMP steam required for AMP turbines. Indeed there is no desuperheater after these kinds of turbines so the flow of AMP steam coming through the inlet is equal to the flow of LP steam at the outlet.

So the flow of LP steam coming from AMP turbines, $AMPT_{out}$ is equal to:

$$AMPT_{out} = AMPT = \sum_{at} \{(1-\varepsilon_{at}) * F_{at}\} \qquad at \in \mathrm{AT} \tag{5.22}$$

MP steam turbines:

For the same reasons and assumptions mentioned above, the LP steam coming from MP turbines (MPT_{out}) is equal to MPT, the flow of MP steam required for these turbines.

So we get: $$MPT_{out} = MPT = \sum_{mt} \{(1-\varepsilon_{mt}) * F_{mt}\} \qquad mt \in \mathrm{MT} \tag{5.23}$$

Letdown stations:

The last two LP steam producers are firstly the letdown station expanding AMP steam to LP steam, bringing the flow LDAL down to the LP line. And secondly the letdown station expanding a total flow of LDM medium pressure steam to LP steam.

So the total inflow is:

$$Inflow = FL + AMPT_{out} + MPT_{out} + LDAL + LDM \tag{5.24}$$

Or, using (5.22) and (5.23):

$$Inflow = FL + \sum_{at} \{(1-\varepsilon_{at}) * F_{at}\} + \sum_{mt} \{(1-\varepsilon_{mt}) * F_{mt}\} + LDAL + LDM \tag{5.25}$$

$at \in \mathrm{AT}$, $mt \in \mathrm{MT}$

5.3.4.2 Outflows

The steam pressure has now become too low for further use in turbines so the only LP steam consumers are the users in processes and the rest of LP steam has to be vented.

So using ULP = {ulp| LP steam users} and U_{ulp} the LP steam flow in tons/day required for user ulp, we get the total LP outflow:

$$Outflow = \sum_{ulp} U_{ulp} + Vent \qquad ulp \in \text{ULP} \tag{5.26}$$

with *Vent*, the flow in tons/day of LP steam vented to atmosphere.

So the LP steam balance is written as follows:

$$\boxed{FL + \sum_{mt}\{(1-\varepsilon_{mt}) * F_{mt}\} + \sum_{at}\{(1-\varepsilon_{at}) * F_{at}\} + LDM + LDAL = \sum_{ulp} U_{ulp} + Vent}$$

$$at \in \text{AT},\ mt \in \text{MT},\ ulp \in \text{ULP} \tag{5.27}$$

5.4 Steam imbalances and error terms

These equations are unfortunately not quite true in an actual industrial problem because they are only a representation of what theoretically happens. By investigating at the refinery we found that we could not rely on these equations to represent accurately the steam distribution due to measurement errors and possible unmeasured flows. Indeed, we proceed to an overall steam balance at the refinery using the computer package Pross (the plant database access system) and asking of the SAPREF staff on site when information was missing. We first met the problem of updating all the tag numbers appearing on the last version of the flowsheet made in 1996. So we found the updated tag numbers required by using Pross, interviewing personnel in the control room and using the computer system in the control room. Once this had been done we were able to proceed to the overall mass balance and found some percentage error imbalances up to 60 %. This can be explained by several reasons:

- Faulty flow meters that are reading negative flows, or are over reading or under reading
- Inaccurate estimations of streams that are lacking a flow meter
- Steam which cannot be accounted for due to leaks
- Missing units recently added or taken out.

Another reason for these imbalances is the confusion that occurs, depending on the flowsheets we are using, between the AMP and MP headers that are sometimes erroneously switched on drawings or considered as only one header. In the case of contradictory information we investigated together with the SAPREF staff. We did these steam balances on both an hourly basis for the 6th December 2000 and on a daily basis for the months of March and December 2000. We made sure that at these times of the year no major shutdowns were done that would have resulted in flow changes. Here is the average percentage errors imbalances we have:

	HP header		AMP header	
Data set	Mass imbalance (tons/day)	% imbalance	Mass imbalance (tons/day)	% imbalance
Mar-00	-668	-11	2033	53
Dec-00	-426	-7	1854	48
06-Dec	-582	-10	1933	51

Table 5.1: HP and AMP header mass imbalances

	MP header		LP header	
Data set	Mass imbalance (tons/day)	% imbalance	Mass imbalance (tons/day)	% imbalance
Mar-00	-687	-21	22	1
Dec-00	-1567	-57	1115	24
06-Dec	-1559	-58	729	17

Table 5.2: MP and LP header mass imbalances

To tackle this problem we decided to incorporate in our mass balances some error terms in order to take into account these imbalances on each header. In so doing we get an accurate model of the measured steam distribution. Note that we want to work on and optimize the steam distribution relatively to what it is currently, so these steam imbalances will stay as they stand as part of our model and will not affect the rest of it. Overall mass imbalance has always been a major issue at the refinery and to correct it is not part of our work. Required figures for us are steam consumption of the components described in the previous chapter, e.g. steam turbines.

In future our steam balance will be:

$$Inflows(X) = Outflows(X) + F_{eX} \tag{5.28}$$

with F_{eX} error term in tons/day on the header X, X ∈ {H, A, M, L}. This term can be either positive (if the theoretical steam demand is lower than the steam produced) or negative (if the calculated demand is higher than the steam coming to the header).

So the mass balances done above become:

HP steam balance:

$$\boxed{FH = \sum_{ht}\{(1-\varepsilon_{ht})*F_{ht}\} + \left(\sum_{hat} LDH_{hat}\right) + LDHM + \sum_{uhp} U_{uhp} + F_{eH}} \tag{5.29}$$

ht ∈ HT, *hat* ∈ HAT, *uhp* ∈ UHP

AMP steam balance:

$$\boxed{\sum_{hat}\left(\{(1-\varepsilon_{hat})*F_{hat}\} + w_{hat} + LDH_{hat}\right) = \sum_{at}\{(1-\varepsilon_{at})*F_{at}\} + \sum_{uap} U_{uap} + LDA + LDAL + F_{eA}}$$

hat ∈ HAT, *at* ∈ AT, *uap* ∈ UAP (5.30)

MP steam balance:

$$\boxed{FM + (1-\varepsilon_{K6101})*F_{K6101}*y_{K6101} + LDA + LDHM + w_{LDHM} = \sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\} + \sum_{ump} U_{ump} + LDM + F_{eM}}$$

mt ∈ MT, *ump* ∈ UMP (5.31)

LP steam balance:

$$\boxed{FL + \sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\} + \sum_{at}\{(1-\varepsilon_{at})*F_{at}\} + LDM + LDAL = \sum_{ulp} U_{ulp} + Vent + F_{eL}}$$

$at \in$ AT, $mt \in$ MT, $ulp \in$ ULP (5.32)

Figure 5.14 at the end of this chapter shows the steam distribution model with these error terms incorporated.

5.5 Energy balance

We have done in the previous section an overall mass balance of the steam distribution. However this is not enough to represent the distribution as temperatures (and thus steam enthalpy) might change when steam turbines are removed from the network, and so steam flow rates would change. To deal with this case we proceed now to energy balances on every header. This is done by using the energy conservation principle:

Power in = Power out (5.33)

Or using what have just been done:

$Inflows*H^{in} = Outflows*H^{out}$ (5.34)

where *Inflows* and *Outflows* are mass steam flow rates in tons/day coming in and leaving the header considered as calculated in the steam balance. And H^{in} and H^{out} are respectively the mass enthalpy (kJ/tons) of the steam flow rates coming in and leaving the header. The pressure on these headers is kept constant so only the change in temperatures can affect enthalpy.

5.5.1 HP energy balance

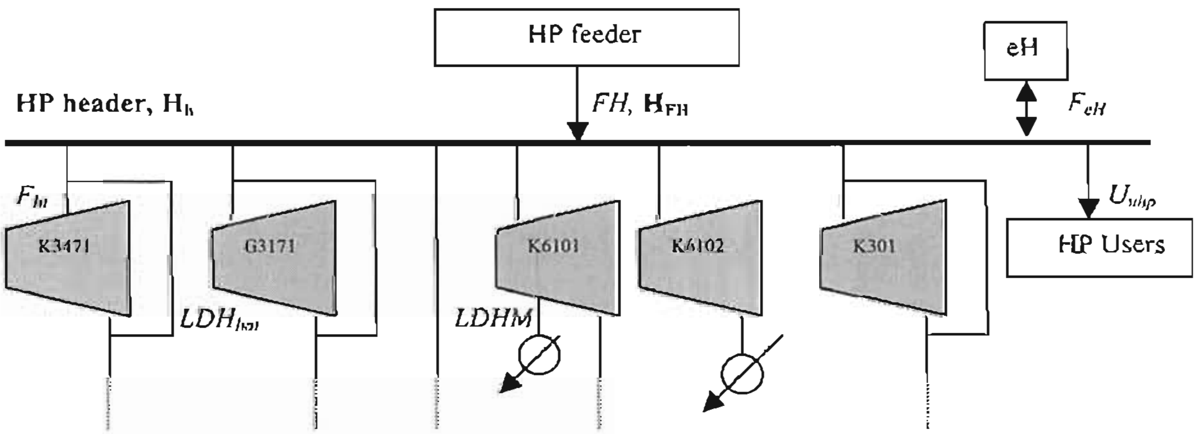


Figure 5.5: HP header energy balance

5.5.1.1 Energy inflows

As seen in equation (5.3), boilers produce a flow rate *FH* of HP steam.

So the power coming to the HP header is the mass enthalpy of this HP steam produced, H_{FH}, times the flow:

$$Power_{in} = FH * H_{FH} \qquad (5.35)$$

5.5.1.2 Energy outflows

Steam turbines

The power required by HP steam turbines, P_{HPT}, is defined as follows:

$$P_{HPT} = \left(\sum_{ht} \{ (1 - \varepsilon_{ht}) * F_{ht} \} \right) * H_h \qquad ht \in \mathrm{HT} \qquad (5.36)$$

where H_h is the mass enthalpy in kJ/tons of the HP header

Turbine bypass

Power leaving the HP header through these bypasses, P_{by}, is the flow rate times enthalpy of the HP header, H_h. Using (5.6) we get:

$$P_{by} = \left(\sum_{hat} LDH_{hat} \right) * H_h \qquad hat \in \mathrm{HAT} \qquad (5.37)$$

Letdown stations

Once again the power inside the letdowns, P_{LDHM}, is the flow rate times H_h:

$$P_{LDHM} = LDHM * H_h \qquad (5.38)$$

HP steam users

The power required for users needs to account for the useful energy available in the steam. So we assume that the steam is expanded to 1 bar and condensed at 100°C. Thus the users outlet enthalpy, h0, is the enthalpy of water at 100°C and 1 bar: 400 kJ/kg. Note that this assumption will be made for steam users from all headers. So the power required by users, P_{UHP} is the flow rate times the enthalpy drop across these users:

$$P_{UHP} = \left\{ \sum_{uhp} U_{uhp} \right\} * (H_h - h_0) \quad uhp \in \mathrm{UHP} \qquad (5.39)$$

Error terms

We have to consider these terms in the energy balance as well as they represent a flow of steam to balance the steam imbalance. As error terms generally come from leaks or faulty measurements on users consumption, we will consider them in the energy balance in the same way we consider steam users so the power loss, P_{eH}, associated with the flow rate steam loss F_{eH} is:

$$P_{eH} = F_{eH} * (H_h - h_0) \tag{5.40}$$

So we have the total power outflow:

$$Power_{out} = P_{HPT} + P_{hy} + P_{LDHM} + P_{UHP} + P_{eH} \tag{5.41}$$

And substituting expressions of these enthalpies we get:

$$Power_{out} = \left(\sum_{ht}\{(1-\varepsilon_{ht}) * F_{ht}\}\right) * H_h + \left(\sum_{hat} LDH_{hat}\right) * H_h + LDHM * H_h +$$
$$\left(\left\{\sum_{uhp} U_{uhp}\right\} + F_{eH}\right) * (H_h - h_0)$$

$$ht \in \mathrm{HT},\ hat \in \mathrm{HAT},\ uhp \in \mathrm{UHP} \tag{5.42}$$

So the HP steam power balance is written, using (5.35) and (5.42) in (5.33):

$$\boxed{FH * H_{FH} = \left(\sum_{ht}\{(1-\varepsilon_{ht}) * F_{ht}\}\right) * H_h + \left(\sum_{hat} LDH_{hat}\right) * H_h + LDHM * H_h + \left(\left\{\sum_{uhp} U_{uhp}\right\} + F_{eH}\right) * (H_h - h_0)}$$

$$ht \in \mathrm{HT},\ hat \in \mathrm{HAT},\ uhp \in \mathrm{UHP} \tag{5.43}$$

5.5.2 AMP energy balance

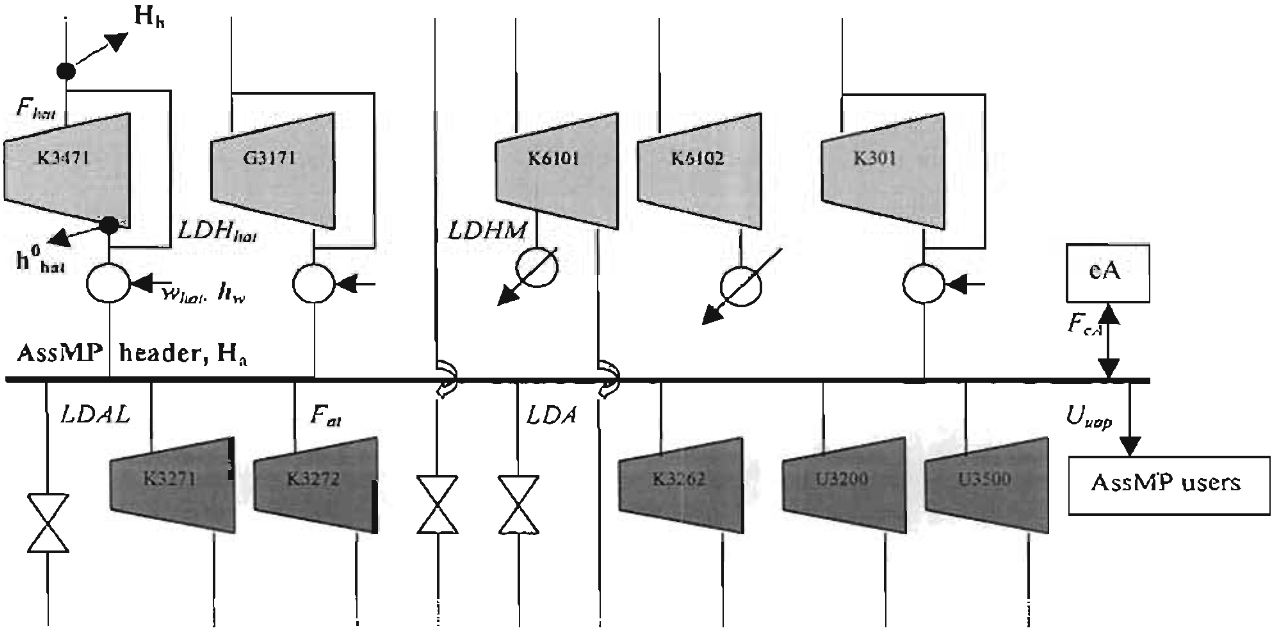


Figure 5.6: AMP header energy balance

5.5.2.1 Energy inflows

As seen in the mass balance, the only AMP steam produced is by the *hat* turbines: K3471, G3171 and K301. So to determine the power coming in we proceed to an energy balance just after the three desuperheaters: If we call h_{hat}^{0} the outlet enthalpy of the steam exhaust after turbine *hat*, we can write the power coming from the turbine *hat* as $(1-\varepsilon_{hat})*F_{hat}*h_{hat}^{0}$.

The steam coming from the *hat* bypass, assuming no loss across the line, has a mass enthalpy H_h as it comes from the HP header. So the power coming through the bypass is merely $LDH_{hat}*H_h$.

Now we have to add the contribution of the flow of water, w_{hat} sprayed into the desuperheater *hat*. Let us call h_w, the enthalpy of this water entering at 50°C, 1 bar (fixed average value for all runs). The power contribution is thus: $w_{hat}*h_w$. Having done this we can write the total energy inflow:

$$Power_{in} = \sum_{hat}\left\{(1-\varepsilon_{hat})*F_{hat}*h_{hat}^{0} + LDH_{hat}*H_h + w_{hat}*h_w\right\} \tag{5.44}$$

hat ∈ HAT

5.5.2.2 Energy outflows

The AMP steam required is for turbines, letdown stations and users so using the same way of modeling as used for the HP header and defining H_a as the mass enthalpy of the AMP header, we have:

Power of the steam required by turbines: $\left(\sum_{at}\left\{(1-\varepsilon_{at})*F_{at}\right\}\right)*H_a$ *at* ∈ AT

Power of the AMP steam coming down through the letdown stations: $(LDA + LDAL)*H_a$

With the same assumption on AMP steam users and the error term *eA* as on the HP header, which is that the power required for users and error terms need to account for the useful energy in its complete transition down to atmospheric condensate, we get the contribution of these two terms in the energy balance as follows:

$$\left(\left\{\sum_{uap}U_{uap}\right\} + F_{eA}\right)*(H_a - h_0) \qquad uap \in \text{UAP}$$

With h0 the enthalpy of water at 100°C and 1 bar: 400 kJ/kg.

So we have the energy outflow:

$$Power_{out} = \left(\sum_{at} \{(1-\varepsilon_{at}) * F_{at}\} \right) * H_a + (LDA + LDAL) * H_a + \left(\left\{ \sum_{uap} U_{uap} \right\} + F_{eA} \right) * (H_a - h_0)$$

$at \in$ AT, $uap \in$ UAP (5.45)

And finally the AMP steam energy balance is:

$$\boxed{\begin{aligned} &\sum_{hat} \{(1-\varepsilon_{hat}) * F_{hat} * h^0_{hat} + LDH_{hat} * H_h + w_{hat} * h_w\} = \\ &\left(\sum_{at} \{(1-\varepsilon_{at}) * F_{at}\} \right) * H_a + (LDA + LDAL) * H_a + \left(\left\{ \sum_{uap} U_{uap} \right\} + F_{eA} \right) * (H_a - h_0) \end{aligned}}$$

$hat \in$ HAT, $at \in$ AT, $uap \in$ UAP (5.46)

5.5.3 MP energy balance

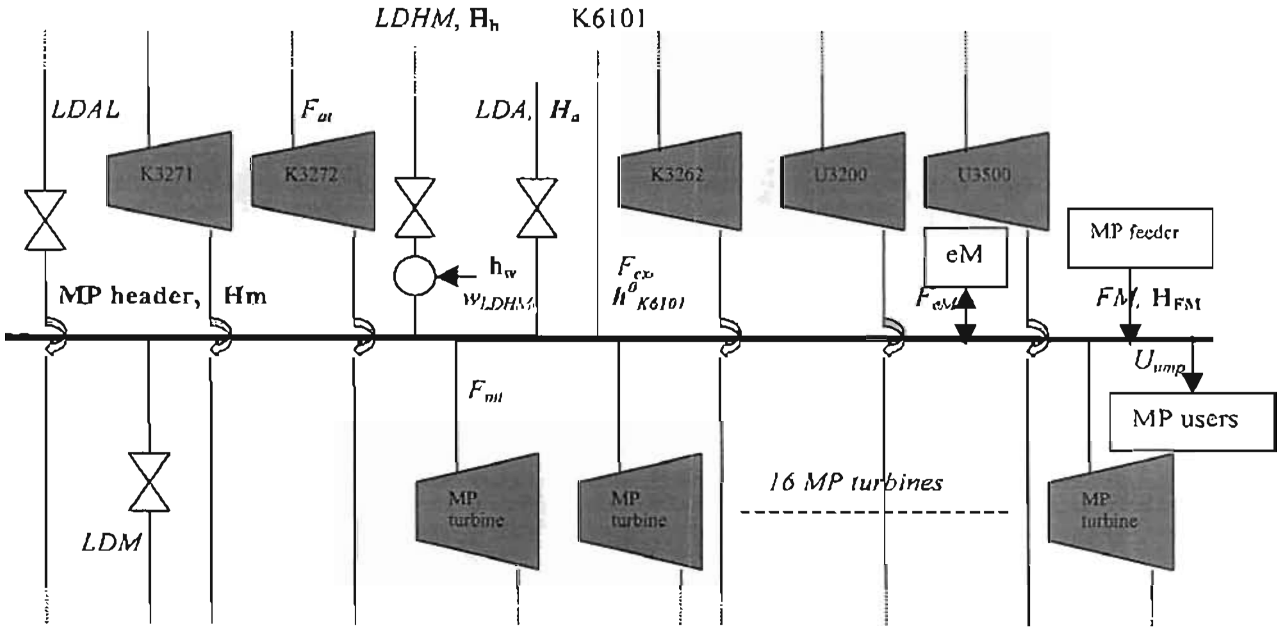


Figure 5.7: MP header energy balance

We are still using the mass balance and multiply flows of steam by their respective mass enthalpies to get the energy balance

5.5.3.1 Energy inflow

We have on the MP header a local steam feeder that produces *FM* tons/day of steam. If we call H_{FM} the mass enthalpy in kJ/tons of this feeder we have the following contribution to the energy inflow on the header: $FM * H_{FM}$

We have the exhaust steam coming from the K6101 turbine with a mass enthalpy h^0_{K6101}. From the mass balance, multiplying the steam flow rate by the mass enthalpy leads to the power term:

$$(1-\varepsilon_{K6101})*F_{K6101}*y_{K6101}*h^0_{K6101}$$

Then we have the letdown station coming from the AMP header, bringing the flow LDA with the enthalpy H_a. We have also the letdown station linking the HP header and the MP one. As there is a desuperheater here as well, the enthalpy contribution will be: $LDHM*H_h+w_{LDHM}*h_w$ with w_{LDHM} the flow of water (in tons/day) vaporized in the desuperheater.

So we have all the terms for the MP steam energy inflow:

$$Power_{in}=FM*H_{FM}+(1-\varepsilon_{K6101})*F_{K6101}*y_{K6101}*h^0_{K6101}+LDA*H_a+LDHM*H_h+w_{LDHM}*h_w \tag{5.47}$$

5.5.3.2 Energy outflow

We have on this level steam turbines, letdown stations and users that require MP steam with the following powers:

MP steam turbines: $\left(\sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\}\right)*H_m$ $\qquad mt\in \mathrm{MT}$

Letdown station: $LDM*H_m$

Users and error terms, as seen for previous headers, with the steam's useful energy in its complete transition down to atmospheric condensate: $\left(\left\{\sum_{ump}U_{ump}\right\}+F_{eM}\right)*(H_m-h_0)$ $\qquad ump\in \mathrm{UMP}$

where H_m is the mass enthalpy of the MP header and h0 the enthalpy of water at 100°C, 1 bar: 400 kJ/kg.

So we can write the total energy outflow as follows:

$$Power_{out}=\left(\sum_{mt}\{(1-\varepsilon_{mt})*F_{mt}\}\right)*H_m+LDM*H_m+\left(\left\{\sum_{ump}U_{ump}\right\}+F_{eM}\right)*(H_m-h_0)$$

$mt\in \mathrm{MT}$, $ump\in \mathrm{UMP}$

And finally the MP energy balance will be:

$$\boxed{\begin{aligned}&FM * H_{FM} + (1-\varepsilon_{K6101}) * F_{K6101} * y_{K6101} * h^0_{K6101} + LDA * H_a + LDHM * H_h + w_{LDHM} * h_w = \\ &\left(\sum_{mt}\{(1-\varepsilon_{mt}) * F_{mt}\}\right) * H_m + LDM * H_m + \left(\left\{\sum_{ump} U_{ump}\right\} + F_{eM}\right) * (H_m - h_0)\end{aligned}}$$

$mt \in$ MT, $ump \in$ UMP (5.48)

5.5.4 LP energy balance

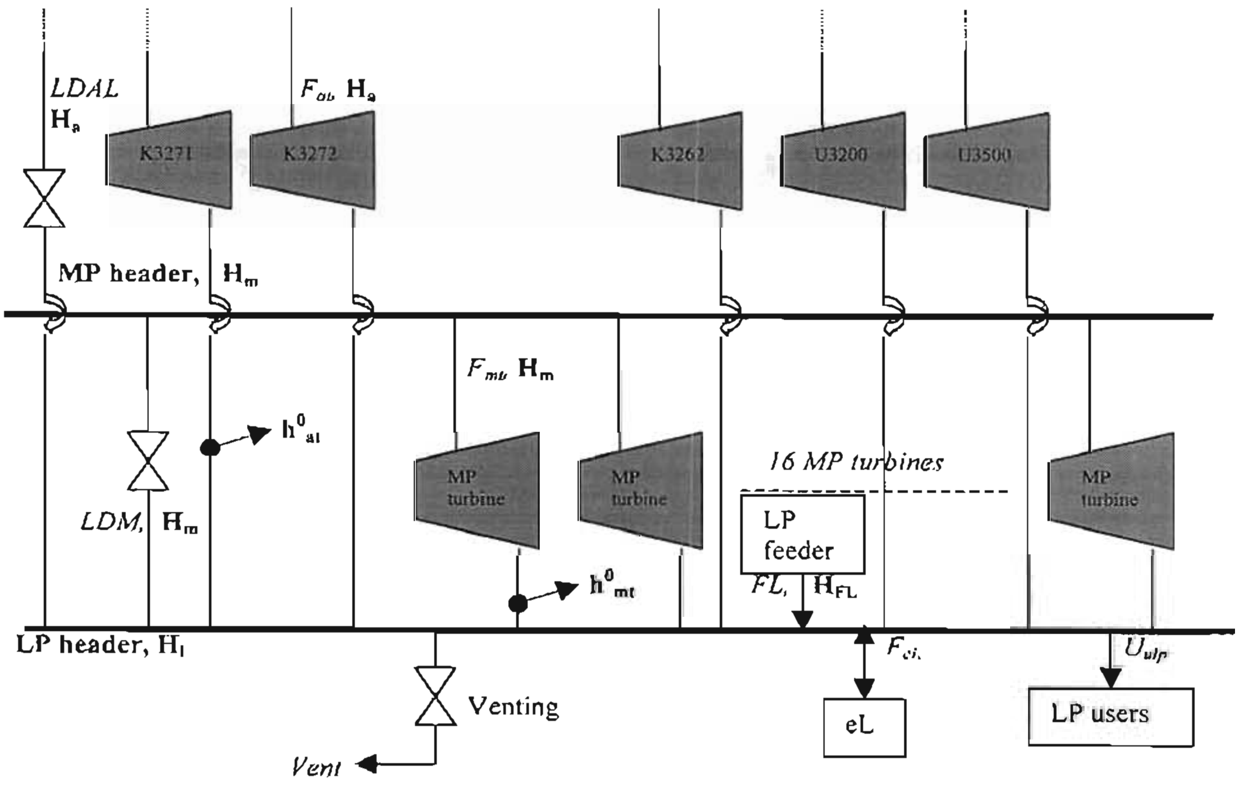


Figure 5.8: LP header energy balance

5.5.4.1 Energy inflow

We have the steam coming from the MP header, either through turbines or letdown stations, and from the AMP header either from turbines or letdown stations as well as a local feeder.

From local feeder:

We have on the LP header a local steam feeder that produces *FL* tons/day of steam. We call H_{FL} the mass enthalpy in kJ/tons of the LP steam produced here. We have its following contribution to the energy inflow on the header: $FL * H_{FL}$

From MP header

As seen before the contribution of the exhausted LP steam from any *mt* MP steam turbine is the flow $(1-\varepsilon_{mt}) * F_{mt}$ times the outlet enthalpy h^0_{mt} of this turbine. So it will appear as follows in the energy balance: $\sum_{mt} \left\{ (1-\varepsilon_{mt}) * F_{mt} * h^0_{mt} \right\}$ $\quad mt \in \mathrm{MT}$

As far as the letdown stations are concerned, we multiply the entire flow *LDM* by H_m the enthalpy of the MP header from where the steam comes: $LDM * H_m$

From AMP header

Powers of the exhausted LP steam from AMP turbines and of the steam coming through the AMP-LP letdown station will be expressed in the same way as above, as only the origin of the steam has changed for a similar structure.

So it will be for the exhausted LP steam, $\sum_{at} \left\{ (1-\varepsilon_{at}) * F_{at} * h^0_{at} \right\}$ $\quad at \in \mathrm{AT}$, with h^0_{at} the outlet steam enthalpy of the *at* turbine. Further, we have $LDAL * H_a$ for the letdown station.

So the total energy inflow is:

$$Power_{in} = FL * H_{FL} + \sum_{at} \left\{ (1-\varepsilon_{at}) * F_{at} * h^0_{at} \right\} + \sum_{mt} \left\{ (1-\varepsilon_{mt}) * F_{mt} * h^0_{mt} \right\} + LDM * H_m + LDAL * H_a$$

$$mt \in \mathrm{MT},\ at \in \mathrm{AT} \tag{5.49}$$

5.5.4.2 Energy outflow

On this bottom level we have only LP steam users and the error term *eL* to consider as well as the steam vented to atmosphere.

So we will have the term $\left(\left\{ \sum_{ulp} U_{ulp} \right\} + F_{eL} \right) * (H_l - h_0)$ $\quad uhp \in \mathrm{UHP}$

and $Vent * H_l$ as the contribution of the flow of vented steam *Vent* to the energy balance with H_l the mass enthalpy of the LP header. Still with h0 the enthalpy of water at 100°C and 1 bar: 400 kJ/kg.

So with these notations the total energy outflow is:

$$Power_{out} = Vent * H_t + \left(\left\{ \sum_{ulp} U_{ulp} \right\} + F_{eL} \right) * (H_t - h_0) \qquad uhp \in \mathrm{UHP} \qquad (5.50)$$

And the total LP energy balance is:

$$\boxed{\begin{aligned} & FL * H_{FL} + \sum_{at} \{(1-\varepsilon_{at}) * F_{at} * h_{at}^0\} + \sum_{mt} \{(1-\varepsilon_{mt}) * F_{mt} * h_{mt}^0\} + LDM * H_m + LDAL * H_n = \\ & Vent * H_t + \left(\left\{ \sum_{ulp} U_{ulp} \right\} + F_{eL} \right) * (H_t - h_0) \end{aligned}}$$

$$mt \in \mathrm{MT},\ at \in \mathrm{AT},\ uhp \in \mathrm{UHP} \qquad (5.51)$$

So in these last two sections we have modeled the steam distribution both with mass and energy balances to be as accurate as possible. These equations will be part of our MINLP model. We still have some constraints to add to this model to fully represent what physically happens.

5.6 Constraints

Indeed in our representation of the steam distribution we have to specify some equality or inequality constraints for both integer and continuous variables.

5.6.1 Constraints on power demand

5.6.1.1 Steam turbines

As far as steam turbines are concerned, the first demand to meet is obviously the power demand. We have to satisfy this demand either by the steam turbine if $\varepsilon = 0$ or with an electrical drive if $\varepsilon = 1$. We call W_n, the constant power demand for turbine n in kW. W_n is the current power production of turbine n calculated as follows:

$$W_n = F_n^0 * \Delta H_n \qquad (5.52)$$

with F_n^0 the current steam flow rate used by turbine n and ΔH_n the enthalpy drop across this turbine.

Since we calculated W_n as the flow multiplied by the enthalpy drop across the turbine, we have to define this enthalpy drop for all kinds of turbines. By investigating at the refinery we were able to find out these enthalpy drops from the Mollier diagram, knowing pressure and temperature at the inlet and outlet of turbines. So we have:

$$\boxed{\Delta H_n = H_t - h_n^0} \qquad (5.53)$$

with ΔH_n, the enthalpy drop across turbine n, H_i the enthalpy of the header from which the steam is used to drive the turbine and h_n^0 the steam outlet enthalpy of turbine n. Note that h_n^0 and H_i are continuous variables subject to changes depending on decision variables ε throughout the steam distribution system. However, ΔH_n is assumed constant as a turbine feature. By proceeding like this we do not need to mention the steam turbine efficiency and the theoretical enthalpy drop across each turbine. Indeed we are already taking this turbine efficiency into consideration because we calculate the power production by using the actual enthalpy drop across turbines, which is equal to the theoretical (or adiabatic) enthalpy drop times the turbine efficiency, from temperature and pressure measurements at the refinery. We have the following average figures for these measurements (note that they did not vary much with steam flow and other refinery conditions):

Averages values of March 2000			
Turbines	Enthalpy drop (kJ/kg)	Steam flows (Tons/day)	Power produced (kW)
K3471	146	447	756
G3171	127	597	878
K6101	167	1552	3000
K6102	600	355	2467
K301	200	2455	5683
AMP turbines	303	from 3 up to 275	from 11 up to 964
MP turbines	340	from 5 up to 833	from 20 up to 3278

Table 5.3: Averages turbine power productions of March 2000

Averages values of Dec 2000			
Turbines	Enthalpy drop (kJ/kg)	Steam flows (Tons/day)	Power produced (kW)
K3471	146	447	756
G3171	127	710	1044
K6101	167	1538	2973
K6102	600	395	2743
K301	200	2462	5699
AMP turbines	303	from 20 up to 275	from 70 up to 964
MP turbines	340	from 8 up to 805	from 31 up to 3168

Table 5.4: Averages turbine power productions of December 2000

Averages values of the 06 Dec 2000			
Turbines	Enthalpy drop (kJ/kg)	Steam flows (Tons/day)	Power produced (kW)
K3471	146	451	762
G3171	127	779	1145
K6101	167	1529	2955
K6102	600	382	2653
K301	200	2401	5558
AMP turbines	303	from 20 up to 275	from 70 up to 964
MP turbines	340	from 8 up to 843	from 31 up to 3317

Table 5.5: Averages turbine power productions of the 6th December 2000

Then we can define the constraint to meet the power demand by steam as:

$$\boxed{W_n * (1 - \varepsilon_n) = F_n * (H_i - h_n^0)} \quad (5.54)$$

with ε_n, F_n and h_n^0 variables relative to turbine n.

We can check with this formulation that if $\varepsilon_n = 0$ then the demand is satisfied by the steam turbine and if $\varepsilon_n = 1$ then the flow F_n is forced to zero (because ΔH_n is a constant) and the demand has to be met by an electrical drive.

5.6.1.2 Steam users

The power demand Wu_i for steam users on header i can be defined as:

$$\boxed{Wu_i = \left(\sum_i U_i\right) * (H_i - h_0)} \quad (5.55)$$

As *Hi* may change depending on the steam production and network configuration the flow of steam required can change as well (decreasing if *Hi* is increasing for instance). So the demand to meet is in terms of power and not only in terms of steam flow. Thus Wu_i will be the power demand to satisfy, and is calculated with the current conditions of enthalpy and flows as follows:

$$Wu_i = \left(\sum_i U_i^0\right) * (H_i^0 - h_0) \quad (5.56)$$

Where U_i^0 and H_i^0 are respectively user steam flows and header enthalpies in the current working conditions of the refinery when no optimization change has yet been implemented (i.e. existing time - varying loads).

5.6.1.3 Error terms

As mentioned earlier, error terms F_{el}, $l \in \{H, A, M, L\}$ are steam flows to balance the mass imbalance due to leaks or faulty measurements on user consumptions, so we will consider them in the same way as we consider steam users: i.e. as requiring a certain power. This means that if enthalpy is changing on header i the corresponding flow F_{el} will change as well to meet the following constant power requirement:

$$\boxed{Werr_l = F_{el} * (H_l - h_0)} \quad (5.57)$$

and we calculate $Werr_l$ using the current working conditions:

$$Werr_l = F_{el}^0 * (H_l^0 - h_0) \quad (5.58)$$

With F_{ci}^{0} the mass imbalance on header i calculated with the current network (i.e. existing time - varying loads).

5.6.2 Constraints on desuperheaters

There is no means of measuring the flow of water sent into the desuperheaters. Though the amount of water estimated does not have a significant impact on the mass balance, and could be neglected, we will represent as follows:

We assumed that the flow of water is a constant fraction X_i^w of the total steam flow passing through the desuperheater *i*. However each one of the four desuperheaters has its own particular fraction.

To determine X_i^w, we proceed to an enthalpy balance on each of the desuperheaters in the working conditions assuming no loss and water fully vaporized. Once again these energy balances will look like:

$$Inflows * H^{in} = Outflows * H^{out}$$

Desuperheaters located just after the hat turbines, K3471, G3171 and K301:

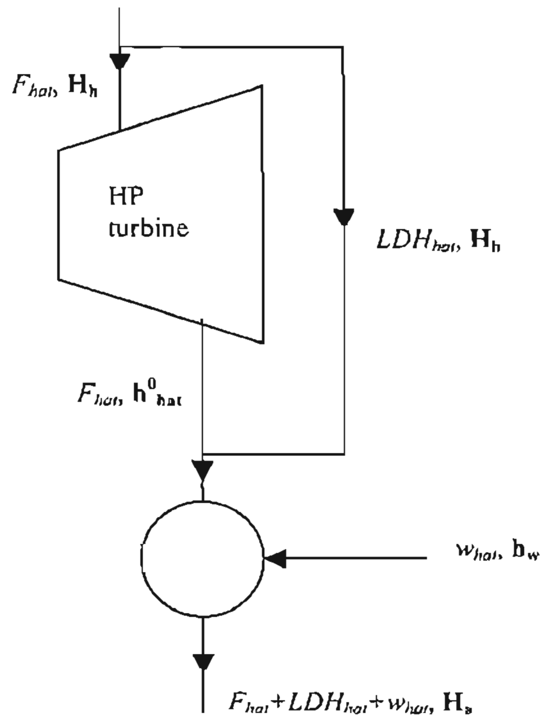


Figure 5.9: Desuperheaters energy balance

Energy balance:

$$w_{hat}^{0} * h_w + LDH_{hat}^{0} * H_h^{0} + F_{hat}^{0} * h_{hat}^{0} = \left(w_{hat}^{0} + LDH_{hat}^{0} + F_{hat}^{0} \right) * H_a^{0} \tag{5.59}$$

with h_w the enthalpy of water (50°C, 1 bar) known and X^0 meaning X in the current conditions with no turbine changes at all. So all the terms in this equation are known except w_{hat}^{0}.

Now we have by definition of X_{hat}^{w}:

$$w_{hat}^{0} = X_{hat}^{w} * \left(F_{hat}^{0} + LDH_{hat}^{0} \right) \tag{5.60}$$

Substituting w_{hat}^{0} from (5.60) in (5.59) leads to:

$$\begin{aligned} &\left[X_{hat}^{w} * \left(F_{hat}^{0} + LDH_{hat}^{0} \right) \right] * h_{w} + LDH_{hat}^{0} * H_{h}^{0} + F_{hat}^{0} * h_{hat}^{0} = \\ &\left(\left[X_{hat}^{w} * \left(F_{hat}^{0} + LDH_{hat}^{0} \right) \right] + LDH_{hat}^{0} + F_{hat}^{0} \right) * H_{a}^{0} \end{aligned} \tag{5.61}$$

And arranging (5.61) to get X_{hat}^{w} gives:

$$\boxed{X_{hat}^{w} = \frac{LDH_{hat}^{0} * \left(H_{h}^{0} - H_{a}^{0} \right) + F_{hat}^{0} * \left(h_{hat}^{0} - H_{a}^{0} \right)}{\left(F_{hat}^{0} + LDH_{hat}^{0} \right) * \left(H_{a}^{0} - h_{w} \right)}} \tag{5.62}$$

Using the same technique we calculate X_{LDHM}^{w} for the desuperheater located on the HP-MP letdown station.

$$\boxed{X_{LDHM}^{w} = \frac{\left(H_{h}^{0} - H_{m}^{0} \right)}{\left(H_{m}^{0} - h_{w} \right)}} \tag{5.63}$$

So using the figures we have from the overall mass balance done at the refinery we get the following results:

	Mar-00	Dec-00	06-Dec-00
XK3471 (%)	3.19	3.19	3.19
XG3171 (%)	4.81	4.32	4.66
XK301 (%)	1.3	1.3	1.3
XLDHM (%)	6.92	6.92	6.92

Table 5.6: Fraction of water, X_{i}^{w}, sent to desuperheaters

Note that X^{w}_{K3471} and X^{w}_{K301} are constant because there is no steam flow through the bypasses of these two turbines ($LDH^{0}_{hat}=0$) both in March and December 2000. We can define in our model the flow of water coming into the desuperheaters by:

$$\boxed{w_{hat} = X_{hat}^{w} * \left((1 - \varepsilon_{hat}) * F_{hat} + LDH_{hat} \right)} \qquad hat \in HAT \tag{5.64}$$

And,

$$\boxed{w_{LDHM} = X_{LDHM}^{w} * LDHM} \tag{5.65}$$

5.7 Cost functions

Now that the modeling of the steam distribution has been completed, we arrive at a significant field which is the costs involved in the optimization. Indeed the purpose of our project is to determine whether and which steam turbines should be switched to electrical drives. It will be necessary to take into account the different costs involved in replacing steam turbines, running electrical drives and maintaining both steam turbines and electrical drives.

5.7.1 High pressure steam production costs

The first cost to look at is obviously the cost of producing HP steam with the refinery boilers. FH, the flow of HP steam produced, will ultimately depend on the optimum distribution found and the corresponding steam demand. We must bear in mind that the problem of venting LP steam does not necessarily come from an excess of HP steam produced but more from a bad management of steam use in the distribution system as a whole, like a lack of LP steam users for the LP steam produced leading to venting.
In considering the option of removing steam turbines we expect this HP steam production to be lower than the current one as the demand would decrease with turbines removed. However, it does not necessarily mean that we will not have any LP steam vented for the optimal solution, as costs involved in replacing steam turbines with electrical drives might be too high and LP steam production might stay higher than the LP steam demand.

The cost of producing one ton per day of HP steam is calculated as follows:

$$\left.\begin{array}{l} 1\text{t HP steam} = 1.16\text{t SRS} \\ 1\text{t SRF} = 15.5\text{t SRS} \\ 1\text{t fuelgas} = 1.25\text{t SRF} \\ 1\text{t fuelgas} = \$225 \\ 1\$ = \text{R}8.36 \end{array}\right\} \Rightarrow 1\text{t HP steam} = \frac{(225*8.36*1.16)}{(15.5*1.25)} = \text{R}112.6$$

where SRS (Standard Refinery Steam) and SRS (Standard Refinery Fuel) are standardised energy measurements at SAPREF. In this way, the cost depends on the conversion of US$/Rands. At this time we proceed on the basis of 1US$ = R8.36 (24/08/2001) so we use a figure of R112.6 per ton of HP steam. As the conversion rate is subject to change we shall have to study the sensitivity to this cost of our optimization work.

5.7.2 Replacing steam turbines costs

We investigated at the refinery and spoke with staff and engineers to assess all of the work involved and equipment to buy for removing a steam turbine on site and putting in an electrical drive instead. This operation is a significant one and obviously involves some major costs.

The first step is to remove the existing turbine and demolish its old bed. Then we have to build a new bed for the new motor and get switchgear. Then we need to buy the electric motor itself and install it. We have to get some electrical cables as well as a VSD (Variated Speed Drive). Finally we consider the manpower involved in all of these operations from the work of artisans and technicians to engineers. At this stage of our investigations we met a major problem after having spoken with electrical engineers: The refinery electrical infrastructure would not support such substantial electrical demand increases due to the installation of electrical drives. So we had to consider costs involved in additional site infrastructure due to existing bottlenecks in order to get a structure able to supply the additional electrical demand.

We managed to get some figures for all these costs from staff and engineers. These figures are characteristic costs for specific sizes of electrical drive, so depend on drive power demand.
We used them as reference points and assuming linear correlation (except for motor cost) we proceed to a linear regression to get costs for a larger range of size. Indeed for the motor cost correlation we assumed a second order correlation to approximate costs vs. size:
Motor cost $= a*Power^2 + b*Power + c$
An example is given below for removal and demolition costs:
The two reference points we get are: Removal and bed demolition of a 1MW: R4000 and for a 6MW: R8000. We assume that

$$\text{Cost (R)} = A*\text{Power (MW)} + B \qquad (5.66)$$

We can easily get A and B by substituting our two reference points in (5.66):

$$\begin{cases} 4000 = A*1 + B \\ 8000 = A*6 + B \end{cases} \Leftrightarrow \begin{cases} A = 800R/MW \\ B = 3200R \end{cases}$$

Appendix B shows the complete calculation of costs involved for every work to do and equipment to buy in the replacement of a steam turbine with a variable speed electrical drive.

We expected a second order correlation of the capital cost for replacement because of the second order relationship between the cost of motors and power:

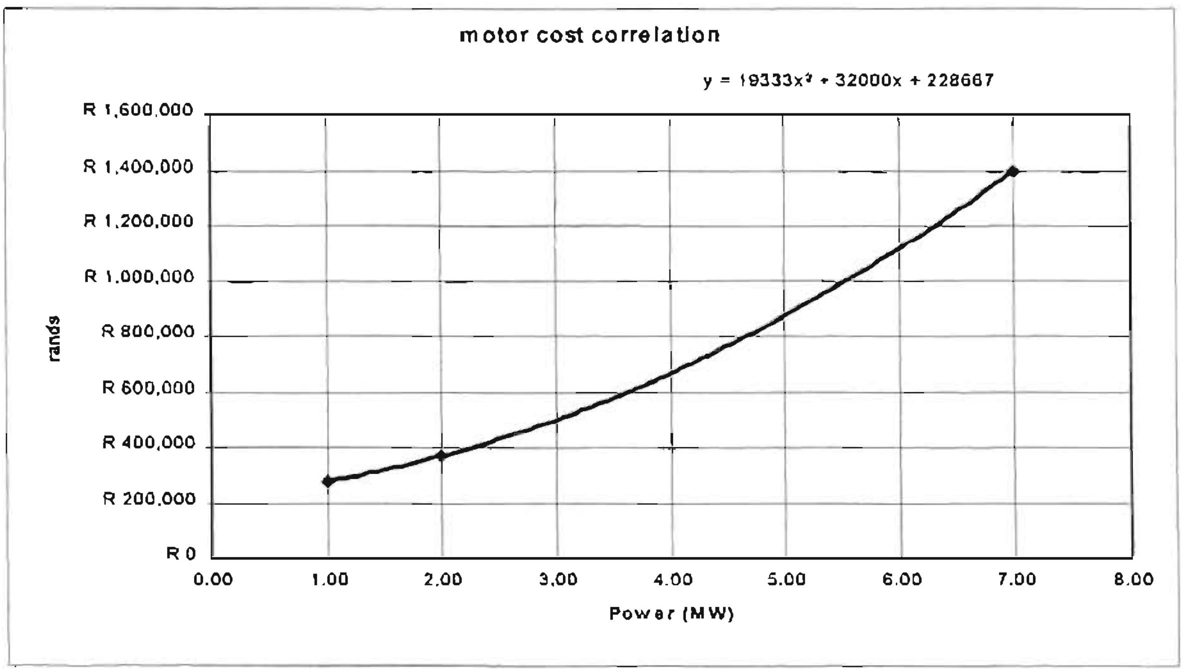


Figure 5.10: motor cost correlation

However the additional site infrastructure cost, that represents approximately 46% of the overall capital cost, causes the relationship to be nearly linear:

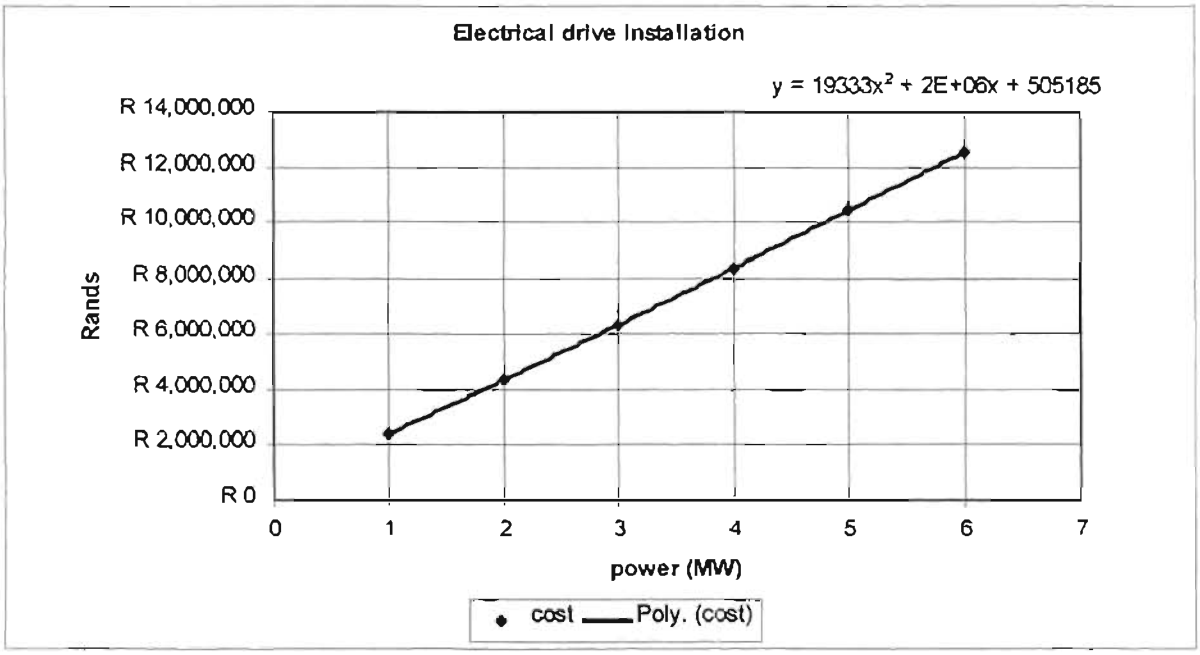


Figure 5.11: Overall electrical drive installation cost

Thus the limitation due to bottlenecks in the electrical infrastructure represents a large drawback for electrical drives installation and penalises heavily in the optimization. Eventually we have:

$$\boxed{CRp_i = 0.019333 * We_i^2 + 2000 * We_i + 505185} \tag{5.67}$$

With CRp_i the capital cost in rands for replacement of steam turbine i, producing W_i kW, with an electrical drive of capacity We_i kW. Indeed we want the electrical drive to satisfy exactly the same power demand as the steam turbine so taking into account the electrical efficiency of such electrical drives we have:

$$\boxed{W_i = \eta_{elec} * We_i} \tag{5.68}$$

From electrical engineers at SAPREF, the electrical efficiency is 95%, so we will assume for all our work below that η_{elec}=0.95.

5.7.3 Maintenance costs

In order to compare the two options we have either steam turbines or electrical drives, and we must evaluate the costs of maintaining these utilities under working conditions.

5.7.3.1 Maintenance of steam turbines

These turbines do not require a lot of attention and normally have a long lifetime around 30 years. Mechanical engineers in charge of the utility section gave us the cost involved per year depending on the size (meaning the power) of turbines. We have the following figures:

Maintenance of steam turbines		
turbines	design power (kW)	cost (R/yr)
small	100	1700
K6101	2700	40000
G3171	5000	75000

Table 5.7 Maintenance costs of steam turbines

From these data we get the following chart:

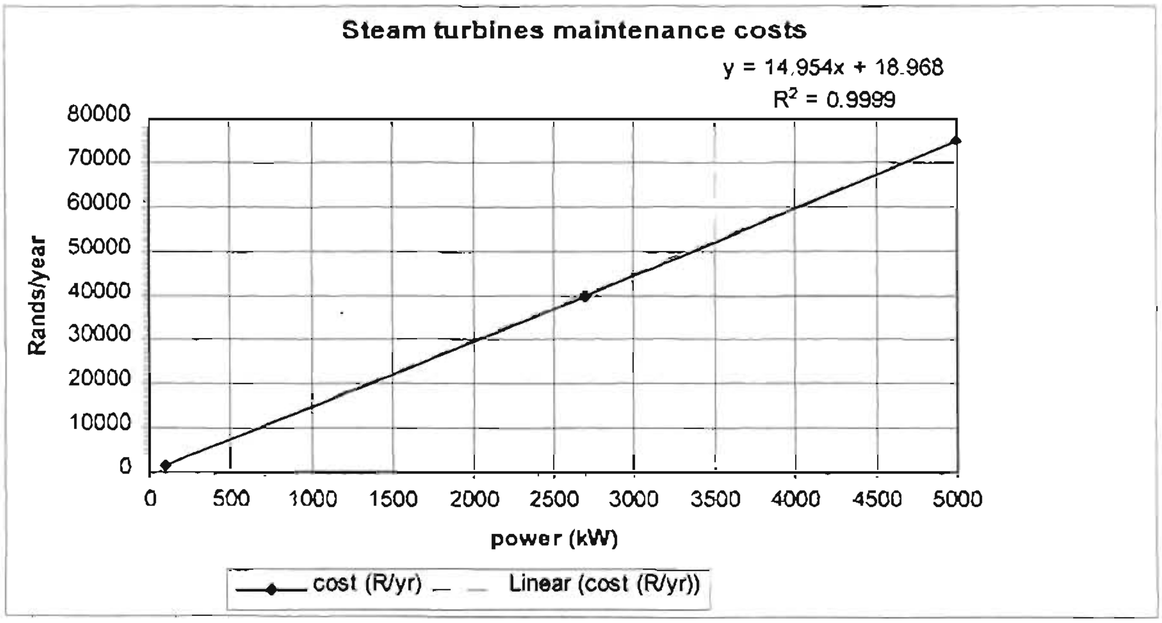


Figure 5.12: Steam turbines maintenance costs correlation

We can approximate with a very good precision the relationship maintenance vs. power as linear and as follows:

$$\boxed{Mst_i = 14.954 * W_i + 18.968} \tag{5.69}$$

giving Mst_i the annual cost in R/year of maintaining turbine i, producing W_i kW.

5.7.3.2 Maintenance of electrical drives

Considering the option of replacing steam turbines with electrical drives we must think as well of the costs involved to run and maintain these electrical drives.

Electrical engineers looking at the maintenance costs of the existing electrical installations at the refinery gave us an average cost per kW and per year: They spend on average R285 per kW per year for maintenance of electrical drives. We can see that it is much more expensive than the maintenance of steam turbines so this cost will not favour the "electrical" choice.

We merely use the following equation in our model:

$$\boxed{Melec_i = 285 * We_i} \tag{5.70}$$

Where $Melec_i$ is the annual cost in R/year of maintaining electrical drive i (in place of turbine i) of size We_i kW.

5.7.4 Running electrical drives costs

The other cost involved in choosing an electrical drive to supply the power demand is obviously the cost of electricity required for the use of such a drive, i.e. as paid by the refinery to the electricity supplier, ESKOM.

So the cost of running an electrical drive of size We_i kW, 24 hours per day for one year, $Celec_i$, is:

$$\boxed{Celec_i = 24 * 365 * a \ * We_i} \tag{5.71}$$

with "a" the cost of one kWh, currently equal to 0.174R.

5.7.5 Salvage value of steam turbines

If one or more steam turbines are switched off and removed, there is a financial benefit in selling it as it should still be in working condition. Mechanical engineers suggest that 10% of the capital cost of the turbine is a good approximation of this value.

Thus we first have to determine the capital cost of steam turbines and then take 10% of this cost.
We get the following reference points from engineers in charge of utilities:

Capital cost of steam turbines		
turbines	design power (kW)	capital cost (Rands)
K6101	2700	R 15,000,000
small	100	R 250,000

Table 5.8: Capital costs of steam turbines

The correlation is known to be of the following form:

$$Cap = a_0 * Wi^b \tag{5.72}$$

with Cap the capital cost of a steam turbine of capacity Wi kW.

We can get a_0 and b with the two reference points we have. It would have been better to use more than two points to get a more accurate correlation but with these two points the approximation is accurate enough for our purpose.

So using the two points in (5.72) gives:

$$\begin{cases} 15000000 = a_0 * 2700^b \\ 250000 = a_0 * 100^b \end{cases} \Rightarrow \left(\frac{15000000}{250000}\right) = \left(\frac{2700}{100}\right)^b \Rightarrow \begin{cases} b = \dfrac{\log(60)}{\log(27)} = 1.242278 \\ a_0 = \dfrac{250000}{100^{1.242278}} = 819.19 \end{cases}$$

So we have: $$\boxed{Cap = 819.2 * Wi^{1.2423}} \tag{5.73}$$

Graphically it gives:

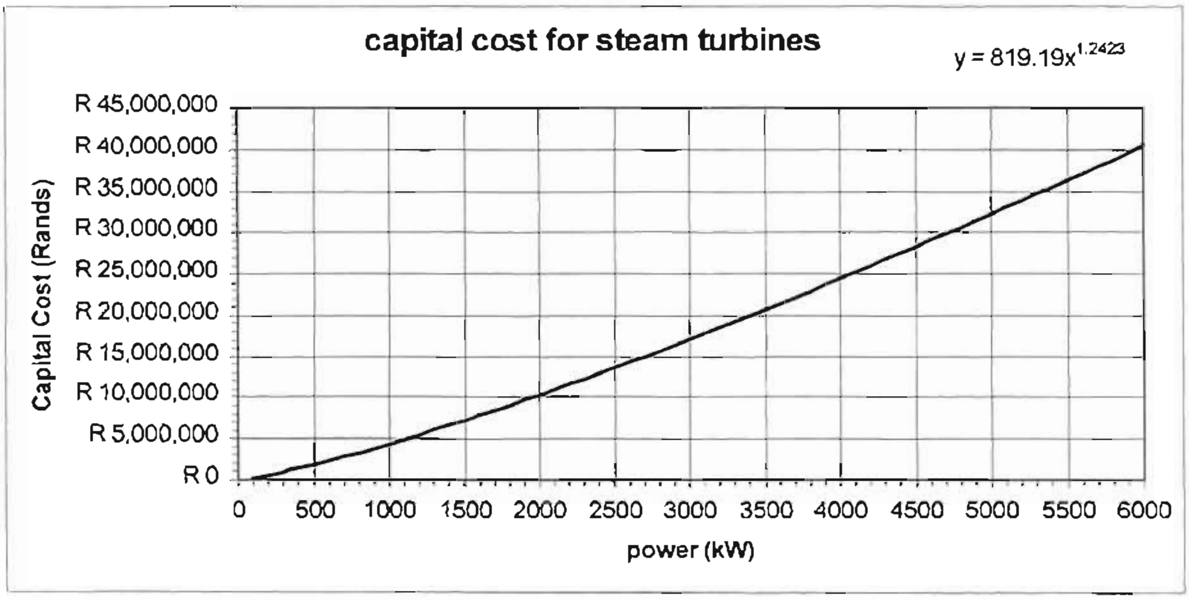


Figure 5.13: Capital costs for steam turbines correlation

So we take 10% of this capital cost as the salvage value:

$$\boxed{Vs_i = 0.1 * (819.2 * Wi^{1.2423}) = 81.92 * Wi^{1.2423}} \tag{5.74}$$

with Vs_i the salvage value (or net realisable value) in Rands for turbine i of size Wi kW.

5.8 Objective function to minimise

We now have the full model of mass and energy balances as well as a full representation of costs involved in our optimization problem.

The remaining model development for the MINLP problem is concerned with defining the objective function to minimise z = z (x, y) with x integer variables and y continuous variables.

We have defined above all the cost functions involved. The objective is to minimise the sum of all these costs put together so Z is in Rands per year.

5.8.1 Cost of HP steam

This term $Z_{HPsteam}$ will appear in Z as the cost of producing HP steam for one year:

$$Z_{HPsteam} = 365 * FH * C_{hp\_steam} \tag{5.75}$$

with C_{hp_steam}, the cost of producing 1 ton per day of HP steam, equal to R112.6 under our conditions.

5.8.2 Cost of maintaining steam turbines in use

This term Z_{steam} is given by:

$$Z_{steam} = \sum_{i=1}^{N} [(1 - \varepsilon_i) * Mst_i] \tag{5.76}$$

with N the number of steam turbines considered for a change. We can see that if turbine i is kept on steam use (ε_i =0) then we have to take into account the maintenance costs involved, otherwise if the turbine is switched off (ε_i =1) then we save these maintenance costs.

5.8.3 Cost of maintaining and running electrical drives

This cost, Z_{elec}, is only considered if we decide to replace one or more steam turbines, i.e. if $\varepsilon_i = 1$. Thus we have:

$$Z_{elec} = \sum_{i=1}^{N} [\varepsilon_i * (Celec_i + Melec_i)] \tag{5.77}$$

5.8.4 Replacement cost

As we are working with functions in Rands per year we have to define a project lifetime over which replacement cost will be spread. Thus calling D this project lifetime, we shall pay each year (assuming the same amount each year) the following cost Z_{repl} for those turbines, which are replaced:

$$Z_{repl} = \sum_{i=1}^{N} \varepsilon_i * \frac{CRp_i}{D} \tag{5.78}$$

5.8.5 Salvage value

Finally if we removed turbine i we can save its salvage value over the same period of time D. So the money saved each year will be Z_{salv}:

$$Z_{salv} = \sum_{i=1}^{N} \varepsilon_i * \frac{Vs_i}{D} \tag{5.79}$$

Now we can write Z the objective function in Rands/year to be minimised, in our MINLP problem, as follows:

$$Z = Z_{HPsteam} + Z_{steam} + Z_{elec} + Z_{repl} - Z_{salv} \tag{5.80}$$

Then, using equations (5.75) up to (5.79), we have:

$$\boxed{Z = 365 * FH * C_{hp\_steam} + \sum_{i=1}^{N}[(1-\varepsilon_i) * Mst_i] + \sum_{i=1}^{N}[\varepsilon_i * (Celec_i + Melec_i)] + \sum_{i=1}^{N}\varepsilon_i * \frac{CRp_i}{D} - \sum_{i=1}^{N}\varepsilon_i * \frac{Vs_i}{D}} \tag{5.81}$$

And replacing cost functions as defined earlier we get:

$$\boxed{\begin{aligned} Z = {} & 365 * FH * C_{hp\_steam} + \sum_{i=1}^{N}(1-\varepsilon_i) * (14.954 * W_i + 18.968) + \\ & \sum_{i=1}^{N}\varepsilon_i * \left(24 * 365 * a * \left(\frac{Wi}{\eta_{elec}}\right) + 285 * \left(\frac{Wi}{\eta_{elec}}\right) + \frac{0.019333 * \left(\frac{Wi}{\eta_{elec}}\right)^2 + 2000 * \left(\frac{Wi}{\eta_{elec}}\right) + 505185}{D} - \frac{81.92 * Wi^{1.2423}}{D} \right) \end{aligned}} \tag{5.82}$$

In this chapter we built a mathematical model of the distribution, both on a physical view by mass and energy balances and on an financial view by cost functions. These equations and constraints are the MINLP problem that we will solve by using optimization techniques and algorithms available in the computer package GAMS.

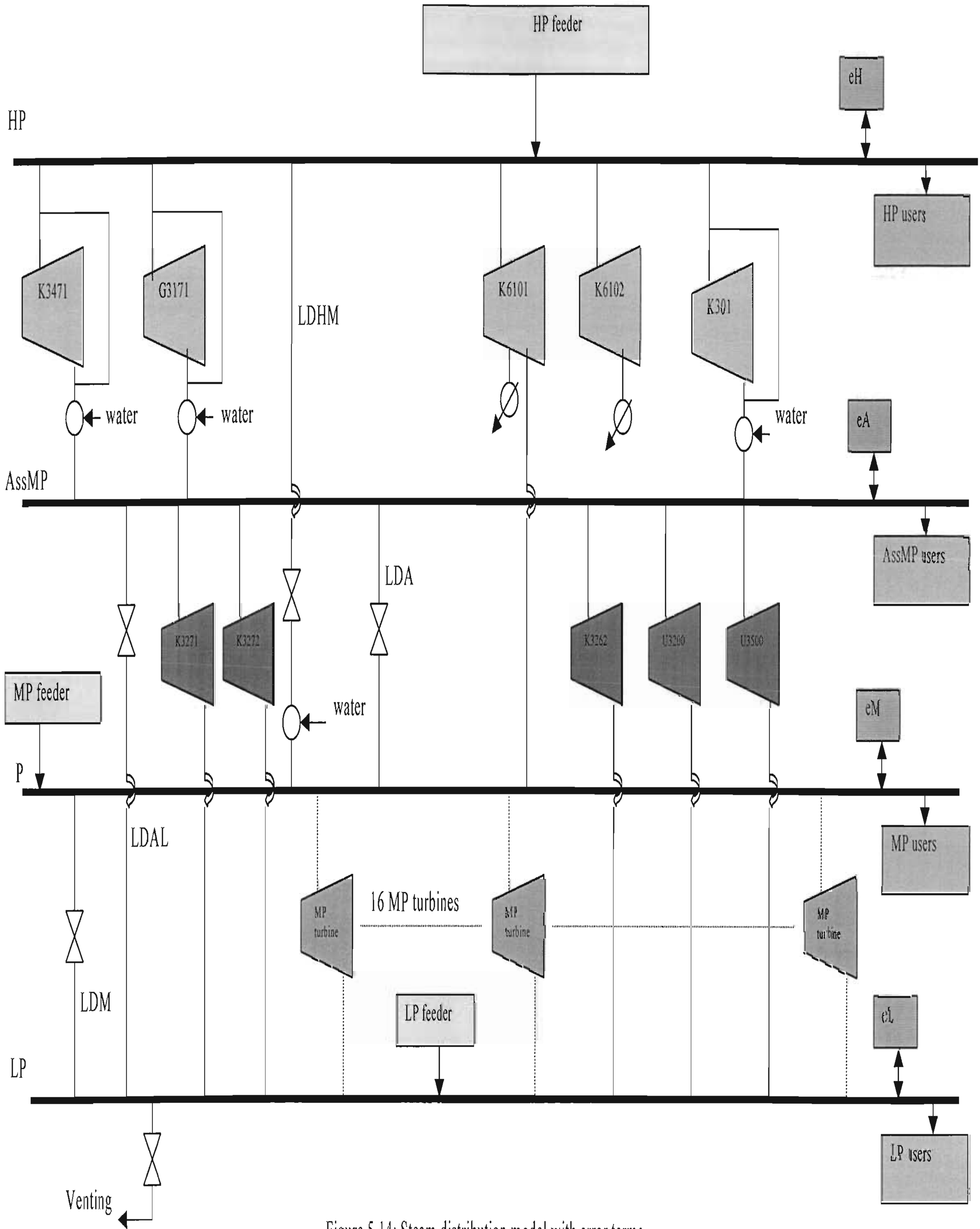


Figure 5.14: Steam distribution model with error terms

Chapter 6

GAMS programming

Now that we have our complete model of the distribution and the MINLP problem formulated, we can work on the optimization itself. In this chapter we introduce the use of the optimization package GAMS (General algebraic modeling system), first on a simplified example and then on our problem. Note that some of the work presented here is based on the GAMS manual as well as on the solvers manual, both from the GAMS Development Corporation (1996).

6.1 Presentation of GAMS

GAMS is a commercial programming language that provides a flexible framework for formulating and solving linear, nonlinear, integer and mixed integer (linear or nonlinear) optimization problems.
Basic features of GAMS are:

- All existing algorithmic methods described previously are available or can be added without changing the user's model representation.
- The optimization problem can be formulated independently of the data it uses. This separation of logic and data allows a problem to be increased in size without causing an increase in the complexity of the representation. It also allows solving the same model for different data sets without having to change the problem formulation.
- Great effort has been invested in making the language as accessible as possible and in formulating the program as "naturally" as possible. The user does not have to worry about details such as array sizes and work space allowed even for large and complex models. All these efforts make it a user-friendly language.

The structure of a GAMS program is as follows:

Input:

- Sets
  - Declaration
  - Assignment of members
- Data (Parameters, Tables, Scalars)
  - Declaration
  - Assignment of values
- Variables
  - Declaration
  - Assignment of type

- Assignment of bounds and/or initial values (optional)
- Equations
  - Declaration
  - Definition
- Model and Solve statements
- Display statement (optional)

Output:

- Echo Print
- Reference Maps
- Equation Listings
- Status Reports
- Results

The syntax of the program looks similar to the typical formulation of optimization problems, making it easier to work with.

6.2 Example with a simplified distribution

We will see in a simplified optimization problem how a typical GAMS program looks and what sort of results it gives. Here is a simplified utility section that we have to optimize:

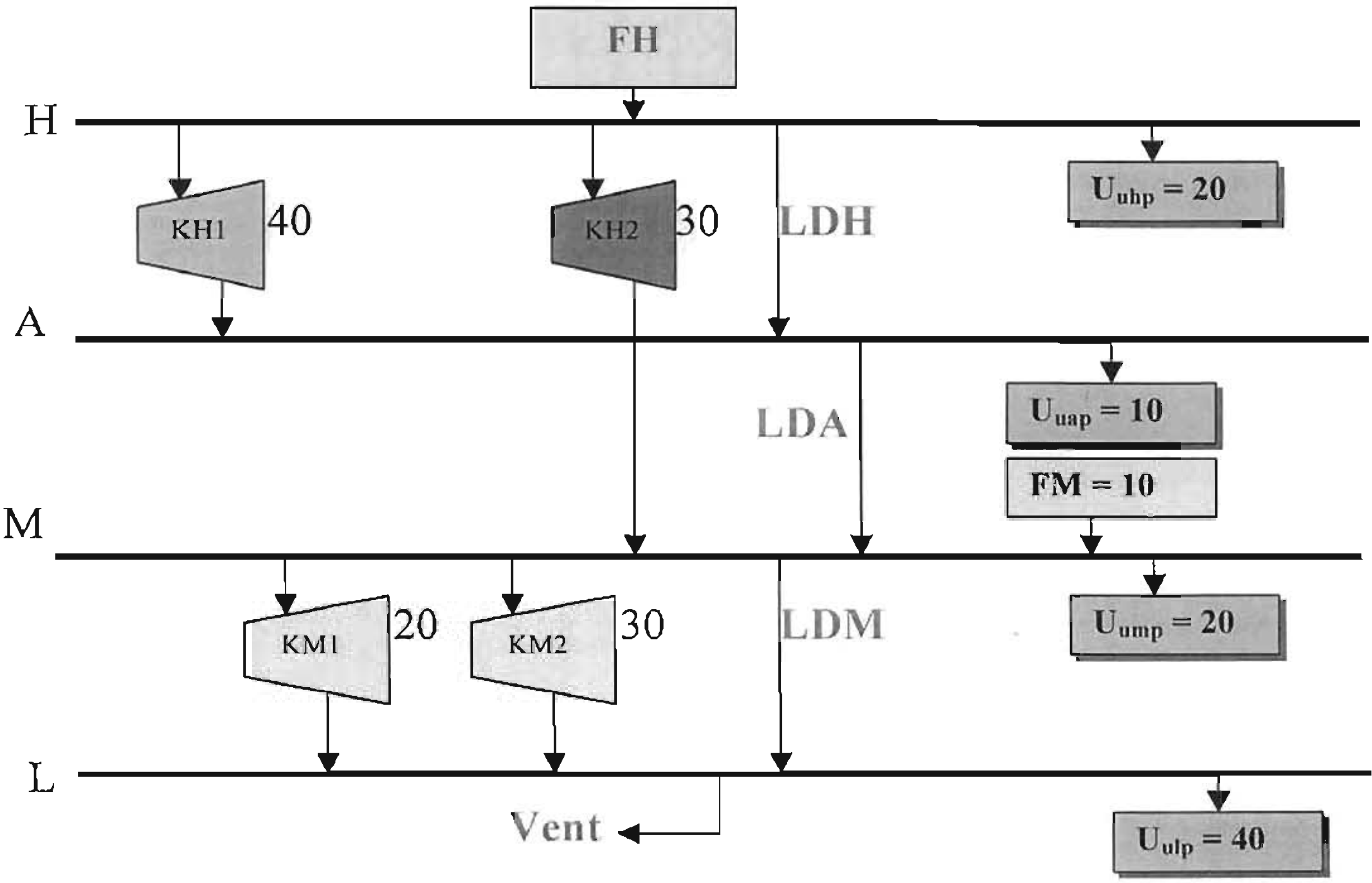


Figure 6.1: Simplified utility plant flowsheet

Key:

FH, FM: Steam producers

KHi, KMi: name of turbine number i using steam from header H or M.

6.2.1 Hypotheses

- Still a linear problem
- No turbine bypass
- No error terms
- Only back-pressure turbines
- Only steam flow requirement (no energy balances)
- Cost of replacing steam turbines proportional to the original steam consumption F_{KHi} or F_{KMi}. (Independent of the header and of the particular turbine considered).

6.2.2 Variables

- 5 Continuous: FH, 3 letdowns and the vented steam
- 4 Integers: ε_{KHi} and ε_{KMi} for every turbine

6.2.3 Equations

4 Mass balances on steam for each header as seen in the previous chapter.

1 constraint on FH: FH ≥ 0

4 constraints on the letdowns and vented steam: LDi ≥ 0 and Vent ≥ 0

Mass balances:

$$\begin{cases} FH = (1-\varepsilon_{KH1})*F_{KH1} + (1-\varepsilon_{KH2})*F_{KH2} + LDH + U_{uhp} \\ (1-\varepsilon_{KH1})*F_{KH1} + LDH = LDA + U_{uop} \\ FM + (1-\varepsilon_{KH2})*F_{KH2} + LDA = (1-\varepsilon_{KM1})*F_{KM1} + (1-\varepsilon_{KM2})*F_{KM2} + U_{ump} + LDM \\ (1-\varepsilon_{KM1})*F_{KM1} + (1-\varepsilon_{KM2})*F_{KM2} + LDM = U_{ulp} + Vent \end{cases} \tag{6.1}$$

6.2.4 Cost of replacement

In order to simplify the model we choose to express the cost of replacing steam turbines as proportional to the original steam consumption. E.g.: If turbine KH1, using F_{KH1} tons/day, is switched off the overall price of replacing it with an electrical drive and operating it, is: $\alpha * F_{KH1}$, where the constant α (R/ton) includes the depreciation of the capital investment.

So the cost of replacement, CR (R/day), for all the turbines will appear as:

$$CR = \left[\left(\varepsilon_{KH1} * F_{KH1} + \varepsilon_{KH2} * F_{KH2}\right) + \left(\varepsilon_{KM1} * F_{KM1} + \varepsilon_{KM2} * F_{KM2}\right)\right] * \alpha \tag{6.2}$$

6.2.5 Objective function Z

The objective function in R/day to minimize here will be the cost of the HP steam production plus the cost of replacement CR:

$$Z = FH * C_{HPsteam} + CR \tag{6.3}$$

with $C_{HPsteam}$ the HP steam production cost (R/ton).

So using (6.1), we get:

$$\boxed{Z = FH * C_{HPsteam} + \left[\left(\varepsilon_{KH1} * F_{KH1} + \varepsilon_{KH2} * F_{KH2}\right) + \left(\varepsilon_{KM1} * F_{KM1} + \varepsilon_{KM2} * F_{KM2}\right)\right] * \alpha} \tag{6.4}$$

6.2.6 MILP problem

Thus our problem is a mixed integer linear programming problem (P0) that has the following form:

$$\min Z(FH, \varepsilon) = FH * C_{HPsteam} + \left[\left(\varepsilon_{KH1} * F_{KH1} + \varepsilon_{KH2} * F_{KH2}\right) + \left(\varepsilon_{KM1} * F_{KM1} + \varepsilon_{KM2} * F_{KM2}\right)\right] * \alpha$$

Subject to:

(P0)

$$\begin{cases} FH = (1 - \varepsilon_{KH1}) * F_{KH1} + (1 - \varepsilon_{KH2}) * F_{KH2} + LDH + U_{uhp} \\ (1 - \varepsilon_{KH1}) * F_{KH1} + LDH = LDA + U_{uap} \\ FM + (1 - \varepsilon_{KH2}) * F_{KH2} + LDA = (1 - \varepsilon_{KM1}) * F_{KM1} + (1 - \varepsilon_{KM2}) * F_{KM2} + U_{ump} + LDM \\ (1 - \varepsilon_{KM1}) * F_{KM1} + (1 - \varepsilon_{KM2}) * F_{KM2} + LDM = U_{ulp} + Vent \end{cases}$$

$$\begin{cases} FH \geq 0 \\ LDH \geq 0 \\ LDA \geq 0 \\ LDM \geq 0 \\ Vent \geq 0 \end{cases}$$

6.2.7 GAMS program

We are now solving problem (P0) by writing the following GAMS program:
Note that lines beginning with a "*" are comments.

```
$title STEAM DISTRIBUTION
```

```
Sets
     i steam turbines / 1,2 /
     j header / H,A,M,L /;

* Sets are corresponding exactly to the indices in the algebraic representations of models.

**************************************************************************************

Parameters
       U(j) steam users on header j
          / H 20
            A 10
            M 20
            L 40 /

       FD(j) local steam feeder on header j
          / M 10 /;

Table F(i,j) steam from header j used by turbine i
            H       A       M      L
       1   40               20
       2   30               30        ;

Scalars alpha replacing cost coefficient /20/
     Csteam HP steam production cost /100/;

* Data can be entered with the three fundamentally different formats above:
* Lists, tables and direct assignments.

**************************************************************************************

Variables
       z objective function
       epsilonK(i,j) decision for turbine i on header j
       FH steam production on header H
       LD(j) letdown from the j header
       Vent LP steam vented to atmosphere;

Binary variable epsilonK ;

* The decision variables of a GAMS-expressed model must be declared with a Variables statement.

**************************************************************************************

Equations

      cost objective function to minimize
      balanceH steam balance for level H
      balanceA steam balance for level A
      balanceM steam balance for level M
      balanceL steam balance for level L
      constraintFH constraint on steam production on level H
      constraintLD(j) constraint on letdown j
      constraintVent constraint on venting ;

cost..     z =e=  FH*Csteam + alpha*sum(j, sum(i, epsilonK(i,j)*F(i,j))) ;

balanceH..  FH -LD('H')- sum(i,(1-epsilonK(i,'H'))*F(i,'H')) - U('H') =e= 0;
```

balanceA.. (1-epsilonK('1','H'))*F('1','H') + LD('H') - LD('A') - U('A') =e= 0;

balanceM.. FD('M') + (1-epsilonK('2','H'))*F('2','H')+ LD('A') - sum(i,(1-epsilonK(i,'M'))*F(i,'M')) - LD('M') - U('M') =e= 0;

balanceL.. sum(i,(1-epsilonK(i,'M'))*F(i,'M')) + LD('M') - U('L') - Vent =e= 0;

constraintFH.. FH =g= 0;

constraintLD(j).. LD(j) =g= 0;

constraintVent.. Vent =g= 0;

* *Equations, inequalities and the objective function must be declared and defined as above.*
* *"=e=" and "=g=" respectively mean "equal to" and "greater than".*
* *The power of such a language is whenever a group of equations or inequalities has the same*
* *algebraic structure, all the members of the group are created simultaneously, not individually.*

****

Model distribution /all/;

* *The model must be named and the word "all" means that all previously defined equations are*
* *to be included.*

Solve distribution using mip minimizing z ;

* *To use the solution procedures available, the model has to be called following by the type*
* *of the problem and the direction to follow for the objective function.*

Display epsilonK.l , FH.l, LD.l, Vent.L;

* *Optimal values of variables are displayed.*

****

We can see in this example how the syntax of the program appears similar to the typical formulation of optimization problems. We just have to properly define the model and then call a solver, depending on the kind of problem, that deals with the mathematical resolution itself. The Display section is not compulsory, as the solver output will show all of the variable optimal values and several other things.

6.2.8 GAMS output

We present below the two most interesting parts of the GAMS output:

S O L V E S U M M A R Y

MODEL distribution	*OBJECTIVE z*
TYPE MIP	*DIRECTION MINIMIZE*
SOLVER OSL	*FROM LINE 88*
**** *SOLVER STATUS*	*1 NORMAL COMPLETION*
**** *MODEL STATUS*	*8 INTEGER SOLUTION*

***** OBJECTIVE VALUE 9200.0000*

RESOURCE USAGE, LIMIT 0.223 1000.000
ITERATION COUNT, LIMIT 6 10000

And:

---- 93 VARIABLE epsilonK.L decision for turbine i on header j

	H	M
1	1.000	1.000

---- 93 VARIABLE FH.L = 80.000 steam production on header H

---- 93 VARIABLE LD.L letdown from the j header
H 30.000, A 20.000, M 10.000

---- 93 VARIABLE Vent.L = 0.000 LP steam vented to atmosphere

So the minimum for the objective function is Z = 9200 R corresponding to two turbines that have to be removed: KH1 and KM1. The solver used is OSL that offers several algorithms for solving LP problems. Here we used a primal simplex method (the default one). These results are highly dependent on the values we set for the example as steam user requirements and the coefficient α for turbine removal and electricity costs.

As we chose the cost of replacing steam turbines and running electrical drives (α =20) lower than the steam production cost (Csteam = 100), it is actually less expensive to switch turbines off than using them. So the overall idea behind the solution is that if steam has to be produced because of users requirements on headers then it is cheaper to pass this steam through turbines because you do not have the cost of replacement involved. But if user demand is low enough then it becomes preferable to pass steam previously used by turbines through letdown stations instead. In so doing you add the cost of replacement, α *flow, but you decrease the steam production and save Csteam*flow.

An illustration of this idea is given below by setting U_{alp} to 40 instead of 10 tons/day. Then the results become:

S O L V E   S U M M A R Y

MODEL distribution OBJECTIVE z
TYPE MIP DIRECTION MINIMIZE
SOLVER OSL FROM LINE 88

***** SOLVER STATUS 1 NORMAL COMPLETION*
***** MODEL STATUS 8 INTEGER SOLUTION*
***** OBJECTIVE VALUE 12000.0000*

RESOURCE USAGE, LIMIT 1.813 1000.000
ITERATION COUNT, LIMIT 3 10000

---- 93 VARIABLE epsilonK.L decision for turbine i on header j
(ALL 0.000)

---- 93 VARIABLE FH.L = 120.000 steam production on header H

---- 93 VARIABLE LD.L letdown from the j header
H 30.000, A 30.000

---- 93 VARIABLE Vent.L = 10.000 LP steam vented to atmosphere

We see that the objective function is Z = 12000 corresponding to none of the turbines switched off. From the previous configuration we have an extra 30 tons/day to produce to satisfy the U_{alip} demand. If we pass this extra 30 tons/day through the letdown station LDH, which was already 30 tons/day, then we get a total of 60 tons/day trough it and an extra cost of: 30*Csteam = 30*100 = + R3000.

But at this stage as this steam has to be produced it becomes then cheaper to produce an extra 40 tons/day to be able to use KH1 (which requires this amount of steam) and then KM1 too, rather than passing an extra 30 tons/day through the letdown. Indeed, compared to the above solution it costs us an extra R4000 (40*100) of steam production, but we then use KH1 and KM1 so we save the replacement cost of (40+20)*20 = R1200. It is thus an overall extra cost of R4000 – R1200 = R2800 compared to the R3000 of the solution that would consist of replacing KH1 and KM1. The objective function then becomes Z = R9200 + R2800 = R12000.

We have seen on a simplified example how the structure of a GAMS program looks and what kind of results GAMS gives. At this low level of complication we can still explain and easily prove why such a solution is better than another. However when it will come to optimizing the real steam distribution the complexity becomes too high in terms of model, cost functions, number of variables and number of solutions to be able to compare solutions so easily.

This means that we will have to trust the solution given by GAMS, and so we have to build a very reliable program and use a solver with a robust algorithm.

6.3 GAMS program for the refinery distribution

We are now working on optimizing the real and current steam distribution.

6.3.1 Link between GAMS and EXCEL

Prior to the program itself we will explain in this part how to link GAMS and Excel to be able to import from Excel a full set of data needed in the GAMS program and then export GAMS results into Excel.

Indeed, one of the feature of GAMS is that the optimization problem is expressible independently of the data it uses. So what we are aiming to do is import an excel spreadsheet with all data from the plant required by our GAMS program such as turbine power demand, current steam flow, mass imbalances and so on...The separation of logic and data allows us to work both on data and the program structure independently.

There are two reasons why we want to import data from an excel file:
The first one being that we can dump data from the refinery straight from Pross (the plant database access system) into Excel and then work on these data to put them in a presentable and usable form for our GAMS program. So it would avoid a waste of time in writing all the data to the GAMS file. The second and most important reason being that we want to run our program for a large range of data to make sure that we have encountered the entire refinery working range. Indeed, refineries are multiperiod plants where demands typically vary from period to period due to market or seasonal changes. Thus, given a period of time, either hourly or daily, we are aiming to run a large sequence of data from the plant much more easily by using Excel. It will allow us to run large data sets as only one input file through the GAMS program and thus save time. We will use too the possibility of exporting GAMS results into Excel to make them more presentable and easier to work on and analyse.

The link between GAMS and Excel has been implemented by using programs downloaded from the GAMS web site: *xllink.exe, xlimport.gms, xlexport.gms and xldump.gms.*

The application *xllink.exe* allows using either *xlimport.gms* to import files from Excel to GAMS or *xlexport.gms* or *xldump.gms* both to export GAMS files into Excel.

With this link we are able to import excel files into the GAMS program with the following instruction in the program:
$libinclude xlimport T name.xls a1:u8
with *T*, the table or parameter defined in the GAMS program, *name* the name of the excel file where data will come from and "*a1:u8*" being the section in *name.xls* to be imported in *T*.

The way to export results given by GAMS into an Excel file is by writing in the program:
$libinclude xldump R res.xls
with *R*, the GAMS table or parameter where results from the optimization are located and *res.xls* the name of the excel files where these results are sent to.

Thus, we are now able to import data from the plant needed for our model in the GAMS program by using Excel. This link allows us to save lots of time and work on data and on the model independently.

6.3.2 Data in Excel files

In this part we shall explain what data are required in our model and how we worked on data available at the refinery to present an Excel file to be imported and used as a data file into our GAMS program.

We have worked at the refinery for several weeks, trying to get the data we needed by using Pross (plant database access system) and interviewing staff and engineers. That was not an easy task as lots of data were missing due to lack of or faulty flow, temperature and pressure meters. From these few weeks of investigation at the refinery we developed the complete utility section flowsheet and thus our model.

As far as cost functions are concerned, they are all in chapter 5. Costs are assumed not to be dependent on the period of time considered so Excel files have to be about "physical" data only.

We have in appendix C1 to C4, as an example, all data available at the refinery for each of the four headers for the month of March 2000.

We have chosen 3 different representative periods of time during the year and for each of these 3 periods we collected data either on an hourly basis if the period is one day or on a daily basis if the period is one month. The three periods are the months of March and December 2000 as well as the 6^{th} of December 2000. We made sure that during these periods there were no major plant shutdowns and that they represent good average working conditions of the utility section. Then for each of these periods we proceed either on an hourly or daily basis to the following calculations:
First, to a mass balance on every header as described in chapter 5. From this, we get eX the error term in tons/day on the header X, $X \in \{H, A, M, L\}$ from the difference between inflows and outflows. Then knowing steam flows and enthalpy drops across turbines, we got the power produced by every turbine W_n. This is the most important data required for a turbine, as it constitutes the demand that has to be satisfied either still by the steam turbine or by an electrical drive. Note that turbine power is not expressed in our Excel files in kW but in a unit that is 86.4 times a kW, as we multiplied flows in tons/day by enthalpies in kJ/kg. It was more convenient for us to keep on using flows in tons/day for the problem (unit used by the refinery) rather than converting all of them to kg/s.

Thus we have:

$$W_n = F_n(Tons/day) * \Delta H_n(kJ/kg) = F_n^0(kg/s) * \frac{3600*24}{1000} * \Delta H_n(kJ/kg) = 86.4 * W_n^0(kW) \qquad (6.5)$$

with: W_n power given by turbine n,

F_n flow of steam through turbine n in tons/day,

ΔH_n enthalpy drop across turbine n,

F_n^0 flow of steam through turbine n in kg/s,

W_n^0 power given by turbine n in kW.

So in our Excel files and in our GAMS program, power for any turbine *n* will appear as W_n in a unit that we will call "xW", with 1xW= 86.4 kW.

Other data presented in Excel files are power required by steam users in xW as well as flows in tons/day, enthalpy and flow of steam produced by local feeders on the MP and LP header, enthalpy of steam produced by the boilers and fraction of HP steam not condensed in K6101 on the HP header.

We have in appendix D, the four Excel files containing these calculations for each of the four headers, showing the changing values of these parameters in time. These are the files we will use as data files for our GAMS program.

6.3.3 Single solve

What we are aiming to do is run the complete data set for every period as only one input file through the Gams program. That means for each line of the data set, i.e. each time period (hour or day), we will have to run the solver with the specific conditions of the hour or the day. So we will run the solver 24 times for the period of the 6^{th} of December 2000 and 31 times each for the months of March and December 2000. The GAMS program structure remains the same, only data will change.

We will in this part concentrate on building this structure, meaning a GAMS program able to proceed to the optimization of the distribution for only one specific working condition, one time period. In other words, for only one line of the full data set.

This will constitute the major work with GAMS in our project, as this algorithm will be the heart of our final program. Indeed the solver in this program will then just have to be used for every time period of the full data set. So running the complete data set, with each day or month considered as only one input file, through GAMS, will only be a matter of calling the solver for every time period.

We will show now the general structure of this main program named *single_solve.gms* presented in appendix E. The program is built as follows:

Sets

All sets used later in the program have to be declared in this first section.

Data imported from excel

As mentioned earlier, we import from Excel data from the time period chosen. Thus we have 4 Excel files, one for each header, to consider. Data are put in 4 GAMS parameters.

Constants

Then all the figures from the model that are constant whatever the time period is, are entered as scalars.

Cost functions

These functions are not dependent on time period either. So we write them as parameter functions of W_{ij} (data imported from Excel) as they were defined in chapter 5.

Variable statement

All variables are listed in this section. They are all positive continuous variables like steam or water flows and enthalpies, except decision variables for every turbine, which are binary variables.

Equations

All equations, including the objective function to be minimized, are declared then defined exactly as defined in our modeling of the flowsheet in chapter 5.

Constraints

Constraints of our model are also declared and defined after the equations.

Initial point

As in any non-linear problem, an initial point has to be given to the solver for proceeding with the first iteration. Here again we import data from an Excel file.

Scaling

In certain problems we have to scale both variables and equations and constraints. This has to be done before the solve statement.

Options

Specific GAMS options including solver options that can be set just before the solve statement.

Solve statement

Then the solver chosen can be called depending on the kind of problem. Here we are dealing with a MINLP problem so we use DICOPT, the only MINLP solver available in our version of GAMS.

Results

We chose to present our results in a more convenient way so we used the "put" writing facility of the GAMS language in a file call *report.gms*, that we include in our main program, to put the results in another GAMS file, *res.put*. This file that can then be easily exported to an Excel file.

An organogram of the method used follows:

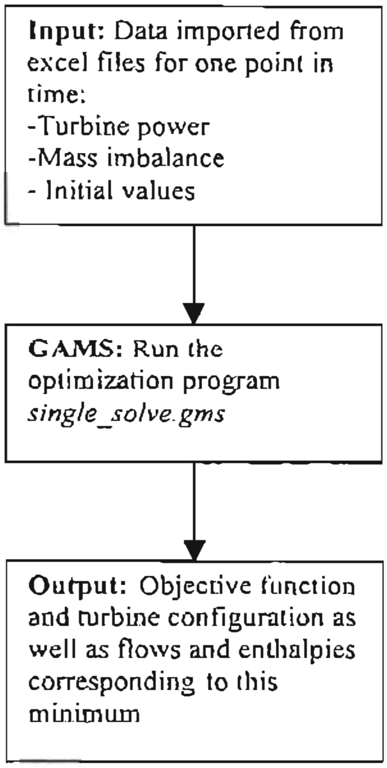


Figure 6.2: Organogram of the method for *single_solve.gms*

We presented here an overview of the main GAMS program. Having this program done properly required lots of time because we first had to get used to the structure and specifications required for writing a GAMS program. Also, because we had to increase complexity and size progressively to be closer and closer to the real problem. Sensitive and tricky points of this program will be discussed in the "problems encountered" paragraph.

6.3.4 GAMS options

In this section we will go into more detail about general options available in the GAMS language. These options are for an advance use of GAMS that normally provides default values that are adequate for the most purposes, but there are always cases when the user would like to maintain control of aspects of the run, and that is our case. Options in the GAMS language are from two different types: The Dollar Control Options and the option statement.

<u>The Dollar Control Options:</u>
They are used to indicate compiler directives and options. They are not part of the GAMS language and must be entered on separate lines recognized by a "$" symbol in the first column. They are grouped into five major functional categories affecting: input comment format, input data format, output format, reference maps and program control.

In *single_solve.gms* we use:
$title MINLP formulation of SAPREF steam distribution

sets the title in the page header of the listing file to 'text'.

$offlisting
turns off echoing input file to listing file.

$include 'report'
inserts the contents of a specified text file at the location of the call.

$libinclude xlimport HP HP_mass_balance_0612.xls b51:j52
performs the same task as the $include facility in that it inserts the contents of the specified text file at the location of the call. In addition it also passes on arguments which can be used inside the include file. If an incomplete path is given, the file name is completed using the library include directory. We used it here specifically with the command "xlimport" to import data from Excel.

The option statement:
The option statement is used to set various global system parameters that control output detail, solution process and the layout of displays. They are processed at execution time unlike the Dollar Control Options. They are provided to give flexibility to the user who would like to change the way GAMS would normally do things. The options available through the option statement are grouped into the following functional categories affecting: output detail, solver specific parameters, choice of solver, input program control.

We have in *single_solve.gms* the following options:
option solprint=off, iterlim=5000 ;

solprint controls the printing of the model solution in the listing file. The default value is "on" but we decided to turn it off so solution details are not printed as we solved the model often and we did not need this printing for every run.

Iterlim sets the maximum iteration number after which the solver interrupts the solution process and return the current solution values to GAMS. To give the solver enough iterations to reach the optimum we set it to 5000 instead of 1000 the default value which was sometimes not large enough.

We will be using later the bratio option which is used to specify whether or not basis information (from an earlier solve) is used. Indeed certain solution procedures can restart from an advanced basis that is constructed automatically. The use of this basis is rejected if the number of basic variables is smaller than bratio times the size of the basis. A bratio of 0 accepts any basis, and a bratio of 1 always rejects the basis, which is sometimes needed with nonlinear problems. Setting bratio to 0 forces GAMS to construct a basis using whatever information is available. If bratio has been set to 0 and there was no previous solve, an "all slack" basis will be provided. In *single_solve.gms* we keep the default value of 0.25.

6.3.5 Solvers and specific solver options

In our case we use the MINLP solver DICOPT++ (DIscrete Continuous OPTimizer), only called DICOPT in the GAMS language (and we will keep this notation even if it is actually the improved version of DICOPT: DICOPT++). This solver is the implementation of the extensions of the Outer-Approximation algorithm for the equality relaxation strategy developed in 1990 by J.Viswanathan and Ignacio E. Grossmann at the Engineering Design Research Centre (EDRC) at Carnegie Mellon University. The MINLP algorithm inside DICOPT solves a series of NLP and MIP sub-problems. These sub-problems can be solved using any NLP or MIP solver that runs under GAMS. That is one of the main features of the GAMS/DICOPT system, it is able to use existing or new development of NLP and MIP solvers available in GAMS.

In our case we work with OSL2 as MIP solver and CONOPT2 as a NLP solver. All these solvers are chosen as default solvers in the File\Options\Solvers directory in GAMS IDE (Integrated Development Environment) which is the GAMS environment for users.

6.3.5.1 MIP solver: OSL2

This is the new version of OSL, the IBM Optimization Subroutine Library, containing high performance solvers for LP, MIP, QP (quadratic programming) problems. It is specifically designed for solving large and difficult problems. It offers several algorithms:

- Primal Simplex algorithm (default method)
- Dual Simplex algorithm
- Network algorithm
- 3 Interior Point algorithms (primal, primal-dual and primal-dual predictor-corrector)

Normally the primal simplex method is a good method to start with. The simplex method is a very robust method, and in most cases we should get good performance with this solver. For large models that have to be solved many times it may be worthwhile to see if one of the other methods gives better results. Also changing the tuning parameters may influence the performance. The method option can be used to select another algorithm.

The simplex method:

This most used method is the primal simplex method. It is very fast, and allows for restarts from an advanced basis. In case the GAMS model has multiple solves, and there are relatively minor changes between those LP models, then the solves after the first one will use basis information from the previous solve to do a 'jump start'. This is completely automated by GAMS and normally we should not worry about this. In case of a 'cold start' (the first solve) we will see on the screen the message 'Crash...'. This will try to create a better basis than the scratch ('all-slack') basis the Simplex method would normally get. The crash routine does not use much time, so it is often beneficial to crash. Crashing is usually not used for subsequent solves because that would destroy the advanced basis. The default rule is to crash when GAMS does not pass on a basis, and not to crash otherwise. Notice that in a GAMS model you can use the bratio option to influence GAMS whether or not to provide the solver with a basis. The default behaviour can be changed by the crash option in the option file.

By default the model is also scaled. Scaling is most of the time beneficial. It can prevent the algorithm from breaking down when the matrix elements have a wide range: i.e. elements with a value of 1.0e-6 and also of 1.0e+6. It can also reduce the solution time.

The presolver is called to try to reduce the size of the model. In addition to these reductions OSL can also remove some redundant rows, and substitute out certain equations. The presolver has several options which can be set through the presolve option.

The presolve may destroy an advanced basis. Sometimes this will result in very expensive restarts. As a default, the presolve is not used if an advanced basis is available. If using the presolve procedure is more useful than the use of an advanced basis, one can still force a presolve by using an option file.
GAMS/OSL uses the order: scale, presolve, crash.

After the model is solved we have to call the postsolver in case the presolver was used. The postsolver will reintroduce the variables and equations that presolve substituted out, and will calculate their values. This solution is an optimal solution, but not a basic solution. By default we call simplex again to find an optimal basis. This allows us to restart from this solution. It is possible to turn off this last step by the option postsolve 0, but in our case we will always keep the default value postsolve 1 to get a basic solution.

Occasionally we may want to use the dual simplex method, for instance when the model is highly primal degenerate, and not dual degenerate (primal form yields multiple solutions but not the dual form), or when we have many more rows than columns. We can use the method dsimplex option to achieve this. In general the primal simplex (the default) is more appropriate: most models do not have the characteristics above, and the OSL primal simplex algorithm is numerically more stable.

The interior point methods

OSL also provides three interior point solvers. It is worthwhile to try them out especially when the model is large and if it is not a restart. The primal-dual barrier method with predictor-corrector is in general the best algorithm. This can be set by the method option in the option file. These methods produce a sequence of points that converge to the optimum along a trajectory through the interior of the feasible region instead of skirting its periphery, as the simplex method does.

The network method

The pure network solver included in OSL is an implementation of the simplex method that takes advantage of the special form of the constraint matrix of the problem. Constraint matrices for pure network programming problems take a particularly simple form. Each column has only two nonzero entries and the values of these entries are plus or minus one. Consequently, many of the intermediate computations of the simplex method, such as matrix multiplications and factoring of certain coefficient matrices, can be performed implicitly or avoided altogether.

6.3.5.2 NLP solver: CONOPT2

Currently, there are two standard NLP algorithms available, MINOS and CONOPT, and CONOPT is available in two versions, the old CONOPT and the new CONOPT2. All algorithms attempt to find a local optimum. The algorithms in MINOS and CONOPT are based on fairly different mathematical algorithms, and they behave differently on most models. This means that while MINOS is superior for some models, CONOPT is superior for others. Even CONOPT and CONOPT2 behave differently. The new CONOPT2 is best for most models and that is why we chose to use it instead of CONOPT, even if there is a small number of models that are best solved with the old CONOPT.

The only reliable way to find which solver to use for a particular class of models is so far to experiment. However there are a few rules of thumb to choose between CONOPT2 (we will for convenience use the generic name CONOPT) and MINOS:

GAMS/CONOPT is well suited for models with very nonlinear constraints. MINOS has much more difficulty in establishing if a model is infeasible, so one would like to use CONOPT for NLP subproblems that are infeasible or barely feasible. We have a model with lots of nonlinearities outside the objective function so it may be better to use CONOPT.

GAMS/CONOPT has a fast method for finding a first feasible solution that is particularly well suited for models with few degrees of freedom. So for problems with roughly the same number of variables and constraints it is advised to use CONOPT. We have 134 variables and 104 constraints so here is another reason to use CONOPT.

GAMS/CONOPT has a preprocessing step in which recursive equations and variables are solved and removed from the model. If many equations can be solved one by one then CONOPT will take advantage of this property. Similarly, intermediate variables only used to define objective terms are eliminated from the model and the constraints are moved into the objective function. We have a good number of these equations and intermediate variables in our model that makes us think that CONOPT could be useful here again. However, especially in our mass and energy balances the difficulty comes from the fact that equations cannot be solved one by one.

GAMS/CONOPT has been designed for large and sparse models. This means that both the number of variables and equations can be large which is our case. The assumption made in CONOPT that the model is sparse (i.e. that most functions only depend on a small number of variables) is satisfied in our model. CONOPT can also be used for denser models, but the performance will suffer significantly.

Both GAMS/CONOPT and GAMS/MINOS are designed for models with smooth functions (i.e., their first derivatives must exist). But GAMS/CONOPT can also be applied to models that do not have differentiable functions although no guaranties are given for this class of model.

GAMS/CONOPT has many built-in tests and messages and is therefore also a helpful debugging tool during model development.

All these points lead us to rather choose the NLP solver CONOPT2 for our specific model for better performance. Besides, experiments with several different data from the refinery using our program *single_solve.gms* (described in section 6.3.3 and presented in appendix D) that are presented above, with both MINOS and CONOPT2 confirm that CONOPT2 gives better solutions for the objective function to minimize z.

Here are presented these results from 15 different runs of our program with 15 points in time (working conditions) from the March 2000 data set:

	Objective function Z (R/year)		Resource usage (sec)		Iteration count	
tests	CONOPT2	MINOS	CONOPT2	MINOS	CONOPT2	MINOS
1	201,826,116	205,104,200	4.89	5.16	1084	1447
2	211,905,038	214,691,879	18.36	6.72	3148	2225
3	203,243,903	203,807,161	3.96	4.89	1730	1955
4	199,023,361	213,354,676	7.56	4.59	1568	1397
5	204,629,053	220,414,345	2.93	2.49	1123	1204
6	201,647,150	215,739,224	2.81	2.26	1315	530
7	221,092,095	221,118,973	2.57	3.91	344	1016
8	206,725,678	213,444,902	8.41	1.81	860	649
9	210,436,386	213,629,376	20.99	5.11	1465	1656
10	207,411,829	207,411,829	22.83	4.28	1430	1309
11	203,594,096	207,881,917	22.98	2.32	1666	1053
12	213,715,466	213,843,489	13.84	2.27	592	781
13	212,637,594	212,789,555	28.36	2.51	643	970
14	196,690,765	203,388,346	15.80	1.66	616	598
15	205,749,325	215,078,846	9.67	2.24	349	606
average	206,688,524	212,113,248	12.40	3.48	1196	1160

Table 6.1: CONOPT2/MINOS comparison

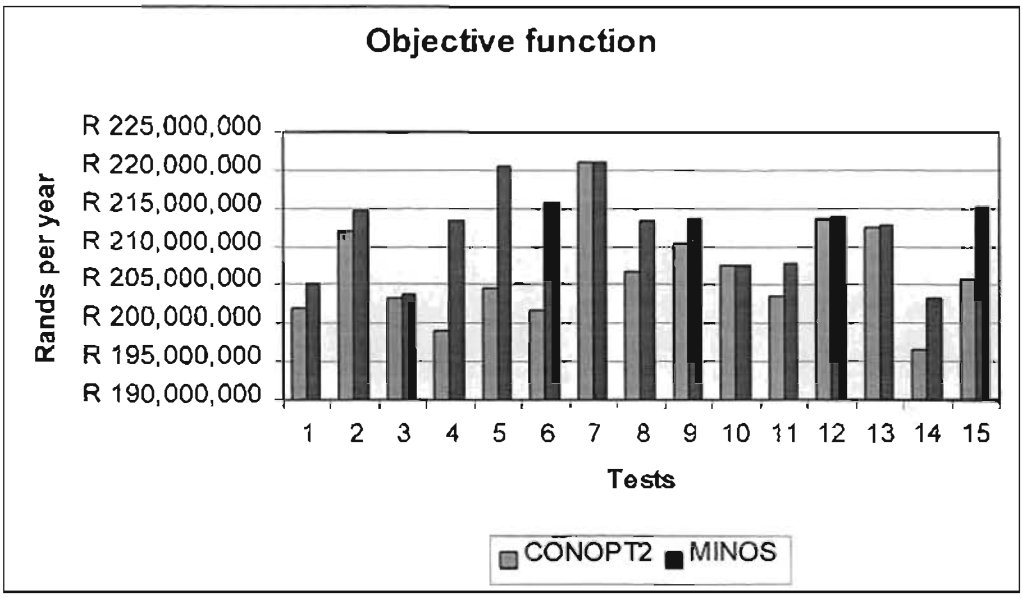


Figure 6.3: Objective functions comparison between CONOPT2 and MINOS

So we see that CONOPT 2 always gives better or in the worst cases identical results to MINOS. CONOPT 2 results are on average 2.6% lower than MINOS.

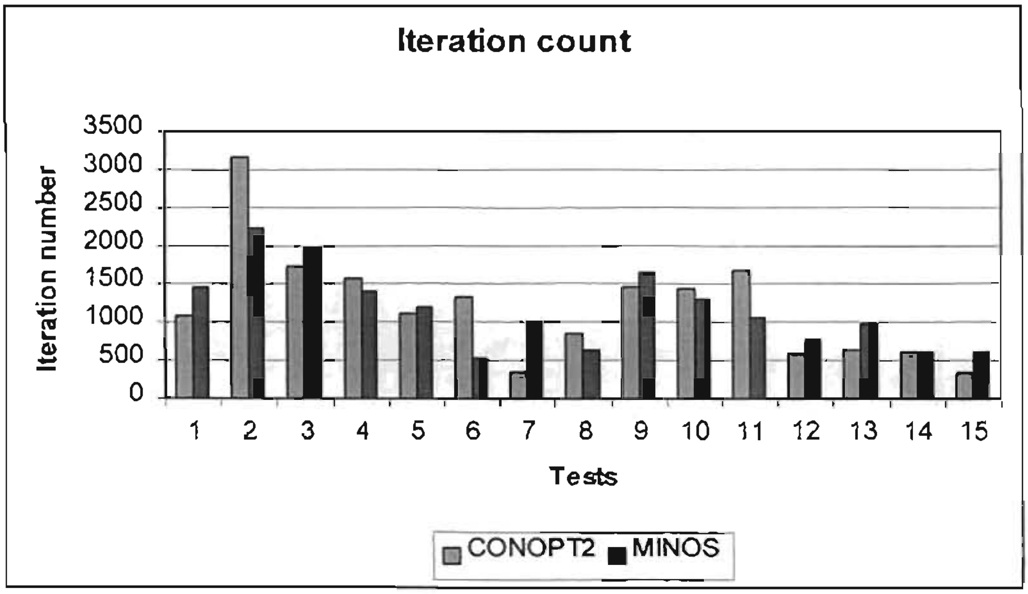


Figure 6.4: Iteration count comparison between CONOPT2 and MINOS

As far as iteration numbers are concerned, MINOS requires on average less iterations before stopping but as seen above with the objective functions, it is because CONOPT 2 continues the search further and reaches some better solutions.

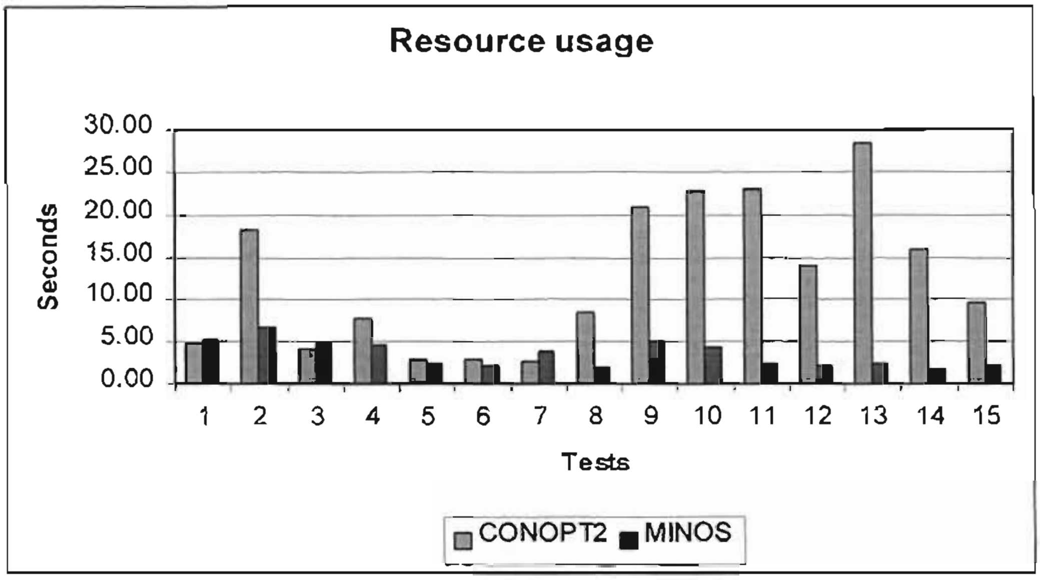


Figure 6.5: Resource usage comparison between CONOPT2 and MINOS

Finally, resource usage required by CONOPT 2 is generally much longer than the time MINOS needs. Once again it is because of the greater number of iterations used by CONOPT 2 and also because one CONOPT 2 iteration needs more time (for instance because of the preprocessing step...).

As the computational requirements (both time and iteration number) involved with CONOPT 2 are still reasonable and do not penalise it too much in our case, we chose CONOPT 2 as a NLP solver because it is the one that gives better solutions.

Most modelers should not be concerned with algorithmic details such as choice of algorithmic sub-components or tolerances. CONOPT2 has considerable build-in logic that selects a solution approach that seems to be best suited for the type of model at hand, and the approach is adjusted dynamically as information about the behaviour of the model is collected and updated.
That is why we will mainly consider options available from the MIP solver, OSL2.

6.3.5.3 Solver options

In addition to the general options available in the GAMS language, most solvers like OSL2 and CONOPT2 allow the user to affect certain algorithmic details and thereby affect their effectiveness. The default values for all the solver specific options are usually appropriate for most problems. However, in some cases, it is possible to improve the performance of a solver by specifying non-standard values for some of the options, as seen above.

To have a solver use an option file, it is necessary to inform the solver in the GAMS model by setting the optfile model suffix to a positive value. So we have in single_solve.gms the following line:

distribution.optfile = 1;
"distribution" being the name of our model. The name of the option file that is specific to each solver being used after this assignment is solvername.opt, where solvername is the name of the solver that is specified and the suffix opt.

So we use with *single_solve.gms*, the option file *dicopt.opt*. There are several options available in DICOPT such as the major iterations limit or the stop criterion to be used but even if we used these options for experimentation purposes, we are not using them in the final version of our program. Actually the only DICOPT option that we are going to use is to specify an option file for the MIP master problems. Indeed DICOPT solves a succession of MIP and NLP subproblems and as mentioned in the previous paragraph we want to be able to tune options of the MIP solver, OSL2. The default options for the NLP solver CONOPT2 are good enough. In order to do so, we have to specify in the DICOPT option file that we create a MIP option file by the following instruction:
mipoptfile osl.opt

So for all the MIP subproblems the option file called *osl.opt* will be used.
We can now tune all the OSL2 options by entering them in the *osl.opt* option file.
We will use in our work the following three options mentioned in the paragraph 6.3.5.1:

Method option:
Select one of the MIP solvers available in OSL2

psimplex	uses primal simplex method as solver (default)
dsimplex	uses dual simplex
network	uses the network solver
interior1,2 or 3	uses primal, primal dual or primal dual predictor-corrector barrier method.

Crash option:
Crash an initial basis if GAMS does not provide a basis. To tell GAMS never to provide the solver with a basis we use option bratio = 1, in the GAMS program.

0	No crash is performed
1	Dual feasibility may not be maintained (default value if no basis provided by GAMS).
2	Dual feasibility is maintained.
3	The sum of infeasibilities is not allowed to increase
4	Dual feasibility is maintained and the sum of infeasibilities is not allowed to increase.

Presolve option:
Perform model reduction before starting the optimization procedure. This should not be with network solver. Presolve can sometimes detect infeasibilities which can cause it to fail, in which case the normal solver algorithm is called for the full problem.

-1 Do not use presolve (default value for restarts).

0 Remove redundant rows, the variables summing to zero are fixed. If just one variable in a row is not fixed, the row is removed and appropriate bounds are imposed on that variable (default value for cold start).

1 As 0, and doubleton rows are eliminated (rows of the form $px_j + qx_k = b$).

2 As 0, and rows of the form x1=x2 + x3..., x>0 are eliminated.

3 All above are performed.

So an example of an *osl.opt* option file is:

```
method dsimplex
crash 0
presolve 2
```

Thus we shall experiment with a large range of these options in order to get the algorithmic method that gives the best solutions for our model.

6.3.6 Stopping Rule

Based on experience with other models, the DICOPT default stopping rule, which is to stop when the NLP becomes worse, performs well in practice. In many cases it finds the global optimum solution, for both convex and non-convex problems.

First, DICOPT often finds the best integer solution in the first few major iterations. Second, in many cases as soon as the NLP's start to give worse integer solutions, no better integer solution will be found anymore. This observation is the motivation to make the stopping rule where DICOPT stops as soon as the NLP's starts to deteriorate the default stopping rule.

However in case we want more reassurance that no better integer solutions are missed we can use the option stop 0 that does not make DICOPT stop unless an iteration limit, resource limit, or major iteration limit is hit or a feasible MIP master problem becomes infeasible.

Here is an example of a DICOPT output:

--- *DICOPT: Stopped on NLP worsening*

The search was stopped because the objective function

of the NLP subproblems started to deteriorate.

DICOPT Log File

Major Step	*Major Iter*	*Objective Function*	*CPU time (Sec)*	*Itera-tions*	*Evaluation Errors*	*Solver*
NLP	*1*	*1.84491*	*0.50*	*70*	*0*	*conopt2*

MIP	*1*	*1.84491*	*0.22*	*117*	*0*	*osl2*
NLP	*2*	*1.93836<*	*0.11*	*4*	*0*	*conopt2*
MIP	*2*	*1.86775*	*1.59*	*310*	*0*	*osl2*
NLP	*3*	*1.86110<*	*0.11*	*7*	*0*	*conopt2*
MIP	*3*	*1.88762*	*3.24*	*560*	*0*	*osl2*
NLP	*4*	*1.87202*	*0.22*	*11*	*0*	*conopt2*

Total solver times : NLP = 0.94 MIP = 5.05

Perc. of total : NLP = 15.66 MIP = 84.34

The integer solution marked with a '<' are an improvement.

6.3.7 Problems encountered

Building this program and going into details concerning solver options was not an easy task and took us some time. We actually built this program step by step increasing difficulty along the way. In this section we will point out the sensitive stages we had to get over.

6.3.7.1 Initial point

Initial values are important for many reasons. Initial values that satisfy or closely satisfy many of the constraints reduce the work involved in finding a first feasible solution. Initial values that in addition are close to the optimal ones also reduce the distance to the final point and therefore indirectly the computational effort. Finally, nonconvex models may have multiple solutions and in such a case the chances of finding a global optimum are usually increased by choosing a starting point that is "sufficiently close" to the global solution.

In our case we are not expecting major changes especially in continuous variables such as steam flows and enthalpies so a starting point close enough to the global optimum is to use the current working conditions of the utilities section of the refinery when no change has been made yet. So the same data from the refinery that were previously used for providing input to the GAMS program are now imported from excel files again but to set initial values to the continuous variables. A few tests with other initial values in a reasonable range around expected optimum values showed no major change nor improvements to the objective function. Besides current working conditions satisfy many of the constraints so reduce the work involved in finding a first feasible solution. They constitute a good point to start with that we will keep for our work.

As DICOPT first solves a NLP subproblem, we have to provide initial values for the discrete variables for the first NLP subproblem as well, in order to try to get a better starting point than the default value of

0 for all of the initial binary variable values that is otherwise used. The best way to find good initial values is once again by experimentation. But as we have 26 discrete variables we could not afford to test the 2^{26} = 67,108,864 combinations involved so we grouped discrete variables (decisions on steam turbines) by headers. Thus we only have to find the best combination between 2^3 = 8 possibilities as a starting point. Results are presented in the following table 6.2:

Discrete variables	Starting points							
HP turbines	0	0	0	0	1	1	1	1
AMP turbines	0	0	1	1	0	0	1	1
MP turbines	0	1	0	1	0	1	0	1
Objective function (R/year)	190 306 149	190 260 741	189 825 078	infeasible	189 099 927	190 796 547	190 260 916	189 735 389

Table 6.2: Tests on starting point

So the starting point corresponding to the HP turbines switched to electrical drives and all the remaining turbines kept on steam use is slightly better than other options. Note that these tests have been done with no option files, i.e. with the default parameters for the OSL2 solver. The best starting point could change a little bit with the use of OSL2 options because the algorithm would not be exactly the same. However we shall keep this starting point as it is for the rest of our work as the idea is to have a starting point close enough to the global solution, and avoid the infeasible region, and as it only constitutes the starting point for discrete variables of the first NLP subproblem. The discrete values for NLP subproblems in the following major iterations are given by the solution of MIP subproblems.

As a conclusion on initial values to be chosen, we have with the current conditions at the refinery the starting point as far as continuous variables are concerned and in addition with the combination above, $\varepsilon_{hl}^{0} = 1$, $\varepsilon_{al}^{0} = 0$, $\varepsilon_{ml}^{0} = 0$ as far as discrete variables are concerned, we get a good initial point to work with.

6.3.7.2 Scaling factors

Nonlinear as well as linear programming algorithms use the derivatives of the objective function and the constraints to determine good search directions, and they use the function values (f and g in (P1), section 3.1) to determine whether constraints are satisfied or not. The scaling of the variables and constraints, i.e. the units of measurement used for the variables and constraints, determine the relative size of the derivatives and of the function values and thereby also the search path taken by the algorithm. So working with a well scaled model is important for getting the optimal solution itself and reducing the solution time as well. The objective of scaling is to get matrix elements in as narrow a size range as possible. A well scaled model means a model where basic and superbasic solution values are expected to be around 1, e.g. from 0.01 to 100. Variables become well scaled if their expected value is around 1. After having scaled variables we must select a unit of measurement for the equations so that the expected values of the individual terms are around 1 as well. Derivatives will usually be well scaled whenever the variables and equations are well scaled.

Scaling in GAMS language is turned off by default. Setting the model suffix.scaleopt to 1 turns on the scaling feature. So we have:

```
Model distribution /all/;
distribution.scaleopt = 1;
```

The scale factor of a variable or an equation is referenced with the suffix .scale. The scale factor on a variable, Vs, is used to relate the variable as seen by the user, Vu, to the variable as seen by the algorithm, Va, as follows:

$$Va = \frac{Vu}{Vs} \tag{6.6}$$

For example with the variable FH, HP steam production is in tons/day. The current (and initial) value is 6028 tons/day in *single_solve.gms*. The aim is to bring this variable as close as possible to 1, as seen by the algorithm. We will use the scaling factor 1000 so we have:

$$Va_{FH} = \frac{Vu_{FH}}{Vs_{FH}} = \frac{6028}{1000} = 6.028 \tag{6.7}$$

So for all continuous variables (discrete variables do not have to be scaled) expecting to be of the form $X * 10^n \quad X \in [1;10]$, based on their initial values, the scaling factor we used is generally 10^n.

Similarly, the scale factor on an equation, Gs, is used to relate the equation as seen by the user, Gu, to the equation as seen by the algorithm, Ga, as follows:

$$Ga = \frac{Gu}{Gs} \tag{6.8}$$

We have to perform a combination of equation and variable scaling until a well scaled model is obtained as variable scaling factors affect the choice of a scaling factor for equations.

Example of scaling on the constraint on the power demand for turbine K6101:

Equation (5.54) for constraints to meet the power demand gives:

$$W_n * (1 - \varepsilon_n) = F_n * (H_i - h_n^0)$$

applied with the following figures:

$$\begin{cases} W_{K6101} = 261687 \\ H_{HP} - h_{K6101}^{0} = 167 \\ Fa_{K6101} = \dfrac{F_{K6101}}{100} \end{cases}$$

The scaling factor for all of the HP turbine steam flows is chosen equal to 100 because initial flows on this header are between 430 and 2397 tons/day. So the equation as seen by the algorithm in GAMS, Ga_{WK6101}, is:

$$Ga_{WK6101} : \frac{(261687 - 261687 * \varepsilon_{K6101} = 100 * Fa_{K6101} * 167)}{Gs_{WK6101}} \tag{6.9}$$

with Gs_{WK6101}, scaling factor for this equation.

Now by choosing $Gs_{WK6101} = 1000000$ we get:

$Ga_{WK6101} : 0.261687 - 0.261687 * \varepsilon_{K6101} = 0.0167 * Fa_{K6101}$

So we have seen in this example how to bring the matrix elements into a smaller range, closer to 1. We proceed to the scaling of all variables and equations as shown in appendix E in the GAMS program listing in the same way. So we finally get matrix elements approximately in the range 0.01 to 100 which avoids numerical inaccuracies due to underflow or overflow. Finally, after GAMS terminates the optimization, it recalculates the values as seen by the user, Vu, of all the optimum variables and gives them in the results section.

We performed some tests on scaling factors of both variables and equations to check what their influences could be on the objective function. It appears that different factors from those we calculated above do not improve the objective function but can lead to small changes, all of them giving a worse objective function than the one we have with our calculated scaling factors. Changes that occurred where only on small MP turbine configurations (as in changes due to different starting points as seen in the previous section) so the objective function was only slightly affected by these small changes in the optimum configuration.

Thus, as no improvements were found from these tests, this confirms that our scaling factors are satisfactory for our model. So we will use these scaling factors for *single_solve.gms* in our following work and possible changes in the GAMS program in order to keep a well-scaled model.

6.3.7.3 RMINLP and intermediate variables

RMINLP

We progressively increased the complexity of the model to eventually get to a MINLP problem. Thus, we first made assumption of linearity in our model to deal with MIP. Once we were confident about it and got some reliable and repeatable results, we moved on to a RMINLP problem (Relaxed Mixed Integer Non Linear Programming) by including non linear equations in our model and relaxing integer variables to continuous variables bounded between 0 and 1. To first solve our problem with relaxed discrete variables allows us to be confident about our model and to make sure that nothing else causes a problem. This class of problem is the same as NLP in terms of difficulty of solution. So after having run our

program with CONOPT2, which is also a RMINLP solver, and getting reliable results, we were able to move on to the MINLP problem simply by setting the solver statement in our GAMS program to the following form:
Solve distribution using minlp minimizing z ;
Instead of:
Solve distribution using rminlp minimizing z ;
As it was previously for solving the corresponding RMINLP problem.

Intermediate variables
Nonlinearities in our model are of two different kinds. We have nonlinearities in some functions as in the replacement cost function of the form: $y(x) = A*x^2 + B*x + C$.
But most of the nonlinearities in our model are of the form: $\varepsilon*V$ with ε a binary variable and V a continuous variable. Indeed, we have many terms of that form in mass and energy balances for instance steam flows F_i are subject to the existence or not (so to the corresponding discrete variable ε_i) of the steam turbine i.

DICOPT cannot handle this second type of nonlinearity because it requires that binary variables only appear in linear terms (requirement for the AP/OA/ER algorithmic method). However, we can easily get over this problem. Indeed, one way to work around this restriction is to introduce intermediate continuous variables in place of binary variables in our model, and a set of equations that makes them equal to the binary variables. In *single_solve.gms*, we call these intermediate continuous variables inthp, intap and intmp, respectively used in place of epshp, epsap and epsmp.

So we have seen in this paragraph 6.3.7 all the sensitive and difficult points of our model. We eventually arrived at an accurate model and a working GAMS program that has to be tested with different OSL solver options to find the algorithm that gives the global optimum. However this program, *single_solve.gms* can only be run with one set of data corresponding to one point in time. To avoid wasting time in changing data for each run, we will now work on doing a GAMS program with the same model but capable of running full data sets using only one input file through the program.

6.3.8 Multi solve

In this paragraph we will explain how to run a full data set as only one input file through a GAMS program. Indeed we have available from the refinery three large data sets, one for the working conditions every hour on the 6th of December 2000 and two for average conditions every day for the months of March and December 2000. As measured working conditions, forming the input data for our model, are different from one point in time to another this could lead to different optimal turbine configurations from one point in time to another. So what we are aiming to do here is by running the sequence of data we have through the model to hopefully get an optimal pattern emerging for most of the working conditions.

To do so we are using the feature that several solve statements can be processed in the same program in GAMS. The model remains the same for each resolution. We write a loop in which we set the data to be used (from the full data set imported from Excel files) following by the solve statement inside the loop as well. Thus we are able to proceed to the optimization of the distribution for each point in time of the full data set within only one program. Then we put the results of each point in time to a GAMS parameter that we export into an Excel file for convenience.

The listing of this program, called *multi_solve.gms*, is presented in appendix E. The structure of the program being very close to *single_solve.gms*, we actually only present in appendix E, the listing of *multi_solve.gms* with differences from *single_solve.gms* written in bold characters.

It was not easy to get the multi-solve version working properly because of specific GAMS requirements when putting the solve statement in a loop. This program takes obviously more time to run than *single_solve.gms* but it is still worth using it rather than changing data in *single_solve.gms* for each point in time. Its computational time is very reasonable as it only takes approximately 5 minutes to run for 31 data points.

Here are a few points of explanation for this new program:

Initial point

Note that in multi_solve.gms we only have to provide a starting point for the first solve, so for the first hour if we are running it with hourly data and for the first day if we are using daily data. We do not have to provide GAMS with initial values for the next points in time because it already uses as much information as possible from the previous solution to provide a starting point in the search for the next solution.

We proceeded to do some tests using the optimum solution variables from the previous solution as a starting point for the following point in time with *single_solve.gms* but it did not lead to a solution as good as the one given for the same point in time within *multi_solve.gms*. These tests prove that GAMS is not only using variable optimum values from the previous solution but an advanced basis including more information on the model and algorithm behaviors.

MIP solver options

The best way to find the good MIP algorithm options is experimentation. So we will test different choices of options such as bratio in the main GAMS program and method, crash and presolve options in the OSL option file. Indeed these options can be useful and beneficial to use so they are worth trying even if it means we have to run *multi_solve.gms* a lot of times in changing these options for each run.

Exporting "0" to excel files

Note here that a specification of xldump is that it can not export directly the figure "0" to Excel. So in order to export zeros from the GAMS parameter called "rep", where the results are, we write an "if" loop to change the possible zeros in rep to a value of "eps" (a small value), which can then be exported and understood as a zero in Excel. This is especially necessary for the binary variables that can only take the value 0 or 1.

We have now a program able to proceed to the optimization of our model for a full set of data imported as only one input file through GAMS.

Below is presented an organogram of the method used:

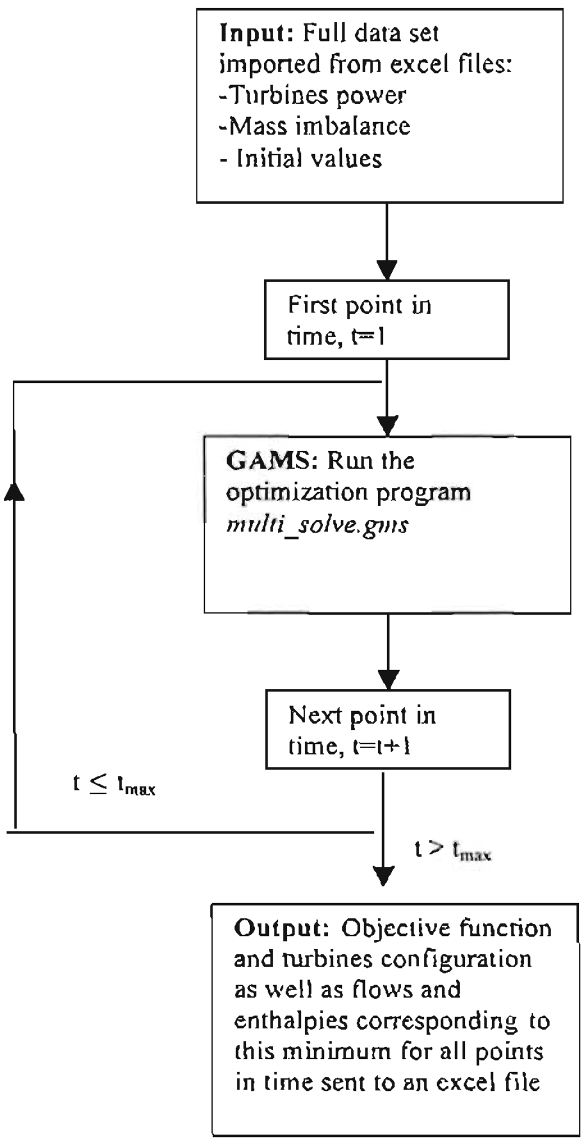


Figure 6.6: Organogram of the method for *multi_solve.gms*

In this chapter we became familiar with the GAMS language and we explained how we used it to progressively build a complex program that solves our current optimization problem. We shall, in the next chapter, use this program to get the final results of our project.

Chapter 7

Optimum turbine configuration and profits

We are, in this chapter, using the GAMS program *multi_solve.gms* elaborated in the previous chapter to proceed to the optimization itself. We run it with our three sets of data available and we conduct sensitivity tests on both algorithm options and model parameters to get to the final optimum turbine configuration and all the costs involved. Note that algorithm options descriptions are based on our experiments and on definitions in the GAMS manual, as well as the solvers manual, both from the GAMS Development Corporation (1996).

7.1 Algorithm options choice

As we have seen in the previous chapter we can set different choices of options into the OSL2 MIP solver. In order to choose options that give the best result we proceed to some experiments with *multi_solve.gms* using the March 2000 set of data. More details about these options can be found in section 6.3.5.3.

7.1.1 Method

We first have to choose the algorithmic method that gives the lowest objective function. Here are tables of our experiments with all other options set at their default values.

	The Simplex methods	
Methods	primal simplex	dual simplex
Solver status	Normal completion	
Model status	Integer solution	
Average objective function (R/day)	210 872 033	**207 984 485**

Table 7.1: Test with the simplex methods

	The interior point methods		
Methods	primal barrier	primal-dual barrier	primal-dual predictor-corrector barrier
Solver status	Error solver failure		
Model status	Error no solution		
Average objective function (R/day)	No result		

Table 7.2: Test with the interior point methods

Methods	network
Solver status	Normal completion
Model status	Integer solution
Average objective function (R/day)	210 647 162

Table 7.3: Test with the network method

So we can see from these results that the dual simplex method suits our model the best. Note that for some particular points in time in the full data set the primal simplex method may be better, due to zero flows or simplifications that the method takes advantage of. All of the interior point methods are faulty for our model - the MIP algorithm fails to find a solution. Our model has not the particular form of a pure network model (special form of the constraint matrix) which explains why the network method does not take much advantage of our model characteristics. So this method is worse than the dual simplex method even if a bit better on average than the primal simplex method. The reasons for the dual simplex method being the best are likely to be that the model is highly primal degenerate and not dual degenerate or that it has many more rows than columns. However our theoretical knowledge here happens to be too limited to explain this in detail so we will rely on our experimentation work. So based on the average objective function of our experiments, we will use the dual simplex method for the rest of our work.

7.1.2 Bratio option

Before speaking of GAMS/OSL algorithmic parameters we have at this stage to test the bratio option which is a general GAMS option. Indeed certain solution procedures can restart from an advanced basis that is constructed automatically. This option is used to specify whether or not basis information (probably from an earlier solve) is used. The use of this basis is rejected if the number of basic variables is smaller than bratio times the size of the basis. Here are the results obtained:

	dual simplex method				
Bratio	0	0.1	0.25	0.9	1
Average objective function (R/day)	209,296,774	**207,984,485**	**207,984,485**	**207,984,485**	208,750,302

Table 7.4: bratio option

Setting bratio at any values between 0 and 1 (but not 0 and 1 themselves) gives the best average answer. Bratio=0.25 is the default value. No changes occur between bratio=0.1 and bratio =0.9. This means that the number of basic variables is larger than 0.9 times the size of the basis. In other words, the advanced basis from earlier solves is used and gives slightly better results than when bratio=1 meaning that all existing basis information is discarded. A bratio of 0 accepts any basis built by GAMS but when there is no previous solve an "all slack" basis (also called 'all logical') is provided, here worse than the one built from an earlier solve. We were expecting results to be better when using a basis from earlier solves. Indeed in this kind of program where several solves are undertaken, GAMS uses as much information as possible from the previous solution to restart from an advanced basis and to provide for instance a starting point in the search for the next solution. So the use of an advance basis is often beneficial in a program with several solve statements for the same model. However setting bratio to 1 allows us to test the crash option to see if any improvements can be achieved by using the crashing procedure. We will speak about this option in a paragraph below.

7.1.3 Presolve

The first procedure that takes place in the GAMS/OSL algorithm is the presolver. The presolve option sets the presolver parameters. Indeed, the presolver is called to try to reduce the size of the model by simple reductions, removing some redundant rows and substituting out certain equations before the model is solved. Here is a table of the results we obtain in trying the range of values available for the presolve option:

	dual simplex method, bratio = 0.25		
presolve	default	-1	0
Action	0 for cold starts Do not presolve for restarts	Do not use presolve	Remove redundant rows variables summing to zero are fixed
Av. objective functions (R/day)	207 984 485	209 140 317	208 457 723

	dual simplex method, bratio = 0.25	
presolve	1	2
Action	As 0, and doubleton rows are eliminated	As 0, and rows of the form x1=x2+x3..., x>0 are eliminated
Av. objective functions (R/day)	208 014 482	209 392 859

Table 7.5: Presolve option

We see that the default value of the presolve option gives the best result. This default value uses the presolver for the first solve ('cold start') and then does not use it for subsequent solves because the presolver may destroy an advanced basis. Even if these solutions are likely to be local optima, we can see in our case that actually using the presolver procedure for all the solves is less beneficial than the use of an advanced basis since we get a better (meaning lower) objective function with the default value of the presolve option than with any other value. On the other hand, to not use the presolver for the first start is also less beneficial because it can reduce significantly the size of the model and there is not an advanced basis yet. So we keep the default value for this presolve option.

7.1.4 Crash

The next step after the presolve procedure is the crash routine. Indeed, in case of a 'cold start' (the first solve) we will see on the screen the message 'Crash...'. This will try to create a better basis than the scratch ('all-slack') basis the dual simplex method would normally get. The crash routine does not use much time, so it is often beneficial to crash. Crashing is usually not used for subsequent solves because that would destroy the advanced basis so to use this option we have to tell GAMS never to provide the solver with a basis by setting the bratio option to 1. We change the crash option in the OSL2 option file to see the results given by GAMS:

	Dual simplex method					
Bratio option	0,25		1			
Crash	0	default	1	2	3	4
Action	No crash performed	crash the first solve only	Dual feasibility may not be maintained	Dual feasibility maintained	Sum of infeasibilities not allowed to increase	Action from 2 and 3
Av. objective functions (R/day)	207 460 890	207 984 485	208 750 302	211 752 581	209 037 906	212 717 442

Table 7.6: Crash option

From these tests on the crash option we can see that not to perform crash is the best option for our model. Note that in that specific case where we do not crash the bratio option does not have to be set to 1 as no crash is performed and so we can still use an advanced basis. These results confirm the fact that crashing is not beneficial for subsequent solves as it destroys the advanced basis used to do a 'jump start'.

As we mentioned above, by default the crashing procedure is used for the first solve to try to create a better basis than the scratch basis but we can see that in our case not to perform a crash even at the first solve gives a better objective function, so the scratch basis is still better.
Therefore we keep the crash option set to 0.

Eventually our *osl.opt* file will be:

```
method dsimplex
crash 0
```

We have now reached a point where all the algorithmic parameters have been tuned with satisfaction. We shall keep them as they are and get the final solution and turbine configuration from the GAMS program.

7.2 Best turbines configuration

We can now run our final program and get the best configuration and the corresponding objective function for each of our three data sets.

7.2.1 March data set

We have in appendix F the Excel file with the results for every point in time in March. From this we calculate the average values of both the decision variables for every turbine and the objective function. Here are these average values:

	HP turbines			
Turbines	K3471	**K6101**	**K6102**	K301
Average Epsilon	0.193548	**0.967742**	**0.741935**	0.322581
Decision	0	1	1	0

	AMP turbines				
Turbines	K3262	K3271	K3272	U3200	U3500
Average Epsilon	0.193548	0.322581	0.258065	0.193548	0.419355
Decision	0	0	0	0	0

	MP turbines							
Turbines	K4402	K7101	K8001	others	P3202	P3261	P3262A	P3263
Average Epsilon	0.258065	0.322581	0.096774	0.16129	0.16129	0	0.290323	0.096774
Decision	0	0	0	0	0	0	0	0

	MP turbines (2)							
Turbines	P3371	P3501	P3701C	P6102A	P6110	P6115	P8030	U300
Average Epsilon	0.322581	0.129032	0.483871	0	0.096774	0.322581	0.387097	0.419355
Decision	0	0	0	0	0	0	0	0

Table 7.7: March optimal configuration

The "average Epsilon" means that for that specific percentage of the time it is beneficial to switch this steam turbine to an electrical drive. So we will keep the turbines which have an average Epsilon under 50 % on steam use and consider for a change those which have an average Epsilon over 50%. Of course this simple procedure for selection of an optimal configuration ignores combinatorial switching of the epsilons. The term "others" in the table 7.7 refers to MP turbines with a low steam consumption, not shown on the steam distribution flowsheet.

So from the March data set it appears that only two HP turbines K6101 and K6102 have to be switched to electrical drives. We shall speak about the actual cost benefit of such results in a following section. We shall first get the results from the others sets of data.

Note that our problem could not be treated as a multiperiod optimization problem with switch on/off costs depending on the period. Indeed we can not switch one turbine to an electrical drive for one period of time and then put it back for the next period. We have not considered the possibility of variable configurations because of the practical difficulties of installing parallel equipment for the large compressors being driven by these turbines (a configuration once chosen is fixed). So we have to keep only one configuration, which is the best average as this configuration will not be changeable.

7.2.2 December data set

Running our GAMS program with the December data set gives the following results for turbine configurations:

	HP turbines			
Turbines	K3471	**K6101**	**K6102**	K301
Average Epsilon	0	**1**	**0.612903**	0.419355
Decision	0	**1**	**1**	0

	AMP turbines				
Turbines	K3262	K3271	K3272	U3200	U3500
Average Epsilon	0.258065	0.258065	0.129032	0.225806	0.354839
Decision	0	0	0	0	0

	MP turbines							
Turbines	K4402	K7101	K8001	others	P3202	P3261	P3262A	P3263
Average Epsilon	0.322581	0.225806	0.16129	0.419355	0.193548	0	0.16129	0.354839
Decision	0	0	0	0	0	0	0	0

	MP turbines (2)							
Turbines	P3371	P3501	P3701C	P6102A	P6110	P6115	P8030	U300
Average Epsilon	0.225806	0.225806	0.129032	0	0.129032	0.290323	0.322581	0.387097
Decision	0	0	0	0	0	0	0	0

Table 7.8: December optimal configuration

We see that this time too only K6101 and K6102 have to be changed.

7.2.3 6th December data set

Finally we run our GAMS program with the hourly basis data set of the 6th of December and we get the following results:

	HP turbines			
Turbines	K3471	**K6101**	**K6102**	K301
Average E	0	**1**	**0.583333**	0.416667
Decision	0	**1**	**1**	0

	AMP turbines				
Turbines	K3262	K3271	K3272	U3200	U3500
Average E	0.375	0.291667	0.291667	0.416667	0.375
Decision	0	0	0	0	0

	MP turbines							
Turbines	K4402	K7101	K8001	others	P3202	P3261	P3262A	P3263
Average E	0.416667	0.375	0.208333	0.416667	0.458333	0	0.375	0.375
Decision	0	0	0	0	0	0	0	0

	MP turbines (2)							
Turbines	P3371	P3501	P3701C	P6102A	P6110	P6115	P8030	U300
Average E	0.375	0.291667	0.25	0	0.416667	0.333333	0.458333	0.25
Decision	0	0	0	0	0	0	0	0

Table 7.9: 6th December optimal configuration

These results, similar to the previous ones with the daily data sets confirm the fact that switching **K6101** and **K6102** to electrical drives constitutes the best option.

7.2.4 Solution

So with the two data sets of March and December 2000 having the same daily basis we can make an average and get the following table:

	HP turbines			
Turbines	K3471	**K6101**	**K6102**	K301
Average Epsilon	0.096774	**0.983871**	**0.677419**	0.370968
Decision	0	**1**	**1**	0

	AMP turbines				
Turbines	K3262	K3271	K3272	U3200	U3500
Average Epsilon	0.225806	0.290323	0.193548	0.209677	0.387097
Decision	0	0	0	0	0

	MP turbines							
Turbines	K4402	K7101	K8001	others	P3202	P3261	P3262A	P3263
Average Epsilon	0.290323	0.274194	0.129032	0.290323	0.177419	0	0.225806	0.225806
Decision	0	0	0	0	0	0	0	0

	MP turbines (2)							
Turbines	P3371	P3501	P3701C	P6102A	P6110	P6115	P8030	U300
Average Epsilon	0.274194	0.177419	0.306452	0	0.112903	0.306452	0.354839	0.403226
Decision	0	0	0	0	0	0	0	0

Table 7.10: Average optimal configuration

The best pattern emerging from our work is so the following one:

Steam turbines **K6101** and **K6102** have to be switched to electrical drives all others turbines remaining on steam use.

Even if in some cases this configuration was not the best to choose, on average only K6101 and K6102 have a percentage widely over 50% with the others percentages never exceed 40% so this new configuration emerges quite clearly as the one to chose. We will thus keep it as it as the best option on average and because we cannot switch on or off turbines or electrical drives depending on the period considered.

7.2.5 Profits

The mean objective functions we get represent the case where each point in time has its own optimal turbine configuration. As we keep only the best option above we have to recalculate the objective function corresponding to this specific configuration for the full data set. We will call this new "objective function" the cost function of the optimum configuration.

We can calculate as well the costs involved before doing any change to the distribution and compare it with the new costs of this new configuration to get the benefits of the optimization. In order to get the cost function of this new configuration (cost of the solution proposed) for each point in time we will set binary variables to 1 for K6101 and K6102 and 0 for the rest of the turbines in a new GAMS program called *solution.gms* and run it. Note that this new program is only using a Non Linear model as integer variables have been fixed. Thus GAMS will proceed to the new mass and energy balances with this fixed distribution and will give the cost of such a configuration for each point in time, especially needed for those times where this solution was not the best one. Here is a table of the average costs of the new distribution for both March and December:

	Mar-00	Dec-00
Objective function (R/year)	207,450,890	240,123,650
Cost function of the new distribution (R/year)	201,511,993	230,877,222
Difference (R/year)	5,938,898	9,246,428
Improvement (Percentage)	**2.86**	**3.85**

Table 7.11: Cost functions of the new distribution

Surprisingly we get even better results with this new turbine distribution (without K6101 and K6102) than results from the optimisation where each point in time has its own best configuration given by GAMS. This means that for some specific points in time our new turbine configuration gives better results than the answer given by GAMS because the solver was unable to give this solution as the best. This is surprising but can be explained by the fact that the solvers we used for our non convex MINLP problem do not rigorously guarantee to find the global optimum. So for these specifics points in time only a local optimality has been achieved. We can nevertheless guess from average values of the decision variables, and check with the above results, that even if we have no rigorous guarantee of our proposed configuration being the global optimum, no others appear to be as good as that one. The full results for March 2000 are given in appendix G. Note that in both cases we do not have any more LP steam vented to atmosphere at the bottom of the steam network. The optimal configuration achieves the following savings compare with the current steam distribution:

	Mar-00	Dec-00
Cost of the new distribution (R/year)	201,511,993	230,877,222
Cost of the current distribution (R/year)	239,847,585	249,315,458
Net savings in Rands per annum	**38,335,592**	**18,438,235**

Table 7.12: Savings per annum achieved

Here is an organogram summarising the method we used to get these results:

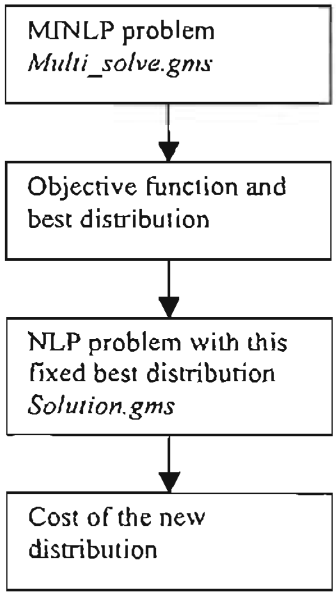


Figure 7.1: Organogram of the method used to get the new cost function

In the table below we have the costs breakdown for the two months:

	March-00		Dec-00	
Total power production of steam turbines (kW)		17 168		18 301
Total steam power to switch to electrical drives (kW)		5 467		5 717
HP steam production (Tons/day)		4 633		5 359
Costs involved	**R/year**	**%**	**R/year**	**%**
Steam production	190 420 000	94,50	220 240 602	95,39
Steam turbines maintenance	256 754	0,13	273 687	0,12
Electricity to run electrical drives	8 970 951	4,45	8 415 479	3,65
Electrical drives maintenance	1 558 138	0,77	1 629 474	0,71
Replacement cost per year	407 903	0,20	425 413	0,18
Salvage value per year	-101 753	-0,05	-107 432	-0,05
Total cost per year	**201 511 993**	100,00	**230 877 222**	100,00

Table 7.13: Costs repartition

So the HP steam production costs are overwhelming all of the other costs, giving 95% of the total cost. The second most important is the electrical cost but this represents only 4% of the total. The 1% left is spread over the remaining costs. This difference comes from the cost of 1kWhr with HP steam, which is approximately R2.22, whereas 1kWhr bought from ESKOM is only R0.174.

The GAMS program is thus trying to reduce steam consumption as much as possible but has to meet steam user demands in the meantime. As explained in a previous section once steam has to be produced it is more profitable to use steam turbines so the extreme solution, one might think about, of switching all turbines to electrical drives, because they are cheaper to run, is avoided. However as far as K6102 is concerned, it is a condensing turbine so switching it to an electrical drive will always be profitable because HP steam can always be saved.

7.2.6 Payback period and Return On Investment (ROI)

The investment done for this new configuration is equal to the cost of replacing K6101 and K6102 by electrical drives minus the salvage value expected on these two turbines. These costs are defined as follows (see section 5.7.2 and 5.7.5):

Replacement costs: $\boxed{CRp_i = 0.019333 * We_i^2 + 2000 * We_i + 505185}$

with CRp_i the capital cost in rands for replacement of steam turbine i, producing W_i kW, with an electrical drive of capacity We_i kW.

Salvage values: $\boxed{Vs_i = 0.1 * (819.2 * Wi^{1.2423}) = 81.92 * Wi^{1.2423}}$

with Vs_i the salvage value (or net realisable value) in Rands for turbine i of size Wi kW.

	March-00	Dec-00
Power switched to electrical (kW)	5 467	5 717
Replacement costs (R)	12 237 081	12 762 376
Salvage values (R)	-3 052 594	-3 222 973
Investments (R)	**9 184 487**	**9 539 403**

Table 7.14: Investments

So we can get the payback period (P) of the project by the following equation:

$$\boxed{P = \frac{\text{Investment}}{\text{Savings per year}}} \tag{7.1}$$

And the return on investment (ROI) by this one:

$$\boxed{ROI = \frac{\left(NS - 0.5 * NS - \frac{I}{D}\right)}{I}} \tag{7.2}$$

with: NS: Net savings per annum in Rands

I: Investments in Rands

D: Project lifetime in years

The 0.5*NS in this equation being the taxation of 50% on benefits.

So we have the following results:

	Mar-00	Dec-00
Investments (R)	9,184,487	9,539,403
Net Savings (R/year)	38,335,592	18,438,235
Project lifetime (years)	30	30
Payback period (months)	**2.87**	**6.21**
ROI (%)	**205**	**93**

Table 7.15: Payback period and ROI

So we can see from these results all the benefits of a conversion to the recommended optimal configuration. Whatever the working conditions of the utility section are, we would get some substantial savings and a payback period in the worst case of approximately 6 months and a corresponding return on invesment around 100 % by using the optimum configuration found. These results depend however on our model parameters such as the cost of HP steam production or the project lifetime.

In the next section we will do a sensitivity analysis of these results as dependent on the model parameters.

7.3 Sensitivity analysis

In order to get the influence of model parameters on the results we will run our GAMS program *multi_solve.gms* with a wide range of these model parameters. As we are interested in relative changes from the starting parameters, working with only one set of data of March 2000 is adequate.

7.3.1 Steam production costs

Indeed this parameter previously chosen equal to R112.6/ton is subject to variation especially with the Rand vs. US$ exchange rate. We shall see what the optimum configuration would be with different values around this figure of R112.6 per ton of HP steam.

Cost of HP steam production (R/tons)	**90**	**100**	**112.6**
HP turbines to be switched off	K6101,K6102	K6101, K6102	K6101, K6102
AP turbines to be switched off	none	none	none
MP turbines to be switched off	P3501	P8030	none
Objective function (R/year)	167,680,614	185,016,950	207,450,890
Cost function of new distribution (R/year)	**163,831,078**	**180,280,581**	**201,511,993**
Cost function of current distribution (R/year)	191,775,603	213,046,392	239,847,585
Net savings per annum (R)	**27,944,526**	**32,765,811**	**38,335,592**
Investment (R)	10,586,668	9,815,145	9,184,487
Payback period (months)	4.55	3.59	2.87
ROI (%)	129	164	205

Cost of HP steam production (R/tons)	120	140
HP turbines to be switched off	K6101, K6102	K6101, K6102
AP turbines to be switched off	none	U3500
MP turbines to be switched off	none	P3371, P3501, P3701C
Objective function (R/year)	220,372,606	251,268,823
Cost function of new distribution (R/year)	**214,026,273**	**249,296,012**
Cost function of current distribution (R/year)	255,587,968	298,129,544
Net savings per annum (R)	**41,561,695**	**48,833,533**
Investment (R)	9,184,487	16,307,413
Payback period (months)	2.65	4.01
ROI (%)	223	146

Table 7.16: HP steam production cost influence

We can observe that as far as the best turbine configuration is concerned, variation in the HP steam production cost does not affect much the main choice of switching K6101 and K6102 to electrical drives until the price of steam becomes so high that it becomes worthwhile to switch more turbines to electrical drives, even on the AMP and MP headers. Note also that we still have this result of the cost of the new configuration being lower than the mean objective function found in the optimisation, because of local optimum problems fixed. Both this new cost function and the cost function of the current distribution increase with the price of producing HP steam. However, savings are growing as well with the price of steam, as reduction in steam production leads to more significant savings.

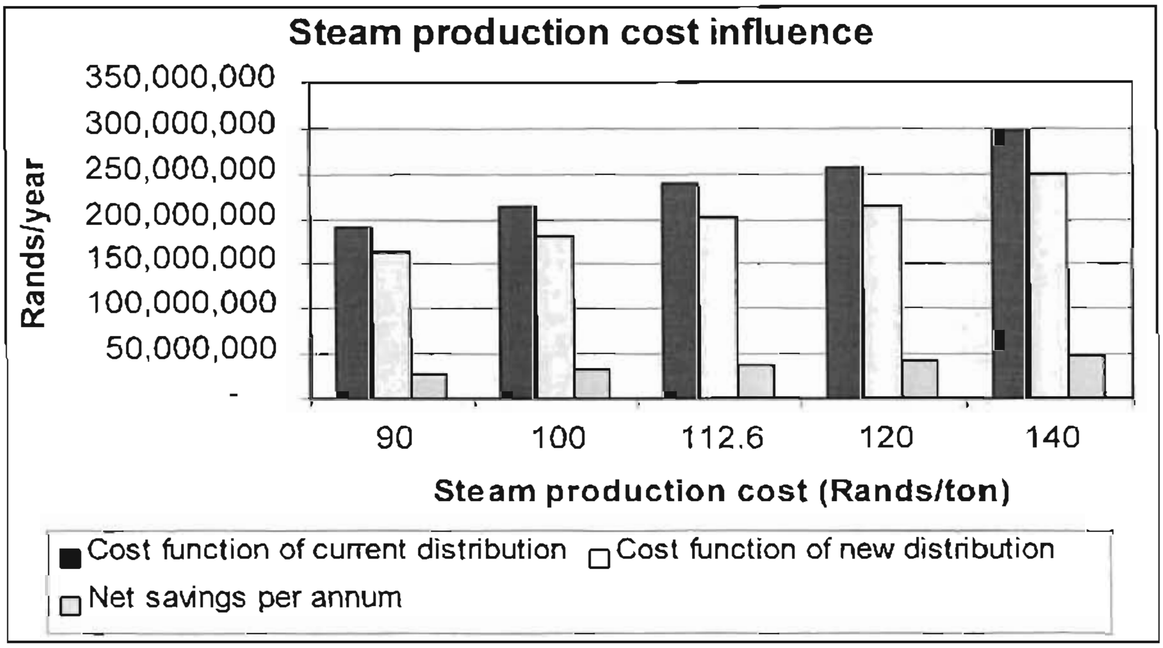


Figure 7.2: HP steam production cost influence

However, as more steam turbines have to be switched to electrical drives for a high steam price, replacement costs rise and give a return on investment a bit lower than for smaller steam prices where less replacement costs are involved.

The main result from this sensitivity analysis on steam production cost is that K6101 and K6102, even if they are not the only ones to be switched off for optimum configurations, always have to be removed.

7.3.2 Electrical costs

This parameter also plays a key role in whether to choose an electrical drive or a steam turbine as it directly gives the cost involved in running electrical drives. So we try a range of values around its default value of 0.174 R/kWh.

Cost of electricity (R/kWh)	**0.1**	**0.15**
HP turbines to be switched off	K6101,K6102	K6101, K6102
AP turbines to be switched off	K3271	none
MP turbines to be switched off	K7101,P3371,P3701C,U300	P3701C
Objective function (R/year)	200,209,145	205,050,169
Cost function of new distribution (R/year)	**198,251,915**	**201,205,849**
Cost function of current distribution (R/year)	239,847,585	239,847,585
Net savings per annum (R)	**41,595,669**	**38,641,735**
Investment (R)	19,748,686	11,417,369
Payback period (months)	5.70	3.55
ROI (%)	**102**	**166**

Cost of electricity (R/kWh)	**0.174**	**0.2**	**0.3**
HP turbines to be switched off	K6101, K6102	K6101, K6102	K6101, K6102
AP turbines to be switched off	none	none	none
MP turbines to be switched off	none	none	none
Objective function (R/year)	207,450,890	208,537,374	207,625,191
Cost function of new distribution (R/year)	**201,511,993**	**202,847,451**	**207,983,829**
Cost function of current distribution (R/year)	239,847,585	239,847,585	239,847,585
Net savings per annum (R)	**38,335,592**	**37,000,134**	**31,863,756**
Investment (R)	9,184,487	9,184,487	9,184,487
Payback period (months)	2.87	2.98	3.46
ROI (%)	**205**	**198**	**170**

Table 7.17 Electrical cost influence

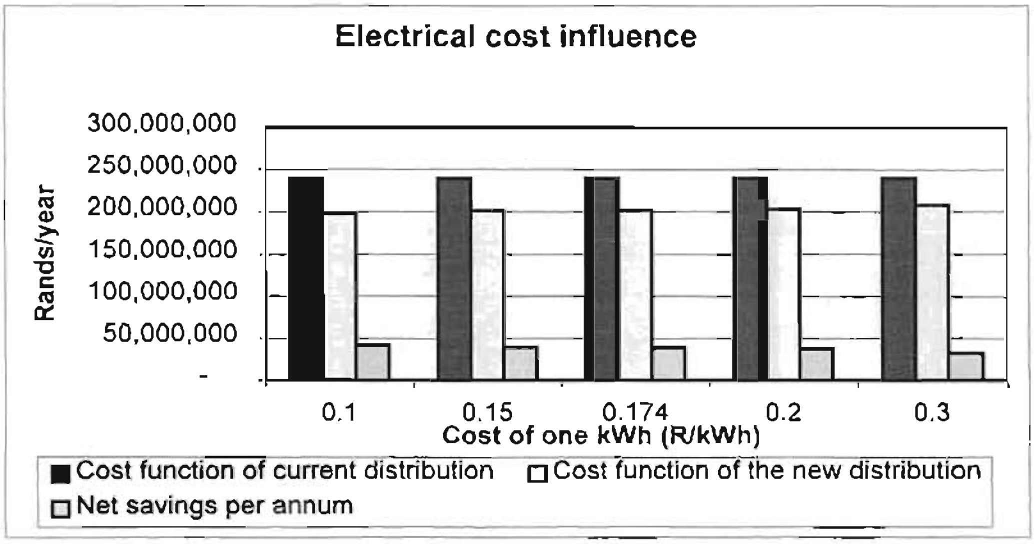


Figure 7.3: Electrical cost influence

We can see that in the case of a rise in the price of one kWh sold by ESKOM, the optimum configuration is not changed. On the other hand as the price decreases, similarly to an increase in the HP steam production cost, it becomes more profitable to use electrical drives as the price of their use becomes more attractive.

The price of electricity has only a slight impact on both the objective function and the cost function of implementing the best configuration. Note that for the specific value of R0.3 per kWh, z is lower because what we have here is only the average objective function from GAMS and not the actual cost function of the new optimum configuration. Previously we have dealt with the case where this actual cost function is lower than the average objective function because of local optimum problems. But here for high values like R0.3 per kWh the optimum configuration consisting of switching only K6101 and K6102 with electrical drives emerges much more often from the GAMS program (ε_{K6101} and ε_{K6102} equal to one in all points in time) so the average objective function is much closer to the actual cost function of the new configuration. The objective function is also slightly under the cost because the new configuration is the best on average and so not always the best for a specific point in time. This means that in that case the solver gives more global optimums.

Note that the cost of electricity has no influence on the cost function of the current distribution as no electrical drives have been used yet. The cost function of the new distribution goes up with the cost of electricity, as we still have to run the two electrical drives in place of steam turbines K6101 and K6102 at a higher cost. This gives decreasing savings as the electricity cost rises. However once again configurations with more steam turbines removed get a lower return on investment because of higher replacement costs. Once again K6101 and K6102 clearly appear as the two turbines to switch to electrical drives.

7.3.3 Project lifetime

Another major model parameter is the choice of the project lifetime period. Based on the expected lifetime of an electrical drive the starting value was 30 years. However one could work with a shorter period, as we cannot guarantee the lifetime of the entire plant, and also because one might ask to see the benefits achieved on a shorter period of time to make sure that the investment is proven to be profitable on a short-term period. Here is a table of results for 10 and 20 year project lifetimes.

project lifetime (years)	10	20	30
HP turbines to be switched off	K6101,K6102	K6101, K6102	K6101, K6102
AP turbines to be switched off	none	none	none
MP turbines to be switched off	none	none	none
Objective function (R/year)	208,832,578	207,083,808	207,450,890
Cost function of new distribution (R/year)	**202,134,054**	**201,667,508**	**201,511,993**
Cost function of current distribution (R/year)	239,847,585	239,847,585	239,847,585
Net savings per annum (R)	**37,713,530**	**38,180,077**	**38,335,592**
Investment (R)	8,413,184	8,413,184	8,413,184
Payback period (months)	2.68	2.64	2.63
ROI (%)	214	222	224

Table 7.18: Project lifetime influence

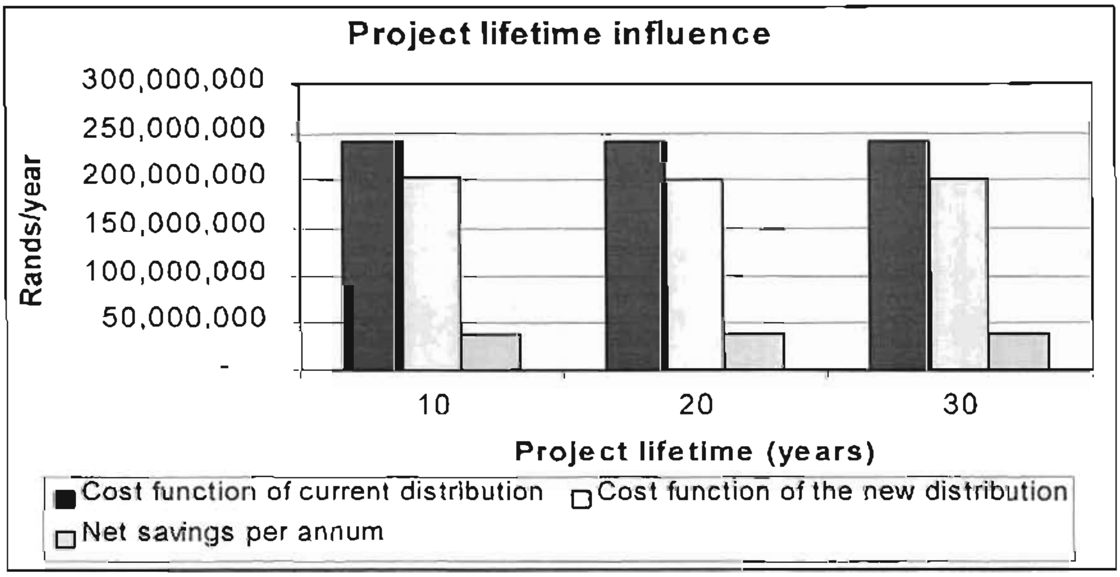


Figure 7.4: Project lifetime influence

So choosing different project lifetimes has little effect on the best turbine configuration emerging from the optimization. Here again the cost function of this best configuration is even lower than the average objective function because of the non-convexity of the problem. Project lifetime does not intervene in calculation of the current distribution cost function, which only involves steam production cost and steam turbines maintenance. So savings rise slightly with the project lifetime increasing, because the same investment is spread over a longer period. Thus return on investment is also a little bit higher for longer project lifetimes.

The interesting result once again being that the same configuration is given as the best to implement in all the cases. This clearly arises because of the dominance of operating costs over capital costs in the ROI calculation.

7.3.4 Generator G3171

Indeed this generator has not been considered for a change as it produces electricity for the refinery and its function is to supply vital utility drives with power in the event of a power cut. However G3171 is currently only producing 0.9MW of power, 18% of its maximum capacity of 5MW. Thus we thought about one way of using G3171 more efficiently for our optimization of the utility system. This would consist in maximising its use and so its steam consumption in our model. This option could be profitable, as producing more electricity with G3171 would reduce the electricity bill of the refinery. However it could involve a substantial rise of the HP steam consumption, unfavourable to the overall objective function.

Changes in G3171 steam consumption may affect steam flows and enthalpies on the AMP header and so indirectly the whole distribution so we have to run our GAMS program again to get the best configuration associated with this change in our model. Our choice is to put the G3171 production as a free variable to maximize in our GAMS program. The idea is to subtract from the objective function the benefit achieved from running G3171 over its current capacity.

We call NP_{G3171} the new power production of G3171 in kW. As the power currently produced by the generator W_{G3171} is already used in the utility system, we can only use the new production minus the current production for the new electrical drives instead of buying all the electricity required for running them. For instance if we produce 3MW with G3171 and its current production is 0.9MW then we can save 3-0.9=2.1 MW of electricity. And if the electrical drives required a total of 5MW we will just have to buy 5-2.1=2.9 MW of electricity to ESKOM. So we add the following equation in our model:

Savings done with a maximum use of G3171, Z_{G3171}, in Rands/year:

$$Z_{G3171} = \left(NP_{G3171} - W_{G3171}\right) * a * 24 * 365 \tag{7.3}$$

with "a" is the cost of electricity in Rands/kWh.

These savings are thus taken into account in the objective function, rewritten as follows:

$$Z = Z_{HPsteam} + Z_{steam} + Z_{elec} - Z_{G3171} + Z_{repl} - Z_{salv} \tag{7.4}$$

Which gives, substituting these terms:

$$Z=365*FH*C_{hp\_steam}+\sum_{i=1}^{N}[(1-\varepsilon_i)*Mst_i]+\sum_{i=1}^{N}[\varepsilon_i*(Celec_i+Melec_i)]-[(NP_{G3171}-W_{G3171})*a*24*365]+\sum_{i=1}^{N}\varepsilon_i*\frac{CRp_i}{D}-\sum_{i=1}^{N}\varepsilon_i*\frac{Vs_i}{D} \quad (7.5)$$

We also have to keep the new power production between the maximum capacity of 5MW and the current power production so we add the following constraints to our model:

$$\begin{cases} NP_{G3171} = F_{G3171}*(H_H - h^{o}_{G3171}) \\ W_{G3171} \le NP_{G3171} \le 5000 \end{cases}$$

We run our new GAMS program, with these changes, called *multi_solveG3171.gms* and get the following results:

Generator G3171	maximized use	current use
HP turbines to be switched off	K6101,K6102	K6101, K6102
AP turbines to be switched off	none	none
MP turbines to be switched off	none	none
Power production of G3171 (MW)	**2.8**	**0.9**
Percentage of maximum capacity of 5 MW	56%	18%
HP steam production (tons/day)	4693	4633
Objective function (R/year)	204,027,668	207,450,890
Cost function of new distribution (R/year)	**201,069,108**	**201,511,993**
Cost function of current distribution (R/year)	239,847,585	239,847,585
Net savings per annum (R)	**38,778,477**	**38,335,592**
Investment (R)	8,413,184	8,413,184
Payback period (months)	2.60	2.63
ROI (%)	227	224

Table 7.19: Maximized use of generator G3171

The best turbine configuration has not been affected by changes in the model. We get a slight improvement of approximately R443, 000 in the net savings achieved per annum.

On one hand we increased the power production of G3171 significantly and benefit approximately 1.9MW leading to a rough R2.86 million saved in the electricity costs per annum. On the other hand, to run the generator at this higher production sending the excess of HP steam (previously letdown through letdown stations) through the generator is not enough and an extra 60 tons/day has to be produced, raising the steam costs of approximately R2.43 million per annum. Indeed it is heavily unfavourable for the overall cost function to increase the HP steam production as we have seen earlier that the steam production costs are overwhelming all the other costs. An extra 1ton/day costs 1*365*Csteam=1*365*112.6= R41, 100 per year but increases the G3171 production by only

$1 * \frac{\Delta H_{G3171}}{86.4} = 1 * \frac{127}{86.4} = 1.47$ kW. Meaning a saving of 1.47*24*365*a=1.47*24.365*0.174=R2, 240 per year in the electrical costs.

What is happening in the new distribution is that all the HP steam previously going through the HP-AMP letdown stations (approximately 1200 tons/day) is now sent through G3171 and then produces an extra 1.7MW (so roughly R2.7 million saved). This solution would have been much better if this change would have been the only one. However the steam enthalpy at the generator outlet is much lower than the one at the letdown stations outlets, making the whole AMP header enthalpy lower than before. So steam flows have to be greater to match the energy requirements and so the HP steam production has to increase by this average of 60 tons/day making this solution of running G3171 at a higher capacity much less profitable.

It is however still more profitable to run G3171 at this higher production rate of 2.8MW corresponding to 56% of its maximum capacity than the current 0.9MW that does not take enough advantage of it. So this solution has to be considered as it also keeps G3171 in better readiness to supply vital utility drives with power in the event of a power cut.

7.3.5 Compressor K6101

After having presented our first results to the refinery engineers and staff in charge of the utility section it appeared that removing K6101 (compressor) and installing an electrical drive instead represents a critical risk to the refinery. Indeed, if it suddenly stops (power failure, etc...) they would lose instrument air and could not control anything in the refinery.

To avoid this risk we will see what the optimum distribution would be if K6101 is not available for replacement by an electrical drive anymore. In order to have this in our model we set ε_{K6101} to 0 and in doing so we force it to remain on steam use. Working with our starting values of our model parameters we get:

K6101 decision	available for a change	not available for a change
HP turbines to be switched off	K6101, K6102	K6102
AP turbines to be switched off	none	none
MP turbines to be switched off	none	none
Objective function (R/year)	207,450,890	216,755,672
Cost function of new distribution (R/year)	**201,511,993**	**215,514,149**
Cost function of current distribution (R/year)	239,847,585	239,847,585
Net savings per annum (R)	**38,335,592**	**24,333,435**
Investment (R)	8,413,184	6,899,837
Payback period (months)	2.63	3.40
ROI (%)	224	173

Table 7.20: K6101 not available for replacement

So we can see that the solution given is to replace only K6102 now that K6101 is not anymore considered for replacement. The benefits are obviously lower than before but still significant and still make the optimization work profitable. This is due to the higher HP steam production required when running K6101 with steam. As only the replacement of K6102 is involved, investments are a bit lower but still lead to a very short payback period and a very high return on investment.

In order to get the best from our optimization work we want to see now what the results would be by considering the maximized use of the generator G3171 coupled with keeping K6101 on steam use.
Here are the results of such a model run with GAMS:

K6101 not available for replacement		
Generator G3171	**maximized use**	**current use**
HP turbines to be switched off	K6102, K3471	K6102
AP turbines to be switched off	U3200	none
MP turbines to be switched off	none	none
Power production of G3171 (MW)	**1.7**	**0.9**
Percentage of maximum capacity of 5 MW	34%	18%
HP steam production (tons/day)	5057	5116
Objective function (R/year)	216,607,650	216,755,672
Cost function of new distribution (R/year)	**215,187,162**	**215,514,149**
Cost function of current distribution (R/year)	239,847,585	239,847,585
Net savings per annum (R)	**24,660,423**	**24,333,435**
Investment (R)	7,969,195	6,899,837
Payback period (months)	3.88	3.40
ROI (%)	**151**	**173**

Table 7.21: Maximized use of generator G3171 when K6101 is not available for replacement

We see that maximizing G3171 use influences the best turbines configuration in that it leads to the removal of two others turbines, K3471 and U3200. The idea behind that choice is to reduce the very expensive HP steam production as much as possible. So removing K3471 on the HP header gives the opportunity to reduce that production and also to remove a turbine on the AMP header, U3200. As far as cost functions are concerned they are quite similar for the following reasons:

In maximizing G3171 production we achieved a saving of 59tons/day in the HP steam production, meaning a saving of R2.4 million per annum. Note that in that case no problems of energy requirements are encountered, as steam is not taken from a letdown station to G3171 but from K3471 so its quality on the AMP header is similar. We got an increase of 0.8MW in the power production of G3171, leading to a R1.2 million per annum save in the electricity bill.

However, this electricity bill is noticeably increasing due to the use of two more electrical drives in place of K3471 and U3200. K3471 and U3200 producing respectively 760 and 960 kW that raise the electricity costs of roughly R1.2 and R1.5 million per annum. We add to that R0.5 million per annum in the maintenance costs of these electrical drives and their replacement costs of R4.4 million spread over a project lifetime of 30 years (R265, 000 per annum). These replacements also involve savings of R400, 000 because of their salvage values and also negligible decrease in the steam turbine maintenance costs.

So we save approximately R3.7 million per annum compare to the previous case but we spend roughly R3.4 million per annum to run this new configuration. That is why we only get approximately R330, 000 per annum of savings compare with the previous case where G3171 was only used at 18% of its maximum capacity. As this new solution involves more replacements and thus a higher investment for similar benefits, it has a slightly lower return on investment.

This solution, of improving generator G3171 use, constitutes though the best one in the case where K6101 is not available for replacement. Note that in case of a power failure, this generator would be use to drive essential drives, which in many cases will not include the new electrical drives that have to be installed.

In this chapter we have finally come to the results of our optimization problem solved with our GAMS program, where best algorithm options have been chosen.

An optimum configuration emerged where only two turbines have to be switched to electrical drives, K6101 and K6102. We then proceeded to a sensitivity analysis to see the influence on the results of changes in the model parameters. Finally we studied the cases where an improved use of the generator G3171 is implemented and where K6101 is not available for a change because it might not be safe to remove it.

In all the cases savings achieved by the optimum configuration found by the solver are substantial and make the optimisation profitable.

We will now move on to the conclusions and recommendations arising from this study.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

An MINLP model that incorporates rigorous equations has been proposed for the analysis and optimization of the SAPREF oil refinery steam distribution for given utility demands and operating parameters. The model predicts the optimal replacement pattern for changing steam turbines to electrical drives, and the optimal operating conditions, such as, flowrates and enthalpies.

Several correlations have been derived to account for purchase, salvage and operating costs of steam turbines and electric motors. The work done has shown the capabilities of the model to select between steam turbines or electrical drives and the best arrangement of different kinds of steam turbines and electrical drives for a given power demand.

The model has been implemented in the user-friendly optimization package GAMS. Recent increased interest in the development and application of MINLP optimisation algorithms make it feasible to solve such problems, even of significant size. GAMS incorporates the latest algorithmic methods of optimization. The profits achieved, and the modest computational effort using DICOPT++ as the MINLP solver of this large model, show all of the potential of such an optimization tool for industrial purposes.

A sensitivity analysis of model parameters has illustrated the flexibility of the model and the possibility to adapt it to other flowsheet optimization and synthesis problems.

Finally, although none of the recent methods can yet give a theoretical guarantee that the global optimum is reached in nonconvex problems, the results achieved show that the proposed method has a high degree of reliability for finding an optimum, making the optimization work worthwhile.

8.2 Recommendations

As far as the results for the specific SAPREF distribution are concerned, we were satisfied to see that in most of the cases one optimal distribution is emerging even if operating parameters move around their current values, proof of a reliable and steady solution. These parameters, generally, only affect the costs involved and not the optimal steam distribution routing to implement at the refinery.

This configuration consists in replacing the two HP turbines K6101 and K6102. Benefits achieved are significant, even if they have to be considered with caution, because they might not represent the actual costs due to approximations and extrapolations. Compared with the current costs, they still make the optimisation work worthwhile, over a large range of model parameters and working conditions. Another achievement is that in nearly all of the best steam distributions given for the different cases studied, we eliminate LP steam venting to atmosphere on the LP header. Average savings, depending on operating conditions of the period of time considered, amount to about 1, 000 tons/day HP steam (at the cost of additional electricity of course).

The net savings per annum calculated are always greater than the required investment, leading to a payback period shorter than one year. The current working conditions and model parameters give in the worst case an investment of R9.5 million for net savings of R18.4 million per annum (December 2000 data set) and an investment of R9.2 million for net savings of R38.3 million per annum in the best case (March 2000 data set). Electrical drives are sized on the exact power required at any instant.

The contribution made to the economics by the salvage value is small enough to make the solution of keeping the steam turbine on standby, should the situation change or the electricity fail, worth considering. However, to keep a whole turbine and lubrication system ready to run at the flick of a switch is likely to cost a lot in maintenance and there is also the problem of mechanical access to the drive-shalf which means equipments like compressors and pumps would probably have to be duplicated.

Finally, using the March 2000 data set, we studied cases where the use of generator G3171 is maximized and where K6101 is not available for replacement for reasons of plant integrity. We came to the conclusion that generator G3171 power production has to be increased to improve the savings whatever the decision on K6101 is. However if K6101 is kept on steam use, allowing G3171 to select its own operating point (instead of being fixed at 0.9MW of its 5MW range) leads to some changes in the optimal turbine configuration where two more turbines have to be replaced (K3471 and U3200). In this last case benefits are only slightly increased, approximately by R330, 000 per annum, for an increase in investment of R1 million, making the choice of a free G3171 more questionable.

As far as K6101 is concerned, the choice is given to the refinery managers whether to replace it or not considering the integrity issue. If they are forced to keep it on steam use, benefits are then reduced by 36%, compare with the case where we could replace it (R24.3 million per annum against R38.3 million per annum). The payback period and return on investment still make the optimization work highly profitable. Replacing only K6102 with an electrical drive leads to a drop in the investment from R8.4 million to R6.9 million.

All these results are summarized in the following table:

		Turbines to be switched	Investment (Rands)	Net savings per annum (Rands)
K6101 available for a change	Dec 2000	K6101, K6102	9, 539, 403	18, 438, 235
	March 2000	K6101, K6102	9, 184, 487	38, 335, 592
	March 2000 and maximised use of G3171	K6101, K6102	8, 413, 184	38, 778, 477
K6101 not available for a change	March 2000	K6102	6, 899, 837	24, 333, 435
	March 2000 and maximised use of G3171	K6102, K3471, U3200	7, 969, 195	24, 660, 423

Table 8.1: Summarized results

References

Adjiman C. S., Androulakis I. P. and Floudas C. A. (1997). Global optimization of MINLP problems in process synthesis and design. *Computers & Chemical Engineering*, **21**, S445-S450.

Babcock and Wilcox Co. (1992). Steam: Its generation and uses. *Barbeton, Ohio.*

Bhatia T.K. and Biegler L.T. (1999). Multiperiod design and planning with interior point methods. *Computers & Chemical Engineering*, **23**, 919-932.

Brooke A., Kendrick D., Meeraus A. and Raman R. (1998). GAMS: A User's Guide. GAMS Development Corporation, Washington, USA.

Bruno J.C., Fernandez F., Castells F. and Grossmann I.E. (1998). A rigourous minlp model for the optimal synthesis and operation of utility plants. *Trans IchemE*, **76**, part A.

Chetty J. and Majola T. (2000). Optimisation of steam consumption. *Report from the vacation work 1999/2000.*

Chou C.C. and Shih Y.S. (1987). A thermodynamic approach to the design and synthesis of plant utility systems. *Ind Eng Chem Res*, **26**, 1100.

Dewar I. (1999). Opportunities for optimising utilities in major refineries. *PTQ, summer 1999.*

DholeV.R. and Linnhoff B. (1993). Total Site Targets for Fuel, Cogeneration, Emissions, and Cooling. *Computers & Chemical Engineering*, **17**, S101-S109.

Diaz M. and Bandoni J.A. (1996). A mixed integer optimization strategy for a large chemical plant in operation. *Computers & Chemical Engineering*, **20**, 531.

Dolan W.B., Cummings P.T. and Levan M.D. (1989). Process optimization via simulated annealing: Application to network design. *AIChE Journal*, **35** (5), 725-736.

Duran M.A. and Grossmann I.E. (1986). An Outer Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Math. Programming*, **36**, 307-339.

Duran M.A. and Grossmann I.E. (1986). A mixed integer nonlinear programming approach for process systems synthesis. *AIChE Journal*, 32(4), 592-606.

Fletcher R. and Leyffer S. (1994). Solving Mixed integer nonlinear programs by Outer Approximation. *Math. Programming*, **66**, 327.

Floudas C.A., Aggarwal A. and Ciric A.R. (1989). Global optimum search for non convex NLP and MINLP problems. *Computers & Chemical Engineering*, **13**, 1117-1132.

Gaines L.D. and Gaddy J.L. (1976). Process optimization by flow sheet optimization. *Ind. Eng. Chem. Process Des. Dev.*, 15, 206.

GAMS Development Corporation, Washington, USA (1996). Solvers manual. *www.gams.com*.

Geoffrion A.M. (1972). Generalised Benders Decomposition. *Journal of optimization theory and applications*, **10**, 237-260.

Goodall P.M. (1970). The efficient use of steam. *Westbury House.*

Grossmann I.E. and Kravanja Z. (1995). Mixed-integer nonlinear programming techniques for process systems engineering. *Computers & Chemical Engineering*, **19**, *suppl.*

Grossmann I.E. (1985). Mixed-Integer Programming approach for the synthesis of integrated process flowsheets. *Computers & Chemical Engineering*, **9**, 463.

Grossmann I.E. (1996). Mixed-integer optimization techniques for algorithmic process synthesis. *Adv. Chem. Eng.*, **23**, 171.

Gupta (1980). Branch-and-Bound Experiments in Nonlinear Integer Programming. *Ph.D. Thesis Sch. Ind. Eng., Purdue Univ., West Lafayette.*

Hostrup M., Gani R., Kravanja Z., Sorsak A. and Grossmann I.E. (2001). Integration of thermodynamic insights and MINLP optimization for the synthesis, design and analysis of process flowsheets. *Computers & Chemical engineering*, **25**, 73-83.

Iyer R.R. and Grossmann I.E. (1997). Optimal Multiperiod Operational Planning for Utility Systems. *Computers & Chemical Engineering*, **21**, no.8, 787-800.

Iyer R.R. and Grossmann I.E. (1998). A bilevel decomposition algorithm for long-range planning of process networks. *Ind. Eng. Chem. Res.*, 37, no.2, 474-481.

Kocis G.R.and Grossmann I.E. (1987). Relaxation Strategy for the structural optimization of process flow sheets. *Ind. Eng. Chem. Res.*, **26**, 1869-1880.

Kocis G.R. and Grossmann I.E. (1989). A modeling and decomposition strategy for MINLP optimization of process flowsheets. *Computers & Chemical Engineering*, **13**, 797-819.

Kocis G.R. and Grossmann I.E. (1989). Computational Experience with DICOPT solving MINLP problems in Process Systems Engineering. *Computers & Chemical Engineering*. **13**, 307-315.

Kovac A. and Glavic P. (1995). Retrofit of complex and energy intensive processes-I. *Computers & Chemical Engineering*, **19**, 1255.

Kravanja Z. and Grossman I.E. (1990). Prosyn-a MINLP process synthesizer. *Computers & Chemical Engineering*, **14**,1363-1373.

Kravanja Z. and Grossmann I.E. (1994). New developments and capabilities in PROSYN – An automated topology and parameter Process Synthesizer. *Computers & Chemical Engineering*, **18**, 1097-1114.

Marechal F. and Kalitventzeff B. (1997). Identify the optimal pressure levels in steam networks using integrated combined heat and power method. *Chemical Engineering science*, **52**, no 17, 2977-2989.

Marechal F. and Kalitventzeff B. (1999). Targeting the optimal integration of steam networks: Mathematical tools and methodology. *Computers & Chemical Engineering*, **Supplement**, 133-136.

Matas-Valiente P. (2000). Benefit from Steam-balance optimization strategies. *Chemical Engineering*, Nov 2000, 70-74.

Mavromatis S.P. and Kokossis A.C. (1998). Conceptual optimisation of utility networks for operational variation. *Chemical Engineering science*, **53**, no 8.

Pahor P. and Kravanja Z. (1995). Simultaneous solution and minlp synthesis of DAE process problems: PFR networks in overall processes. *Computers & Chemical Engineering*, **19**, suppl.

Papalexandri K. P., Pistikopoulos E. N. and Kalitventzeff B. (1998). Modelling and optimization aspects in energy management and plant operation with variable energy demands-application to industrial problems. *Computers & Chemical Engineering*, **22**, 1319-1333.

Papalexandri K. P., Pistikopoulos E. N., Kalitventzeff B., Dumont M. N., Urmann K. and Gorschluter J. (1996). Operation of a steam production network with variable demands modelling and optimization under uncertainty. *Computers & Chemical Engineering,* **20**, S763-S768.

Papalexandri K.P. and Pistikopoulos E.N. (1994). A multiperiod MINLP model for the synthesis of flexible Heat and mass exchange Networks. *Computers & Chemical Engineering,* **18**, no. 11/12, 1125-1139.

Peterson J.F. and Mann W.L. (1985). Steam system design: how it evolves. *Chemical Engineering,* October 14.

Petroulas T. and Reklaitis G.V. (1984). Computeraided synthesis and design of plant utility systems. *AIChE Journal,* **30**(1), 69.

Petterson F. and Westerlund T. (1997). Global optimization of pump configurations using binary separable programming. *Computers & Chemical Engineering,* **21**, no 5.

Raman R. and Grossmann I.E. (1993). Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Computers & Chemical Engineering,* **17**, 909-927.

Raman R. and Grossmann I.E. (1994). Modeling and computational techniques for logic based integer programming. *Computers & Chemical Engineering,* **18**, 563-578.

Ryoo H.S. and Sahindis N.V. (1995). Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers & Chemical Engineering,* **19**, no 5.

Salcedo R.L. (1992). Solving Nonconvex Nonlinear Programming and Mixed-Integer Nonlinear Programming Problems with Adaptive Random Search. *Ind.Eng.Chem.Res.* **31**, no 1.

Skrifvars H., Harjunkoski I., Westserlund T., Kravanja Z. and Porn R. (1996). Comparison of different minlp methods applied on certain chemical engineering problems. *Computers & Chemical Engineering,* **20**, suppl.

Smith E. M. B. and Pantelides C. (1997). Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering,* **21**, S791-S796.

Ternet D.J. and Biegler L.T. (1999). Interior-point methods for reduced Hessian successive quadratic programming. *Computers & Chemical Engineering,* **23**, 859-873.

Turkay M. and Grossmann I.E. (1996). Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs- multiple size regions. *Ind. Eng. Chem. Res.*, 35, 2611-2623.

Turkay M. and Grossmann I.E. (1998). Structural flowsheet optimization with complex investment cost functions. *Computers & Chemical Engineering,* 22, no 4/5.

Van den Heever S.A. and Grossmann I.E. (1998). Disjunctive multiperiod optimization models for process networks. *FOCAPO Conference procedings,* July 1998.

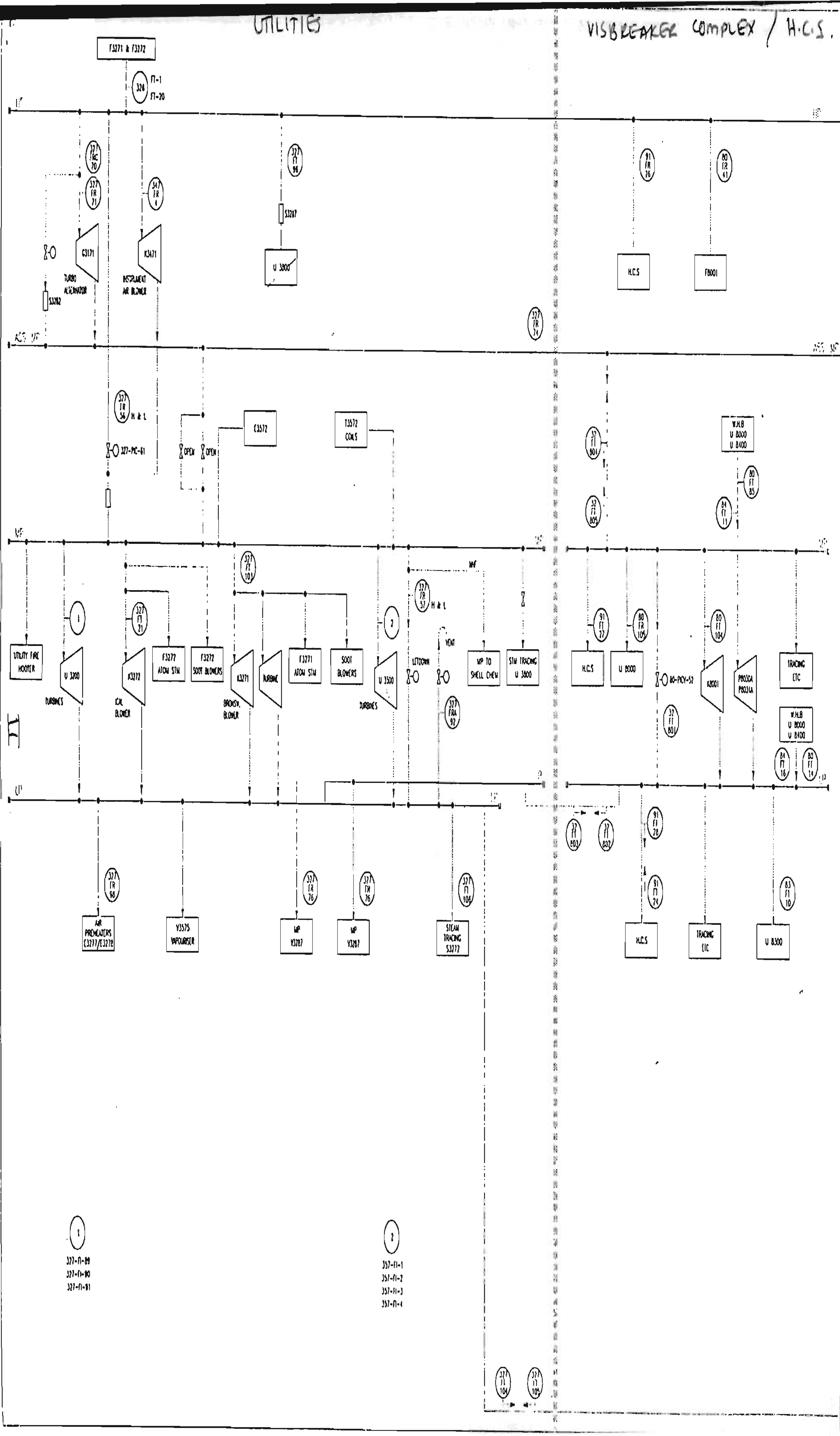
Varvarezos D.K., Grossmann I.E., and Biegler L.T. (1992). An outer-approximation method for multiperiod design optimization. *Ind. Eng. Chem. Res.*, 31, no 6, 1466-1477.

Viswanathan J. and Grossmann I. E. (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering,* 14, 769-782.

Westerlund T., Pettersson F. and Grossmann I.E. (1994). Optimization of pump configurations as a minlp problem. *Computers & Chemical Engineering,* **18**, no 9.

APPENDIX A

Steam distribution flowsheet



KEY PLAN

GENERAL NOTES

1. This drawing must not be lent, copied or reproduced in any way without the permission of SHELL & BP of South African Petroleum Refineries (Pty.) Ltd

Rev.	By	Date	Description	
C	A.F.A	9/3/91	REVISED & UPDATED ON CAD	400HGR

Chk'd | Appr.

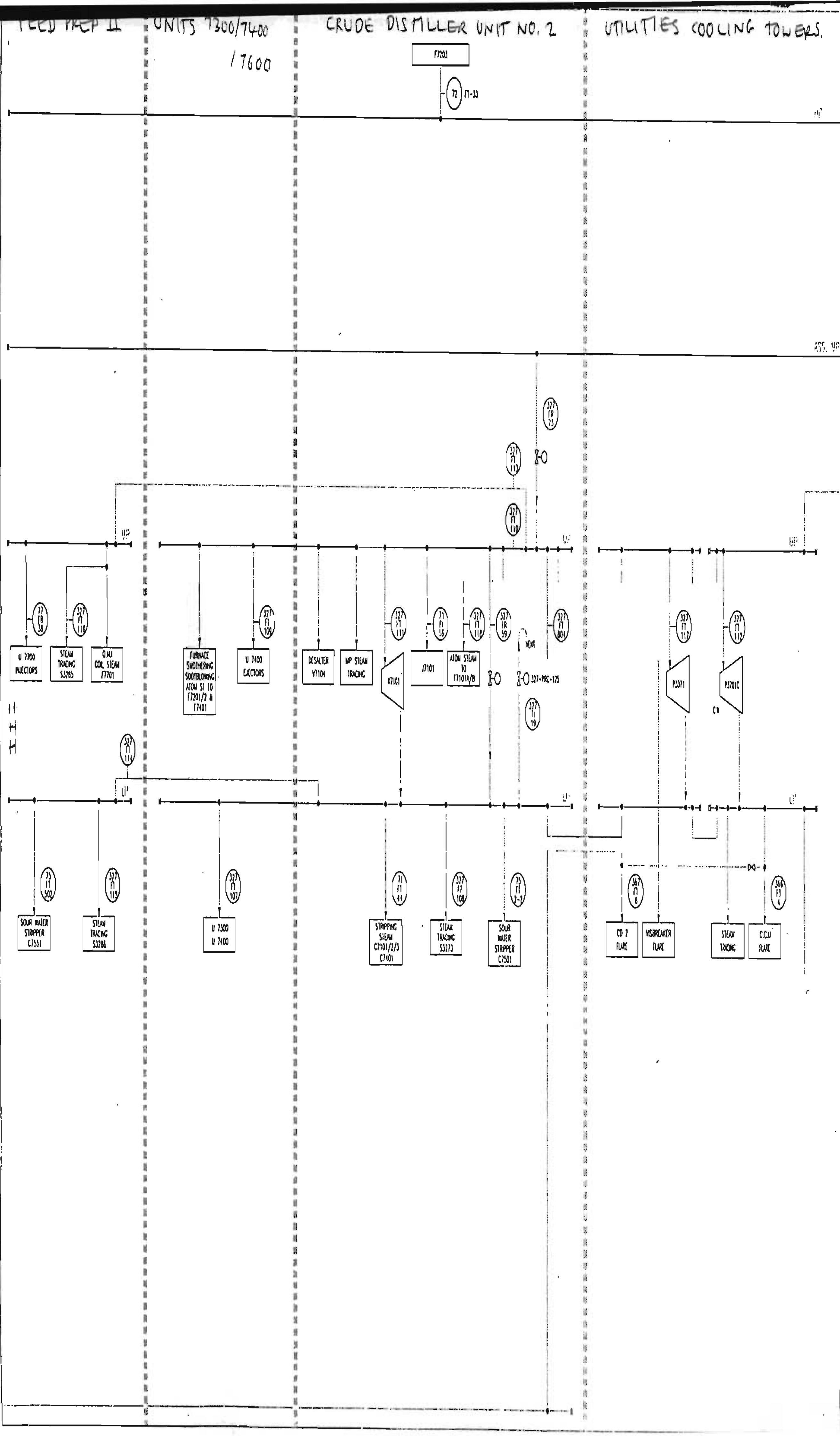
Proj. No.

SHELL & BP SOUTH AFRICAN PETROLEUM REFINERIES (PTY.) LTD. DURBAN SOUTH AFRICA

FLOW METERING
OF STEAM
FLOW SCHEME

Drawn	Date	Scale	Checked
R.EDDEN	06/06/89	1:1	JC
APPROVED			

Drg. No. C4-5000-14 Sht. 1 of 5



KEY PLAN

GENERAL NOTES

1. This drawing must not be lent, copied or reproduced in any way without the permission of SHELL & BP of South African Petroleum Refineries (Pty.) Ltd

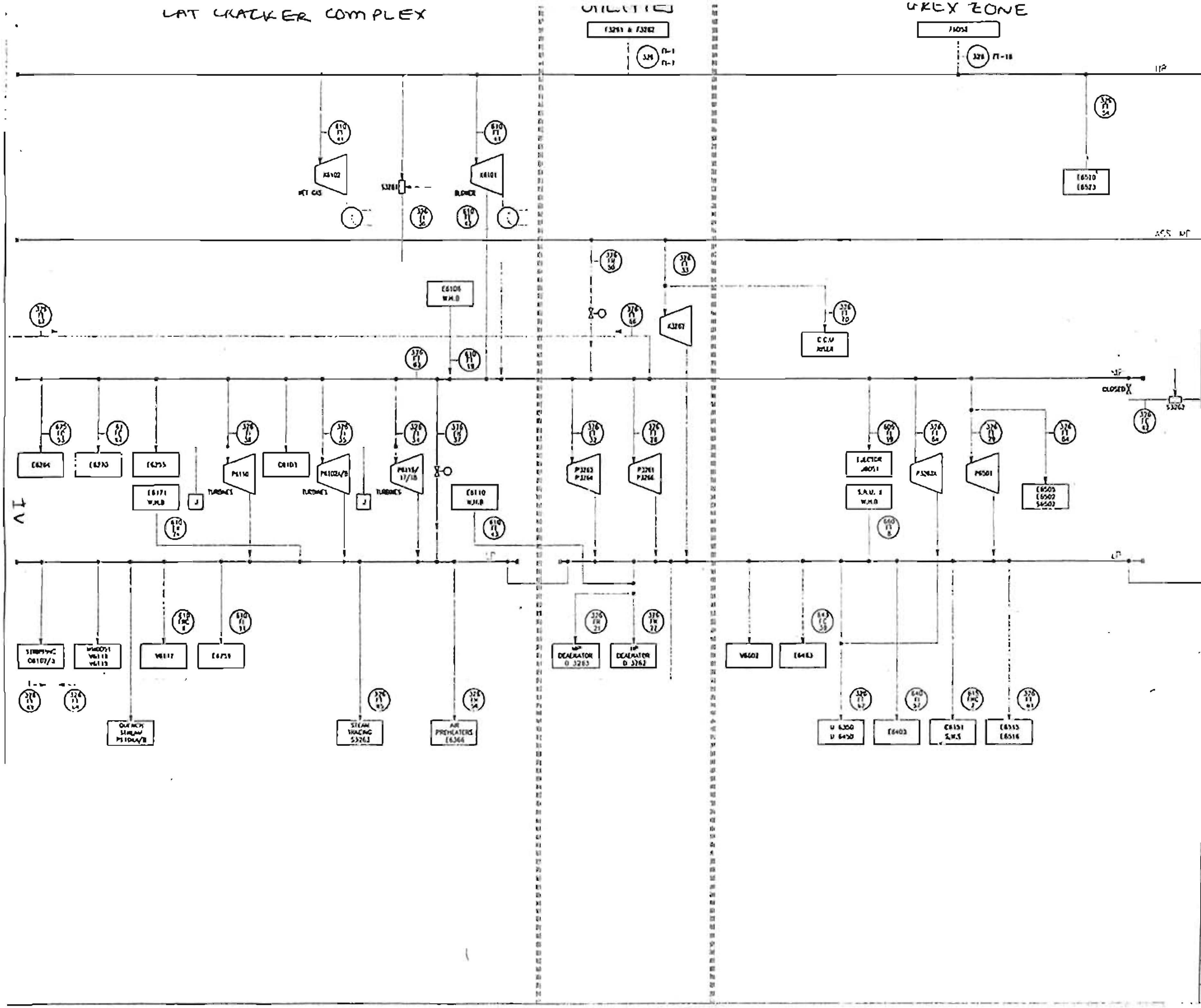
Rev.	By	Date	Description	
C	A.E.A	9/3/?	REVISED & UPDATED ON CAD	ADUKCR A
				Chk'd Appvd
				Proj. No.

SHELL & BP SOUTH AFRICAN PETROLEUM REFINERIES (PTY.) LTD. DURBAN SOUTH AFRICA

ZONE METERING OF STEAM FLOW SCHEME

Drawn	Date	Scale	Checked
R.LODEN	06/06/89	1:1	JC
APPROVED			

Drg. No. 04-5000-14 Sht. 2 of 5.



KEY PLAN

GENERAL NOTES

1. This drawing must not be lent, copied or reproduced in any way without the permission of SHELL & BP of South African Petroleum Refineries (Pty.) Ltd

C	A.T.A		REVISED & UPDATED ON CAD	100RCK A
Rev	By	Date	Description	Chk'd / Appr'd
				Proj. No.

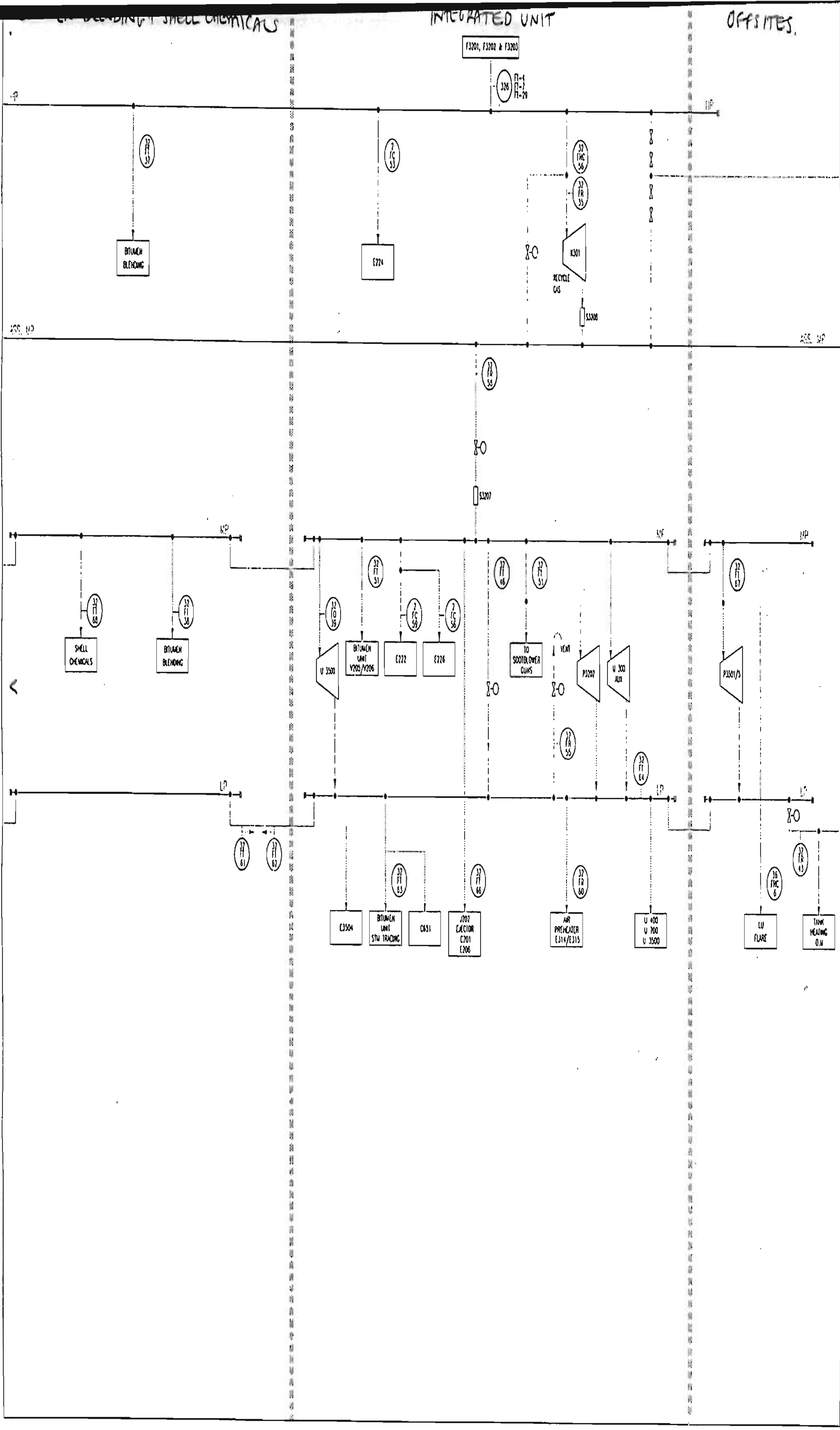
SHELL & BP SOUTH AFRICAN PETROLEUM REFINERIES (PTY.) LTD. DURBAN SOUTH AFRICA

ZONE METERING OF STEAM FLOW SCHEME

Drawn	Date	Scale	Checked
R.EBDEN	06/05/89	1:1	J C

APPROVED

Drg. No. 04-5000-14 Sht. 3 of 5.



KEY PLAN

GENERAL NOTES

1. This drawing must not be lent, copied or reproduced in any way without the permission of SHELL & BP of South African Petroleum Refineries (Pty.) Ltd

Rev.	By	Date	Description	Chk'd/Appvd
C	A.C.A		REVISED & UPDATED ON CAD	400RGR

Proj. No.

SHELL & BP SOUTH AFRICAN PETROLEUM REFINERIES (PTY.) LTD. DURBAN SOUTH AFRICA

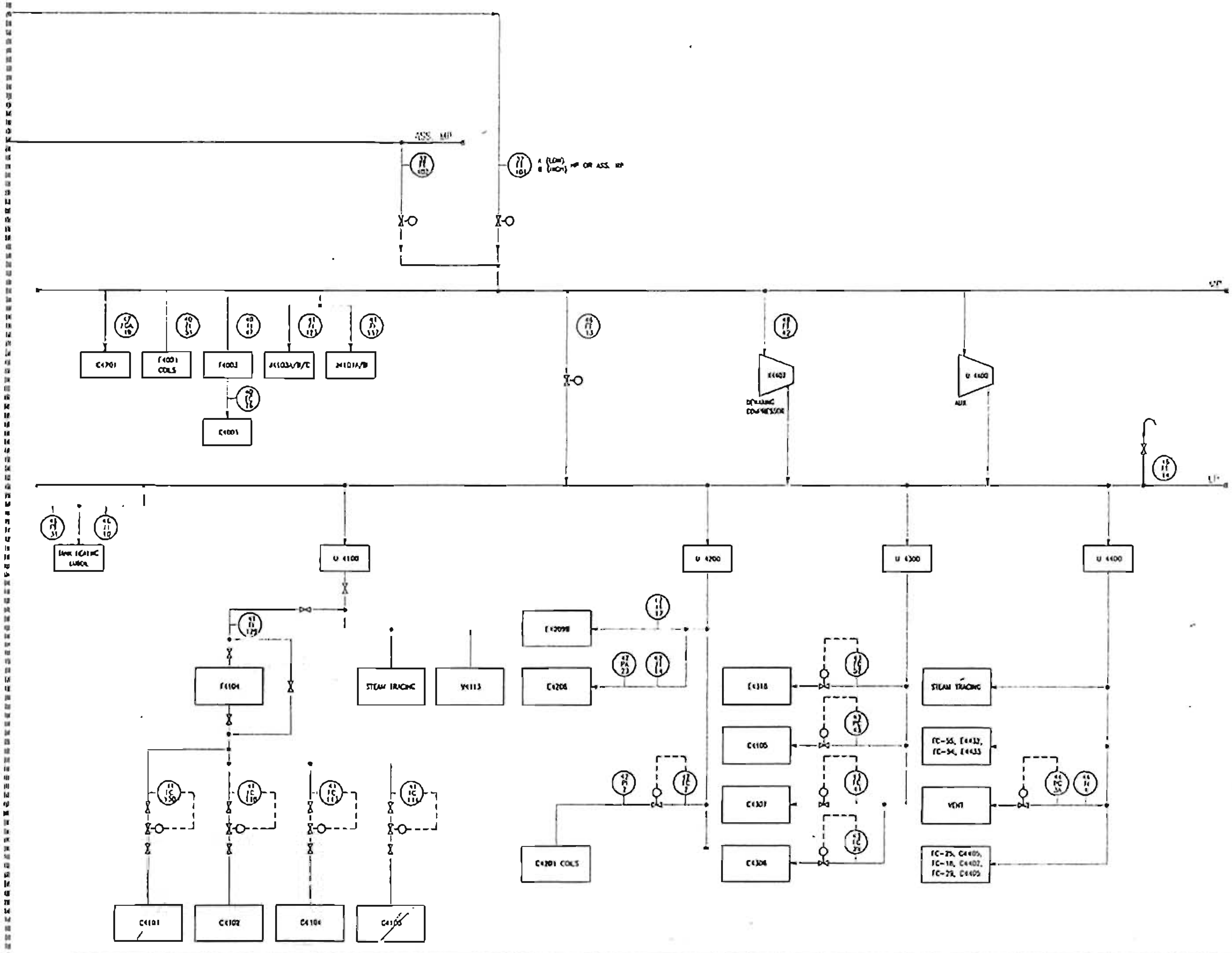
ZONE METERING
OF STEAM
FLOW SCHEME

Drawn	Date	Scale	Checked
R.EDDEN	06/06/89	1:1	J.C

APPROVED

Drg. No. 04-5000-14 Sht. 4 of 5

LUBOLL



KEY PLAN

GENERAL NOTES

1. This drawing must not be lent, copied or reproduced in any way without the permission of SHELL & BP of South African Petroleum Refineries (Pty.) Ltd

Rev.	By	Date	Description	Chk'd/Appvd
C	ACA		REVISED & UPDATED ON CAD	DONCK A

Proj. No.

SHELL & BP SOUTH AFRICAN PETROLEUM REFINERIES (PTY.) LTD. DURBAN SOUTH AFRICA

ZONE METERING OF STEAM FLOW SCHEME

Drawn	Date	Scale	Checked
R.EDDEN	06/06/89	1:1	J.C
APPROVED			

Drg. No. 0-1 5000-14 Sht. 5 of 5

VI

APPENDIX B

Cost correlation and table for VSD installation

								Costs (R) involved for desired size (MW)					
		reference points						1	2	3	4	5	6
Work and equipment		MW	R	MW	R	MW	R						
Demolition	Remove Turbine & Demolish Bed	1.00	4000	6.00	8000			4000	4800	5600	6400	7200	8000
New Civils	150m Trench @ R120/m	1.00	18000	6.00	18000			18000	18000	18000	18000	18000	18000
	Bed for new Motor	1.00	15000	6.00	45000			15000	21000	27000	33000	39000	45000
Switchgear	1 Panel HV (all as double-busbar)	1.00	120000	6.00	120000			120000	120000	120000	120000	120000	120000
Motor	Motor	1.00	280000	2.00	370000	7.00	1400000	280000	369999	498664	665995	871992	1116655
Installation	Per cable Termination	1.00	18000	6.00	18000			18000	18000	18000	18000	18000	18000
Cables	150m For 1MW : 70mm2 @ R200/m For 4MW : 300mm2 @ R700/m	1.00	14000	4.00	210000			14000	79333	144667	210000	275333	340667
VSD	VSD	0.70	450000	4.00	3000000			681818	1454545	2227273	3000000	3772727	4545455
Manpower	(Hours: [1MW ; 6MW]) Engineer @ R150/h [600 ; 720] Technician @ R100/h [900 ; 1080] Artisan @ R90/h [900 ; 1080]	1.00	261000	6.00	313200			261000	271440	281880	292320	302760	313200
Infrastructure	Additional Site Infrastructure due to bottleknecks	1.00	1000000	5.00	5000000			1000000	2000000	3000000	4000000	5000000	6000000
TOTAL								**2411818**	**4357118**	**6341083**	**8363715**	**10425013**	**12524976**

APPENDIX C.1

Overall steam balance of March 2000 – HP header

	HP feeders		
	Total HP produced		
	FH	average T	average P
TIME	*Tons/day*	*Degres C*	*kPa*
01-Mar-00	5861.40	397.90	5103.20
02-Mar-00	5906.65	398.58	5105.88
03-Mar-00	6251.21	397.09	5089.79
04-Mar-00	6086.15	397.18	5095.13
05-Mar-00	5858.38	394.87	5114.82
06-Mar-00	5944.08	394.64	5109.48
07-Mar-00	5866.08	397.09	5098.37
08-Mar-00	5975.59	398.34	5098.98
09-Mar-00	5739.57	396.50	5064.44
10-Mar-00	5998.25	399.65	5089.10
11-Mar-00	5888.14	401.25	5091.84
12-Mar-00	5645.17	402.99	5086.82
13-Mar-00	5760.23	399.78	5121.83
14-Mar-00	5550.63	397.48	5122.32
15-Mar-00	5796.30	400.06	5118.61
16-Mar-00	5821.98	399.63	5123.55
17-Mar-00	6058.76	402.86	5098.90
18-Mar-00	5931.54	401.96	5084.82
19-Mar-00	5739.77	399.30	5124.52
20-Mar-00	5753.47	386.36	5121.15
21-Mar-00	5687.71	404.58	5108.30
22-Mar-00	5786.36	402.36	5112.36
23-Mar-00	5719.15	401.34	5118.75
24-Mar-00	5697.67	398.99	5122.01
25-Mar-00	5494.83	399.77	5123.24
26-Mar-00	5890.29	401.82	5112.06
27-Mar-00	5817.70	402.14	5119.18
28-Mar-00	5822.92	401.96	5282.47
29-Mar-00	5720.47	402.10	5107.16
30-Mar-00	5794.86	402.70	5095.04
31-Mar-00	5790.70	401.32	5106.20
average	**5827.61**	**399.44**	**5111.95**

Table C.1.1: HP feeders

HP consumed					
turbines					
HP to Ass MP			HP to MP		HP condensed
F G3171 only	F K301 +bypass	F K3471 only	F K6101	F K6101 to MP	F K6102
327FI71	*32FI35*	*347FI4*	*61F041*	*61F042*	61F044
720.51	2305.11	447.98	1570.39	765.16	365.03
732.50	2348.34	447.59	1565.07	764.08	375.51
679.20	2505.44	447.91	1575.31	776.73	378.06
686.48	2498.45	447.39	1561.95	762.42	379.12
757.56	2329.48	447.07	1595.69	786.61	386.78
629.01	2333.26	447.08	1599.08	796.61	384.70
495.82	2455.73	446.71	1587.07	1167.92	374.85
517.56	2502.58	447.30	1599.82	1194.13	369.06
450.70	2491.37	447.56	1622.99	1227.60	342.04
530.99	2413.99	447.84	1584.08	1227.60	323.14
535.53	2556.94	448.07	1493.68	1227.60	312.33
409.73	2697.73	448.00	1454.09	1227.60	306.66
590.27	2478.37	447.24	1480.46	1227.60	310.46
659.10	2378.24	446.76	1448.81	1227.60	312.84
716.82	2308.96	446.76	1476.72	1227.60	357.35
708.39	2316.46	446.79	1510.89	1227.60	376.81
598.11	2554.77	447.51	1514.23	1227.60	374.03
433.07	2739.38	447.70	1528.30	1227.60	362.87
647.72	2289.21	446.84	1547.75	1227.60	352.85
784.13	2266.22	446.86	1527.00	1227.60	308.65
467.58	2524.13	447.40	1566.42	1227.60	332.68
550.82	2464.36	446.98	1587.89	1227.60	340.60
633.64	2347.05	446.61	1585.95	1227.60	348.71
612.99	2248.12	446.61	1587.22	1227.60	368.68
596.30	2348.98	446.92	1559.17	1227.60	364.51
656.74	2486.14	447.15	1527.11	1291.97	345.76
671.26	2414.50	446.85	1584.03	1450.42	352.30
600.07	2555.64	447.36	1579.97	1452.24	362.27
512.03	2616.30	447.69	1602.57	1354.88	377.62
470.80	2711.59	448.03	1552.93	1393.52	379.34
465.11	2616.53	447.82	1539.53	1393.52	387.46
597.44	**2454.95**	**447.30**	**1552.13**	**1168.37**	**355.26**

Table C.1.2: HP consumers-turbines

	letdowns							
	LDHM				LDH			Total Letdowns
	LDHM S3261	LDHM S3274	LDHM Luboil	total	F G3171 bypass	F K3471 bypass	total	
TIME	326FI36	327FI58	324FC1		32X016	327FI56		
01-Mar-00	0.00	0.00	361.08	361.08	174.28	0.00	174.28	535.36
02-Mar-00	0.00	0.00	348.54	348.54	186.25	0.00	186.25	534.79
03-Mar-00	0.00	0.00	350.88	350.88	160.62	0.00	160.62	511.50
04-Mar-00	0.00	0.00	350.30	350.30	153.06	0.00	153.06	503.35
05-Mar-00	0.00	0.00	360.56	360.56	117.01	0.00	117.01	477.57
06-Mar-00	0.00	0.00	356.83	356.83	160.84	0.00	160.84	517.67
07-Mar-00	0.00	0.00	350.96	350.96	229.49	0.00	229.49	580.44
08-Mar-00	0.00	0.00	353.73	353.73	221.73	0.00	221.73	575.46
09-Mar-00	0.00	0.00	363.17	363.17	210.19	0.00	210.19	573.36
10-Mar-00	0.00	0.00	361.31	361.31	284.35	0.00	284.35	645.67
11-Mar-00	0.00	0.00	359.66	359.66	201.07	0.00	201.07	560.73
12-Mar-00	0.00	0.00	344.40	344.40	133.91	0.00	133.91	478.31
13-Mar-00	0.00	0.00	344.25	344.25	138.55	0.00	138.55	482.80
14-Mar-00	0.00	0.00	359.31	359.31	147.14	0.00	147.14	506.46
15-Mar-00	0.00	0.00	354.01	354.01	165.58	0.00	165.58	519.59
16-Mar-00	0.00	0.00	359.09	359.09	135.08	0.00	135.08	494.18
17-Mar-00	0.00	0.00	351.97	351.97	131.70	0.00	131.70	483.67
18-Mar-00	0.00	0.00	346.37	346.37	104.64	0.00	104.64	451.00
19-Mar-00	0.00	0.00	348.97	348.97	199.55	0.00	199.55	548.51
20-Mar-00	0.00	0.00	344.03	344.03	126.38	0.00	126.38	470.41
21-Mar-00	0.00	0.00	352.81	352.81	157.86	0.00	157.86	510.67
22-Mar-00	0.00	0.00	349.46	349.46	122.29	0.00	122.29	471.75
23-Mar-00	0.00	0.00	357.74	357.74	107.35	0.00	107.35	465.09
24-Mar-00	0.00	0.00	359.65	359.65	137.02	0.00	137.02	496.67
25-Mar-00	0.00	0.00	353.94	353.94	137.54	0.00	137.54	491.47
26-Mar-00	0.00	0.00	346.98	346.98	191.19	0.00	191.19	538.17
27-Mar-00	0.00	0.00	345.08	345.08	183.49	0.00	183.49	528.56
28-Mar-00	0.00	0.00	341.94	341.94	144.39	0.00	144.39	486.33
29-Mar-00	0.00	0.00	342.16	342.16	140.25	0.00	140.25	482.41
30-Mar-00	0.00	0.00	319.89	319.89	125.31	0.00	125.31	445.20
31-Mar-00	0.00	0.00	332.59	332.59	211.34	0.00	211.34	543.93
average	**0.00**	**0.00**	**350.70**	**350.70**	**162.56**	**0.00**	**162.56**	**513.26**

Table C.1.3: HP consumers – Letdown stations

	users						
	E3801	Alkyl	HCS(TCS)		Bit Blend	E224	Total HP to users
	Fs S3287	Fs	E9105	E9104	Fs	Fs	
TIME	327FI96	32F654	91FC16	91FC11	32FI37	02FC53	
01-Mar-00	33.85	19.53	50.43	155.85	130.88	267.08	657.63
02-Mar-00	38.89	19.19	0.00	230.28	26.66	265.09	580.10
03-Mar-00	39.87	22.16	0.00	273.91	160.70	257.81	754.46
04-Mar-00	38.31	20.55	0.00	298.67	102.87	252.49	712.88
05-Mar-00	36.56	17.50	0.00	304.05	57.25	240.82	656.18
06-Mar-00	36.33	19.40	0.00	305.90	102.64	246.06	710.33
07-Mar-00	35.61	16.86	0.00	266.85	118.29	253.63	691.25
08-Mar-00	36.43	20.55	0.00	200.26	131.54	252.50	641.28
09-Mar-00	23.54	19.97	0.00	126.66	110.84	240.45	521.46
10-Mar-00	26.57	19.12	0.00	173.07	157.27	246.60	622.63
11-Mar-00	35.82	19.83	0.00	204.13	129.38	244.98	634.13
12-Mar-00	35.37	18.09	0.00	206.69	111.33	251.27	622.75
13-Mar-00	35.42	18.75	0.00	214.01	101.30	248.85	618.32
14-Mar-00	34.73	16.62	0.00	224.52	100.84	249.93	626.63
15-Mar-00	35.21	15.93	0.00	224.98	100.84	251.00	627.96
16-Mar-00	35.28	16.74	0.00	99.55	100.84	228.40	480.81
17-Mar-00	35.30	18.06	0.00	233.62	100.84	239.15	626.96
18-Mar-00	36.03	10.39	0.00	235.38	100.84	250.63	633.27
19-Mar-00	34.24	7.73	0.00	237.72	100.84	248.55	629.07
20-Mar-00	33.67	3.21	0.00	249.94	100.84	249.94	637.59
21-Mar-00	32.83	2.85	0.00	253.82	100.84	246.85	637.19
22-Mar-00	31.97	3.41	0.00	253.26	100.84	245.70	635.17
23-Mar-00	32.10	7.21	0.00	255.27	100.84	240.13	635.54
24-Mar-00	32.37	7.37	0.00	132.32	100.84	243.59	516.49
25-Mar-00	33.37	8.12	0.00	0.00	100.84	228.37	370.69
26-Mar-00	36.43	7.81	0.00	0.00	100.84	249.61	394.68
27-Mar-00	35.82	9.83	0.00	0.00	100.84	236.91	383.39
28-Mar-00	35.38	10.43	0.00	0.00	100.84	221.31	367.96
29-Mar-00	34.05	7.62	0.00	0.00	100.84	228.89	371.38
30-Mar-00	36.16	10.07	0.00	0.00	100.84	264.30	411.36
31-Mar-00	36.13	10.97	0.00	0.03	100.84	267.41	415.37
average	**34.63**	**13.74**	**1.63**	**172.93**	**105.03**	**247.04**	**575.00**

Table C.1.4: HP consumers - users

		eH	
TIME	Total HP consumed	imbalance = production - consumption	% imbalance
01-Mar-00	6602.00	-740.60	-12.64
02-Mar-00	6583.90	-677.24	-11.47
03-Mar-00	6851.87	-600.66	-9.61
04-Mar-00	6789.62	-703.47	-11.56
05-Mar-00	6650.32	-791.94	-13.52
06-Mar-00	6621.13	-677.05	-11.39
07-Mar-00	6631.85	-765.78	-13.05
08-Mar-00	6653.05	-677.46	-11.34
09-Mar-00	6449.47	-709.90	-12.37
10-Mar-00	6568.35	-570.10	-9.50
11-Mar-00	6541.41	-653.27	-11.09
12-Mar-00	6417.27	-772.10	-13.68
13-Mar-00	6407.90	-647.68	-11.24
14-Mar-00	6378.84	-828.20	-14.92
15-Mar-00	6454.16	-657.86	-11.35
16-Mar-00	6334.32	-512.34	-8.80
17-Mar-00	6599.27	-540.51	-8.92
18-Mar-00	6595.58	-664.04	-11.20
19-Mar-00	6461.95	-722.18	-12.58
20-Mar-00	6440.86	-687.38	-11.95
21-Mar-00	6486.08	-798.36	-14.04
22-Mar-00	6497.56	-711.20	-12.29
23-Mar-00	6462.59	-743.44	-13.00
24-Mar-00	6276.77	-579.10	-10.16
25-Mar-00	6178.04	-683.21	-12.43
26-Mar-00	6395.75	-505.46	-8.58
27-Mar-00	6380.89	-563.19	-9.68
28-Mar-00	6399.59	-576.67	-9.90
29-Mar-00	6410.01	-689.54	-12.05
30-Mar-00	6419.25	-624.40	-10.78
31-Mar-00	6415.76	-625.06	-10.79
average	**6495.34**	**-667.72**	**-11.46**

Table C.1.5: HP steam imbalance

APPENDIX C.2

Overall steam balance of March 2000 – AMP header

	Ass MP feeders							
	HP turbines							
	K3471			G3171		K301		Total AssMP feeders
	F K3471 only	F K3471 bypass	desuperheater water	F G3171+bypass	desuperheater	F K301 + bypass	desuperheater	
TIME	347FI4	327FI56	calcul	327FC70	calcul	32FI35	calcul	
01-Mar-00	447.98	0.00	4.21	894.79	14.69	2305.11	26.16	3692.94
02-Mar-00	447.59	0.00	4.21	918.74	15.09	2348.34	26.65	3760.61
03-Mar-00	447.91	0.00	4.21	839.82	13.79	2505.44	28.43	3839.60
04-Mar-00	447.39	0.00	4.21	839.53	13.79	2498.45	28.35	3831.72
05-Mar-00	447.07	0.00	4.20	874.57	14.36	2329.48	26.43	3696.11
06-Mar-00	447.08	0.00	4.20	789.85	12.97	2333.26	26.48	3613.83
07-Mar-00	446.71	0.00	4.20	725.27	11.91	2455.73	27.87	3671.68
08-Mar-00	447.30	0.00	4.21	739.29	12.14	2502.58	28.40	3733.91
09-Mar-00	447.56	0.00	4.21	660.89	10.85	2491.37	28.27	3643.14
10-Mar-00	447.84	0.00	4.21	815.35	13.39	2413.99	27.39	3722.17
11-Mar-00	448.07	0.00	4.21	736.60	12.10	2556.94	29.01	3786.93
12-Mar-00	448.00	0.00	4.21	543.64	8.93	2697.73	30.61	3733.12
13-Mar-00	447.24	0.00	4.20	728.82	11.97	2478.37	28.12	3698.73
14-Mar-00	446.76	0.00	4.20	806.24	13.24	2378.24	26.99	3675.66
15-Mar-00	446.76	0.00	4.20	882.40	14.49	2308.96	26.20	3683.01
16-Mar-00	446.79	0.00	4.20	843.48	13.85	2316.46	26.29	3651.06
17-Mar-00	447.51	0.00	4.21	729.81	11.98	2554.77	28.99	3777.27
18-Mar-00	447.70	0.00	4.21	537.70	8.83	2739.38	31.09	3768.91
19-Mar-00	446.84	0.00	4.20	847.26	13.91	2289.21	25.98	3627.40
20-Mar-00	446.86	0.00	4.20	910.50	14.95	2266.22	25.72	3668.45
21-Mar-00	447.40	0.00	4.21	625.44	10.27	2524.13	28.64	3640.10
22-Mar-00	446.98	0.00	4.20	673.11	11.05	2464.36	27.96	3627.67
23-Mar-00	446.61	0.00	4.20	740.99	12.17	2347.05	26.63	3577.65
24-Mar-00	446.61	0.00	4.20	750.01	12.32	2248.12	25.51	3486.76
25-Mar-00	446.92	0.00	4.20	733.83	12.05	2348.98	26.66	3572.64
26-Mar-00	447.15	0.00	4.20	847.93	13.92	2486.14	28.21	3827.55
27-Mar-00	446.85	0.00	4.20	854.75	14.04	2414.50	27.40	3761.73
28-Mar-00	447.36	0.00	4.21	744.46	12.22	2555.64	29.00	3792.89
29-Mar-00	447.69	0.00	4.21	652.28	10.71	2616.30	29.69	3760.89
30-Mar-00	448.03	0.00	4.21	596.11	9.79	2711.59	30.77	3800.50
31-Mar-00	447.82	0.00	4.21	676.45	11.11	2616.53	29.69	3785.81
average	447.30	0.00	4.21	867.13	14.24	2454.95	27.86	3815.68

Table C.2.1: AssMP feeeders

	Ass MP consumed					
	Turbines					Users
	K3262	K3272	K3271+Soot Blowers	U3200	U3500	CCU
	F K3262	K3272	K3271+Soot Blowers	U3200	U3500	F CCU riser
TIME	32X075	*327FI21*	*327FI103*	*estimate*	*32F039*	*61FI53*
01-Mar-00	1.07	32.11	93.69	275.00	20.16	153.15
02-Mar-00	1.53	41.13	97.70	275.00	20.58	153.18
03-Mar-00	0.23	34.07	96.45	275.00	16.29	157.06
04-Mar-00	1.21	30.88	99.07	275.00	20.22	154.33
05-Mar-00	0.25	38.11	93.28	275.00	17.92	153.79
06-Mar-00	0.00	31.24	91.65	275.00	16.45	156.40
07-Mar-00	1.50	30.51	92.74	275.00	19.00	154.81
08-Mar-00	0.02	44.84	95.84	275.00	16.66	156.26
09-Mar-00	0.41	23.36	93.28	275.00	17.24	155.03
10-Mar-00	0.00	30.75	92.77	275.00	19.83	154.95
11-Mar-00	0.00	33.80	94.84	275.00	17.60	156.47
12-Mar-00	4.18	32.09	91.00	275.00	17.58	158.65
13-Mar-00	3.77	31.05	92.55	275.00	16.23	156.73
14-Mar-00	4.62	31.32	91.72	275.00	17.79	155.63
15-Mar-00	6.27	29.03	92.10	275.00	17.30	154.53
16-Mar-00	1.58	31.21	93.19	275.00	16.14	155.99
17-Mar-00	3.83	30.69	95.51	275.00	21.80	156.41
18-Mar-00	3.45	35.56	96.78	275.00	22.42	158.13
19-Mar-00	3.49	31.64	94.76	275.00	20.05	155.11
20-Mar-00	5.74	27.57	95.77	275.00	20.46	153.55
21-Mar-00	3.79	17.58	94.87	275.00	17.14	157.87
22-Mar-00	4.15	35.05	94.08	275.00	17.38	156.69
23-Mar-00	6.83	39.78	94.49	275.00	16.29	154.89
24-Mar-00	0.81	32.23	94.44	275.00	15.72	154.54
25-Mar-00	3.48	41.69	88.23	275.00	17.36	153.20
26-Mar-00	3.28	71.33	87.98	275.00	23.24	151.33
27-Mar-00	3.06	63.72	87.18	275.00	18.82	150.59
28-Mar-00	6.02	7.86	86.23	275.00	18.92	149.19
29-Mar-00	5.68	16.50	86.09	275.00	16.00	147.88
30-Mar-00	5.83	57.22	84.22	275.00	18.92	146.82
31-Mar-00	10.14	36.75	83.27	275.00	18.51	144.57
average	3.10	34.54	92.44	275.00	18.39	154.12

Table C.2.2: AssMP consumers – Turbines and users

	letdowns						
	LDA						LDAL
	LDA CDU2	LDA FCCU	LDA Luboil	LDA Visbreaker		total LDA	LDAL Utilities
	Fs CDU2	Fs FCCU	Fs Luboil	AssMP to MP	MP to AssMP		Fs
TIME	327FI73	326FI50	32FI402	328FI4	328FI5		327 FI 57
01-Mar-00	394.85	492.96	9.40	228.68	0.00	1125.89	5.66
02-Mar-00	408.13	531.72	10.53	274.32	0.00	1224.69	57.55
03-Mar-00	432.43	411.33	11.34	292.78	0.00	1147.87	105.26
04-Mar-00	439.73	471.83	12.49	278.44	0.00	1202.48	39.97
05-Mar-00	470.93	296.24	13.33	272.85	0.00	1053.34	12.70
06-Mar-00	516.06	314.03	12.67	274.63	0.00	1117.39	3.61
07-Mar-00	391.08	648.95	11.98	259.00	0.00	1311.01	23.37
08-Mar-00	354.26	692.00	11.86	245.48	0.00	1303.61	82.40
09-Mar-00	461.56	398.96	12.47	208.05	0.00	1081.03	29.43
10-Mar-00	443.29	436.68	11.99	234.36	0.00	1126.32	7.82
11-Mar-00	382.97	505.22	14.45	239.26	0.00	1141.90	32.39
12-Mar-00	400.35	453.94	16.09	252.70	0.00	1123.08	52.42
13-Mar-00	464.79	317.71	12.26	253.82	0.00	1048.57	103.31
14-Mar-00	498.01	358.74	12.15	253.43	0.00	1122.33	51.39
15-Mar-00	491.68	423.94	10.08	264.51	0.00	1190.22	32.89
16-Mar-00	490.73	353.55	11.51	213.13	0.00	1068.92	83.72
17-Mar-00	483.73	398.88	14.25	198.82	0.00	1095.67	104.69
18-Mar-00	491.70	373.07	13.25	194.50	0.00	1072.52	100.70
19-Mar-00	468.97	245.21	12.24	174.43	0.00	900.85	70.41
20-Mar-00	468.28	436.94	11.65	236.86	0.00	1153.74	35.46
21-Mar-00	511.59	323.81	16.07	214.29	0.00	1065.75	81.27
22-Mar-00	470.12	309.79	16.07	173.20	0.00	969.19	90.37
23-Mar-00	507.32	299.11	15.58	170.23	0.00	992.24	66.42
24-Mar-00	467.70	323.34	14.23	194.95	0.00	1000.21	25.37
25-Mar-00	467.57	401.56	13.86	180.22	0.00	1063.20	157.99
26-Mar-00	470.82	677.10	15.16	111.66	0.00	1274.73	296.31
27-Mar-00	478.55	601.45	14.24	118.67	0.00	1212.91	268.16
28-Mar-00	501.13	584.36	17.74	90.85	0.00	1194.09	176.74
29-Mar-00	504.73	477.13	17.23	91.22	0.00	1090.31	129.67
30-Mar-00	526.05	513.27	15.29	88.54	0.00	1143.15	211.58
31-Mar-00	525.98	498.72	14.60	85.30	0.00	1124.60	67.64
average	464.03	437.79	13.42	205.46	0.00	1120.70	84.09

Table C.2.3: AssMP consumers – Letdown stations

TIME	Total AssMP consumed	eA imbalance = production - consumption	% imbalance
01-Mar-00	1706.74	1986.21	53.78
02-Mar-00	1871.36	1889.25	50.24
03-Mar-00	1832.24	2007.36	52.28
04-Mar-00	1823.15	2008.56	52.42
05-Mar-00	1644.39	2051.72	55.51
06-Mar-00	1691.74	1922.09	53.19
07-Mar-00	1907.94	1763.74	48.04
08-Mar-00	1974.63	1759.28	47.12
09-Mar-00	1674.78	1968.36	54.03
10-Mar-00	1707.45	2014.72	54.13
11-Mar-00	1752.00	2034.93	53.74
12-Mar-00	1754.01	1979.11	53.01
13-Mar-00	1727.21	1971.51	53.30
14-Mar-00	1749.81	1925.86	52.39
15-Mar-00	1797.34	1885.67	51.20
16-Mar-00	1725.75	1925.31	52.73
17-Mar-00	1783.60	1993.67	52.78
18-Mar-00	1764.55	2004.36	53.18
19-Mar-00	1551.33	2076.07	57.23
20-Mar-00	1767.30	1901.15	51.82
21-Mar-00	1713.28	1926.82	52.93
22-Mar-00	1641.91	1985.76	54.74
23-Mar-00	1645.93	1931.72	53.99
24-Mar-00	1598.31	1888.45	54.16
25-Mar-00	1800.15	1772.49	49.61
26-Mar-00	2183.19	1644.36	42.96
27-Mar-00	2079.45	1682.29	44.72
28-Mar-00	1914.04	1878.85	49.54
29-Mar-00	1767.13	1993.76	53.01
30-Mar-00	1942.73	1857.77	48.88
31-Mar-00	1760.48	2025.33	53.50
average	1782.38	2033.29	53.29

Table C.2.4: AssMP steam imbalance

APPENDIX C.3

Overall steam balance of March 2000 – MP header

	MP feeders											
	Local feeders					HP turbine	LDHM					
	E6106/7			E8401	V8011	K6101	LDHM S3261		LDHM S3274			LDHM Luboil
	T 6 to 7 ?	FM E6106/7	P 6 to 7(barg)	FM	FM	F K6101 to MP	Fs S3261	desuperheater	Fs S3274	desuperheater	Ts S3274	Fs Luboil
TIME	32T667	61F019	61P015	84FA11	80FI85	61F042	326FI36		327FI58		327TC53	324FC1
01-Mar-00	205.96	418.11	1659.42	147.09	55.04	765.16	0.00	0.00	0.00	0.00	283.75	361.08
02-Mar-00	205.97	413.67	1655.50	147.23	51.40	764.08	0.00	0.00	0.00	0.00	281.53	348.54
03-Mar-00	205.54	447.92	1653.69	150.56	50.50	776.73	0.00	0.00	0.00	0.00	283.15	350.88
04-Mar-00	205.63	426.61	1647.61	155.57	50.29	762.42	0.00	0.00	0.00	0.00	285.36	350.30
05-Mar-00	205.49	423.42	1646.95	154.63	50.50	786.61	0.00	0.00	0.00	0.00	292.23	360.56
06-Mar-00	205.46	428.26	1648.76	156.07	48.75	796.61	0.00	0.00	0.00	0.00	288.80	356.83
07-Mar-00	205.87	379.93	1651.45	156.57	45.17	1167.92	0.00	0.00	0.00	0.00	286.37	350.96
08-Mar-00	205.91	364.32	1655.80	152.35	46.65	1194.13	0.00	0.00	0.00	0.00	282.53	353.73
09-Mar-00	204.85	359.52	1619.89	150.61	44.73	1227.60	0.00	0.00	0.00	0.00	290.26	363.17
10-Mar-00	205.66	343.61	1645.20	145.84	47.62	1227.60	0.00	0.00	0.00	0.00	287.05	361.31
11-Mar-00	205.67	297.08	1646.82	151.56	43.73	1227.60	0.00	0.00	0.00	0.00	288.86	359.66
12-Mar-00	205.70	279.23	1644.80	143.75	45.71	1227.60	0.00	0.00	0.00	0.00	287.08	344.40
13-Mar-00	205.34	280.11	1637.76	144.69	46.07	1227.60	0.00	0.00	0.00	0.00	286.77	344.25
14-Mar-00	205.50	301.54	1637.35	141.55	46.91	1227.60	0.00	0.00	0.00	0.00	290.10	359.31
15-Mar-00	205.51	335.19	1640.86	140.10	41.60	1227.60	0.00	0.00	0.00	0.00	289.18	354.01
16-Mar-00	205.41	299.32	1637.42	131.82	39.91	1227.60	0.00	0.00	0.00	0.00	289.99	359.09
17-Mar-00	205.65	343.04	1641.83	140.13	44.44	1227.60	0.00	0.00	0.00	0.00	282.85	351.97
18-Mar-00	205.90	435.80	1653.61	138.44	44.95	1227.60	0.00	0.00	0.00	0.00	278.09	346.37
19-Mar-00	205.75	419.47	1651.75	151.82	47.60	1227.60	0.00	0.00	0.00	0.00	286.60	348.97
20-Mar-00	205.85	393.41	1648.80	150.93	45.78	1227.60	0.00	0.00	0.00	0.00	285.57	344.03
21-Mar-00	205.52	371.85	1643.73	145.69	46.26	1227.60	0.00	0.00	0.00	0.00	291.66	352.81
22-Mar-00	205.48	376.36	1646.13	149.33	46.48	1227.60	0.00	0.00	0.00	0.00	289.97	349.46
23-Mar-00	205.56	396.72	1647.62	149.46	47.22	1227.60	0.00	0.00	0.00	0.00	290.13	357.74
24-Mar-00	205.57	420.77	1652.80	139.99	51.01	1227.60	0.00	0.00	0.00	0.00	287.17	359.65
25-Mar-00	205.72	426.78	1653.21	124.34	22.00	1227.60	0.00	0.00	0.00	0.00	290.08	353.94
26-Mar-00	205.85	395.41	1646.44	105.50	0.15	1291.97	0.00	0.00	0.00	0.00	277.93	346.98
27-Mar-00	206.60	399.09	1685.25	107.12	0.21	1450.42	0.00	0.00	0.00	0.00	287.23	345.08
28-Mar-00	207.77	415.42	1725.00	118.47	0.23	1452.24	0.00	0.00	0.00	0.00	284.78	341.94
29-Mar-00	207.56	421.23	1725.06	124.01	0.24	1354.88	0.00	0.00	0.00	0.00	288.93	342.16
30-Mar-00	207.83	367.68	1724.84	117.87	0.32	1393.52	0.00	0.00	0.00	0.00	288.77	319.89
31-Mar-00	207.83	385.62	1724.97	122.68	0.36	1393.52	0.00	0.00	0.00	0.00	287.52	332.59
average	205.93	379.56	1658.14	140.51	37.16	1168.37	0.00	0.00	0.00	0.00	286.78	350.70

Table C.3.1: MP feeders

MP feeders					
LDA					
LDA CDU2	LDA FCCU	LDA Luboil	LDA Visbreaker		Total MP produced
Fs CDU2	Fs FCCU	Fs Luboil	AssMP to MP	MP to AssMP	
327FI73	326FI50	32FI402	328FI4	328FI5	
394.85	492.96	9.40	228.68	0.00	2872.36
408.13	531.72	10.53	274.32	0.00	2949.60
432.43	411.33	11.34	292.78	0.00	2924.45
439.73	471.83	12.49	278.44	0.00	2947.67
470.93	296.24	13.33	272.85	0.00	2829.06
516.06	314.03	12.67	274.63	0.00	2903.91
391.08	648.95	11.98	259.00	0.00	3411.55
354.26	692.00	11.86	245.48	0.00	3414.78
461.56	398.96	12.47	208.05	0.00	3226.67
443.29	436.68	11.99	234.36	0.00	3252.30
382.97	505.22	14.45	239.26	0.00	3221.52
400.35	453.94	16.09	252.70	0.00	3163.77
464.79	317.71	12.26	253.82	0.00	3091.29
498.01	358.74	12.15	253.43	0.00	3199.24
491.68	423.94	10.08	264.51	0.00	3288.72
490.73	353.55	11.51	213.13	0.00	3126.67
483.73	398.88	14.25	198.82	0.00	3202.86
491.70	373.07	13.25	194.50	0.00	3265.67
468.97	245.21	12.24	174.43	0.00	3096.32
468.28	436.94	11.65	236.86	0.00	3315.49
511.59	323.81	16.07	214.29	0.00	3209.97
470.12	309.79	16.07	173.20	0.00	3118.41
507.32	299.11	15.58	170.23	0.00	3170.97
467.70	323.34	14.23	194.95	0.00	3199.24
467.57	401.56	13.86	180.22	0.00	3217.86
470.82	677.10	15.16	111.66	0.00	3414.74
478.55	601.45	14.24	118.67	0.00	3514.84
501.13	584.36	17.74	90.85	0.00	3522.39
504.73	477.13	17.23	91.22	0.00	3332.82
526.05	513.27	15.29	88.54	0.00	3342.43
525.98	498.72	14.60	85.30	0.00	3359.37
464.03	**437.79**	**13.42**	**205.46**	**0.00**	**3197.00**

Table C.3.2: MP feeders - part 2

MP consumers								
Luboil								
Turbines		Users						
K4402	others	J4101A/B	J4401	J4102	C4701	J4301	F4101	F4001
44FI42	*estimate*	*41FI117*	*44FI46*	*41FI123*	*47FCA19*	*43FI42*	*41FI142*	*40FI51*
874.47	40.00	40.09	4.55	27.29	18.98	-0.02	0.70	0.00
829.78	40.00	40.84	4.61	27.56	19.00	-0.01	0.69	0.00
848.73	40.00	40.84	4.58	27.54	19.03	-0.02	0.73	0.00
841.12	40.00	40.82	4.60	27.46	19.01	-0.02	0.71	0.00
869.23	40.00	40.94	4.53	27.23	19.00	-0.02	0.74	0.00
861.71	40.00	40.93	4.55	27.46	18.96	-0.02	0.74	0.00
855.37	40.00	40.80	4.61	27.65	18.99	-0.01	0.70	0.00
869.63	40.00	40.77	4.63	27.69	18.99	0.03	0.73	0.00
890.80	40.00	41.14	4.53	27.01	18.98	-0.02	0.73	0.00
875.44	40.00	40.87	4.60	27.21	19.22	-0.02	0.73	0.00
880.86	40.00	40.36	4.58	27.29	20.99	-0.01	0.72	0.00
822.30	40.00	40.15	4.66	27.66	21.02	-0.01	0.72	0.00
789.42	40.00	40.16	4.63	27.56	21.01	-0.01	0.74	0.00
838.19	40.00	40.30	4.60	27.32	21.00	-0.02	0.73	0.00
807.28	40.00	40.24	4.62	27.50	20.99	-0.02	0.72	0.00
830.59	40.00	40.29	4.59	27.44	21.00	-0.02	0.74	0.00
821.97	40.00	40.27	4.65	27.55	21.01	-0.01	0.70	0.00
808.05	40.00	40.13	4.69	27.89	20.72	-0.01	0.69	0.00
797.69	40.00	40.21	4.65	27.70	21.02	-0.01	0.71	0.00
781.89	40.00	40.16	4.65	27.80	21.00	-0.01	0.68	0.00
845.17	40.00	40.21	4.62	27.69	21.02	-0.02	0.74	0.00
823.79	40.00	40.20	4.63	27.64	20.99	-0.01	0.75	0.00
847.38	40.00	41.89	4.60	27.57	21.01	-0.02	0.73	0.00
852.56	40.00	44.24	4.59	27.54	21.00	-0.02	0.74	0.00
845.46	40.00	44.40	4.61	27.62	20.98	-0.02	0.72	0.00
831.63	40.00	44.43	4.66	27.85	21.00	-0.02	0.68	0.00
812.20	40.00	44.42	4.65	27.85	21.00	-0.02	0.72	0.00
813.97	40.00	44.37	4.68	27.94	21.01	-0.01	0.72	0.00
830.13	40.00	41.51	4.65	28.03	21.00	-0.01	0.73	0.00
748.37	40.00	41.15	4.74	28.47	19.29	0.00	0.70	0.00
779.62	40.00	45.04	4.68	28.21	18.00	0.00	0.70	0.00
833.06	**40.00**	**41.36**	**4.62**	**27.62**	**20.20**	**-0.01**	**0.72**	**0.00**

Table C.3.3: MP consumers – Luboil section

MP consumers													
CDU2													
Turbines					Users								
P3701C in	P3701C back	P3701C	P3371	K7101	J7101	U7700	S3285	QMI	F7101	U7400	Desalter	Soot Blowi	Visb flare
326FI66	*326FI67*	*calculated*	*337FI12*	*71FI111*	*71FI16*	*327FI113*	*327FI135*	*estimate*	*327FI118*	*74FI10*	*estimate*	*estimate*	*368FI6*
336.11	0.00	336.11	54.09	137.01	13.28	102.31	37.87	1.00	15.73	48.10	0.00	5.00	1.03
316.98	0.00	316.98	54.05	129.15	13.28	102.29	37.75	1.00	15.72	48.16	0.00	5.00	0.33
308.88	0.00	308.88	54.02	133.05	13.31	102.15	37.46	1.00	15.70	48.31	0.00	5.00	0.25
312.43	0.00	312.43	53.97	125.19	13.25	101.98	37.57	1.00	15.71	48.09	0.00	5.00	0.03
244.27	0.00	244.27	53.98	117.36	13.24	100.60	36.82	1.00	15.57	48.04	0.00	5.00	0.41
205.52	0.04	205.47	54.11	118.75	13.26	100.95	36.99	1.00	15.62	48.12	0.00	5.00	0.29
330.83	0.00	330.83	54.17	119.49	13.23	102.13	37.87	1.00	15.70	48.06	0.00	5.00	0.41
372.20	0.00	372.20	54.22	126.69	13.34	101.49	37.35	1.00	15.75	48.39	0.00	5.00	0.43
272.60	2.11	270.49	53.63	137.48	13.05	99.54	36.60	1.00	13.65	47.29	0.00	5.00	4.92
298.42	0.11	298.31	53.92	146.50	13.15	101.10	37.37	1.00	11.00	47.62	0.00	5.00	0.38
356.21	0.00	356.21	53.95	118.04	13.29	101.33	37.26	1.00	11.06	48.20	0.00	5.00	0.08
348.44	0.00	348.44	53.96	117.14	13.41	102.55	37.78	1.00	11.23	48.69	0.00	5.00	0.46
281.01	0.00	281.01	53.94	129.58	13.40	102.14	37.44	1.00	11.13	48.63	0.00	5.00	0.22
258.67	0.00	258.67	53.95	123.12	13.29	101.91	37.60	1.00	11.03	48.21	0.00	5.00	0.52
258.66	0.00	258.66	53.93	111.19	13.27	101.30	37.34	1.00	11.00	48.08	0.00	5.00	0.11
255.65	0.10	255.55	54.00	119.47	13.30	101.25	37.24	1.00	11.02	48.26	0.00	5.00	0.71
261.92	0.00	261.92	53.96	124.08	13.37	103.38	38.64	1.00	11.26	48.46	0.00	5.00	0.20
260.49	0.00	260.49	54.05	142.62	13.47	104.13	38.89	1.00	11.34	48.95	0.00	5.00	0.09
249.74	0.11	249.63	54.04	119.13	13.20	102.19	37.95	1.00	11.05	47.98	0.00	5.00	0.63
262.41	0.26	262.16	54.00	115.95	13.30	102.28	38.04	1.00	11.07	48.16	0.00	5.00	0.05
223.93	0.20	223.73	53.99	121.84	13.29	101.32	37.46	1.00	10.98	48.11	0.00	5.00	0.31
251.46	0.10	251.36	54.07	125.59	13.35	101.69	37.46	1.00	11.04	48.41	0.00	5.00	0.00
229.99	3.79	226.20	54.10	130.27	13.30	101.51	37.46	1.00	11.02	48.20	0.00	5.00	0.29
259.25	0.00	259.25	54.16	132.41	13.30	101.28	37.26	1.00	11.01	48.19	0.00	5.00	0.46
262.91	0.00	262.91	54.28	125.70	13.29	101.46	37.52	1.00	11.00	48.15	0.00	5.00	1.36
290.65	0.00	290.65	54.65	132.67	13.31	103.23	38.63	1.00	11.28	48.28	0.00	5.00	0.32
279.83	0.00	279.83	54.50	125.52	13.27	101.44	37.52	1.00	16.86	48.10	0.00	5.00	0.06
275.31	0.30	275.01	54.50	132.88	13.31	102.50	38.31	1.00	14.12	48.27	0.00	5.00	0.12
263.35	0.12	263.23	54.50	125.68	13.39	102.75	38.40	1.00	11.49	48.56	0.00	5.00	0.13
264.68	0.00	264.68	54.38	130.77	13.42	103.66	38.94	1.00	11.60	48.76	0.00	5.00	1.00
263.88	0.00	263.88	54.28	140.58	13.31	103.51	39.00	1.00	11.55	48.29	0.00	5.00	1.16
279.25	0.23	279.01	54.11	126.93	13.30	101.98	37.74	1.00	12.69	48.23	0.00	5.00	0.54

Table C.3.4: MP consumers – Crude Distiller Unit No 2 section

MP consumers								
CCU								
Turbines								
P6110/11/1	P6102A/B	P6115/17/1	P3202	P3501/5	P3263/4	P3261/6	P3262A	U300
326F138	*326F135*	*326F134*	*32F051*	*32F167*	*326F132*	*326F128*	*32F664*	*estimate*
227.34	0.00	43.82	108.83	139.60	-0.63	0.00	5.99	8.00
227.40	0.00	44.12	117.90	139.60	-0.67	0.00	6.16	8.00
225.64	0.00	43.92	125.17	139.60	-0.62	0.00	6.36	8.00
226.02	0.00	43.96	121.32	139.60	-0.59	0.00	6.77	8.00
225.02	0.00	46.45	119.95	139.60	-0.54	0.00	6.57	8.00
225.34	0.00	48.00	113.98	139.60	-0.49	0.00	7.55	8.00
227.57	0.00	49.58	105.71	139.60	-0.65	0.00	7.02	8.00
227.69	0.00	49.49	115.27	139.60	-0.84	0.00	6.99	8.00
223.80	0.00	46.28	119.23	139.60	-0.62	0.00	6.46	8.00
227.40	0.00	46.76	134.07	139.60	-0.64	0.00	6.67	8.00
227.41	0.00	46.81	138.56	139.60	-0.59	0.00	6.60	8.00
227.10	0.00	46.66	145.91	139.60	-0.46	0.00	6.90	8.00
225.92	0.00	46.79	134.56	139.60	-0.31	0.00	6.77	8.00
227.79	0.00	47.28	110.50	139.60	-0.34	0.00	6.59	8.00
226.37	0.00	48.00	118.76	139.60	-0.32	0.00	6.09	8.00
225.08	0.00	47.15	124.07	139.60	-0.27	0.00	5.80	8.00
226.79	0.00	46.80	131.87	139.60	-0.27	0.00	3.53	8.00
227.16	0.00	44.14	108.71	139.60	-0.17	0.00	4.16	8.00
227.04	0.00	34.46	132.89	139.60	156.57	0.00	4.77	8.00
227.42	0.00	39.77	136.16	139.60	266.79	0.00	3.59	8.00
226.87	0.00	46.45	133.10	139.60	117.02	0.00	4.01	8.00
244.75	0.00	45.46	134.30	139.60	177.86	0.00	3.10	8.00
226.60	0.00	45.40	121.42	139.60	266.84	0.00	4.03	8.00
227.26	0.00	42.98	128.71	139.60	262.67	0.00	3.95	8.00
227.55	0.00	41.81	124.90	139.60	258.95	0.00	3.87	8.00
229.90	0.00	42.08	107.29	139.60	267.50	0.00	3.28	8.00
229.24	0.00	42.15	114.99	139.60	275.78	0.00	2.89	8.00
229.11	0.00	41.84	114.34	139.60	338.73	0.00	3.22	8.00
228.44	0.00	41.86	110.93	139.60	272.16	0.00	2.88	8.00
228.85	0.00	41.85	108.82	139.60	265.55	0.00	3.63	8.00
229.05	0.00	44.42	104.92	139.60	261.47	0.00	2.91	8.00
227.71	**0.00**	**44.73**	**121.52**	**139.60**	**102.54**	**0.00**	**5.13**	**8.00**

Table C.3.5: MP consumers – turbines on the Cat Cracker Complex section

MP consumers													
CCU													
Users													
E6264	C6101	U6500	J6051	Bit Unit	Bit Blendin	Shell chem	U200	IU flare	E222	E226	HDS1B	PA line trac	C201
625FC53	*estimate*	*32F629*	*605FI19*	*32FT50*	*32FI38*	*32FI68*	*32F066*	*36FC6*	*02FC59*	*02FC56*	*02FI23*	*estimate*	*estimate*
10.00	0.00	6.30	27.99	not avail	211.26	22.23	117.87	21.36	17.04	81.73	7.32	10.00	69.00
10.00	0.00	6.59	27.90		187.71	22.44	117.67	40.91	15.39	100.97	7.30	10.00	69.00
10.00	0.00	7.48	27.70		190.43	23.33	120.21	57.35	14.78	96.12	7.33	10.00	69.00
10.00	0.00	7.52	27.66		220.95	22.74	120.10	32.65	21.17	82.53	7.29	10.00	69.00
10.00	0.00	6.38	27.57		229.76	22.04	118.90	18.49	33.38	68.25	7.25	10.00	69.00
10.00	0.00	6.51	27.62		203.14	22.83	120.46	19.53	27.96	69.21	7.28	10.00	69.00
10.00	0.00	5.89	27.93		198.51	23.09	118.91	20.40	27.59	70.47	7.27	10.00	69.00
10.00	0.00	6.68	27.93		201.42	24.74	120.10	32.36	29.04	73.54	7.24	10.00	69.00
10.00	0.00	5.93	27.14		172.98	23.21	130.09	49.12	27.07	77.12	7.25	10.00	69.00
10.00	0.00	6.47	27.90		195.34	23.63	125.57	24.73	25.31	89.99	7.30	10.00	69.00
10.00	0.00	6.60	28.01		202.07	23.17	123.16	17.20	14.68	100.69	7.33	10.00	69.00
10.00	0.00	6.28	28.06		215.91	23.25	126.77	16.31	19.59	86.36	7.30	10.00	69.00
10.00	0.00	6.32	27.80		218.44	22.99	69.54	19.35	16.32	88.25	6.76	10.00	69.00
10.00	0.00	6.42	27.81		219.67	23.01	0.00	30.80	13.57	73.65	4.07	10.00	69.00
10.00	0.00	6.57	27.75		219.67	23.01	0.00	18.05	5.01	104.06	1.84	10.00	69.00
10.00	0.00	6.73	27.76		219.67	23.01	0.76	20.21	3.11	107.24	0.15	10.00	69.00
10.00	0.00	6.95	27.89		219.67	23.01	115.70	17.28	8.44	109.41	7.28	10.00	69.00
10.00	0.00	4.76	27.85		219.67	23.01	137.89	16.33	16.59	77.11	7.26	10.00	69.00
10.00	0.00	1.82	27.91		219.67	23.01	140.17	113.19	16.99	73.06	7.25	10.00	69.00
10.00	0.00	1.08	27.90		219.67	23.01	133.03	103.25	22.56	72.51	7.26	10.00	69.00
10.00	0.00	5.50	27.72		219.67	23.01	125.47	20.80	24.42	71.49	7.28	10.00	69.00
10.00	0.00	5.33	27.73		219.67	23.01	144.91	18.02	23.04	73.20	7.34	10.00	69.00
10.00	0.00	5.65	27.73		219.67	23.01	144.04	21.45	23.55	72.12	7.33	10.00	69.00
10.00	0.00	6.05	27.78		219.67	23.01	143.37	40.00	22.82	71.22	7.36	10.00	69.00
10.00	0.00	3.65	27.76		219.67	23.01	139.65	23.66	29.09	73.78	7.31	10.00	69.00
10.00	0.00	6.91	27.80		219.67	23.01	119.01	23.49	32.14	75.88	7.34	10.00	69.00
10.00	0.00	7.38	27.59		219.67	23.01	110.96	53.31	27.29	78.10	7.30	10.00	69.00
10.00	0.00	7.12	27.71		219.67	23.01	110.94	65.42	40.44	68.79	7.30	10.00	69.00
10.00	0.00	7.46	27.68		219.67	23.01	111.67	17.92	33.89	71.70	7.32	10.00	69.00
10.00	0.00	6.98	27.84		219.67	23.01	109.93	18.41	12.50	83.08	7.30	10.00	69.00
10.00	0.00	7.92	27.91		219.67	23.01	110.15	21.04	12.71	80.33	7.30	10.00	69.00
10.00	**0.00**	**6.11**	**27.78**	**0.00**	**212.97**	**23.03**	**110.55**	**32.66**	**21.21**	**81.35**	**6.77**	**10.00**	**69.00**

Table C.3.6: MP consumers – users on the Cat Cracker Complex section

MP consumers													
Visbreaker													
Turbines		Users											
K8001	P8030/1	F8401	Solvents	J9101/2	J9201/2	S8001	80-RV-6		J8001A/B	J8002A/B	C8001	C8002	C8003
80FI104	*estimate*	*84FI4*	*91FI27*	*91FI20*	*92FI11*	*80FC205*	*80FA160*	*80FA161*	*80FI164*	*80FI175*	*80FC68*	*80FC69*	*80FC71*
158.04	18.00	0.00	66.90	5.12	0.23	10.98	0.06	1.08	18.01	56.08	3.59	10.80	13.10
157.31	18.00	0.00	108.35	0.00	0.22	10.52	0.07	1.07	18.84	55.80	3.59	10.78	13.09
158.12	18.00	0.00	131.71	0.00	0.19	11.07	0.10	1.08	18.76	54.13	3.28	10.80	12.77
154.59	18.00	0.00	131.96	0.00	0.24	10.27	0.07	1.07	18.01	51.08	3.19	10.80	12.54
157.94	18.00	0.00	126.59	0.00	0.23	10.51	0.08	1.07	16.90	48.21	3.15	10.80	12.66
157.04	18.00	0.00	127.94	0.00	0.24	10.90	0.09	1.07	19.17	48.06	3.32	10.79	11.79
151.05	18.00	0.00	115.45	0.00	0.23	9.96	0.06	1.05	21.81	48.18	3.36	10.71	11.63
156.36	18.00	0.00	81.76	0.00	0.23	10.19	0.08	1.06	19.22	47.82	2.72	10.59	12.20
142.84	18.00	0.00	50.09	0.00	0.23	10.19	0.09	1.05	22.42	51.72	2.58	10.39	11.29
154.11	18.00	0.00	73.98	0.00	0.23	10.60	0.07	0.98	18.41	50.05	2.95	10.56	11.95
156.56	18.00	0.00	84.32	0.00	0.23	10.30	0.06	1.06	21.17	50.24	3.10	10.65	11.62
156.05	18.00	0.00	88.03	0.00	0.23	10.57	0.04	1.05	25.22	51.62	3.09	10.79	10.89
157.94	18.00	0.00	86.62	0.00	0.24	11.07	0.07	1.07	15.04	51.99	2.88	10.80	11.03
160.29	18.00	0.00	81.49	0.00	0.22	10.80	0.03	1.06	20.86	51.18	4.03	10.79	12.21
163.15	18.00	0.00	80.41	0.00	0.23	10.63	0.04	1.07	17.06	50.71	5.53	10.53	13.99
159.92	18.00	0.00	12.96	0.98	0.23	11.05	0.05	1.08	21.84	50.70	5.15	10.33	13.49
160.27	18.00	0.00	0.79	2.81	0.22	10.62	0.01	1.06	16.39	50.66	5.79	10.74	13.16
158.37	18.00	0.00	0.58	3.73	0.23	10.05	0.03	1.06	16.15	46.59	4.53	10.70	12.55
161.55	18.00	0.00	0.45	3.74	0.22	10.71	0.02	1.06	15.23	47.45	3.05	10.78	11.42
166.15	18.00	0.00	70.44	3.76	0.23	10.15	0.02	1.07	9.57	41.30	3.30	10.78	11.69
167.20	18.00	0.00	42.05	3.75	0.23	10.13	0.03	1.06	19.61	39.19	3.45	10.77	12.02
165.20	18.00	0.00	0.30	3.72	0.19	10.77	0.06	1.08	19.81	39.94	3.20	10.71	12.27
163.18	18.00	0.00	0.38	3.73	0.17	10.59	0.05	1.08	15.86	42.08	3.91	10.61	12.46
166.55	18.00	0.00	0.81	2.22	0.22	10.97	0.05	1.09	18.59	46.33	6.66	10.50	13.03
78.50	18.00	0.00	0.80	0.00	0.22	10.41	0.08	1.07	19.52	40.30	0.85	5.94	7.37
0.00	18.00	0.00	1.18	0.00	0.23	3.69	0.48	0.01	12.16	29.53	0.00	1.56	1.22
0.00	18.00	0.00	0.96	0.00	0.23	6.48	0.55	0.00	11.50	29.39	0.00	1.50	1.20
0.00	18.00	0.00	1.05	0.00	0.23	6.66	0.49	0.00	15.10	29.20	0.00	1.46	1.21
0.00	18.00	0.00	1.02	0.00	0.23	7.01	0.40	0.00	14.94	29.06	0.00	1.50	1.21
0.00	18.00	0.00	0.90	0.00	0.23	6.98	0.25	0.25	10.33	29.19	0.00	1.45	1.23
0.00	18.00	0.00	1.01	0.00	0.23	6.00	0.01	0.52	5.61	29.46	0.00	1.40	1.21
125.43	**18.00**	**0.00**	**50.69**	**1.08**	**0.22**	**9.70**	**0.12**	**0.88**	**17.20**	**44.75**	**2.91**	**8.75**	**9.98**

Table C.3.7: MP consumers – Visbreaker section

	letdowns					
	LDM					
	LDM Utilities	LDM FCCU	LDM CDU1	LDM Visbreaker	LDM Luboil	total LDM
	Fs	Fs	Fs	Fs	Fs	
TIME	327FI59	326FI37	32FI46	328FI1	46FI13	
01-Mar-00	196.48	391.70	0.03	0.00	0.00	588.21
02-Mar-00	216.98	391.44	0.08	0.00	0.00	608.50
03-Mar-00	223.05	390.08	10.28	0.00	0.00	623.41
04-Mar-00	244.96	390.84	23.89	0.00	0.00	659.69
05-Mar-00	228.90	393.41	0.28	0.00	0.00	622.59
06-Mar-00	231.41	392.57	0.57	0.00	0.32	624.86
07-Mar-00	226.82	392.17	3.11	0.00	0.56	622.66
08-Mar-00	215.02	392.73	1.71	8.95	1.08	619.48
09-Mar-00	216.06	395.74	0.78	4.28	0.00	616.85
10-Mar-00	206.77	396.14	30.30	0.00	0.00	633.22
11-Mar-00	245.71	394.22	45.95	0.00	0.00	685.87
12-Mar-00	253.45	393.24	77.91	0.00	0.00	724.60
13-Mar-00	233.25	393.64	63.55	0.00	0.00	690.44
14-Mar-00	254.47	397.54	52.75	0.00	0.00	704.76
15-Mar-00	273.00	393.06	52.75	0.00	0.00	718.81
16-Mar-00	253.04	386.68	52.75	0.00	0.00	692.48
17-Mar-00	239.27	389.37	52.75	0.00	0.00	681.38
18-Mar-00	214.79	390.56	52.75	0.00	0.00	658.10
19-Mar-00	225.15	397.06	52.75	0.00	0.00	674.97
20-Mar-00	238.53	398.77	52.75	0.00	0.23	690.28
21-Mar-00	234.49	391.13	52.75	0.00	0.00	678.37
22-Mar-00	216.93	393.28	52.75	0.00	0.00	662.96
23-Mar-00	216.67	396.93	52.75	0.00	0.00	666.36
24-Mar-00	198.27	397.70	52.75	0.00	0.00	648.73
25-Mar-00	217.45	395.44	52.75	39.38	0.00	705.03
26-Mar-00	235.60	394.19	52.75	46.26	0.00	728.80
27-Mar-00	251.14	393.81	52.75	71.52	0.00	769.22
28-Mar-00	258.42	397.24	52.75	93.71	2.47	804.59
29-Mar-00	264.43	397.20	52.75	102.94	0.00	817.32
30-Mar-00	272.60	399.62	52.75	92.79	1.24	819.00
31-Mar-00	259.90	396.58	52.75	99.88	0.00	809.11
average	**234.29**	**394.00**	**38.97**	**18.06**	**0.19**	**685.51**

Table C.3.8: MP consumers – Letdown stations

TIME	MP to turbines	MP users	total MP consumed	eM imbalance	% imbalance
01-Mar-00	2150.67	1103.94	3842.83	-970.47	-33.79
02-Mar-00	2087.77	1154.42	3850.69	-901.08	-30.55
03-Mar-00	2108.87	1193.50	3925.79	-1001.34	-34.24
04-Mar-00	2090.39	1186.05	3936.13	-988.46	-33.53
05-Mar-00	2045.83	1164.31	3832.72	-1003.66	-35.48
06-Mar-00	1997.05	1140.78	3762.70	-858.79	-29.57
07-Mar-00	2105.72	1127.64	3856.02	-444.47	-13.03
08-Mar-00	2183.30	1113.51	3916.29	-501.51	-14.69
09-Mar-00	2095.99	1082.38	3795.21	-568.54	-17.62
10-Mar-00	2148.14	1104.26	3885.61	-633.31	-19.47
11-Mar-00	2190.01	1115.81	3991.69	-770.17	-23.91
12-Mar-00	2129.60	1134.70	3988.90	-825.12	-26.08
13-Mar-00	2031.21	1068.64	3790.29	-699.00	-22.61
14-Mar-00	2031.63	993.17	3729.56	-530.31	-16.58
15-Mar-00	1998.71	996.31	3713.82	-425.10	-12.93
16-Mar-00	2026.96	937.33	3656.77	-530.10	-16.95
17-Mar-00	2036.52	1052.39	3770.29	-567.43	-17.72
18-Mar-00	2013.19	1042.69	3713.98	-448.30	-13.73
19-Mar-00	2143.36	1129.46	3947.79	-851.47	-27.50
20-Mar-00	2259.48	1174.76	4124.53	-809.03	-24.40
21-Mar-00	2144.98	1068.36	3891.72	-681.75	-21.24
22-Mar-00	2231.08	1045.45	3939.50	-821.09	-26.33
23-Mar-00	2291.02	1048.02	4005.39	-834.42	-26.31
24-Mar-00	2336.09	1076.35	4061.17	-861.94	-26.94
25-Mar-00	2229.54	1040.24	3974.81	-756.95	-23.52
26-Mar-00	2165.26	993.95	3888.01	-473.27	-13.86
27-Mar-00	2142.69	1017.30	3929.21	-414.37	-11.79
28-Mar-00	2209.20	1036.13	4049.92	-527.53	-14.98
29-Mar-00	2135.42	981.33	3934.08	-601.25	-18.04
30-Mar-00	2052.49	966.26	3837.76	-495.32	-14.82
31-Mar-00	2086.73	963.95	3859.79	-500.41	-14.90
average	**2125.77**	**1072.69**	**3883.97**	**-686.97**	**-21.49**

Table C.3.9: MP steam imbalance

APPENDIX C.4

Overall steam balance of March 2000 – LP header

	LP producers							
	LP feeders							Total
	P3501 + ex Samco	R&R stripper	E6110	E6117 WHB	SRU2	V8016	E8402/3/4	FL
TIME	46FI31 (estimated)	61FI80	61F043	61F074	66FI8	80FI94	84FI16	
01-Mar-00	69.00	26.75	87.01	29.60	124.71	229.76	40.83	607.65
02-Mar-00	69.00	26.76	85.79	25.85	124.38	225.27	40.57	597.62
03-Mar-00	69.00	26.27	77.25	25.37	124.21	213.94	42.52	578.56
04-Mar-00	69.00	26.61	68.92	26.59	124.60	214.73	44.46	574.91
05-Mar-00	69.00	27.01	58.92	64.60	124.83	217.01	44.75	606.13
06-Mar-00	69.00	27.66	44.10	113.67	124.17	219.51	45.44	643.54
07-Mar-00	69.00	27.75	51.78	44.40	128.33	213.59	44.77	579.62
08-Mar-00	69.00	27.40	45.67	93.80	126.10	207.78	43.29	613.04
09-Mar-00	69.00	29.22	54.68	85.14	122.70	218.39	43.36	622.48
10-Mar-00	69.00	28.65	48.70	55.46	124.34	219.16	40.33	585.64
11-Mar-00	69.00	28.38	64.02	11.46	110.81	216.43	43.57	543.68
12-Mar-00	69.00	27.96	53.17	43.08	105.93	214.88	39.28	553.29
13-Mar-00	69.00	28.00	37.71	35.31	119.22	215.08	39.48	543.80
14-Mar-00	69.00	29.29	22.34	53.61	112.19	201.18	37.59	525.20
15-Mar-00	69.00	28.81	19.51	33.42	104.26	192.54	36.49	484.03
16-Mar-00	69.00	26.74	32.52	23.91	107.34	198.38	33.27	491.16
17-Mar-00	69.00	27.46	20.61	31.08	117.34	198.02	36.41	499.93
18-Mar-00	69.00	27.92	19.64	33.67	119.65	201.28	36.46	507.63
19-Mar-00	69.00	27.54	41.51	38.90	123.53	219.00	43.21	562.69
20-Mar-00	69.00	27.67	35.64	45.66	126.16	213.59	42.70	560.43
21-Mar-00	69.00	27.17	44.81	46.63	127.39	220.05	41.47	576.52
22-Mar-00	69.00	27.13	46.22	45.06	123.29	221.58	43.14	575.42
23-Mar-00	69.00	27.41	47.13	44.39	118.65	226.14	43.24	575.97
24-Mar-00	69.00	27.39	45.85	48.50	119.18	261.67	39.82	611.41
25-Mar-00	69.00	27.07	41.97	31.89	121.20	143.59	32.29	467.01
26-Mar-00	69.00	27.91	34.59	42.45	120.68	0.00	23.51	318.14
27-Mar-00	69.00	27.66	31.88	39.67	119.19	1.82	24.54	313.75
28-Mar-00	69.00	26.82	37.53	40.55	119.25	0.00	28.62	321.77
29-Mar-00	69.00	26.40	45.86	38.21	126.87	0.00	31.69	338.02
30-Mar-00	69.00	26.13	33.06	36.40	125.43	0.00	29.46	319.49
31-Mar-00	69.00	25.86	44.35	43.32	123.38	0.00	31.94	337.85
average	69.00	27.44	45.90	44.25	120.62	171.75	38.34	517.30

Table C.4.1: LP local feeders

AssMP turbines					MP turbines				
F K3262	K3272	K3271+Soot Blowers	U3200	U3500	K4402	others	P3701C	P3371	K7101
32X075	327FI21	327FI103	estimate	32F039	44FI42	estimate	calculated	337FI12	71FI111
1.07	32.11	93.69	275.00	20.16	874.47	40.00	336.11	54.09	137.01
1.53	41.13	97.70	275.00	20.58	829.78	40.00	316.98	54.05	129.15
0.23	34.07	96.45	275.00	16.29	848.73	40.00	308.88	54.02	133.05
1.21	30.88	99.07	275.00	20.22	841.12	40.00	312.43	53.97	125.19
0.25	38.11	93.28	275.00	17.92	869.23	40.00	244.27	53.98	117.36
-1.30	31.24	91.65	275.00	16.45	861.71	40.00	205.47	54.11	118.75
1.50	30.51	92.74	275.00	19.00	855.37	40.00	330.83	54.17	119.49
0.02	44.84	95.84	275.00	16.66	869.63	40.00	372.20	54.22	126.69
0.41	23.36	93.28	275.00	17.24	890.80	40.00	270.49	53.63	137.48
-0.20	30.75	92.77	275.00	19.83	875.44	40.00	298.31	53.92	146.50
-1.21	33.80	94.84	275.00	17.60	880.86	40.00	356.21	53.95	118.04
4.18	32.09	91.00	275.00	17.58	822.30	40.00	348.44	53.96	117.14
3.77	31.05	92.55	275.00	16.23	789.42	40.00	281.01	53.94	129.58
4.62	31.32	91.72	275.00	17.79	838.19	40.00	258.67	53.95	123.12
6.27	29.03	92.10	275.00	17.30	807.28	40.00	258.66	53.93	111.19
1.58	31.21	93.19	275.00	16.14	830.59	40.00	255.55	54.00	119.47
3.83	30.69	95.51	275.00	21.80	821.97	40.00	261.92	53.96	124.08
3.45	35.56	96.78	275.00	22.42	808.05	40.00	260.49	54.05	142.62
3.49	31.64	94.76	275.00	20.05	797.69	40.00	249.63	54.04	119.13
5.74	27.57	95.77	275.00	20.46	781.89	40.00	262.16	54.00	115.95
3.79	17.58	94.87	275.00	17.14	845.17	40.00	223.73	53.99	121.84
4.15	35.05	94.08	275.00	17.38	823.79	40.00	251.36	54.07	125.59
6.83	39.78	94.49	275.00	16.29	847.38	40.00	226.20	54.10	130.27
0.81	32.23	94.44	275.00	15.72	852.56	40.00	259.25	54.16	132.41
3.48	41.69	88.23	275.00	17.36	845.46	40.00	262.91	54.28	125.70
3.28	71.33	87.98	275.00	23.24	831.63	40.00	290.65	54.65	132.67
3.06	63.72	87.18	275.00	18.82	812.20	40.00	279.83	54.50	125.52
6.02	7.86	86.23	275.00	18.92	813.97	40.00	275.01	54.50	132.88
5.68	16.50	86.09	275.00	16.00	830.13	40.00	263.23	54.50	125.68
5.83	57.22	84.22	275.00	18.92	748.37	40.00	264.68	54.38	130.77
10.14	36.75	83.27	275.00	18.51	779.62	40.00	263.88	54.28	140.58
3.02	**34.54**	**92.44**	**275.00**	**18.39**	**833.06**	**40.00**	**279.01**	**54.11**	**126.93**

Table C.4.2: LP producers – AMP and MP turbines

MP turbines										
P6110/11/16	P6102A/B	P6115/17/1	P3202	P3501/5	P3263/4	P3261/6	P3262A	U300	K8001	P8030/1
326FI38	*326FI35*	*326FI34*	*32F051*	*32FI67*	*326FI32*	*326FI28*	*32F664*	*estimate*	*80FI104*	*estimate*
227.34	0.00	43.82	108.83	139.60	-0.63	0.00	5.99	8.00	158.04	18.00
227.40	0.00	44.12	117.90	139.60	-0.67	0.00	6.16	8.00	157.31	18.00
225.64	0.00	43.92	125.17	139.60	-0.62	0.00	6.36	8.00	158.12	18.00
226.02	0.00	43.96	121.32	139.60	-0.59	0.00	6.77	8.00	154.59	18.00
225.02	0.00	46.45	119.95	139.60	-0.54	0.00	6.57	8.00	157.94	18.00
225.34	0.00	48.00	113.98	139.60	-0.49	0.00	7.55	8.00	157.04	18.00
227.57	0.00	49.58	105.71	139.60	-0.65	0.00	7.02	8.00	151.05	18.00
227.69	0.00	49.49	115.27	139.60	-0.84	0.00	6.99	8.00	156.36	18.00
223.80	0.00	46.28	119.23	139.60	-0.62	0.00	6.46	8.00	142.84	18.00
227.40	0.00	46.76	134.07	139.60	-0.64	0.00	6.67	8.00	154.11	18.00
227.41	0.00	46.81	138.56	139.60	-0.59	0.00	6.60	8.00	156.56	18.00
227.10	0.00	46.66	145.91	139.60	-0.46	0.00	6.90	8.00	156.05	18.00
225.92	0.00	46.79	134.56	139.60	-0.31	0.00	6.77	8.00	157.94	18.00
227.79	0.00	47.28	110.50	139.60	-0.34	0.00	6.59	8.00	160.29	18.00
226.37	0.00	48.00	118.76	139.60	-0.32	0.00	6.09	8.00	163.15	18.00
225.08	0.00	47.15	124.07	139.60	-0.27	0.00	5.80	8.00	159.92	18.00
226.79	0.00	46.80	131.87	139.60	-0.27	0.00	3.53	8.00	160.27	18.00
227.16	0.00	44.14	108.71	139.60	-0.17	0.00	4.16	8.00	158.37	18.00
227.04	0.00	34.46	132.89	139.60	156.57	0.00	4.77	8.00	161.55	18.00
227.42	0.00	39.77	136.16	139.60	266.79	0.00	3.59	8.00	166.15	18.00
226.87	0.00	46.45	133.10	139.60	117.02	0.00	4.01	8.00	167.20	18.00
244.75	0.00	45.46	134.30	139.60	177.86	0.00	3.10	8.00	165.20	18.00
226.60	0.00	45.40	121.42	139.60	266.84	0.00	4.03	8.00	163.18	18.00
227.26	0.00	42.98	128.71	139.60	262.67	0.00	3.95	8.00	166.55	18.00
227.55	0.00	41.81	124.90	139.60	258.95	0.00	3.87	8.00	78.50	18.00
229.90	0.00	42.08	107.29	139.60	267.50	0.00	3.28	8.00	0.00	18.00
229.24	0.00	42.15	114.99	139.60	275.78	0.00	2.89	8.00	0.00	18.00
229.11	0.00	41.84	114.34	139.60	338.73	0.00	3.22	8.00	0.00	18.00
228.44	0.00	41.86	110.93	139.60	272.16	0.00	2.88	8.00	0.00	18.00
228.85	0.00	41.85	108.82	139.60	265.55	0.00	3.63	8.00	0.00	18.00
229.05	0.00	44.42	104.92	139.60	261.47	0.00	2.91	8.00	0.00	18.00
227.71	**0.00**	**44.73**	**121.52**	**139.60**	**102.54**	**0.00**	**5.13**	**8.00**	**125.43**	**18.00**

Table C.4.3: LP producers – MP turbines

letdowns							
LDAL	LDM						
LDAL Utilities	LDM Utilities	LDM FCCU	LDM CDU1	LDM Visbreaker	LDM Luboil	total LDM	Total LP produced
Fs	Fs	Fs	Fs	Fs	Fs		
327 FI 57	327FI59	326FI37	32FI46	328FI1	46FI13		
5.66	196.48	391.70	0.03	0.00	0.00	588.21	3774.23
57.55	216.98	391.44	0.08	0.00	0.00	608.50	3787.38
105.26	223.05	390.08	10.28	0.00	0.00	623.41	3838.15
39.97	244.96	390.84	23.89	0.00	0.00	659.69	3791.34
12.70	228.90	393.41	0.28	0.00	0.00	622.59	3711.80
3.61	231.41	392.57	0.57	0.00	0.32	624.86	3682.11
23.37	226.82	392.17	3.11	0.00	0.56	622.66	3750.12
82.40	215.02	392.73	1.71	8.95	1.08	619.48	3930.58
29.43	216.06	395.74	0.78	4.28	0.00	616.85	3774.03
7.82	206.77	396.14	30.30	0.00	0.00	633.22	3792.98
32.39	245.71	394.22	45.95	0.00	0.00	685.87	3871.99
52.42	253.45	393.24	77.91	0.00	0.00	724.60	3879.77
103.31	233.25	393.64	63.55	0.00	0.00	690.44	3787.36
51.39	254.47	397.54	52.75	0.00	0.00	704.76	3733.43
32.89	273.00	393.06	52.75	0.00	0.00	718.81	3654.14
83.72	253.04	386.68	52.75	0.00	0.00	692.48	3711.43
104.69	239.27	389.37	52.75	0.00	0.00	681.38	3749.34
100.70	214.79	390.56	52.75	0.00	0.00	658.10	3712.82
70.41	225.15	397.06	52.75	0.00	0.00	674.97	3876.38
35.46	238.53	398.77	52.75	0.00	0.23	690.28	3970.21
81.27	234.49	391.13	52.75	0.00	0.00	678.37	3889.53
90.37	216.93	393.28	52.75	0.00	0.00	662.96	3985.49
66.42	216.67	396.93	52.75	0.00	0.00	666.36	4032.15
25.37	198.27	397.70	52.75	0.00	0.00	648.73	4039.79
157.99	217.45	395.44	52.75	39.38	0.00	705.03	3985.32
296.31	235.60	394.19	52.75	46.26	0.00	728.80	3969.33
268.16	251.14	393.81	52.75	71.52	0.00	769.22	3941.61
176.74	258.42	397.24	52.75	93.71	2.47	804.59	3906.32
129.67	264.43	397.20	52.75	102.94	0.00	817.32	3819.70
211.58	272.60	399.62	52.75	92.79	1.24	819.00	3843.75
67.64	259.90	396.58	52.75	99.88	0.00	809.11	3725.00
84.09	**234.29**	**394.00**	**38.97**	**18.06**	**0.19**	**685.51**	**3836.05**

Table C.4.4: LP producers – LDAL and LDM letdown stations

	Integrated unit						
	users						venting
	E3504	C651	V805 Bit Unit	E702	E314/E315	Oil Mouv	Fs
TIME	*estimate*	*65FC3*	*08FI12*	*07FC3*	*32FI60*	*32FT64*	*32FI55*
01-Mar-00	1.00	39.19	5.06	19.50	104.37	12.12	17.54
02-Mar-00	1.00	40.97	5.06	19.92	106.78	14.02	15.03
03-Mar-00	1.00	42.04	4.79	21.00	105.91	16.03	14.77
04-Mar-00	1.00	40.93	5.15	21.00	106.51	14.92	7.97
05-Mar-00	1.00	37.16	5.39	21.00	106.29	12.81	8.66
06-Mar-00	1.00	34.71	5.21	21.00	109.61	12.08	8.54
07-Mar-00	1.00	29.98	5.01	20.99	126.35	13.29	5.48
08-Mar-00	1.00	30.05	4.90	20.99	140.91	19.78	6.74
09-Mar-00	1.00	30.00	3.53	21.00	138.50	19.00	9.23
10-Mar-00	1.00	30.00	4.39	21.00	109.95	51.23	38.99
11-Mar-00	1.00	28.80	4.17	20.78	118.16	13.61	26.37
12-Mar-00	1.00	25.00	3.98	21.00	118.59	13.43	25.67
13-Mar-00	1.00	25.00	3.81	21.00	95.84	13.88	25.59
14-Mar-00	1.00	25.00	4.53	21.00	92.86	13.18	25.55
15-Mar-00	1.00	26.82	4.38	21.00	114.34	13.97	25.55
16-Mar-00	1.00	31.10	4.27	21.00	116.22	13.46	25.55
17-Mar-00	1.00	27.58	4.24	21.00	119.25	13.69	25.55
18-Mar-00	1.00	27.00	3.99	21.00	120.33	14.59	25.55
19-Mar-00	1.00	27.00	4.48	21.00	108.01	13.46	25.55
20-Mar-00	1.00	27.00	4.41	21.00	104.42	14.59	25.55
21-Mar-00	1.00	28.63	4.46	21.00	119.72	13.83	25.55
22-Mar-00	1.00	27.97	4.34	21.00	121.34	13.94	25.55
23-Mar-00	1.00	27.02	4.41	21.00	112.25	13.74	25.55
24-Mar-00	1.00	27.31	4.31	21.00	100.79	13.96	25.55
25-Mar-00	1.00	8.32	4.51	21.00	104.70	13.83	25.55
26-Mar-00	1.00	0.00	4.08	21.00	108.54	15.15	25.55
27-Mar-00	1.00	0.00	4.17	20.27	116.24	15.03	25.55
28-Mar-00	1.00	0.29	4.02	20.00	114.92	15.23	25.55
29-Mar-00	1.00	0.00	4.14	20.00	108.46	15.05	25.55
30-Mar-00	1.00	0.00	4.41	20.00	104.65	14.69	25.55
31-Mar-00	1.00	0.02	4.65	20.00	99.18	14.82	25.55
average	**1.00**	**24.03**	**4.46**	**20.76**	**112.06**	**15.56**	**21.63**

Table C.4.5: LP consumers – Integrated unit section

Utilities												
users												Venting
O3263	O3262	V6602	U6350	E6403	C6151	E6515	E3277/8	V3287	V3279	S3272	Tank coil	Fs
326FI21	*326FI22*	*estimate*	*32F662*	*64FC57*	*615FC2*	*32F661*	*327FI98*	*327FI76*	*327FI76*	*327fi106*	*estimate*	*327FA92*
70.83	352.15	0.00	258.14	164.42	93.74	54.99	0.19	62.08	62.08	8.99	15.00	1.00
76.10	357.90	0.00	260.43	165.26	98.44	60.27	4.02	62.16	62.16	9.16	15.00	0.31
72.06	381.21	0.00	256.45	165.41	95.67	59.51	1.90	61.65	61.65	9.87	15.00	1.69
64.81	373.98	0.00	260.50	164.97	91.71	56.60	0.19	59.78	59.78	9.17	15.00	0.33
64.74	355.96	0.00	264.14	161.93	87.71	49.54	0.20	59.80	59.80	9.11	15.00	0.79
65.93	354.26	0.00	264.63	160.65	83.35	47.33	0.21	59.92	59.92	9.48	15.00	2.78
61.08	364.37	0.00	269.33	163.63	70.33	38.01	0.21	60.57	60.57	9.12	15.00	2.07
63.09	347.18	0.00	281.18	160.17	72.76	40.96	4.07	56.14	56.14	9.45	15.00	8.18
53.14	314.08	0.00	277.69	147.31	93.14	56.26	0.18	51.13	51.13	9.19	15.00	22.46
50.08	333.85	0.00	265.55	131.59	93.43	54.67	0.18	55.33	55.33	9.22	15.00	3.13
46.65	337.16	0.00	267.15	122.41	84.22	48.17	0.19	57.11	57.11	9.62	15.00	2.37
41.40	319.57	0.00	258.30	128.54	88.94	51.70	0.21	49.68	49.68	9.35	15.00	0.42
43.48	327.72	0.00	251.24	144.45	88.31	51.66	0.22	51.37	51.37	9.87	15.00	2.61
43.31	323.36	0.00	245.14	135.96	85.53	47.27	0.23	49.75	49.75	9.10	15.00	0.07
44.13	319.98	0.00	245.06	139.34	87.35	49.08	0.23	50.50	50.50	9.40	15.00	0.23
42.73	309.28	0.00	246.31	145.99	81.56	44.94	0.24	52.94	52.94	9.50	15.00	0.02
44.79	280.88	0.00	257.58	153.56	95.63	56.31	0.26	53.48	53.48	9.34	15.00	1.40
56.79	349.89	0.00	265.16	160.32	99.75	62.29	0.76	55.92	55.92	9.59	15.00	4.57
56.66	341.30	0.00	259.32	157.28	97.77	59.56	0.21	60.10	60.10	9.29	15.00	9.52
58.46	346.55	0.00	270.80	158.77	99.66	48.63	0.21	60.90	60.90	9.30	15.00	5.07
57.96	353.17	0.00	280.36	161.41	100.40	62.67	0.21	59.62	59.62	9.32	15.00	1.55
56.38	344.04	0.00	283.97	151.83	101.08	62.40	0.20	59.51	59.51	9.67	15.00	0.26
56.51	343.26	0.00	281.52	148.61	101.85	62.98	0.22	59.25	59.25	9.61	15.00	0.13
58.39	311.84	0.00	282.16	153.89	102.33	63.57	0.19	58.31	58.31	9.70	15.00	8.77
57.21	312.76	0.00	276.72	161.10	103.95	63.46	0.22	47.41	47.41	9.58	15.00	0.68
63.30	371.63	0.00	273.73	163.00	104.85	70.33	0.43	33.19	33.19	9.95	15.00	0.03
65.05	374.11	0.00	274.89	159.93	106.51	77.36	0.31	32.50	32.50	9.50	15.00	0.03
66.02	373.44	0.00	272.62	162.54	106.49	78.03	0.27	34.32	34.32	9.56	15.00	2.07
68.73	357.43	0.00	270.49	163.31	106.20	72.63	0.18	34.22	34.22	9.89	15.00	2.88
65.22	362.48	0.00	283.16	160.00	106.47	74.50	0.34	32.80	32.80	9.61	15.00	34.13
67.97	346.43	0.00	292.21	159.84	106.51	77.73	0.29	33.95	33.95	9.52	15.00	1.58
58.16	**343.26**	**0.00**	**267.61**	**154.11**	**94.70**	**58.17**	**0.55**	**52.11**	**52.11**	**9.45**	**15.00**	**3.91**

Table C.4.6: LP consumers – Utilities section

Luboil											
users											
LUBOIL TANK FARM	OM TANK FARM	C4001	C4104	C4103	C4102	C4101	C4201	E4206	E4209A/B	C4202	C4204
324FI5	*324FI6*	*40FC26*	*41FC113*	*41FC114*	*41FC118*	*41FC130*	*42FC2*	*42FI14*	*42FI17*	*42FC8*	*42FC10*
61.10	232.95	28.00	2.00	6.80	21.41	27.54	83.91	238.13	40.58	10.93	12.00
62.02	247.50	27.99	2.17	4.94	21.61	27.78	85.93	238.17	40.93	11.42	12.00
62.71	243.47	27.99	1.93	2.37	21.94	28.09	83.56	237.07	40.74	11.72	12.00
61.80	240.49	28.00	1.50	1.70	22.28	28.55	87.26	237.72	41.29	12.71	12.00
59.45	227.06	28.00	1.50	1.70	22.34	28.56	85.43	237.17	41.68	12.75	12.00
60.63	244.97	28.00	1.56	1.74	22.27	28.48	82.12	236.99	41.22	12.42	12.00
60.70	263.26	28.00	1.80	2.00	22.17	28.43	86.84	238.72	41.26	12.59	12.00
63.18	252.46	28.01	1.87	2.01	22.18	28.39	87.68	238.17	41.68	12.66	12.00
59.66	241.32	28.00	1.80	2.00	22.05	28.22	87.77	239.32	41.26	12.42	12.00
60.53	240.41	28.01	5.24	1.90	21.63	27.77	88.96	240.14	40.61	11.75	12.00
60.21	224.09	28.00	7.00	2.29	20.60	27.68	84.52	239.94	40.21	11.55	12.00
61.13	227.89	28.00	7.26	2.35	19.00	27.84	79.79	239.62	39.46	11.34	12.00
61.59	228.73	28.00	7.27	1.43	19.80	27.81	79.96	239.91	39.71	11.39	12.00
60.83	229.47	29.06	4.74	2.26	21.65	27.87	79.69	239.39	39.87	11.70	12.00
62.56	254.61	30.00	1.00	4.50	21.75	27.95	80.75	239.49	40.38	11.87	12.00
62.54	257.35	30.01	1.12	4.50	21.76	27.94	75.14	236.64	39.98	11.62	12.00
64.41	238.63	30.01	1.60	4.50	21.70	27.99	76.18	234.71	40.32	11.73	12.00
66.20	238.68	30.00	1.60	4.50	21.64	27.91	84.20	237.43	41.07	12.09	12.00
62.18	261.25	30.00	1.60	4.50	21.61	27.83	80.95	239.41	40.49	11.70	12.00
61.98	267.46	30.00	1.60	4.50	21.58	27.79	89.44	241.92	40.82	12.05	12.00
59.47	223.61	30.00	1.60	4.50	21.75	27.94	88.23	241.82	40.70	12.27	12.00
59.95	218.44	29.92	1.60	4.45	21.78	27.95	82.87	240.21	40.21	11.97	12.00
60.93	240.50	29.67	1.60	4.79	21.69	27.85	83.28	237.78	40.10	11.87	12.00
61.43	253.60	29.38	1.60	5.01	21.53	27.65	83.24	228.17	39.85	11.97	12.00
58.89	240.08	29.09	1.60	5.00	21.51	27.62	82.26	219.86	40.85	12.41	12.00
63.34	251.72	29.63	1.60	4.99	21.51	27.72	84.67	209.03	42.41	12.78	12.00
61.68	233.53	29.80	1.60	5.00	21.59	27.74	84.52	206.20	42.90	12.81	12.00
64.90	266.90	29.80	1.16	1.98	21.81	28.01	83.53	205.15	42.86	13.03	12.00
62.28	273.83	29.90	1.44	1.31	21.76	27.92	79.15	218.00	40.95	12.72	12.00
61.06	265.65	29.29	2.25	1.64	21.48	27.61	62.83	244.25	37.29	12.28	12.00
59.70	249.26	29.57	1.44	3.38	21.46	27.61	58.15	247.30	37.09	12.22	12.00
61.58	**244.49**	**29.00**	**2.38**	**3.37**	**21.58**	**27.94**	**82.03**	**234.45**	**40.61**	**12.09**	**12.00**

Table C.4.7: LP consumers – Users on Luboil section

Luboil											
users											Venting
C4306	C4307	C4305	E4433	E4432	C4402	C4404	C4405	E4428	E4405	C4501	Fs
43FI39	*43FC41*	*43FC43*	*44FC54*	*44FC55*	*44FC18*	*44FC25*	*44FC29*	*44FC50*	*44FC9*	*45FC2*	*46FI14*
390.00	13.00	20.00	0.47	9.24	40.00	19.00	9.70	0.67	29.61	10.50	439.12
390.00	13.00	20.00	1.48	9.14	40.00	19.00	8.39	0.44	29.38	10.50	358.02
390.00	13.00	20.00	0.00	9.43	40.00	19.27	7.41	0.39	34.97	10.50	380.51
390.00	13.00	20.00	0.84	9.29	40.00	21.00	8.32	0.58	34.90	10.50	354.00
390.00	13.00	20.00	0.87	7.88	40.00	21.00	12.45	0.49	32.67	10.50	400.77
390.00	13.00	20.00	0.88	3.54	40.92	21.00	7.87	0.68	33.54	10.50	383.35
390.00	13.75	20.00	29.50	3.61	42.00	21.00	12.69	1.48	22.13	10.10	352.69
390.00	14.00	20.00	33.97	3.76	42.01	21.00	8.10	1.59	21.48	0.02	398.48
390.00	14.00	20.00	39.46	3.52	42.01	21.00	15.53	1.29	20.01	1.62	424.98
390.00	14.00	20.00	35.24	3.50	41.99	21.00	15.82	1.59	21.15	7.50	419.07
390.00	14.00	20.00	38.26	3.51	42.00	21.00	15.85	1.21	18.67	7.50	448.95
390.00	14.00	20.00	19.14	3.49	41.99	21.00	15.87	0.77	18.55	7.50	415.72
390.00	14.00	20.00	0.00	3.92	42.01	21.00	15.97	0.85	43.62	7.50	360.60
390.00	14.00	20.00	0.00	9.40	41.99	21.00	15.90	1.08	41.67	7.50	427.58
390.00	14.00	20.00	0.00	9.43	42.00	21.00	15.67	0.83	37.03	7.50	372.47
390.00	14.00	20.00	0.00	9.43	42.01	21.00	15.73	0.68	27.58	7.50	406.16
390.00	14.00	20.00	0.00	9.44	42.01	21.01	15.23	0.55	30.00	7.50	402.14
390.00	14.00	20.00	0.94	9.82	41.99	20.99	14.93	0.41	33.02	7.50	375.55
390.00	14.00	20.00	0.00	5.37	42.00	21.00	14.94	0.40	32.85	7.50	308.41
390.00	14.00	20.00	0.00	4.06	41.99	21.00	9.89	0.50	33.95	7.50	281.72
390.00	14.00	20.00	0.00	9.43	42.00	21.00	6.79	0.53	36.46	7.50	412.12
390.00	14.00	20.00	0.00	9.43	42.00	20.66	7.96	0.82	34.83	7.50	402.53
390.00	14.00	20.00	1.32	8.33	41.99	19.00	7.91	0.67	33.92	7.50	417.24
390.00	14.00	20.00	20.85	7.03	42.00	19.54	7.72	1.32	23.56	7.50	430.71
390.00	14.00	20.00	22.85	7.04	42.00	21.00	8.54	1.60	20.01	7.50	445.20
390.00	14.00	20.00	22.22	6.63	42.01	20.98	5.74	1.05	19.60	7.50	410.06
390.00	14.00	20.00	21.38	6.34	41.99	21.10	7.33	1.57	19.26	7.50	425.42
390.00	14.00	20.00	22.05	6.97	41.99	22.94	6.85	1.36	18.48	7.50	383.59
390.00	14.00	20.00	24.46	6.56	42.00	23.05	8.44	1.08	18.41	7.50	357.62
390.00	14.00	20.00	2.51	8.67	42.00	23.10	10.52	1.70	44.68	7.50	314.55
390.00	14.00	20.00	8.05	7.40	42.00	23.09	10.51	0.98	45.69	7.50	362.60
390.00	**13.80**	**20.00**	**11.18**	**6.92**	**41.64**	**20.96**	**11.12**	**0.94**	**29.41**	**7.73**	**389.42**

Table C.4.8: LP consumers – Rest of the users and venting on the Luboil section

CDU2							
Users							Venting
SWS	S3286	U7300	STRIPPING	SWS	CCU FLAR	CDU2 FLAR	Fs
75FC2	327FI134	327FI107	75FC2	755FC2	366FA14	367FI11	327FA119
16.17	5.66	165.62	16.17	22.63	0.00	5.67	13.59
18.47	6.52	171.54	18.47	28.19	0.00	5.14	13.19
18.44	7.67	182.87	18.44	32.77	0.00	5.03	13.04
18.44	3.08	185.31	18.44	48.23	0.00	4.49	13.03
17.79	2.06	162.74	17.79	40.90	0.00	4.71	13.11
17.31	2.75	161.87	17.31	47.43	0.00	4.73	13.12
16.03	5.11	161.45	16.03	51.24	0.00	4.62	13.13
17.00	11.70	160.82	17.00	50.85	0.00	6.12	13.08
17.97	5.06	175.76	17.97	47.87	0.00	5.90	12.72
18.95	8.02	186.20	18.95	44.36	0.00	4.98	13.03
17.83	5.15	180.40	17.83	45.97	0.00	3.00	13.06
17.11	3.32	189.18	17.11	39.31	0.00	3.86	13.07
17.88	2.77	194.44	17.88	23.71	0.00	4.38	13.20
18.19	4.58	209.37	18.19	25.25	0.00	5.42	13.01
18.69	2.22	207.94	18.69	24.51	0.00	4.33	12.96
18.99	4.70	201.32	18.99	6.88	0.00	4.15	12.99
18.70	6.56	191.13	18.70	27.14	0.00	3.06	13.17
18.73	6.63	190.63	18.73	32.60	0.00	2.61	13.29
17.53	10.27	170.01	17.53	30.58	0.00	4.24	13.21
18.58	6.00	179.61	18.58	30.32	0.00	2.48	13.25
18.13	4.75	188.36	18.13	22.53	0.00	2.77	13.20
17.94	7.52	182.84	17.94	29.51	0.00	2.85	13.15
17.04	4.90	178.99	17.04	47.51	0.00	3.25	13.19
16.49	5.43	177.55	16.49	46.43	0.00	3.44	12.96
20.60	4.91	177.83	20.60	47.01	0.00	2.95	13.06
31.33	8.88	188.55	31.33	49.01	0.00	2.14	13.79
29.63	5.37	192.66	29.63	41.88	0.00	2.30	12.90
29.53	7.77	198.91	29.53	56.63	0.00	3.74	13.12
28.45	3.62	201.37	28.45	54.59	0.00	2.76	13.06
28.45	3.41	206.63	28.45	64.61	0.00	2.98	12.90
28.26	3.59	210.82	28.26	67.34	0.00	2.85	12.98
20.15	5.48	184.93	20.15	39.61	0.00	3.90	13.12

Table C.4.9: LP consumers – Crude Distiller Unit No 2

CCU									Visbreaker				
Users									Users				
C6102	C6103	E6436	V6118/9	V6112	E6259	P6104	S3263	E3264	HCS ex	HCS back	HCS	E8303	Tracing
61FC21	*61FC22*	*643FCA39*	*estimate*	*61FC8*	*625FI11*	*estimate*	*32F665*	*326FI58*	*91FI28*	*91FI24*	*calculated*	*83FC10*	*32F665*
20.00	20.00	55.00	5.00	110.00	9.93	0.00	6.61	2.12	1.10	48.38	-47.27	219.15	6.61
20.01	26.49	55.00	5.00	110.00	9.84	0.00	6.98	2.27	9.57	31.29	-21.72	218.31	6.98
20.00	33.20	55.00	5.00	110.01	9.75	0.00	7.22	3.38	3.93	35.26	-31.32	225.26	7.22
20.00	33.20	55.00	5.00	110.00	9.94	0.00	7.12	2.25	0.14	40.15	-40.00	222.02	7.12
20.01	33.20	55.00	5.00	109.99	10.00	0.00	6.70	3.51	0.00	41.42	-41.42	220.00	6.70
20.01	33.20	55.00	5.00	110.00	9.93	0.00	6.58	3.32	0.16	39.81	-39.65	216.91	6.58
20.00	33.20	55.00	5.00	110.00	9.92	0.00	6.71	1.76	10.05	31.46	-21.41	215.29	6.71
20.00	31.56	55.00	5.00	110.00	10.09	0.00	7.16	2.64	26.93	14.71	12.23	221.60	7.16
20.00	25.00	55.00	5.00	109.55	10.13	0.00	6.31	2.74	28.34	9.94	18.40	228.03	6.31
20.00	25.00	55.00	5.00	109.99	10.06	0.00	6.83	2.68	28.06	11.69	16.38	232.76	6.83
20.00	25.00	55.00	5.00	110.00	9.95	0.00	6.70	2.44	27.77	13.51	14.27	233.24	6.70
20.00	25.00	55.00	5.00	110.00	10.27	0.00	6.53	3.92	27.83	13.05	14.78	225.92	6.53
20.00	25.00	55.00	5.00	110.00	10.17	0.00	6.71	3.71	27.54	13.65	13.89	234.31	6.71
20.00	25.00	55.00	5.00	110.00	10.30	0.00	6.54	4.01	27.38	14.38	13.00	234.30	6.54
20.01	25.00	55.00	5.00	110.00	10.25	0.00	6.63	2.21	27.23	14.71	12.52	232.00	6.63
20.00	25.00	55.00	5.00	110.00	9.87	0.00	6.43	4.72	21.46	14.57	6.88	223.46	6.43
20.01	25.00	55.00	5.00	110.00	9.95	0.00	6.70	1.94	0.10	36.71	-36.61	230.94	6.70
20.00	25.00	55.00	5.00	110.00	9.96	0.00	7.30	1.99	0.09	36.42	-36.33	236.18	7.30
20.00	25.00	55.00	5.00	110.00	10.38	0.00	6.94	3.11	0.10	37.96	-37.86	238.18	6.94
20.01	25.00	55.00	5.00	110.01	10.55	0.00	7.07	0.89	0.01	48.73	-48.71	229.43	7.07
20.00	25.00	55.00	5.00	110.00	10.03	0.00	6.78	0.98	0.04	41.73	-41.69	229.10	6.78
20.00	25.00	55.00	5.00	110.00	10.23	0.00	6.79	1.94	0.00	40.92	-40.92	238.11	6.79
20.00	25.00	55.00	5.00	110.01	10.42	0.00	6.95	3.05	0.04	41.01	-40.97	234.97	6.95
20.00	25.00	55.00	5.00	110.00	10.57	0.00	7.13	2.78	15.21	22.72	-7.51	236.43	7.13
20.01	25.00	55.00	5.00	110.00	10.30	0.00	6.86	3.32	30.00	3.90	26.10	226.21	6.86
20.00	25.00	55.00	5.00	110.00	10.16	0.00	7.69	4.31	29.59	1.93	27.66	223.36	7.69
20.00	25.00	55.00	5.00	110.00	10.08	0.00	7.42	3.58	29.55	3.72	25.83	223.78	7.42
20.00	25.00	55.00	5.00	110.00	10.46	0.00	7.57	4.63	29.70	3.38	26.31	239.26	7.57
20.00	25.00	55.00	5.00	110.00	10.50	0.00	7.35	6.27	29.98	3.36	26.62	233.27	7.35
20.00	25.00	55.00	5.00	110.00	11.10	0.00	7.05	2.46	30.10	3.32	26.78	229.25	7.05
20.00	25.00	55.00	5.00	109.89	10.68	0.00	7.10	2.92	30.05	2.87	27.18	219.59	7.10
20.00	**26.42**	**55.00**	**5.00**	**109.98**	**10.19**	**0.00**	**6.92**	**2.96**	**15.87**	**23.12**	**-7.24**	**228.08**	**6.92**

Table C.4.10: LP consumers – Cat Cracker complex and Visbreaker complex

TIME	Total LP to users	Total LP vented	Total LP consumed	eL Imbalance	% imbalance
01-Mar-00	3319.94	471.26	3791.19	-16.96	-0.45
02-Mar-00	3410.73	386.54	3797.27	-9.89	-0.26
03-Mar-00	3438.82	410.02	3848.84	-10.69	-0.28
04-Mar-00	3419.65	375.33	3794.98	-3.64	-0.10
05-Mar-00	3334.18	423.32	3757.50	-45.70	-1.23
06-Mar-00	3336.87	407.79	3744.65	-62.54	-1.70
07-Mar-00	3411.03	373.37	3784.40	-34.27	-0.91
08-Mar-00	3457.54	426.47	3884.01	46.57	1.18
09-Mar-00	3420.83	469.39	3890.22	-116.19	-3.08
10-Mar-00	3444.27	474.22	3918.49	-125.51	-3.31
11-Mar-00	3361.16	490.74	3851.91	20.08	0.52
12-Mar-00	3297.07	454.87	3751.94	127.82	3.29
13-Mar-00	3304.43	401.99	3706.42	80.94	2.14
14-Mar-00	3295.46	466.22	3761.67	-28.24	-0.76
15-Mar-00	3339.98	411.21	3751.19	-97.05	-2.66
16-Mar-00	3280.89	444.73	3725.62	-14.19	-0.38
17-Mar-00	3257.29	442.26	3699.55	49.79	1.33
18-Mar-00	3398.08	418.96	3817.04	-104.22	-2.81
19-Mar-00	3364.01	356.69	3720.70	155.67	4.02
20-Mar-00	3381.25	325.58	3706.82	263.39	6.63
21-Mar-00	3383.36	452.43	3835.79	53.74	1.38
22-Mar-00	3369.19	441.49	3810.68	174.81	4.39
23-Mar-00	3380.31	456.10	3836.42	195.73	4.85
24-Mar-00	3386.31	477.99	3864.30	175.50	4.34
25-Mar-00	3356.33	484.49	3840.82	144.50	3.63
26-Mar-00	3438.12	449.44	3887.56	81.76	2.06
27-Mar-00	3422.03	463.90	3885.93	55.68	1.41
28-Mar-00	3501.36	424.33	3925.69	-19.37	-0.50
29-Mar-00	3476.62	399.12	3875.73	-56.03	-1.47
30-Mar-00	3496.06	387.13	3883.20	-39.44	-1.03
31-Mar-00	3474.98	402.71	3877.69	-152.68	-4.10
average	3385.75	428.07	3813.81	22.24	0.58

Table C.4.11: LP steam imbalance

APPENDIX D

EXCEL data tables sent to GAMS

	Turbines power					Users power	K6101 ratio	Steam enthalpy	Error term power
TIME	K6102	K6101	G3171	K3471	K301	usehp	y	HFH	errorHP
1	219016.80	262254.80	91504.90	65405.66	461021.20	1907117.72	0.49	3300.00	-2147735.94
2	225307.20	261367.02	93026.99	65348.58	469667.00	1682300.15	0.49	3300.00	-1964004.70
3	226833.60	263076.60	86258.78	65394.57	501087.20	2187933.71	0.49	3300.00	-1741907.91
4	227470.20	260845.15	87182.83	65319.09	499689.60	2067361.28	0.49	3300.00	-2040063.58
5	232068.60	266480.73	96209.61	65271.49	465895.40	1902927.80	0.49	3300.00	-2296634.41
6	230819.40	267046.19	79884.40	65273.24	466651.60	2059970.92	0.50	3300.00	-1963432.82
7	224907.00	265039.86	62968.76	65219.95	491145.20	2004611.66	0.74	3300.00	-2220747.79
8	221433.00	267169.11	65730.25	65305.51	500516.80	1859704.75	0.75	3300.00	-1964621.24
9	205224.60	271038.83	57238.39	65343.03	498273.40	1512243.28	0.76	3300.00	-2058715.80
10	193885.20	264541.86	67436.24	65384.93	482798.60	1805622.65	0.77	3300.00	-1653284.78
11	187398.00	249444.73	68011.68	65418.80	511387.00	1838989.18	0.82	3300.00	-1894486.48
12	183995.40	242833.70	52036.22	65408.00	539545.20	1805974.71	0.84	3300.00	-2239096.09
13	186273.00	247235.99	74964.67	65296.46	495674.60	1793117.27	0.83	3300.00	-1878267.07
14	187704.60	241951.94	83705.57	65226.23	475648.00	1817229.90	0.85	3300.00	-2401786.38
15	214410.00	246611.91	91036.14	65226.96	461792.20	1821081.10	0.83	3300.00	-1907791.68
16	226085.40	252318.63	89965.78	65231.78	463291.60	1394335.95	0.81	3300.00	-1485796.15
17	224420.40	252875.74	75959.72	65336.46	510953.60	1818172.69	0.81	3300.00	-1567471.46
18	217719.60	255225.93	54999.38	65363.91	547876.60	1836473.14	0.80	3300.00	-1925728.47
19	211709.40	258474.92	82260.19	65237.91	457842.00	1824303.87	0.79	3300.00	-2094322.87
20	185190.00	255008.50	99583.88	65242.14	453243.80	1849010.71	0.80	3300.00	-1993409.83
21	199606.20	261592.14	59383.04	65320.55	504826.80	1847860.86	0.78	3300.00	-2315248.06
22	204357.60	265177.96	69953.63	65258.93	492872.00	1841988.65	0.77	3300.00	-2062478.55
23	209225.40	264853.65	80472.79	65205.64	469409.60	1843062.23	0.77	3300.00	-2155978.03
24	221206.20	265065.24	77849.48	65205.06	449623.40	1497812.30	0.77	3300.00	-1679387.10
25	218706.00	260381.06	75729.85	65249.74	469796.00	1075008.25	0.79	3300.00	-1981310.45
26	207456.60	255026.87	83406.49	65283.32	497227.80	1144577.80	0.85	3300.00	-1465831.10
27	211378.20	264532.68	85250.02	65239.66	482900.80	1111839.70	0.92	3300.00	-1633251.00
28	217362.00	263854.16	76208.38	65314.12	511128.20	1067075.30	0.92	3300.00	-1672331.40
29	226570.20	267629.86	65027.94	65363.18	523260.60	1077015.63	0.85	3300.00	-1999670.93
30	227601.00	259339.81	59791.85	65412.09	542318.60	1192954.44	0.90	3300.00	-1810758.84
31	232476.60	257101.84	59069.35	65381.43	523306.20	1204574.45	0.91	3300.00	-1812678.35
average	213155.40	259206.37	75874.43	65306.08	490989.37	1667492.00	0.75	3300.00	-1936394.49
Power in kW	2467.08	3000.07	878.18	755.86	5682.75	19299.68			-22411.97

Table D.1: HP header data required for the GAMS program

	Turbines power					Users power	Error term power
TIME	K3262	K3272	K3271	U3200	U3500	useap	errorAP
1	325.63	9729.00	28387.22	83325.00	6109.33	407841.11	5289274.08
2	464.20	12462.84	29603.28	83325.00	6234.86	407926.33	5031063.01
3	69.51	10323.36	29224.71	83325.00	4935.57	418245.45	5345603.97
4	366.36	9355.64	30017.27	83325.00	6126.45	410972.80	5348798.27
5	76.84	11547.12	28262.93	83325.00	5429.73	409534.78	5463718.52
6	0.00	9466.11	27771.01	83325.00	4984.02	416479.89	5118526.21
7	453.50	9245.26	28101.13	83325.00	5756.64	412259.03	4696835.90
8	6.15	13587.70	29040.00	83325.00	5047.31	416131.03	4684960.91
9	123.90	7076.93	28264.75	83325.00	5223.27	412839.56	5241740.91
10	0.00	9318.55	28110.70	83325.00	6008.97	412639.84	5365207.47
11	0.00	10242.46	28736.52	83325.00	5333.38	416674.28	5419009.33
12	1266.90	9724.06	27573.39	83325.00	5327.86	422484.95	5270377.29
13	1141.43	9407.27	28043.59	83325.00	4917.75	417377.32	5250136.47
14	1400.56	9490.90	27790.28	83325.00	5391.28	414450.68	5128554.77
15	1900.11	8794.79	27906.15	83325.00	5242.72	411524.04	5021529.77
16	477.38	9457.05	28235.54	83325.00	4889.03	415406.70	5127109.80
17	1160.10	9300.07	28939.05	83325.00	6604.61	416519.83	5309139.84
18	1044.50	10775.13	29323.04	83325.00	6792.56	421097.53	5337606.92
19	1056.83	9588.37	28713.61	83325.00	6075.45	413068.58	5528582.66
20	1740.43	8352.86	29019.76	83325.00	6200.14	408908.98	5062770.40
21	1149.82	5327.83	28744.55	83325.00	5192.78	420415.80	5131125.08
22	1256.42	10619.09	28506.54	83325.00	5266.47	417270.80	5288086.21
23	2068.28	12054.31	28631.71	83325.00	4934.78	412461.42	5144181.19
24	244.04	9766.17	28616.41	83325.00	4761.68	411545.35	5028946.14
25	1052.96	12631.52	26734.33	83325.00	5260.14	407968.94	4720142.22
26	994.57	21612.17	26657.18	83325.00	7040.87	402997.12	4378929.87
27	928.57	19306.13	26414.90	83325.00	5703.34	401013.18	4479925.12
28	1822.85	2380.10	26128.99	83325.00	5732.18	397287.64	5003372.53
29	1719.95	4998.14	26085.00	83325.00	4848.52	393815.09	5309372.85
30	1766.07	17338.05	25517.87	83325.00	5732.76	390979.00	4947246.10
31	3072.03	11136.22	25232.05	83325.00	5609.11	384976.60	5393448.46
average	**940.32**	**10465.01**	**28010.76**	**83325.00**	**5571.40**	**410423.02**	**5414661.69**
Power in kW	**10.88**	**121.12**	**324.20**	**964.41**	**64.48**	**4750.27**	**62669.70**

Table D.2: AMP header data required for the GAMS program

	Turbines power									
TIME	K4402	others	P3701C	P3371	K7101	P6110	P6102A	P6115	P3202	P3501
1	297319.12	13600.00	114278.76	18389.75	46582.38	77296.62	0.00	14897.27	37003.56	47464.00
2	282124.18	13600.00	107772.52	18376.76	43910.32	77315.66	0.00	15000.19	40084.30	47464.00
3	288567.86	13600.00	105018.52	18368.26	45235.98	76718.28	0.00	14934.43	42558.48	47464.00
4	285981.48	13600.00	106226.20	18351.43	42563.24	76847.14	0.00	14945.52	41249.82	47464.00
5	295537.52	13600.00	83050.51	18353.54	39901.04	76508.16	0.00	15793.95	40782.66	47464.00
6	292982.42	13600.00	69860.48	18397.50	40373.98	76613.90	0.00	16318.95	38753.54	47464.00
7	290826.82	13600.00	112480.84	18418.85	40625.24	77372.10	0.00	16855.74	35940.04	47464.00
8	295675.56	13600.00	126549.36	18435.96	43072.90	77415.62	0.00	16825.34	39190.10	47464.00
9	302871.32	13600.00	91965.92	18233.83	46744.56	76093.02	0.00	15735.00	40537.86	47464.00
10	297650.62	13600.00	101425.33	18332.19	49811.02	77314.98	0.00	15897.79	45585.16	47464.00
11	299493.76	13600.00	121111.40	18342.90	40134.96	77320.76	0.00	15914.07	47109.72	47464.00
12	279583.36	13600.00	118468.92	18345.69	39826.58	77212.98	0.00	15862.87	49610.42	47464.00
13	268402.12	13600.00	95543.40	18338.27	44056.86	76813.82	0.00	15907.10	45750.74	47464.00
14	284984.94	13600.00	87946.44	18343.48	41860.12	77446.90	0.00	16076.08	37568.64	47464.00
15	274475.20	13600.00	87944.06	18335.52	37805.28	76964.44	0.00	16319.29	40378.74	47464.00
16	282398.90	13600.00	86888.26	18360.88	40618.44	76527.88	0.00	16029.71	42183.46	47464.00
17	279468.44	13600.00	89054.16	18347.90	42186.86	77107.92	0.00	15912.17	44834.44	47464.00
18	274737.34	13600.00	88567.28	18376.56	48492.16	77235.76	0.00	15006.44	36961.40	47464.00
19	271212.90	13600.00	84873.76	18374.52	40503.18	77192.92	0.00	11715.96	45183.28	47464.00
20	265843.96	13600.00	89133.35	18361.39	39421.30	77324.16	0.00	13522.45	46292.70	47464.00
21	287356.10	13600.00	76068.23	18357.55	41426.96	77137.16	0.00	15791.98	45255.02	47464.00
22	280088.94	13600.00	85462.67	18382.30	42700.26	83214.32	0.00	15456.23	45663.36	47464.00
23	288109.88	13600.00	76908.44	18392.91	44292.48	77043.32	0.00	15437.53	41281.44	47464.00
24	289868.70	13600.00	88143.30	18413.75	45018.04	77269.76	0.00	14614.53	43760.38	47464.00
25	287457.08	13600.00	89389.74	18455.37	42738.00	77366.66	0.00	14215.16	42466.00	47464.00
26	282755.56	13600.00	98819.98	18582.26	45108.14	78164.98	0.00	14307.51	36478.60	47464.00
27	276146.64	13600.00	95143.22	18529.73	42676.46	77939.90	0.00	14330.73	39097.96	47464.00
28	276750.82	13600.00	93501.90	18530.85	45177.84	77898.42	0.00	14225.84	38874.92	47464.00
29	282243.86	13600.00	89498.03	18531.05	42731.54	77670.28	0.00	14233.52	37716.54	47464.00
30	254445.80	13600.00	89990.52	18487.67	44463.16	77809.34	0.00	14228.73	36998.12	47464.00
31	265069.10	13600.00	89718.38	18454.96	47796.86	77878.36	0.00	15102.60	35673.82	47464.00
average	**283239.69**	**13600.00**	**94864.64**	**18396.89**	**43156.65**	**77420.50**	**0.00**	**15206.92**	**41316.94**	**47464.00**
Power in kW	**3278.24**	**157.41**	**1097.97**	**212.93**	**499.50**	**896.07**	**0.00**	**176.01**	**478.21**	**549.35**

Table D.3: MP header data required for the GAMS program – part I

Turbines power						Users power	Local feeder		Error term power
P3263	P3261	P3262A	U300	K8001	P8030	usemp	FM	HFM	errorMP
0.00	0.00	2035.24	2720.00	53734.96	6120.00	375340.21	620.24	3100.00	-329959.02
0.00	0.00	2094.88	2720.00	53485.74	6120.00	392501.47	612.30	3100.00	-306367.54
0.00	0.03	2161.72	2720.00	53760.12	6120.00	405790.65	648.97	3100.00	-340454.65
0.00	0.00	2303.19	2720.00	52561.28	6120.00	403257.48	632.47	3100.00	-336076.43
0.00	0.14	2233.36	2720.00	53700.96	6120.00	395864.52	628.55	3100.00	-341244.81
0.00	0.07	2566.29	2720.00	53393.94	6120.00	387865.51	633.07	3100.00	-291987.07
0.00	0.10	2385.92	2720.00	51357.00	6120.00	383398.38	581.67	3100.00	-151119.46
0.00	0.07	2375.68	2720.00	53163.42	6120.00	378594.32	563.31	3100.00	-170514.22
0.00	0.07	2196.06	2720.00	48565.26	6120.00	368007.57	554.86	3100.00	-193303.40
0.00	0.03	2269.02	2720.00	52395.70	6120.00	375447.45	537.07	3100.00	-215324.07
0.00	0.00	2243.63	2720.00	53229.04	6120.00	379374.28	492.36	3100.00	-261857.63
0.00	0.03	2344.98	2720.00	53057.68	6120.00	385797.66	468.69	3100.00	-280541.51
0.00	0.00	2303.26	2720.00	53699.60	6120.00	363337.60	470.87	3100.00	-237661.53
0.00	0.00	2240.67	2720.00	54498.26	6120.00	337677.12	490.00	3100.00	-180307.03
0.00	0.00	2070.67	2720.00	55471.00	6120.00	338744.38	516.89	3100.00	-144535.50
0.00	0.03	1972.48	2720.00	54374.16	6120.00	318692.30	471.05	3100.00	-180232.98
0.00	0.00	1198.77	2720.00	54491.12	6120.00	357812.91	527.61	3100.00	-192927.22
0.00	0.00	1415.90	2720.00	53846.14	6120.00	354513.10	619.19	3100.00	-152422.88
53232.78	0.00	1622.41	2720.00	54927.00	6120.00	384018.00	618.90	3100.00	-289500.79
90708.94	0.00	1221.72	2720.00	56490.32	6120.00	399418.40	590.12	3100.00	-275071.46
39787.48	0.00	1362.14	2720.00	56846.64	6120.00	363243.01	563.80	3100.00	-231794.80
60471.38	0.00	1055.22	2720.00	56169.02	6120.00	355454.26	572.17	3100.00	-279169.21
90724.92	0.00	1369.18	2720.00	55481.54	6120.00	356326.12	593.39	3100.00	-283702.43
89308.14	0.00	1343.65	2720.00	56628.02	6120.00	365960.09	611.78	3100.00	-293059.09
88042.32	0.00	1317.43	2720.00	26691.53	6120.00	353683.06	573.12	3100.00	-257361.74
90950.68	0.00	1115.68	2720.00	0.00	6120.00	337943.85	501.05	3100.00	-160912.99
93764.18	0.00	981.82	2720.00	0.00	6120.00	345880.84	506.43	3100.00	-140886.58
115169.22	0.00	1095.17	2720.00	0.00	6120.00	352283.04	534.12	3100.00	-179358.74
92535.42	0.00	979.20	2720.00	0.00	6120.00	333652.68	545.48	3100.00	-204426.22
90285.64	0.00	1233.86	2720.00	0.00	6120.00	328528.60	485.87	3100.00	-168409.89
88900.14	0.00	990.01	2720.00	0.00	6120.00	327741.88	508.66	3100.00	-170140.15
34963.91	**0.02**	**1745.14**	**2720.00**	**42645.79**	**6120.00**	**364714.54**	**557.23**	**3100.00**	**-233568.74**
404.67	**0.00**	**20.20**	**31.48**	**493.59**	**70.83**	**4221.23**			**-2703.34**

Table D.4: MP header data required for the GAMS program – part 2

	Local feeder		Users power	Error term power
TIME	FL	HFL	uselp	errorlp
1	607.65	2760.00	7835048.96	-40018.99
2	597.62	2760.00	8049326.10	-23345.83
3	578.56	2760.00	8115613.31	-25220.38
4	574.91	2760.00	8070363.85	-8595.12
5	606.13	2760.00	7868660.08	-107842.56
6	643.54	2760.00	7875011.08	-147601.72
7	579.62	2760.00	8050026.08	-80887.82
8	613.04	2760.00	8159803.84	109907.56
9	622.48	2760.00	8073158.33	-274205.10
10	585.64	2760.00	8128467.52	-296200.30
11	543.68	2760.00	7932348.22	47390.69
12	553.29	2760.00	7781079.77	301663.46
13	543.80	2760.00	7798450.32	191022.41
14	525.20	2760.00	7777278.76	-66654.90
15	484.03	2760.00	7882348.32	-229036.58
16	491.16	2760.00	7742902.52	-33478.02
17	499.93	2760.00	7687197.08	117511.95
18	507.63	2760.00	8019466.20	-245959.67
19	562.69	2760.00	7939063.60	367389.46
20	560.43	2760.00	7979739.38	621592.85
21	576.52	2760.00	7984737.39	126815.07
22	575.42	2760.00	7951286.98	412547.82
23	575.97	2760.00	7977536.56	461926.58
24	611.41	2760.00	7991684.52	414169.38
25	467.01	2760.00	7920936.91	341015.52
26	318.14	2760.00	8113972.17	192958.79
27	313.75	2760.00	8075991.51	131405.27
28	321.77	2760.00	8263208.42	-45705.88
29	338.02	2760.00	8204822.26	-132234.10
30	319.49	2760.00	8250708.68	-93083.59
31	337.85	2760.00	8200952.09	-360332.82
average	517.30	2760.00	771629.29	41569.15
Power in kW			8930.89	481.12

Table D.5: LP header data required for the GAMS program

APPENDIX E

GAMS programs *single_solve.gms* and *multi_solve.gms* with the March 2000 data set

Note that characters in bold represent additions made in *single_solve.gms* to lead to *multi_solve.gms*. So bold text is only appearing in *multi_solve.gms*.

```
$title MINLP formulation of SAPREF steam distribution
$offlisting

Sets
      t time /1*31/
      le header level / h,a,m,l /
      dhp data from HP level / K6102,K6101,G3171,K3471,K301,usehp,y,HFH,errorHP /
      dap data from AssMP level / K3262,K3272,K3271,U3200,U3500,useap,errorAP /
      dmp data from MP level / K4402,others,P3701C,P3371,K7101,P6110,P6102A,P6115,
                     P3202,P3501,P3263,P3261,P3262A,U300,K8001,P8030,
                     usemp,FM,HFM,errorMP /
      dlp data from LP steam level / FL,HFL,uselp,errorLP /
      ht(dhp) hp turbines / K6102,K6101,G3171,K3471,K301 /
      at(dap) assmp turbines / K3262,K3272,K3271,U3200,U3500 /
      mt(dmp) mp turbines / K4402,others,P3701C,P3371,K7101,P6110,P6102A,P6115,
                    P3202,P3501,P3263,P3261,P3262A,U300,K8001,P8030 /
      n0 number of initial values from the plant / FH0,Uh0,Ua0,Um0,Ul0,Eh0,Ea0,Em0,El0,
                   LDH0_K3471,LDH0_G3171,LDH0_K301,LDHM0,LDA0,LDAL0,LDM0,vent0,
                   Hh0,Ha0,Hm0,Hl0 /;

****************************** Data required from excel spreadsheet *******************

* THP steam Data include from the file HP_mass_balance_march.xls

parameter THP(t,dhp) data dhp from the plant
$libinclude xlimport THP HP_mass_balance_march.xls a59:j90

* TAP steam Data include from the file AssMP_mass_balance_march.xls

parameter TAP(t,dap) data dap from the plant
$libinclude xlimport TAP AssMP_mass_balance_march.xls a64:h95

* TMP steam Data include from the file MP_mass_balance_march.xls

parameter TMP(t,dmp) data dmp from the plant
$libinclude xlimport TMP MP_mass_balance_march.xls a56:u87

* TLP steam Data include from the file LP_mass_balance_march.xls

parameter TLP(t,dlp) data dlp from the plant
$libinclude xlimport TLP LP_mass_balance_march.xls a51:e82

********************************** Turbines and users power ***************************

* HP steam Data include from the file HP_mass_balance_march.xls

parameter HP(dhp) data dhp from the plant
$libinclude xlimport HP HP_mass_balance_march.xls b59:j60

* AP steam Data include from the file AssMP_mass_balance_march.xls
```

```
parameter AP(dap) data dap from the plant
$libinclude xlimport AP AssMP_mass_balance_march.xls b64:h65

* MP steam Data include from the file MP_mass_balance_march.xls

parameter MP(dmp) data dmp from the plant
$libinclude xlimport MP MP_mass_balance_march.xls b56:u57

* LP steam Data include from the file LP_mass_balance_march.xls

parameter LP(dlp) data dlp from the plant
$libinclude xlimport LP LP_mass_balance_march.xls b51:e52

**************************** Constants *******************************************

Scalars h0 enthalpy of condensate steam at users outlets in kJ per kg / 400 /
     hw enthalpy of the water coming through desuperheaters / 210 /
     Csteam cost of HP steam production in rands per ton / 112.6 /
     a cost of one kWh / 0.174 /
     D project lifetime in years / 30 /
     eff_elec electrical drive efficiency / 0.95 /
     deltaH_K3471 enthalpy drop across K3471 in kJ per kg / 146 /
     deltaH_G3171 enthalpy drop across G3171 in kJ per kg / 127 /
     deltaH_K6101 enthalpy drop across K6101 in kJ per kg / 167 /
     deltaH_K6102 enthalpy drop across K6102 in kJ per kg / 600 /
     deltaH_K301 enthalpy drop across K301 in kJ per kg / 200 /
     deltaH_ap enthalpy drop across AP turbines in kJ per kg / 303 /
     deltaH_mp enthalpy drop across MP turbines in kJ per kg / 340 /
     Xwater_K3471 % of water coming at the desuperheater K3471 / 0.0319 /
     Xwater_G3171 % of water coming at the desuperheater G3171 / 0.0481 /
     Xwater_K301 % of water coming at the desuperheater K301 / 0.0130 /
     Xwater_LDHM % of water coming at the desuperheater LDHM / 0.0692 /;

**************************** Parameters *******************************************

*** Calculation of Celec ***

parameter Celec_hp(ht) cost of electricity in running an electrical drive on hp header;
Celec_hp(ht) = 24*365*a*(HP(ht)/86.4)/eff_elec;

parameter Celec_ap(at) cost of electricity in running an electrical drive on ap header;
Celec_ap(at) = 24*365*a*(AP(at)/86.4)/eff_elec;

parameter Celec_mp(mt) cost of electricity in running an electrical drive on mp header;
Celec_mp(mt) = 24*365*a*(MP(mt)/86.4)/eff_elec;

*** Calculation of Mst ***

parameter Mst_hp(ht) cost of maintenance for HP steam turbine;
Mst_hp(ht) = 14.954*(HP(ht)/86.4) + 18.968;

parameter Mst_ap(at) cost of maintenance for AP steam turbine;
Mst_ap(at) = 14.954*(AP(at)/86.4) + 18.968;

parameter Mst_mp(mt) cost of maintenance for MP steam turbine;
Mst_mp(mt) = 14.954*(MP(mt)/86.4) + 18.968;

*** Calculation of Melec ***

parameter Melec_hp(ht) cost of maintenance for an electrical drive on HP header;
```

```
Melec_hp(ht) = 285*(HP(ht)/86.4);

parameter Melec_ap(at) cost of maintenance for an electrical drive on AP header;
Melec_ap(at) = 285*(AP(at)/86.4);

parameter Melec_mp(mt) cost of maintenance for an electrical drive on MP header;
Melec_mp(mt) = 285*(MP(mt)/86.4);

*** Calculation of Replacement cost ***

parameter Rhp(ht) cost of replacing a steam turbine on HP header;
Rhp(ht) = 19333*(HP(ht)/(86.4*1000))**2 + 2000000*(HP(ht)/(86.4*1000))+505185;

parameter Rap(at) cost of replacing a steam turbine on AP header;
Rap(at) = 19333*(AP(at)/(86.4*1000))**2 + 2000000*(AP(at)/(86.4*1000))+505185;

parameter Rmp(mt) cost of replacing a steam turbine on MP header;
Rmp(mt) = 19333*(MP(mt)/(86.4*1000))**2 + 2000000*(MP(mt)/(86.4*1000))+505185;

*** calculation of the salvage value ***

parameter Vhp(ht) salvage value of HP turbines;
Vhp(ht) = 0.1*(819.2*(HP(ht)/86.4)**1.2423);

parameter Vap(at) salvage value of AP turbines;
Vap(at) = 0.1*(819.2*(AP(at)/86.4)**1.2423);

parameter Vmp(mt) salvage value of MP turbines;
Vmp(mt) = 0.1*(819.2*(MP(mt)/86.4)**1.2423);

**************************** Variables statement ************************************

Variables
        z new cost
        zsteam steam turbines use cost
        zelec electrical drives use cost
        zrepl replacement of steam turbines cost
        salv net realizable value of existing turbines

        epshp(ht) decision for turbine on hp level
        epsap(at) decision for turbine on assmp level
        epsmp(mt) decision for turbine on mp level

        inthp(ht) intermediate continuous variable
        intap(at) intermediate continuous variable
        intmp(mt) intermediate continuous variable

        FH steam production on level H

        Fht(ht) steam consumption of hp steam turbines
        Fat(at) steam consumption of assmp steam turbines
        Fmt(mt) steam consumption of mp steam turbines

        Uh steam consumption for users on hp level
        Ua steam consumption for users on ap level
        Um steam consumption for users on mp level
        Ul steam consumption for users on lp level

        Eh mass imbalance for header HP
        Ea mass imbalance for header AP
```

```
        Em mass imbalance for header MP
        El mass imbalance for header LP

        LDH_K3471 letdown from the H level to the Ass MP level at K3471
        LDH_G3171 letdown from the H level to the Ass MP level at G3171
        LDH_K301 letdown from the H level to the Ass MP level at K301
        LDHM double letdown from H level to MP level
        LDA letdown from the Ass MP level to the MP level
        LDAL double letdown from A level to LP level
        LDM letdown from the MP level to the LP level
        Vent venting from level LP

        w_K3471 water coming at the K3471 outlet desuperheater
        w_G3171 water coming at the G3171 outlet desuperheater
        w_K301 water coming at the K301 outlet desuperheater
        w_LDHM water coming at the LDHM desuperheater

        H(le) enthalpy on header le

        Hout_K3471 outlet enthalpies for K3471 turbine
        Hout_G3171 outlet enthalpies for G3171 turbine
        Hout_K6101 outlet enthalpies for K6101 turbine
        Hout_K6102 outlet enthalpies for K6102 turbine
        Hout_K301 outlet enthalpies for K301 turbine
        Hout_ap(at) outlet enthalpies for AP turbines
        Hout_mp(mt) outlet enthalpies for MP turbines;

binary variable epshp, epsap, epsmp;
positive variable FH,Fht,Fat,Fmt,Uh,Ua,Um,Ul,LDH_K3471,LDH_G3171,LDH_K301,LDHM,
        LDA,LDAL,LDM,Vent,w_K3471,w_G3171,w_K301,w_LDHM;
positive variable H,Hout_K3471,Hout_G3171,Hout_K6101,Hout_K6102,Hout_K301,
        Hout_ap,Hout_mp,inthp,intap,intmp;

**************************** Equations statement ***********************************

Equations

        cost objective function
        cost_steam cost of running steam turbines
        cost_elec cost of running electrical drives
        replacement cost of installation and purchase of electrical drives
        salvage net realizable values of existing steam turbines

        mass_balance_HP steam balance on level H
        mass_balance_AP steam balance on level A
        mass_balance_MP steam balance on level M
        mass_balance_LP steam balance on level L

        energy_balance_HP energy balance on level H
        energy_balance_AP energy balance on level A
        energy_balance_MP energy balance on level M
        energy_balance_LP energy balance on level L

        HP_outlet_enthalpy_K3471 calculation of K3471 turbines outlet enthalpy
        HP_outlet_enthalpy_G3171 calculation of G3171 turbines outlet enthalpy
        HP_outlet_enthalpy_K6101 calculation of K6101 turbines outlet enthalpy
        HP_outlet_enthalpy_K6102 calculation of K6102 turbines outlet enthalpy
        HP_outlet_enthalpy_K301 calculation of K301 turbines outlet enthalpy
        AP_outlet_enthalpy(at) calculation of ap turbines outlet enthalpies
        MP_outlet_enthalpy(mt) calculation of mp turbines outlet enthalpies
```

desuper_K3471 calculation of water coming through K3471 outlet desuperheater
desuper_G3171 calculation of water coming through G3171 outlet desuperheater
desuper_K301 calculation of water coming through K301 outlet desuperheater
desuper_LDHM calculation of water coming through LDHM desuperheater

interhp(ht) intermediate continuous variable equal to epshp
interap(at) intermediate continuous variable equal to epsap
intermp(mt) intermediate continuous variable equal to epsmp

constraint_users_hp power required for users on hp level
constraint_users_ap power required for users on hp level
constraint_users_mp power required for users on mp level
constraint_users_lp power required for users on lp level

constraint_error_hp energy imbalance term on hp header
constraint_error_ap energy imbalance term on ap header
constraint_error_mp energy imbalance term on mp header
constraint_error_lp energy imbalance term on lp header

constraint_hp_K3471 power required for K3471 turbine
constraint_hp_G3171 power required for G3171 turbine
constraint_hp_K6101 power required for K6101 turbine
constraint_hp_K6102 power required for K6102 turbine
constraint_hp_K301 power required for K301 turbine
constraint_ap(at) power required for ap turbine
constraint_mp(mt) power required for mp turbine

generator_G3171 keep it switched on;

**************************** Mathematical model ************************************

*** Objective function to minimize ***

cost.. z =e= 365*FH*Csteam + zsteam + zelec + zrepl - salv ;

*** cost functions involved in z ***

cost_steam.. zsteam =e= sum(ht, (1-inthp(ht))*Mst_hp(ht)) + sum(at, (1-intap(at))*Mst_ap(at)) + sum(mt, (1-intmp(mt))*Mst_mp(mt));

cost_elec.. zelec =e= sum(ht, inthp(ht)*(Celec_hp(ht)+Melec_hp(ht))) + sum(at, intap(at)*(Celec_ap(at)+Melec_ap(at))) + sum(mt, intmp(mt)*(Celec_mp(mt)+Melec_mp(mt)));

replacement.. zrepl =e= sum(ht, inthp(ht)*Rhp(ht)/D) + sum(at, intap(at)*Rap(at)/D) + sum (mt, intmp(mt)*Rmp(mt)/D);

salvage.. salv =e= sum(ht, inthp(ht)*Vhp(ht)/D) + sum(at, intap(at)*Vap(at)/D) + sum (mt, intmp(mt)*Vmp(mt)/D);

*** Mass balances ***

mass_balance_HP.. FH - sum(ht,(1-inthp(ht))*Fht(ht)) - Uh - LDH_K301 - LDH_G3171 - LDH_K3471 - LDHM - Eh =e= 0;

mass_balance_AP.. ((1-inthp('K301'))*Fht('K301') + LDH_K301 + w_K301) + ((1-inthp('K3471'))*Fht('K3471') + LDH_K3471 + w_K3471) + ((1-inthp('G3171'))*Fht('G3171') + LDH_G3171 + w_G3171) - sum(at,(1-intap(at))*Fat(at)) - LDA - LDAL - Ua - Ea =e= 0;

```
mass_balance_MP.. MP('FM') + (1-inthp('K6101'))*Fht('K6101')*HP('y') + LDA + LDHM + w_LDHM
- sum(mt,(1-intmp(mt))*Fmt(mt)) - LDM - Um - Em =e= 0;

mass_balance_LP.. LP('FL') + sum(mt,(1-intmp(mt))*Fmt(mt)) + sum(at,(1-intap(at))*Fat(at)) + LDM
+ LDAL - Ul - Vent - El =e= 0;

*** Energy balances ***

energy_balance_HP.. FH*HP('HFH') =e= (sum (ht,(1-inthp(ht))*Fht(ht)) + Uh + LDH_K3471 +
LDH_G3171 + LDH_K301 + LDHM + Eh )*H('h');

energy_balance_AP.. ((1-inthp('K301'))*Fht('K301')*Hout_K301 + LDH_K301*H('h') + w_K301*hw)
+ ((1-inthp('G3171'))*Fht('G3171')*Hout_G3171 + LDH_G3171*H('h') + w_G3171*hw)+ ((1-
inthp('K3471'))*Fht('K3471')*Hout_K3471 + LDH_K3471*H('h') + w_K3471*hw) =e= (sum(at,(1-
intap(at))*Fat(at)) + LDA + LDAL + Ua + Ea) * H('a');

energy_balance_MP.. LDA*H('a') + LDHM*H('h') + w_LDHM*hw + MP('FM')*MP('HFM') + (1-
inthp('K6101'))*Fht('K6101')*HP('y')*Hout_K6101 =e= (LDM + sum(mt,(1-intmp(mt))*Fmt(mt)) +
Um + Em )*H('m');

energy_balance_LP.. LP('FL')*LP('HFL') + LDM*H('m') + LDAL*H('a') + sum(at,(1-
intap(at))*Fat(at)*Hout_ap(at)) + sum(mt,(1-intmp(mt))*Fmt(mt)*Hout_mp(mt)) =e= (Vent + Ul +
El)*H('l');

*** outlet enthalpies calculation ***

HP_outlet_enthalpy_K3471.. Hout_K3471 =e= H('h') - deltaH_K3471 ;
HP_outlet_enthalpy_G3171.. Hout_G3171 =e= H('h') - deltaH_G3171 ;
HP_outlet_enthalpy_K6101.. Hout_K6101 =e= H('h') - deltaH_K6101 ;
HP_outlet_enthalpy_K6102.. Hout_K6102 =e= H('h') - deltaH_K6102 ;
HP_outlet_enthalpy_K301.. Hout_K301 =e= H('h') - deltaH_K301 ;
AP_outlet_enthalpy(at).. Hout_ap(at) =e= H('a') - deltaH_ap ;
MP_outlet_enthalpy(mt).. Hout_mp(mt) =e= H('m') - deltaH_mp ;

*** water to desuperheaters ***

desuper_K3471.. w_K3471 =e= Xwater_K3471*((1-inthp('K3471'))*Fht('K3471')+LDH_K3471);
desuper_G3171.. w_G3171 =e= Xwater_G3171*((1-inthp('G3171'))*Fht('G3171')+LDH_G3171);
desuper_K301.. w_K301 =e= Xwater_K301*((1-inthp('K301'))*Fht('K301')+LDH_K301);
desuper_LDHM.. w_LDHM =e= Xwater_LDHM*LDHM;

**************************** Constraints ******************************************

*** Intermediate continuous variable ***

interhp(ht).. inthp(ht) =e= epshp(ht);
interap(at).. intap(at) =e= epsap(at);
intermp(mt).. intmp(mt) =e= epsmp(mt);

*** Wusers required ***

constraint_users_hp.. HP('usehp') =e= Uh*(H('h')-h0);
constraint_users_ap.. AP('useap') =e= Ua*(H('a')-h0);
constraint_users_mp.. MP('usemp') =e= Um*(H('m')-h0);
constraint_users_lp.. LP('uselp') =e= Ul*(H('l')-h0);

*** Werrors required ***

constraint_error_hp.. HP('errorhp') =e= Eh*(H('h')-h0);
constraint_error_ap.. AP('errorap') =e= Ea*(H('a')-h0);
```

```
constraint_error_mp..   MP('errormp') =e=  Em*(H('m')-h0);
constraint_error_lp..   LP('errorlp') =e=  El*(H('l')-h0);


*** Wturbines required ***

constraint_hp_K3471..  HP('K3471')*(1-epshp('K3471'))=e= Fht('K3471')*deltaH_K3471;
constraint_hp_G3171..  HP('G3171')*(1-epshp('G3171'))=e= Fht('G3171')*deltaH_G3171;
constraint_hp_K6101..  HP('K6101')*(1-epshp('K6101'))=e= Fht('K6101')*deltaH_K6101;
constraint_hp_K6102..  HP('K6102')*(1-epshp('K6102'))=e= Fht('K6102')*deltaH_K6102;
constraint_hp_K301..  HP('K301')*(1-epshp('K301'))=e= Fht('K301')*deltaH_K301;
constraint_ap(at)..  AP(at)*(1-epsap(at))=e= Fat(at)*(H('a')-Hout_ap(at));
constraint_mp(mt)..  MP(mt)*(1-epsmp(mt))=e= Fmt(mt)*(H('m')-Hout_mp(mt));

*** specific constraint for G3171 ***

generator_G3171..  epshp('G3171')=e= 0;

***************************** Initial point *******************************************

parameter F0ht(ht) intial flows for hp turbines
$libinclude xlimport F0ht HP_mass_balance_march.xls m59:q60
Fht.l(ht) = F0ht(ht) ;

parameter F0at(at) intial flows for ap turbines
$libinclude xlimport F0at AssMP_mass_balance_march.xls j64:n65
Fat.l(at) = F0at(at) ;

parameter F0mt(mt) intial flows for mp turbines
$libinclude xlimport F0mt MP_mass_balance_march.xls x56:am57
Fmt.l(mt) = F0mt(mt) ;

parameter Init(n0) initial values for variables from the plant
$libinclude xlimport Init Initial_values_march.xls b3:v4

FH.l = Init('FH0');
Uh.l = Init('Uh0');
Eh.l = Init('Eh0');
LDH_K3471.l = Init('LDH0_K3471');
LDH_G3171.l = Init('LDH0_G3171');
LDH_K301.l = Init('LDH0_K301');
LDHM.l = Init('LDHM0');
w_K3471.l = Xwater_K3471*(Fht.l('K3471')+LDH_K3471.l);
w_G3171.l = Xwater_G3171*(Fht.l('G3171')+LDH_G3171.l);
w_K301.l = Xwater_K301*(Fht.l('K301')+LDH_K301.l);
w_LDHM.l = Xwater_LDHM*LDHM.l;
Ua.l = Init('Ua0');
Ea.l = Init('Ea0');
LDA.l = Init('LDA0');
LDAL.l = Init('LDAL0');
Um.l = Init('Um0');
Em.l = Init('Em0');
LDM.l = Init('LDM0');
Ul.l = Init('Ul0');
El.l = Init('El0');
vent.l = Init('vent0');
H.l('h') = Init('Hh0');
H.l('a') = Init('Ha0');
H.l('m') = Init('Hm0');
H.l('l') = Init('Hl0');
```

```
Hout_K3471.l = H.l('h') - deltaH_K3471 ;
Hout_G3171.l = H.l('h') - deltaH_G3171 ;
Hout_K6101.l = H.l('h') - deltaH_K6101 ;
Hout_K6102.l = H.l('h') - deltaH_K6102 ;
Hout_K301.l = H.l('h') - deltaH_K301 ;
Hout_ap.l(at) = H.l('a') - deltaH_ap ;
Hout_mp.l(mt) = H.l('m') - deltaH_mp ;
epshp.l(ht) = 1;
epsap.l(at) = 0;
epsmp.l(mt) = 0;

zsteam.l = 1e5;
zelec.l = 1e7;
zrepl.l = 1e6;
salv.l = 1e5;
z.l = 365*FH.l*Csteam + zsteam.l + zelec.l + zrepl.l - salv.l ;

*** calculation of Z0 the previous cost with no change at all ***

parameter z0 previous cost of the old distribution;
z0 =  365*FH.l*Csteam + sum(ht, Mst_hp(ht)) + sum(at, Mst_ap(at)) + sum(mt, Mst_mp(mt));


***************************** Solve statment *****************************************

Model distribution /all/;
distribution.scaleopt = 1;

************* Variables scaling **************
z.scale = 1e8;
zsteam.scale = 1e5;
zelec.scale = 1e7;
zrepl.scale = 1e6;
salv.scale = 1e5;
FH.scale = 1000;
Fht.scale(ht) = 100;
Fat.scale(at) = 100;
Fmt.scale(mt) = 100;
LDH_K3471.scale = 100;
LDH_G3171.scale = 100;
LDH_K301.scale = 100;
LDHM.scale = 100;
w_K3471.scale = 10;
w_G3171.scale = 10;
w_K301.scale = 10;
w_LDHM.scale = 10;
LDA.scale = 100;
LDAL.scale = 100;
LDM.scale = 100;
Vent.scale = 100;
Uh.scale = 100;
Ua.scale = 100;
Um.scale = 100;
Ul.scale = 100;
Eh.scale = 100;
Ea.scale = 1000;
Em.scale = 1000;
El.scale = 100;
H.scale(le) = 1000;
Hout_K3471.scale = 1000;
```

```
Hout_G3171.scale = 1000;
Hout_K6101.scale = 1000;
Hout_K6102.scale = 1000;
Hout_K301.scale = 1000;
Hout_ap.scale(at) = 1000;
Hout_mp.scale(mt) = 1000;

************** Equations and constraints scaling **************
cost.scale = 1e7;
cost_steam.scale = 1e6;
cost_elec.scale = 1e6;
replacement.scale = 1e6;
salvage.scale = 1e5;
mass_balance_HP.scale = 100;
mass_balance_AP.scale = 100;
mass_balance_MP.scale = 100;
mass_balance_LP.scale = 100;
energy_balance_HP.scale = 1e6;
energy_balance_AP.scale = 1e6;
energy_balance_MP.scale = 1e6;
energy_balance_LP.scale = 1e6;
HP_outlet_enthalpy_K3471.scale = 1000;
HP_outlet_enthalpy_G3171.scale = 1000;
HP_outlet_enthalpy_K6101.scale = 1000;
HP_outlet_enthalpy_K6102.scale = 1000;
HP_outlet_enthalpy_K301.scale = 1000;
AP_outlet_enthalpy.scale(at) = 1000;
MP_outlet_enthalpy.scale(mt) = 1000;
desuper_K3471.scale = 10;
desuper_G3171.scale = 10;
desuper_K301.scale = 10;
desuper_LDHM.scale = 10;
constraint_users_hp.scale = 1e6;
constraint_users_ap.scale = 1e6;
constraint_users_mp.scale = 1e6;
constraint_users_lp.scale = 1e6;
constraint_error_hp.scale = 1e6;
constraint_error_ap.scale = 1e7;
constraint_error_mp.scale = 1e7;
constraint_error_lp.scale = 1e6;
constraint_hp_K3471.scale= 1e6;
constraint_hp_G3171.scale= 1e6;
constraint_hp_K6101.scale= 1e6;
constraint_hp_K6102.scale= 1e6;
constraint_hp_K301.scale= 1e6;
constraint_ap.scale(at) = 1e6;
constraint_mp.scale(mt) = 1e6;

******************************************************************
parameter rep (*,*);
option solprint=off, iterlim=10000;
distribution.optfile = 1;

Solve distribution using minlp minimizing z ; ***  only in single_solve.gms ***

loop (t,
      HP(dhp)=THP(t,dhp);
      AP(dap)=TAP(t,dap);
      MP(dmp)=TMP(t,dmp);
      LP(dlp)=TLP(t,dlp);
```

```
    Solve distribution using minlp minimizing z ;
    rep(t,'z') = z.l;
    rep(t,ht) = epshp.l(ht);
    rep(t,at) = epsap.l(at);
    rep(t,mt) = epsmp.l(mt);
    rep(t,'FH') = FH.l;
    rep(t,'LDH_K3471') = LDH_K3471.l;
    rep(t,'LDH_G3171') = LDH_G3171.l;
    rep(t,'LDH_K301') = LDH_K301.l;
    rep(t,'LDHM') = LDHM.l;
    rep(t,'LDA') = LDA.l;
    rep(t,'LDAL') = LDAL.l;
    rep(t,'LDM') = LDM.l;
    rep(t,'Vent') = Vent.l;
   );

loop(t,
    if (rep(t,'FH')=0, rep(t,'FH') = eps);
    if (rep(t,'LDH_K3471')=0, rep(t,'LDH_K3471') = eps);
    if (rep(t,'LDH_G3171')=0, rep(t,'LDH_G3171')= eps);
    if (rep(t,'LDH_K301')=0, rep(t,'LDH_K301')= eps);
    if (rep(t,'LDHM')=0, rep(t,'LDHM')= eps);
    if (rep(t,'LDA')=0, rep(t,'LDA')= eps);
    if (rep(t,'LDAL')=0, rep(t,'LDAL')= eps);
    if (rep(t,'LDM')=0, rep(t,'LDM')= eps);
    if (rep(t,'Vent')=0, rep(t,'Vent')= eps);

    loop(ht,
         if (rep(t,ht)=0,
            rep(t,ht)=eps;
           );
      );
    loop(at,
         if (rep(t,at)=0,
            rep(t,at)=eps;
           );
      );
    loop(mt,
         if (rep(t,mt)=0,
            rep(t,mt)=eps;
           );
      );
  );

$libinclude xldump rep solution_march.xls

Display epshp.l, epsap.l, epsmp.l, FH.l, LDH_K3471.l, LDH_G3171.l, LDH_K301.l, LDHM.l, LDA.l,
LDAL.l, LDM.l, Vent.l,
Uh.l, Ua.l, Um.l, Ul.l, Eh.l, Ea.l, Em.l, El.l, Fht.l, Fat.l, Fmt.l, H.l, Hout_K3471.l,
Hout_G3171.l, Hout_K6101.l, Hout_K6102.l, Hout_K301.l, Hout_ap.l,
Hout_mp.l, zsteam.l, zelec.l, zrepl.l, salv.l, z0, rep;
```