

Multi-objective optimisation using the cross-entropy method in CO gas management at a South African ilmenite smelter

Johan Georg Stadler

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in the Faculty of Engineering at Stellenbosch University



Supervisor: Dr James Bekker

December 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Abstract

In a minerals processing environment, stable production processes, cost minimisation and energy efficiency are key to operational excellence, safety and profitability. At an ilmenite smelter, typically found in the heavy minerals industry, it is no different. Management of an ilmenite smelting process is a complex, multi-variable challenge with high costs and safety risks at stake. A by-product of ilmenite smelting is superheated carbon monoxide (CO) gas, or furnace off-gas. This gas is inflammable and extremely poisonous to humans. At the same time the gas is a potential energy source for various on-site heating applications. Re-using furnace off-gas can increase the energy efficiency of the energy intensive smelting process and can save on the cost of procuring other gas for heating purposes.

In this research project, the management of CO gas from the *Tronox KZN Sands* ilmenite smelter in South Africa was studied with the aim of optimising the current utilisation of the gas. In the absence of any buffer capacity in the form of a pressure vessel, the stability of the available CO gas is directly dependent on the stability of the furnaces. The CO gas has been identified as a partial replacement for methane gas which is currently purchased for drying and heating of feed material and pre-heating of certain smelter equipment. With no buffer capacity between the furnaces and the gas consuming plants, a dynamic prioritisation approach had to be found if the CO was to replace the methane. The dynamics of this supply-demand problem, which has been termed the “CO gas problem”, needed to be studied.

A discrete-event simulation model was developed to match the variable supply of CO gas to the variable demand for gas over time – the demand being a function of the availability of the plants requesting the gas, and the feed rates and types of feed material processed at those plants. The problem was formulated as a multi-objective optimisation problem with the two main, conflicting objectives, identified as: 1) the average production time lost per plant per day due to CO-methane switchovers; and 2) the average monthly saving on methane gas costs due to lower consumption thereof. A metaheuristic, namely *multi-objective optimisation using the cross-entropy method*, or MOO CEM, was applied as optimisation algorithm to solve the CO gas problem. The performance of the MOO CEM algorithm was compared with that of a recognised benchmark algorithm for multi-objective optimisation, the NSGA II, when both were applied to the CO gas problem.

The background of multi-objective optimisation, metaheuristics and the usage of furnace off-gas, particularly CO gas, were investigated in the literature review. The simulation model was then developed and the optimisation algorithm applied.

The research aimed to comment on the merit of the MOO CEM algorithm for solving the *dynamic, stochastic* CO gas problem and on the algorithm’s performance compared to the benchmark algorithm. The results served as a basis for recommendations to *Tronox KZN Sands* in order to implement a project to optimise usage and management of the CO gas.

Opsomming

In mineraalprosessering is stabiele produksieprosesse, kostebeperking en energie-effektiwiteit sleuteldrywers tot bedryfsprestasie, veiligheid en wins. 'n Ilmenietsmelter, tipies aangetref in swaarmineraleprosessering, is geen uitsondering nie. Die bestuur van 'n ilmenietsmelter is 'n komplekse, multi-doelwit uitdaging waar hoë kostes en veiligheidsrisiko's ter sprake is. 'n Neweproduk van die ilmenietsmeltproses is superverhitte koolstofmonoksiedgas (CO gas). Hierdie gas is ontvlambaar en uiters giftig vir die mens. Terselfdertyd kan hierdie gas benut word as energiebron vir allerlei verhittingstoepassings. Die herbenutting van CO gas vanaf die smelter kan die energie-effektiwiteit van die energie-intensiewe smeltproses verhoog en kan verder kostes bespaar op die aankoop van 'n ander gas vir verhittingsdoeleindes.

In hierdie navorsingsprojek is die bestuur van die CO gasstroom wat deur die ilmenietsmelter van *Tronox KZN Sands* in Suid-Afrika geproduseer word, ondersoek met die doel om die huidige benuttingsvlak daarvan te verbeter. Weens die afwesigheid van enige bufferkapasiteit in die vorm van 'n drukbestande tenk, is die stabiliteit van CO gas beskikbaar vir hergebruik direk afhanklik van die stabiliteit van die twee hooggoonde wat die gas produseer. Die CO gas kan gedeeltelik metaangas, wat tans aangekoop word vir die droog en verhitting van voermateriaal en vir die voorverhitting van sekere smeltoerusting, vervang. Met geen bufferkapasiteit tussen die hooggoonde en die aanlegte waar die gas verbruik word nie, was die ondersoek van 'n dinamiese prioriteringsbenadering nodig om te kon vasstel of die CO die metaangas kon vervang. Die dinamika van hierdie vraag-aanbod probleem, getiteld die "CO gasprobleem", moes bestudeer word.

'n Diskrete-element simulasiemodel is ontwikkel as probleemoplossingshulpmiddel om die vraag-aanbodproses te modelleer en die prioriteringsbenadering te ondersoek. Die doel van die model was om oor tyd die veranderlike hoeveelhede van geproduseerde CO teenoor die veranderlike gasaanvraag te vergelyk. Die vlak van gasaanvraag is afhanklik van die beskikbaarheidsvlak van die aanlegte waar die gas verbruik word, sowel as die voertempo's en tipes voermateriaal in laasgenoemde aanlegte. Die probleem is geformuleer as 'n multi-doelwit optimeringsprobleem met twee hoof, teenstrydige doelwitte: 1) die gemiddelde verlies aan produksietyd per aanleg per dag weens oorgeskakelings tussen CO en metaangas; 2) die gemiddelde maandelikse besparing op metaangaskoste weens laer verbruik van dié gas. 'n Metaheuristiek, genaamd MOO CEM (*multi-objective optimisation using the cross-entropy method*), is ingespan as optimeringsalgoritme om die CO gasprobleem op te los. Die prestasie van die MOO CEM algoritme is vergelyk met dié van 'n algemeen aanvaarde riglynalgoritme, die NSGA II, met beide toepas op die CO gasprobleem.

The agtergrond van multi-doelwit optimering, metaheuristieke en die benutting van hooggoond af-gas, spesifiek CO gas, is ondersoek in die literatuurstudie. Die simulasiemodel is daarna ontwikkel en die optimeringsalgoritme is toegepas.

Die navorsing het dit ten doel gehad om kommentaar te lewer op die toepaslikheid van die MOO CEM algoritme vir die oplos van die *dinamiese, stogastiese* CO gasprobleem, sowel as op die algoritme se prestasie-vergelyking met die NSGA II algoritme. Die resultate het as 'n grondslag gedien vir aanbevelings aan *Tronox KZN Sands* aangaande die implementering van 'n projek om die benutting en bestuur van die CO gas te optimeer.

Acknowledgements

I wish to extend my gratitude to the following people who supported me toward this thesis:

- My supervisor, James Bekker, for his excellent guidance, support and humour at the right times, especially since I was working on the thesis as a part time student, 2 000 km from campus.
- My former employer, *Exxaro KZN Sands* (January 2007 to July 2012), and current employer, *Tronox KZN Sands* (July 2012 to date), for affording me the valuable resources of financial support and study leave.
- My wife, for her constant encouragement, belief in me and patience with me to complete the thesis in our first year of marriage.
- My manager, Eben Scholtz, for his mentorship, supportive attitude and practical assistance towards my studies, and for allowing me at critical times to give my attention to the research in order to get it completed.
- My colleagues in the Business Improvement section, for absorbing some of my work at times and for supporting me in many other ways.
- My colleagues with whom I am working on the CO gas project at *Tronox KZN Sands*, especially Michael Smith, Maintenance Superintendent at the Mineral Separation Plant, for getting the *Howden* Plant operational and implementing the project at the Reductant Plant; Ashley Windwaai, Maintenance Superintendent at the Smelter, for prioritising the CO gas project amongst a list of many others; and Paul Diedericks, Project Manager at the Smelter and Slag Plant.
- My parents, for encouraging me to remain committed to complete my post graduate studies.
- Anita van der Spuy, for proofreading the document and making valuable suggestions.

My last and special word of gratitude is extended to my Lord and Saviour, Jesus Christ, for the blessings of work and study. I honour You, Lord, as Creator of all and I thank You for enabling us to discover Your genius, amongst others, through scientific research.

Table of Contents

Declaration	i
Abstract	ii
Opsomming	iii
Acknowledgements	v
Table of Contents	vi
List of Figures	x
List of Tables	xii
Glossary	xiii
1. Introduction	1
1.1 Background to the research project	1
1.2 Approach to the experimental part of the research	3
1.3 The research hypothesis	4
1.4 Research aim, specific benefits and objectives	4
1.5 Chapter overview	6
2. Multi-Objective Optimisation and Metaheuristics	8
2.1 Introduction to Multi-Objective Optimisation	8
2.2 Historical Overview of Multi-Objective Optimisation	10
2.2.1 First notion of MOO and early applications	10
2.2.2 Mathematical foundations	11
2.3 Defining optimality in a Multi-Objective Optimisation Problem	12
2.4 Terminology and definitions used in Multi-Objective Optimisation	14
2.4.1 Classification techniques	14
2.4.2 Single-Objective Optimisation	15
2.4.3 The Multi-Objective Optimisation Problem	16
2.5 Metaheuristics in Multi-Objective Optimisation	25
2.5.1 Heuristics	25
2.5.2 Metaheuristics	26
2.6 A General Overview of Optimisation Algorithms	29

2.6.1	Enumerative schemes	31
2.6.2	Deterministic algorithms	31
2.6.3	Stochastic search and optimisation	33
2.7	Summary: Chapter 2	37
3.	The Cross-Entropy Method	39
3.1	Introduction to the CEM	39
3.2	Mathematical foundations of the CEM	40
3.3	Continuous and Discrete Optimisation with the CEM	42
3.3.1	The CEM for continuous optimisation	42
3.3.2	The CEM for discrete optimisation	43
3.4	Steps in the CEM procedure in layman's terms	46
3.4.1	Steps in the CEM procedure in the continuous case	46
3.4.2	Steps in the CEM procedure in the discrete case	48
3.5	Applications of the CEM for Single-Objective Optimisation	50
3.6	CEM for Multi-Objective Optimisation	52
3.7	Summary: Chapter 3	58
4.	Modelling and Simulation	60
4.1	Modelling	60
4.2	Simulation defined	62
4.3	Simulation in perspective	63
4.4	Components of a simulation model	64
4.4.1	Inputs, outputs and states	64
4.4.2	Entities and attributes	65
4.4.3	Activities and events	65
4.4.4	Resources	66
4.4.5	Statistical collectors	66
4.5	Steps in a simulation study	66
4.6	Some advantages, drawbacks and limitations of simulation	67
4.7	Simulation and metaheuristics	69
4.8	Summary: Chapter 4	72
5.	Problem context: Ilmenite smelting and furnace-offgas	73
5.1	Heavy minerals industry	73
5.2	Tronox KZN Sands	74
5.3	Company history and operational background	78
5.3.1	Process description	79

<i>Table of Contents</i>		viii
5.3.2	Mining of heavy minerals	79
5.3.3	Downstream beneficiation at <i>Tronox KZN Sands</i>	81
5.3.4	The furnace off-gas system	85
5.4	Short overview of two common industrial gases	86
5.4.1	Carbon dioxide	86
5.4.2	Carbon monoxide	88
5.5	Examples of the usage of CO gas as a furnace off-gas in industry	89
5.5.1	Pre-heating of scrap in steel making	90
5.5.2	Model Predictive Control	90
5.5.3	Co-generation	91
5.5.4	Risks and challenges	92
5.6	Summary: Chapter 5	92
6.	Problem formulation	94
6.1	Conceptualising the research problem	94
6.1.1	Supply of CO gas	95
6.1.2	Demand for gas	95
6.1.3	Matching the supply of CO gas with the demand for gas	98
6.2	Scope clarification	103
6.3	Solving the problem in the medium term versus the longer term	104
6.4	Summary: Chapter 6	105
7.	Defining the experiment	107
7.1	Simulation applied to the CO gas problem	107
7.2	Preliminary experiment	109
7.3	Selecting the parameters to vary and observe	114
7.4	Collecting and statistical analysis of input data	115
7.5	Translation of the model into a computer simulation language	115
7.6	Summary: Chapter 7	117
8.	Production runs and applying MOO CEM	119
8.1	Initial run to determine preliminary statistics and the number of replications required	119
8.2	Execution of production runs	121
8.2.1	Initial scenarios	121
8.2.2	Extending the number of scenarios	124
8.2.3	Archiving of candidates and solution sets	128
8.3	Model verification and validation	128
8.4	Optimisation with the simulation model	130
8.4.1	Calculating a “decision factor” for MOO	132
8.4.2	Applying a metaheuristic for MOO	134

8.5	Programming the MOO CEM algorithm and finding the Pareto front	136
8.6	Results achieved with the MOO CEM algorithm	137
8.7	Summary: Chapter 8	141
9.	A comparison of MOO CEM with the NSGA II	144
9.1	Performance indicators when assessing MOO algorithm performance	144
9.2	MOO CEM performance when applied to the CO gas problem	146
9.3	Comparative assessment of MOO CEM with a benchmark MOO algorithm	147
9.3.1	Background to algorithm assessment	147
9.3.2	Quality indicators for algorithm assessment	148
9.3.3	Performing the assessment	150
9.3.4	Results of the algorithm assessment	152
9.4	Summary: Chapter 9	154
10.	Research summary, conclusions and recommendations	156
10.1	Summary and conclusions	156
10.2	Recommendations	160
11.	References	162
Appendices		I
Appendix A	Input data analysis	II
Appendix B	Microsoft® Visio model of the CO gas problem	VI
Appendix C	Arena® model of the CO gas problem	XI
Appendix D	Howden plant distribution priorities	XVII
Appendix E	Project Implementation Plan	XIX

List of Figures

Figure 1: Thesis outline	7
Figure 2: Tjalling C. Koopmans.....	10
Figure 3: John von Neumann.....	10
Figure 4: Harold W. Kuhn	12
Figure 5: Albert W. Tucker.....	12
Figure 6: Leonid Hurwicz.....	12
Figure 7: Depiction of a Pareto front for two objectives, f_1 and f_2	14
Figure 8: Two examples of convex sets	21
Figure 9: Two examples of concave sets.....	21
Figure 10: Example of a Pareto front of a bi-objective minimisation problem.....	24
Figure 11: MOP evaluation mapping	25
Figure 12: Global search and optimisation techniques	30
Figure 13: Classification of metaheuristics	31
Figure 14: Hill-climbing algorithm stuck at a local optimum.	32
Figure 15: Functions that present difficulties for hill-climbing algorithms.	32
Figure 16: A unimodal function	32
Figure 17: Reuven Rubinstein	39
Figure 18: Stochastic mechanism for creating sample solutions in the continuous case of the CEM	47
Figure 19: Variance minimisation as applied by the CEM.....	48
Figure 20: Stochastic mechanism for creating sample solutions in the discrete case of the CEM, for DV_1 with $m_1 = 40$	49
Figure 21: Truncated normal distribution on $-1 \leq x \leq 2$, $\mu = 1$, $\sigma = 1$	53
Figure 22: Example of a histogram for decision variable x_i and $r = 5$. Source: (Bekker & Aldrich, 2011)	55
Figure 23: The effect of adjusting histogram frequencies for $DV x_i$. Source: (Bekker & Aldrich, 2011).....	57
Figure 24: Models are abstractions of reality	61
Figure 25: Simulation in perspective	64
Figure 26: A model where a known analytical objective function exists.....	70
Figure 27: A model where no known analytical objective function exists.....	70
Figure 28: Rutile crystals.....	74
Figure 29: Zircon grains.....	74
Figure 30: Heavy minerals operations in South Africa.....	75
Figure 31: Location of the three <i>Tronox KZN Sands</i> sites on the South African east coast.....	76
Figure 32: Central Processing Complex outside Empangeni (Northerly aerial view).....	76
Figure 33: The ilmenite smelting process at <i>Tronox KZN Sands</i>	77
Figure 34: Flaring of CO gas at the furnace emission stacks.....	77
Figure 35: Monitor mining of a heavy mineral deposit	79
Figure 36: Mechanical mining of a heavy mineral deposit	80
Figure 37: Dredge mining of a heavy mineral deposit	80
Figure 38: High level overview of the processes at <i>Tronox KZN Sands</i>	82
Figure 39: PWP at <i>Tronox KZN Sands</i>	83
Figure 40: HMC truck being loaded at the PWP	83
Figure 41: Northerly view the CPC outside Empangeni, South Africa.	83
Figure 42: <i>Tronox KZN Sands</i> MSP at the CPC	84
Figure 43: Tapping operations of titania slag (left) and a metal tap on the right.....	84
Figure 44: Secondary cooling of titania slag "buttons"	84
Figure 45: A disintegrator scrubber	85
Figure 46: Basic concept model of the CO gas problem.....	94
Figure 47: Monthly gas energy required per plant.....	96
Figure 48: CO and methane gas burner at the URIC	97
Figure 49: Increase in methane gas unit cost since 2006.....	97
Figure 50: Actual methane gas cost January to June in 2009, 2010 and 2011	98
Figure 51: CO gas poses serious occupational health and safety risks.....	99
Figure 52: <i>Howden</i> gas compression and distribution plant.....	100
Figure 53: Extended basic concept model.....	101
Figure 54: Detailed concept model of the CO gas problem at <i>Tronox KZN Sands</i>	102
Figure 55: CO gas supply vs. demand	108
Figure 56: CO gas production showing warm-up periods	113
Figure 57: Screen print of overview of submodels in <i>Arena</i> ®.....	116

Figure 58: Screen print of the animation view in <i>Arena</i> ®.....	117
Figure 59: User input form for scenario selection.....	123
Figure 60: Selected scenario saved to <i>Microsoft</i> ® <i>Excel</i>	123
Figure 61: Estimated <i>Howden</i> plant maintenance cost.....	126
Figure 62: Model energy consumption vs. actual energy consumption in <i>Arena</i> ® animation view.....	130
Figure 63: Ranking of the seven initial scenarios using the decision factor value	133
Figure 64: Comparison of the true Pareto front with the MOO CEM-estimated Pareto front.....	138
Figure 65: Exhaustive enumeration of CO gas problem with MOO CEM-estimated Pareto front.....	138
Figure 66: Example of a simple hyperarea and reference point	150
Figure 67: Box-whisker plot of CO gas problem for hyperarea (HV) and epsilon indicator (eps).....	153
Figure 68: Example of input data analysis – FC1 CO gas production	III
Figure 69: Example of input data analysis – FC2 CO gas production	IV
Figure 70: Example of input data analysis – URIC MTTF and MTTR.....	V
Figure 71: <i>Microsoft</i> ® <i>Visio</i> depiction of CO gas supply.....	VII
Figure 72: <i>Microsoft</i> ® <i>Visio</i> depiction of establishing the demand for gas per minute.....	VIII
Figure 73: <i>Microsoft</i> ® <i>Visio</i> depiction of dynamic prioritisation process	IX
Figure 74: <i>Microsoft</i> ® <i>Visio</i> depiction of dynamic prioritisation process (continued from previous page)	X
Figure 75: <i>Arena</i> ® model. Overview of the submodels	XII
Figure 76: <i>Arena</i> ® model. Submodel 1 – Creating the demand for gas	XIII
Figure 77: <i>Arena</i> ® model. Submodel 2 – Creating the supply of gas by the two furnaces	XIV
Figure 78: <i>Arena</i> ® model. Submodels 3 and 4 combined – Prioritisation of the CO gas.....	XV
Figure 79: <i>Arena</i> ® model. Submodels 5 and 6 combined – Registration of gas usage at the users	XVI
Figure 80: <i>Howden</i> plant prioritisation logic.....	XVIII
Figure 81: <i>Microsoft</i> ® <i>Project</i> CO Gas Project implementation plan	XX

List of Tables

Table 1: Working matrix with population of DVs generated	47
Table 2: Structure of the working matrix W	52
Table 3: Plants where CO or methane gas is consumed at the CPC	95
Table 4: List of gas consumers included in the project in order of priority.....	96
Table 5: Scope clarification table	104
Table 6: Input variable information	110
Table 7: Constant values used in the model	110
Table 8: Output variable information	111
Table 9: Depiction of entities in <i>Arena</i> ®.....	113
Table 10: Decision variables identified	114
Table 11: Objectives of the CO gas problem identified.....	115
Table 12: Submodels of the <i>Arena</i> ® model.....	116
Table 13 a-c: Calculating the number of replications required	121
Table 14: The seven combinations of CO gas users analysed initially	122
Table 15: Default values of input variables selected to include in full	125
Table 16: Comparison of actual monthly energy used vs. model energy used	129
Table 17: Average energy usage per plant in the model	130
Table 18: Decision factor values for the seven scenarios initially investigated.....	132
Table 19: Top three initial scenarios ranked on the decision factor value.....	133
Table 20: Values of the true Pareto front and associated Pareto optimal set.....	140
Table 21: MOO CEM-estimated Pareto front values and associated Pareto optimal set	140
Table 22: Quality indicators calculated for the MOO CEM algorithm in the CO gas problem	147
Table 23: Pareto non-compliant unary quality indicator values	152
Table 24: Mean values and 95% confidence interval half-widths for the hyperarea and epsilon indicator..	153
Table 25: Outcome of the hypothesis test for the hyperarea indicator.....	153

Glossary

AC	Alternating current
ASP	Associated Stochastic Problem
BI	Business Improvement
CEM	Cross-Entropy Method
CI	Continuous Improvement
CO	Carbon monoxide
CO ₂	Carbon dioxide
CPC	Central Processing Complex
DC	Direct current
DV	Decision variable
EA	Evolutionary Algorithm
EAF	Electric arc furnace
EC	Evolutionary Computation
EMOO	Evolutionary multi-objective optimisation
EP	Evolutionary Programming
Eps	Epsilon Indicator
FC1	Furnace 1
FC2	Furnace 2
GA	Genetic Algorithm
HA	Hyperarea Indicator
HMC	Heavy Minerals Concentrate
IE	Industrial Engineering
LIMS	Low Intensity Magnetic Separators
LL	Lower Limit
MOEA	Multi-objective optimisation using Evolutionary Algorithms
MOO	Multi-objective optimisation
MOO CEM	Multi-objective optimisation using the Cross-Entropy Method
MOO GA	Multi-objective optimisation Genetic Algorithm
MOP	Multi-objective optimisation problem
MPC	Model Predictive Control
MSP	Mineral Separation Plant
MTP	Metal Treatment Plant
MTTF	Mean-time-to-failure
MTTR	Mean-time-to-repair

NSGA II	2 nd Generation Non-dominated Sorting Genetic Algorithm
OMS	Operations Management System
OR	Operations Research
PWP	Primary Wet Plant
ROM	Run-of-mine
RPP	Reductant Processing Plant
SA	Simulated Annealing
SHEQ	Safety, Health Environment and Quality
SPP	Slag Processing Plant
TS	Tabu Search
UL	Upper Limit
URIC	Unroasted Ilmenite Circuit



1. Introduction

1.1 Background to the research project

The South African mining industry has long been an industry in which industrial engineers have been at work to add value for companies. Compared to other economic sectors, the mining industry is known for the large scale of its operations, its capital intensive nature and robust engineering environment. At the same time the mining industry is synonymous with advanced technology and scientific research, multifaceted logistics and cutting edge business. All these aspects form part of the essence of the mining industry and make this industry a world filled with optimisation opportunities for the ardent industrial engineer. Arguably, it is also for this reason that many mining companies incorporate Business Improvement (BI) or Continuous Improvement (CI) departments into their organisational structures. While completing this thesis, the author attended the *Business Improvement in Mining Africa 2012* conference in Johannesburg, South Africa, (<http://www.businessimprovementinmining.com/Event.aspx?id=658656>) where BI practitioners from most of the large mining groups in South Africa convened. Some of the mining companies represented at the conference included *Anglo American, Anglo Platinum, Kumba Iron Ore, Lonmin, De Beers Marine, Sasol Mining, Goldfields, BHP Billiton, Rio Tinto*, as well as *Exxaro Resources*, where the author was an employee at the time.

The problem that forms the essence of the research covered in this thesis was identified by the author in 2007 while working as an industrial engineer-in-training at a business unit of South African top forty mining company, *Exxaro Resources*. More specifically, the problem was identified at the mining group's heavy minerals mine and smelter, *Exxaro KZN Sands*¹, on the east coast of South Africa. Currently, the author is working at the same business unit as BI Manager. The author's role in that position is to lead a small team of industrial engineers to effect improvement of the business over a broad spectrum of activities, ranging from production improvement and cost saving projects, to initiatives to make the workplace safer and efforts aimed at improving business procedures or service from internal service departments.

The operations of *Tronox KZN Sands* will be described in a later chapter where the research problem will be placed in context of the business' operations. In short, the problem can be described as originating from the smelter plant on the Central Processing Complex (CPC) of *Tronox KZN Sands*, situated outside the town of Empangeni in South Africa. The business mines heavy minerals, alternatively referred to as mineral sands, predominantly containing the minerals

¹ *Exxaro Resources* sold its heavy minerals business, including *Exxaro KZN Sands*, to US based chemicals firm, *Tronox*. The transaction resulted in a name change from *Exxaro KZN Sands* to *Tronox KZN Sands* in July 2012. Hereafter the name *Tronox KZN Sands* will be used in the thesis.



ilmenite, rutile and zircon. The ilmenite is smelted in two 36 MW direct current (DC) arc furnaces to produce titania slag and pig iron. Although “slag” is often a waste product of smelting reactions, in ilmenite smelting, the “slag” is in fact a layer of highly valuable molten titanium dioxide which becomes an important final product. Titania slag from *Tronox KZN Sands* is exported as a feedstock to the *Tronox* pigment plants in the Netherlands and United States as well as to other international pigment producers, whilst the pig iron is exported into international foundry markets.

A third product of the smelting reaction is carbon monoxide (CO) gas, which is also referred to as “furnace off-gas”, or simply, “off-gas”. Other industries in South Africa that produce carbon monoxide gas on a similar scale are steel producers and coke ovens. At the time when the project was identified, *Tronox KZN Sands* was flaring all its produced CO gas into the atmosphere at the top of the two emission stacks of the smelter building. It was always clear that the CO gas had potential to be utilised for economic as well as environmental benefits, instead of being flared into the atmosphere, but this has to date never been achieved sustainably at the business. The main uses of CO gas from large scale industrial operations are heating or drying of feed material, heating of equipment and electricity generation.

With the different possibilities for CO gas usage clear, the need arose at the business to understand how the available volume and energy content of the CO gas would compare to the requirement for CO gas associated with specific uses.

For commercial reasons *Tronox KZN Sands* decided to pursue the option of using the CO gas for heating and drying feed material into its plants and the pre-heating of ladles before iron tapping. Different studies were undertaken to compare the supply and the demand for CO gas. Various models were developed by engineers on site to do this comparison. Although the indications from all these models were that the average supply of CO gas exceeds the peak demand for gas, the project to start using the gas was never successfully launched. The main reason for the latter was the inconsistency of the furnaces in the absence of a pressure vessel to build up a buffer capacity of CO gas. *It was realised by the author that a more dynamic model that could handle stochastic, time-dependent changes of the CO gas supply versus demand had to be developed in order to compare the supply-demand balance over shorter periods. It would not be enough to know that on average, say, over a month, there is sufficient CO gas to meet the average monthly gas levels required. A model had to be developed to indicate on a near instantaneous basis, for example a minute-by-minute basis, how the gas supply compared with the requirement for gas. This problem forms the core of the research project and will be explained in detail later. It will be referred to as the “CO gas problem” or “research problem”.*

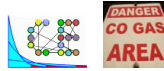


1.2 Approach to the experimental part of the research

The research project originated right at this point when the author, as one of the few industrial engineers on site, was tasked with developing a discrete-event simulation model of the CO gas supply-demand process. Discrete-event simulation was found an adequate approach to this problem, due to the stochastic nature of most of the variables of the problem. As soon as the model for the current process was validated and verified, production runs could be performed. Thereafter, a number of *hand-picked scenarios* could be tested to search for the optimum scenario, that at which operational stability and economic value of the project would be optimised. In the search for the optimum scenario the author was guided by his supervisor to apply an *optimisation algorithm* in conjunction with the simulation model to evaluate whether a *Pareto optimum set* of solutions could be found by applying an optimisation algorithm. In other words, it would be evaluated by means of an *iterative, automated process* whether a *Pareto front* existed in the objective space when a *much wider selection of scenarios* from the decision space was tested on the simulation model. From the myriad optimisation techniques available, a metaheuristic, the cross-entropy method (CEM), more specifically the multi-objective version of the latter metaheuristic, namely the MOO CEM algorithm, was selected. The CEM is a relatively recently developed and computationally powerful metaheuristic which will be discussed in depth in Chapter 3, including the reasons for its selection. The performance of the MOO CEM, when applied to the specific multi-objective problem (MOP), could be evaluated – firstly by determining the degree of Pareto dominance of the approximated Pareto front, and, secondly, by comparing the MOO CEM algorithm's performance to that of a benchmark MOO algorithm, the NSGA II, when the latter is applied to the same MOP.

The problem posed some interesting challenges in terms of developing a simulation model that represented the reality at the smelter satisfactorily, as well as in “linking” the simulation model and the metaheuristic from a programming perspective. Careful thought had to go into the *level of abstraction* at which the problem was to be modelled, seeing that an ilmenite smelter production process can easily become a complicated modelling exercise. The simulation model was developed in *Arena*® software and the MOO CEM algorithm was programmed in *Matlab*®. The two software packages had to be interlinked in order to perform the optimisation.

Later in the thesis, further background to the research problem and the formulation thereof will be described in considerably more detail. In the next two sections, the research hypothesis, and the aims and objectives of the research are stated. In the last section of this chapter a chapter overview of the thesis is provided.



1.3 The research hypothesis

The research hypothesis investigated in this study was stated as:

Applying the MOO CEM algorithm to the dynamic, stochastic CO gas problem reduces the computational burden of finding an approximated Pareto front that compares satisfactorily with the true Pareto front.

A positive outcome to the research hypothesis will achieve the following contribution to the body of knowledge:

1. It will provide a case study in the successful application of the MOO CEM algorithm to a specific multi-objective dynamic, stochastic problem. The focus is on the terms *dynamic*, which refers to a process that evolves over time and *stochastic*, which refers to the statistical variation in such a process. If the CEM converges fast in the multi-objective case of the CO gas problem, then similar optimisation problems can be studied using this case study as a reference.
2. It will provide a case study for the *South African industrial engineering community* of a successful application of this metaheuristic to a real-world problem from the *South African heavy minerals industry*. This will add to the limited number of case studies available within the South African industrial engineering community on the application of the CEM, as well as to the limited number of case studies illustrating an intersection between industrial engineering and heavy minerals mining and processing.

The *aim* of the research and the *objectives* serve to support the hypothesis are discussed next.

1.4 Research aim, specific benefits and objectives

The *research aim*, defined in (Muller & Lategan, 2008) as the macro purpose of the study, is to demonstrate that the CEM can be used effectively to optimise a specific dynamic, stochastic MOP from the South African heavy minerals industry. In other words, the aim is to demonstrate the CEM's ability to find a Pareto front displaying good proximity and diversity whilst reducing the computational effort of solving the selected dynamic, stochastic MOP. The aim is extended to attempt to prove the MOO CEM algorithm's superiority to another optimisation technique, generally accepted in Operations Research (OR) circles as a benchmark for multi-objective optimisation (MOO), the second generation non-dominated sorting genetic algorithm (NSGA II). It was not the aim of the research to make a general statement regarding the CEM's ability to solve



dynamic, stochastic MOPs, yet positive outcome of this research hypothesis would add to the case studies found in the literature supporting the CEM's ability to solve various hard optimisation problems, and would be one of the first case studies supporting the MOO CEM algorithm's success with dynamic, stochastic MOPs.

As alluded to in the sections above, the expected *value from*, or *contribution by*, the research was twofold: The outcome of the research would contribute to the academic body of knowledge, would the outcome of the research hypothesis be positive, and, successful results from the research would allow the author to make recommendations to *Tronox KZN Sands* regarding the implementation of the project. The latter could have significant economic and other value for the company. These contributions can be broken down into the following *specific benefits*:

1. If the CEM can be applied successfully to find the Pareto optimum set of solutions to the CO gas problem within acceptable computational time, this would serve as a supporting case study to the supervisor's doctoral research. The latter investigates the suitability and efficacy of the CEM for MOO of dynamic, stochastic processes.
2. Depending on the findings in 1) a comparison of performance between MOO CEM and the NSGA II would position the MOO CEM algorithm in terms of its performance ability within the fields of metaheuristics and MOO.
3. If a suitable optimum scenario could be found and implemented, it could result in a cost saving for *Tronox KZN Sands* potentially exceeding R 1 million per month².

The research objectives are the specific research tasks which needed to be done, according to [Muller & Lategan \(2008\)](#), and these are:

1. Review of the literature and positioning of this research within the body of existing literature.
2. Determining whether the Pareto front for the dynamic, stochastic MOP (the CO gas problem) can be created economically, in terms of computational time, by means of MOO CEM.
3. Determining whether the approximated Pareto front is effective and efficient by comparing it to the true Pareto front and applying appropriate quality indicators.
4. Comparing the performance of the MOO CEM algorithm relative to the NSGA II, using appropriate Pareto compliant indicators.
5. Providing recommendations to *Tronox KZN Sands* based on the outcome of the research, in order to implement the project.

² Based on 2012 methane prices



1.5 Chapter overview

The thesis is divided into three main themes. In Chapters 2 to 4 a picture is painted of the academic fields with which this research intersect, thereby introducing the first theme of the thesis: the theoretical background. This is done by providing a scholarly overview of the development and some of the key aspects of those fields.

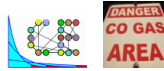
In Chapter 2 a literature review provides an overview, a brief history and foundational definitions of MOO. Chapter 2 continues the literature study by discussing metaheuristics as a popular approach towards MOO, followed by a discussion on the broader field of global search and optimisation wherein metaheuristics is just a subfield. Hereby metaheuristics is placed in context of other search and optimisation techniques and a number of the most popular metaheuristics are discussed. The latter discussion includes references to different evolutionary as well as non-evolutionary algorithms.

Chapter 3 focuses on the selected metaheuristic applied in the solution development phase of this research project, namely the CEM. The CEM, in turn, forms the foundation of the MOO CEM. Its origin is discussed, mathematical formulation is presented and the main algorithm is presented and explained. Chapter 3 also discusses its application to continuous and discrete optimisation, as well as its application to single and multi-objective optimisation, including references of some real-world applications.

Since discrete-event simulation was used extensively in the solution development phase of the research, its basic principles, the components of a simulation model, the steps in a simulation study, the main benefits and drawbacks of simulation and the interplay between simulation and metaheuristics are discussed in Chapter 4. Chapter 4 concludes the first theme of the thesis.

Chapter 5 covers the second thesis theme which entails discussing the operational context of the research problem. In section 5.1 the heavy minerals industry in South Africa is briefly discussed. Arguably, heavy minerals mining and beneficiation processes are relatively complex compared to those of other sectors of the mining industry, such as coal or iron ore. A proper understanding of the heavy minerals industry and the basic processes involved will aid in the reader's understanding of the research problem and the recommendations made based on the research findings. In section 5.2 the reader is introduced to *Tronox KZN Sands* where the research problem was identified, while a short history and a basic process overview of the company are discussed in section 5.3. Sections 5.4 and 5.5 provide a brief look at the usage of industrial gases, with a focus on furnace off-gas management, as this was the specific subfield of mining and minerals processing where the research problem originated.

The third and final theme covers the experimental work and solution development over Chapters 6 to 10. The simulation and optimisation methodology that were followed and results



achieved are discussed. As part of this theme, based on the optimisation results and results of the comparative exercise of the MOO CEM algorithm and the NSGA II, the suitability of MOO CEM for solving the CO gas problem is discussed. Chapter 10 concludes the thesis with a research summary, by discussing the degree to which the research aims and objectives were achieved, and by providing a list of recommendations to *Tronox KZN Sands*. Figure 1 provides a graphical depiction of the outline of the thesis.

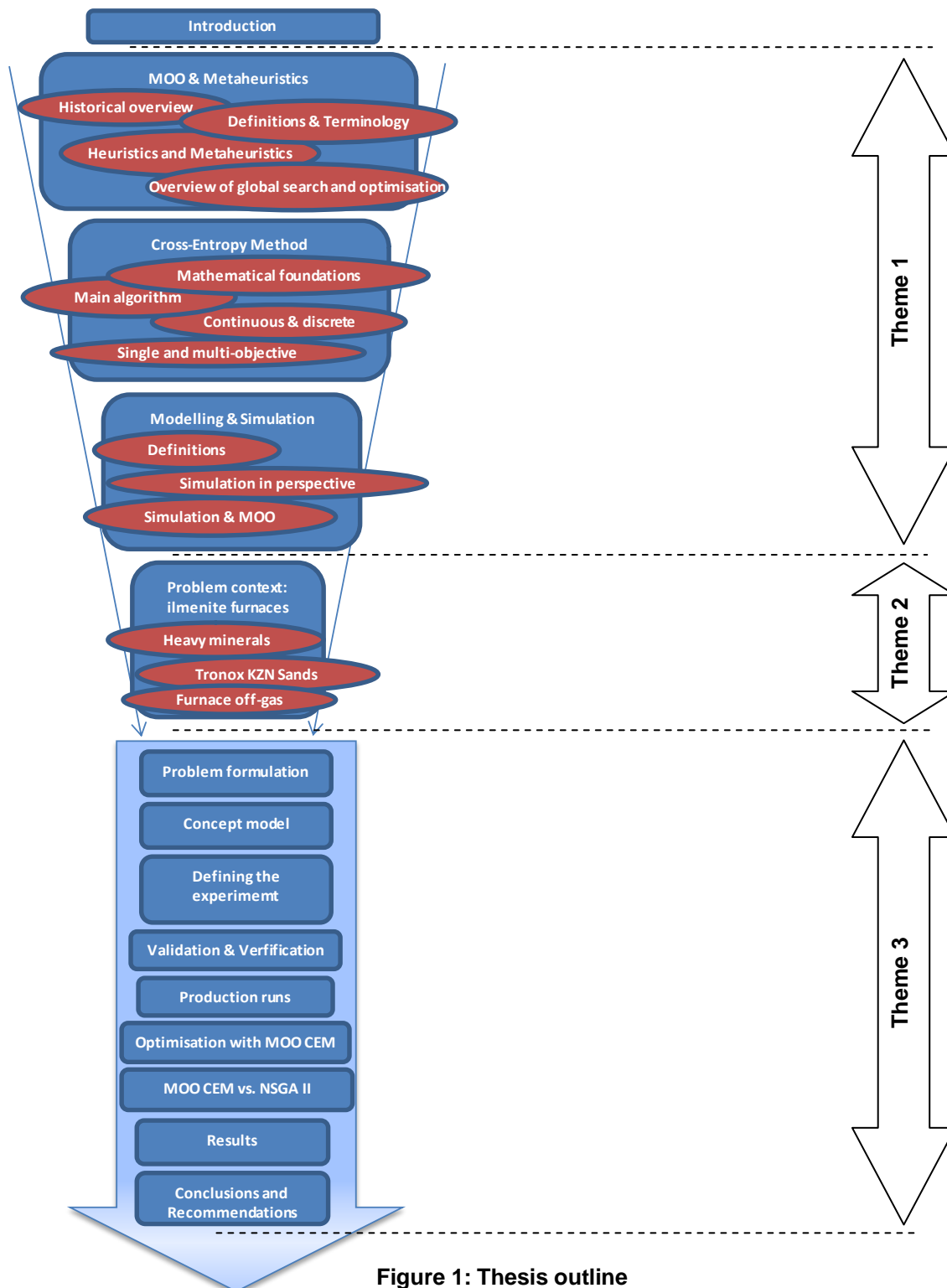
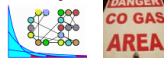


Figure 1: Thesis outline



2. Multi-Objective Optimisation and Metaheuristics

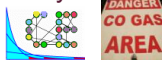
In the previous chapter the research project was introduced. The chapter started with the background to the research problem, explained the approach to the experimental part of the research, stated the research hypothesis, aim and objectives, and provided a chapter overview of the thesis.

In this chapter the field of multi-objective optimisation (MOO) will be discussed, with a further focus on metaheuristics applied in MOO. These fields form the bulk of the theoretical background to the thesis. This chapter will provide an introduction and historical overview of MOO followed by a discussion of its mathematical foundations. Next in this chapter the concept of “optimality” in a multi-objective context is defined and some important terminology and definitions related to MOO are discussed. Subsequently, the concepts of heuristics and metaheuristics are discussed, while the relationship between MOO and metaheuristics is explored. The chapter closes with a general overview of global search and optimisation approaches, in order to better contextualise metaheuristics as one approach in that universe. In this last section of this chapter some of the most popular metaheuristics are also discussed in more detail.

2.1 Introduction to Multi-Objective Optimisation

Humans are daily faced with making decisions. These decisions range from as simple as what cereal to eat for breakfast or where to go for the long weekend, to more complex such as which car to buy, long term investment decisions or which person to marry. Most everyday decisions require that a number of factors need to be considered simultaneously. Effectively, humans are faced with multi-objective decision making on a regular basis. Arguably, humans naturally have a fairly well developed capacity to make such decisions, many even in an instant. This capacity can also be enlarged and refined with knowledge and experience, which might develop into wisdom over time. The acclaimed author, journalist, cultural commentator and intellectual adventurer, Malcolm Gladwell, in his book *Blink, The Power of Thinking without Thinking* (Gladwell, 2005), calls this capacity of humans “snap judgement”.

In many cases, especially in a context where a more scientific or reasoned approach is required, for example in business, economics or engineering, these decisions need to be taken as objectively as possible, to be well-justified and the process of reaching a decision is often recorded. Often these decisions are indeed too complex to be made instantly. In those cases computers are usually relied on to aid in the decision making. It should naturally be inferred that when making a multi-objective decision, the decision maker also aims to optimise the result,

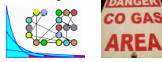


effect or implication of the decision, hence it is not only multi-objective *decision making* that is being dealt with, but also multi-objective *optimisation*.

MOO has probably been practised since the first human was around, but the concept has, especially over the last 50 years, developed into a multi-faceted research field. According to [Coello Coello et al. \(2007\)](#) the term came into use formally only in the 1960s, although research in the application of MOO in fields like economics and sociology dates back much earlier. A brief historical overview of the field will be presented later in this chapter.

Multi-objective optimisation also goes by the names “multi-criteria optimisation”, “multi-performance optimisation”, “combinatorial optimisation” and “vector optimisation” in the literature. The term multi-objective optimisation, with its acronym, MOO, will be used in this thesis.

Optimisation refers to finding the best possible solution to a problem given a set of limitations or constraints. When dealing with a single objective to be optimised (e.g., the cost of a design), the designer aims to find the best possible solution available or at least a good approximation of it. This is called a “global optimum”. However, when devising optimisation models for many real-world problems, it is frequently the case that there is not one, but several objectives that require optimisation. Think of deciding on a new job: some considerations would be salary, work hours, location, associated quality of life, growth potential, intellectual stimulation and availability of work opportunities for a spouse nearby. These considerations are often conflicting, since one can in general not find a high paying job and live at the beach and be close to one’s family and find the perfect job for one’s spouse nearby. An example from an engineering context would be designing an automotive part. The designer would like to minimise its cost and weight, while maximising its performance and strength. In fact, problems with multiple objectives arise in a natural fashion in most disciplines and the solutions of some have challenged researchers for a long time. Consider here the Buffer Allocation Problem, the Travelling Salesman Problem and the Knapsack 0/1 problem. MOPs require different mathematical and algorithmic tools than those adopted to solve single-objective optimisation problems. [Coello Coello et al. \(2007\)](#) state that, despite the considerable variety of techniques developed in Operations Research (OR) and other disciplines to tackle MOPs, the complexities of their solution have called for researchers to find alternative approaches. Considering the increase in the number of publications on applications of MOO in various fields, it is clear that MOO has become a very important research topic for practitioners from different fields ([Corne et al., 1999](#)). It is especially scientists and engineers who find MOO a very powerful tool with which to solve complex problems. ([Coello Coello et al., 2007](#)) state that research in the field has grown considerably over the last 15 to 20 years and mention that there are still many unanswered questions in this field. For example, [Coello Coello \(1999; 2006\)](#) states that “there is not even a universally accepted definition of ‘optimum’ as in single-objective optimisation, which makes it difficult to even compare results of one method to the other”.



Normally the decision about what the “best” answer is remains the prerogative of the so called (human) decision maker.

This thesis will take a look at selected approaches developed to solve MOPs. The thesis will focus on a specific MOP from engineering practice in the South African heavy minerals industry and take the reader through the process followed towards solving this MOP. However, before that is done, a historical overview of MOO is presented, followed by discussion on the mathematical foundations of MOO.

2.2 Historical Overview of Multi-Objective Optimisation

2.2.1 First notion of MOO and early applications

MOO theory is not as recent as is often thought. In fact, some authors, for example [Stadler \(1979\)](#), indicate that the notion of MOO is an inherent part of economic equilibrium and, in consequence, can be traced back to Adam Smith’s well known 1776 treatise, *The Wealth of Nations*. [Coello Coello et al. \(2007\)](#) state that the general concept of economic equilibrium is often attributed to Léon Walras. However, William Stanley Jevons, Carl Menger, Francis Ysidro Edgeworth and Vilfredo Pareto also did very important work in this regard in the period between 1874 and 1906. Another field that developed with an inherent link to MOO is the theory of psychological games and the notion of game strategy (based on analysing the psychology of the enemy) which is attributed to Félix Édouard Émile Borel. The latter field, called Game Theory, dates back to the work done by Borel in 1921 which led to the writing of his book *Applications aux Jeux de Hasard (Applications for Gaming)* in 1938. Most historians, however, tend to attribute the origins of Game Theory to a paper from the Hungarian-born American mathematician John von Neumann, which was orally presented in 1926 and published as a seminal paper of his in 1928. [Coello Coello et al. \(2007\)](#) quote John von Neumann and Oskar Morgenstern in their 1944 classic, *Theory of Games and Economic Behaviour*, as saying that an optimisation problem in the context of a social exchange economy was “a peculiar and disconcerting mixture of several conflicting problems” that was “nowhere dealt with in classical mathematics”. Apparently, according to [Coello Coello et al.](#), von Neumann and Morgenstern did not discuss this problem any further



Figure 3: John von Neumann

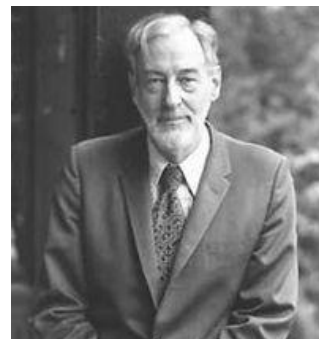
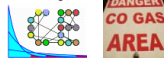


Figure 2: Tjalling C. Koopmans



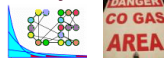
in their book and no real contribution in this regard, i.e. formalising the notion of MOO as a field of study, was made until the 1950s. Only in 1951, the Nobel prize winner in Economic Sciences, Tjalling C. Koopmans, edited a book called *Activity Analysis of Production and Allocation* (Koopmans, 1951), where the concept of “efficient” vector was first used in a significant way. At this stage researchers were on the brink of pinning down the characteristics of MOO as an independent study field.

In the 1960s multi-objective public investment problems became more common and “trade-off” became a favourite term used by managers, planners, and decision makers, write Cohon and Marks in their research on water resources management, Cohon & Marks (1975). In this way MOO arose in a natural fashion in mathematical economics, and increasingly more techniques were developed by systems analysts and decision theorists for the solving of private and public sector problems, by control theorists for engineering (guidance and design) problems, and by water resource economists and systems analysts for water resource planning problems. There was also some renewed interest in Kuhn and Tucker’s vector maximum theory during the early 1960s, as is reflected in papers by Zadeh (1963), Klingler (1964) and Da Cunha & Polak (1967). Application of MOO to fields outside economics began with the work by Koopmans (1951) in production theory and that of Marglin (1967) in water resources planning. Coello Coello *et al.* (2007) state that the first engineering application accounted for in the literature was a paper in the early 1960s on automated control by Lofti Zadeh (Zadeh, 1963). However, the use of MOO did not become general until the 1970s when it was treated in the works of Cohon & Marks (1975) and Cohon (1978). Coello Coello *et al.* provide a comprehensive list, from a wide variety of sources, of good reviews of existing mathematical programming techniques for MOO (Coello Coello *et al.*, 2007, p. 31).

Recently, MOPs and their solutions are widely reported in the literature. Chapter 7 in Coello Coello *et al.* (2007) is devoted to comprehensive discussions of applications in engineering, science, industry and miscellaneous fields like investment portfolio optimisation and stock ranking, while Bekker (2012, pp. 35-40) discusses a list of applications. With specific reference to evolutionary algorithms (EAs) for MOO, or evolutionary multi-objective optimisation (EMOO), a comprehensive list of references is maintained at the EMOO home page (<http://www.lania.mx/~ccoello/>, cited 26 May 2012).

2.2.2 Mathematical foundations

According to Stadler (1979) in his review of the history of “Multicriteria Optimisation”, as he refers to the field of MOO, the origins of the mathematical foundations of MOO can be traced back to the period between 1895 and 1906. During that period, German mathematician, Georg Cantor,



produced two works published in the *Mathematische Annalen*, Cantor (1895; 1897), and another German, Felix Hausdorff, in his publication, Hausdorff (1906), laid the foundations of infinite dimensional ordered spaces. Cantor also introduced equivalence classes and stated the first sufficient conditions for the existence of a utility function, while Hausdorff gave the first example of a complete ordering. However, it was the concept of a “vector maximum problem” introduced by Princeton University Professors of Mathematics, Harold W. Kuhn and Albert W. Tucker, in their publication on non-linear programming, Kuhn & Tucker (1951), which made MOO a mathematical discipline in its own right. The so-called “proper efficiency” in the context of MOO was also formulated in this seminal paper that can be considered as the first serious attempt to derive a theory in this area. This same direction was followed two years later in Arrow *et al.* (1953) where the term “admissible” instead of “efficient” points was used in a publication called *Contributions to the Theory of Games*. However, state Coello Coello *et al.* (2007), MOO theory remained relatively undeveloped during the 1950s, and the subject was sparsely covered, by only a few authors, see for example the works of Koopman (1953; 1956), Hitch (1953), Hoag (1956), Klahr (1958) and Karlin (2003). Arguably the most important research outcome related to MOO towards the late 1950s was goal programming, introduced in Charnes & Cooper (1957) and further developed in Charnes & Cooper (1961). It took until the 1960s for the foundations of MOO to be consolidated and taken seriously by pure mathematicians. This can be dated to Russian-born American mathematician and Nobel Laureate, Leonid Hurwicz, who eventually generalised the results of Kuhn and Tucker to topological vector spaces in his work *Studies in Linear and Nonlinear Programming* in Hurwicz (1964). Thereafter the field of MOO, with its mathematical backbone generally established and accepted, started developing increasingly into an applied scientific field.



Figure 4: Harold W. Kuhn



Figure 5: Albert W. Tucker

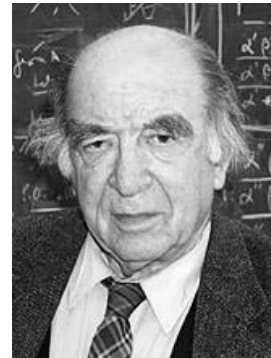
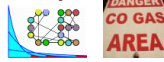


Figure 6: Leonid Hurwicz

2.3 Defining optimality in a Multi-Objective Optimisation Problem

Before going further, a definition of a MOP is required. In a classic definition, celebrated scholars of the field, Oszczka (1985) and Coello Coello (1999), define a MOP as follows:

A problem of finding a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions form a mathematical description of performance



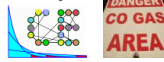
criteria which are usually in conflict with each other. Hence, the term “optimise” means finding such a solution which would give the values of all the objective functions acceptable to the designer.

In simpler terms, a MOP is a problem with *two or more objectives that need to be optimised simultaneously*. Formal mathematical definitions will be presented in section 2.4.

It is important to mention that there might be constraints imposed on the objectives. Further, Coello Coello (2006) states it is important to emphasise that normally the objectives of a MOP are in conflict with each other. If this is not the case, he states, a single solution exists for the MOP. Purshouse & Fleming (2007), however, state that it is useful to recognise that a number of relationships can exist between objectives and that these relationships can vary within the domain of the search. This view of Purshouse and Fleming is substantiated by an earlier work of the same authors on relationships between objectives published in Purshouse & Fleming (2003). The latter authors agree with Coello that objectives may, and most often do, exhibit conflict, in which a solution modification leads to improved performance in one objective and worsened performance in another, but may also exhibit harmony, in which the solution modification improves performance in all objectives. Purshouse and Fleming note that early multi-objective analyses have tended to focus on conflict (understandably so, seeing that this has been demonstrated to represent significant challenges to MOO researchers and practitioners), but write that other relationships “undoubtedly exist in real-world applications and are also worthy of research”.

Most MOPs do not lend themselves to a single solution and have, instead, a set of solutions. Such solutions are really “trade-offs” or good compromises among the objectives. In order to generate these trade-off solutions, an old notion of optimality is normally adopted. This notion of optimality was originally introduced by Irish philosopher and political economist Francis Ysidro Edgeworth in Edgeworth (1881) and later generalised by famous Italian engineer, sociologist and economist Vilfredo Pareto in his accoladed work, *Cours D'Economie Politique*, (Pareto, 1896). As mentioned in the introduction to this chapter, it is important to remember that Coello Coello commented that at present there is still no universally accepted definition of “optimum”.

At the same time, the *Edgeworth-Pareto optimum* or, simply, *Pareto optimum* serves as a very handy and widely accepted concept in the MOO field. A solution to a MOP is defined as *Pareto optimal* if *no other feasible solution exists which would decrease some criterion without causing a simultaneous increase in at least one other criterion* – assuming it refers to a minimisation problem. It should not be difficult to realise that the use of this concept almost always gives, not a single solution, but a set of them, which is called the *Pareto optimal set*. Other terms used in the literature for the Pareto optimal set are *efficient vector*, *non-dominated vectors*



or *non-dominated solutions*. The author will use the term Pareto optimal set. The vectors of the decision variables corresponding to the solutions included in the Pareto optimal set are called *non-dominated*. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the *Pareto front* or *trade-off surface/plain* (Coello Coello, 2006). Figure 7 illustrates the concept of a Pareto front for two objectives. Pareto Optimality will be discussed further and formally defined in section 2.4.3.7, along with other terms and definitions associated with the broader field of MOO.

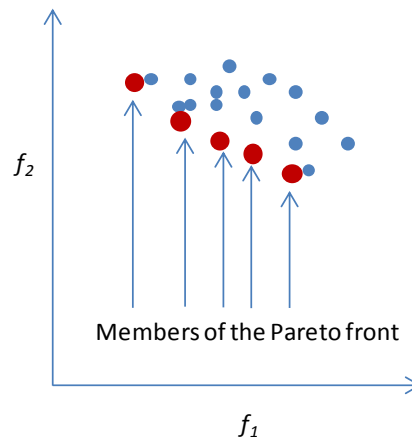


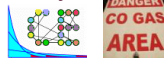
Figure 7: Depiction of a Pareto front for two objectives, f_1 and f_2 .

2.4 Terminology and definitions used in Multi-Objective Optimisation

In order to develop an understanding of MOPs and the ability to design adequate optimisation algorithms to solve them, a series of formal non-ambiguous definitions is required. These definitions provide a precise set of symbols and formal relationships that permit proper analysis of MOPs.

2.4.1 Classification techniques

Several attempts have been made to classify the many MOO techniques currently in use. Section 2.6 will also make such an attempt based on the literature covered by the author, with special recognition given to the work of Coello Coello *et al.* (2007). Yet, first of all, it is quite important to distinguish two stages into which the solution of a MOP can be divided: 1) the optimisation of the several objective functions involved and 2) the process of deciding what kind of “trade-offs” are appropriate from the decision maker’s perspective. The latter is called the “multi-criteria decision making” process. In this section, some techniques available for these two



stages of a MOP are mentioned. For a detailed discussion and analysis of their advantages and disadvantages, refer to [Coello Coello *et al.* \(2007\)](#). [Cohon & Marks \(1975\)](#) proposed one of the most popular classifications of techniques within the OR community. Other classifications are obviously possible (see for example [Duckstein \(1984\)](#)). However, the classification proposed by [Cohon & Marks \(1975\)](#) has been adopted for the purposes of this thesis, *because it focuses the classification on the way in which each technique handles the two problems of searching and making (multi-criterion) decisions* ([Van Veldhuizen & Lamont, 2000](#); [Horn, 1997](#)):

1. **A priori Preference Articulation**: make decisions *before* searching (decide → search).
2. **A posteriori Preference Articulation**: search *before* making decisions (search → decide).
3. **Progressive Preference Articulation**: integrate search and decision making (decide ↔ search).

Examples of *a priori* articulation are the Global Search Method, Goal Programming, the Goal-Attainment Method, the Lexicographic Method, Min-Max Optimisation, Multi-attribute Utility Theory, Surrogate Worth Trade-off, the ELECTRE technique and the PROMETHEE technique.

Examples of *a posteriori* techniques include Linear Combination of Weights and the ε -Constraint Method.

Examples of *progressive* preference articulation include the Probabilistic Trade-off Development method, the STEP method and the Sequential Multi-objective Problem Solving Method.

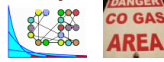
It is beyond the scope of this thesis to discuss these techniques in detail, but the reader is referred to [Coello Coello *et al.* \(2007\)](#) for a rather detailed discussion of each. Yet, they are worth mentioning in this thesis as they allude to different ways in which the decision maker goes through the two stages of “search” and “decide” when solving a MOP.

The two subsections below present generic mathematical and formal definitions pertaining to the field of optimisation, with the emphasis on MOO.

2.4.2 Single-Objective Optimisation

Firstly, it is necessary to define the single objective optimisation problem which continues to be addressed by many search techniques.

Definition 1 (General Single-Objective Optimisation Problem): A general single-objective optimisation problem is defined as minimising (or maximising) $f(\mathbf{x})$ subject to $g_i(\mathbf{x}) \leq 0$,



$i = \{1, \dots, m\}$, and $h_j(\mathbf{x}) = 0$, $j = \{1, \dots, p\}$, $\mathbf{x} \in \Omega$. A solution minimises (or maximises) the scalar $f(\mathbf{x})$ where \mathbf{x} is a n -dimensional decision variable vector $\mathbf{x} = (x_1, \dots, x_n)$ from some universe Ω .

Observe that $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$ represent constraints that must be fulfilled while optimising (minimising or maximising) $f(\mathbf{x})$. The set Ω contains all possible \mathbf{x} 's that can be used to satisfy an evaluation of $f(\mathbf{x})$ and its constraints, while \mathbf{x} can be a vector of continuous or discrete variables, with f also being either continuous or discrete.

The method for finding the global optimum (which may or may not be unique) of any function is referred to as **Global Optimisation**. In general, the global minimum of a single-objective problem is presented in Definition 2 below:

Definition 2 (Single-Objective Global Minimum Optimisation):

Given a function $f: \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $\Omega \neq \emptyset$, for $\mathbf{x} \in \Omega$ the value $f^* \triangleq f(\mathbf{x}^*) > -\infty$ is called a global minimum if and only if the value

$$\forall \mathbf{x} \in \Omega: f(\mathbf{x}^*) \leq f(\mathbf{x}). \quad \dots 1$$

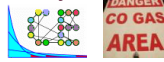
The vector \mathbf{x}^* is by definition the global minimum solution, f is the objective function, and the set Ω is the feasible region of \mathbf{x} . The goal of determining the global minimum solution(s) is called the global optimisation problem for a single-objective problem. While single-objective optimisation problems may have a unique optimal solution, MOPs (as a rule) present a possibly uncountable set of solutions which, when evaluated, produce vectors whose components represent trade-offs in the objective space. As mentioned, this requires a decision maker to choose an acceptable solution (or solutions) by selecting one or more of these vectors.

2.4.3 The Multi-Objective Optimisation Problem

Osyczka's definition of a MOP as worded in his work, *Multi-criteria optimisation for engineering design* (Osyczka, 1985), was shown in section 2.3. While accepting his definition for the purpose of this thesis, the following subsections will discuss formal mathematical definitions for some of the most important aspects of a MOP.

2.4.3.1 Decision variables

The **decision variables** are the numerical quantities for which values are to be chosen in an optimisation problem. These quantities are denoted as \mathbf{x}_j , $j = 1, 2, \dots, n$.



The vector \mathbf{x} of n decision variables is represented by

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad \dots 2$$

This can be written more conveniently as

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T, \quad \dots 3$$

where T indicates the transposition of the column vector to the row vector.

2.4.3.2 Constraints

Optimisation problems normally have restrictions imposed on them by the particular characteristics of the environment or available resources (e.g., physical limitations, time restrictions, etc.). In order to consider a certain solution feasible, these restrictions must be satisfied. These restrictions are generally called **constraints**, and they describe dependences among decision variables and constants (or parameters) involved in the problem. These constraints are expressed in the form of mathematical inequalities

$$g_i(\mathbf{x}) \leq 0, i = 1, \dots, m \quad \dots 4$$

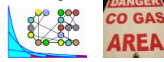
or equalities

$$h_j(\mathbf{x}) = 0, j = 1, \dots, p. \quad \dots 5$$

Note that p , the number of equality constraints, must be less than n , the number of decision variables, because if $p \geq n$ the problem is said to be over-constrained, since there are no degrees of freedom left for optimising (i.e., in other words, there would be more unknowns than equations). The number of degrees of freedom is given by $n - p$. Also, constraints can be explicit (i.e. given in algebraic form) or implicit, in which case the algorithm to compute $g_i(\mathbf{x})$ for any given vector \mathbf{x} must be known.

2.4.3.3 Commensurable vs. Non-Commensurable

In order to know how “good” a certain solution is, it is necessary to have some criteria by which to evaluate it. These criteria are expressed as computable functions of the decision variables, called **objective functions**. As already mentioned, in real-world problems objective functions are often in conflict with others, with some requiring to be minimised while others require to be maximised.



These objective functions may be **commensurable** (measured in the same units) or **non-commensurable** (measured in different units). Coello Coello *et al.* (2007) state that the multiple objectives being optimised almost always conflict, placing a partial, rather than total, ordering on the search space. In fact, finding the global optimum of a general MOP is an NP-Complete problem, states Bäck (1996).

2.4.3.4 Attributes, criteria, objectives, and goals

It is a practice in OR to differentiate among attributes, criteria, objectives and goals (see for example MacCrimmon (1973)). The term **attributes** often refers to differentiating aspects, properties or characteristics of alternatives or consequences. **Criteria** generally refer to evaluative measures, dimensions or scales against which alternatives may be evaluated as to their worth. **Objectives** are sometimes viewed in the same way, but may also denote specific desired levels of attainment or vague ideals. **Goals** usually indicate either of the latter notions, i.e. criteria or objectives. A common distinction in OR is to use the term goal to designate potentially achievable levels, and objective to designate unattainable ideals.

The convention adopted in the authoritative work, Coello Coello *et al.* (2007), is the same as that assumed by several researchers (see for example Horn (1997) and Fishburn (1978)) of using the terms objective, criteria, and attribute interchangeably to represent a MOP's goals or objectives (i.e. distinct mathematical functions) to be achieved. The terms **objective space** or **objective function space** are used to denote the coordinate space within which vectors resulting from evaluating a MOP's solutions are plotted.

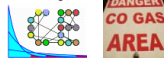
The objective functions are designated: $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$, where k is the number of objective functions in the MOP being solved. Therefore, the objective functions form a vector function $\mathbf{f}(\mathbf{x})$ which is defined by

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}. \quad \dots 6$$

This can be written more conveniently as

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T. \quad \dots 7$$

The set of all n -tuples of real numbers denoted by \mathbb{R}^n is called the **Euclidean n -space**. The following two Euclidean spaces are relevant to MOPs:



- The n -dimensional space of the decision variables in which each coordinate axis corresponds to a component of vector \mathbf{x} .
- The k -dimensional space of the objective functions in which each coordinate axis corresponds to a component vector $\mathbf{f}_k(\mathbf{x})$, as shown for example by f_1 and f_2 in [Figure 7](#).

Every point in the n -dimensional space represents a solution and gives a certain point in the k -dimensional space which determines the *quality* of this solution in terms of the objective function values.

The single-objective formulation is extended to reflect the nature of MOPs where there is not one objective function to optimise, but many. Thus, there is not a unique solution but a set of solutions. This set of solutions is found through the use of Pareto Optimality Theory ([Ehrgott, 2005](#)). Note that MOPs require a decision maker to make a choice of \mathbf{x}_i^* values. This selection is essentially a trade-off of one complete solution \mathbf{x} over another in multi-objective space. More precisely, MOPs are those problems where the goal is to optimise k objective functions simultaneously. This may involve the maximisation of all k functions, the minimisation of all k functions or a combination of maximisation of some and minimisation of some of these k functions. A MOP global minimum (or maximum) problem is formally defined by Definition 3 ([Van Veldhuizen, 1999](#)):

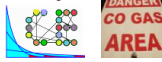
Definition 3 (General MOP): A general MOP is defined as minimising (or maximising) $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$ subject to $g_i(\mathbf{x}) \leq 0$, $i = \{1, \dots, m\}$, and $h_j(\mathbf{x}) = 0$, $j = \{1, \dots, p\}$, $\mathbf{x} \in \Omega$. A MOP solution minimises (or maximises) the components of a vector $\mathbf{F}(\mathbf{x})$ where \mathbf{x} is a n -dimensional decision variable vector $\mathbf{x} = (x_1, \dots, x_n)$ from some universe Ω . It is noted that $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$ represent constraints that must be fulfilled while minimising (or maximising) $\mathbf{F}(\mathbf{x})$ and Ω contains all possible \mathbf{x} that can be used to satisfy an evaluation of $\mathbf{F}(\mathbf{x})$.

Thus, a MOP consists of k objectives reflected in the k objective functions, $m + p$ constraints on the objective functions and n decision variables. The k objective functions may be linear or nonlinear and continuous or discrete in nature. The evaluation function, $F : \Omega \rightarrow A$, is a mapping from the vector of decision variables ($\mathbf{x} = x_1, \dots, x_n$) to output vectors ($\mathbf{y} = a_1, \dots, a_k$) ([Van Veldhuizen, 1999](#)). The vector of decision variables \mathbf{x}_i can also be continuous or discrete.

2.4.3.5 Ideal vector

Definition 4 (Ideal Vector) : Let

$$\mathbf{x}^{0(i)} = [x_1^{0(i)}, x_2^{0(i)}, \dots, x_n^{0(i)}]^T \quad \dots 8$$



be a vector of variables which optimises (either minimises or maximises) the i -th objective function $f_i(\mathbf{x})$. In other words, the vector $\mathbf{x}^{0(i)} \in \Omega$ is such that

$$f_i(\mathbf{x}^{0(i)}) = \underset{\mathbf{x} \in \Omega}{\text{opt}} f_i(\mathbf{x}). \quad \dots 9$$

Then, the vector

$$\mathbf{f}^0 = [f_1^0, f_2^0, \dots, f_k^0]^T, \quad \dots 10$$

(where \mathbf{f}_i^0 denotes the optimum of the i -th function) is ideal for a MOP, and the point in \mathbb{R}^n which determined this vector is the ideal (utopical) solution, and is consequently called the **ideal vector**.

In other words, the ideal vector contains the optimum for each separately considered objective achieved at the same point in \mathbb{R}^n .

2.4.3.6 Convexity and Concavity

Definition 5 (Convexity): A function $\phi(\mathbf{x})$ is called **convex** over the domain of \mathbb{R} if for any two vectors \mathbf{x}_1 and $\mathbf{x}_2 \in \mathbb{R}$,

$$\phi(\theta\mathbf{x}_1 + (1 - \theta)\mathbf{x}_2) \leq \theta\phi(\mathbf{x}_1) + (1 - \theta)\phi(\mathbf{x}_2), \quad \dots 11$$

where θ is a scalar in the range $0 \leq \theta \leq 1$.

A convex function cannot have any value larger than the function values obtained by linear interpolation between $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$.

If the reverse inequality of the (11) holds, the function is **concave**. Thus $\phi(\mathbf{x})$ is **concave** if $-\phi(\mathbf{x})$ is **convex**. Linear functions are convex and concave at the same time.

A set of points (or region) is defined as a convex set in n -dimensional space if, for all pairs of two points \mathbf{x}_1 and \mathbf{x}_2 in the set, the straight-line segment joining them is also entirely in the set. Thus, every point \mathbf{x} , where

$$\mathbf{x} = \theta\mathbf{x}_1 + (1 - \theta)\mathbf{x}_2 \quad 0 \leq \theta \leq 1 \quad \dots 12$$

is also in the set. So, for example, the sets shown in [Figure 8](#) are convex, but the sets shown in [Figure 9](#) are not.

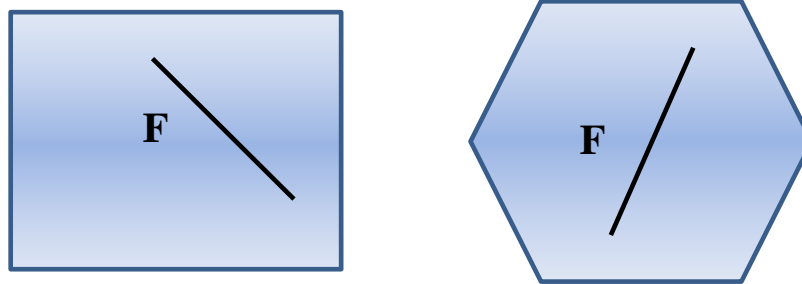
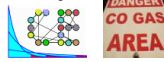


Figure 8: Two examples of convex sets

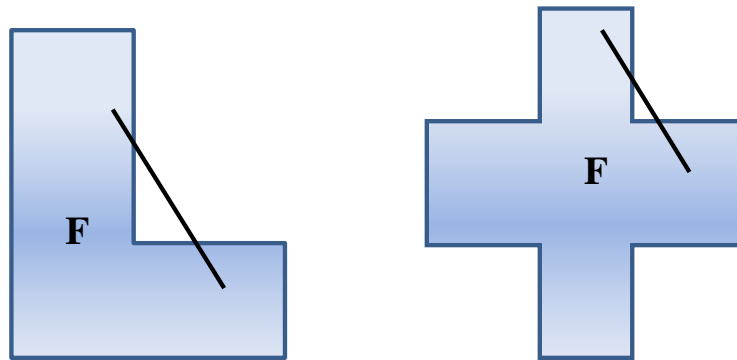


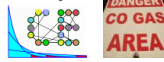
Figure 9: Two examples of concave sets

2.4.3.7 Pareto terminology

The notion of “optimum” changes when there are several objective functions. In MOO, the aim is to find good compromises or “trade-offs” rather than a single solution as in global optimisation. As mentioned in section 2.3, the notion of “optimum” most commonly adopted is that originally proposed in Edgeworth (1881) and later generalised in Pareto (1896). Although some authors such as Stadler (1988) call this notion the *Edgeworth-Pareto optimum*, the most commonly accepted term is *Pareto optimum*. Before the formal definition is given and discussed, the most popular non-Pareto based approaches to optimality should be mentioned.

There are examples in the literature where the optimisation of MOPs was attempted, and in some cases relatively successfully achieved (Coello Coello, 1999), by applying techniques that *do not make use of the concept of the Pareto optimum*. These approaches are listed below. Note that some have already been mentioned in 2.4.1:

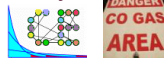
- **Weighted Sum Approach**, where the decision maker transforms the MOP into a scalar optimisation function by assigning different weighting coefficients to each of the objective functions.



- **Goal Programming**, where the decision maker assigns targets or goals that he/she wishes to achieve for each objective function.
- **Goal-Attainment**, where the decision maker assigns a vector of weights relating to the under or over-attainment of the desired goals in addition to the goal vector for the objective functions.
- The **ϵ -Constraint Method** where the most preferred or primary objective function is minimised (assuming a minimisation problem) whilst considering the other objectives as constraints bound by some allowable levels ϵ_i . This way a single objective minimisation is performed on the primary objective function subject to additional constraints on the other objective functions. The levels ϵ_i are then altered to generate the entire Pareto optimum set, although it might not be called that when using this approach.
- The **VEGA**, David Schaffer's extension of the Simple Genetic Algorithm (SGA) where the way in which selection is done is altered ([Schaffer, 1985](#)).
- **Lexicographic ordering**, whereby the objectives are ranked by the decision maker in order of importance. The optimum solution is then obtained by minimising (assuming a minimisation problem) the objective functions, starting with the most important one and proceeding according to the assigned ranked order.
- Using **Game Theory** in a more graphical approach, whereby the Nash equilibrium ([Nash, 1953](#)) is found, assuming a non-cooperative game.
- Three others are the **Weighted Min-max approach**, the **Contact Theorem**, and the use of **Randomly Generated Weights** and **Elitism**. All these different approaches are discussed in more detail in [Coello Coello \(1999\)](#).

The idea of using Pareto-based fitness assignment was first proposed in [Goldberg \(1989\)](#) to solve the problems of Schaffer's approach. He suggested the use of *non-dominated ranking and selection* to move a population toward the Pareto front in a MOP. A significant amount can be said about Pareto optimality ranking or non-dominated ranking, but for now the reader is referred to [Goldberg & Richardson \(1987\)](#), [Goldberg \(1989\)](#) and [Coello Coello \(1999\)](#) for further discussions. At this point it is sufficient that the reader take notice that other approaches besides the Pareto-based ones exist.

A number of researchers, among whom [Hilliard et al. \(1988\)](#), [Liepins et al. \(1990\)](#) and [Ritzel et al. \(1994\)](#), found that the Pareto optimality ranking was superior to non-Pareto based approaches such as VEGA for the problems they tested.



Formal definitions for key concepts in the subfield of Pareto optimal ranking, when solving MOPs, are now presented:

Definition 6 (Pareto Optimality): A solution $\mathbf{x} \in \Omega$ is said to be Pareto Optimal with respect to (w.r.t.) Ω if and only if (iff) there is no $\mathbf{x}' \in \Omega$ for which $\mathbf{v} = \mathbf{F}(\mathbf{x}') = (f_1(\mathbf{x}'), \dots, f_k(\mathbf{x}'))$ dominates $\mathbf{u} = \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$. The phrase **Pareto Optimal** is taken to mean with respect to the entire decision variable space unless otherwise specified.

This definition says that \mathbf{x}^* is Pareto optimal if there exists no feasible vector \mathbf{x} which would decrease some criterion without causing a simultaneous increase in at least one other criterion (assuming minimisation). The concept of Pareto Optimality is integral to the theory and the solving of MOPs. Additionally, there are a few more definitions that are also adopted in MOO:

Definition 7 (Pareto Dominance): A vector $\mathbf{u} = (u_1, \dots, u_k)$ is said to **dominate** another vector $\mathbf{v} = (v_1, \dots, v_k)$ if and only if \mathbf{u} is partially less than \mathbf{v} , i.e.,

$$\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\}: u_i < v_i \quad \dots 13$$

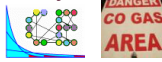
Definition 8 (Pareto Optimal Set): For a given MOP, $\mathbf{F}(\mathbf{x})$, the **Pareto Optimal Set**, \mathcal{P}^* , is defined as:

$$\mathcal{P}^* := \{\mathbf{x} \in \Omega \mid \neg \exists \mathbf{x}' \in \Omega \quad \mathbf{F}(\mathbf{x}') \preceq \mathbf{F}(\mathbf{x})\}. \quad \dots 14$$

Pareto optimal solutions are those solutions within the decision search space whose corresponding objective vector components cannot all be simultaneously improved. These solutions form the Pareto Optimal Set and are also termed *non-inferior*, *admissible*, or *efficient solutions*, with the entire set represented by \mathcal{P}^* . Their corresponding vectors are termed *non-dominated*. Selecting (a) vector(s) from this vector set (the Pareto front set \mathcal{PF}^*) implicitly indicates acceptable Pareto Optimal solutions or decision variables. These solutions may have no apparent relationship besides their membership of the Pareto optimal set. They form the set of all solutions whose associated vectors are non-dominated. Pareto optimal solutions are classified as such based on their evaluated functional values.

Definition 9 (Pareto front): For a given MOP, $\mathbf{F}(\mathbf{x})$, and Pareto Optimal Set \mathcal{P}^* , the **Pareto front** \mathcal{PF}^* is defined as

$$\mathcal{PF}^* := \{\mathbf{u} = \mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^*\}. \quad \dots 15$$



When plotted in objective space, the non-dominated vectors are collectively known as the **Pareto front**. Again, \mathcal{P}^* is a subset of some solution set. Its evaluated objective vectors form \mathcal{PF}^* , of which each is non-dominated with respect to all objective vectors produced by evaluating every possible solution in Ω . In general, it is not easy to find an analytical expression of the line or surface that contains these points and in most cases it turns out to be impossible. The normal procedure to generate the Pareto front is to compute many points in Ω and their corresponding $f(\Omega)$. When there are a sufficient number of these, it is then possible to determine the non-dominated points and to produce the Pareto front.

A sample Pareto front of a bi-objective problem is shown in Figure 10 with the solid curve marking the Pareto front of the feasible area \mathbf{F} . Note that $\mathcal{PF}^{*_{\text{true}}}$ is often used interchangeably with \mathcal{PF}^* in the literature. Although single-objective optimisation problems may have a unique optimal solution, MOPs usually have a possibly uncountable set of solutions on a Pareto front. Each solution associated with a point on the Pareto front is a vector whose components represent trade-offs in the decision space or Pareto solution space.

The MOP's evaluation function, $\mathbf{f} : \Omega \rightarrow \mathcal{A}$, maps decision variables ($\mathbf{x} = x_1, \dots, x_n$) to vectors ($\mathbf{y} = a_1, \dots, a_k$). This situation is represented in Figure 11 for the case $n = 2$, $m = 0$, and $k = 3$. This mapping may or may not be onto some region of objective function space, dependent upon the functions and constraints composing the particular MOP.

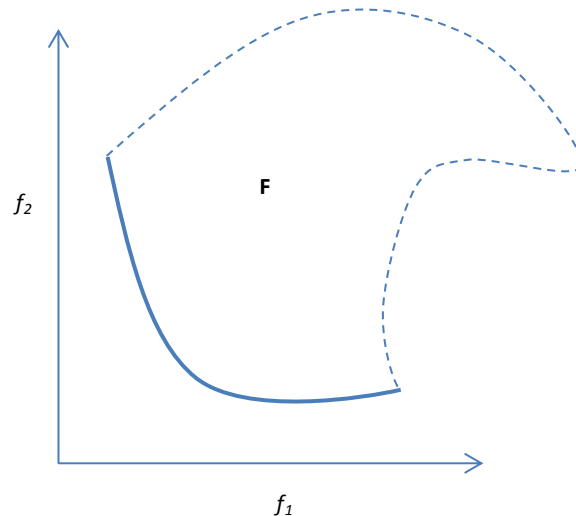


Figure 10: Example of a Pareto front of a bi-objective minimisation problem

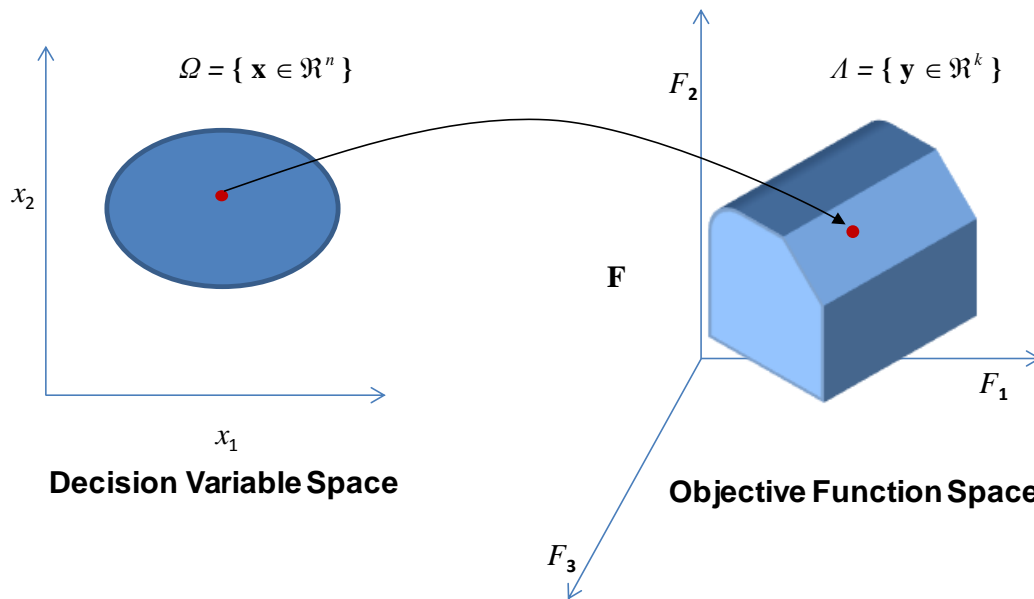
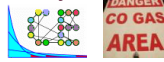


Figure 11: MOP evaluation mapping

2.5 Metaheuristics in Multi-Objective Optimisation

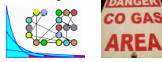
Optimisation using metaheuristics has become a very popular research topic in the last few years, state [Corne et al. \(1999\)](#) in their *New Ideas in Optimisation*. These authors state that this is especially true for problems with two or more objective functions. They also mention, importantly, that when applying metaheuristics to MOPs, different mathematical and algorithmic tools are required than those adopted to solve single-objective optimisation. In fact the reader is reminded that [Coello Coello \(2006\)](#) comments that even the notion of “optimality” changes when dealing with MOPs.

The general nonlinear single-objective optimisation problem is as much an open problem as the general nonlinear MOP. Therefore, the use of metaheuristics is a valid choice which, in fact, has rapidly gained acceptance among researchers from a wide variety of disciplines.

But what are metaheuristics? The next two sections will aim to define first heuristics and then metaheuristics.

2.5.1 Heuristics

The term *heuristic* in Greek means “to find” or “to discover” and refers to experience-based techniques for problem solving, learning, and discovery. The website www.freedictionary.com (cited 26 May 2012) defines the term heuristic in three different contexts:



- Of or relating to a usually speculative formulation serving as a guide in the investigation or solution of a problem. For example: "The historian discovers the past by the judicious use of such a heuristic device as the ideal type" (a quotation by Karl J. Weintraub, German-born American historian).
- Of or constituting an educational method in which learning takes place through discoveries that result from investigations made by the student.
- In Computer Science: Relating to or using a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a program for use in the next step of the program.

The most fundamental heuristic is trial and error. [Wikipedia](#) comments that heuristic methods are used to speed up the process of finding a good enough solution where an exhaustive search is impractical. Examples of this method include using a "rule of thumb", an educated guess, an intuitive judgment, or common sense. More examples of other commonly used heuristics can be found in the book, *How to Solve It* ([Polya, 2008](#)), of which a few are:

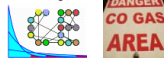
- If you are having difficulty understanding a problem, try drawing a picture.
- If you cannot find a solution, try assuming that you have a solution and seeing what you can derive from that ("working backwards").
- If the problem is abstract, try examining a concrete example.
- Try solving a more general problem first (the "inventor's paradox": the more ambitious plan may have more chance of success).

In more precise terms, Israeli-American computer scientist Judea Pearl defines heuristics as "strategies using readily accessible, though loosely applicable, information to control problem solving in human beings and machines" ([Pearl, 1984](#)).

Regardless of whose definition one prefers, "heuristics" is about seeking the solution to a problem and the processes by which a problem solver learns to solve a problem more efficiently.

2.5.2 Metaheuristics

The term "metaheuristic" is credited to University of Colorado professor of computer science, Fred Glover ([Bekker, 2012](#)). It could be argued that "metaheuristics" is a rather unfortunate term often used to describe a major subfield, indeed the primary subfield, of stochastic optimisation. Stochastic optimisation is the category of algorithms and techniques which employ *some degree of randomness* to find optimal (or as optimal as possible) solutions to difficult problems. The next section will look at global search and optimisation techniques and discuss *stochastic optimisation* as a subfield thereof (see [Figure 12](#)). Metaheuristics are the most general method within this



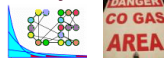
subfield and are applied to a very wide range of problems. Metaheuristics are even sometimes referred to simply as stochastic optimisation, but that is too general a term as it includes important algorithms like the Markov Chain Monte Carlo (MCMC) and Gibbs Sampling which do not belong to the same subfield as metaheuristics (Luke, 2009). Metaheuristics has lately been a popular term of use, “but it is quite a misleading and weird term”, comments Luke (2009), and then he continues, “when I hear ‘metadiscussion’, I think: a discussion about discussions; or ‘metadata’: I think data about data. Likewise when I hear ‘metaheuristic’ I think: a heuristic about (or for) heuristics. But that is not at all what these algorithms are about!” Perhaps the lesser-used term “black box optimisation” would be better, though it too comes with some additional baggage adds the same author. “Weak methods” is also too broad a term: it does not imply stochasticity. Sometimes the term “stochastic search” is used: but often “search” problems are defined as all-or-nothing: either you find the solution or you do not. The goal of metaheuristics is not doing search, it is doing optimisation. So what does the term “metaheuristics” mean? In an attempt to answer that question, a consideration of the type of problems that metaheuristics are applied to is required.

Luke (2009) attempts to clarify the kind of problems suited to optimisation by metaheuristics by using the following curious, but rather effective, analogy. In a famous 1964 United States court case regarding obscenity, popularly referred to as *Jacobellis v. Ohio*, the US Supreme Court Justice Potter Stewart famously wrote (O'Meara & Shaffer, 1964),

“I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description [referring to hard-core pornography]; and perhaps I could never succeed in intelligibly doing so. But I know it when I see it, and the motion picture involved in this case is not that.”

The phrase “I know it when I see it” became a colloquial expression within the United States by which a speaker attempts to categorise an observable fact or event, although the category is subjective or lacks clearly defined parameters.

Luke (2009) suggests that metaheuristics are applied to “I know it when I see it” problems. They are algorithms used to find answers to problems when the analyst has very little to help him or her: Luke puts it this way, “You do not know what the optimal solution looks like, you do not know how to go about finding it in a principled way, you have very little heuristic information to go on and brute-force search is out of the question because the search space is too large. But if you are given a candidate solution to your problem, you can test it and assess how good it is. That is, you know a good solution when you see it.”



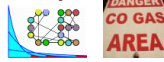
An example used in [Luke \(2009\)](#) is quite illustrative: imagine the objective is trying to find an optimal set of robot behaviours for a soccer goalie robot. You have a simulator for the robot and can test any given robot behaviour set and assign it a quality. You know a good one when you see it. You have come up with a definition for what robot behaviour sets look like in general, but you have no idea what the optimal behaviour set is, nor even how to go about finding it. The simplest thing you could do in this situation is Random Search: just try random behaviour sets as long as you have time, and return the best behaviour set you discovered. (Notice, you have applied a heuristic: trial and error, basically). Before you give up, consider the following alternative, known as Hill-Climbing. Start with a random behaviour set. Then make a small, random modification to it and try the new version. If the new version is better, throw the old one away, else throw the new version away. Now make another small, random modification to your current version (whichever one you didn't throw away). If this newest version is better, throw away your current version; else throw away the newest version. Repeat as long as you can.

Hill-climbing is a simple metaheuristic algorithm. It exploits a heuristic *belief* (i.e. a belief that *through some search mechanism a solution can be found*) about your space of candidate solutions, which is usually true for many problems. This heuristic belief says that similar solutions tend to behave similarly (and tend to have similar quality), so small modifications will generally result in small, well-behaved changes in quality, allowing the analyst to “climb the hill” of quality up to good solutions. *This heuristic belief is one of the central defining features of metaheuristics*: indeed, nearly all metaheuristics are essentially elaborate combinations of hill-climbing and random search ([Luke, 2009](#)). In section 2.6 these two metaheuristics (random search and hill-climbing), among many others, will be discussed in more detail and placed in context of their appropriate subfields.

While keeping Luke's definition and explanation in mind, the author found [Wikipedia's](#) definition of a metaheuristic concise and descriptive:

In computer science, metaheuristic designates a computational method that optimises a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Metaheuristics make few or no assumptions about the problem being optimised and can search very large spaces of candidate solutions. However, metaheuristics do not guarantee an optimal solution is ever found. Many metaheuristics implement some form of stochastic optimisation. Other terms having a similar meaning as metaheuristic, are: derivative-free, direct search, black-box, or indeed just heuristic optimiser.

Now, knowing that heuristics refer to a process of problem solving by searching and learning and that metaheuristics are not heuristics about or for heuristics, but rather a major subfield of



stochastic optimisation applied to “I know it when I see it” problems, the reader should have a clearer understanding of the term metaheuristic. Discussing metaheuristic in the context of the other subfields of stochastic optimisation in the next section should further add to the reader’s understanding of what a metaheuristic is.

2.6 A General Overview of Optimisation Algorithms

As discussed in section 2.2, the OR community has been developing approaches to solve MOPs since the 1950s. Currently, a wide variety of mathematical programming techniques to solve MOPs are available in the specialised literature. See for example [Miettinen \(1999\)](#) and [Ehrgott \(2005\)](#). However, mathematical programming techniques have certain limitations when tackling MOPs.

For example, many of them are *susceptible to the shape of the Pareto front* and may not work when the Pareto front is concave or disconnected. Others *require differentiability* of the objective functions and the constraints. Also, most of them *generate only a single solution from each run*. Thus, several runs (using different starting points) are required in order to generate several elements of the Pareto optimal set ([Miettinen, 1999](#)). In contrast, metaheuristics deal simultaneously with a *set of possible solutions* (the so-called *population*) which allows an analyst to find several members of the Pareto optimal set in a single run of the algorithm. Additionally, most metaheuristics are *less susceptible to the shape or continuity of the Pareto front* (e.g., they can easily deal with discontinuous and concave Pareto fronts).

[Coello Coello \(2006\)](#) points out that from the several optimisation algorithms currently available, Evolutionary Algorithms (which are based on the emulation of the mechanism of natural selection) are among the most popular, especially to solve MOPs, see [Goldberg \(1989\)](#) and [Fogel \(1999\)](#).

Authors such as [Coello Coello et al. \(2007\)](#), [Luke \(2009\)](#), [\(Venter, 2010\)](#) and [Bekker \(2012\)](#) describe a wide variety of different optimisation algorithms or metaheuristics. The aim of this section is to provide the reader with an overview of the field of optimisation algorithms, especially those suitable to solve MOPs. Considering the broad scope of this field, the different optimisation algorithms are not discussed in detail. At this point, though, the reader is reminded again that at the centre of this thesis is the discussion of one specific metaheuristic and to evaluate its ability to solve a MOP from a real-world engineering environment. This metaheuristic will be discussed in detail in Chapter 3. However, at this stage, the broad spectrum of optimisation algorithms will first be discussed. [Coello Coello et al. \(2007\)](#) classify general search and optimisation techniques into three categories:



- **Enumerative,**
- **Deterministic,** and
- **Stochastic.**

Although it must be mentioned that an enumerative search is deterministic, a distinction is made here as it employs no heuristics. Figure 12 shows each category with its subfields as found in the literature and thereby also gives a visual overview of the broad field of global search and optimisation.

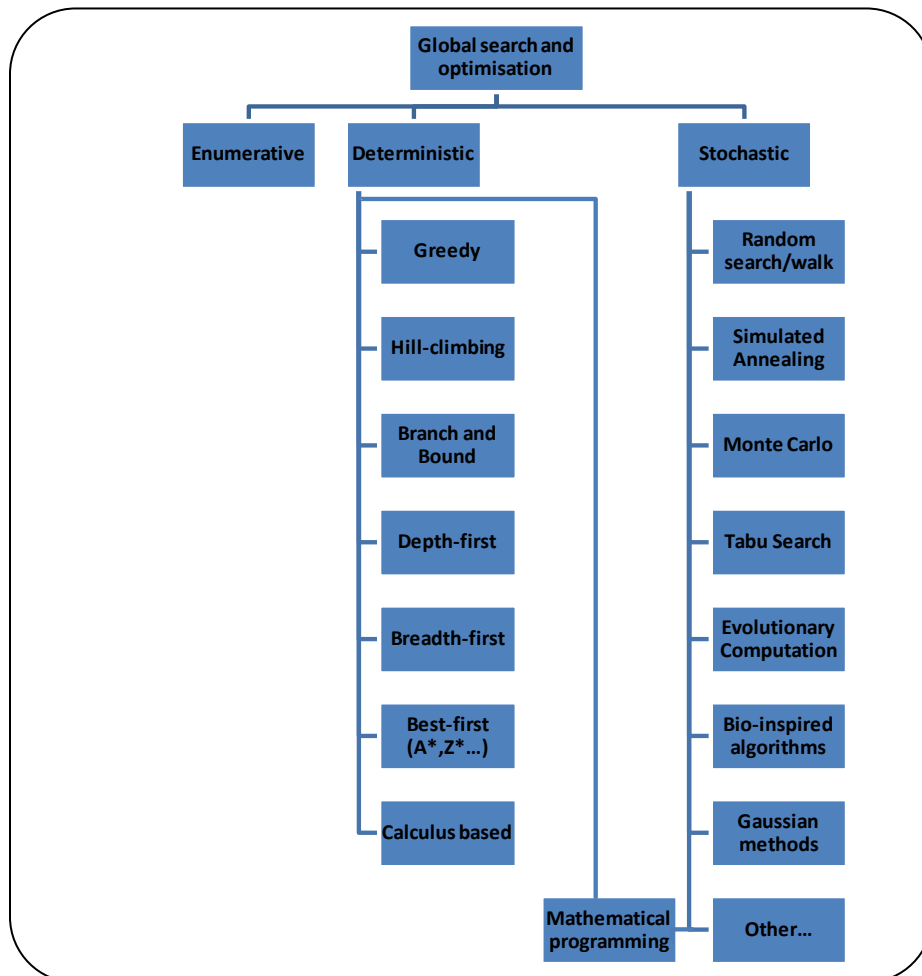
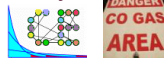


Figure 12: Global search and optimisation techniques

Another possible way of classifying optimisation algorithms is shown in Figure 13. As indicated by these two figures, not all researchers classify the subfields in the same way, especially not those in the stochastic category. Some researchers, like Venter (2010), for example, classify Simulated Annealing and Tabu Search under Evolutionary Computation. The author prefers the grouping shown in Figure 12, which is in line with Coello Coello *et al.* (2007), Luke (2009) and



Bekker (2012), hence the discussion below will pertain to the subdivision of algorithms as shown in Figure 12.

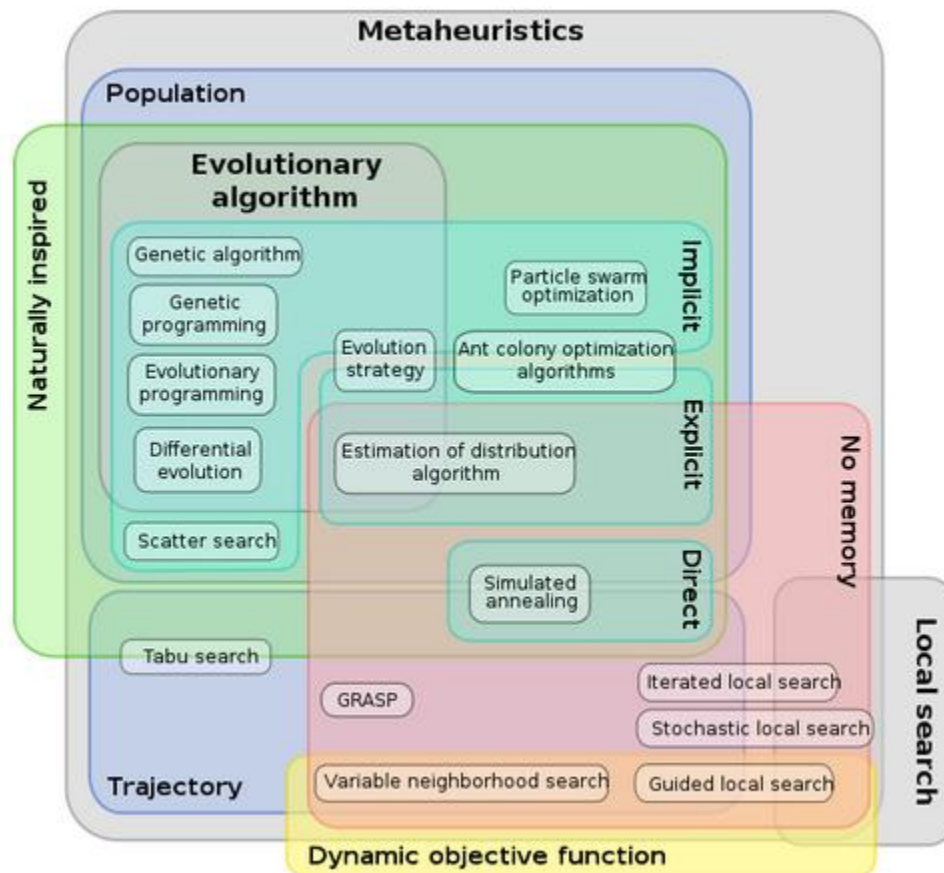


Figure 13: Classification of metaheuristics

2.6.1 Enumerative schemes

Enumerative schemes are probably the simplest search and optimisation methods. Each possible solution is evaluated within some defined finite search space. However, it is easily seen that this technique is inefficient, or even infeasible, as search spaces become large. Because many real-world problems are computationally intensive, some way of limiting the search space must be applied to find acceptable solutions in an acceptable time (Michalewicz & Fogel, 2004).

2.6.2 Deterministic algorithms

Deterministic algorithms incorporate problem domain knowledge in an attempt to achieve this limiting of the search space. Many of these are considered graph or tree search algorithms and are described here as such. **Greedy algorithms** make locally optimal choices, assuming optimal subsolutions are always part of the globally optimal solution (Brassard & Bratley, 1988;



Husbands, 1992). Thus, if that assumption does not hold, these algorithms fail. **Hill-climbing algorithms** search in the direction of steepest ascent from the current position. These algorithms work best on unimodal functions, but the presence of local optima, plateaus, or ridges in the fitness or search landscape reduces algorithm effectiveness (Russell & Norvig, 1995). Figure 14 and Figure 15 show instances where hill-climbing algorithms are typically ineffective, except for the case of a unimodal function, as shown in Figure 16.

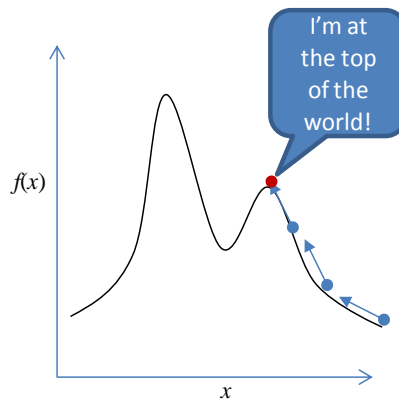


Figure 14: Hill-climbing algorithm stuck at a local optimum.

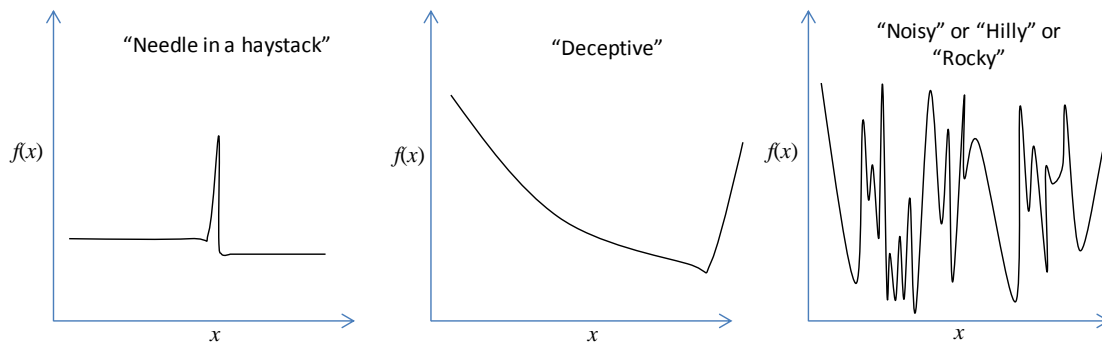


Figure 15: Functions that present difficulties for hill-climbing algorithms.

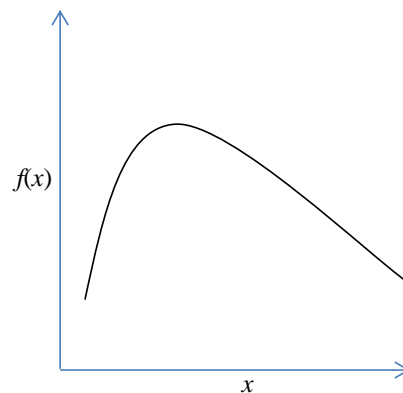
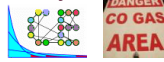


Figure 16: A unimodal function



Greedy and hill-climbing strategies can be classed as *irrevocable*. In other words, they repeatedly expand a node, examine all possible successors (then expanding the “most promising” node), whilst no record is kept of past expanded nodes (Pearl, 1984). **Branch and bound search techniques** need problem specific *heuristics/decision algorithms to limit* the search space (Garey & Johnson, 1979; Pearl, 1984). These algorithms compute some bound at a particular node which determines whether that node is “promising”; several different nodes’ bounds are then compared whereafter the algorithm branches to the “most promising” node (Neapolitan & Naimipour, 1996). **Basic depth-first search** is classed as *blind or uninformed*. This means the search order is independent of the location of solutions (except when the search is terminated). It expands a node, generates all successors, expands a successor, and continues like that continuously. If the search is blocked (e.g. it reaches a tree’s bottom level) it resumes from the deepest node left behind (Pearl, 1984). **Backtracking** is a *depth-first* search variant which “backtracks” to a node’s parent if the node is determined “unpromising” (Neapolitan & Naimipour, 1996). **Breadth-first search** is also uninformed. It differs from depth-first search in its actions after node expansion. These algorithms progressively explore the graph one layer at a time (Pearl, 1984). **Best-first search** uses *heuristic information* to place numerical values on a node’s “promise”; the node with highest promise is examined first (Pearl, 1984). **A***, **Z***, and others are popular *best-first search variants*. These select a node to expand based both on “promise” of the node and the overall cost of arriving at that node. Lastly, **Calculus-based search** methods *require*, at the minimum, *continuity* in some variable domain for an optimal value to be found (Anton, 1984).

The abovementioned algorithms are all deterministic methods successfully used in solving a wide variety of problems. Any experienced scientist or engineer will know that many MOPs are high-dimensional, discontinuous, multimodal, and/or NP-Complete. Often, deterministic methods are ineffective when applied to NP-complete or other high-dimensional problems. The reason is that they are handicapped by their requirement for problem domain knowledge (heuristics) to direct or limit search in these exceptionally large search spaces. Problems exhibiting one or more of the above characteristics are termed *irregular* (Lamont, 1993). Because many real-world scientific and engineering MOPs are irregular, enumerative and deterministic search techniques are unsuitable.

2.6.3 Stochastic search and optimisation

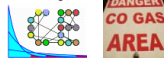
Stochastic search and optimisation approaches such as **Simulated Annealing** (SA) (Metropolis *et al.*, 1953) and later formalised by Kirkpatrick *et al.* (1983), **Monte Carlo methods** (Oszycza, 1978), **Tabu search** (Glover & Laguna, 1998), and **Evolutionary Computation** (EC) (Goldberg, 1989; Michalewicz, 1996; Bäck, 1996) were developed as alternative approaches for



solving these irregular problems. In addition to these methods, there are also **Bio-inspired algorithms** and **Gaussian Convolution**, a variation of Hill-Climbing (Luke, 2009). Bio-inspired algorithms include **Ant Systems** (AS) or **Ant Colony Optimisations** (ACO) (Dorigo *et al.*, 1996), **Particle Swarm Optimisation** (PSO) (Kennedy *et al.*, 2001), the **Firefly Algorithm** (Yang, 2008) and **Artificial Immune Systems** (Coello Coello & Cortés, 2005). There are yet others that are not shown in Figure 12, such as **Distributed Reinforcement Learning** (Watkins & Dayan, 1992), **Differential Evolution** (Storn & Price, 1997) and **Memetic algorithms** (Moscatto, 1989), which are all discussed in Bekker (2012). Another metaheuristic is the **Cross-Entropy Method** (CEM), which is a variance minimisation algorithm. It was developed by Rubinstein (1997) and later improved in Rubinstein (1999). As mentioned before, this metaheuristic forms part of the main focus of this thesis. Yet there are even more algorithms such as the original **No Free Lunch Algorithm** (Wolpert & Macready, 1997) and **Harmony Search** (Geem & Kim, 2001). Wikipedia also briefly discusses eight further metaheuristics (which the author has not specifically referenced nor expanded on), namely the **Altruism Algorithm**, the **Artificial Bee Colony** (ABC) Algorithm, **Glowworm Swarm Optimisation** (GSO), **Charged System Search**, **Cuckoo Search**, **Gravitational Search Algorithm**, **Intelligent Water Drops** (IWD), **Multi-Swarm optimisation**, **River Formation Dynamics** (RFD) (a gradient version of ACO), **Self-Propelled Particles** (SPP) and **Stochastic Diffusion Search** (SDS). It is quite astonishing how new stochastic approaches have developed from different fields of study in attempts to provide more effective and efficient methods to solve MOPs.

Coello Coello *et al.* (2007) state that stochastic methods require a function assigning fitness values to possible solutions and an encode/decode (mapping) mechanism between the problem and algorithm domains. Although some algorithms are shown to “eventually” find an optimum, most cannot guarantee the optimal solution, but rather an acceptable approximation. In general they provide good solutions to a wide range of optimisation problems which traditional deterministic search methods find difficult (Goldberg, 1989; Husbands, 1992). Some of the most popular stochastic optimisation algorithms of those listed above will be discussed below in slightly more detail: a brief description given of the mechanism of each algorithm and how it was adopted for MOO, if applicable.

A *random search* is the simplest stochastic search strategy. A given number of randomly selected solutions are simply evaluated. Similar to the random search is a *random walk*, the only difference being that the next solution evaluated is randomly selected using the last evaluated solution as a starting point (Vicini & Quagliarella, 1997). As is the case with enumeration, though, the latter two strategies are not efficient for many MOPs because of their failure to incorporate



problem domain knowledge. Goldberg (1989) states that random searches can generally expect to do no better than enumerative ones.

Simulated Annealing (SA) is an algorithm that explicitly models the annealing of metal. Annealing is a technique applied in materials science to alter a property of a material, such as its hardness. A common way of achieving this is heating a metal to liquid phase and then gradually cooling until it freezes. By mimicking this process mathematically, SA has been developed to an effective optimisation technique. Bekker (2012) provides a concise, yet thorough explanation of this technique with good references to applications. Where hill-climbing chooses the best move from some node, SA picks a random one. If the move improves the current optimum it is always executed, else the move is made with some probability $p < 1$. This probability exponentially decreases by either the time or the amount by which the current optimum is worsened (Dowland, 1993; Russell & Norvig, 1995). If a liquid material's temperature is lowered slowly enough, it attains a lowest-energy configuration. The analogy for SA is that if the "move" probability decreases slowly enough, the global optimum is found. Bekker (2012) states that Kirkpatrick *et al.* (1983) and Černý (1985) independently showed the analogy to combinatorial optimisation.

In general, *Monte Carlo methods* involve simulations dealing with stochastic events; they employ a pure random search where any selected trial solution is fully independent of any previous choice and its outcome (Osyczka, 1985; Schwefel, 1995). The current "best" solution and associated decision variables are stored as a comparator.

Tabu search (TS) is a meta-strategy developed to avoid getting "stuck" on local optima. TS acts as a local search procedure and iteratively allows a move to a good neighbour. Record is kept of both the visited solutions and the "paths" which reached them in different "memories." This information restricts the choice of solutions to evaluate next. TS is often integrated with other optimisation methods (Glover, 1986; Schwefel, 1995; Glover & Laguna, 1998). TS has been extended to MOO to become multi-objective tabu search (MOTS). Once again Bekker (2012) provides a good concise discussion of TS and references cases where TS has been applied for MOO.

Evolutionary Computation (EC) is a generic term for several stochastic search methods that simulate the natural evolutionary process computationally. As a recognised research field, EC is comparatively young, although its associated techniques have existed for about 45 years (Fogel, 1998). EC includes the techniques of genetic algorithms (GAs), evolution strategies (ESs), and evolutionary programming (EP), collectively known as Evolutionary Algorithms (EAs) (Fogel, 1995). These techniques are loosely based on the Darwinian concept of "Survival of the Fittest" (Goldberg, 1989). Aspects that are common between different EAs are the reproduction,



random variation, competition, and selection of contending individuals within some population (Fogel, 1995). In general, an EA consists of a population of encoded solutions (individuals), each comprising a number of genes which is equal across the individuals. The solutions are manipulated by a set of operators and the “fitness” of each is evaluated by some fitness function. The associated fitness of a solution determines which survive into the next generation. Although sometimes considered equivalent, the terms EA and EC can be distinguished to preserve the distinction between EAs and other EC techniques (e.g., genetic programming (GP) (Koza, 1992; Banzhaf *et al.*, 1998) and learning classifier systems (Lanzi *et al.*, 2002; Bull, 2004).

The complexity of MOPs and the shortcomings of deterministic search methods drove the creation of a number of optimisation techniques by the OR community. These methods can be linear or nonlinear, deterministic or stochastic, and can be grouped under the category *mathematical programming*. In these methods, constraints are treated as the main problem aspect (Schwefel, 1995). Linear programming is designed to solve problems in which the objective function and all constraint relations are linear (Hillier & Lieberman, 1967). Conversely, nonlinear programming techniques solve some MOPs not meeting those restrictions, but require convex constraint functions (Schwefel, 1995). Obviously, many problem domain assumptions must be satisfied when using linear programming and many real-world scientific and engineering problems can only be modelled by nonlinear functions. This is discussed in Hillier & Lieberman (1967) and Goel & Deb (2002). Finally, *stochastic programming* is used when random-valued parameters and objective functions subject to statistical perturbations are part of the problem formulation. Depending on the type of variables used in the problem, several variants of these methods exist (i.e., discrete, integer, binary, and mixed-integer programming) (Schwefel, 1995).

MOO using *evolutionary algorithms* (MOEA) has been actively researched over the past 25 years and the applications of MOEA have grown (Coello Coello *et al.*, 2007, p. 64). Two major references among the many journal publications are the books by Deb (2001) and Coello Coello *et al.* (2007). In a recent article by Coello Coello (2009), future research paths in the field of MOEA were highlighted. These include understanding the dynamics of the population of a GA over different generations better; defining the stopping criteria for a GA-based MOP technique better; and reformulating many real-world problems that are currently treated with EAs as if they had a single objective to handle all the objectives simultaneously. It is also noted that there is much focus on designing MOEAs that reduce the number of objective function evaluations, as these evaluations can be very expensive when optimising some real-world problems.

To end off this chapter the most popular and most applied MOEAs from the literature are highlighted. Arguably the best-known evolutionary-based algorithms are:



- The Multi-Objective Genetic Algorithm (MOGA) of [Fonseca & Fleming \(1993\)](#),
- The 2nd Generation Non-Dominated Sorting Genetic Algorithm (NSGA II) of [Deb et al. \(2002\)](#),
- The Niche-Pareto Genetic Algorithm (NPGA) of [Erickson et al. \(2002\)](#),
- The Pareto Archived Evolution Strategy (PAES) of [Knowles & Corne \(2000\)](#),
- The Strength Pareto Evolutionary Algorithm (SPEA) of [Zitzler & Thiele \(1999\)](#),
- The Multi-Objective Messy Genetic Algorithm (MOMGA) of [Van Veldhuizen & Lamont \(2000\)](#),
- The Pareto Envelope-based Selection Algorithm (PESA) of [Corne et al. \(2000\)](#).

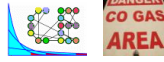
All of these algorithms and some of their variants are discussed extensively in [Coello Coello et al. \(2007\)](#). The Adaptive Range Multi-objective Genetic Algorithm (ARMOGA) of [Sasaki & Obayashi \(2005\)](#) requires relatively few objective evaluations to find the Pareto front and has been applied successfully in optimisation problems where computationally intensive objective evaluations are needed, for example in transonic wing design ([Oyama et al., 2001](#)). For a thorough, yet concise, discussion on MOEAs, the reader is referred to [Bekker \(2012\)](#).

2.7 Summary: Chapter 2

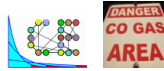
This chapter has aimed to lay the broad theoretical foundation for the research project. The field of MOO was introduced, presenting a brief historical overview of the development of the field as well as a short discussion on the mathematical foundations of the field. The concept of “optimality” in a multi-objective context, as well as other key terms and definitions from within the field of MOO, was discussed and formally defined.

The discussion of MOO was followed by an exposition of heuristics and metaheuristics, as metaheuristics are often applied to MOPs in the literature. Metaheuristics form a major subfield of stochastic optimisation which, in turn, forms a subfield of global search and optimisation. The chapter included a discussion on the broader field of global search and optimisation, which not only included metaheuristics, but also other subfields such as enumerative and deterministic techniques. The shortcomings and areas of best application of different techniques in each subfield were discussed. Some of the most popular metaheuristics were briefly discussed, whilst references to the researchers that developed and/or improved those algorithms were included. Lastly, also under stochastic techniques, one of its most popular subfields for MOO, namely MOEA, was discussed. The latter subfield is in itself rather broad and has really only been touched on in this chapter. Once again, authoritative works covering the important aspects of MOEA were referenced.

With the foundations laid for the reader to have a good understanding of the fields of MOO and metaheuristics, in the next chapter the focus will shift to a specific metaheuristic, the cross-



entropy method (CEM). With the research aim centering around the optimisation of a dynamic, stochastic real-world MOP, the CEM was the selected metaheuristic for the optimisation exercise, and needs to be discussed next.



3. The Cross-Entropy Method

In the previous chapter a scholarly review of the fields of MOO and metaheuristics was presented. Metaheuristics are often used in the optimisation of hard MOPs. Metaheuristics form a major subfield of stochastic optimisation which, in turn, forms a subfield of global search and optimisation. The different branches of global search and optimisation were discussed to put metaheuristics in perspective as an optimisation approach. It was shown that a wide variety of metaheuristics exists and the most popular metaheuristics, including the cross-entropy method (CEM), were briefly discussed. The focus of this chapter is to present the CEM in more detail by reviewing its origin and mathematical foundation, its main algorithm, how it was adapted for MOO and examples of its successful application.

The question may be asked why the CEM, of all the numerous existing metaheuristics was selected. In short, the answer would be that the CEM has been tested on complex mathematical test problems as well as noisy (stochastic) problems (Kroese *et al.*, 2006) and has been proven to be computationally inexpensive and effective in optimising many of those test problems. Further, the supervisor's research involved adapting the CEM for MOO and it was an aim of this thesis to serve as a case study for the supervisor's research. It was mentioned in section 1.3 that one of the most important contributions expected from this research, should the research hypothesis not be rejected, was that it would serve as a case study to the OR and IE communities. It would be a case study of the MOO CEM's successful application in optimising a dynamic, stochastic MOP from a real-world engineering environment.

3.1 Introduction to the CEM

The CEM is a relatively recent development by Israeli scientist, Reuven Rubinstein. He originally developed it to estimate probabilities of rare events in complex, stochastic networks (Rubinstein, 1997), but it has been extended to many other types of problems, including continuous optimisation and combinatorial optimisation (Rubinstein, 1999).

The core of the theory supporting the CEM for optimisation is briefly outlined in this chapter, but for detail the reader is referred to Rubinstein & Kroese (2004), Kroese & Rubenstein (2005) and the CEM website, www.cemethod.org (cited on 8 June 2012). The CEM for optimisation has its origin in *Importance Sampling* and the *Kullback–Leibler distance*. These concepts are discussed in depth in Rubinstein & Kroese (2004) and are briefly presented in section 3.2 of this chapter.



Figure 17: Reuven Rubinstein



The main idea behind using the CEM, whether for continuous multi-extremal optimisation or for combinatorial optimisation, is the same. The first step is to associate with each optimisation problem a rare event estimation problem. The latter is called the *associated stochastic problem* or ASP. The ASP is then tackled efficiently by an adaptive algorithm. The principal outcome of this approach is the construction of a random sequence of solutions which converges probabilistically to the optimal or near-optimal solution. As soon as the ASP is defined, the CEM involves the following two iterative phases:

1. Generation of a sample of random data (trajectories, vectors, etc.) according to a specified random mechanism.
2. Updating the parameters of the random mechanism, typically parameters of probability distribution functions, on the basis of the data, to produce a “better” sample in the next iteration.

These two iterative phases will be explained when the CEM algorithm discussed later in this chapter but, before the CEM algorithm can be considered, a look at the mathematical foundations underpinning the CEM is required.

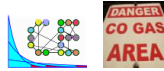
3.2 Mathematical foundations of the CEM

As mentioned at the beginning of this chapter, *Importance Sampling* and the *Kullback-Leibler distance* form the foundation of the CEM for optimisation and are discussed in this section. The discussion of the mathematical formulation of the CEM to follow in this section is based on that by [Rubinstein & Kroese \(2004\)](#).

A random sample vector, $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ from some space \mathcal{X} (i.e. a collection of all possible solutions), has a real function f . The real function, f , of each solution is the performance of that solution that needs to be optimised. In general, the probability that the performance, $f(\mathbf{X})$, is smaller than or equal to a real number γ under a family of probability density functions $h(\cdot; \mathbf{u})$ on \mathcal{X} needs to be determined. In this probability,

$$l = P_{\mathbf{u}}(f(\mathbf{X}) \leq \gamma) = E_{\mathbf{u}} I_{\{f(\mathbf{X}) \leq \gamma\}} \dots 16$$

If l is very small, the probability that $f(\mathbf{X})$ is smaller than or equal to γ , i.e. $f(\mathbf{X}) \leq \gamma$, is a *rare event*. *Importance sampling* can be used to efficiently estimate this probability. A random sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ is taken from a different density g on \mathcal{X} for this purpose. The following likelihood ratio estimator is used to estimate the value of l in this case ([Rubinstein & Kroese, 2004](#)):



$$\hat{l} = \frac{1}{N} \sum_{i=1}^N I_{\{f(\mathbf{X}_i) \leq \gamma\}} \frac{h(\mathbf{X}_i; \mathbf{u})}{g(\mathbf{X}_i)}. \quad \dots 17$$

This can be written as

$$g^*(\mathbf{x}) = \frac{I_{\{f(\mathbf{x}) \leq \gamma\}} h(\mathbf{x}; \mathbf{u})}{l}, \quad \dots 18$$

which means

$$l = \frac{I_{\{f(\mathbf{X}_i) \leq \gamma\}} h(\mathbf{X}_i; \mathbf{u})}{g^*(\mathbf{X}_i)}. \quad \dots 19$$

Here, g^* is an estimation of the optimum and depends on the unknown value of l . However, the value of g^* can be approximated within the family of densities $\{h(\cdot; \mathbf{v})\}$. The reference parameter in this case is \mathbf{v} . The approximation of the value of g^* is done by minimising the distance between g^* and $h(\cdot; \mathbf{v})$. The *Kullback-Leibler distance* or *cross-entropy* between g and h is the measure of this distance, and it is defined as:

$$\mathcal{D}(g, h) = E_g \ln \frac{g(\mathbf{X})}{h(\mathbf{X})} \quad \dots 20$$

$$= \int g(\mathbf{x}) \ln g(\mathbf{x}) d\mathbf{x} - \int g(\mathbf{x}) \ln h(\mathbf{x}) d\mathbf{x}. \quad \dots 21$$

For the distance between g^* in (18) and $h(\cdot; \mathbf{v})$ to be minimised, the value of \mathbf{v} must be selected in a way that will minimise $-\int g^*(\mathbf{x}) \ln h(\mathbf{x}; \mathbf{v}) d\mathbf{x}$. This minimisation can be achieved by solving the following maximisation problem

$$\max_{\mathbf{v}} \int g^*(\mathbf{x}) \ln h(\mathbf{x}; \mathbf{v}) d\mathbf{x}. \quad \dots 22$$

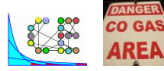
The replacing of g^* of (18) in (22), results in the following maximisation formulation

$$\max_{\mathbf{v}} \int \frac{I_{\{f(\mathbf{x}) \leq \gamma\}} h(\mathbf{x}; \mathbf{u})}{l} \ln h(\mathbf{x}; \mathbf{v}) d\mathbf{x}. \quad \dots 23$$

This corresponds to

$$\max_{\mathbf{v}} D(\mathbf{v}) = \max_{\mathbf{v}} E_{\mathbf{u}} I_{\{f(\mathbf{X}) \leq \gamma\}} \ln h(\mathbf{X}; \mathbf{v}). \quad \dots 24$$

With reference to the above, the CEM for optimisation will now be formulated in the next section.



3.3 Continuous and Discrete Optimisation with the CEM

The mathematical foundation explained above underpins both the continuous and discrete forms of the CEM. The different forms were developed to allow the CEM to be applied to the optimisation for both problems of continuous nature (i.e. the search space comprises continuous variables) and discrete nature (i.e. the search space comprises discrete variables). The discrete case of the CEM is analogous to the continuous case and the formal mathematical formulation of both is presented in this section. The steps of the CEM process will be explained in layman's terms after the mathematical formulations.

3.3.1 The CEM for continuous optimisation

Suppose one wishes to find the minimum of some performance function $f(\mathbf{x})$ over all states \mathbf{x} in some set \mathcal{X} . Let the minimum be γ^* , then

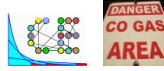
$$\gamma^* = \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}). \quad \dots 25$$

The deterministic problem is randomised by defining a family of probability density functions $\{h(\cdot; \mathbf{v}); \mathbf{v} \in \mathcal{V}\}$ on the set \mathcal{X} . The associated stochastic problem of (25) is the estimation problem

$$l(\gamma) = P_{\mathbf{u}}(f(\mathbf{X}) \leq \gamma) = E_{\mathbf{u}} I_{\{f(\mathbf{X}) \leq \gamma\}}. \quad \dots 26$$

The vector \mathbf{X} is a random vector with probability density function $h(\cdot; \mathbf{u})$ for some $\mathbf{u} \in \mathcal{V}$. When estimating l , $\{f(\mathbf{X}) \leq \gamma\}$ can be considered a rare event, and l can be estimated by making adaptive changes to the probability density function using the Kullback-Leibler cross-entropy. A sequence of probability density functions $h(\cdot; \mathbf{u}); h(\cdot; \mathbf{v}_1); h(\cdot; \mathbf{v}_2), \dots$ is created, which is steered in the direction of the theoretical optimal density. One generates a sequence of tuples $\{(\hat{\gamma}_t; \hat{\mathbf{x}}_t)\}$ that converges to the optimal tuple $(\gamma^*; \mathbf{v}^*)$, and setting $\mathbf{v}_0 = \mathbf{u}$, the procedure is as follows (Rubinstein & Kroese, 2004):

- 1. Adaptive updating of γ_t .** For a fixed \mathbf{v}_{t-1} , let γ_t be the $(1 - \delta)$ -quantile of $f(\mathbf{X})$ under \mathbf{v}_{t-1} . That is, $P_{\mathbf{v}_{t-1}}(f(\mathbf{X}) \leq \gamma_t) \geq \delta$, with δ typically chosen as $\delta = 10^{-2}$. Now estimate t by drawing a random sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ from $h(\cdot; \mathbf{v}_{t-1})$ and determine the sample $(1 - \delta)$ -quantile of the performances:



$$\hat{\gamma}_t = f_{(\lceil(1-\delta)N\rceil)}. \quad \dots 27$$

2. Adaptive updating of \mathbf{v}_t . For t and \mathbf{v}_{t-1} derive \mathbf{v}_t by solving the program

$$\min_{\mathbf{v}} D(\mathbf{v}) = \min_{\mathbf{v}} E_{\mathbf{v}_{t-1}} I_{\{f(\mathbf{X}) \leq \gamma_t\}} \ln h(\mathbf{X}; \mathbf{v}). \quad \dots 28$$

The value of $\min_{\mathbf{v}} D(\mathbf{v})$ in (28) can be estimated with the stochastic program

$$\min_{\mathbf{v}} \hat{D}(\mathbf{v}) = \min_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^N I_{\{f(\mathbf{X}_i) \leq \hat{\gamma}_t\}} \ln h(\mathbf{X}_i; \mathbf{v}). \quad \dots 29$$

The parameter vector $\hat{\mathbf{v}}$ can be updated using a smoothing function, as follows:

$$\hat{\mathbf{v}}_t = \alpha \tilde{\mathbf{v}}_t + (1 - \alpha) \hat{\mathbf{v}}_{t-1} \quad \dots 30$$

where $\tilde{\mathbf{v}}_t$ is obtained from (30) and α is a smoothing constant typically in the range 0.6 to 0.8. Based on the above, the main Algorithm of the CEM for continuous optimisation by Rubinstein & Kroese (2004) is shown as Algorithm 1. The same authors proved that the CE optimal density is often the atomic density at \mathbf{x}^* . The discrete case of the CEM for optimisation will be shown next.

Algorithm 1 Main Algorithm of the CEM: continuous optimisation

1. Choose some $\hat{\mathbf{v}}_0$. Set $t = 1$.
 2. Generate a sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ from the density $h(\cdot; \hat{\mathbf{v}}_{t-1})$ and compute the sample $(1 - \delta)$ -quantile $\hat{\gamma}_t$ of the performances according to (27).
 3. Use the same sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ and solve the stochastic program in (29). This solution is $\tilde{\mathbf{v}}_t$.
 4. Smooth the vector $\tilde{\mathbf{v}}_t$ using (30).
 5. If for some $t \geq \omega$, say $\omega = 5$, $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \dots = \hat{\gamma}_{t-\omega}$, then stop, otherwise set $t \leftarrow t + 1$ and return to Step 2.
-

3.3.2 The CEM for discrete optimisation

The discrete optimisation case of the CEM is analogous to the continuous case. Suppose the minimum of Ψ over \mathcal{X} is γ^* , then similar to (25):

$$\Psi(\mathbf{x}^*) = \gamma^* = \min_{\mathbf{x} \in \mathcal{X}} \Psi(\mathbf{x}). \quad \dots 31$$



The CEM requires that an estimation problem be associated with the optimisation problem of (31). To do so, one defines a collection of indicator functions on \mathcal{X} for different $\{I_{\{\Psi(\mathbf{x}) \leq \gamma\}}\}$ values of the threshold $\gamma \in \mathbb{R}$. Let $\{f(\cdot; \mathbf{v}); \mathbf{v} \in \mathcal{V}\}$ be a family of *discrete* probability mass functions on \mathcal{X} that are parameterised by a real-value vector \mathbf{v} . To solve the problem associated with (31), assume $\mathbf{u} \in \mathcal{V}$ and estimate the probability

$$l = P_{\mathbf{u}}\{\Psi(\mathbf{X}) \leq \gamma\} = \sum_{\mathbf{x}} I_{\{\Psi(\mathbf{x}) \leq \gamma\}} f(\mathbf{x}; \mathbf{u}) = E_{\mathbf{u}} I_{\{\Psi(\mathbf{X}) \leq \gamma\}} \quad \dots 32$$

with $f(\mathbf{x}; \mathbf{u})$ being the probability mass function on \mathcal{X} and γ some chosen level. Suppose now γ is equal to γ^* , then $l = f(\mathbf{x}^*; \mathbf{u})$, which is a very small probability. It can be estimated with importance sampling by taking a random sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ from a different probability mass function g and estimating l via:

$$\hat{l} = \frac{1}{N} \sum_{k=1}^N I_{\{\Psi(\mathbf{X}_k) \leq \gamma\}} \frac{f(\mathbf{X}_k; \mathbf{u})}{g(\mathbf{X}_k)} \quad \dots 33$$

which is the unbiased *importance sampling estimator* of l . The optimal way to estimate l is to use the change of measure with a different probability mass function

$$g^*(\mathbf{x}) := \frac{I_{\{\Psi(\mathbf{x}) \leq \gamma\}} f(\mathbf{x}; \mathbf{u})}{l}. \quad \dots 34$$

Since this optimal probability mass function is generally difficult to obtain and depends on the unknown l , one chooses g such that the cross-entropy or Kullback-Leibler distance between g and g^* is minimal. The Kullback-Leibler distance between two probability mass functions g and h is defined as before:

$$\mathcal{D}(g, h) = E_g \left[\log \frac{g(\mathbf{X})}{h(\mathbf{X})} \right] \quad \dots 35$$

$$= \sum_{\mathbf{x}} g(\mathbf{x}) \left[\log \frac{g(\mathbf{x})}{h(\mathbf{x})} \right] \quad \dots 36$$

$$= \sum_{\mathbf{x}} g(\mathbf{x}) \log g(\mathbf{x}) - \sum_{\mathbf{x}} g(\mathbf{x}) \log h(\mathbf{x}). \quad \dots 37$$

Since $\{I_{\{\Psi(\mathbf{x}) \leq \gamma\}}\}$ is non-negative, and the probability mass function f is parameterised by a finite dimensional vector \mathbf{v} , therefore $f(\mathbf{x}) = f(\mathbf{x}; \mathbf{v})$, $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{v})$ for some reference



parameter, \mathbf{v} , (Kroese, 2011). To estimate l in (33), one chooses \mathbf{v} such that $\mathcal{D}(g^*; f(\cdot; \tilde{\mathbf{v}}))$ is minimal. That means $E_{\mathbf{v}}\{I_{\{\Psi(\mathbf{X}_k) \leq \gamma\}}\} \log f(\mathbf{X}; \tilde{\mathbf{v}})$ should be maximal. Alon et al. (2005) showed that for discrete random vectors \mathbf{X} the components of $\tilde{\mathbf{v}}$ will always be of the form

$$\frac{E_{\mathbf{v}} I_{\{\Psi(\mathbf{X}) \leq \gamma\}} I_{\{\mathbf{X} \in A\}}}{E_{\mathbf{v}} I_{\{\Psi(\mathbf{X}) \leq \gamma\}} I_{\{\mathbf{X} \in B\}}} \quad \dots 38$$

with $A \subset B \subset \mathcal{X}$. This number can be estimated by taking a random sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ from the probability mass function $f(\cdot; \mathbf{v})$ and evaluating

$$\frac{\sum_{k=1}^N I_{\{\Psi(\mathbf{X}_k) \leq \gamma\}} I_{\{\mathbf{X}_k \in A\}}}{\sum_{k=1}^N I_{\{\Psi(\mathbf{X}_k) \leq \gamma\}} I_{\{\mathbf{X}_k \in B\}}} \quad \dots 39$$

One can use distributions p_j in the discrete optimisation problem to draw observations for random vectors $\mathbf{X}_i = (\mathbf{X}_{i1}, \dots, \mathbf{X}_{ind})$, for $j = 1, \dots, n_d$ elements in the decision vector. The estimator for p_j is (Alon et al., 2005),

$$\hat{p}_j = \frac{\sum_{i=1}^N I_{\{\hat{\Psi}(\mathbf{X}_i) \leq \gamma\}} I_{\{\mathbf{X}_{ij} = j\}}}{\sum_{i=1}^N I_{\{\hat{\Psi}(\mathbf{X}_i) \leq \gamma\}}} \quad \dots 40$$

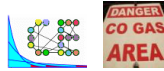
which has the same form as (39), with $P_0 = 0.5$. The smoothing update rule is similar to that in (30), and is shown in (41):

$$\hat{P}_t = \alpha \tilde{P}_t + (1 - \alpha) \hat{P}_{t-1} \quad \dots 41$$

The probabilities in P will approach zero or one after a sufficient number of iterations. The optimisation algorithm for the discrete case is shown in Algorithm 2 for N row vectors and n_d elements in the decision vector \mathbf{X} .

Algorithm 2 Main Algorithm of the CEM: discrete optimisation

1. Assign the elements of \hat{P}_0 the value 0.5. Set $t = 1$.
2. Generate a sample $\mathbf{X}_1, \dots, \mathbf{X}_N$ using \hat{P}_{t-1} , and compute the sample quantile $(1 - \delta)$ -quantile $\hat{\gamma}_t$ of the performance function according to (27). (1)



3. Using the same sample $\mathbf{X}_1, \dots, \mathbf{X}_N$, update $\hat{\mathbf{P}}_t$ with (40).
4. Smooth $\hat{\mathbf{P}}_t$ with (41).
5. If for some $t \geq \omega$, say $\omega = 5$, $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \dots = \hat{\gamma}_{t-\omega}$, then stop, otherwise set $t \leftarrow t + 1$ and return to Step 2.

This concludes the mathematical formulation of the CEM. In the next section, the steps in the procedure of the CEM are discussed in layman's terms while illustrative examples are included.

3.4 Steps in the CEM procedure in layman's terms

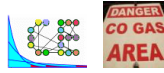
When applying the CEM, an iterative procedure is performed. The iterative stages that are followed once the ASP is defined were mentioned in the introduction to this chapter. References to the iterative stages were also made in the mathematical formulations in the previous section. This section expands on the iterative stages by breaking them up into more steps, as implemented by the author. The aim is to show, in a simplified way, how the CEM, in both the continuous and discrete case, progresses through the steps in its procedure.

3.4.1 Steps in the CEM procedure in the continuous case

1. Define the initial limits of the decision variables (DVs) as $LL_{i,t}$ and $UL_{i,t}$ with LL being the lower limit of a specific DV and UL the upper limit of the same DV. Let $i = 1, \dots, d$, where d is the number of DVs in the specific problem and t is the iteration counter.
2. Set $\sigma_{initial} = \sigma_{i,t}$, arbitrarily large at $10 \cdot (UL_{i,t} - LL_{i,t})$ and $\mu_{initial} = \mu_{i,t}$ a random value between $LL_{i,t}$ and $UL_{i,t}$, with $t = 0$.
3. Generate a population of predefined size N of random sample solutions based on a stochastic mechanism. With reference to Figure 18, the stochastic mechanism is explained below:

Let $\sigma_{i,t}$, $\mu_{i,t}$, $LL_{i,t}$ and $UL_{i,t}$ be the parameters of the stochastic mechanism. By calculating the ratio between the areas of the two differently shaded portions of a normal distribution ϕ , a value for population member $x_{ji,t}$ can be generated. Let the ratio between the two truncated portions of ϕ be shown as

$$\frac{\phi_{x_{ji,t}} - \phi_{LL_{i,t}}}{\phi_{UL_{i,t}} - \phi_{LL_{i,t}}} = r, \quad \dots 42$$



where r is a random value between 0 and 1. Then, if $x_{ji,t}$ represents DV_i in the j^{th} set of DVs out of N sets in the population after iteration t , $x_{ji,t}$ can be calculated by

$$x_{ji,t} = \phi^{-1}(r \cdot (\phi_{UL_{i,t}} - \phi_{LL_{i,t}}) - \phi_{UL_{i,t}}) \quad \dots 43$$

The stochastic mechanism allows for the generation of $x_{ji,t}$ for each DV_i for each set j of DVs in the population after iteration t . By applying the stochastic mechanism, a working matrix is created containing the entire generated population of DVs for a specific iteration. Table 1 shows an example of such a working matrix.

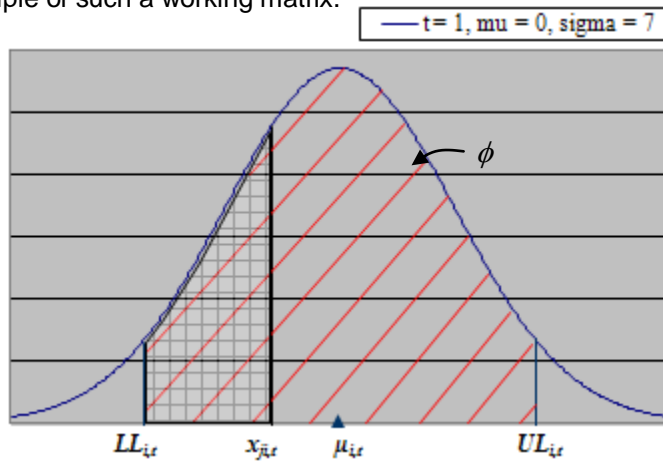
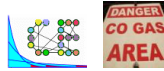


Figure 18: Stochastic mechanism for creating sample solutions in the continuous case of the CEM

Population of DVs after iteration t				
DV set	DV_1	DV_2	...	DV_d
$j=1$	x_{11}	x_{12}	...	x_{1d}
\vdots	\vdots	\vdots		\vdots
$j=N$	x_{N1}	x_{N2}	...	x_{Nd}

Table 1: Working matrix with population of DVs generated

4. For each set of DVs in the population, calculate the objective function value.
5. The objective function values are sorted from worst to best.
6. A specified percentage of the objective function values is selected from the bottom up as an **Elite** set. The **Elite** set thus contains a specified percentage of the best objective values from a population.
7. The **Elite** set after iteration t , together with a smoothing parameter α , are then used to adjust the parameters of the stochastic mechanism to $\sigma_{i,t+1}$, $\mu_{i,t+1}$, $LL_{i,t+1}$ and $UL_{i,t+1}$.



8. These adjusted parameters are then used in (43) to generate the new population of N solutions in the next iteration, $t = t+1$.
9. This process is repeated until the best, or a good enough, solution has been found in the **Elite** set.

Step 7 is effectively a variance minimisation exercise that causes the algorithm to converge towards the optimum values for each DV. Figure 19 illustrates how the mean value ($\mu_{i,t}$) of DV_i shifts towards the optimum value through variance minimisation.

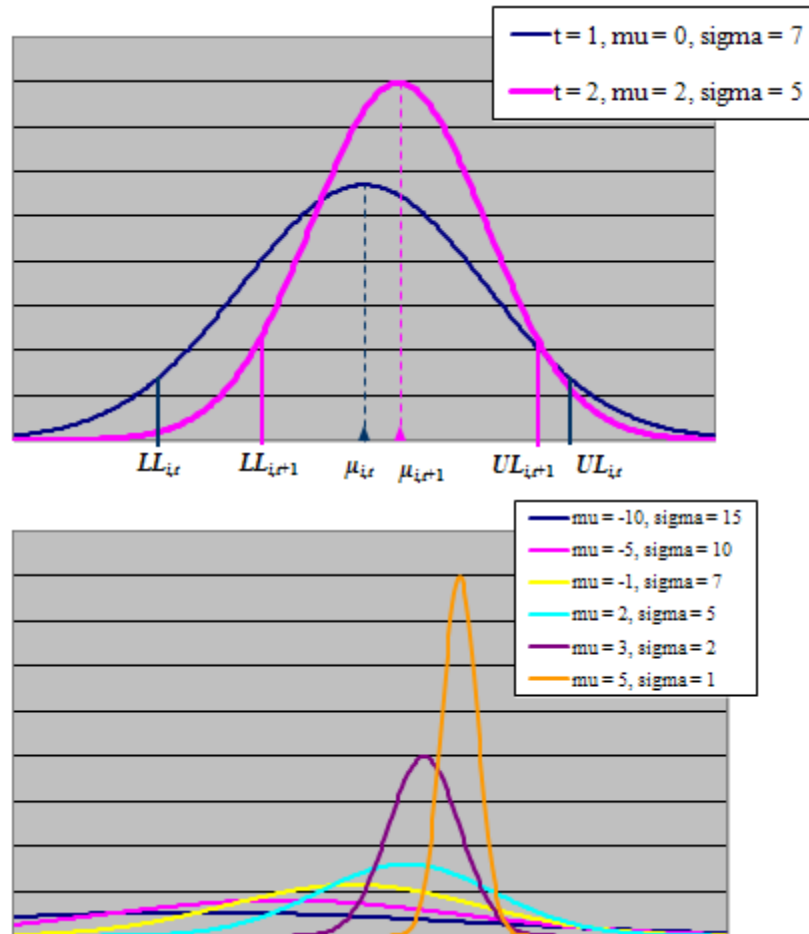


Figure 19: Variance minimisation as applied by the CEM

3.4.2 Steps in the CEM procedure in the discrete case

In the discrete case, the steps are similar to those for the continuous case, the main difference being the algorithm initialisation and the stochastic mechanism. Steps 1 to 3 for the continuous case change to the following:



1. Instead of setting the limits, a probability vector is created for each DV. Set the size of the probability vector \mathbf{P}_i of DV_i equal to the number of classes which the value of the specific discrete DV_i can take, say m_i . (It will be shown later how \mathbf{P}_i will become the stochastic mechanism). Let $i = 1, \dots, d$, where d is the number of DVs in the specific problem and t is the iteration counter.
2. Assign the initial values of each class of $\mathbf{P}_{initial} = \mathbf{P}_{i,t}$ as $1/m_i$, with $m_i > 0$, effectively making the initial probability vector a uniform discrete distribution at iteration $t = 0$.
3. Similar to the continuous case, generate a population of predefined size N of random sample solutions using $\mathbf{P}_{i,t}$. If $x_{ji,t}$ again represents DV_i in the j^{th} set of DVs out of N sets in the population after iteration t , $x_{ji,t}$ can be generated according to the probability $p_{iq,t}$ in $\mathbf{P}_{i,t}$, where $q = 1, \dots, m_i$. The working of \mathbf{P}_i as the stochastic mechanism is explained below with reference to Figure 20.

Probability vector for discrete CEM

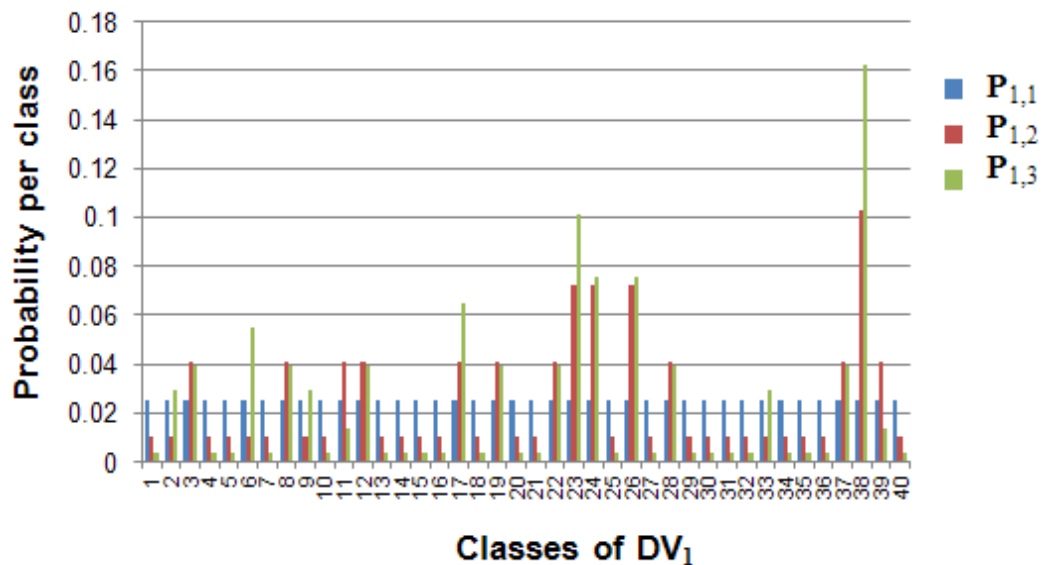
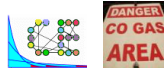


Figure 20: Stochastic mechanism for creating sample solutions in the discrete case of the CEM, for DV_1 with $m_1 = 40$

Referring to the example shown in Figure 20, where values are generated for each class of DV_1 with $m_1 = 40$, it is clear that after iteration $t = 3$, the probabilities, $p_{1,38,3}$ and $p_{1,23,3}$ have been adjusted to be the highest, whilst other probabilities, like $p_{1,1,3}$, have decreased. Thereby the likelihood is increased that values from classes 38 and 23 (and accordingly for other



classes) of DV_1 will be generated in the next iteration $t = 4$. A working matrix similar the one shown in [Table 1](#) is generated.

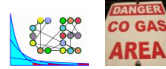
4. Repeat steps 4 to 6 of the continuous case.
5. The **Elite** set with the best objective values, together with a smoothing parameter, α , are then used to adjust and update the probabilities in \mathbf{P}_i to get $\mathbf{P}_{i,t+1}$ with probabilities $p_{iq,t+1}$. Simply put, the probability of each class of a specific DV is updated based on the occurrence of values from each class for that DV in the **Elite** set.
6. These adjusted and updated probabilities in $\mathbf{P}_{i,t+1}$ are then used to generate the new population of size N in the next iteration, $t = t+1$.
7. This process is repeated until the best, or a good enough, solution has been found in the **Elite** set.

This concludes the theoretical overview of the CEM and its formulation for optimisation. In the next section examples of the successful application of the CEM to a few single objective optimisation problems are discussed.

3.5 Applications of the CEM for Single-Objective Optimisation

From the literature it is clear that the CEM has been successfully applied to a variety of optimisation problems, including combinatorial optimisation problems and to rare-event estimation problems, the latter with both light and heavy-tailed distributions. It is important to note that the CEM deals successfully with both deterministic problems, such as the travelling salesman problem, and noisy (i.e., simulation-based) problems, such as the buffer allocation problem, state [Kroese et al. \(2006\)](#) and [Bekker & Aldrich \(2011\)](#). [Kroese et al. \(2006\)](#) also show the accurate performance of the CEM for solving both constrained and unconstrained, difficult continuous multi-extremal problems. The latter article in particular shows that typically the CEM finds the optimal (or near-optimal) solution very fast and provides more accurate results than those obtained by other metaheuristics as reported in the literature.

Most recently, [Bekker \(2012\)](#) illustrated the CEM for optimisation by applying it to four continuous, single-objective functions defined on finite spaces. These are the bowl function, the Rosenbrock function, the Shekel function and the Rastrigin function. The reader is referred to [\(Bekker, 2012\)](#) for details. Bekker shows that the CEM was effectively applied to solve all four of the problems in a minimum number of iterations.



In 2011 the CEM was applied in a study by [Leonard \(2011\)](#) involving apron layouts for planned new developments at Lanseria Airport in Gauteng, South Africa. Real-life flight schedules from South Africa's biggest airport, O.R. Tambo International, were used to simulate the arrivals and departures of flights with true passenger counts. O.R. Tambo Airport's flight schedules were used to forecast flight arrivals and departures at Lanseria Airport after its expansion. Aircraft movement on the apron was also simulated, while passenger transport distances were measured. [Leonard \(2011\)](#) applied the CEM and computer simulation in an original way by determining gate assignments that will minimise passenger transport distance.

[Lü et al. \(2008\)](#) developed a parallel leader-based algorithm using the CEM to solve the maximum clique problem. Twenty-five selected benchmark problems were tested and it was established that the CEM algorithm was significantly faster than benchmark algorithms in finding the solutions. [Evans et al. \(2007\)](#) presented a general method for designing parallel cross-entropy algorithms to be used on multiple instruction multiple data (MIMD) machines and message passing interface (MPI) library routines. Application of the CEM to project management is illustrated in [Cohen et al. \(2005\)](#). The CEM found good project loading parameters, even in stochastic environments. Other application areas, listed from the most recent to the oldest found in the literature, include ([Kroese et al., 2006](#); [Bekker, 2012](#)):

- heavy-tail distributions ([Asmussen et al., 2005](#); [Asmussen & Kroese, 2006](#)),
- buffer allocation ([Alon et al., 2005](#)),
- vehicle routing ([Chepuri & Homem-de-Mello, 2005](#)),
- network reliability estimation ([Hui et al., 2005](#)),
- CE convergence ([Margolin, 2005](#)),
- reinforcement learning ([Menache et al., 2005](#)),
- repairable systems ([Ridder, 2005](#)),
- queuing models of telecommunication systems ([De Boer, 2000](#); [De Boer et al., 2004](#)),
- Image segmentation ([Dubin, 2004](#)),
- policy search ([Mannor et al., 2003](#)),
- neural computation ([Dubin, 2002](#)),
- static simulation models ([Homem-de-Mello & Rubinstein, 2002](#)),
- DNA sequence alignment ([Keith & Kroese, 2002](#)),
- scheduling ([Margolin, 2002](#)),
- max-cut and bipartition problems ([Rubenstein, 2002](#)),
- control and navigation ([Helvik & Wittner, 2001](#)).

The CEM tutorial ([De Boer et al., 2005](#)) and [Rubinstein & Kroese \(2004\)](#) both contain further excellent references to CEM applications. As far as could be determined, the author did not find



examples where the CEM has been used for optimisation in heavy minerals processing or the field of furnace off-gas handling.

As discussed in section 2.6, there are many different optimisation techniques. It is not the intention of this chapter to compare the CEM with other optimisation techniques. Kroese *et al.* (2006) argue that, broadly speaking, in comparison with other metaheuristics the CEM is not only based on fundamental principles (cross-entropy distance, maximum likelihood, etc.), but is also relatively easy to program (with far fewer parameters than many other metaheuristics). The same authors also add that the CEM gives consistently accurate results and is therefore worth considering when faced with a difficult optimisation problem.

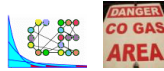
3.6 CEM for Multi-Objective Optimisation

So far in this chapter, discussions and formulations around the CEM have focused mainly on its application in single objective optimisation. In previous sections the theoretical basis of the CEM for optimisation has been presented. Now, the MOO CEM algorithm (Bekker, 2012) can be discussed. MOO CEM is the acronym for *multi-objective optimisation using the cross-entropy method* and is the algorithm that was applied in the optimisation exercise in this research project. The proposed algorithm is based on Algorithm 1 outlined in section 3.3.1 and is explained in this section by description and pseudocode.

The algorithm requires the construction of a working matrix \mathbf{W} comprising of N rows and $D+K+1$ columns, with N random, initial solutions. D is the number of decision variables (DVs) and K is the number of objectives. Sample values of the first DV are stored in column 1, of the second DV in column 2, and so on up to column D . Values of the first objective function are stored in column $D + 1$, of the second objective in column $D + 2$, and for the K^{th} objective K in column $D+K$. The last column, $D+K+1$, is reserved to store the rank value ρ of each solution. The ranking mechanism is explained later in this section. The structure of the working matrix is shown in Table 2.

Decision variables				Objectives				Rank
X_{11}	X_{12}	...	X_{1D}	f_{11}	f_{12}	...	f_{1K}	ρ_1
\vdots	\vdots		\vdots	\vdots	\vdots		\vdots	\vdots
X_{N1}	X_{N2}	...	X_{ND}	f_{N1}	f_{N2}	...	f_{NK}	ρ_N

Table 2: Structure of the working matrix \mathbf{W}



By using a truncated normal distribution for each DV a sample vector \mathbf{X}_i is formed from the density $h_i(\cdot; \hat{\mathbf{v}}_{i-1})$. Figure 21 shows an example of a truncated normal distribution. For the D decision variables defined over ranges $[l_i; L_i]$, l_i is the lower limit and L_i the upper limit of DV x_i , $1 \leq i \leq D$. The truncated normal distribution ϕ_i , defined in the range $[l_i; L_i]$ with mean μ_i and variance σ_i^2 , is given by

$$\phi_i(x) = \begin{cases} 0, & x < l_i \\ \frac{h_n(x)}{\int_{l_i}^{L_i} h_n(x) dx} & l_i < x \leq L_i \\ 0, & x > L_i. \end{cases} \quad \dots 44$$

The function $h_n(x)$ is the normal probability density function defined on $-\infty < x < \infty$. The reason for using truncated distributions is that it simplifies containing the search. The CEM requires that an arbitrarily large value for σ_i is initially assigned, hence $\sigma_i = 10(L_i - l_i)$ is set as the initial value. The first D columns of the working matrix are filled with sample values from each applicable truncated normal distribution, ϕ_i .

Subsequently, each of the objective functions is evaluated using the row vectors $\mathbf{X}_{1i}, \dots, \mathbf{X}_{Ni}$ of Table 2. Two or more performance vectors $f_j(\mathbf{X})$ are calculated with $1 \leq j \leq K$. Notice that this is

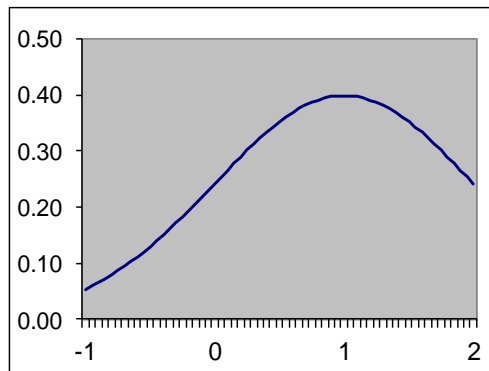
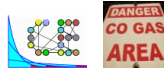


Figure 21: Truncated normal distribution on $-1 \leq x \leq 2$, $\mu = 1$, $\sigma = 1$

opposed to the single vector $f(\mathbf{X})$ in the original CEM. The $(1 - \delta)$ -quantile (γ) cannot be estimated, because ranking one objective function will not necessarily yield a good estimate of the other objective function(s). The best combinations of objective function values need to be found by doing a Pareto ranking as shown in Algorithm 3 (Goldberg, 1989). The working matrix containing N rows and $D + K + 1$ columns is provided for the algorithm, and the columns numbered $D + i - 1$, $(1 < i \leq K)$ are sorted consecutively.



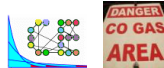
After sorting column i , $D + 1 \leq i \leq D + K - 1$, the $i + 1^{\text{th}}$ column is ranked. The ranking value of a solution indicates the number of solutions in the population that dominates that solution. This number is stored in column $D + K + 1$ of the working matrix. Recall that the aim is to obtain non-dominated solutions in the MOO exercise, thus a solution with a ranking value of zero is a non-dominated solution. When all the solutions have been ranked, those with a ranking value of less than or equal to a specified threshold value ρ_E are appended to an elite vector called **Elite**, which represents the current (weakly) non-dominated set.

A histogram is constructed for each decision variable using the decision variable values in the elite vector provided by Algorithm 3. While the algorithm searches for non-dominated solutions, the histograms provide guiding information for the MOO CEM algorithm and are thus maintained.

Algorithm 3 Pareto ranking algorithm (Minimisation)

1. Input: working matrix \mathbf{W} with (1 to N) rows and (1 to $D + K + 1$) columns, and user-selected threshold ρ_E .
 2. Set $j = D + 1$.
 3. Sort the working matrix \mathbf{W} with the values in column j in descending order.
 4. Set $r_p = 1$.
 5. Set $r_q = r_p$.
 6. If $\mathbf{W}(r_p, j+1) \geq \mathbf{W}(r_q + 1, j+1)$, increment the rank value ρ_{r_p} in $\mathbf{W}(r_p, D + K + 1)$.
 7. Increment r_q .
 8. If $\mathbf{W}(r_p, D + K + 1) < \rho_E$ and $r_q < N$, return to Step 6.
 9. Increment r_q .
 10. If $r_p < N$, return to Step 5.
 11. Increment j .
 12. If $j < D + K + 1$, return to Step 3, otherwise return the rows in \mathbf{W} with rank value not exceeding ρ_E as the weakly dominated or non-dominated vector **Elite**.
-

The histogram concept is implemented as follows: a decision variable x_i is defined on the range $[l_i; L_i]$. The lower boundary of the first class is set equal to l_i , and the upper boundary of the last class is set equal to L_i . Next, the minimum value of the decision variable x_i in the elite vector, i.e. $\min(\mathbf{Elite}(:, i))$, is set equal to the upper boundary of the first class. The lower boundary of the last class is equal to the maximum value of the decision variable in the elite vector, namely $\max(\mathbf{Elite}(:, i))$. A number of equal-sized classes are formed between these two



boundaries using $(\max(\mathbf{Elite}(\cdot; i)) - \min(\mathbf{Elite}(\cdot; i)))/r$ if r of these classes are formed, resulting in a total number of $r + 2$ classes (see Figure 22).

The class boundaries for the histogram of decision variable x_i are recorded in a vector $\mathbf{C}_i = \{c_{i1}; c_{i2}, \dots, c_{i(r+2)}; c_{i((r+2)+1)}\}$, with $c_{i1} = l_i$ and $c_{i((r+2)+1)} = L_i$. Note that \mathbf{C}_i contains $r + 3$ elements because the histogram has $r + 2$ classes, and that the class widths of the first class $([c_{i1}; c_{i2}])$ and the last class $([c_{i(r+2)}; c_{i((r+2)+1)}])$ can differ from each other and from the widths of the r classes.

The elite vector has the same columns as the working matrix shown in Table 2, and the values in column i , $1 \leq i \leq D$ are used to determine frequency values for decision variable x_i . The decision variable values are classified according to the following rule: X_{ij} belongs to the class $[c_{i\kappa}; c_{i(\kappa+1)})$ if $c_{i\kappa} \leq X_{ij} < c_{i(\kappa+1)}$, $1 \leq \kappa \leq r + 2$. The histogram frequency values are recorded in a vector $\mathbf{R}_i = \{\tau_{i1}; \tau_{i2}, \dots, \tau_{i(r+1)}; \tau_{i(r+2)}\}$, where τ_{i1} is equal to the frequency count of decision variable x_i in the range $[c_{i1}; c_{i2})$, τ_{i2} represents the count in range $[c_{i2}; c_{i3})$, and so on.

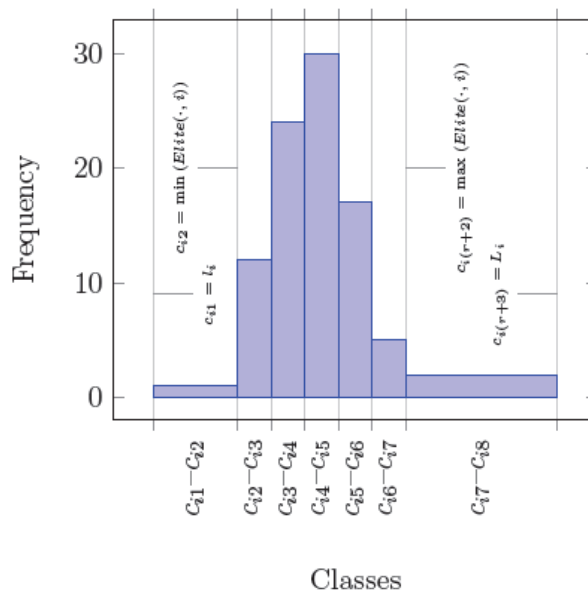
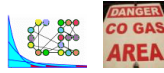


Figure 22: Example of a histogram for decision variable x_i and $r = 5$. Source: (Bekker & Aldrich, 2011)

In preparation for the next iteration of the algorithm, the new population of possible solutions is formed proportionally according to the class frequencies for each decision variable: Suppose the elite vector contains E_r rows and there are $\tau_{i\kappa}$ occurrences in class $[c_{i\kappa}; c_{i(\kappa+1)})$ for a given decision variable x_i , then $\lfloor N\tau_{i\kappa} / E_r \rfloor$ values will be created from this class range for this



variable (the population size is N and $1 \leq \kappa \leq r + 2$). When the proportional numbers do not add up to N due to the rounding down of the proportion calculation, the small difference is arbitrarily added to the last class. When generating observations from a class range $[c_{i\kappa}; c_{i(\kappa+1)}]$, temporary $\mu'_{i\kappa}$ and $\sigma'_{i\kappa}$ values are used. These values are associated with the specific histogram class ranges, so for the class $[c_{i\kappa}; c_{i(\kappa+1)})$ for decision variable x_i , the parameter estimators are $\mu'_{i\kappa} = c_{i\kappa} + U(c_{i(\kappa+1)} - c_{i\kappa})$, whereas $\sigma'_{i\kappa} = (c_{i(\kappa+1)} - c_{i\kappa})$, where $1 \leq \kappa \leq r + 2$ and U is a uniformly distributed random number.

To prevent premature convergence, the histogram frequencies are adjusted during each iteration t with a preset probability of typically $p_h = 0.1$ to $p_h = 0.3$. To do this, the maximum frequency over all classes is determined for a given decision variable. The frequency in each class is then subtracted from this value, which results in an inverted histogram as shown in [Figure 23](#). This ensures that search ranges that were given small proportions of population candidate allocations receive higher proportions of allocations, while search ranges with high proportions of population allocations receive fewer allocations after frequency inversion. The algorithm will re-adjust the frequencies according to the rankings returned by the candidates, so that a class that does not contribute to the elite vector effectively becomes eliminated as the search advances.

[Bekker & Aldrich \(2011\)](#) state that the histogram approach allows for the accommodation of discontinuous search spaces. Independent samples are, however, drawn for the DVs, while one expects them to be correlated because certain combinations of decision variable values yield the non-dominated objective values. By increasing the number of classes as the search progresses it is possible to maintain good combinations of DV values, because the resolution of the DV spaces becomes finer. To avoid the algorithm becoming inefficient, the number of classes should, of course, not grow too large. In line with the “No free lunch for optimisation” theorem ([Wolpert & Macready, 1997](#)) for single-objective optimisation (and extended by [Corne & Knowles \(2003\)](#) for MOO) the construction of the histogram sets adds to the computational burden.

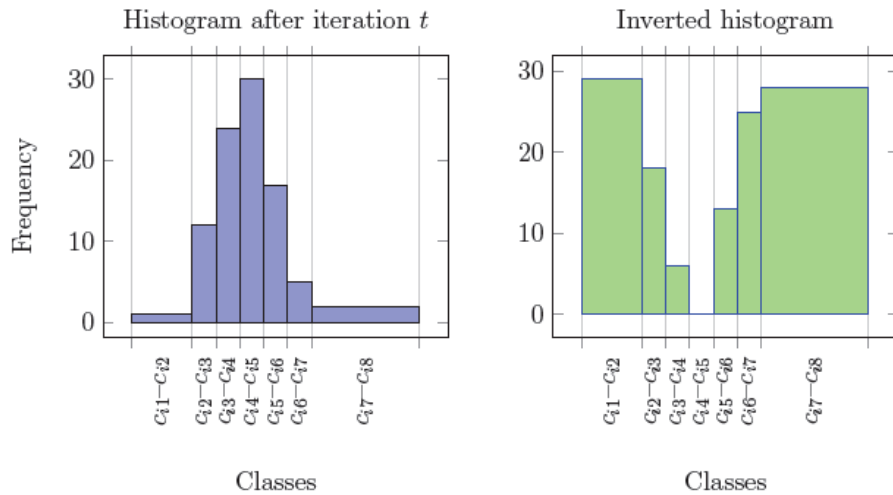
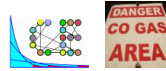


Figure 23: The effect of adjusting histogram frequencies for DV x_i .
Source: (Bekker & Aldrich, 2011)

Since one deals with more than one objective and to ensure exploitation, the process described above is repeated several times as an outer loop of the algorithm. After each loop, the elite vector is ranked again and the number of classes of the histograms is incremented. The algorithm is presented in pseudocode as Algorithm 4.

Algorithm 4 MOO CEM Algorithm

1. Set **Elite** = \emptyset , $t = 1$, $k = 1$.
2. Initialise variable vectors $\mathbf{X}_i = \emptyset$, $1 \leq i \leq D$, and compute initial objective values.
3. For each decision variable \mathbf{x}_i , $1 \leq i \leq D$, initialise a histogram class vector $\mathbf{C}_i = \{c_{i1}, c_{i2}, \dots, c_{i(r+2)}, c_{i((r+2)+1)}\}$ and histogram frequency vector $\mathbf{R}_i = \{\tau_{i1}, \tau_{i2}, \dots, \tau_{i(r+2)}, \tau_{i((r+2)+1)}\}$
4. Set $i = 1$.
5. Set $\kappa = 0$.
6. Increment κ .
7. Do for frequency element $\tau_{i\kappa}$ in \mathbf{R}_i :
8. Generate a class-based $\tilde{\mathbf{v}}'$ in the range $[c_{i\kappa}, c_{i(\kappa+1)})$, $1 \leq \kappa \leq r+2$.
9. Generate a subsample \mathbf{Y} according to the pdf $\phi_i(x_i, \tilde{\mathbf{v}}')$ with $\mathbf{x}_i \in [c_{i\kappa}, c_{i(\kappa+1)})$ and $|\mathbf{Y}| = \tau_{i\kappa}$, $1 \leq \kappa \leq r+2$.
10. Append \mathbf{Y} to \mathbf{X}_i .
11. If $\kappa < r+2$ return to Step 6.
12. Invert the histogram counts with probability p_h .
13. Increment i .
14. If $i \leq D$, return to Step 5.



15. Compute the $N K$ objective values using $\mathbf{X}_i, 1 \leq i \leq D$.
16. Rank the objective values using the Pareto ranking of Algorithm 3 with a relaxed $\rho_E = 2$ to obtain an updated elite vector **Elite**.
17. Form new histogram class vectors \mathbf{C}_i and histogram frequency vectors \mathbf{R}_i based on **Elite**, $1 \leq i \leq D$.
18. Use the values in **Elite** and compute $\tilde{\mathbf{v}}'_{it}$ for all $i, 1 \leq i \leq D$.
19. Smooth the vectors $\tilde{\mathbf{v}}'_{it}$ for all $i, 1 \leq i \leq D$, using (30).
20. If all $\sigma_{it} > \varepsilon_c$ or less than the allowable number of evaluations has been done, increment t and reiterate from Step 4.
21. Rank the elite vector **Elite** using the Pareto ranking of Algorithm 3 with $\rho_E = 1$.
22. Increment k .
23. If k is less than the allowable number of loops, return to Step 2.
24. Rank the elite vector **Elite** using the Pareto ranking of Algorithm 3 with $\rho_E = 0$ to obtain the final elite vector.

The parameter vectors $(\mu_i; \sigma_i)$ are updated using (30) and the values in the DV columns of the elite vector. For example, the $\sigma_{1,t}$ -value of the first decision variable is updated as follows after iteration t :

$$\hat{\sigma}_{1,t} = \alpha \tilde{\sigma}_{1,t} + (1 - \alpha) \hat{\sigma}_{1,t-1}, \quad \dots 45$$

with $\alpha = 0.7$ in all cases. This process is continued until the σ_i -value of each decision variable has decreased below a common threshold ε_c . When the algorithm terminates, the elite vector should contain the solutions on the Pareto front, as well as the associated decision variable values. This allows the Pareto front to be easily plotted if the objective functions are two or three.

The initial ranking threshold is relaxed to ensure exploration and exploitation of the search. For this, a value of $\rho_E = 2$ is selected. This means that solutions having a ranking of zero to two are included in the initial elite vector, while the true dominating set will have a ranking value of zero for all solutions. At the start of a new loop, when a new population is formed, the elite vector is trimmed by setting the threshold to one. When the algorithm terminates, the existing elite vector is refined a last time. The threshold then used is zero, which means all solutions selected are non-dominated.

3.7 Summary: Chapter 3

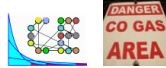
In this chapter the selected metaheuristic for optimisation of the research problem, namely the CEM, has been discussed in detail. The chapter started with an introduction to the CEM with references to its origins.



This was followed by an exposition of the mathematical foundations of the CEM. Based on the mathematical background, the formal formulation of the CEM was presented for both the continuous and discrete cases. In order to make the mechanism of the CEM more understandable, the steps in its procedure were explained, and illustrated using graphical depictions.

Next, a number of successful applications of the CEM for single-objective optimisation were discussed – a list, with references, of applications that were not explicitly discussed was also included. The chapter concluded with an in depth discussion of the MOO CEM algorithm. The in depth discussion is important as the MOO CEM algorithm was the technique used during the solution development phase in this research project.

The next chapter will discuss modelling and simulation as an OR problem solving technique, particularly with reference to its application in the research problem. The way in which MOO CEM was applied in conjunction with the simulation model will also be discussed in the next chapter.



4. Modelling and Simulation

The previous chapter was dedicated to a detailed discussion of the CEM, the metaheuristic that was selected for the optimisation of the multi-objective research problem. The discussion of the CEM included a review of its mathematical foundation, main algorithm, applications and how it was adapted for MOO. The chapter on the CEM followed naturally on its preceding chapter, wherein the broader fields of MOO and metaheuristics were discussed. The latter fields ultimately frame the narrative of this research project as they form the bulk of the theoretical background to the research problem. The research problem has been briefly introduced, in section 1.1, and has been referred to as the “CO gas problem” in previous chapters.

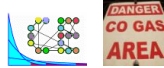
In this chapter the first theme, i.e. the theoretical background, of the thesis is concluded. The concepts and practice of modelling and simulation, more specifically discrete-event simulation, are discussed in this chapter. In order to study the behaviour of real-world entities, objects or systems, scientists and engineers often turn to modelling them. Modelling can take place on different levels and can be *physical*, *mathematical* or *conceptual*. This chapter will discuss some important aspects of modelling, particularly computer modelling.

It is often the case that a real-world system cannot easily be described analytically. In these cases a simulation model of the system can be developed in order to study the behaviour of the system under different sets of circumstances and over time. Simulation can, indeed, be seen as a subfield of modelling. A major aim of this chapter is to present simulation as a problem solving tool in the decision maker’s toolkit.

4.1 Modelling

Modelling is a broad term that may be described as the *representation of an entity, object or system* in any form other than itself. *Abstraction* is required during modelling and the reversal of abstraction is necessary for model interpretation (Taha, 2003). Models can be *prescriptive* (represent a proposed system) or *descriptive* (represent an existing system). Abstraction of a real-world system is achieved through identifying and incorporating into a model the dominant and/or relevant factors that control the real-world system’s behaviour (Taha, 2003). Through this, the real-world system can be represented to an acceptable degree of accuracy. Figure 24 shows a graphic depiction of how a model is, at best, an abstraction of a real-world system.

Models vary in the degree to which they represent reality. *Isomorphic models* comprise an exact agreement between elements of the model and the object itself. Exact relationships and interactions between elements are preserved in these models. *Homomorphic models* are similar to the real system in form, but different in fundamental structure. This difference can be attributed



to abstraction in representation. Simulation models are typically homomorphic, but the degree of isomorphism (extent to which the model agrees with reality) needs to be stated and tested if conclusions are to be drawn from the model. This process is known as model validation (Banks, 1998), and is a typical step in the overall simulation process which will be discussed later in this chapter.

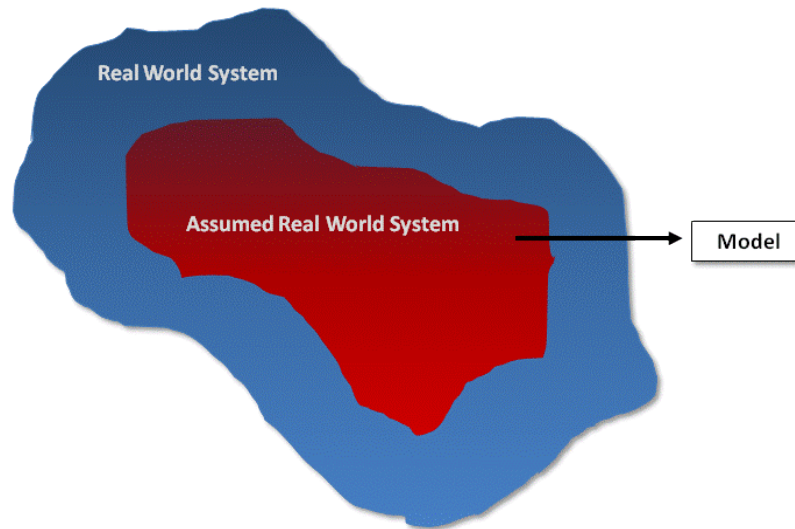


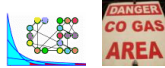
Figure 24: Models are abstractions of reality

A simulated system may be classified according to the level of abstraction required by the analyst. (Du Plessis, 2008) identifies three levels of abstraction:

Strategic level: The aim is to identify and analyse strategic organisational issues. Less detail is required for modelling on this level. Typical applications would include: a study of a corporate business model, a study of advertising strategies, estimation of the speed of the spread of a disease.

Operational level: Systems are usually modelled at this level when tactical decision making is needed. More detail is required here than on the strategic level. The reason for modelling at this level may be to investigate problems at an operational level, such as minimisation of stock levels in a warehouse, balancing production lines or locating bottlenecks by measuring process flow in a laboratory.

Physical level: The physical level may be seen as the “detail level” where the behaviours of individual objects with exact descriptions in terms of physical characteristics, e.g. size, weight, velocity, energy content etc. are relevant and are also modelled. Examples of modelling on this



level could include modelling the smelting process in an electric arc furnace, modelling the growth of a sugar cane plant or modelling the impact of a vehicle in a crash test.

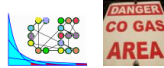
4.2 Simulation defined

OR incorporates creative scientific research into fundamental properties of operations (Hillier & Lieberman, 2005). Problems are generally approached with an operational optimisation or improvement outlook. Queuing and simulation together form one of the branches of OR (Taha, 2003), but are not limited exclusively to OR.

Hogg (2009) comments that simulation has, over the past two or three decades, consistently been reported as the most popular OR tool. Simulation refers to a wide compilation of methods and applications to predict real system behaviour through numerical evaluation using software – this specialised software being designed to replicate system operations and/or characteristics, usually over time (Kelton *et al.*, 2007). It involves the construction of a model of a real-world system, and experimenting with that model to understand the system's behaviour and/or evaluate operation alternatives (Pegden *et al.*, 1995). Banks (1998) defined simulation as “the imitation of the operation of a real-world process or system over time”. Banks went on to note in the same article that simulation “involves the generation of an artificial history of the system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented”. Simulation can therefore be seen as experimentation with a model of a real-world system, given certain starting conditions, to observe behaviour of the model and relate the behaviour back to the real-world system which the model represents. Asikainen (1995) claimed simulation to be “the next best thing to observing a real system”. Simulation is one of the most powerful tools available for evaluation and design of complex operating systems (Gallis, 1996).

It must be understood that *simulation is not an optimisation technique as such, but rather provides estimates of system performance through modelling* (Goulet *et al.*, 1980; Gallis, 1996; Hillier & Lieberman, 2005). It can thus be used to evaluate different alternatives within the system, acting as a tool for system improvement, but there is no guarantee that the final improved system is in fact an optimisation of the original (Hillier & Lieberman, 2005).

Simulation application is generally used in the analysis of complex real-world systems which cannot be assessed using analytic OR techniques due to the complexities of system component interaction. Numerous built-in parameters, variables and functions have led to simulation software coping with these interactions, which other analysis tools could not (Ziesak *et al.*, 2004).



4.3 Simulation in perspective

Simulation is included under the broader umbrella of modelling. It must be noted that simulation is not the only problem solving/system analysis tool with which to analyse complex problems. A number of other branches of modelling exist under, of which some are more appropriate than simulation for certain problems. Before a problem is modelled to be studied and potentially optimised, the analyst should make sure he/she has a good understanding of the different modelling tools in the OR field and select the appropriate one for the problem at hand. The following are OR techniques, other than simulation, which come under the umbrella of modelling:

- Queuing theory
- Linear Programming
- Assignment algorithms
- Integer programming
- Graph theory
- Markov chains
- Stochastic inventory models
- Dynamic programming

Linear programming can be used only when all the objective functions and constraints can be expressed linearly. If not, which is often the case with real-world problems, one moves into the field of non-linear OR techniques. As mentioned, simulation is often used to study complex, stochastic processes. Queuing theory and Markov chains, for example, allow for studying some stochastic processes analytically, but are often constrained by assumptions and process complexities. Consider for example how block arrivals at a simple queue soon cause the analysis to become difficult, while the situation is fairly easily dealt with by simulation (Bekker, 2005).

Simulation itself is made up of many branches, each of which is classified according to the type of model it produces. Figure 25 shows a breakdown of several modelling techniques. As mentioned before, the branch of simulation used in this study was discrete-event simulation and can be identified by the shaded blocks.

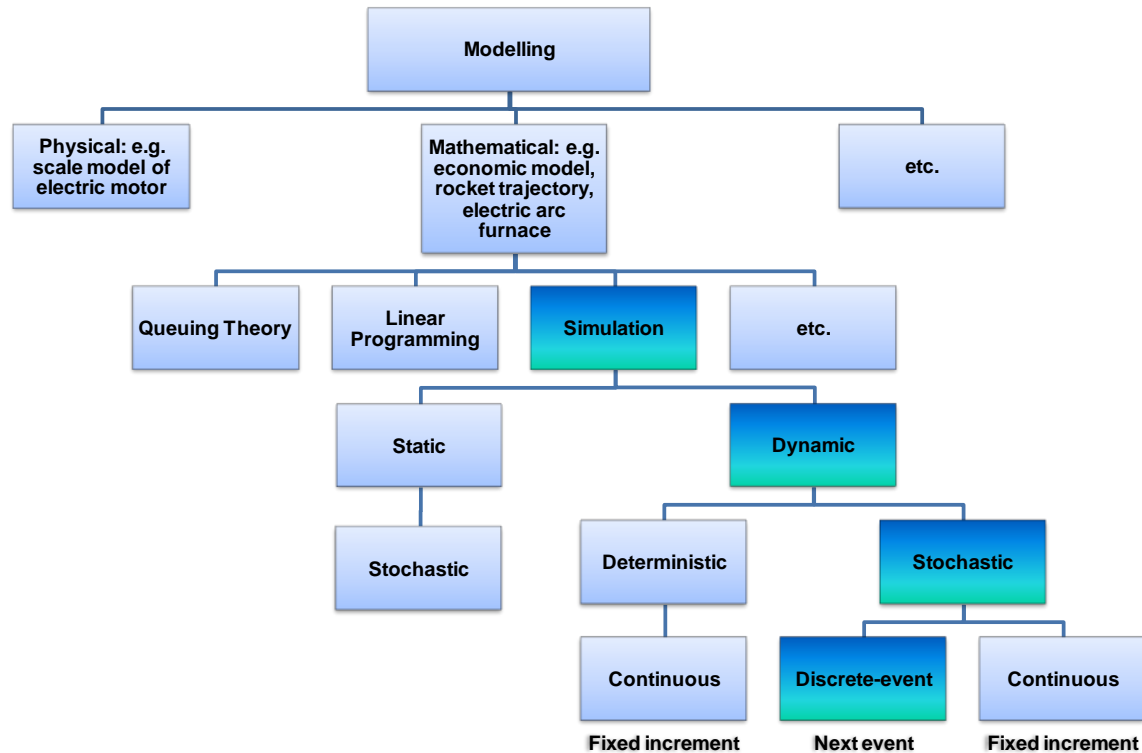
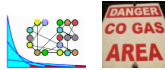


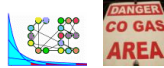
Figure 25: Simulation in perspective

4.4 Components of a simulation model

A simulation model comprises different components. These include the inputs, outputs and states, the entities and their attributes, activities and events, resources and statistical collectors. These components will be discussed in section 4.4.1 to 4.4.5.

4.4.1 Inputs, outputs and states

Certain happenings in the environment in which the system exists influence the system. These are the inputs into the system. In the CO gas problem one input is the specific combination of plants which are to receive CO gas for heating and/or drying purposes. Although numerous inputs may exist, the analyst must define a finite number of inputs in order to collect and feed input data into the simulation. The state of the simulated system depends on changes in the system conditions as a result of the inputs. The state of the CO gas distribution and burning system would include: the volume of CO gas produced by the furnaces; or the CO gas consumers that are operational and requesting CO gas at any given time, in other words, the supply-demand balance of CO gas. The outputs of the system are needed to answer questions about the system



and can be derived from the system state. In the CO gas problem, one of the outputs could be the average period of uninterrupted supply of CO gas at any of the gas consuming plants, or another the overall saving on methane gas costs (White & Ingalls, 2009).

4.4.2 Entities and attributes

In discrete-event simulation, the arrival of entities is the input for the simulation. The entities flow through the system and trigger events that change the state of the system variables. In the CO gas problem the entities are imaginary “batches” of CO gas produced per minute and a fictitious “checking element” that checks the combination of CO gas consumers that are requesting CO gas on a minute-by-minute basis. These entities have unique characteristics, called attributes (White & Ingalls, 2009). The attributes of a “batch” of CO gas could be its volumetric value and energy content.

Entities can be external and explicitly created and manipulated by the modeller, or internal and implicitly created and manipulated by the software. Resources provide a service to entities and often have limited capacity. For entities to be served by a resource, they have to compete or wait for the resource. When an entity has to wait for the use of the resource, it experiences a delay. The states of the entities change as they move through the system. These states can be the *active state*, *ready state*, *time-delayed state*, *condition delayed state* and *dormant state*. An entity is in the active state if it is moving. It will continue moving until it reaches a delay of some type. It will then go into another state and another entity will possibly enter the active state. In the case where two or more entities need manipulation at the same time, only one will continue moving while the others will experience a delay. The entities that are delayed are in what is known as the ready state. These entities are waiting to go into the active state. Sometimes, entities are delayed for a known time period before entering the ready state. These entities are in the time-delayed state. Entities may also be delayed until some condition is satisfied. These entities are in the condition-delayed state. When entities are delayed and cannot move into another state through changes in the condition of the model, they are in the dormant state (Scriber & Brunner, 2011).

4.4.3 Activities and events

Processes in a simulation are called activities and changes in the state of the system are called events. Entities interact with activities to create events. Activities are divided into three categories: *Delays*, *queues* and *logic*.

A *delay* is caused when the flow of an entity is put on hold for a definite time period. A delay in the flow of a batch of CO gas would be the time the system waits for a burner to switch over



from methane gas to CO gas, while there is actually enough CO gas to meet the demand. This time period may be either constant or random. When the delay of any entity starts, an event occurs. If the time period of the delay is d time units, then the entity will be delayed until the current time plus d time units has expired, after which it will carry on with its movement (be consumed by the burner which triggers the registration of another minute of uninterrupted CO gas supply) and create another event.

A *queue* occurs when an entity's movement is put on hold for an unspecified time period. In the CO gas problem there were no queues while there is no buffer tank, as the excess CO gas would simply be routed to another CO gas consumer or emitted into the atmosphere if no consumer can receive the CO gas.

Logic activities are manipulation of the state variables – thus, allowing an entity to affect the state of the system only if this is desirable. An example of a logic activity is the decision of which combination of consumers to supply with CO gas, which affects the supply-demand balance of CO gas (White & Ingalls, 2009).

4.4.4 Resources

Resources in a computer simulation are those things (units/equipment) that service entities and that have constrained capacities. In the CO gas problem the resources are the burners that burn CO or methane gas (White & Ingalls, 2009).

4.4.5 Statistical collectors

The parts of the simulation used to gather statistics on the values of performance measures or global variables are called statistical collectors. There are three types of statistics, namely *counts*, *time-persistent statistics* and *tallies*. *Counts* are used to count the number of minutes a burner runs on CO gas without interruption. *Time-persistent* collectors give a time-weighted average of a variable, for example the average methane gas saving at a gas consumer over a 24 hour period. *Tally* statistics are concerned with only one observation regardless of other observations or the time between them. An example of a tally statistic is the number of CO gas batches that are produced by Furnace 1 over a replication (White & Ingalls, 2009).

4.5 Steps in a simulation study

The following seven steps are used in the process of model development (Banks, 1998):

- problem formulation
- objective setting and overall project plan
- model conceptualisation



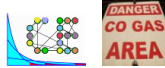
- data collection
- translation of concept model to computer model
- verification and validation
- experimental design and analysis

These steps were followed in the development of the simulation model in this study. Although, it should be added, these steps are not necessarily followed in the exact order proposed by Banks.

4.6 Some advantages, drawbacks and limitations of simulation

Most of the following advantages of simulation are discussed in (Banks, 1998):

- Simulation allows for the *study and evaluation of alternatives*.
- Simulation allows for a plan or the running of a system *to be visualised* by the simulation analyst in order to study the system.
- *Problems can be diagnosed and insight can be gained* into the working of a *complex system*. Certain otherwise difficult questions about real-world systems can be answered. *The question “why?”* regarding certain behaviour of a system can be addressed, as simulation enables a *microscopic examination* of the system in order to determine the reason for phenomena which occur.
- Through simulation, *constraints or critical parameters can be identified* and studied.
- Through simulation, changes to a system can be tested and experimented with, *without having to use any resources*.
- New policies or methods can be explored, *without disrupting the real system*.
- *Analysis* of the operations of a new system *before it is implemented*, thus helping to *minimise incurrence of unnecessary cost*.
- Through simulation a problem can be *studied at several levels of abstraction*. By approaching a system at a higher level of abstraction the designer is able to better understand the overall behaviour and interactions of the system. This prevents the designer from becoming mired in detail and losing focus.
- Simulation can be used as an effective means *for illustrating or demonstrating* concepts or designs, particularly where computer simulation software has *graphics and animation* functionalities. This can be very handy when *communicating simulation results* from different concepts/designs/alternatives to management.
- By *compressing and expanding time*, simulation allows the speeding up or slowing down of time in order to aid investigative phenomena.

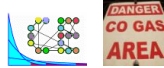


- Long (“never ending”) processes can be *studied in a relatively short time*.

Banks (1998) also includes most of the following disadvantages of simulation:

- Simulation *does not automatically generate optimal solutions* to problems. It just predicts the outcomes under a certain set of assumptions with certain input variables.
- Simulation studies *can be expensive* in many ways. Software packages are generally expensive. It takes time and effort from many stakeholders to complete a simulation study, which must be compared to the window period available in which an answer is required, as well as the opportunity cost of opting for a simpler, possibly less accurate method of study. In short, simulation can be too costly for a given problem.
- *Computationally intensive processing* is sometimes required for simulation and is further dependent on computational capacity of the software and hardware involved. As a consequence, the results of the simulation may take longer than expected before it becomes available.
- Data acquisition can be a *long, costly process*.
- The development of a model that imitates a real-world system accurately enough for meaningful inferences is an art, hence *adequately experienced/qualified simulation analysts* might not always be readily available. In an attempt to perform a simulation in the absence of the right analyst, inappropriate use of simulation may occur.
- The output of a simulation model normally has a *probabilistic nature* and will require *statistical background of the interpreter* (often of both the analyst who presents the results and the audience). Often simulation results are presented to an audience that does not fully grasp the probabilistic nature of the results and thereby loses some of the real meaning of the results.
- A model is often very *specific to a certain problem* and often cannot be generally applied to all related systems
- *Analysis quality and reliability depend on model quality and input data accuracy*. An incorrect model or poor input data may result in important decisions being taken based on incorrect model outputs.

Despite these disadvantages, simulation was deemed the appropriate technique for the experiment in this study. Simulation allowed for the representative modelling of this dynamic, stochastic system much more realistically than a static model would. The simulation model can be altered to evaluate the effect of future changes to the system such as more stable furnace operations, automated switchover between CO and methane gas or to determine the optimum volume required of a buffer tank.



4.7 Simulation and metaheuristics

Metaheuristics were discussed in sections 2.5.2 and 2.6.3. It was shown that metaheuristics is a subfield of stochastic optimisation, and indeed a major subfield thereof. A quick re-look at the first sentence of the [Wikipedia](#) definition of metaheuristics, says:

In computer science, metaheuristic designates a computational method that optimises a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

That “measure of quality” is often termed “objective function” in the literature. For example, consider again the hill-climbing algorithm aiming to find the top of a unimodal function such as the function $f(x) = x^2$. In this case the objective function is a simple parabola and a simple hill-climbing algorithm, or even an analytical method, the derivative, can easily be utilised to find the optimum. As soon as the objective function gets more complex, stronger optimisation algorithms are required. [Coello Coello et al. \(2007\)](#) and [Bekker \(2012\)](#) contain a number of test problems that are often used to compare the *performance quality* of optimisation algorithms. Yet although these objective functions are discontinuous, complex functions they are still mathematical functions which can be plotted with the correct software program, like *Matlab*®. Many real-world problems do not have known, or easy-to-find, mathematical functions or analytical models. This type of problem often requires a simulation model to estimate the objective function, although, as shown in section 4.3, simulation must not be taken as the only tool capable of modelling more complex problems.

Some of the well known more complex problems in the OR environment include the travelling salesman problem (TSP), the quadratic assignment problem (QAP) and the max-cut problem. These can be seen as a representative sample of combinatorial optimisation problems where the problem being studied is completely known and static. In contrast, the buffer allocation problem (BAP) is a noisy estimation problem where the objective function needs to be estimated, since it is unknown. For a problem like the BAP, discrete-event simulation is a very effective technique for estimating an unknown objective function.

Simulation therefore enables the analyst to estimate the unknown objective function of a system/problem, in fact, *a simulation model of a real-world problem (with its inputs and outputs) becomes the objective function where no (known) analytical model can be found.* [Figure 26](#) and [Figure 27](#) show the parallel between a (known) analytical objective function, and a simulation model that serves as an objective function.

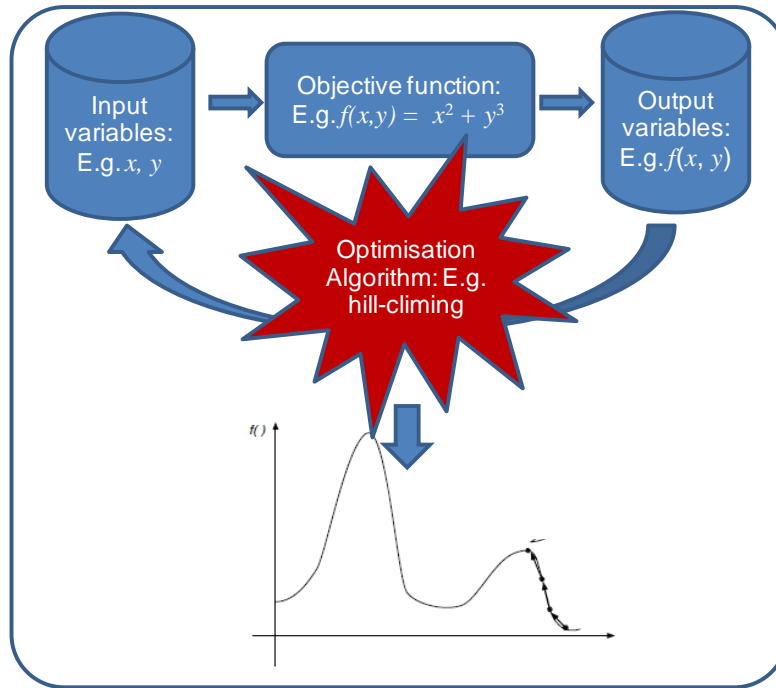
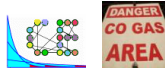


Figure 26: A model where a known analytical objective function exists

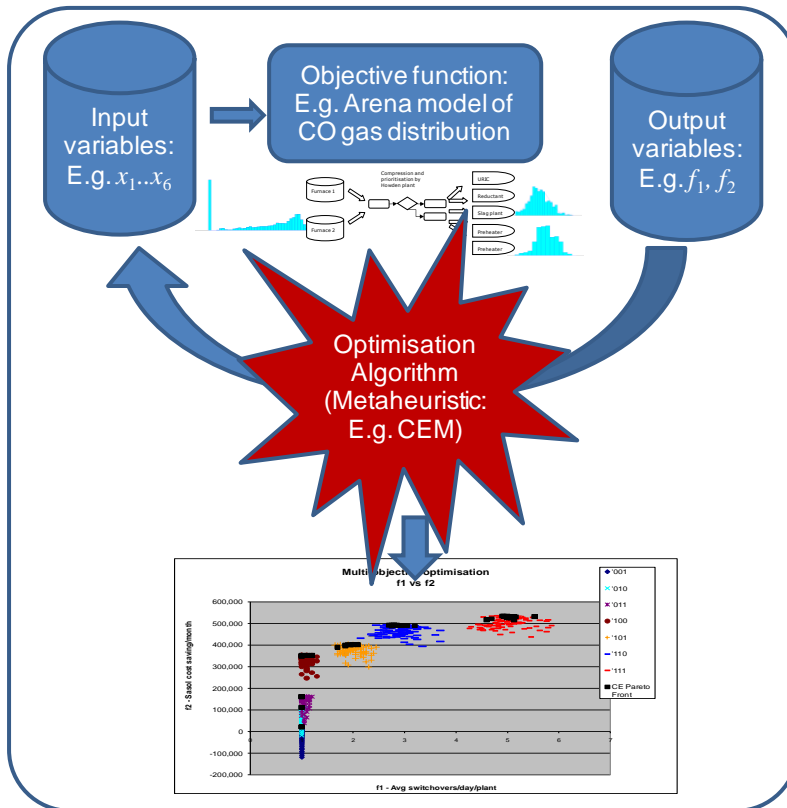
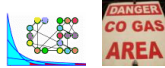


Figure 27: A model where no known analytical objective function exists

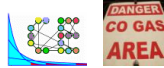


It was also noted in section 4.2 that simulation is not an optimisation technique in itself, but it provides estimates of system performance through modelling. Simulation is effectively used to evaluate different alternatives within the system. Simulation thereby acts as a tool for decision making and system improvement. However, as has been pointed out, *simulation itself does not guarantee that the final improved system is in fact the optimum system*. In order to find an optimum system, an optimisation technique needs to be applied to search for the optimum. This optimisation technique can be as simple as intelligently selected scenarios by the user, or as broad as random search but in large search spaces, metaheuristics are often applied for optimising. Some simulation software packages come with add-in optimisation packages. For example the *Arena*® software package comes with *OptQuest* as its optimisation package. This means that *OptQuest*, with some input (starting criteria and constraints) from the analyst, searches through different scenarios on the same simulation model until a satisfactory optimum is found. Notice how in both cases in [Figure 26](#) and [Figure 27](#) an optimisation algorithm (the red star) is required to search for the optimum outputs.

In a case where the objective function is a simulation model, depending on the software used, advanced programming skills may be required to “link” the objective function to the metaheuristic. Then again, with some software packages the optimisation package (containing certain metaheuristics), can be “linked”, without any programming, to the simulation model, for example *Arena*® and *OptQuest*. For the study of the CO gas problem, the author developed the simulation model in *Arena*® and coded the metaheuristic in *Matlab*®. The latter was then integrated with *Arena*® to allow the interaction between the simulation model and the metaheuristic.

The question may be asked, “How does the metaheuristic ‘know’ when to stop, i.e. how does the metaheuristic ‘know’ that an optimum has been found?” The answer to this question is dependent on the view of the simulation analyst. In the case of a mathematical function, it is often possible to plot the objective function and read the optimum from the graph. Often, with a simulation model, a plot is not arbitrary. The optimum is then taken as the optimum value or range of values found in a constrained objective space that the analyst has had to define before running the model. This optimum range of values will form the Pareto front. Especially in an unconstrained objective space, the user should decide whether he or she is satisfied with the optimum(s) found by the metaheuristic. This “satisfaction” is often a function of the metaheuristic terminating when a specified error value (ε -value) is reached, or after a predetermined number of simulation runs.

This section has aimed to explain the relationship between simulation and metaheuristics. This relationship will be further demonstrated in the chapters to follow when the solution development phase of the research project will be discussed in detail.



4.8 Summary: Chapter 4

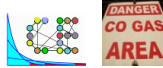
Chapter 4 has presented the reader with some important theoretical background to modelling and simulation. Simulation was placed in perspective as an OR problem solving tool to emphasise that, although simulation is a popular tool, care should still be taken to consider whether simulation is really the most suitable tool for solving a particular problem with.

The components of a simulation model were defined and discussed, as well as the generic simulation process as presented by [Banks \(1998\)](#). Some of the differentiating strengths and most commonly experienced drawbacks of simulation were listed.

With the aim of elucidating the link between this chapter and the previous one, the relationship between simulation and optimisation, particularly optimisation by means of metaheuristics, was discussed next. An important fact was reiterated: although simulation is effective for solving complex, stochastic real-world problems, simulation in itself does not necessarily guarantee an optimum solution. Either the simulation analyst must intelligently evaluate potential optimum scenarios with a simulation model or some of optimisation algorithm must be applied in conjunction with the simulation model.

This chapter concludes the first theme of the thesis as outlined in [Figure 1](#). The aim of the first theme was to lay the theoretical foundation of the study fields with which this research intersected. This was done through a literature review of the relevant fields and by establishing the connections between the most relevant aspects of those fields and the research problem.

The next chapter will cover the next short theme of the thesis, which is to contextualise the research problem in terms of the practical environment where it originated.



5. Problem context: Ilmenite smelting and furnace-offgas

In the previous chapter the theory and principles of modelling and simulation, which form a central part of the solution development in this research project, were discussed. The aim of the previous chapter was to discuss the most important aspects of modelling and simulation, as well as to show the relationship between simulation and optimisation, with the focus on optimisation with metaheuristics.

At this point the author regards it as important to elaborate on the research problem which was briefly discussed in section 1.1 and which has so far been termed simply the “CO gas problem”. This chapter forms the second theme of the thesis, which contextualises the CO gas problem by providing the reader with an overview of the industry, company and operational environment in which the problem arose. It is believed that an understanding of the problem context will equip the reader with meaningful insight into the logic and level of complexity of the CO gas problem.

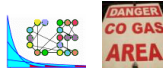
The first part of this chapter will focus on the most important aspects of the heavy minerals industry and *Tronox KZN Sands*, while the second half of the chapter will drill a level deeper to discuss the specific processes at *Tronox KZN Sands*, where the problem was identified. A brief discussion on CO and CO₂ gas is included. A good understanding of the nature of CO gas, especially, should add to the reader’s appreciation of the practical side of the research problem.

5.1 Heavy minerals industry

Tronox KZN Sands is a heavy minerals mine and processing company. Mineral sand deposits are widely distributed and are found on almost every continent in the world with the most significant of these deposits occurring in South Africa, Australia and North America. Once mined, the ore undergoes a series of unique beneficiation processes to separate and upgrade the valuable products. These deposits are usually composed of a number of minerals, however it is the zircon, ilmenite and rutile that are of main economic importance (Tyler & Minnitt, 2004; Oomardath, 2009).

- **Titanium minerals**

Titanium is widely distributed in the earth’s crust, and exists as the ninth most abundant element (Oomardath, 2009). It occurs naturally in two forms, namely *ilmenite* and *natural rutile*. The *ilmenite* form is a black sand or rock, with a chemical formula FeTiO₃. The TiO₂ content of ilmenite usually varies from 45% to 60% depending on the ore body. The other source of titanium is naturally occurring *rutile*. This varies in colour from brown to reddish black. The TiO₂ content of



this mineral is normally 94% to 95%. Apart from the naturally occurring forms of titanium, there are also *two enriched ores* which are important for commercial reasons. The first ore is *titania slag* which is obtained when one removes the iron from the ilmenite through smelting. The second enriched ore is *beneficiated ilmenite*, which is obtained when the natural ilmenite is treated by physical and chemical means to increase the TiO_2 content. An important product of the latter process is *synthetic rutile*.

Although titanium was discovered in 1791, commercial exploitation of the mineral only started in 1913, when titanium dioxide pigment was first manufactured by utilising a fusion process. This process has subsequently been superseded by the *sulphate and chloride processes*, the latter being the more favoured of the two today. The sulphate process, so named because of its extensive use of sulphuric acid, uses mainly ilmenite and titania slag; while the chloride process can utilise mineral rutile, beneficiated ilmenite (synthetic rutile), titania slag and natural ilmenite.



Figure 28: Rutile crystals

At *Tronox KZN Sands*, the smelting of ilmenite produces titania slag and low-manganese pig iron (LMPI). These products are sold to the pigment and foundry markets, respectively. Rutile is not smelted and is exported for the production of welding rods, titanium metal and also pigment.

- **Zirconium minerals**

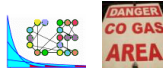
Zirconium is the other element commonly found in mineral sand deposits. This generally occurs in the form of Zirconium Silicate (Zircon). Zircon has a chemical formula ZrSiO_4 , and is usually a colourless mineral however, it may also range in colour from yellow to reddish. The radioactive decay of uranium and thorium also affects the colour of the minerals. Zircon is a very versatile mineral, however it is primarily used for the manufacture of ceramics, tile glazing and refractories (Selby, 2007).



Figure 29: Zircon grains

5.2 *Tronox KZN Sands*

This research was done on a project at the *Tronox KZN Sands* heavy minerals operation in Kwazulu-Natal, South Africa.



Heavy minerals operations in South Africa involve the mining of ilmenite, and zircon in coastal dunes. There are three active heavy minerals operations in South Africa: *Tronox KZN Sands* and *Richards Bay Minerals* (a joint venture between international mining houses *Rio Tinto* and *BHP Billiton*) on the east coast; and *Tronox Namakwa Sands* on the west coast. Although these different South African operations are similar in many ways, there are important differences, and it is important to understand that this research project refers to processes and business conditions at *Tronox KZN Sands*, where it was performed.

Figure 30 indicates the location of heavy minerals operations in South Africa and Figure 31 shows the three different sites – the Hillendale Mine, Fairbreeze mine and CPC – of *Tronox KZN Sands*. In Figure 32 an aerial picture shows the layout of the CPC that is situated outside the town of Empangeni.



Figure 30: Heavy minerals operations in South Africa

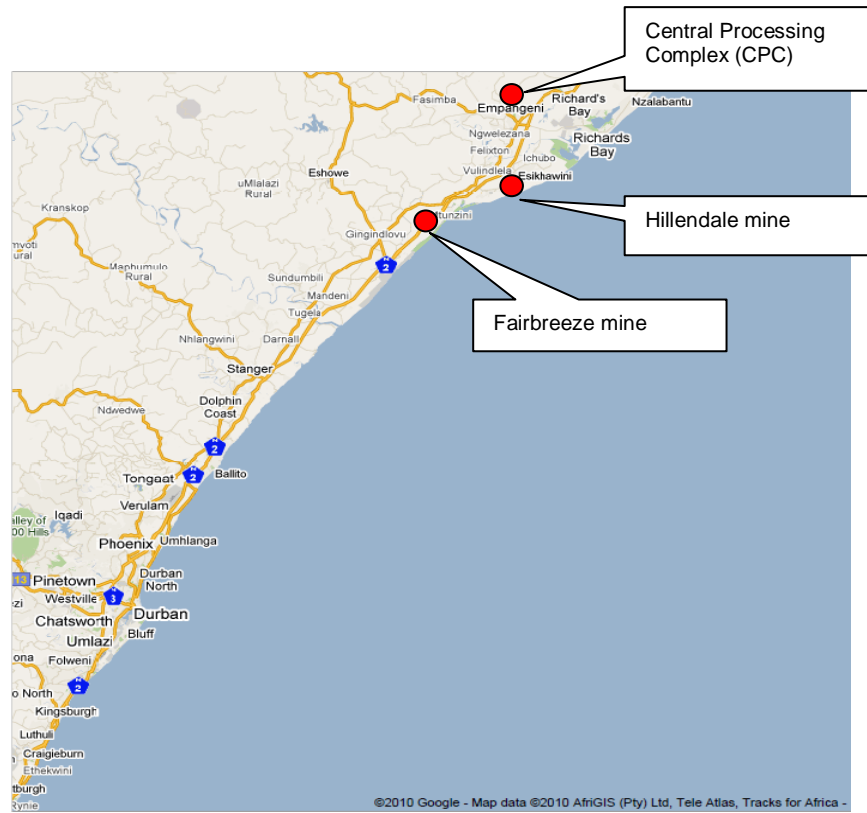
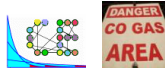


Figure 31: Location of the three *Tronox KZN Sands* sites on the South African east coast

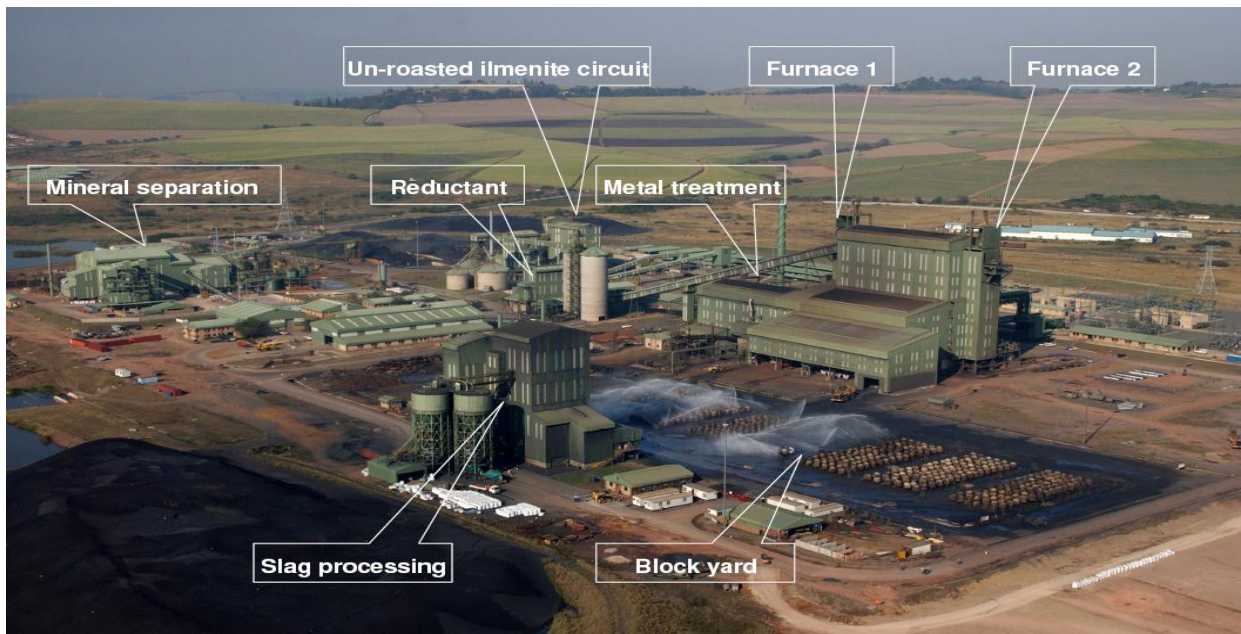
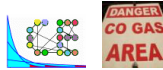


Figure 32: Central Processing Complex outside Empangeni (Northerly aerial view)



Ilmenite is smelted in two direct current (DC) arc furnaces. Zircon and rutile are not smelted, but are separated using gravitational, electrostatic and electromagnetic beneficiation technology. One of the by-products of the ilmenite smelting process is carbon monoxide gas (CO gas) and this is often called “furnace off-gas” or simply “off-gas”. This gas stream is captured in the off-gas system where the raw gas (a combination of CO gas, furnace dust and fine ilmenite particles) is cleaned. Currently at *Tronox KZN Sands*, most of the cleaned off-gas is flared into the atmosphere through emission stacks above the two furnaces. The basic chemical reaction taking place inside the ilmenite furnaces is shown in Figure 33, while Figure 34 shows the flaring of CO gas at the furnace emission stacks.

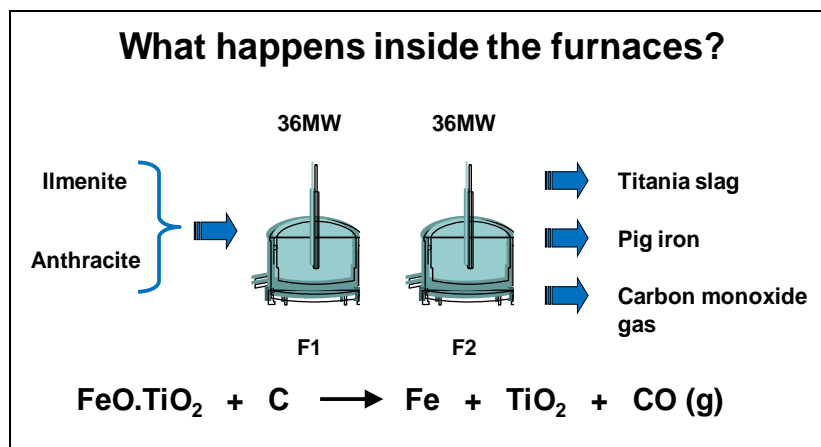


Figure 33: The ilmenite smelting process at *Tronox KZN Sands*

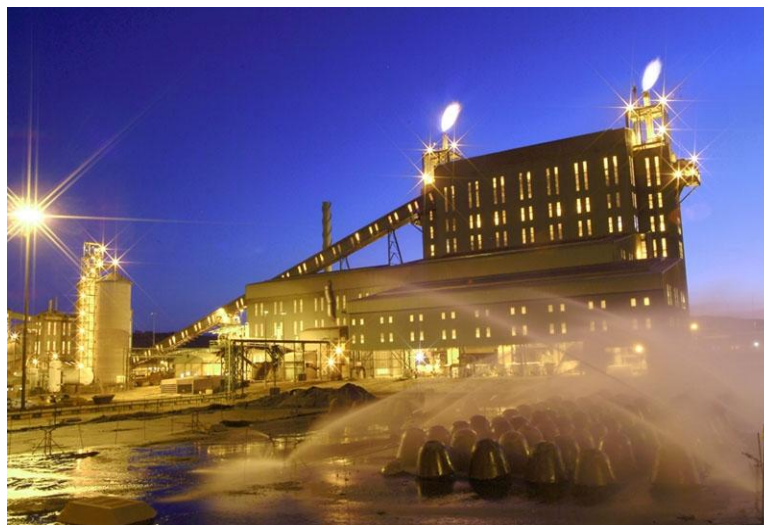
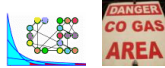


Figure 34: Flaring of CO gas at the furnace emission stacks

Some of the beneficiation processes at the CPC entail electrostatic and electromagnetic separation, while other processes involve screening to separate material into fractions of different

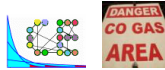


size. Electrostatic and electromagnetic separation and screening all require dry feed material in order to operate efficiently. Since the inception of *Tronox KZN Sands*, methane has been purchased from an external supplier to dry and heat feed material. The opportunity has always existed to utilise the furnace off-gas instead of buying methane gas. However, over the last decade this has not been achieved sustainably. The main benefits of utilising the furnace off-gas will be a significant cost saving due to a reduction in the consumption of methane gas and a reduction in the carbon footprint of the business. By burning CO gas instead of methane gas at the burners, the emission of CO₂ gas (formed during the combustion of methane in atmospheric oxygen) will be lowered. CO₂ is also formed when the CO gas is flared at the stacks but, by distributing CO gas to the burners, the necessity to burn methane gas there will be lowered significantly and thereby also the emission of CO₂ gas from the CPC. The potential exists to reduce the methane consumption by over 50%. The flaring of CO gas at the furnace stacks is the main contributor to the *direct* CO₂ emissions of the business and resulted in the emission of roughly 120 000 tons per year of CO₂. The latter calculates to approximately 24% of the total carbon footprint of the business. A 50% reduction in methane gas consumption could result in a 1.5% reduction of the carbon footprint, or a 5% reduction in direct CO₂ emissions.

From a capital projects point of view at *Tronox KZN Sands*, the aim of this research was to provide input into the feasibility study of the utilisation of the furnace off-gas at the burners at the various beneficiation plants at the CPC.

5.3 Company history and operational background

The business currently known as *Tronox KZN Sands*, started as a project in 1994 as *Iscor Heavy Minerals*, owned by the then *Iscor Mining*, a large South African mining company at the time. It later changed name to *IHM Heavy Minerals* when *Iscor Mining* became *Kumba Resources*, and belonged to the latter company. Later, the management of the company was temporarily sold by *Kumba Resources* to Australian company, *Ticor*, and the name changed again to *Ticor South Africa*. The first product was produced in 2001 as *Ticor South Africa*. Thereafter it was acquired back by *Kumba Resources* and since November 2006 the company has formed part of the diversified South African mining group *Exxaro Resources*, after *Kumba's* iron ore assets were sold to *Anglo American* and the rest of the company, including heavy minerals, formed *Exxaro*. In October 2011, *Exxaro* announced a planned deal with an international pigment and chemicals producer, *Tronox*, and since July 2012, the former *Exxaro KZN Sands* forms part of *Tronox* (www.miningweekly.com/article/exxaro-and-tronox-to-combine-mineral-sands-operations-2011-09-26, cited 2 June 2012), hence adopting the name *Tronox KZN Sands*. The new company listed on the New York Stock Exchange as *New Tronox*. *Exxaro Resources* will hold 38.5% share in *New Tronox* and maintain 26% direct interest in the South African operations.



Tronox KZN Sands is a major exporter to various parts of the world, including Europe, North America and the Far East. Due to its export activities, the business is an earner of foreign exchange for South Africa with a turnover exceeding R1.5 billion per annum, with the potential to more than double the current turnover with the commissioning of the new Fairbreeze mine. The company employs approximately 750 full time personnel and 300 temporary staff. The annual salary bill exceeds R300 million and approximately R530 million is spent on local procurement in the Kwazulu-Natal province. This total inflow of over R800 million per year into Kwazulu-Natal provides a significant boost to an otherwise poverty stricken province.

The existing Hillendale mine is at the end of its life. Full production will stop by the middle of 2013. The Fairbreeze mine is required to provide Heavy Mineral Concentrate (HMC) as feed to the Mineral Separation Plant (MSP) and the Furnaces beyond Hillendale. The regulatory processes to finalise the South African government's approval of Fairbreeze is currently still under way. Due to the location of the Fairbreeze ore body in an environmentally sensitive part of the northern Kwazulu-Natal coastal region, the plan to start the new mine is heavily contested by some public groups in the region. At the same time the opening of the mine is welcomed by other groups who hope that the mine will provide new employment opportunities, stimulate new economic activity, and secure the 1000 existing jobs and the economic injection that the business currently sustains. Without the Fairbreeze mine, the downstream mineral separation and smelting processes will also have to stop operations. Despite the numerous challenges, the Fairbreeze mine is expected to be operational by 2015.

In 2011, the South African government identified the beneficiation of minerals, specifically titanium and pigment, as a priority. *Tronox KZN Sands* is well placed to support this national strategy. In November 2011 the Minister of Mineral Resources confirmed the importance of titanium mining in South Africa, with titanium downstream beneficiation as one of the national growth path priorities. The transaction of selling the heavy minerals operations of *Exxaro Resources* to pigment producer, *Tronox*, was well aligned with government priorities and opened the opportunity to create the first mine-to-pigment company in the world.

5.3.1 Process description

5.3.2 Mining of heavy minerals

The method, by which the mineral sand deposit is mined varies, depending on the location and type of the ore body. There are three general methods by which this is done. The first of these methods is by means of slurring the ore, called "hydraulic



Figure 35: Monitor mining of a heavy mineral deposit



monitoring". In this process, the sand from the mine face is washed off by using a high pressure water gun – water pressure range in the order of 25 kPa. As this slurry flows off, it is directed to pumping stations, from where it is pumped to the primary processing plant, or primary wet plant (PWP). This process is generally used where the minerals occur in a sand form and no size reduction is needed. This form of mining is applied at *Tronox KZN Sands* and is illustrated in [Figure 35](#).

The second method of mining a heavy mineral deposit is by mechanical means. This is usually done where the deposit comprises both sand and rocks. The deposit is mined by means of front end loaders and diggers, which then load trucks or conveyor belts that transport the ore to the primary processing plant. This is a dry mining method and in this method, some size reduction may also be necessary. [Figure 36](#) shows the mechanical mining of heavy minerals as practiced at *Tronox Namakwa Sands*.



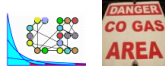
Figure 36: Mechanical mining of a heavy mineral deposit

The third method generally used to mine a heavy mineral deposit, is by dredging. In this method, the spiral plant is located inside the mining pit. The pit is then flooded with water, and as this is done, the dredge is raised in this artificial dam. The material is then pumped to the dredge, with the concentrate from the spirals being pumped to the primary processing plant. A prerequisite for this mining method is an ore body with low slimes content. Slimes are material which is less than 75 microns in size and remain in suspension in the water. *Richards Bay Minerals* makes use of this method of mining and [Figure 37](#) shows a picture of dredge mining of heavy minerals.



Figure 37: Dredge mining of a heavy mineral deposit

Often the terms heavy minerals mining and mineral sands mining are used interchangeably. This practice may be somewhat misleading seeing that although heavy minerals in many ore bodies worldwide are mined in sand deposits along coastlines (as is the case with the three major ore bodies in South Africa), some of these deposits are found in a solid rock form. The *Eramet Titan and Iron* smelter in Tyssedal, south west Norway, smelts ilmenite from a hard rock



Norwegian ilmenite mine. The Gravelotte mine area near Palaborwa in the Limpopo province of South Africa also produces low volumes of hard rock ilmenite.

5.3.3 Downstream beneficiation at *Tronox KZN Sands*

- **Primary concentration plant at the mine**

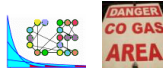
Once mined and screened, the run-of-mine (ROM) usually consists of about 5% heavy minerals, thus the first stage of beneficiation involves upgrading the heavy mineral content to about 90%. This is primarily done by means of spiral concentrators. It separates minerals primarily on the basis of density. Once this is done, magnetite is also removed from the ore by means of low intensity magnetic separators (LIMS). This magnetite needs to be removed since this adversely affects the subsequent magnetic separation processes. The HMC which emerges from the process, usually consists of about 90% valuable heavy minerals. The main constituents of this concentrate have already been mentioned: ilmenite, zircon, rutile, and in some cases, Leucoxene³.

Another important aspect of the PWP at *Tronox KZN Sands* is that most of the water needed by the process is recycled by means of high rate thickeners which remove any slimes from the water before returning it to the process. These slimes are then pumped to a residue dam for drying. A different approach occurs during dredging, where if the slimes content is low, the slimes are pumped back to the pond.

- **Secondary beneficiation**

Once the HMC is obtained there are a number of methods which exist to separate out the valuable minerals. Although the approach may vary between plants, the overall methodology remains the same. The first mineral usually removed is the ilmenite. This is done, by taking utilising its magnetic properties. The rutile and zircon rich stream which remains from the above process is then beneficiated by means of a series of gravity separation techniques (generally spirals and shaking tables) to remove lighter minerals. The zircon is then separated from the rutile by means of a series of electrostatic separators. These take advantage of the dissimilar conductivities between rutile and zircon, where rutile is a conductor and zircon is a non-conductor. Other gangue minerals such as monazite and leucoxene are also removed in this step by means of high strength magnetic separators. On some plants, depending on the presence of metallic staining on the particles, the rutile and zircon are leached, usually with sulphuric acid, in order to dissolve the coatings.

³ Leucoxene is a transformed ilmenite. Depending on its stage of transformation, its TiO₂ content may vary significantly, and it also brings in many impurities such as silica. The leucoxene quantity depends largely on the age of the mine.



- **Ilmenite treatment**

The ilmenite which is obtained from the secondary beneficiation process is usually utilised in one of three ways. The first is to smelt the ilmenite. This method is used predominantly in South African mines where the cost of smelting was relatively cheap due to historic low electricity tariffs. The process involves charging a smelter with ilmenite and a reductant (usually anthracite). This material is then electrically heated by means of an arc to temperatures in the range of 1700°C. Once the reaction occurs, the material in the smelter stratifies into two layers. These are the slag (top) and metal (bottom) layers. The slag is composed of a titanium dioxide content exceeding 85%, while the metal tapped out is essentially low manganese pig iron. This iron is cast into ingots and sold primarily for casting applications. The titania slag is cooled, crushed and then screened before it is sold to the titanium pigment market.

Other approaches to treating the ilmenite include firstly the manufacturing of synthetic rutile from the ilmenite. This is a popular process for ore bodies which are not suitable for smelting or where electricity costs are high. The TiO_2 content of the synthetic rutile is usually higher than that of titania slag. The final route for processing ilmenite, provided it is of good quality, is to use it directly to manufacture of titanium pigment by processing it in the chloride or sulphate processes by which both titania slag and synthetic rutile are also processed. Figure 38 shows a high level overview of the mining and beneficiation processes at *Tronox KZN Sands*. Note that the Slag Processing Plant is not shown in the high level overview.

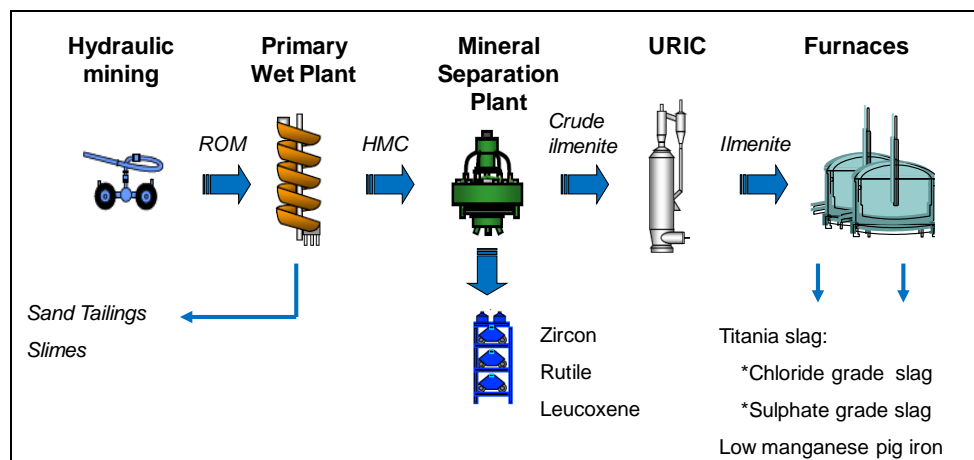


Figure 38: High level overview of the processes at *Tronox KZN Sands*.

Figure 39 shows an aerial view of the PWP at *Tronox KZN Sands* while Figure 40 shows a truck being loaded with HMC at the PWP before it is transported to the CPC 25km away. Figure 41 shows the aerial view taken from the north of the CPC with the town of Empangeni in the distance.

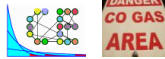


Figure 39: PWP at Tronox KZN Sands



Figure 40: HMC truck being loaded at the
PWP



Figure 41: Northerly view the CPC outside Empangeni, South Africa.

The MSP is shown in [Figure 42](#). The MSP is where secondary beneficiation of HMC takes place to separate the HMC into ilmenite, rutile and zircon. [Figure 43](#) shows pictures tapping operations at the CPC with a slag tap on the left and a metal tap on the right. A picture of a slag “button”, weighing approximately 18 ton, is shown in [Figure 44](#). These are cooled and later crushed and screened into the end products of chloride grade and sulphate grade titania slag.



Figure 42: Tronox KZN Sands MSP at the CPC



Figure 43: Tapping operations of titania slag (left) and a metal tap on the right

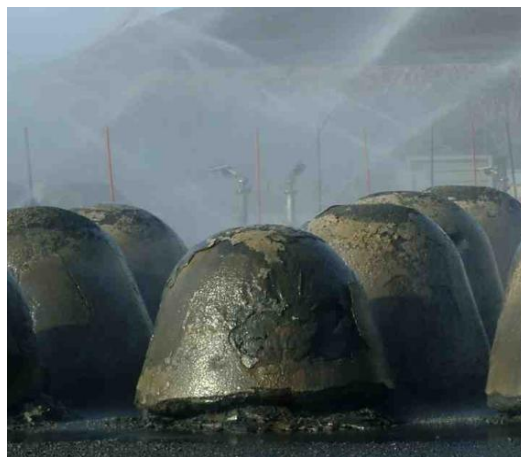
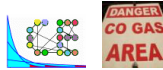


Figure 44: Secondary cooling of titania slag "buttons"



5.3.4 The furnace off-gas system

It was mentioned in section 5.2 and depicted Figure 33 that the ilmenite smelting process not only produces titania slag and iron, but a by-product of the reaction is large volumes of CO gas. Some properties of CO gas will be discussed in the next section. Its predominant properties include its highly poisonous nature and its flammability, especially at the high temperatures at which it is emitted from the smelting reaction. This necessitates careful management of the hot CO gas. An off-gas duct system with suction fans extracts the off-gas from the furnace freeboard (the space between the slag crust and furnace roof) and out into the disintegrator system. Initially the off-gas also contains fine ilmenite particles and furnace dust from the smelting reaction and requires cleaning.

Off-gas cleaning is done by ejector-venturi scrubbers which remove the residual dust from the furnace gas. The disintegrator scrubber plant minimises the explosion hazard in the presence of the CO gas. In contrast to this, a conventional scrubber, which is used in some furnace operations, does not reduce this hazard as much, due to a high risk of air ingress, which, in turn, can result in explosion damage. Figure 45 depicts a typical disintegrator scrubber.

Disintegrated scrubber systems for furnace off-gas systems are successfully operated at *Tronox KZN Sands*, *Tronox Namakwa Sands*, *Richards Bay Minerals*, *SACC (South African Calcium Carbide)* and *Samancor Ferrochrome* in Middelburg.

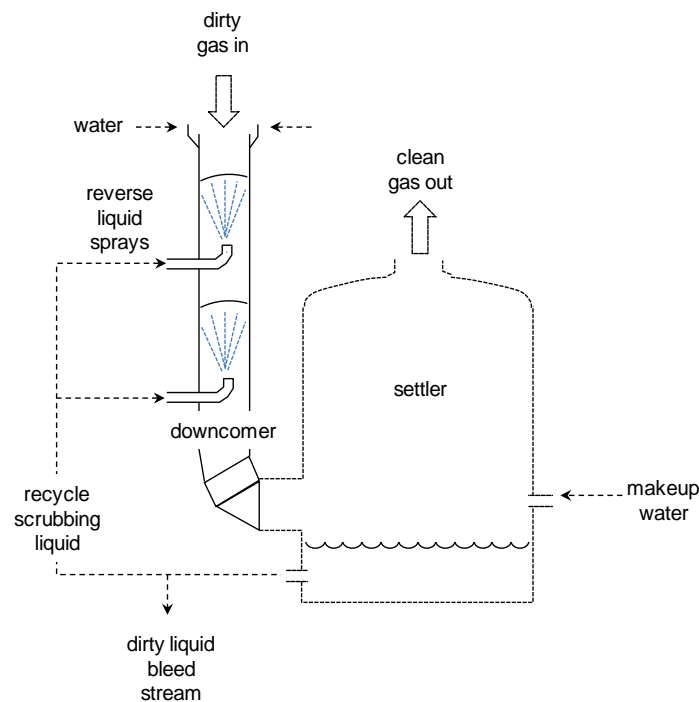


Figure 45: A disintegrator scrubber



The clean gas from the disintegrator scrubber can be distributed to the stack at the top of each furnace where the gas is flared, or to the distribution plant where the gas is compressed and distributed to the different plants where it is used to dry and heat feed material or pre-heat equipment.

5.4 Short overview of two common industrial gases

This research project entailed finding an improved way of managing CO and methane gas at *Tronox KZN Sands*. CO and methane gas are only two of a number of gases that are commonly applied in industrial applications. This section aims to provide some background to CO and CO₂ gas, another common industrial gas. These two gases are very similar on a molecular level, but significantly different in their properties and applications.

Next, some of the most important properties and applications of CO and CO₂ gas will be discussed. CO₂ gas will be dealt with first. The intention is to provide the reader with a better understanding of CO gas, which is an important entity in this project, and to allow the reader to contrast CO gas with the better known CO₂ gas.

5.4.1 Carbon dioxide

In recent times, the first thought with regards to CO₂ gas, at least in the popular media, is the relationship of CO₂ gas with climate change and global warming through increased human induced CO₂ emissions. It is not in the scope of this project to discuss this aspect of CO₂ gas. It is, though, acknowledged as a contemporary and highly debated topic and, in fact, has some bearing on the CO gas problem. It is therefore briefly discussed. The recent pressure on companies and individuals to reduce their carbon footprint has also forced the *Tronox KZN Sands* to embark on projects to achieve this. Thus, with reference to the CO gas problem, apart from it being a cost saving project, it is also in *Tronox's* interest to implement a project of this type that will reduce the company's carbon footprint. As mentioned in section 5.2, the CO gas project has the potential to reduce the direct CO₂ emissions by over 7 400 tons (5%) per year. The motivation to implement "green projects" has thus become an additional important driver of this project.

- **Properties and applications**

CO₂ is colourless, stable and non-combustible. Compared to N₂, the heavier CO₂ is not as chemically inactive and accordingly finds only limited use as an inerting gas in petrochemical applications. CO₂ is an acid which reacts with bases such as ammonia or carbanions. It can be used for reaction or co-polymerisation with epoxides. Overall, CO₂ serves as a building block for only a few chemical reactions, but covers a wide variety of applications. The gas is used as:



- A component for chemical syntheses such as the production of urea, organic acids (e.g., hydroxybenzoic acids such as salicylic acid) and cyclic and polymeric carbonates
- Feed for CO production, especially if H₂ is available
- A reagent for neutralisation of aqueous fluids with too high pH – often replacing sulphuric acid in this function
- A solvent (in the supercritical state), as in the production of fluorinated polymers
- A foaming agent in production of voluminous polymers
- An inert gas mainly for blanketing, as it is heavier than air
- A fire-extinguishing agent
- A gassing agent in soft drinks and sparkling wine
- As a solid, i.e., “dry ice” where CO₂ is used as a coolant and a surface cleaning agent, especially in dry-ice blasting applications, where accelerated CO₂ pellets are used to maintain heat exchangers or pipelines.

CO₂ is a greenhouse gas (GHG). Human activities account for generation of more than 20 billion ton per year of CO₂ (Schreiner & Reinhardt, 2008). As mentioned above, new efforts are made not only to minimise CO₂ output but also to discover new potential uses (Walther, 2007). Such novel applications have been found in the resource opening such as enhanced oil recovery (EOR), in which CO₂ reduces the viscosity of in-situ oil deposits for facilitated recovery. However, in petrochemistry, bulk CO₂ uses are not easy to find even when the various reduction options (including the Boudouard equilibrium and the Calcor process) are considered. (Schreiner & Reinhardt, 2008). Efforts to exploit CO₂ as a supercritical solvent have not yet led to a major breakthrough of commodity chemicals production. Another interesting approach for bulk use of this gas is the dehydrogenation of saturated hydrocarbons where CO₂ could have potential as a supporting oxidant (Bhasin, 2001). Such an application does not, however, appear to have been commercialised.

- **Sources of CO₂ gas suitable for capturing and storage**

Schreiner & Reinhardt (2008) states that with a CO₂ content of about 380 parts per million by volume (ppmv) in air, generation of this gas via air separation units is not economic, especially since high-volume gas mixtures with much higher CO₂ concentrations are available. It is accordingly obtained from natural sources such as CO₂-rich natural gases but also from a variety of production processes, where it is generally a by-product. Such processes include fermentation (mainly bio-ethanol production), lime burning and reactions of carbonates with strong acids. Petrochemical CO₂ sources of industrial importance, besides syngas production units, are ethylene oxide plants. Here the comparatively poor selectivity of the catalytic oxidation step is responsible for the high CO₂ generation rate. Normally, CO₂ is delivered in cylinders but also in



tanks as liquefied gas with purity up to 99.999 vol%. Special high-purity grades and mixtures with other gases are also available. In many cases the emission volumes are impractically high for capturing economically, for example at the *Tronox KZN Sands* furnaces, where the emitted CO gas burns in atmospheric air to form large volumes of CO₂ and water vapour. In such cases the CO₂ is simply lost to the atmosphere and thereby impacting negatively on the environment by contributing to a green house effect. To lower this negative impact, the CO₂ emissions at *Tronox KZN Sands* needed to be reduced.

- **Humans and CO₂**

It must be appreciated that normally CO₂ cannot be detected by human senses. This gas is significantly heavier than air and human exposure at a concentration greater than 10% by volume can be fatal, therefore, strict compliance with safety regulations is mandatory in all applications.

5.4.2 Carbon monoxide

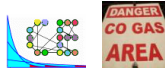
Carbon monoxide gas is molecularly very similar to CO₂ gas, but differs in structure in that it consists of a single carbon and oxygen atom covalently bound. This small difference gives it vastly different properties compared to CO₂.

The most commonly known source of CO gas is probably motor vehicle exhaust systems. However, CO gas has many industrial purposes. In many industrial petrochemical processes, CO gas is a key reactant for products, including:

- Phosgene (a valued industrial reagent and building block in synthesis of pharmaceuticals)
- Acetic acid via carbonylation⁴ of methanol
- Acetic anhydride via carbonylation of methyl acetate or dimethyl ether
- Oxidative carbonylation of methanol to dimethyl carbonate
- Meat colouring, to keep it looking fresh longer
- In medicine as a biological regulator that acts as an anti-inflammatory

CO gas also plays a role in complementing well-established technologies. For example, methyl methacrylate monomer (used in the manufacturing of PMMA plastics – a substitute for PVC and a material used as cement in total hip replacements) is produced mainly from hydrogen cyanide and acetone in the presence of CO gas. A novel two-stage synthesis route involving ethylene, methanol and formaldehyde as starting materials, along with CO, is becoming competitive (Schreiner & Reinhardt, 2008). One production plant of the latter type has been constructed in Singapore in 2008.

⁴ Reactions that introduce CO gas to organic or inorganic substrates.



- **Sources of CO gas suitable for capturing and storage**

The common way to obtain CO is by cryogenic distillation (essentially distillation at very low temperatures) of syngas. CO gas from this source is available in different purities:

- For bulk supply via pipeline or from a nearby production plant, up to 99.99 volume % CO gas. This is adequate for most chemical syntheses (typical contaminants are H₂ and hydrocarbons along with air gases including O₂, N₂ and argon as well as water).
- Qualities up to "five nines" (quality grade 5.0, standing for a gas purity of 99.999%) for testing and calibration.

On a laboratory scale, CO gas is produced by the dehydration of formic acid, e.g. with sulphuric acid, or by heating a mixture of powdered zinc and calcium carbonate.

Petrochemical and other processing can consume relatively large quantities of CO gas that is generally produced by onsite plants (for example the *Tronox* ilmenite smelters in South Africa). Some chemical complexes have pipeline networks to supply the gas to various processes. Production capacities range up to and beyond 30 000 m³/h (Schreiner & Reinhardt, 2008). The volumes at the *Tronox KZN Sands* smelters are in the order of 9 000 m³/h. For minor demands, such as in laboratories, CO is normally delivered as a compressed gas.

- **Humans and CO gas**

CO is a highly toxic, combustible, colourless and odourless gas (Penney, 2000). Thus, stringent safety requirements govern its shipping, storage and handling. In South Africa, the SANS 329⁵ standard is one of the important standards that govern the handling of CO gas. The latter regulation also impacted on this research project, especially with regards to the implementation of the project.

5.5 Examples of the usage of CO gas as a furnace off-gas in industry

Although the CO gas at *Tronox KZN Sands* has to date not been utilised successfully to tap into its benefits, there are many industry examples of how CO gas as a furnace off-gas is indeed used with success. These examples serve as further motivators to focus on the implementation of this project.

⁵ South African National Standard 329 is prescribed by the South African Gas Pipeline Association (SAPGA)



5.5.1 Pre-heating of scrap in steel making

Electric arc furnaces (EAFs) are used in the steel industry to convert pig iron, scrap and other iron units to steel. CO gas forms as off-gas in the EAF when oxygen is introduced to burn carbon. Carbon is burned to decarburise the steel (to lower the carbon content of the steel), or to generate gas in the slag to foam the slag (a practice which improves heat transfer from the arc to the metal bath).

In this industry, EAF off-gas is used in the *pre-heating*, especially of scrap material. The scrap is loaded into large buckets called “baskets”. After loading, the basket may pass to a scrap pre-heater, which uses hot EAF off-gas to heat the scrap and recover energy to increase plant overall efficiency. The scrap basket is then taken to the melt shop, the roof is swung off the furnace, and the furnace is charged with the pre-heated scrap from the basket. Another operation in the steel industry where EAF off-gas is utilised is *continuous charging*. This process involves the discharging of pre-heated scrap from a conveyor belt into the furnace, or, charging the scrap from a shaft set above the furnace, with off-gases directed through the shaft (Turkdogan, 1996).

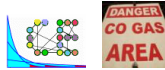
In all the processes within the steel making industry mentioned above, the CO gas will be combusted in air, resulting in the emission of CO₂. The steel making industry is looking at alternatives that will reduce CO₂ emissions. Increased international emphasis on the relationship between CO₂ emissions and climate change may also lead to increase regulatory pressure. This pressure, combined with economic pressures has already stimulated research into the improved usage and control of EAF off-gas. One of the most interesting developments for the author is evaluating the feasibility of hydrogen-based steelmaking, which generates H₂O rather than CO and CO₂.

EAF off-gas system research and a more complete understanding of the related combustion chemistry process may play a part in decreasing the environmental effects of CO₂ gas emissions from steelmaking.

5.5.2 Model Predictive Control

The primary purpose of an off-gas system in steelmaking is to extract the gases emitted by the EAF, to combust hazardous gases such as CO and to filter out the dust. The off-gas system also influences the steelmaking process by influencing the furnace pressure through the rate of gas extraction. It therefore determines the degree of air entrainment, and hence the process energy requirement, as well as nitrogen pick-up in the steel (Jones *et al.*, 1998).

In order to investigate the use of the off-gas variable set in an appropriate control structure, a suitable process model of the EAF and off-gas system is necessary. (Bekker *et al.*, 2000) developed such a model and used model predictive control (MPC) to investigate the automation



of manually controlled variables such as the forced-draught fan power and the air-entrainment slip-gap width for the EAF off-gas process. These two variables can be manipulated and are used to control the relative pressure inside the furnace and the temperature and composition of the gas that exits the cooling duct. Bekker *et al.* (2000) found that the EAF off-gas system can provide valuable variables for feedback control, which, if well manipulated, can improve furnace efficiency and contribute to safety in the workplace. Wojsznis *et al.* (2007) looked at a more generic application of MOO MPC and found it successful on control of a distillation column and the standard Shell heavy oil fractionators problem.

5.5.3 Co-generation

Co-generation is a term loosely used to describe a variety of applications. Typical uses include the generation of electricity in a power station and then using the thermal heat for heating of buildings (Schubert & Gottschling, 2011). It is also widely used where various thermodynamic processes are combined to become more energy efficient. In the latter case co-generation implies the use of CO- and H₂-rich off-gases from a closed EAF to produce electricity by internal combustion engines.

In a good, sealed, electric, reduction furnace no oxygen enters the furnace and the furnace off-gas system from the surrounding environment. The off-gases are thus not “burnt” and therefore contain CO and H₂. The volume and composition of the gas depends on variables such as the feed material, pre-treatment methods (e.g., pre-reduced feed generates less gas than other feed materials), the design of the furnace, furnace controls and the metallurgical condition of the process. The solid matter content in the gas off-take of a furnace varies with the design and dimensions of the furnace, electrode arrangement, material feed details, and operational conditions. In closed AC and DC furnaces wet scrubber systems (disintegrator systems, see Figure 45) are mostly used to cool and clean the gas extracted from the furnace.

Currently some of the CO-rich off-gas is used for sintering and pre-heating as described in section 5.5.1, but in many cases, the majority of it is flared to convert the poisonous CO gas to CO₂. Schubert & Gottschling (2011) state that the percentage of the total gas volume produced in various South African smelters that is currently being used for heating purposes, is negligible compared to the available potential of energy recovery, i.e. electricity generation.

It has been indicated by GE Jenbacher, one of the suppliers of generator equipment designed to use furnace off-gas as fuel source, that approximately 30% of the (electrical) energy supplied to the furnace can be recovered. This is possible if the total furnace gas volume produced is used as a fuel source for a gas engine generator set. In addition, the thermal heat energy of the gas engine’s exhaust gas, at approximately 500°C, is also available for recovery, to be used in steam generation, material drying, pre-heating, etc. (Schubert & Gottschling, 2011).



Currently in South Africa, the potential use of the off-gas as a primary fuel source in a power-generation facility to reduce dependency on the national electric grid is becoming more and more important. Typical producers of CO gas in South Africa at levels high enough for electricity generation are *Samancor Ferrochrome* in Middelberg, the ilmenite smelters of *Tronox Mineral Sands* (in Saldanha and Empangeni) and *Richards Bay Minerals* (Richards Bay) (Schubert & Gottschling, 2011).

Recent developments have led operators of some of these furnaces to revise their strategy and to consider utilising these gases as a source of fuel for producing electric energy, thus becoming more energy efficient by operating a so-called co-generation installation.

5.5.4 Risks and challenges

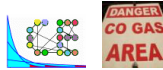
As mentioned already, CO gas is a highly toxic, combustible, colourless and odourless gas. Stringent safety requirements govern its shipping, storage and handling. At *Tronox KZN Sands* precautionary measures include zoning of certain areas as high hazardous areas which is demarcated with appropriate signage, the usage of specialised valves to safely distribute and discharge CO gas in pipelines, installation of fixed CO gas monitors, providing personnel working in CO gas areas with personal CO gas monitors and alarms built into the Digital Control System (DCS) whereby CO gas leaks can be detected by operators.

In the United States (US) and other countries where CO gas poisoning (often unintentional) may be a household risk due to certain fuel burning appliances, an effort is being made to educate even the public on the hazards and prevention measures of CO gas incidents. In the US, household scale detectors are available that will warn when carbon monoxide has reached a dangerous level. The Consumer Product Safety Commission and the Federal Emergency Management Agency in the State of Colorado have even recommended that each home have at least one CO gas detector outside bedrooms and another in the vicinity of fuel-burning appliances (www.asse.org, 2003, Professional Safety, 2003).

5.6 Summary: Chapter 5

The aim of this chapter was to place the research problem in context in terms of the real-world environment where it originated.

The heavy minerals industry is not as well known an industry as coal, diamond, gold, platinum and iron ore mining in South Africa, hence it was the author's intent to provide an overview of the industry. It was also necessary to focus on two sub-processes of this industry, namely ilmenite smelting and furnace off-gas management. The author is of the opinion that a



reasonable understanding of these fields will aid the general reader's appreciation of the practical challenges surrounding the project and the practical value of the research.

With the IE or OR reader in mind, the author could not find many examples in the literature of successful applications of simulation and metaheuristics, particularly the CEM, in the heavy minerals industry. This research could therefore potentially serve as a catalyst (as a case study) for further research in the application of these techniques in the heavy minerals industry, hence the importance to contextualise the research problem properly.

The problem context was narrowed down to the specific South African heavy minerals operation where the research problem arose, namely *Tronox KZN Sands*. A brief overview of the company was presented, including a discussion and diagrams of where in its processes the research problem fitted in.

As explained in previous sections, the research problem deals with the management of CO gas from the *Tronox KZN Sands* furnaces and was termed the "CO gas problem". In the light of this, an overview of CO and CO₂ gas was discussed with the intent of allowing the reader to grasp some of the practical aspects surrounding the problem. The discussion of CO and CO₂ gas included its main industrial sources, some key applications and its hazards to humans.

The chapter concluded with illustrations of other ways of how CO gas as off-gas from EAF operations are utilised for industrial benefit. These applications pertained mainly to EAF operations in South Africa.

The next chapter will introduce the third and final theme of the thesis which entails the problem formulation and solution development phase of the research project. Chapter 6 will address the first part of the final theme, namely the formulation of the CO gas problem.



6. Problem formulation

The research problem has been briefly introduced in the opening section of Chapter 1 of the thesis, and it was named, the “CO gas problem”. In the previous chapter the real-world context of the CO gas problem was discussed. Some of the most important characteristics of CO gas were also presented and then compared with the characteristics of the more commonly known CO₂. Industrial and laboratory scale sources of CO and CO₂ gas were discussed, as well as their applications and the associated risks to humans.

Chapter 6 introduces the third and final theme of the thesis, which involves the experimental work and solution development. This chapter will focus on how the research problem was formulated, or defined. In other words, the way in which the real-world problem was conceptualised and how practical aspects were incorporated into the problem formulation will be presented. Since it was *not a deterministic problem*, which would have allowed a mathematical formulation, the formulation in this case involved the development of a *concept model*, and eventually a *computer simulation model*.

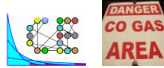
The problem formulation shows how the real-world environment and the academic environment intersect. Appropriately then, the chapter includes a clarification of the scope of the research project, indicating which real-world aspects were included and which excluded in the academic exercise. The chapter concludes with a discussion of the approach taken towards solving the real-world problem in the short to medium term, with the academic exercise as guiding input, compared with the longer term solution, which is inhibited by certain practical obstacles.

6.1 Conceptualising the research problem

The conceptualising of the research problem involves the “translation” of the real-world problem into a research problem. Remember that due to the stochastic nature of the research problem, a mathematical formulation of the problem was not possible, and a concept model needed to be developed. The concept model developed for the CO gas problem is shown in Figure 46. It depicts the CO gas problem as a *supply and demand problem* requiring the *dynamic prioritisation* of the available CO gas. In the subsections to follow the supply side, demand side and the dynamic prioritisation of CO gas will be discussed.



Figure 46: Basic concept model of the CO gas problem



6.1.1 Supply of CO gas

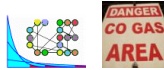
The supply side of the problem represents the two *Tronox KZN Sands* furnaces that produce CO gas as a by-product. The production process of the furnaces has been discussed in sections 5.2 and 5.3. The two furnaces operate independently of each other and produce variable levels of CO gas. The volumetric flow and stability of the CO gas production is mainly dependent on the feed rate, the quality of the feed material and the operational stability of the furnaces. Various operational incidents, such as foaming incidents, off-gas duct blockages, furnace centre piece changes or electrode “additionings”, can result in changes in feed rate and/or energy input that, in turn, affect the volume and stability of CO gas produced.

6.1.2 Demand for gas

The demand side of the problem represents different plants situated at the *Tronox KZN Sands* CPC where other intermediate processing of the mined heavy minerals occurs. The reader is referred to [Figure 38](#) where an overview of the different plants and processes at the business is shown. The plants which are of importance here are those plants at the CPC where drying and pre-heating of feed material, or pre-heating of certain equipment, takes place. These plants are shown in [Table 3](#) – the reason for the greyed out plants will be discussed further down:

Plant	Reason for gas requirement
Unroasted Ilmenite Circuit (URIC)	Drying of stockpiled crude ilmenite before separation into furnace quality ilmenite.
Slag Processing Plant (SPP)	Drying of crushed slag “buttons” from the secondary cooling yard before separating it into final products.
Reductant Plant (RPP)	Drying of anthracite from the reductant silos before feeding the furnaces.
Metal Treatment Plant (MTP)	Pre-heating of ladles in preparation for a metal tap.
Mineral Separation plant (MSP)	Drying zircon and rutile for effective magnetic and electrostatic separation.
Pre-heater 1	Pre-heating of ilmenite and anthracite before being fed into Furnace 1 to increase energy efficiency.
Pre-heater 2	Pre-heating of ilmenite and anthracite before being fed into Furnace 2 to increase energy efficiency.

Table 3: Plants where CO or methane gas is consumed at the CPC



The drying and pre-heating are currently being done predominantly by burning methane gas at these plants. The burners at these plants are, however, equipped to burn both methane gas and CO gas. Piping, valves, instrumentation and a control system exist to distribute and control either the CO gas from the furnaces or methane gas from an underground gas line to the burners. Each plant has a different requirement for gas. The volumetric gas requirement at the burners is a function of the type and rate of material feed, as well as the type of gas burned at the burner. The latter is due to the different calorific value of different gases.

The graphs (Figure 47) show the relative requirement for gas at the different plants (the values shown are averaged). The gas requirement of a plant also determines its priority. It must be noted that some of the plants shown in Table 3 were not considered as part of this project because of specific operational reasons or management decisions – these are greyed out in Table 3: the pre-heaters have not been operated sustainably since 2001 and have been taken out of the business strategy for the foreseeable future. At the time of modelling the problem no CO gas distribution infrastructure or data on actual gas consumption existed at the MTP. Lastly, a management decision had been taken, based on unacceptable process risks, that methane gas would not be replaced with CO gas in the MSP. At the MTP there is a greater likelihood that the burners could be converted to burn CO gas. In that case, the scope of the project can be enlarged to include the MTP pipelines and burners in the simulation model. Table 4 shows the priorities of the three plants included in the project and a picture of the burner at the URIC is shown in Figure 48.

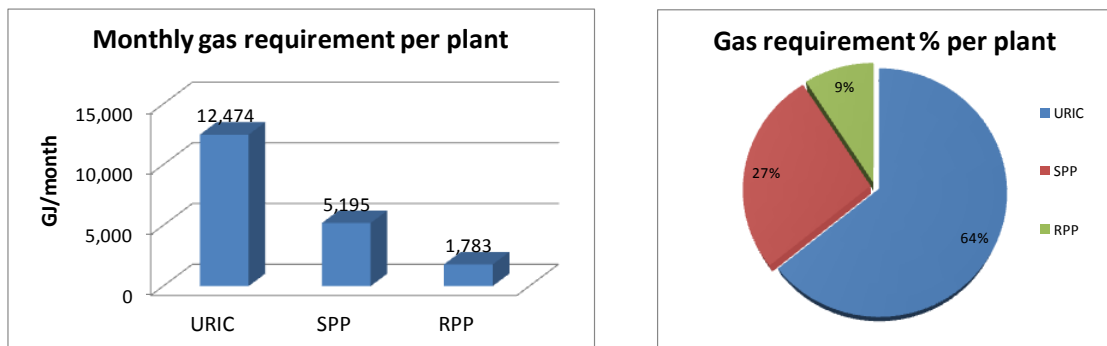


Figure 47: Monthly gas energy required per plant

Priority	Plant
1	Unroasted Ilmenite Circuit (URIC)
2	Slag Processing Plant (SPP)
3	Reductant Plant (RPP)

Table 4: List of gas consumers included in the project in order of priority

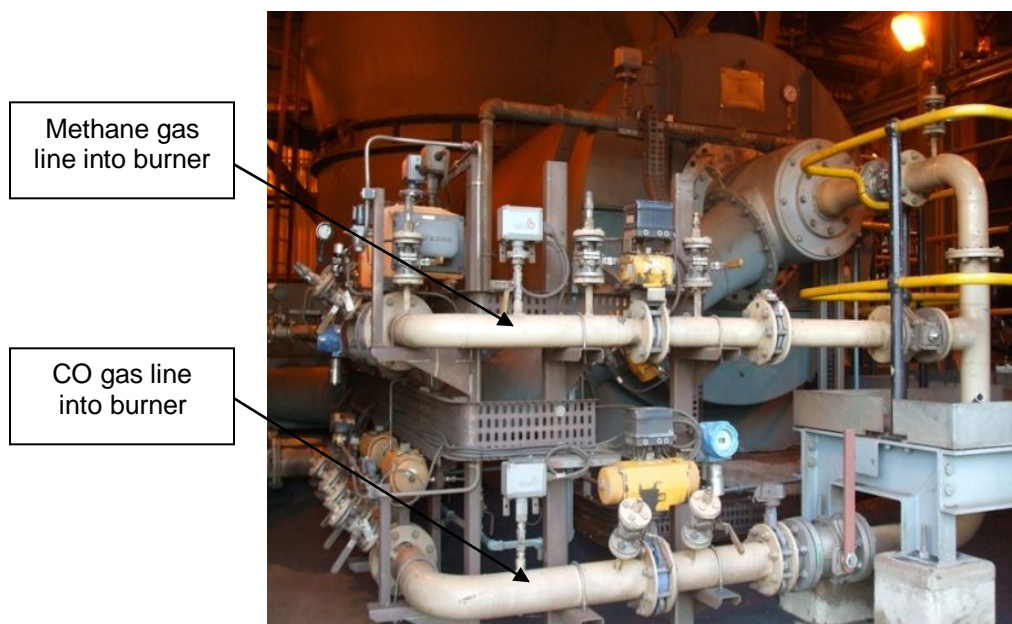


Figure 48: CO and methane gas burner at the URIC

Methane gas is currently purchased in bulk from an external supplier to meet the demand for gas. The methane is tapped from an underground gas pipeline which ensures constant availability. Ensuring constant availability of methane gas is costly. The average monthly cost of methane gas consumption exceeded R 800 000 in 2010 and rose to over R 1.5 million per month in 2011. With the unit price of methane gas continually rising, pressure has mounted to start making use of the freely available CO gas. Figure 49 indicates the steep increase in the unit cost of methane gas between 2006 and 2012, while the total actual monthly expenditure on methane gas in the first six months of 2009, 2010 and 2011 is shown in Figure 50.

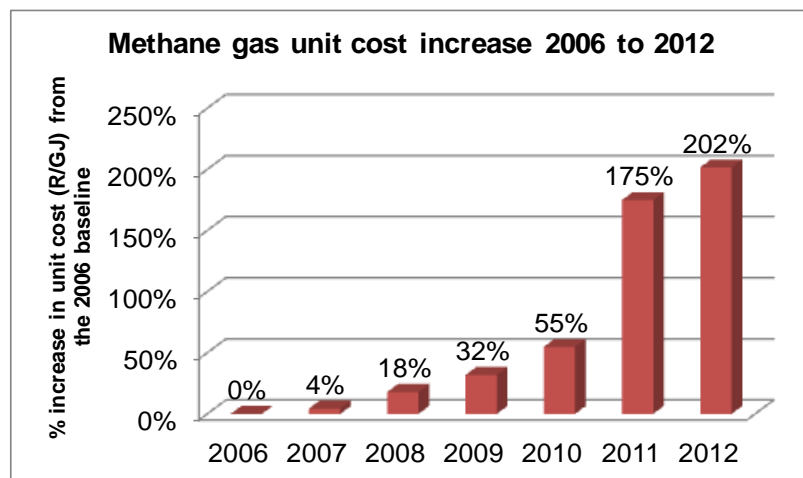


Figure 49: Increase in methane gas unit cost since 2006

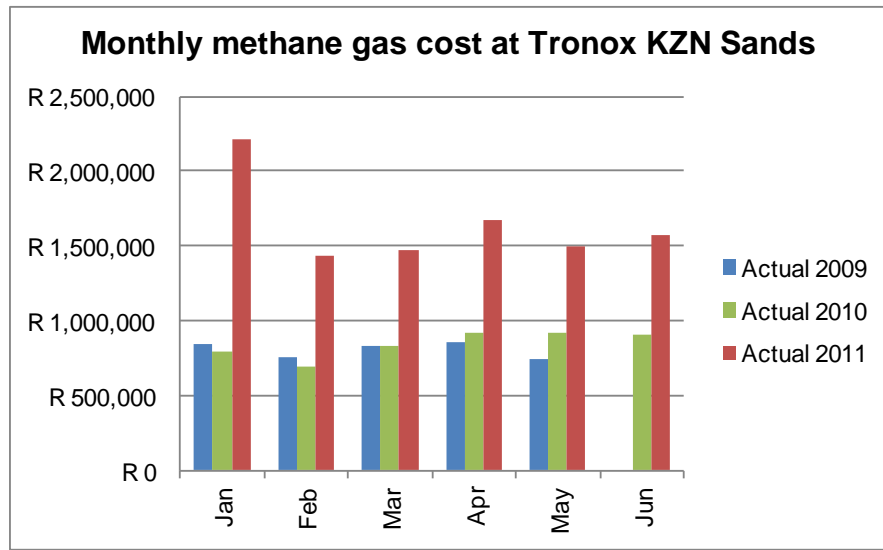


Figure 50: Actual methane gas cost January to June in 2009, 2010 and 2011

Although the CPC was built with the intention to distribute CO gas from the furnaces to the various burners, since 2001 this has not yet been achieved sustainably. A quotation from an official company Post Implementation Review (PIR) for the upgrade of the URIC, dated 19 March 2012, confirms the latter unfortunate reality (Rajkumar, 2012). This reads:

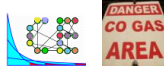
“The benefits that were originally proposed were not achieved due to unstable furnace operation and only one furnace being operational for majority of the past 2 years, as well as the poor Howden plant availability.”

Due to the significant rise in unit cost of methane gas, the project started receiving the attention it deserved from 2011, to the extent that implementation of the project commenced in the second semester of 2011. The author, as industrial engineer and BI Manager on site, is involved in the planning and capital approval phase of the project. [Appendix E](#) shows the latest project plan for implementation of the project.

6.1.3 Matching the supply of CO gas with the demand for gas

6.1.3.1 Reasons for unstable CO gas supply

The key reason why the CO gas has not been utilised at the burners for so long was the unstable availability of CO gas. The instability of the CO gas supply is a result of process instabilities at the furnaces. Ilmenite smelting is known to be a pyrometallurgical process which constantly poses challenges to process metallurgists. It should be mentioned that from the time the author joined



Tronox KZN Sands in 2007, until 2012, operational efficiency and stability at the furnaces have improved significantly. This improvement is due to increased process knowledge and the implementation of various continuous improvement projects.

This instability in CO gas supply presumably caused the process metallurgists to lose confidence in the possibility of replacing the methane gas with CO gas at the burners. Due to the inconstant gas availability too many operational challenges were foreseen. Should the CO gas be distributed to the different burners, in order to maintain a steady flow of gas at the burners, the burners would have to be switched over from CO gas to methane gas every time an undersupply of CO gas occurred. Existing technology at the CPC does not allow for fully automated control of these switchovers. Before and after every switchover, a section of the gas line needs to be purged with nitrogen. This is to prevent flashbacks into the gas lines. Every switchover results in a loss of production time and requires man hours from the operator manning the switchover and purging. Despite the significant cost savings potential of this project, the view prevailed that it was still more reliable and manageable, despite the higher cost, to continue purchasing the methane gas and distributing only that to the burners. This view prevailed early 2011 when the methane gas unit cost almost doubled, causing management to request new focus on this project (Figure 49). Two further contributing factors added to the difficulty of the inconsistent levels of CO gas.

First, CO gas is a highly poisonous gas, whilst also being colourless and odourless (refer to section 5.5.4). Implementing the project would imply that many new areas at the CPC would be exposed to CO gas. The hazardous nature of CO gas has serious health and safety risks and additional costs for *Tronox KZN Sands* associated with its use, and thereby led to further hesitation from management to prioritise the project. Figure 51 is a picture taken at the CPC of a warning sign in a CO gas area.



Figure 51: CO gas poses serious occupational health and safety risks.

Secondly, *Tronox KZN Sands* does not have a pressure vessel (or gas holder) in which buffer capacity of CO gas can be built up in order to smooth out fluctuations in the supply of CO gas from the furnaces. Constructing and integrating such a pressure vessel would require significant capital input and would be subject to strict safety specifications. Although the estimated payback from such a step always seemed in favour of continuing with it, for various reasons it never aligned convincingly with the business strategy over the period of 2006 to 2011. The world economic crisis of 2008/9 had significant negative effects on business conditions, especially on sales in 2009 and early 2010 and led to cutbacks on capital expenditure. Additionally, delays in the approval of the new Fairbreeze mine that is supposed to replace the



existing Hillendale mine (which will reach end of life by end 2013) brought more urgent immediate priorities and even cast a shadow of doubt over the continuation of the business in 2010. In that context, plans to construct a multi million rand pressure vessel were simply not a priority.

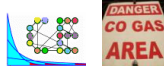
Although it is not within the scope of this project, the author is of the opinion that a re-evaluation of the potential construction of a pressure vessel to fully control CO gas availability should be re-motivated. Considering the approval of the new Fairbreeze mine by top management in February 2011, the fundamental shifts in the supply of and demand for heavy minerals products (which affected market prices positively) and the confidence displayed by *Tronox* in their acquisition of the business, *Tronox KZN Sands* now has a 15 to 20 year time span ahead of it. This gives enough time to plan and build a pressure vessel for CO gas utilisation. The ever increasing unit cost of methane gas adds further impetus to the business case for the construction of the pressure vessel.

6.1.3.2 Compression and distribution of CO gas

The compression and distribution of CO gas is supposed to be done by the *Howden* plant. This plant is situated next to the furnaces and consists of a series of high and low pressure compressors. The *Howden* plant receives CO gas from the gas cleaning plant that was described in section 5.3.4. Due to the fact that the CO gas has never been distributed sustainably to the burners to date, the *Howden* plant fell into a state of unavailability. The distribution pipelines and valves also deteriorated over time. This further lessened the likelihood of utilising the CO gas further. However, as a result of the new focus on the project since end 2010, refurbishments have been started at the *Howden* plant. Figure 52 shows a picture of the *Howden* plant.



Figure 52: *Howden* gas compression and distribution plant



One of the low pressure compressors at the *Howden* plant was repaired in the second semester of 2011. Subsequent to this, the CO gas lines to the RPP were pressure tested and inspected for corrosion in order to be certified under the SANS 329 standard for usage of gases. Thereafter the burners and valves at the RPP were also upgraded to comply with the SANS 329 standard. These were important first steps towards the actual implementation of this long awaited project at *Tronox KZN Sands*. The first CO gas was supplied to the RPP towards the end of 2011.

6.1.3.3 Dynamic prioritisation of CO gas

In the light of the abovementioned practical challenges, it was clear that, in order to replace the expensive methane gas with freely available CO gas, a constant supply of CO gas, which required stable furnace production or a pressure vessel, or effective dynamic prioritisation of the available CO gas was required. Although the dynamic prioritisation is not intuitively seen as the optimum solution, compared with having a pressure vessel that provides sufficient buffer capacity of CO gas, it must be emphasised that dynamic prioritisation of CO gas can nevertheless effect a significant saving in methane gas cost. Dynamic prioritisation implies that lower priority users will be denied CO gas at times of low supply and switched to methane gas. The aim is to minimise the consumption of methane gas in the process. Without a constant CO gas supply and/or effective dynamic prioritisation, the CO gas will never be utilised and the significant benefits thereof will be lost.

The dynamic prioritisation is dictated by the digital control system (DCS) which also controls the *Howden* plant operation. Figure 53 depicts how the basic concept model of Figure 46 has been extended to include the CO gas suppliers, the consumers and the *Howden* plant, while a detailed concept model of the problem is shown in Figure 54. The final formulation of the problem as a supply and demand problem was therefore done through the detailed concept model in Figure 54.

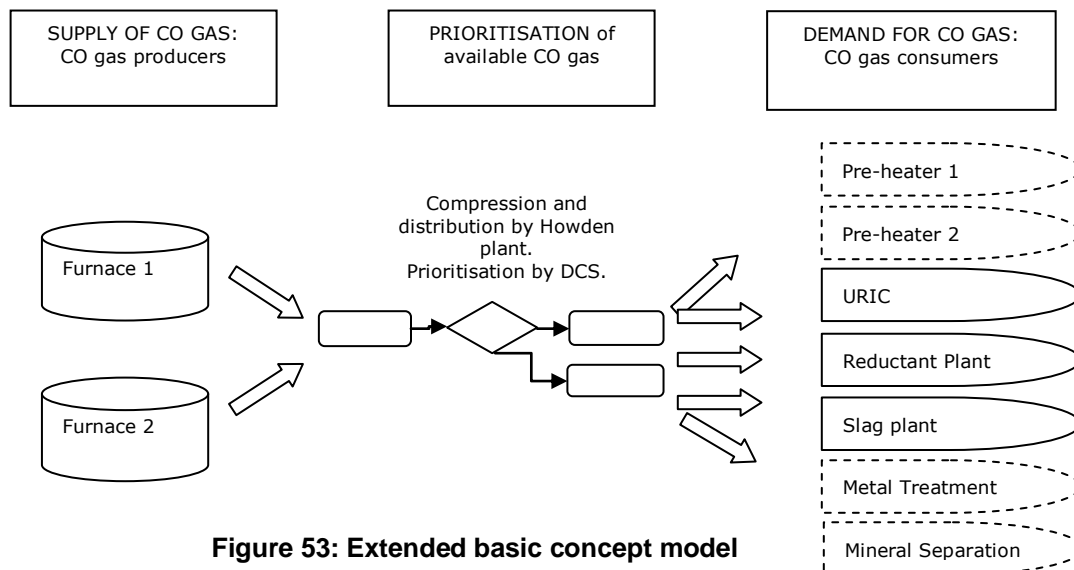
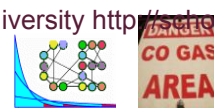


Figure 53: Extended basic concept model



6. Problem formulation

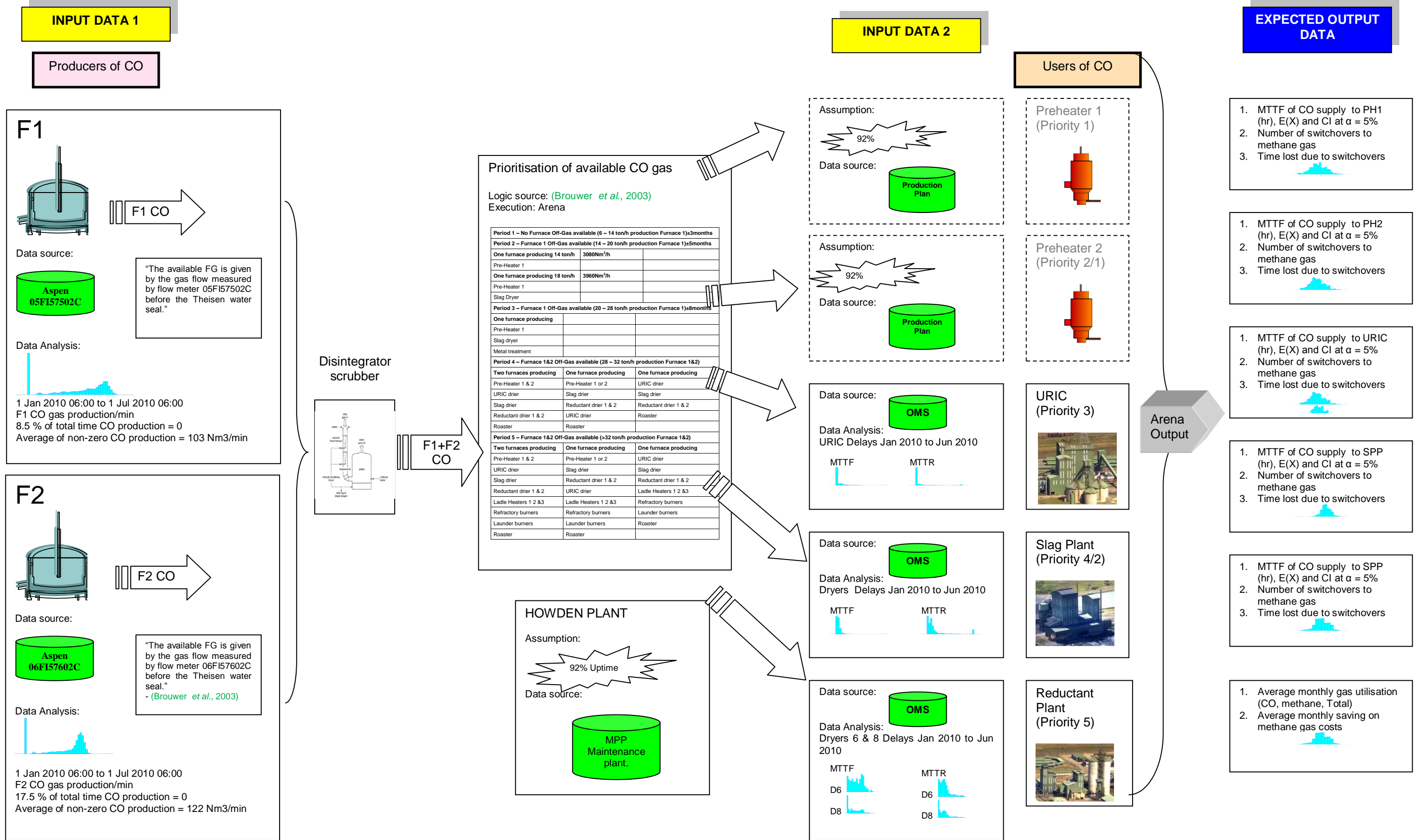
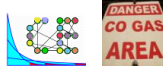


Figure 54: Detailed concept model of the CO gas problem at Tronox KZN Sands



The demand is determined by the combination of gas users assigned by the simulation analyst to receive CO gas, as well as the mean time to failure (MTTF) and material feed rate of each of those users. The detailed concept model in [Figure 54](#) was used in the development of the computer simulation model. The detailed concept model also indicates the main system parameters (input data) used in the simulation study, the sources of this data and the outputs (output data) that were expected from the simulation model. [Appendix B](#) and [Appendix C](#) show the concept model translated into a *Microsoft® Visio* flow diagram and *Arena®* simulation model respectively.

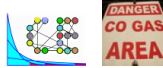
Note that the extended basic concept model in [Figure 53](#) shows the MTP and MSP in dotted lines as part of the concept model, while the detailed concept model excludes both. The reasons for excluding these plants have been discussed in section 6.1.2. It is possible to connect the MTP and MSP physically to the CO gas distribution network, as well as to include them in the simulation model. The pre-heaters have been included in the detailed concept model and simulation model, but they were not “switched on” in the simulation model.

6.2 Scope clarification

The original aim of the simulation was to forecast the availability of CO gas at each of the burners. More specifically, the simulation was aimed at forecasting the *average duration of uninterrupted CO gas availability to each of the different consumers* and the *associated cost saving on methane gas cost*. The duration of uninterrupted CO gas availability at a plant is basically indirectly proportional to the number of switchovers per plant per day, and in the same way the production hours lost per plant per day are related. The aim was extended to *forecast the production hours lost per plant per day* in order to confirm the operational implication of implementing a certain prioritisation philosophy.

Different scenarios could be tested with the simulation model. For this purpose a simplified user input form was developed in *Microsoft® Visual Basic* to allow the analyst to evaluate different scenarios easily with the simulation model.

After certain handpicked scenarios had been evaluated on the simulation model, a full optimisation exercise was embarked on by applying the MOO CEM algorithm in conjunction with the simulation model. The MOO CEM algorithm allowed for the multi-objective nature of the problem to be accounted for. The reader is referred to the detailed concept model ([Figure 54](#)) to appreciate the multi-objective nature of the problem. The MOP was simplified to consider only two objectives, namely the average production hours lost per plant per day and the average total cost saving on methane gas consumption at these plants. These two conflicting objectives were affected chiefly by the specific combination of plants selected to receive CO gas. The higher the



average demand for CO gas, the greater would be the opportunity for a larger cost saving, and simultaneously, the higher the likelihood of CO-methane switchovers. More switchovers would result in operational disruptions that would result in increased production time losses.

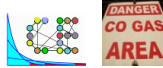
In order to achieve the abovementioned aims, the scope of the project had to be defined clearly. The following table lists aspects of the CO gas problem that were included or excluded from the scope of the project.

Included	Excluded
<ul style="list-style-type: none"> • Priorisation of the instantaneously available CO gas to the highest priority gas consumers according to the <i>Howden</i> plant prioritisation logic (Appendix D). • Estimation of the average time period of uninterrupted CO supply at each consumer over a defined period (replication). • Estimation of the number of CO-methane switchovers at each consumer over a defined period (replication). • Estimation of the production hours lost at each consumer over a defined period (replication). • Estimation of the average saving on methane gas costs by each consumer over a defined period (replication). 	<ul style="list-style-type: none"> • Recommendations on the physical distribution of the CO gas to the consumers, e.g. upgrading of pipes, valves and instrumentation. • Recommendations on maintaining the <i>Howden</i> plant. • Recommendations on maintaining the gas consuming plants. • The economic value of lost production time at the consumers during a switchover. (The reasons for this are that different <i>intermediate products</i> are dried or heated at the burners and that <i>stockpiles</i> of these intermediate products exist between plants. These factors make the comparison of economic value of lost production time quite speculative.) • The distribution of CO gas to the MTP, MSP and the two pre-heaters.

Table 5: Scope clarification table

6.3 Solving the problem in the medium term versus the longer term

A logical solution to this problem would be to determine the optimum size of a pressure vessel to provide a buffer of CO gas between the supply and demand sides of the system. It was, however, *not within the scope of this project* to make a recommendation on the optimum size or the economic feasibility of a buffer container. Some of the reasons for excluding the latter were



discussed in the last two paragraphs of section 6.1.3.1. In the short to medium term the company is not yet in a position to spend capital on the purchase, installation and integration of such a pressure vessel. As the BI Manager on site, the author is the custodian of the medium term capital budget and is fully aware of how current business priorities dictate the capital budget. It can therefore be said with a fair amount of confidence that a pressure vessel is not likely to be included the medium term capital budget.

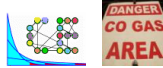
The investigation into the purchase and installation of such a pressure vessel has been identified as a long term prospect and the simulation model can become an effective tool to provide input into such a study. The author has visited the operations of *Tronox Namakwa Sands* on the South African west coast as well as *Eramet Titan & Iron* in Tyssedal, Norway, where such pressure vessels are used in ilmenite smelting operations. This strengthens the argument for the consideration of purchasing a CO gas pressure vessel.

At the time of completing this thesis, the project focused on the short to medium term solution of dynamic prioritisation of the CO gas – a suboptimal solution, as mentioned, but one that can nonetheless effect a significant positive impact whilst being much less capital intensive than purchasing a pressure vessel. The expected impact of dynamic prioritisation was judged as significant enough to continue with the research project and to commence with the implementation. One important aspect to address regarding the implementation was the upgrade of existing equipment (including burner inlets, valves, piping, controllers, *Howden* plant compressors etc.) to a fully operational state and to a level of compliance with legislative requirements. Two other important aspects to address involved the training of staff and the integration of the different plants that will run on CO gas for the first time in a decade. The short to medium term aims of the project were to reap, albeit only a portion then, of the significant cost savings potential of this project and to reduce the environmental impact caused by the emission of CO₂ gas from the burning of methane and CO gas.

6.4 Summary: Chapter 6

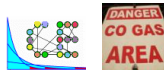
In this chapter it was argued that the CO gas problem is a supply and demand problem where dynamic prioritisation in a stochastic environment is required. Understanding the latter should have added to the validation of the decision to apply simulation as the problem solving tool.

The chapter continued by discussing some key practical aspects that had to be incorporated into formulating the problem. The latter included the usage of methane gas as an alternative to CO gas and operational challenges standing in the way of finally implementing the project. With the operational challenges and conditions in mind, the scope for the simulation study was



clarified. The chapter concluded with a discussion of the immediate to medium term solution proposed, as opposed to a long term solution to the CO gas problem.

The next chapter continues the third and final theme of the thesis, which covers the development of the solution to the research problem. The problem formulation is taken a step further in that it will be demonstrated how the simulation experiment was defined and how the concept model was translated into a simulation model – two steps both in line with the simulation process prescribed by [Banks \(1998\)](#).



7. Defining the experiment

In the previous chapter the formulation of the research problem was presented. The problem formulation included discussing the development of the concept model, both a basic one and a more detailed one. The detailed concept model indicated the input variables, the expected output variables and gave an indication of the relationship between the inputs and outputs. It also specified the sources from which the input data were to be collected and the source of the most important problem logic, i.e. the *Howden* plant prioritisation logic. A clarification of the scope of the project has been included in the previous chapter, as well as a section discussing the approach that would be followed towards developing a short to medium term solution to the problem.

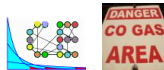
In Chapter 7 the simulation experiment will be defined. First, it will be shown how simulation theory, as discussed in Chapter 4, was applied to the CO gas problem. Subsequently, this chapter will discuss the preliminary experiment in which the level of confidence, the model time span, the input variables and their feasible ranges and the output variables to be observed, will be defined. This chapter will also discuss the process of collecting the input data, the statistical analysis thereof and the translation of the concept model into a computer simulation model.

7.1 Simulation applied to the CO gas problem

As discussed in section 4.2, simulation is generally used in the analysis of complex real-world systems which cannot be assessed using analytic OR techniques due to the complexities of system component interaction. Numerous built-in parameters, variables and functions of simulation software enable it to cope with these interactions better than most other analysis tools (Ziesak *et al.*, 2004). This chapter will illustrate how a simulation software package was used to model the CO gas problem for analysis and optimisation.

First, the *level of abstraction* of this simulation model was taken as the level of an *operational model* (as opposed to a *strategic* or *physical model*, as discussed in section 4.1). The anatomy of this operational model was illustrated in the previous chapter in the form of a detailed conceptual model. It was shown how the process of production, distribution and consumption of CO gas was formulated as a supply and demand problem.

Actual plant data for the period January 2010 to June 2010 was collected as input data to be used in the simulation study. The reason for the selected time period was that it was period of steady operations at *Tronox KZN Sands*. This period excluded any cyclicity or major interruptions to operations and was therefore viewed as a generally representative sample of plant data. It should be noted that, although the input data was collected for a six month period,



the simulation model would not necessarily also have to be run for a full six month period. The duration of simulation runtime would depend on the variability of the collected data. The calculations done to determine the runtime required for the simulation of the CO gas problem are discussed in the next chapter.

Figure 55 shows an example of the actual supply of CO gas versus the demand. The data represents actual data over one day (3 January 2010 06:00 to 4 January 2010 05:59). The blue trend line indicates the supply of CO gas at one minute intervals. The green trend line indicates the demand observed every minute for a certain combination of plants. From this example it is clear that, on average, the supply of CO gas exceeds the demand, i.e. the red trend line is generally above the yellow trend line. When considering the supply and demand balance, every time the blue supply trend line drops below the green demand trend line, a switchover between CO and methane gas is likely to occur, which would result in lost production time. Arrows A and B indicate periods when the supply remained higher than the demand, while C indicates instances where CO-methane switchovers would occur. As mentioned in section 6.2, the main aims of the simulation model were to predict the average production hours lost per plant per day and the average saving on methane gas cost per month. The production hours lost is a function the number of CO-methane switchovers per plant per day, i.e. the “C” instances. During optimisation, the aim was to minimise the average number of “C” instances per day and to maximise average the length of the “A” and “B” type periods.

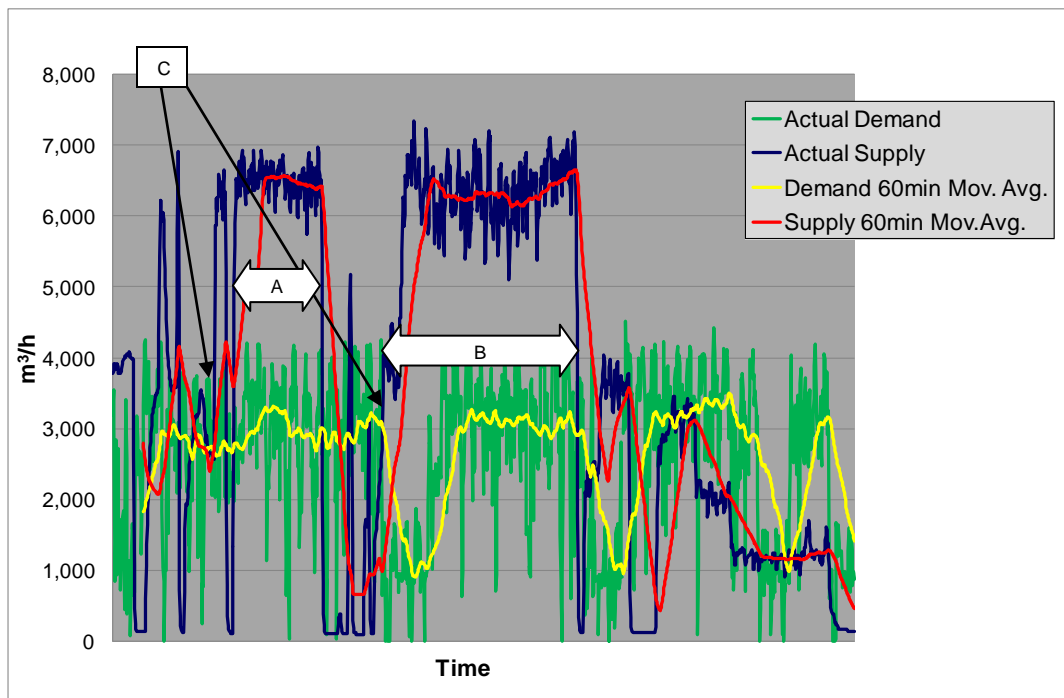


Figure 55: CO gas supply vs. demand



7.2 Preliminary experiment

Following the due process of performing a simulation study (Bekker, 2005) as discussed in section 4.5, the following important aspects of the simulation exercise were defined before the input data could be collected and before the simulation model could be developed:

- The level of confidence for the confidence intervals: 95%
- Model data time span:
 - Input data collection period: six months
 - Model run time: 24 hours per day, seven days per replication⁶.

Next, the input variables (Table 6), constant values (Table 7) and output variables (Table 8) to be used in the simulation were defined. The variable names that were used in *Arena*® to represent the values from these three tables are shown on the list of variables on the CD-ROM.

No	Input variables	Unit	Observed time span	Data source
1	Furnace 1 CO production > 0	m ³ /min	01/01/10 – 31/06/10	<i>Aspen Process Explorer</i> ® ⁷ Tag: 05FI57502C
2	Furnace 2 CO production > 0	m ³ /min	01/01/10 – 31/06/10	<i>Aspen Process Explorer</i> ® Tag: 06FI57602C
3	Furnace 1 downtime (i.e. CO production = 0)	% of total time	01/01/10 – 31/06/10	<i>Aspen Process Explorer</i> ® Tag: 05FI57502C
4	Furnace 2 downtime (i.e. CO production = 0)	% of total time	01/01/07 – 31/07/07	<i>Aspen Process Explorer</i> ® Tag: 06FI57602C
5	Availability of Pre-heater 1 (i.e. MTTF & MTTR)	minutes	NA	Not sufficient actual data available: Assumption 92 % availability.
6	Availability of Pre-heater 2 (i.e. MTTF & MTTR)	minutes	NA	Not sufficient actual data available: Assumption 92% availability.
7	Crude ilmenite feed rate through URIC dryer	ton/min	01/01/10 – 31/06/10	OMS/MPP/URIC Circuit/URIC Feed Measured Ton
8	Slag feed rate through SPP dryer	ton/min	01/01/10 – 31/06/10	OMS/Slag/Dryer/ CO15 Ton
9	Anthracite feed rate through RPP dryer 6	ton/min	01/01/10 – 31/06/10	OMS/Reductant/Overview/ Dryer 6 Ton

⁶ Replication length will only be calculated later in the simulation process and thus discussed in the next chapter

⁷ *Aspen Process Explorer*® is a software package at *Tronox KZN Sands* containing process data. The tag numbers are indexes to these datasets in the software package.



No	Input variables	Unit	Observed time span	Data source
10	Anthracite feed rate through RPP dryer 8	ton/min	01/01/10 – 31/06/10	OMS/Reductant/Overview/ Dryer 8 Ton
12	Availability of URIC (i.e. MTTF & MTTR)	minutes	01/01/10 – 31/06/10	OMS/MPP/Delays/URIC Circuit ⁸
13	Availability of SPP dryer (i.e. MTTF & MTTR)	minutes	01/01/10 – 31/06/10	OMS/Slag/Delays/Dryer delays
14	Availability of RPP dryer 6 (i.e. MTTF & MTTR)	minutes	01/01/10 – 31/06/10	OMS/Reductant/Delays/Dryer 6
15	Availability of RPP dryer 8 (i.e. MTTF & MTTR)	minutes	01/01/10 – 31/06/10	OMS/Reductant/Delays/Dryer 8
16	Availability of <i>Howden</i> Plant (i.e. MTTF & MTTR)	minutes	01/01/10 – 31/06/10	Availability was ranged between 80% and 100% with steps of 0.5%

Table 6: Input variable information

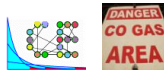
No	Constant	Value	Unit	Data Source
1	CO gas calorific value	0.0112	GJ/m ³	PowerDocs ⁹ #203930
2	Methane gas calorific value	0.03558	GJ/m ³	PowerDocs #203930
3	Methane gas unit cost	62.93	R/GJ	<i>Tronox KZN Sands</i> SAP ¹⁰
4	Typical energy requirement: Preheater 1	1.138	GJ/ton	(Brouwer <i>et al.</i> , 2003)
5	Typical energy requirement: Preheater 2	1.138	GJ/ton	(Brouwer <i>et al.</i> , 2003)
6	Typical energy requirement: URIC dryer	0.382	GJ/ton	(Brouwer <i>et al.</i> , 2003)
7	Typical energy requirement: Slag dryer	0.334	GJ/ton	(Brouwer <i>et al.</i> , 2003)
8	Typical energy requirement: Reductant dryers	0.450	GJ/ton	(Brouwer <i>et al.</i> , 2003)
9	Switchover period at URIC	20	minutes	Assumption
10	Switchover period at URIC	20	minutes	Assumption
11	Switchover period at URIC	20	minutes	Assumption

Table 7: Constant values used in the model

⁸ The OMS is the online Operations Management at *Tronox KZN Sands* containing process data

⁹ *PowerDocs* is the *Tronox KZN Sands* Electronic Document Management System (EDMS)

¹⁰ *SAP* is the *Tronox KZN Sands* Enterprise Management System containing cost and production data



No	Output variables	Unit
1	Average period of uninterrupted CO gas supply at URIC dryer	min
2	Average period of uninterrupted CO gas supply at SPP dryer	min
3	Average period of uninterrupted CO gas supply at RPP dryers	min
4	Average number of switchovers at URIC dryer	times/day
5	Average number of switchovers at SPP dryer	times/day
6	Average number of switchovers at RPP dryers	times/day
7	Average total gas energy used per replication at the URIC dryer	GJ/replication
8	Average total gas energy used per replication at the SPP dryer	GJ/replication
9	Average total gas energy used per replication the RPP dryers	GJ/replication
10	Average total gas energy used per replication by all users	GJ/replication
11	Average CO gas energy used per replication by all users	GJ/replication
12	Average methane gas cost incurred per replication	R/replication
13	Average saving on methane gas cost per replication	R/replication
14	Average hours lost per plant per day	hr/day
15	Average overall saving on methane gas cost per month	R/month

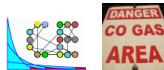
Table 8: Output variable information

- **The system type (i.e. a terminating system or non-terminating system)**

This problem was modelled as a terminating system. The process was defined as terminating at the end of each predefined period (replication). This period must be set in *Arena*® before a simulation run is performed and the results must be interpreted accordingly. The process can therefore easily be studied as a terminating system over a period of any desired length (replication length). When changing the replication length one should ensure that the total simulation run time (replication length multiplied by the number of replications) does not exceed the duration of six months over which the input data was collected and that the statistical integrity of the data is kept intact.

- **Warm-up period**

It should be noted that no warm-up period was explicitly designed into the simulation. This is due to the fact that the way in which the input data was generated by the model implicitly included warm-up periods typical of real furnace behaviour. The physical furnace operation and smelting reaction were not explicitly modelled in this simulation, as this was deemed unnecessary for the purposes of this study. This decision by the author refers back to the level of abstraction that was

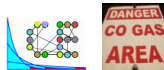


discussed in the section on modelling in 4.1. When modelling, the level of abstraction should fit the purpose of the outcomes expected from the model. The production and management of furnace off-gas was modelled on an *operational level* rather than a *physical level*. This might not be acceptable from a *pyrometallurgical point of view*, but was sufficient from an *operations research point of view* in this case.

Large pyrometallurgical furnace processes often contain much complexity and uncertainty which makes it difficult to reason intuitively about their dynamic behaviour, as well as their products. The ilmenite smelting furnace process presents a number of interesting and hard process control challenges (Zietsman & Pistorius, 2006). For a detailed investigation of the production and utilisation of furnace products, including furnace off-gas, a suitable process model of the furnace and off-gas system would be required. There are only limited references to dynamic EAF models in the literature, as many such models tend to be proprietary. For example, Burstrom *et al.* (1992) derived a dynamic EAF model based on mass and energy balances. In (Bekker *et al.*, 1998; 1999a) extensive modelling efforts were applied to model the operation of an EAF for model predictive control of a steel making EAF. This model by Bekker *et al.* comprised 17 non-linear ordinary differential equations and was approximated by a linear time invariant state space system for controller design purposes (Bekker *et al.*, 1998).

As the research discussed in this thesis is focused on the level of operational management of the furnace off-gas, the modelling of the furnace operations (which is not an arbitrary exercise, as shown above) was not undertaken. Rather, the historical production of CO gas at the *Tronox KZN Sands* smelters was used in the statistical analysis of CO gas production as input into the simulation model. Statistical distributions could be fitted onto these stochastic input data sets in order to generate similar data with simulation.

Considering the above, a warm-up period was not included, seeing that the input data that was generated in the simulation would automatically mimic real furnace behaviour and consequently include warm-up periods. Warm-up periods would occur after every time one or both of the furnaces experienced a downtime (power off) or reduced feed or power. Even when a furnace is “switched off” or the production rate is lowered, this would not happen instantaneously, but as a gradual, controlled process or “ramp-down” period. The data generated would therefore also automatically include such ramp-down periods. The graph in Figure 56 illustrates the intrinsically included warm-up and ramp-down periods in simulation input data. The data shown in the graph illustrates the total CO gas produced in one day by both furnaces. The warm-up periods after furnace dips or downtimes are clearly shown. The replication length has been defined as seven days (one week). A replication would thus consist of seven consecutive days, similar to the day shown in Figure 56. Because the system was modelled as a terminating system, a full simulation run would represent a number of weeks. Using the defined replication length and performing enough replications (observations) per simulation, real furnace behaviour



could be modelled to a sufficient level of accuracy for meaningful study and for conclusions to be reached.

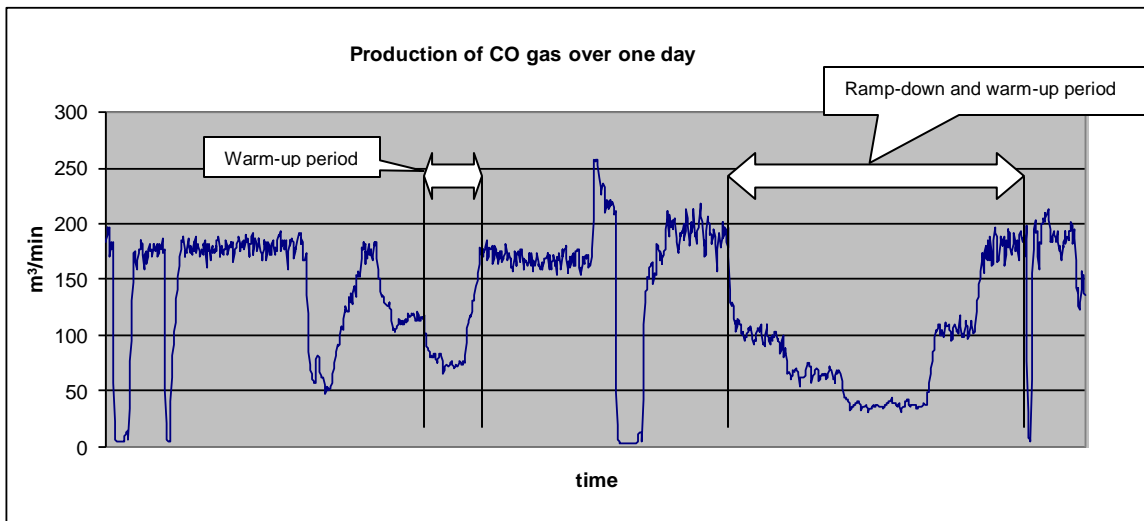


Figure 56: CO gas production showing warm-up periods

• **Definition of the simulation entity:**

- On the supply side of CO gas, an entity was defined as a “batch” of CO gas supplied by the combination of the two furnaces in the last minute of simulation time. In the model this entity was graphically shown as a bubble of CO gas moving through the system and triggering system changes. The continuous phenomenon of CO gas production was thus made discrete. This entity represents $x \text{ m}^3$ of CO gas.
- On the consumption side of CO gas, an entity was defined as an “inspecting element”, depicted as an inspector that “checks” or “inspects” the status of all the gas consumers, thereby effectively establishing the demand for CO gas every minute.
- Table 9 shows how these two entities are graphically depicted in *Arena*®.



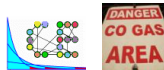
Entity 1	Function: moves through the system in the simulation model as a batch of CO gas that triggers system changes such as the prioritising of CO users, recording of switchovers etc.	
Entity 2	Function: checks the status of resources, i.e. gas consumers, to confirm whether each is operating or down, from which the total demand for gas is calculated.	

Table 9: Depiction of entities in *Arena*®



- **Attributes per entity**

This model makes use of only two attributes for entity 1 on the supply side of the model: the volume and the energy content of a “batch” of CO gas in m³/batch and GJ/batch. The volume and the energy content of a batch are directly related to each other through the calorific value (in GJ/m³) of the gas.

7.3 Selecting the parameters to vary and observe

7.3.1.1 Input variables

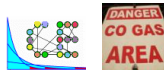
From the list of input variables shown in Table 6, only variables 12 to 16 were identified as control variables (variables 14 and 15 from Table 6 were combined into a single variable), whilst one to 11 were distribution variables. Additional to the input data for which historical data could be collected (Table 6), three additional input variables were identified. These were defined as boolean variables that could be set to zero or one to evaluate different combinations of plants to which CO gas needed to be distributed. These seven input variables thus formed the *decision variables* (DVs), shown in Table 10, while different combinations of the DVs would form the *search space* or *decision space*.

Index	Input variables to study (decision variables)	Type	Range
DV ₁	URIC set to receive CO gas?	Discrete	0 or 1
DV ₂	SPP set to receive CO gas?	Discrete	0 or 1
DV ₃	RPP set to receive CO gas?	Discrete	0 or 1
DV ₄	Availability of <i>Howden</i> Plant	Discrete	80%-100% step 0.5%
DV ₅	Availability of URIC dryer	Discrete	86%,90%,94%
DV ₆	Availability of Slag dryer	Discrete	86%,90%,94%
DV ₇	Availability of Reductant dryers 6 and 8 combined	Discrete	86%,90%,94%

Table 10: Decision variables identified

7.3.1.2 Output variables

From the list of output variables shown in Table 8, only the last two, i.e. 14 and 15, were selected for optimisation, and these are described in Table 11. These two output variables are therefore the objectives in the optimisation exercise and are termed f_1 and f_2 . The two objectives were selected such that they are descriptive of the system behaviour that the analyst intended to study.



Index	Output variables to study (Objectives)	Type	Requirement
f_1	Average hours lost per plant per day	Continuous	Minimise
f_2	Average overall saving on methane gas cost per month	Continuous	Maximise

Table 11: Objectives of the CO gas problem identified

Typical of the simulation process, a statistical analysis was performed on each set of output data generated by the simulation model (Bekker, 2005). The variance in the output data set was used to determine the number of replications required for production runs in order to make statistically sound inferences. The calculation of the number of replications required per production run will be demonstrated in the next chapter.

For each of the output parameters the simulation model returns an estimated expected value, $E(X)$, as well a confidence interval at the 95% confidence level.

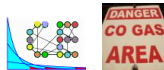
7.4 Collecting and statistical analysis of input data

All the input variables and constants were discussed in the two previous sections. These input variables and constants were collected from the sources indicated. Statistical analyses were performed on each set of input data using *Input Analyzer* from *Arena®* and *Microsoft® Excel* in order to fit statistical distributions that could be used in the model. Examples of the statistical analysis performed on each set of input data can be viewed for three sets of input data in [Appendix A](#).

7.5 Translation of the model into a computer simulation language

The simulation model was developed in the *Arena®* simulation package (Kelton *et al.*, 2007). *Arena®* is a product of *Rockwell Automation Technologies*. Before developing the *Arena®* model, the detailed concept model was first translated into a *Microsoft® Visio* flow diagram. A *Microsoft® Excel* sheet and *Microsoft® Visual Basic* user form were also used to allow a user-friendly way of feeding user selected inputs (for handpicked scenarios) into the *Arena®*. The user form was developed before the MOO CEM metaheuristic was programmed in order to automate the optimisation process.

The simulation model, as developed in *Arena®*, consisted of six different submodels and 173 variables. The submodels can each be viewed on a separate screen in *Arena®*. The simulation model also included an animation view. [Table 12](#) shows the different submodels that constitute the *Arena®* model. For a screen print of each of the submodels, the reader is referred to [Appendix C](#).



Submodel 1	Creating the demand for gas
Submodel 2	Creating the supply of gas by the two furnaces
Submodel 3	Prioritising the CO gas with one furnace down
Submodel 4	Prioritising the CO gas with both furnaces running or both down
Submodel 5	Registering the gas consumed with one furnace down
Submodel 6	Registering the gas consumed with both furnaces running or both down

Table 12: Submodels of the Arena® model

Figure 57 and Figure 58 are two screen prints from the Arena® model. The first shows an overview of the submodels and the second a screen print of the animation view. In both cases the screen prints were taken when the model was not running. Figure 57 can also be seen in a larger view in Appendix C.

The Microsoft® Visio flow diagrams can be seen in Appendix B. The Arena® simulation model and the Matlab® code in which the MOO CEM algorithm was programmed are available on the CD-ROM that was submitted with this thesis.

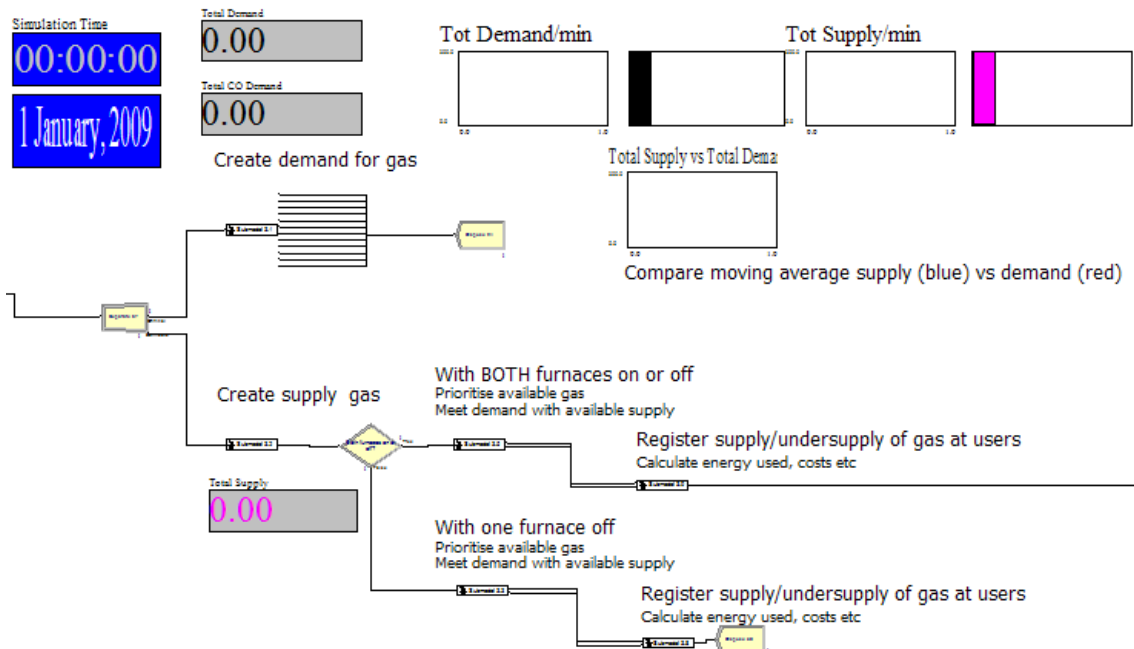


Figure 57: Screen print of overview of submodels in Arena®

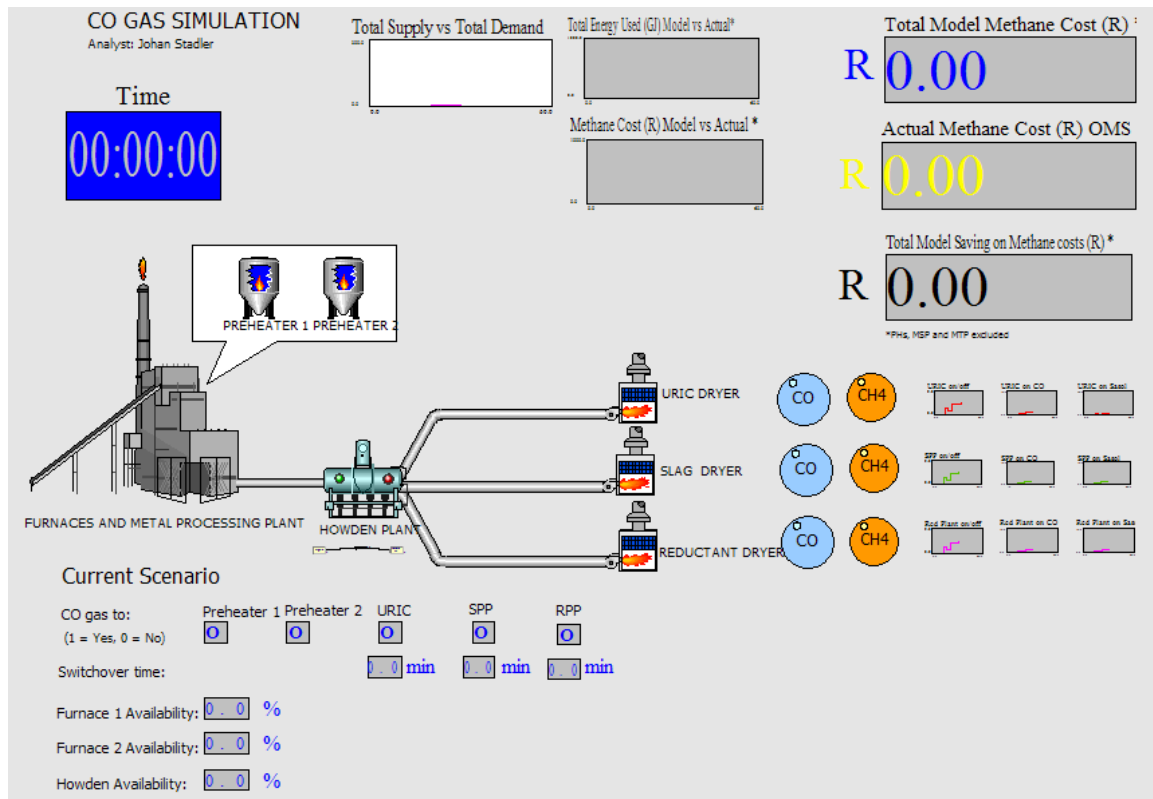
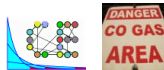


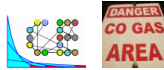
Figure 58: Screen print of the animation view in *Arena*®

7.6 Summary: Chapter 7

In this chapter the simulation experiment has been defined. This step was necessary before simulation production runs could be performed and involves defining the level of confidence, the model time span, as well as the input variables impacting on the system and the output variables that would be observed.

After defining the input variables, the next step was to identify which of these were to be varied in the simulation. In other words, the control variables (decision variables or DVs) together with their feasible regions, needed to be defined. At this point the problem was identified as of the discrete type, as all the DVs identified were defined as discrete in nature.

A similar exercise was required for the output variables. Once they had been identified, those to be observed (the objectives) and their feasible regions needed to be defined. The objectives were limited to two, namely the average hours lost per plant per day, indicated as f_1 , and the average overall saving on methane gas cost per replication, identified as f_2 . With these

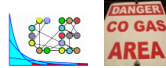


two conflicting objectives identified for optimisation, it was shown that the CO gas problem can be classified as a MOP.

The collection and analysis of the sets of input data were discussed next, with the results of three sets of the input data analyses being included in [Appendix A](#) as examples.

The chapter ended with a discussion on the translation of the concept model into a computer simulation model. The software package used for simulation was *Arena*®. The final section of the chapter included a table showing the different submodels in the *Arena*® model, an *Arena*® screen print showing how the different submodels relate to each other, and a screen print from the animation view in *Arena*®.

With the problem formulated, the experiment defined and the computer simulation model developed, the simulation process could progress to the execution of the production runs. The next chapter will take the reader through the performing of the production runs and the optimisation process, i.e. the process of applying the MOO CEM algorithm to the *Arena*® simulation model.



8. Production runs and applying MOO CEM

The simulation problem was formulated, through the development of a detailed concept model, in Chapter 6, while the simulation experiment was defined in Chapter 7. Two key aspects in the defining of the experiment were the identification of the parameters to vary and observe, i.e. the control input variables and output variables to observe. Thereby the decision variables and the objectives for the optimisation exercise were defined. Chapter 7 also discussed the decision to model the problem as a terminating system, defined entities and attributes in the context of the CO gas problem and discussed the translation of the concept model into a simulation model.

In this chapter the final steps of the simulation process (Banks, 1998; Bekker, 2005), as discussed in section 4.5, are discussed. These steps include the calculation of the number of replications required to make statistically sound inferences, the validation and verification of the simulation model, and the execution of the production runs.

The chapter is concluded with a discussion of the optimisation that was done on the CO gas problem. This included an explanation of how the outputs from simulation runs (objectives) were optimised. The optimisation was first attempted by applying a self-designed “decision factor” which combined selected simulation outputs in a mathematical formula. Thereafter the optimisation was done by applying the MOO CEM algorithm to the simulation model. As part of discussing the optimisation of the problem with the MOO CEM algorithm, the approximated Pareto front obtained with the metaheuristic is also shown, compared to the true Pareto front and interpreted.

8.1 Initial run to determine preliminary statistics and the number of replications required

An initial simulation run was performed to determine the preliminary statistics and the number of replications required for a confidence level of 95%. The *replication length* of the initial run was set at seven days and the initial run was performed for a randomly chosen number of ten replications, i.e. ten pseudo-independent observations of one week of system behaviour. Although the input data was collected and analysed over a six month period, preliminary simulation runs with replication indicated that the variability in the data was such that a replication length of one week would not violate the statistical accuracy of the study, as long as the correct number of observations (replications) was performed.

Using (46), it was determined that 14 replications per simulation run were required for a satisfactory confidence level in the simulation output. (Refer to n^* for f_2 , marked in yellow in

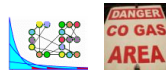


Table 13a.) Thus, to perform the production runs, each simulation run would be set to 14 replications, each with a replication length of seven days, for the simulation to yield statistically acceptable results within the minimum time. The required number of replications, n^* , was calculated by

$$n^* = n \left(\frac{h}{h^*} \right)^2, \quad \dots 46$$

where

h^* = the desired confidence interval half-width,

n^* = the required number of replications,

n = the number of initial replications,

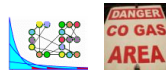
h = the confidence interval half-width that results from n replications calculated from (47).

$$h = t_{n-1; 1-\alpha/2} \frac{s_{\bar{x}}}{\sqrt{n}} \quad \dots 47$$

The student t-distribution was used to calculate the confidence interval half-width by making use of the $s_{\bar{x}}$, calculated from the outputs of the preliminary simulation run. The reader is referred to (Bekker, 2005) for a full discussion on the calculation of the required number of replications. When using *Arena*® as the simulation package, the value of h is calculated and displayed in the output results. This allows the analyst to easily apply (46) to calculate n^* .

Table 13 indicates the number of replications required (n^*) to reach an average value, \bar{X} or $E(\bar{X})$ or X_{barbar} , per output variable that falls within a preselected confidence interval half-width. The preselected confidence interval half-widths for each of the three output variables shown are shown in the column, h^* (per month).

As an example, refer to the interval half-width of f_2 . The confidence interval half-width was preselected as R50 000. This will give a sufficiently accurate estimate of the monthly methane gas cost saving, considering the average monthly methane gas cost for the particular set of 2010 data used is around R850 000 (refer to Figure 50). The last column indicates the number of replications required (n^*) when the model was run with ten replications per simulation run and replication lengths of seven, 14 and 30 days respectively. It follows from Table 13 that as the replication length increases, n^* decreases. For replication lengths of 14 and 30 days the initial ten replications per simulation run were sufficient, with the simulation runtimes of each of those experiments 0.48 minutes and one minute respectively. Selecting a replication length of seven days, with 14 replications per run, resulted in 0.32 minutes (not shown) of simulation runtime. Thus, for time purposes, selecting the production runs with a replication length of seven days and



14 replications per run would not only yield acceptable results, but also require the shortest simulation runtime.

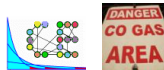
		Replication length (days)						
		7						
a)		Number of replications per run	10					
	Arena® output variable	<i>n</i>	<i>X_{bar}</i> (per rep)	<i>X_{bar}</i> (per mnth)	<i>h</i> (per rep)	<i>h</i> (per mnth)	<i>h*</i> (per mnth)	<i>n*</i>
<i>f</i> ₁	AvgHrsLostPerUserPerRep	10	36.39	155.95	2.98	12.77	20	5
<i>f</i> ₂	CostSavingTotalRperRep	10	211,495.99	906,411.40	13,546.40	58,056.00	50,000	14
	Energy Total GJ per rep	10	3,835.36	16,437.24	152.99	655.67	800	7
		Simulation time	0.25 min					
		Replication length (days)						
		14						
b)		Number of replications per run	10					
	Arena® output variable	<i>n</i>	<i>X_{bar}</i> (per rep)	<i>X_{bar}</i> (per mnth)	<i>h</i> (per rep)	<i>h</i> (per mnth)	<i>h*</i> (per mnth)	<i>n*</i>
<i>f</i> ₁	AvgHrsLostPerUserPerRep	10	74.91	160.52	5.08	10.89	20	3
<i>f</i> ₂	CostSavingTotalRperRep	10	420,550.82	901,180.33	14,766.94	31,643.44	50,000	5
	Energy Total GJ per rep	10	7,598.70	16,282.94	273.61	586.31	800	6
		Simulation time	0.48 min					
		Replication length (days)						
		30						
c)		Number of replications per run	10					
	Arena® output variable	<i>n</i>	<i>X_{bar}</i> (per rep)	<i>X_{bar}</i> (per mnth)	<i>h</i> (per rep)	<i>h</i> (per mnth)	<i>h*</i> (per mnth)	<i>n*</i>
<i>f</i> ₁	AvgHrsLostPerUserPerRep	10	162.16	162.16	8.33	8.33	20	2
<i>f</i> ₂	CostSavingTotalRperRep	10	894,102.01	894,102.01	29,173.45	29,173.45	50,000	4
	Energy Total GJ per rep	10	16,175.03	16,175.03	482.78	482.78	800	4
		Simulation time	1.00 min					

Table 13 a-c: Calculating the number of replications required

8.2 Execution of production runs

8.2.1 Initial scenarios

When the simulation model was originally developed, the first aim was to provide *Tronox KZN Sands* with a tool to evaluate the CO gas problem, whilst incorporating its stochastic nature. Recall that the existing static models were unable to provide meaningful answers to the CO gas problem (refer to the last paragraph of section 1.1). The simulation model was originally developed to evaluate at least seven scenarios, which were essentially the seven different combinations of gas consumers that could be set to receive CO gas, i.e. all the possible combinations of only DV₁, DV₂ and DV₃ shown in Table 10. The scenario where all three of these DVs were set to zero was excluded. These seven scenarios are shown in Table 14. The reason for evaluating different combinations of the CO gas users was to understand the relevant benefits and challenges associated with each of the seven combinations. The objective was to allow the



analyst to decide what the optimum combination of users would be in terms of the theoretical benefits estimated and the practicality of the solution. For some reason, the prevailing view amongst operational personnel was that, since CO gas could not be distributed to *all* the users, it could not be used *at all*, which led to the usage of CO gas simply being mothballed for a number of years. The author felt that instead of getting *no* benefit from the CO gas by not distributing it to any users, it had to be established what the benefit and practical implications of distributing the available gas to at least to *some* users would be.

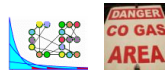
The simulation model, with a user input form, was developed so that the seven different combinations of users could be evaluated easily. The output of the model would guide the analyst to recommend to management which of the user combinations would be the simplest to implement at the time, whilst showing the trade-off of financial benefit versus practicality associated with each combination of users.

Table 14 shows the seven scenarios that were tested originally, where $DV_i = 1$ means “Yes the plant receives CO gas” and $DV_i = 0$ means “No CO gas to the plant”. Figure 59 shows the user input form that was developed to allow the analyst to select a combination of plants to receive CO gas. The selections made on the user form would then be saved to the *Microsoft® Excel* sheet (shown in Figure 60), which would feed the inputs into the *Arena®* model. The specific combination of CO gas users selected by the analyst would constitute a *scenario* and require a *simulation production run*.

The user input form did provide the analyst with the option to change some of the other DVs, but initially the focus was on the seven scenarios discussed above.

	DV ₁	DV ₂	DV ₃
Scenario nr	URIC on CO	SPP on CO	RPP on CO
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1

Table 14: The seven combinations of CO gas users analysed initially



Analyst Input Form

Users to get CO

	Yes	No
Preheater 1	<input type="checkbox"/>	<input type="checkbox"/>
Preheater 2	<input type="checkbox"/>	<input type="checkbox"/>
URIC	<input type="checkbox"/>	<input type="checkbox"/>
RPP	<input type="checkbox"/>	<input type="checkbox"/>
SPP	<input type="checkbox"/>	<input type="checkbox"/>

Clear Users

Switchover time per user
(Default = 5 minutes)

URIC	5 min
RPP	5 min
SPP	5 min

Howden Availability 92 %
(0% - 100%, Default = 92%)

User Availability
(Default = 92%)

Preheater 1	92 %
Preheater 2	92 %
URIC	92 %
RPP	92 %
SPP	92 %
Furnace 1	76 %
Furnace 2	81.4 %

OK

Figure 59: User input form for scenario selection

	A	B	C	D	E	F	G
1	CO gas to users:		0 or 1		Availability		(%)
2							
3	PH1		0		PH1		92.0%
4	PH2		0		PH2		92.0%
5	URIC		0		URIC		94.0%
6	RPP		0		RPP		94.0%
7	SPP		0		SPP		94.0%
8					FC1		92.0%
9					FC2		92.0%
10							
11							
12	Howden Availability:		0 - 100 %		Switchover time		(minutes)
13							
14	HOWDEN		92.0%		URIC		20
15					RPP		20
16					SPP		20
17							
18							
19							
20							
21							

Choose simulation scenario

Figure 60: Selected scenario saved to Microsoft® Excel



The relationship between inputs and outputs in the model is such that the more plants selected to receive CO gas, the higher the peak demand for CO gas will be. In turn, a higher peak demand results in a higher likelihood of situations where an undersupply of CO gas could occur and thus a higher likelihood of interruptions in CO gas supply at the plants. More interruptions in CO gas supply will cause more CO-methane switchovers which, in turn, will result in production interruptions, will take personnel out of their normal work to man a line purge and inspect the switchover, and effectively reduce the saving on methane gas cost.

Remember that for this study CO gas was not distributed to the pre-heaters. However, the potential exists for the pre-heaters to be brought on line in future. For further evaluations any additional consumers, including the pre-heaters and the MTP, can be included in the simulation.

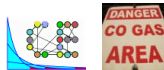
8.2.2 Extending the number of scenarios

The impact of some important input variables was not considered when the seven initial scenarios were evaluated. In fact, the other important variables were not excluded, but their values were set to default values and kept fixed for all seven initial scenarios. For a more realistic study of the problem the impact of the other important input variables, i.e. the other DVs (with their allowed ranges as shown in [Table 10](#)) had to be incorporated into the simulation.

An example of the impact of one of these input variables that needed to be added is the *Howden* plant availability. *Howden* plant availability was defined to range between 80% and 100% in steps of 0.5%, but the default value was set at 92%. With the *Howden* plant being the distributing link between the furnaces and the CO gas consumers, it is logical that the higher its availability, the better the CO gas availability to the consumers would be. The whole feasible range of values for this input variable thus needed to be evaluated. In turn, this could affect simulation outputs significantly: over the long term, maintaining high availability at the *Howden* plant will induce a higher average monthly maintenance cost, which must be offset against the methane gas cost saving.

This resulted in a significant increase in the number of scenarios to be evaluated. The other important input variables selected to be incorporated in the simulation are listed in [Table 15](#). The default values used in evaluating the seven initial scenarios are also shown. The problem therefore had many more than seven scenarios. In fact, it will be shown later that the problem, as defined, comprised 7 560 possible scenarios.

Some important considerations regarding the availabilities listed in [Table 15](#) are discussed below.



Input variable	Identified as DV?	Default value
Furnace 1 availability	No	92%
Furnace 2 availability	No	92%
<i>Howden</i> plant availability	Yes, DV ₄	92%
URIC availability	Yes, DV ₅	94%
SPP availability	Yes, DV ₆	94%
RPP availability	Yes, DV ₇	94%
URIC switchover time	No	20 min
SPP switchover time	No	20 min
RPP switchover time	No	20 min

Table 15: Default values of input variables selected to include in full

- **Furnace availabilities:**

Furnace availability was defined as the percentage of time that the CO gas production from a furnace exceeded zero m³/min for longer than five minutes. This could be calculated from the historical trends of CO gas production at each furnace. Refer to [Appendix A](#) to view the statistical analysis and histogram for the production of CO gas per furnace.

- ***Howden* plant availability**

Seeing that the *Howden* plant has never been operated at its full capacity, nor been operated for periods of time longer than a few weeks since the inception of *Tronox KZN Sands*, an assumption of 92% as the default value was made. However, for a more detailed investigation, the *Howden* plant was given a range as shown in [Table 10](#).

Since there is a lack of history on *Howden* plant operation, there was also no accurate history on maintenance cost. Periodically, relatively large repairs were made to the *Howden* plant to render it operational. These once off capital expenditures, together with a forecast monthly maintenance cost (estimated to be incurred when the plant is fully operational), were used to estimate the monthly running cost. [Figure 61](#) shows the estimated relationship between availability and monthly maintenance cost.

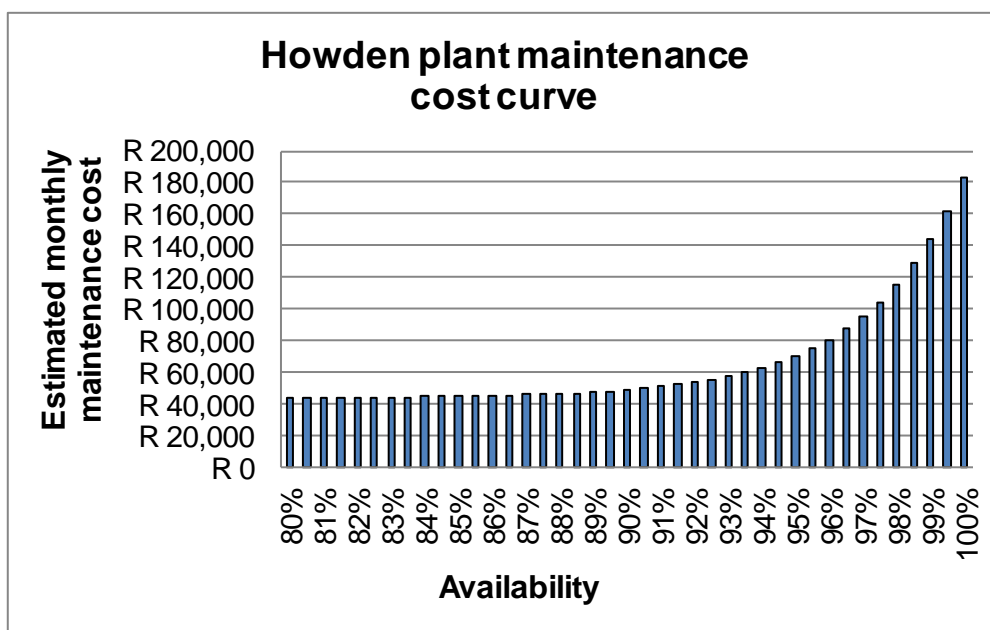
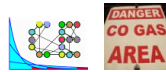


Figure 61: Estimated *Howden* plant maintenance cost

- **Plant availabilities**

Table 10 and Table 15 show plant availabilities, i.e. those of the URIC, SPP and RPP, as single percentages. In reality, detailed information on the availabilities of these plants is available on the *Tronox KZN Sands* Operations Management System (OMS). The mean-time-to-failure (MTTF), in other words duration of runtimes, and mean-time-to-repair (MTTR), or duration of downtimes, were drawn from the OMS. These data sets were also analysed during the collecting and statistical analysis of input data, as discussed in section 7.4. The reasoning is that the longer a plant is running, the longer it will require gas to keep its burner running. Thus the runtime of the plants has an effect on the demand for gas, whether it is CO gas or methane gas. It must also be noted that, although the plants process material in sequential processes, the different plants operate to a large extent independently of each other. The reason for the independence is that there are buffers of intermediate material. In other words, although the URIC supplies ilmenite to the furnaces, it may be that the URIC is down while the furnaces are drawing ilmenite from the URIC silos, and similarly for the other plants.

The run and downtime of gas consumers over time, therefore, had to be taken into account when simulating the distribution of CO gas. This implies that a burner operating on CO gas could stop as a result of either *an undersupply of CO gas*, or as a result of *a higher priority user starting up and demanding CO gas*, or when the *plant where the burner is situated, goes down for*



maintenance – either planned maintenance or due a plant failure. Two further considerations here are:

1. When a gas user goes down for maintenance purposes, while there is in fact enough CO gas available to supply it, the supply side of CO gas cannot be blamed for the end of the CO gas run. Such stoppages of CO gas runs need to be distinguished from stoppages due to undersupply. Yet, in reality, the plant will go down and potentially trigger a lower priority plant to come online and receive the CO gas that has become available, hence triggering a switchover from methane gas to CO gas at the lower priority plant. In order to investigate only the effect of undersupply of CO gas on the number of switchovers caused, the MTTR at the consumers could be set to zero.
2. A CO gas consuming plant that goes down for maintenance can go down for planned or unplanned maintenance due to an operational failure. In the case of planned maintenance, other plants should be scheduled in advance to replace the plant exiting the CO distribution network, in order to limit losses due to switchover times at the incoming plant(s). In the case of an operational problem causing unplanned downtime, plant operators should warn other plants as far as possible before the plant exits the CO gas network, to allow other plant(s) to prepare for a switchover. The simulation model was not designed to include the detail of planned and unplanned downtimes, but this is an important practical consideration.

- **Switchover time**

Once again, since the CO gas distribution system has never been operational for a meaningful period of time, no information existed to determine accurate switchover times. In consultation with plant engineers, a conservative estimate of 20 minutes per switchover was used in the model. The reader is reminded that the switchover and purge is partially a manual process, increasing its duration. Furthermore, the conveyor belts feeding the burners cannot be stopped immediately, but would require a ramp down before stopping and a ramp up back to normal speed after a switchover.

These are some aspects that led to the selection of the default switchover time of 20 minutes. It has also been confirmed that some of the activities at a switchover can be automated but due to the hazardous nature of CO gas, a manual inspection is still required. Once the burners have proved to be operationally successful for a considerable period of time, for example three consecutive months, further improvements such as the full automation of the burners can be reviewed and the switchover period per burner can be calculated from actual performance history.



8.2.3 Archiving of candidates and solution sets

Guided by the supervisor, the author applied archiving of the solution sets to save simulation time. This was particularly useful when applying the MOO CEM algorithm. This allowed that, during algorithm execution, previous candidate solutions could be selected again by the MOO CEM algorithm. An archive of previously evaluated candidates and their solutions was maintained. Before a candidate solution from a new population was evaluated with the simulation model, the algorithm first checked whether or not it existed in the archive. If it was found, its solution was retrieved from the archive, otherwise the simulation model was called to evaluate the candidate. When the simulation run was done, the result was then appended to the archive. Simple sequential search of the non-ordered archive list was used, so the complexity was $O(N_a)$ and the expected number of searches is $N_a/2$, where N_a is the number of members in the archive. The archive search required additional storage space. Initially the archive is empty, but it grows with the algorithm iterations.

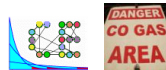
8.3 Model verification and validation

Before the production runs could be done with confidence, the model had to be verified and validated.

Verification allows the analyst to confirm that the model was built correctly. It involves the debugging of the model to ensure that all syntax errors are eliminated, logic is checked and compiler and runtime errors are addressed. If no errors are found during the first few simulation rounds, it cannot be assumed that the model is error free. It would be more likely that errors have not manifested yet, with the given data sets (Bekker, 2005). In a sense, verification is an ongoing exercise, performed as the analyst works through different simulation runs and different scenarios. As errors are found, they must be corrected. It is, however, important that the analyst comes to a point where he/she is confident that no further major errors will occur to affect the performance and accuracy of the model.

Validation requires the analyst to confirm that the correct model has been built. In other words, the analyst essentially needs to confirm that the model is a sufficiently accurate representation of the real-world system that is being imitated. There are three aspects to validation, the first one having already been mentioned. These aspects are (Bekker, 2005):

- **Conceptual validity** – the degree to which the model represents the real-world system adequately



- **Operational validity** – the degree to which the model's generated data can be associated with the real-world system's behavioural data
- **Credibility** – the degree to which the model's end user has confidence in the model's results.

The CO gas model was validated predominantly by comparing the monthly average total gas energy (GJ) that is used in the model to the average total gas energy per month that was used in reality for the period under study. The reason for comparing energy and not volumes of gases is that CO and methane gas have different calorific values (Table 7). In other words, higher volumes of CO gas would be required to supply the same tonnes fed into a burner compared to methane gas. The energy required per ton of feed, however, is fixed per burner and by converting the volumes of CO gas produced per minute, and distributed, to energy supplied to the burners, the energy consumption from CO gas in the model could be compared with actual methane gas energy consumed at the burners over the equivalent period of time in reality. The degree to which the total energy from CO gas supplied to all the burners in the model correlated with the total energy from methane gas consumed in reality by the burners, was a good indication of the model's validity. It was found that if this correlation was close, most of the other output variables from the model and actual figures corresponded.

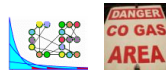
The model validation was done by comparing the energy used by the model when *all the burners were set to run on CO gas* in the model, with a *switchover time of zero minutes*, with the actual energy used. Table 16 compares the actual average monthly energy consumption over the study period of 1 January 2010 to 1 July 2010 with the average monthly energy consumption in the model. The actual energy used were calculated by multiplying the average monthly tonnes fed per plant by the typical energy requirement as indicated in Table 7.

Actual (calculated)	19,452 GJ/month
Model	20,686 GJ/month
Model's deviation from actual	6.34 %

Table 16: Comparison of actual monthly energy used vs. model energy used

From the results of the validation, it was found that the model corresponds satisfactorily with reality. A deviation of 6.34% from the average actual monthly energy burned was considered a sufficiently accurate representation of reality, considering the high level of variability of many system variables and inaccuracies in gas flow meters in the plants.

Figure 62 shows a graph from the *Arena*® model that was included in the animation view to indicate to the analyst during a simulation run how well the model is correlating with actual data.



The blue line represents the gas energy consumed in the model and the yellow line the energy that was consumed in reality over the same period of time.

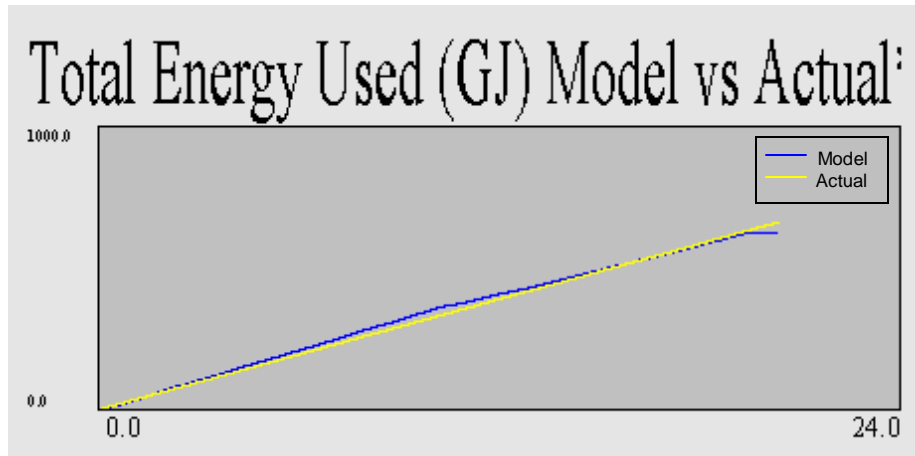


Figure 62: Model energy consumption vs. actual energy consumption in Arena® animation view

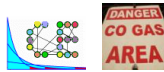
Table 17 shows the average gas consumption at each of the plants. This corresponds with the values shown in Figure 47. Notice the small difference in energy consumption per plant between Table 17 and Figure 47. This is because the values in Figure 47 were calculated from actual feed in 2010, whilst the figures in Table 17 were outputs from the model.

Plant	Average monthly gas energy consumed
URIC	13,269.0 GJ/month
SPP	5,374.6 GJ/month
RPP	2,042.1 GJ/month
Total	20,685.7 GJ/month

Table 17: Average energy usage per plant in the model

8.4 Optimisation with the simulation model

With the simulation model developed, the analyst was in a position to evaluate different scenarios and make recommendations to *Tronox KZN Sands*. Seven scenarios were discussed in section 8.2.1. In section 8.2.2 it was explained that there are actually many more than seven scenarios that needed to be evaluated for a proper understanding of the problem. At this point it is important to reiterate a statement made in section 4.2 on simulation and modelling. This states that

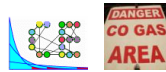


simulation is not an optimisation technique, but rather provides estimates of system performance through modelling. It can thus be used to evaluate different alternatives within the system, acting as a tool for system improvement, but there is *no guarantee that the final improved system is in fact an optimum* (Hillier & Lieberman, 2005).

While the discrete-event simulation exercise allowed a more accurate and meaningful analysis of this stochastic problem than did the existing static models at *Tronox KZN Sands*, it did not, as a matter of course, optimise the problem. The simulation model does not necessarily point the analyst to the set of values of the input variables (DVs) that would lead to the set of optimum output parameters (objectives). To achieve this, an optimisation exercise was needed.

Because this problem has, from the outset, been a problem with multiple output variables, i.e. a MOP, the author originally proposed a “decision factor” that mathematically combined different output parameters to give a dimensionless value that could be used to compare scenarios in an attempt to optimise the problem. The aim of the decision factor was to find the “optimum” scenario, and the associated sets of input and output variables. It was a spontaneous approach to optimising the problem and was developed by the author before he was aware of concepts such as Pareto optimality and techniques such as metaheuristics. Referring back to section 2.4.3.7 where Pareto terminology was discussed, the decision factor approach can arguably be grouped as a technique that *does not make use of the concept of Pareto optimum*. While the decision factor approach is not at all the focus of this chapter, a brief discussion of it has been included to clarify the contrast between that and the more sophisticated and more effective optimisation technique of metaheuristics, particularly the MOO CEM algorithm (Bekker & Aldrich, 2011) discussed Chapter 3.

The decision factor approach is discussed in the next subsection, whilst the application of the MOO CEM algorithm is discussed in the subsection thereafter. The MOO CEM algorithm approach was introduced to the author by his supervisor. As a result of all the DVs in the CO gas problem being discrete in nature, the discrete version of the MOO CEM (section 3.3.2) was applied. At the time the decision was made to apply the MOO CEM algorithm to the CO gas problem, the supervisor was in the process of doctoral research on the suitability of the CEM for optimisation in multi-objective dynamic, stochastic systems. An outcome of the supervisor’s research was developing the MOO CEM algorithm, i.e. adapting the CEM algorithm for MOO. The CO gas problem thus fitted in perfectly as a real-world test problem for the MOO CEM algorithm.



8.4.1 Calculating a “decision factor” for MOO

The decision factor was calculated as follows:

$$Decision\ factor = \frac{f_2}{\sum_i^3 \bar{s}_i}, \quad \dots 48$$

where

f_2 = average monthly methane gas saving (as defined in Table 11),

\bar{s}_i = average number of switchovers per user per day, and

$i = 1,2,3$, represents each of the CO gas users incorporated in the simulation, i.e. the URIC, SPP and RPP.

The output variable, \bar{s}_i , with $i = 1,2,3$, has not been identified as an objective in section 7.3 and is therefore not listed in Table 11. However, it has been identified and included in the list of output variables, as variables 4 to 6 in Table 8.

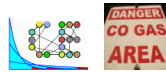
A larger decision factor indicates a better scenario, as the monthly saving on methane gas cost requires maximisation and the number of switchovers per day per plant requires minimisation. For the scenarios that were evaluated in the base case, i.e. the seven scenarios with the input variables shown in Table 15 set to their default values, Table 18 shows the results achieved with the decision factor approach. The symbols shown in the first row, e.g. “110”, represent the seven scenarios initially investigated and is reads follows: “110”, for example, means $DV_1 = 1$, $DV_2 = 1$, $DV_3 = 0$, and similarly for the other symbols.

Scenario	001	010	011	100	101	110	111
Arena® Output							
CostSavingTotalRperMonth	R 68,233	R 295,818	R 409,247	R 779,956	R 826,578	R 911,056	R 893,455
Switchovers per day URIC	0.00	0.00	0.00	0.24	0.41	0.77	1.07
Switchovers per day SPP	0.00	0.06	0.10	0.00	0.00	1.73	2.02
Switchovers per day RPP	0.04	0.00	0.18	0.00	1.01	0.00	1.48
Decision factor	1,651,076	5,176,815	1,507,753	3,287,555	580,263	364,051	195,662

Table 18: Decision factor values for the seven scenarios initially investigated

Notice that in the above evaluation, the replication length was set to 14 days and a simulation run consisted of 10 replications, as discussed in section 8.1.

Figure 63 shows the ranking of the seven scenarios using the decision factor. While it is clear that the scenarios where the saving is the highest are the last three, i.e. “110”, “111”, and “101”, it is also clear that the scenarios that ranked the highest with the decision factor, are in fact



totally different ones, i.e. “010”, “100” and “001”. From Figure 63, the top three scenarios rated on the decision factor are shown in Table 19.

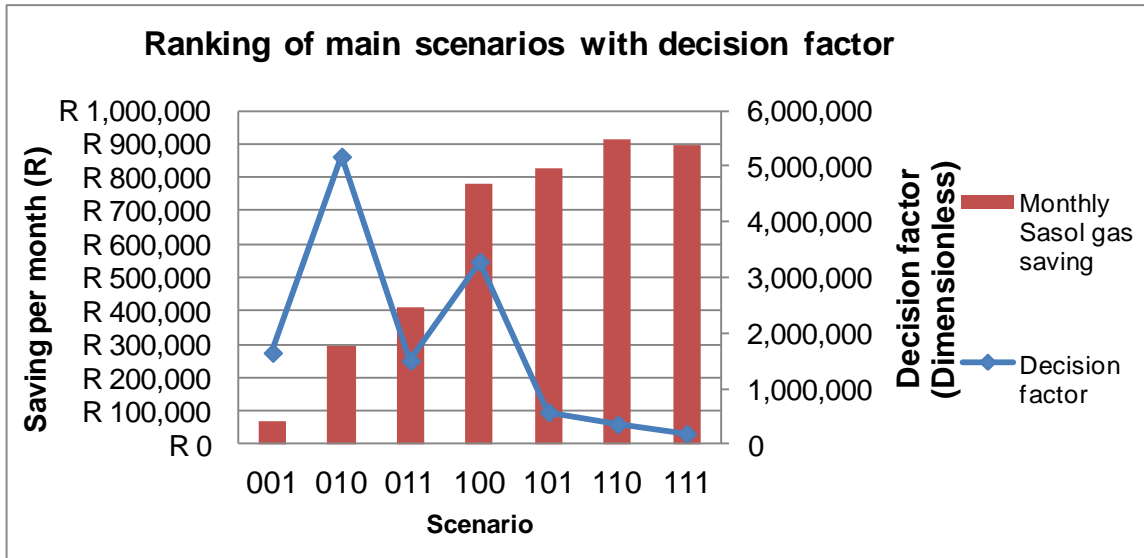


Figure 63: Ranking of the seven initial scenarios using the decision factor value

Top 3 scenarios		
Description	Symbol	Decision factor
CO gas only to SPP	010	5,176,815
CO gas only to URIC	100	3,287,555
CO gas only to RPP	001	1,651,076

Table 19: Top three initial scenarios ranked on the decision factor value

Therefore, although the cost savings are not the highest in the above three scenarios, they are rated top due to the other scenarios being penalised by the number of switchovers per replication associated with each of them.

Although the decision factor is a rather inelegant approach to optimisation, it was found to be fairly helpful as a simple decision making aid. The decision factor combined the important output variables, in their correct relation to each other, into a single number that could be ranked in order to find an “optimum”. This was a very simplified attempt to perform MOO.

When one refers back to the discussion on optimisation techniques in section 2.6, it is clear that a number of different approaches to optimising this simulation problem exist in the OR field. Before continuing to discuss the application of the MOO CEM algorithm, a few reasons why the



decision factor approach is not the most adequate way of optimising a problem like the CO gas problem will be presented.

- **Decision factor easily becomes a meaningless figure**

Although the decision factor can incorporate conflicting output parameters, even measured in different units, into a single dimensionless figure to assist in finding an optimum, the single number can easily become a *meaningless* number that is *difficult to interpret*. This will be the case particularly when the number of objectives becomes greater, or when the magnitude of the output parameters differs significantly.

- **Requires an exhaustive search**

Further, the calculation of the decision factor and finding the optimum set of output variables and the associated input variables do not apply any method for incorporating domain knowledge into the optimisation process. In other words, in order to find the optimum, the decision factor for every single scenario needs to be calculated and then ranked. This becomes computationally expensive when the number of scenarios increases significantly.

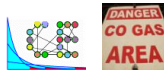
- **No Pareto front**

When trying to combine too many output variables into the decision factor, the analyst is not provided with a trade-off line or surface (*Pareto front*) where the trade-off relationship between output parameters can easily be observed.

8.4.2 Applying a metaheuristic for MOO

When aiming to incorporate more realistic system information into the simulation model to increase the accuracy and significance of the outputs, the input variables shown in [Table 15](#) need to be included. This significantly increases the number of scenarios to be evaluated. The exact increase in the number of scenarios will be discussed next.

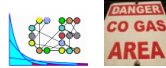
For each combination of potential CO gas users, the *Howden* plant can be operated at a different availability. It seems natural to operate the *Howden* plant at the highest possible availability, but it needs to be kept in mind that in the long term there will be a higher maintenance cost associated with higher *Howden* plant availability. When ranging the *Howden* plant availability between 80% and 100%, with an interval of 0.5%, **40** additional scenarios are added to each of the seven initial scenarios. On a large methane gas cost saving the higher maintenance cost of higher *Howden* plant availability might be insignificant, but the same higher maintenance cost at



the same higher availability could offset a much larger percentage of the methane gas cost saving for the scenario when all CO gas is sent only to a small CO gas consumer such the RPP. This step of incorporating the effect of different availabilities for the *Howden* plant increased the scenarios from seven to **280**.

The next step was to revisit the assumption that each of the CO gas consuming plants is consistently available at 92% (refer to the default values in [Table 15](#)) and thus will require CO gas 92% of the time. When referring back to the detailed conceptual model in [Figure 54](#), the data sources for input variables 12 to 15 in [Table 6](#) and statistical analyses of the input data in [Appendix A](#), it should be clear that the actual MTTF and MTTR of the different plants are more complex than simply a value of, say, 92%. To assume availability (MTTF) of a constant 92% of the time for every plant is a simplifying assumption. To incorporate the statistical distributions into the simulation model would be a more accurate reflection of the runtimes and downtimes of the plants in the model, but it would also complicate the model and would not necessarily assist in finding the Pareto front for f_1 and f_2 . However, taking a narrow range of discrete values (as shown in [Table 10](#)) for the availabilities of the plants and using those as inputs into the model would provide the analyst with an idea of the impact of different plant availabilities on the outputs. As shown in [Table 10](#), each plant has been given a range of three availabilities, being 86%, 90% and 94%. It must be kept in mind that the greater the uptime of a plant, the higher its requirement for gas will be, which increases the likelihood of undersupply situations to a lower priority user. This is especially relevant when either the furnace supply drops or the *Howden* plant is being operated at lower availability. On the other hand, the lower the uptime of a plant, the higher the likelihood of that plant exiting the CO gas distribution network, which would increase the likelihood of lower priority users switching from methane gas to CO gas. Taking this view of the simulation problem adds **27** scenarios to each of the 280 previously defined scenarios. This brings the total number of scenarios to **7 560**.

Instead of manually running the simulation, saving the outputs and then calculating the decision factor for all 7 560 scenarios to determine the best combination of consumers, the MOO CEM algorithm was used to automatically rerun the simulation to search for the Pareto front. Even if the decision factor approach was discarded and an automated exhaustive run of the problem was performed in order to find and rank the output parameters in some way, considerable computation time would be required. With each scenario taking 0.32 minutes of simulation time to execute, an exhaustive run of the CO gas problem would take 40.32 hours to execute. The reasonably long computation time required to evaluate the entire population is exactly the reason for applying the metaheuristic. One of the strengths of metaheuristics are to search “intelligently” through the search space to find the Pareto front in minimum computational time ([Luke, 2009](#)).



8.5 Programming the MOO CEM algorithm and finding the Pareto front

As discussed in Chapter 4, the simulation model of the CO gas problem was implemented in the simulation package, *Arena*® (Kelton *et al.*, 2007). Algorithm 4 was implemented in *Matlab*® code, with the code calling the *Arena*® simulation model to automate the search for the Pareto front and the associated Pareto optimal set of input variables. The logic of the simulation model, in *Microsoft*® *Visio* flow diagram format, can be viewed in Appendix B, while screen prints from *Arena*® of the model logic and animation view can be viewed in Appendix C. The *Matlab*® code for the MOO CEM algorithm is available on the CD-ROM submitted with the thesis.

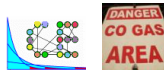
The MOO CEM algorithm samples possible binary values for each of the different CO gas users (DV_1 , DV_2 , DV_3) indicating whether each is set to receive CO gas (a value of 1 is given), or not (a value of 0 is given) in a specific scenario (as illustrated in Table 14). The algorithm also samples values within the defined ranges for the *Howden* availability (DV_4) and each of the plant availabilities (DV_5 , DV_6 and DV_7).

It should be noted that when Algorithm 4 is implemented for the discrete case, it is slightly differently interpreted and applied in the following ways:

The histogram concept that is implemented for each DV_i using the class vector C_i (section 3.6) in the continuous case, is represented by a probability vector, \mathbf{P} , in the discrete case. \mathbf{P} is essentially a probability mass function, represented by in a probability vector $\mathbf{P}_{t-1,i}$ after iteration t ($t > 0$) for DV_i (Algorithm 2). Instead of updating the frequencies of the histogram for a continuous DV by applying $c_{i\kappa} \leq X_{ij} < c_{i(\kappa+1)}$, $1 \leq \kappa \leq r + 2$, the probabilities of $\mathbf{P}_{t-1,i}$ is updated with the frequency count per class of the probability mass function from the values of the DVs in the most recent **Elite** vector. In preparation for the next iteration of the algorithm, the new population of possible solutions is created proportionally according to the class frequencies (updated probabilities in $\mathbf{P}_{t-1,i}$) for each DV.

In the discrete case, no probability p_h for histogram inversion is used and the number of classes is not increased as the search continues, because the classes are fixed according to the allowed increments of the discrete variables. However, the discrete case also applies the smoothing rule of (30) and (41), by applying α to smooth the probabilities in $\mathbf{P}_{t-1,i}$ and ensures exploration and exploitation by decreasing the threshold ρ_E in the Pareto ranking algorithm (Algorithm 3).

The *Matlab*® code calls the *Arena*® model for evaluation of the objective functions. Thus each combination of input variables in the population is evaluated via simulation. Note that



feasible ranges were set for the DVs as shown in (Table 10), resulting in a discrete solution space of 7 560 possibilities as discussed in the previous section.

The aim of the MOO CEM algorithm was to find an approximate Pareto front for objectives f_1 and f_2 . As discussed in 8.1, each combination of input variables in the generated population of possible solutions was evaluated using 14 simulation replications, each with a replication length of seven days. A further aim was to minimise the simulation runtime. It was more important to obtain an approximate Pareto front showing a trend than to find near-accurate estimations of the objectives. Once a front was obtained, the objectives could be evaluated in more depth using the values of the **Elite** vector.

Remember that no warm-up period was included in the simulation model as was explained in section 7.2.

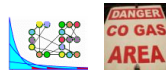
The exhaustive enumeration would yield the *true Pareto front* (PF^*_{true} as defined and discussed in section 2.4.3.7 on *Pareto optimality*), which could be used as a reference Pareto front for those found by the MOO CEM algorithm. In searching for the true front the DVs were evaluated according to their allowed increments, thus exploring the entire solution space. The resulting true front for the CO gas problem comprised 22 points and is shown in Figure 64, together with an approximation front (13 points) found by the MOO CEM. The latter will be discussed in the next section.

The settings for the MOO CEM algorithm (discrete case) were as follows: $N = 20$, $\varepsilon = 0.0001$, $\alpha = 0.6$ and, $p_h = 2$ and the number of main loops was set at 10. The random number generator started at the same seed number for each evaluation. The results achieved with the MOO CEM algorithm are discussed in the next section.

8.6 Results achieved with the MOO CEM algorithm

In section 8.4, two different ways were discussed in which the simulation outputs were processed with the aim of optimisation. The first way (section 8.4.1) was to combine the most important output variables into a single number, called the “decision factor”, to simplify and aid decision making. The results obtained by this approach were shown in Table 18. The decision factor approach was helpful, but far from ideal, hence the second, and preferred, approach was to apply the MOO CEM algorithm. The results achieved with MOO CEM are shown and discussed below.

When performing an exhaustive enumeration requiring evaluation of all 7560 scenarios, the computation time was 40.32 hours, and this exercise yielded the true Pareto front. By applying the MOO CEM algorithm the approximated Pareto front was obtained after 240 evaluations



(scenarios), which is about 3% of the decision space. This decreased the computation time from 40.32 hours to 1.28 hours. The approximated front seems to fit well on the front obtained via the exhaustive search. The shapes of the fronts are similar and the MOO CEM-estimated front will afford the decision maker the same information as the true Pareto front, or “Pareto front known”. The values of the MOO CEM-estimated Pareto front and the values found with the exhaustive run are shown in Figure 65. Depending on the values of N, ϵ, α, p_h and the number of outer loops, j , in the MOO CEM algorithm, the metaheuristic could converge at different rates, with different levels of accuracy, to the approximate Pareto front.

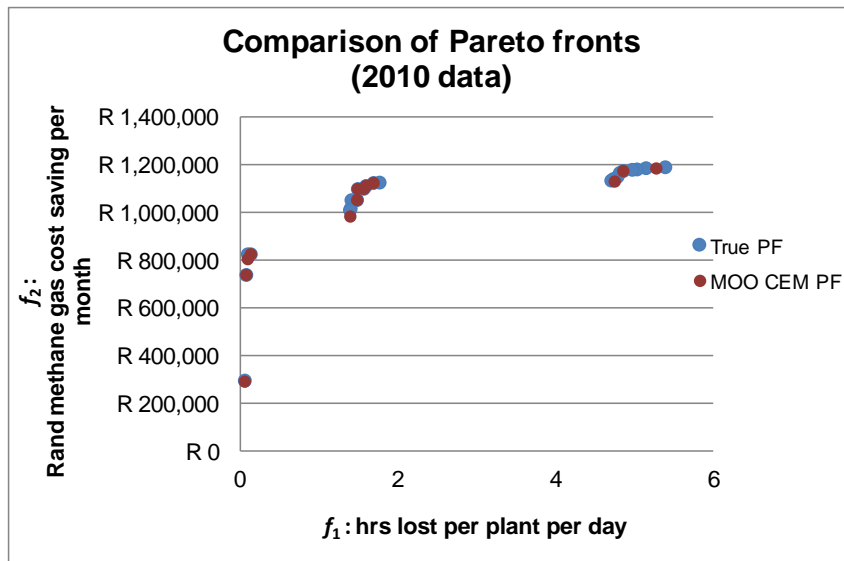


Figure 64: Comparison of the true Pareto front with the MOO CEM-estimated Pareto front

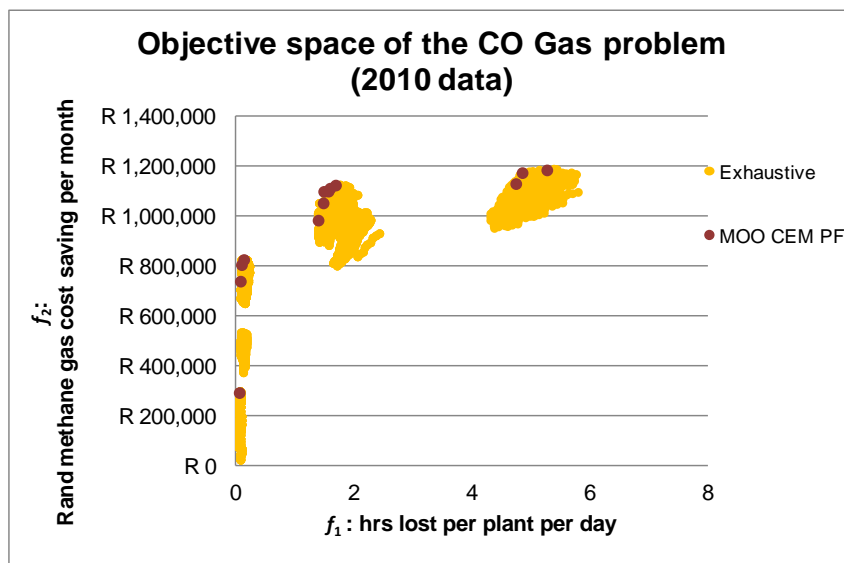
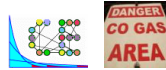


Figure 65: Exhaustive enumeration of CO gas problem with MOO CEM-estimated Pareto front



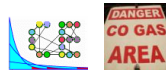
Each data point in Figure 65 indicates a combination of the objectives (f_{1i}, f_{2i}), associated with a certain combination of combination of DVs (DV_{1i}, \dots, DV_{7i}), where $1 \leq i \leq 7560$. The clusters of points observed in the graph are due to step changes in lost production time associated with different combinations of CO gas users. The *Howden* plant and other plants' availabilities (DV_4 to DV_7) seemed to affect the average monthly methane gas saving quite significantly, i.e. availabilities affected f_2 more than they affected f_1 . Although the different combinations of CO gas users caused the clustering points in the objective space, they also affected both f_1 and f_2 significantly. The true Pareto front lies somewhere on the top left edge of these clusters of points, where the hours lost are minimised and the cost saving are maximised.

From the two graphs in Figure 64 and Figure 65, it can be inferred that the MOO CEM algorithm was effectively applied to obtain the Pareto front for the CO gas problem. Not only was the correct form found (when compared with the true Pareto front), but the approximate Pareto front was also obtained in just 3% of the time required to find the true front. This is a significant reduction in computation time. From these conclusions it can be said that the first research aim, identified in section 1.4, was achieved.

It was found that the results from the decision factor approach and those from the MOO CEM algorithm were difficult to compare. Yet, in an attempt to compare the results, it was found that the decision factor tells the analyst that high monthly savings, in fact, were associated with a low decision factor value. If considering only the decision factor value, the decision factor approach rated the scenarios shown in Table 19 as the optimum ones. No explicit indication is given of what the practical implication (i.e. the hours lost) of the more poorly rated scenarios (with the high monthly savings) were. It simply rated those scenarios low. The metaheuristic, though, provided a more comprehensive answer by 1) indicating more sets of optimum scenarios and 2) making the trade-off of f_1 vs. f_2 for different points on the Pareto front clearly visible to the analyst.

When using the Pareto front for decision making, the analyst or decision maker will have to select the *most desirable scenario* of those *optimum* or *non-dominated scenarios* shown in the Pareto optimal set.

Table 20 shows the values of the true Pareto front, as well as its associated Pareto optimal set, obtained for the CO gas problem. Instances where values on the approximated front matched values on the true front exactly have been highlighted in yellow. Table 21 shows the Pareto optimal set associated with the MOO CEM-approximated Pareto front.



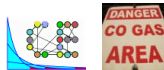
Pareto optimal set of true Pareto front										True Pareto front		
Nr	DV ₁	DV ₂	DV ₃	DV ₄	DV ₅	DV ₆	DV ₇	f ₁	f ₂	Rank		
1	1	1	1	0.945	0.94	0.94	0.94	5.381	1,190,942	0		
2	1	1	1	0.945	0.94	0.9	0.94	5.137	1,186,977	0		
3	1	1	1	0.945	0.94	0.9	0.90	5.022	1,181,788	0		
4	1	1	1	0.945	0.94	0.86	0.94	4.962	1,180,183	0		
5	1	1	1	0.945	0.94	0.86	0.90	4.848	1,174,995	0		
6	1	1	1	0.945	0.94	0.86	0.86	4.803	1,166,960	0		
7	1	1	1	0.92	0.94	0.86	0.86	4.775	1,150,801	0		
8	1	1	1	0.91	0.94	0.86	0.86	4.724	1,141,527	0		
9	1	1	1	0.945	0.86	0.94	0.90	4.695	1,134,978	0		
10	1	1	0	0.955	0.94	0.94	0.94	1.757	1,126,747	0		
11	1	1	0	0.97	0.94	0.94	0.94	1.681	1,125,109	0		
12	1	1	0	0.97	0.94	0.9	0.94	1.586	1,112,618	0		
13	1	1	0	0.945	0.94	0.86	0.94	1.557	1,100,714	0		
14	1	1	0	0.97	0.94	0.86	0.94	1.481	1,099,853	0		
15	1	1	0	0.97	0.86	0.94	0.94	1.471	1,053,565	0		
16	1	1	0	0.885	0.94	0.86	0.94	1.400	1,052,995	0		
17	1	1	0	0.885	0.9	0.86	0.94	1.390	1,017,992	0		
18	1	1	0	0.885	0.86	0.94	0.94	1.381	1,010,983	0		
19	1	0	0	0.94	0.94	0.94	0.94	0.124	826,615	0		
20	1	0	0	0.96	0.94	0.94	0.94	0.086	826,534	0		
21	1	0	0	0.825	0.94	0.94	0.94	0.067	739,997	0		
22	0	1	0	0.93	0.94	0.94	0.94	0.048	297,926	0		

Table 20: Values of the true Pareto front and associated Pareto optimal set

Pareto optimal set of MOO CEM Pareto front								MOO CEM Pareto front		
Nr	DV ₁	DV ₂	DV ₃	DV ₄	DV ₅	DV ₆	DV ₇	f ₁	f ₂	Rank
1	1	1	1	0.945	0.94	0.94	0.90	5.267	1,185,872	0
2	1	1	1	0.945	0.94	0.86	0.90	4.848	1,174,995	0
3	1	1	1	0.945	0.90	0.86	0.86	4.740	1,131,085	0
4	1	1	0	0.970	0.94	0.94	0.94	1.681	1,125,109	0
5	1	1	0	0.970	0.94	0.90	0.90	1.586	1,112,618	0
6	1	1	0	0.945	0.94	0.86	0.94	1.557	1,100,714	0
7	1	1	0	0.970	0.94	0.86	0.90	1.481	1,099,853	0
8	1	1	0	0.970	0.86	0.94	0.90	1.471	1,053,565	0
9	1	1	0	0.825	0.94	0.86	0.90	1.386	984,602	0
10	1	0	0	0.940	0.94	0.94	0.94	0.124	826,615	0
11	1	0	0	0.900	0.94	0.90	0.94	0.086	806,234	0
12	1	0	0	0.825	0.94	0.90	0.94	0.067	739,997	0
13	0	1	0	0.945	0.94	0.94	0.94	0.048	293,498	0

Table 21: MOO CEM-estimated Pareto front values and associated Pareto optimal set

The approximated Pareto front should be interpreted as follows: for example, the decision maker can decide that a minimum of R 1 000 000 per month needs to be saved. This will require that the CO gas is distributed to at least the URIC and the SPP. The decision maker will also have to accept that the operational implication of the decision is that a minimum of roughly 1.47



hours of production time will be lost at each plant per day due to switchovers. It would be prudent to warn operational personnel to expect these losses in production time, in order for them to be prepared to handle them effectively. Operational personnel will have to accept the production disruptions that the decision implies, which include the sending of personnel to manually purge lines in order to switch between CO and methane gas when an undersupply of CO gas occurs. For implementing this decision the *Howden* Plant needs to be maintained, running at 97% availability. The URIC should be operating at 86%, the SPP at 94% and the RPP at 90% availability. The scenario on the approximated Pareto front that corresponds best to the highest rating of the decision factor is that in which the CO gas is distributed only to the SPP, while keeping the availabilities of the *Howden* Plant and other three plants as shown in the last row of [Table 21](#). This is also a feasible scenario and also a non-dominated scenario which would imply a monthly saving of only R 293 498, but where the time lost per day is negligible, at 0.048 hours.

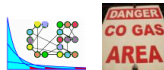
Note that the Pareto fronts shown in [Figure 64](#) and [Figure 65](#) were obtained when running the simulation with 2010 input data. When using 2012 data, the form of the fronts change in favour of the business in that the expected monthly saving increases and the expected hours lost per plant per day decreases significantly. This positive change is mainly due to much improved furnace performance from 2010 to 2012.

Both the decision factor and metaheuristic methods provided the decision maker with a means of selecting an optimum scenario, but MOO CEM (apart from considering a vastly wider number of realistically possible scenarios) gave the decision maker more insight into what the decision implied. In other words the MOO CEM algorithm showed clearly how the benefit of a higher cost saving is penalised by the operational disruptions of more switchovers.

It can thus be concluded that the application of MOO CEM allowed the analyst to find a good Pareto front which presented him or her with the near Pareto optimal set for the CO gas problem. The latter is significant in that it allows the decision maker to now make *a very good and quantitatively justifiable decision* about a problem that initially seemed daunting and complicated, due to its dynamic, stochastic nature. It is important to notice that even though the MOO CEM algorithm finds the Pareto front, the *discretion of the decision maker is still required* to select a point on the Pareto front, whilst, understanding the impact thereof and requirements necessary to achieve that scenario. An element of subjectivity therefore remains part of the decision making.

8.7 Summary: Chapter 8

With the work done, as described in Chapters 6 and 7, to formulate the problem, define the simulation experiment and translate the concept model into a computer model, just about all was set to perform the simulation production runs. The two remaining tasks were: 1) to perform a



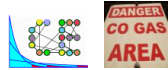
preliminary simulation run to determine the number of replications required per simulation run, in order to allow the analyst to draw statistically sound conclusions from the results; and 2) to ensure that the model had been verified and validated.

This chapter started by describing the process followed to determine the number of replications required. It was found that for simulation results within the desired confidence interval half-widths, 14 replications per simulation run were required, each with a replication length of seven days. A full simulation run took 0.32 minutes to complete.

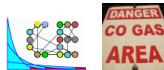
The model was verified and validated through continuous experimentation with different test scenarios. Ultimately, the model was validated by ensuring that the gas energy consumption of the model corresponded with the gas energy consumption of the real system over the same period of time.

Subsequently the production runs could be performed. Initially, only a selected few runs were performed for certain handpicked scenarios. Initially, seven scenarios were identified and tested on the simulation model. Finding the best amongst the outputs obtained from these handpicked scenarios required the analyst, for the first time, to apply some form of optimisation. A unary indicator called the “decision factor” was calculated manually for each scenario in order to rank the scenarios whilst incorporating the different conflicting outputs. The decision factor proved helpful for optimisation to an extent, but the result was not particularly meaningful. It was clear that to evaluate the problem in full, by incorporating all the DVs to the full extent of the allowed range of each, significantly more scenarios had to be tested and optimised. This would be impractical to perform manually. For this purpose the MOO CEM algorithm was applied. The algorithm would call the simulation model repeatedly until an optimum set of solutions had been found. In this way MOO CEM-approximated Pareto front could be obtained and compared with the true Pareto front which could be determined from an exhaustive enumeration of the scenarios.

The optimisation exercise yielded 1) that an approximated Pareto front could be obtained for the CO gas problem and 2) that the latter front corresponded well with the true Pareto front. It was also encouraging to find that the MOO CEM algorithm obtained the approximated Pareto front in only 3% of the observations required to obtain the true front. It was interesting to note that the top ranked scenario according to the decision factor was also found on the true Pareto front, although not too much should be made of the latter observation, as the decision factor did not consider all the DVs that the MOO CEM algorithm did. Changing some of the DVs not explicitly incorporated by the decision factor might deliver results that would not correspond with the MOO CEM algorithm results. However, for further work and final conclusions, only the results of the MOO CEM algorithm were considered.



With the MOO CEM algorithm's results obtained and archived, a last task remained in this research project which was to compare the MOO CEM algorithm with a benchmark algorithm suitable for optimising a dynamic, stochastic MOP. Both MOO CEM and the benchmark algorithm will be applied to the CO gas problem. The next chapter will discuss the comparative exercise.



9. A comparison of MOO CEM with the NSGA II

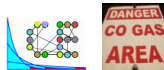
In the previous chapter the execution of the production runs and the application of the MOO CEM algorithm for optimisation were discussed. The chapter included the results obtained by applying the metaheuristic, as well as results found with a “decision factor”, the latter being a simplified unary indicator to assist with the ranking of solutions. The previous chapter made it clear that the metaheuristic was a much better means of approaching the optimisation of a problem such as the CO gas problem compared to the decision factor approach, as the latter does not make use of the concept of Pareto optimality. In this chapter, no further reference will be made to the results obtained with the decision factor.

This chapter will discuss the idea of measuring the performance of MOO algorithms. It follows naturally that if the approximated Pareto front corresponds well with the true Pareto front and is obtained in less time (fewer observations) than the time it takes to find the true front (assuming the true front can be found at all), that using the optimisation algorithm is a better option than an exhaustive enumeration. Yet, for more specific information, it needs to be established how good and how much better the optimisation algorithm is.

In this chapter various different performance indicators for MOO algorithms will be discussed. This is followed by showing the results obtained when selected performance indicators are calculated for the MOO CEM algorithm, when the latter is applied to the research problem specifically. Finally, this chapter compares the performance of the MOO CEM algorithm, as applied to the research problem, with that of the MOO genetic algorithm (GA) of *Matlab*®, when the MOO GA is also applied to the research problem. The MOO GA of *Matlab*® is a commercial product based on the NSGA II of [Deb et al. \(2002\)](#) and the NSGA II is often used as a benchmark in the literature, when algorithms are compared.

9.1 Performance indicators when assessing MOO algorithm performance

From the literature it is clear that several quality performance indicators for MOO algorithms exist. These indicators normally consider some estimation of the deviation between the approximate front found by the MOO algorithm (\mathcal{PF}_{test}^*) and the true Pareto front (\mathcal{PF}_{true}^*) of a benchmark problem. [Bekker \(2012\)](#) explains why the term *quality performance* should be used instead of the general term *performance* (which pertains to both time and quality measurements) in comparative studies where the time quality indicator is implicitly assessed by limiting the number of objective evaluations. In the evaluation of the suitability of the MOO CEM algorithm for optimising the CO gas problem, however, no limitation was placed on the number of objective evaluations (except



when comparing the MOO CEM algorithm with the MOO GA of *Matlab*®) hence the author has preferred the term *performance* instead of *quality performance*. Further, the term *performance indicator* is used in this thesis, while terms like “performance measure” and “performance metric” are used in some of the literature. Knowles *et al.* (2006) argue that the term “indicator” is more correct, since “measure” and “metric” have specific mathematical meanings.

Knowles *et al.* (2006) propose two approaches that can be followed to quantify quality of performance: 1) the *indicator approach*, which quantifies the result of an algorithm run as a numerical value and 2) the *attainment approach* that models the outcome as a probability density function. The indicators assume unary numerical values, which often have some subjectivity included. The indicators are used mainly to obtain the *quality of convergence* and *of diversity*. This involves the concepts of *proximity* and *diversity*, which refer to the fact that it is expected of an algorithm to properly *explore* and *exploit* the solution space. This must be done in order to obtain a Pareto approximation set that is *close to the true front*, and that it is *well populated with solutions*. Laumanns *et al.* (2002) developed the concept of ε -dominance and developed updating strategies for iterative searches that allow for the desired convergence and distribution of solutions. It is often mentioned amongst the leading researchers in the field of MOO that “finding a close and dense Pareto front is in itself a multi-objective problem”. Yet good proximity and diversity is necessary for confidence in the approximation solution set. Put differently, a good approximation set must be close to or, even better, coincide with the true front, while it must also extend uniformly and be well populated over the front (Purshouse & Fleming, 2007). The reader is referred to (Zitzler *et al.*, 2002; Zitzler *et al.*, 2003) for comprehensive discussions and theoretical depth of performance or quality assessment of MOO algorithms.

From the different unary performance indicators discussed in Knowles *et al.* (2006), three were selected and applied in this study. These indicators are listed below, as are the expressions to calculate each. For the expressions below, let $\mathcal{P}\mathcal{F}^*_{test} = \mathcal{P}_K$ and $\mathcal{P}\mathcal{F}^*_{true} = \mathcal{P}_T$.

1. *Generation distance (GD)*, which measures the average distance between \mathcal{P}_K and \mathcal{P}_T . It is defined as

$$GD \triangleq \frac{(\sum_{i=1}^{|\mathcal{P}_K|} d_i^2)^{\frac{1}{2}}}{|\mathcal{P}_K|}, \quad \dots 49$$

with d_i the Euclidean distance between solution value i of \mathcal{P}_K and the closest member in \mathcal{P}_T to the solution i . When $\mathcal{P}_K = \mathcal{P}_T$, then $GD = 0$.



2. *Spacing (SP)*, which numerically describes the spread of the vectors in \mathcal{F}_K . It is defined as

$$SP \triangleq \sqrt{\frac{1}{|\mathcal{F}_K| - 1} \sum_{i=1}^{|\mathcal{F}_K|} (\bar{d} - d_i)^2} \quad \dots 50$$

and

$$d_i = \min_{k=1}^K |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})|, \quad \dots 51$$

with $i, j = 1, \dots, |\mathcal{F}_K|$, K the number of objectives, and \bar{d} the mean of all d_i . The members of the approximation front are equally spaced if $SP = 0$. The true Pareto front is not required for this test measure.

3. *Maximum Pareto front error (ME)*, which measures how well two vector sets conform in terms of shape and distance apart. It is determined as follows:

$$ME \triangleq \max_j \left\{ \min_i \left(\sum_{k=1}^M |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})|^2 \right)^{1/2} \right\} \quad \dots 52$$

More indicators are listed in [Knowles et al. \(2006, Appendix A\)](#). It is important to know that performance indicators do have limitations and [Bekker \(2012\)](#) briefly discusses some of these. It is recommended that the approaches recommended in [Knowles et al. \(2006\)](#) should be followed when comparing the performance or performance quality of two or more algorithms.

9.2 MOO CEM performance when applied to the CO gas problem

While the previous section introduced the notion of performance indicators for MOO algorithms and listed the indicators that have been used in this study, this section shows the calculated values of each of the indicators selected for the MOO CEM algorithm when applied to the CO gas problem.

While [Zhou et al. \(2011\)](#) show that there are many accepted quality indicators of a unary nature available, the *quality performance* of the MOO CEM algorithm for the CO gas problem has been evaluated using the quality indicators defined in the previous section. The indicator values obtained are shown in [Table 22](#). The computation time of 1.28 hours to obtain the approximate



Pareto front has been mentioned in section 8.6 and is also shown. By combining the quality performance and the computation time, the algorithm's *performance* can be stated. The true Pareto front and the approximate front generated by the MOO CEM algorithm for the various problems were shown in Figure 64.

<i>GD</i>	<i>SP</i>	<i>ME</i>	Execution time (min)	Size of Elite
1092.818	4509.521	44.156	77	22

Table 22: Quality indicators calculated for the MOO CEM algorithm in the CO gas problem

The values of the three quality indicators in Table 22 can be interpreted as relatively small compared to the scale of the axes of Figure 64, which is an indication of good correlation with the true Pareto front. It has been shown, in section 8.6, that the execution time for finding the approximated front is only 3% of the time required to obtain the true front. It has also been shown in the same section that the values on the approximated front were 13, compared with the 22 in the **Elite** set of the true front.

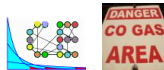
9.3 Comparative assessment of MOO CEM with a benchmark MOO algorithm

As mentioned at the beginning of this chapter, the MOO GA of *Matlab*® was selected for comparison with the MOO CEM algorithm when both are applied to the CO gas problem. The *Matlab*® MOO GA is commercially available and based on the NSGA II algorithm (Deb *et al.*, 2002), which is often used as a benchmark in algorithm-comparative studies. The subsections to follow will introduce the notion of algorithm assessment, discuss quality indicators suitable for algorithm assessment, guide the reader through the assessment experiment and discuss the results obtained for the assessment.

9.3.1 Background to algorithm assessment

Before the comparative assessment is discussed, the following needs clarification:

1. Much of the comparative assessment will be based on the tutorial by Knowles *et al.* (2006). The latter suggests assessment methods and quality indicators. Only essential aspects of the tutorial will be explicitly shown again in this section.



2. Where sections 9.1 and 9.2 used the broader term *algorithm performance* (pertaining to *time* and *quality*) when assessing only how well the MOO CEM algorithm was able to solve the CO gas problem, this section will use the narrower term *quality performance* in the comparative assessment. This is because the execution time indicator is *implicitly* assessed by limiting the number of objective evaluations in the comparative assessment.

The reader is reminded that the solutions produced by the two MOO algorithms are approximations of the true Pareto front of the optimisation problem. These are referred to as "approximation sets". The quality assessment will be done in the objective (or solution) space. For assessment, these approximation sets will be compared, while considering the true Pareto front.

Since both algorithms contain stochastic elements and also operate on a stochastic problem, the resulting Pareto approximation sets will vary from experiment to experiment. Importantly, any quality statements made based on experimental assessments are therefore probabilistic in nature.

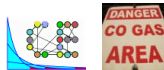
There is no "best" performance assessment technique, state Knowles *et al.* (2006). Three approaches to algorithm performance assessment, which are specifically applicable when algorithms are compared, are discussed in detail in the cited tutorial. These approaches are:

- The *attainment function approach*, where the result of an optimisation algorithm is represented as a probability density function
- The *indicator approach*, where the result of an algorithm is summarised in one or more quantitative indicators (similar to the discussion in section 9.1 and illustration in 9.2)
- The *dominance-based ranking approach*, where pooling and ranking of the many approximation sets created by the algorithms are done. The set of ranks is then used to determine whether there is a significant statistical difference between the rank distributions of the output of the algorithms.

The experiment discussed in this section will use the *indicator approach*.

9.3.2 Quality indicators for algorithm assessment

A selected few quality indicators were discussed in section 9.1 while performance quality values for the MOO CEM algorithm as applied to the CO gas problem, were presented in Table 22. Although the quality indicators discussed thus far give good indications of the quality performance of an algorithm, it has been showed by Knowles *et al.* (2006) that these should not be used when comparing algorithms. The same authors state that a "good" quality indicator is *Pareto compliant*.



This means that if an approximation set X is preferable to Y , the quality indicator value for X should be at least as good as the indicator value for Y , in respect of weak Pareto dominance. Formally, it is stated as: The indicator $I_q : \Omega_q \rightarrow \mathbb{R}$, where \mathbb{R} is the set of real numbers, is *Pareto compliant*, if and only if, for all $X, Y \in \Omega_q : X \preceq Y \Rightarrow I_q(X) \leq I_q(Y)$, assuming that minimisation of indicator values results in higher quality. Once values for the indicators have been calculated, the results produced by different MOO algorithms can be compared by comparing the corresponding indicator values.

In this comparative experiment the *hypervolume* and the *epsilon* indicators will be used, as recommended by Knowles *et al.* (2006). Both of these indicators are Pareto compliant and can be estimated with reasonable ease. It is important to note that the quality indicators in Table 22 are all Pareto non-compliant.

The hypervolume indicator I_H must be maximised. This indicator measures the part of objective space that is weakly dominated by an approximation set X . Note that in this experiment, I_H will be referred to as a “hyperarea”, since the CO gas problem has only two objectives, f_1 and f_2 (defined in section 7.3.1.2 and shown in Table 11). To calculate the hyperarea a strictly dominated reference point has to be provided. Figure 66 shows a simple example of a hyperarea with a minimisation Pareto set $\{(3; 6); (6; 4.5); (9; 1.5)\}$, $I_H = 58.5$ and the reference point at (12; 10.5). It is not difficult to understand that I_H measures *diversity* (spread) and *proximity* (closeness) of the approximation set (\mathcal{P}_K) to the true Pareto front (\mathcal{P}_T).

The unary epsilon indicator is the second indicator to consider. It has a multiplicative and an additive version. The additive version, $I_{\varepsilon+}$, was used in this study. It is defined as follows, with respect to solution set X and \mathcal{P}_R as reference set:

$$I_{\varepsilon+}(X) = I_{\varepsilon+}(X, \mathcal{P}_R) \quad \dots 53$$

The epsilon indicator is interpreted as the minimum value by which each coordinate in the approximation set X must be adjusted so that this set X will dominate the reference set \mathcal{P}_R . Both versions of the epsilon indicator must be minimised.

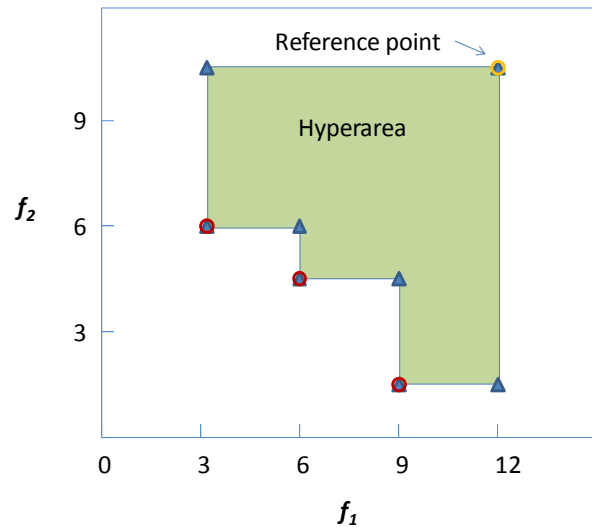


Figure 66: Example of a simple hyperarea and reference point

9.3.3 Performing the assessment

First, approximation sets needed to be generated with each algorithm, from which quality indicator values could be calculated. Using standard statistical procedures the indicator values could be compared, because both the algorithms and the assessment problem have stochastic elements. It is popular practice amongst researchers to use non-parametric statistical tests to compare the output of algorithms, since no assumption of the underlying output distributions is made. Igel *et al.* (2007) suggest, for example, the Mann-Whitney-Wilcoxon rank-sum (MWW) test.

One thousand observations per sample were made in this experiment and the standard two-sample t-test was used to test the hypotheses. The latter test allows for a statement regarding the *direction of difference* between the two samples; in other words, it allows the analyst to conclude that the mean of sample *A* is equal/less/greater than the mean of sample *B*.

The basic null hypothesis of the standard two-sample t-test states that data in two test sets are independent random samples from normal distributions with equal means and equal but unknown variances, against the alternative hypothesis stating that the means are not equal. A right-tailed test was performed on the hyperarea output, whilst the epsilon output was evaluated with a left-tailed test.

The test statement for this experiment, therefore, was: *the data in the two test sets are independent random samples from normal distributions with equal means and equal but unknown variances; the alternative being that the mean due to the MOO CEM algorithm is greater than the mean of the Matlab® MOO GA* (when considering the hyperarea indicator).



A significance level of 5% was used in the hypothesis tests. The hypotheses are formulated as follows:

1. Hyperarea case:

Let μ_{CI_H} be the mean of the approximation sets produced by the MOO CEM algorithm and that of the *Matlab*® set μ_{MI_H} , then the one-sided right-tail hypothesis for assessment is stated as:

$$H_0 : \mu_{CI_H} \leq \mu_{MI_H}$$

$$H_1 : \mu_{CI_H} > \mu_{MI_H}$$

2. Epsilon indicator case:

Let $\mu_{CI_{\epsilon^+}}$ be the mean of the approximation sets produced by the MOO CEM algorithm and that of the *Matlab*® set $\mu_{MI_{\epsilon^+}}$, then the one-sided left-tail hypothesis for assessment is stated as:

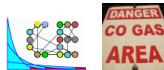
$$H_0 : \mu_{CI_{\epsilon^+}} \geq \mu_{MI_{\epsilon^+}}$$

$$H_1 : \mu_{CI_{\epsilon^+}} < \mu_{MI_{\epsilon^+}}$$

In order to conclude that MOO CEM outperformed the *Matlab*® MOO GA, both null hypotheses needed to have been rejected in favour of the alternative hypotheses. Additionally, in an attempt to find further supporting evidence of the research hypothesis, the number of evaluations was limited for both algorithms. The archives developed during production runs (refer to section 8.2.3) were regarded as the populations of solutions. For the CO gas problem, the maximum generation number was set to $\lfloor N_a / (2N) \rfloor$, with N_a the archive size, and $N = 25$. This limited the maximum number of generations to 150, which is even lower than 240 generations that were required to find the approximate Pareto front shown in Figure 64. When applying the *Matlab*® MOO GA, the Pareto fraction was set to one, while the default settings were otherwise accepted. The test procedure followed is shown next.

Test procedure for algorithm comparison:

1. Run the MOO CEM algorithm for 1 000 pseudo-independent replications, allowing the maximum number of objective function evaluations per replication.
2. Observe the values of the unary Pareto non-compliant quality indicators, as well as the hyperarea and the epsilon indicator for each replication.



3. Repeat steps 1 and 2 using the MOO GA in the Optimisation Toolbox of *Matlab*®, while allowing the same maximum number of objective function evaluations as allowed with MOO CEM.
4. Calculate the means and confidence interval half-widths for the three unary quality indicators *SP*, *GD* and *ME*.
5. Conduct the standard two-sample t-test (`ttest2` in *Matlab*®), the right-tailed on the hyperarea set and the left-tailed on the epsilon indicator set, whilst recording the outcomes of the respective hypothesis tests.
6. Interpret the hypothesis tests' outcomes as follows: the algorithm that produces the largest significant hyperarea can be considered superior, while the algorithm producing the smallest epsilon indicator can be considered superior.
7. Do a box-whisker plot with notches for each indicator.

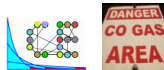
9.3.4 Results of the algorithm assessment

Table 23 shows the mean values of the Pareto non-compliant unary quality indicators. Although the indicator values in Table 23 are lower for the *Matlab*® algorithm, these do not indicate the latter algorithm's superiority, as they are not Pareto-compliant indicators and thus not suitable for algorithm comparison.

	<i>GD</i>	<i>SP</i>	<i>ME</i>
MOO CEM	1092.82	4509.52	44.16
<i>Matlab</i> ® MOO GA	198.63	675.45	5.76

Table 23: Pareto non-compliant unary quality indicator values

The means and 95% confidence interval half-widths h_w of the hyperarea (*HA*) and epsilon (*Eps*) quality indicators for both algorithms are displayed in Table 24. The same table also shows the true (reference) hyperarea, which was calculated using the true Pareto front, as well as the search space size and the number of generations of the *Matlab*® MOO GA. The results of the hypothesis tests using the hyperarea indicator are shown in Table 25. Figure 67 shows the box-whisker plot for the CO gas problem, in further support of the tabled results.



	Reference <i>HA</i>	Mean <i>HA</i>	<i>h_w</i>	Mean <i>Eps</i>	<i>h_w</i>	Search space size	Number of generations allowed
MOO CEM	9,743,602	9,486,125	15,262	-	-	7,560	150
Matlab® MOO GA		8,083,320	165,405	-	-		

Table 24: Mean values and 95% confidence interval half-widths for the hyperarea and epsilon indicator

Hyperarea				
p-value	CI Lower	CI Upper	t-statistic	Outcome
0	1 263 445	∞	16.57	Reject

Table 25: Outcome of the hypothesis test for the hyperarea indicator

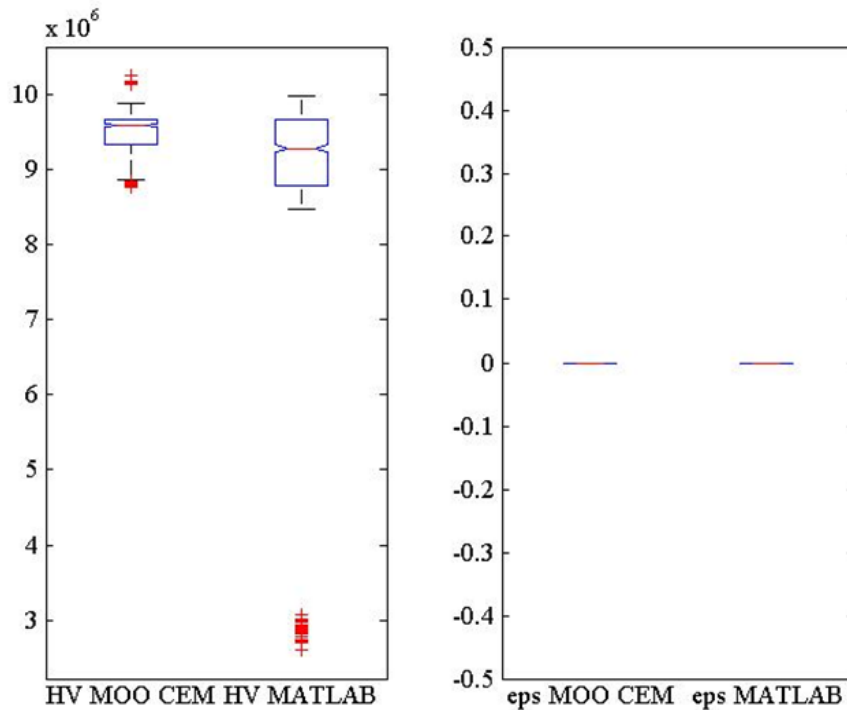
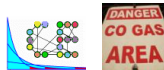


Figure 67: Box-whisker plot of CO gas problem for hyperarea (HV) and epsilon indicator (eps)

The epsilon indicator for the CO gas problem was found to be zero. Being a discrete problem, both algorithms found at least one coordinate in the solution space of the approximation



set that corresponded with the exact coordinate on the true Pareto front. This indicator was therefore of no further relevance.

The null hypothesis for the hyperarea indicator is rejected, which leads to the conclusion that the MOO CEM algorithm outperforms the *Matlab*® MOO GA in optimising the CO gas problem under the stated conditions. The box-whisper plot supports this observation by showing that the mean of the hyperarea set obtained with the MOO CEM algorithm is greater than that produced by the *Matlab*® MOO GA.

9.4 Summary: Chapter 9

This penultimate chapter of this research project has entailed describing an experiment performed to assess the performance quality of the MOO CEM algorithm. Before discussing the comparative assessment per se, some theory on quality indicators for assessing MOO algorithms was discussed, mainly based on an authoritative tutorial by Knowles *et al.* (2006). Three of these quality indicators, the Generation Distance indicator (*GD*), Spacing indicator (*SP*) and Maximum Pareto front Error (*ME*) were individually discussed and the indicator values for each of those were calculated for the MOO CEM algorithm. Note that these quality indicators are used to compare the approximated Pareto front with the true, or reference, Pareto front. All three of these indicators are Pareto non-compliant and therefore not suitable for a comparative exercise.

Of the different approaches to algorithm comparison, the indicator approach was used. The comparative assessment was therefore done using the CO gas problem, while creating values for two recognised qualitative quality indicators (the hyperarea indicator and the additive epsilon indicator) by the MOO CEM algorithm and the commercial MOO GA of *Matlab*®. It was presumed that the performance quality of the MOO CEM algorithm should be comparable to that of at least one other similar algorithm. In order to ensure that that the assessment and its outcomes were sound, it was attempted to follow the best-known and accepted experimental procedure as prescribed by Knowles *et al.* (2006).

The MOO CEM algorithm found a quality indicator value that was superior to that found by the *Matlab*® MOO GA for the hyperarea indicator, while the epsilon indicator had to be discarded due to the test problem's discrete nature. Based on the work described in this chapter it is concluded that the MOO CEM algorithm has merit for application as an optimisation technique for the multi-objective, dynamic, stochastic CO gas problem. It cannot be concluded that the *Matlab*® MOO GA will always be outperformed – the conclusion made in this study pertains only to the CO gas problem and must be understood in the context of the quality indicators used. It is possible that different option settings for the *Matlab*® algorithm could lead to its superiority over the MOO CEM algorithm. However, Bekker (2012), who performed the same comparative assessment on



34 different test problems, commented that changes of *Matlab*[®] option settings were investigated during informal, unreported experiments, which did not lead to results in favour of the *Matlab*[®] MOO GA.

Hereby the description of the experimental work and the results obtained is concluded. In the next and final chapter the research summary, final conclusions and recommendations to the author's employer, based on of the overall research project, are presented.



10. Research summary, conclusions and recommendations

The research project is summarised in this chapter and research conclusions are presented. Recommendations, based on the conclusions, to be presented to *Tronox KZN Sands* have also been included.

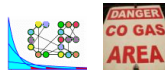
10.1 Summary and conclusions

For the purpose of this thesis a simulation study was performed at the *Tronox KZN Sands* CPC to model and analyse a problem that involves dynamic prioritisation governing the distribution of CO gas to three different gas consuming plants. This problem was termed the “CO gas problem”. The CO gas is produced as a by-product of the ilmenite smelting reaction in the smelter and can be used as a cost effective alternative to the methane gas that is purchased, at an ever increasing cost, for drying and heating purposes at the CPC.

The aim of the study was to recommend a way of optimising the usage of the freely available CO gas, effectively maximising the cost saving achieved on methane gas consumption. Apart from the financial benefit of lower methane gas consumption, this also holds an environmental benefit in that the direct emission of CO₂ gas from the CPC can be decreased by over 7 400 tons per year, or 5%. The latter would decrease the total carbon footprint of *Tronox KZN Sands* by approximately 1.5%. In aiming to maximise the usage of CO gas and thereby the cost saving on methane gas, it needed to be kept in mind that the usage of CO gas could have a negative impact on operations due to production time losses which could occur if the supply of CO gas became insufficient or ceased. In the case of an undersupply of CO gas, a switchover between CO and methane gas would be required, which is disruptive and takes time.

As CO gas is a direct product of the smelting reaction in the furnaces, any significant process changes (such as a furnace idle or downtime) would affect the supply of CO gas. This reality was aggravated by the absence of any buffer capacity (i.e. a pressure vessel) of CO gas at the CPC. The inclusion of a pressure vessel to create buffer capacity was not an option in the short to medium term, for a number of reasons mentioned in the thesis.

The potential benefit of a saving on methane gas costs needed to be balanced against the potential loss of production hours due to switchovers. In the absence of a buffer facility, the selected approach to the problem was the dynamic prioritisation of the available CO gas to the users requiring gas. This involved evaluating different scenarios wherein CO gas was distributed to different combinations of gas consuming plants in the presence of a number of other



probabilistic process variables. The CO gas problem was, from the outset, categorised as a complex, dynamic, stochastic MOP.

The thesis was divided into three themes, with the first theme being a discussion on the theoretical background to the problem and the selected problem solving approach.

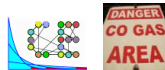
Tackling the CO gas problem from a theoretical (academic) perspective required a literature review to be done to investigate the fields of MOO and metaheuristics. This was done to position the CO gas problem and the anticipated problem solving approach in terms of work done by other authors on similar MOPs. Although no examples could be found in the literature of the application of metaheuristics for the management of furnace off-gas, various illuminating examples were found of effective application of different metaheuristics for solving a wide variety of complex MOPs.

Chapters 2 and 3 included the literature review and discussed the fields of MOO, global search and optimisation, metaheuristics in general and the specific metaheuristic, the MOO CEM, selected for the optimisation of the CO gas problem.

Discrete-event simulation was identified as the preferred tool for incorporating and addressing the real-world dynamics, constraints and uncertainty of the research problem. Hence Chapter 4 was dedicated to a discussion of modelling and simulation as a problem solving tool. Chapter 4 concluded the first thesis theme on the theoretical background of the research project.

In Chapter 5 the research problem was contextualised in terms of the industry, company and specific processes where it originated. This involved an explanation of the heavy minerals industry (with a focus on ilmenite smelting), a discussion of the history and processes of *Tronox KZN Sands*, the company where the CO gas problem was identified. Emphasis was placed on the furnace off-gas management system which formed the specific operational environment relevant to the CO gas problem. The chapter concluded with a brief overview of the important aspects CO gas and CO₂ gas. Hereby the second theme of the thesis was concluded.

The solution development phase involved the concept modelling, the collecting and analyses of input data, selecting the parameters to study, developing the computer simulation model, verification and validation of the model, identifying scenarios to evaluate, generating results per scenario and analysis of the results. These aspects of the solution development phase were covered in Chapters 6, 7 and 8. In fact, the problem solving phase of the project was twofold. The first part, as just mentioned, involved developing the simulation model and evaluating the different pre-selected scenarios with the simulation model. The second part involved a full optimisation exercise to obtain the optimum scenario(s) from all the feasible scenarios in the search space. For the optimisation process, both a simplified approach using a decision factor (a mathematical

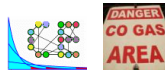


combination of different output parameters in their appropriate relationship to each other) were followed and a metaheuristic suitable for MOO, the MOO CEM algorithm, were applied. The optimisation exercise were discussed in sections 8.4 and 8.5.

From a *commercial perspective*, the performance measures found relevant to the evaluation of the problem were identified as the loss in production time resulting from a specific scenario and the monthly saving on methane gas cost. The results obtained for the performance measures to solve the CO gas problem were included in section 8.6. From an *academic perspective*, the performance of the MOO CEM algorithm was evaluated using three Pareto non-compliant quality indicators and by considering the algorithm's ability to lower the computational burden. The results obtained from comparing the performance quality of the MOO CEM algorithm with that of the NSGA II (as incorporated into the commercially available *Matlab*® MOO GA) by means of two recognised Pareto compliant indicators were discussed in Chapter 9. The NSGA II is often used as a benchmark for new metaheuristics.

The results obtained for the CO gas problem lead to the following conclusions:

- The first conclusion drawn from this research is simply that it was worthwhile to perform a simulation and optimisation exercise for this specific problem. The aim was to determine the optimum scenario(s) for distributing the CO gas in a dynamic, probabilistic environment. That said, the existing static problem solving models at *Tronox KZN Sands* were not adequate for solving the problem. The simulation exercise allowed for an appropriate study of a problem of this type.
- It became clear that with the prevailing operational conditions and equipment configuration at the *Tronox KZN Sands CPC* it is not *necessarily* optimal to run *all* the CO gas consuming plants on CO gas and it is definitely not optimal to distribute CO gas to *none* of those plants. It was found that a significant cost saving on methane gas and reduction of CO₂ emissions could be achieved, *even though no buffer capacity exists*. This could be achieved through *dynamic prioritisation*.
- It was further found that different combinations of gas consumers could be set to run on CO gas, but that every combination of gas consumers had an *impact on the operational stability* of the relevant plants' burners. It could quantitatively be shown how setting more plants to receive the CO gas increased the expected number of disruptions at all the plants, due to switchovers between CO and methane gas. Although this had been anticipated, it could now be quantified.
- A Pareto front could be obtained that clearly indicates the trade-off between the two identified objectives. The Pareto front served as an effective aid whereby the decision maker could

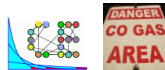


select an optimum combination of decision variables, i.e. which gas consumers were to receive CO gas, the optimum availabilities of the *Howden* plant as well as of the gas consuming plants. By selecting a certain combination of gas consumers and plant availabilities from the Pareto front, the decision maker can have confidence in the expected monthly cost saving, but will also have to accept the probable operational disruptions associated with the decision. Knowledge of the latter should allow the decision maker to prepare plant personnel to be proactive in order to minimise the effect of gas switchovers.

- The decision factor that was applied to give an indication of the optimum scenario was helpful to an extent and correlated somewhat with the results of the metaheuristic. The metaheuristic, however, produced a more comprehensive answer which allowed the decision maker to make a decision with more awareness of the implication thereof, as well as how it compares to other Pareto-optimum decisions.
- It was found that the metaheuristic, compared with an exhaustive enumeration, converged significantly faster (in about 3% of the computation time, or roughly 30 times faster) to obtain an approximated Pareto front. Thereby its *efficiency* was shown. Using the three Pareto non-compliant quality indicators the *effectiveness* of the metaheuristic was proven in showing the approximated Pareto front's good proximity and diversity compared to the true Pareto front. It was thus concluded that the metaheuristic was helpful to 1) reduce the computational burden significantly in optimising this dynamic, stochastic problem, thereby 2) allowing optimum decision making.
- In terms of the comparative exercise between the MOO CEM algorithm and the NSGA II, the results revealed that the MOO CEM algorithm performed superior to the NSGA II based on the hyperarea quality indicator, under the conditions stated for the CO gas problem. The epsilon indicator yielded no result, due to the discrete nature of the CO gas problem.

It can be inferred from the conclusions that the research aim and objectives set in Chapter 1 were achieved. The particular value of the research project was twofold:

- i. The results from the project have added value to the *academic body of knowledge* in that they confirmed the suitability of MOO CEM for optimising a dynamic, stochastic real-world MOP, thereby serving as a supportive case study for the supervisor's research and to academia at large.
- ii. The results also have *commercial value* in that they provide valuable input into the *implementation of the CO gas project at Tronox KZN Sands*. The study in itself was not the sole reason for action being taken to implement the project. Yet, the process of acquiring information for the study, the availability of the simulation model to test scenarios, getting



different role players together to discuss the project and intermittently bringing important aspects pertaining to the project (such as the rising methane gas cost) as well as the research results to the attention of management, all contributed towards the project becoming a reality.

It can therefore be concluded that a positive outcome to the research hypothesis was achieved.

10.2 Recommendations

Based on the conclusions discussed above, the following recommendations were made to *Tronox KZN Sands* management regarding the implementation of the CO gas project:

1. Instead of running no plants on CO gas, as has been the case for most of the life of the CPC to date, the gas should be distributed to any of the combinations of plants that form part of the Pareto optimal set. This will save the company methane gas cost, lower the emission of CO₂ and provide the company with an opportunity to build knowledge and practical experience of operating burners in the plants on CO gas. The extent of the saving can be quantified with high confidence using the simulation model, for example for budgeting purposes. It must be remembered that CO gas remains an occupational hazard, which introduces risks for which the mitigating actions were not addressed by this research. However, with the SHEQ infrastructure and procedures in place at the company, the company should have the capacity to manage these risks successfully.
2. With reference to the Pareto optimum set, it is recommended that the distribution of CO gas should be gradually ramped up to include more plants over time. In other words, it is recommended that the CO gas is initially distributed to one plant only, even though the intention is to eventually achieve a scenario where all the plants operate on CO gas. This recommendation is based on the awareness that, in practice, gaining knowledge and experience in operating a new system successfully one plant at a time, will contribute to gaining sufficient confidence to eventually operate the full system. As indicated by the calculated decision factor values and Pareto front, a decision to start with CO gas to only one plant means that the SPP should initially receive CO gas. This should already incur a saving of around R 300 000 per month with less than one switchover expected per day, hence minimum disruptions to the learning process. With a ramp-up approach, disruptions will initially have to be handled at one plant only. Planned downtimes at the furnaces should be communicated to SPP personnel in advance, so that the plant can be prepared to switch to methane gas over those periods as seamlessly as possible. Once the SPP personnel is comfortable with operating on CO gas and management has seen the cost benefit realised,



other combinations of users from the Pareto optimal set should be supplied with CO gas. It must be mentioned that a dedicated maintenance effort will be required at the *Howden* plant to maintain its availability at 95% and its maintenance cost at a minimum.

3. It is recommended that once the project has been implemented successfully at the SPP, RPP and URIC, the knowledge and experience gained should be utilised to convert the burners of the MTP to also receive CO gas instead of methane gas.
4. The use of a pressure vessel to build a CO gas buffer must be investigated. It is believed that capital expenditure on a pressure vessel will be economically justifiable. As an interim means to improve steady availability of gas at the burners, full automation of the switchovers should be investigated. The latter will still be a suboptimal solution, as production time will still be lost during a switchover, but automation should shorten the switchover period and reduce the need to disrupt personnel in their work. This will also mitigate the risk of having a human in the vicinity of CO gas during a switchover.

The potential to benefit from the CO gas at *Tronox KZN Sands* has for long been seen as a “low hanging fruit” by the author, hence it is recommended to implement these recommendations without delay.



11. References

- ALON, G., KROESE, D. P., RAVIV, T. & RUBINSTEIN, R. Y. 2005. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134, 137-151.
- ANTON, H. 1984. *Calculus with Analytic Geometry*, New York, John Wiley & Sons.
- ARROW, K. J., BARANKIN, E. W. & BLACKWELL, D. 1953. Admissible points of convex sets. *Contributions to the Theory of Games*, 2, 87-91.
- ASIKAINEN, A. 1995. *Discrete-event simulation of mechanized wood-harvesting systems*, Joensuu : University of Joensuu.
- ASMUSSEN, S. & KROESE, D. P. 2006. Improved algorithms for rare event simulation with heavy tails. *Advances in Applied Probability*, 38, 545-558.
- ASMUSSEN, S., KROESE, D. P. & RUBINSTEIN, R. Y. 2005. Heavy Tails, Importance Sampling and Cross-Entropy. *Stochastic Models*, 21, 57-76.
- BÄCK, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, USA.
- BANKS, J. 1998. *Handbook of simulation*, Wiley Online Library.
- BANZHAF, W., NORDIN, P., KELLER, R. E. & FRANCONI, F. D. 1998. *Genetic Programming: An Introduction*. San Francisco, California: Morgan Kaufmann Publishers.
- BEKKER, J. F. 2005. Short notes on aspects of Discrete-event Simulation. Stellenbosch: University of Stellenbosch.
- BEKKER, J. F. 2012. *Applying the cross-entropy method in multi-objective optimisation of dynamic, stochastic systems*. Ph.D. research, University of Stellenbosch.
- BEKKER, J. F. & ALDRICH, C. 2011. The cross-entropy method in multi-objective optimisation: An assessment. *European Journal of Operational Research*, 211, 112-121.
- BEKKER, J. G., CRAIG, I. K. & PISTORIUS, P. C. Modelling and specification for control of an EAF off-gas process. Proceedings of the 9th IFAC symposium on automation in mining mineral and metal processing, 1998 Cologne, Germany. 55-60.
- BEKKER, J. G., CRAIG, I. K. & PISTORIUS, P. C. 1999a. Modelling and simulation of an electric arc furnace process. *ISIJ International*, 39, 23-32.
- BEKKER, J. G., CRAIG, I. K. & PISTORIUS, P. C. 2000. Model predictive control of an electric arc furnace off-gas process. *Control Engineering Practice*, 8, 445-455.
- BHASIN, M. M. 2001. Dehydrogenation and oxydehydrogenation of parafin to olefins. *Applied Catalysis*, A221, 397-419.
- BRASSARD, G. & BRATLEY, P. 1988. *Algorithmics: theory & practice*, Prentice-Hall, Inc.
- BROUWER, G., JANSE VAN VUUREN, R. & VOLK, B. 2003. FUNCTIONAL DESCRIPTION KUMBA HEAVY MINERALS EMPANGENI CPC-SMELTER FURNACE OFF-GAS DISTRIBUTION CIRCUIT 783. Johannesburg: Howden Process Compressors (Pty) Ltd.
- BULL, L. 2004. *Applications of learning classifier systems*, Springer Verlag.
- BURSTROM, E., HÄHLIN, P. & PETERSOHN, C. MEFARC - A Dynamic Process Control System for Electric Arc Furnaces. 4th European Electric Steel Congress, 1992. 495-499.
- CANTOR, G. 1895. Contributions to the Foundation of Transfinite Set Theory. *Mathematische Annalen*, 46, 481-512.
- CANTOR, G. 1897. Contributions to the Foundation of Transfinite Set Theory. *Mathematische Annalen*, 49, 207-246.
- ČERNÝ, V. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45, 41-51.
- CHARNES, A. & COOPER, W. W. 1957. Management models and industrial applications of linear programming. *Management Science*, 4, 38-91.
- CHARNES, A. & COOPER, W. W. 1961. *Management Models and Industrial Applications of Linear Programming*, New York, John Wiley.



- CHEPURI, K. & HOMEM-DE-MELLO, T. 2005. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134, 153-181.
- COELLO COELLO, C. A. 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1, 129-156.
- COELLO COELLO, C. A. 2006. 20 years of evolutionary multi-objective optimization: what has been done and what remains to be done. *Computational Intelligence: Principles and Practice*, 73-88.
- COELLO COELLO, C. A. 2009. Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Frontiers of Computer Science in China*, 3, 18-30.
- COELLO COELLO, C. A. & CORTÉS, N. C. 2005. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6, 163-190.
- COELLO COELLO, C. A., LAMONT, G. B. & VAN VELDHUIZEN, D. A. 2007. *Evolutionary algorithms for solving multi-objective problems*, Springer-Verlag New York Inc.
- COHEN, I., GOLANY, B. & SHTUB, A. 2005. Managing stochastic, finite capacity, multi-project systems through the cross-entropy methodology. *Annals of Operations Research*, 134, 183-199.
- COHON, J. L. 1978. *Multiobjective programming and planning*, Academic Press.
- COHON, J. L. & MARKS, D. H. 1975. A Review and Evaluation of Multi-Objective Programming Techniques. *Water Resources Research*, 11, 208-220.
- CORNE, D., DORIGO, M. & GLOVER, F. 1999. New ideas in optimisation. *Recherche*, 67, 02.
- CORNE, D. & KNOWLES, J. No free lunch and free leftovers theorems for multiobjective optimisation problems. Conference Proceedings: Evolutionary Multi-Criterion Optimization, 2003. Springer, 66.
- CORNE, D., KNOWLES, J. & OATES, M. The Pareto envelope-based selection algorithm for multiobjective optimization. Conference Proceedings: Parallel Problem Solving from Nature PPSN VI, 2000. Springer, 839-848.
- DA CUNHA, N. O. & POLAK, E. 1967. Constrained minimization under vector-valued criteria in finite dimensional spaces. *Journal of Mathematical Analysis and Applications*, 19, 103-124.
- DE BOER, P. T. 2000. *Analysis and sufficient simulation of queuing models of communication systems*. Ph.D., University of Twente.
- DE BOER, P. T., KROESE, D. P., MANNOR, S. & RUBINSTEIN, R. Y. 2005. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134, 19-67.
- DE BOER, P. T., KROESE, D. P. & RUBENSTEIN, R. Y. 2004. A fast cross-entropy method for estimating buffer overflows in queuing networks. *Management Science*.
- DEB, K. 2001. *Multi-objective optimization using evolutionary algorithms*, Wiley.
- DEB, K., PRATAP, A., AGARWAL, S. & MEYARIVAN, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182-197.
- DORIGO, M., MANIEZZO, V. & COLORNI, A. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*, 26, 29-41.
- DOWSLAND, K. A. Simulated annealing. Conference Proceedings: Modern heuristic techniques for combinatorial problems, 1993. John Wiley & Sons, Inc., 20-69.
- DU PLESSIS, A. J. 2008. *Determining tactical operations policies for an auto carrier using discrete-event simulation*. Master's, University of Stellenbosch.
- DUBIN, U. 2002. Application of the cross-entropy method to neural computation. *Unpublished Master's thesis*. Haifa: Technion.
- DUBIN, U. 2004. Application of the cross-entropy method for image segmentation. *Annals of Operations Research*.
- DUCKSTEIN, L. 1984. Multiobjective Optimization in Structural Design: The Model Choice Problem. In: ATREK, E., GALLAGHER, R. H., RAGSDELL, K. M. & ZIENKIEWICZ, O. C. (eds.) *New Directions in Optimum Structural Design*. John Wiley & Sons.



- EDGEWORTH, F. Y. 1881. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*, CK Paul.
- EHRGOTT, M. 2005. *Multicriteria optimization*, Springer Verlag.
- ERICKSON, M., MAYER, A. & HORN, J. 2002. Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA). *Advances in Water Resources*, 25, 51-65.
- EVANS, G. E., KEITH, J. M. & KROESE, D. P. Parallel cross-entropy optimization. Winter Simulation Conference., 2007. IEEE, 2196-2202.
- FISHBURN, P. C. 1978. A Survey of Multiattribute / Multicriterion Evaluation Theories. In: ZIONTS, S. (ed.) *Multiple Criteria Problem Solving*. Berlin: Springer-Verlag.
- FOGEL, D. B. 1995. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*.
- FOGEL, D. B. 1998. *Evolutionary Computation. The Fossil Record. Selected Readings on the History of Evolutionary Algorithms*. New York: *The Institute of Electrical and Electronic Engineers*.
- FOGEL, L. J. 1999. *Artificial Intelligence through Simulated Evolution. Forty Years of Evolutionary Programming*, New York, John Wiley & Sons, Inc.
- FONSECA, C. M. & FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. Proceedings of the 5th International Conference on Genetic Algorithms, 1993. Citeseer, 416.
- GALLIS, C. T. 1996. Activity oriented stochastic computer simulation of forest biomass logistics in Greece. *Biomass and Bioenergy*, 10, 377-382.
- GAREY, M. & JOHNSON, D. 1979. *Computers and Interactibility: A Guide to the Theory of NP-Completeness*, Freeman.
- GEEM, Z. W. & KIM, J. H. 2001. A new heuristic optimization algorithm: harmony search. *Simulation*, 76, 60-68.
- GLADWELL, M. 2005. *Blink: The power of thinking without thinking*, Little, Brown and Company.
- GLOVER, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13, 533-549.
- GLOVER, F. & LAGUNA, M. 1998. *Tabu search*, Springer.
- GOEL, T. & DEB, K. Hybrid methods for multi-objective evolutionary algorithms. Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL, '02), 2002 Singapore. 188-192.
- GOLDBERG, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*, Addison-wesley.
- GOLDBERG, D. E. & RICHARDSON, J. Genetic algorithm with sharing for multimodal function optimization. In: GREFENSTETTE, J. J., ed. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987. Lawrence Erlbaum, 41-49.
- GOULET, D. V., IFF, R. H. & SIROIS, D. L. 1980. Analysis of five forest harvesting simulation models. Part II. Paths, pitfalls, and other considerations. *Forest Products Journal*, 30, 18-22.
- HAUSDORFF, F. 1906. Investigations Concerning Order Types. *Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematische-Physische Klasse*, 58, 106-169.
- HELVIK, B. & WITTNER, O. 2001. Using the cross-entropy method to guide/govern mobile agent's path finding in networks. *Mobile Agents for Telecommunication Applications*, 255-268.
- HILLIARD, M. R., LIEPINS, G. E., PALMER, M. R. & RANGARAJEN, G. The computer as a partner in algorithmic design: Automated discovery of parameters for a multiobjective scheduling heuristic. Proceedings of the Impact of Recent Computer Advances on Operations Research, 1988. 321-331.
- HILLIER, F. S. & LIEBERMAN, G. J. 1967. *Introduction to Operations Research. Operations research*.
- HILLIER, F. S. & LIEBERMAN, G. J. 2005. *Introduction to Operations Research*, Boston, McGraw-Hill.



- HITCH, C. 1953. Sub-optimization in operations problems. *Journal of the Operations Research Society of America*, 1, 87-99.
- HOAG, M. W. 1956. The relevance of costs in operations research. *Operations research*, 448-459.
- HOGG, G. A. 2009. *Multi-stem mechanised harvesting operation analysis: application of discrete-event simulation*. Master's, University of Stellenbosch.
- HOMEM-DE-MELLO, T. & RUBINSTEIN, R. Y. 2002. Rare event estimation for static models via cross-entropy and importance sampling.
- HORN, J. 1997. F1. 9 Multicriterion decision making. *Handbook of Evolutionary Computation*. [HTTP://WWW.LANIA.MX/~CCOELLO/](http://www.lania.mx/~ccoello/). Available: <http://www.lania.mx/~ccoello/>.
- HUI, K. P., BEAN, N., KRAETZL, M. & KROESE, D. P. 2005. The cross-entropy method for network reliability estimation. *Annals of Operations Research*, 134, 101-118.
- HURWICZ, L. 1964. Programming in Linear Spaces. In: ARROW, K. J. K., L. & UZAWA, H. (ed.) *Studies in Linear and Nonlinear Programming*. London, UK: Oxford University Press.
- HUSBANDS, P. Genetic algorithms in optimisation and adaptation. Conference Proceedings: Advances in Parallel Algorithms, 1992. John Wiley & Sons, Inc., 227-276.
- IGEL, C., HANSEN, N. & ROTH, S. 2007. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15, 1-28.
- JONES, J. A. T., BOWMAN, B. & LEFRANK, P. A. 1998. Electric furnace steelmaking.
- KARLIN, S. 2003. *Mathematical Methods and Theory in Games, Programming, and Economics: Two Volumes Bound as One*, Dover Pubns.
- KEITH, J. & KROESE, D. P. SABRES: Sequence Alignment By Rare Event Simulation. Proceedings of the Winter Simulation Conference, 2002 San Diego. 320-327.
- KELTON, W. D., SADOWSKI, R. P. & STURROCK, D. T. 2007. *Simulation with Arena*, New York: McGraw-Hill.
- KENNEDY, J., EBERHART, R. C. & SHI, Y. 2001. Swarm intelligence. 2001. Kaufmann, San Francisco.
- KIRKPATRICK, S., GELATT JR, C. D. & VECCHI, M. P. 1983. Optimization by simulated annealing. *Science*, 220, 671-680.
- KLAHR, C. N. 1958. Multiple objectives in mathematical programming. *Operations research*, 849-855.
- KLINGER, A. 1964. Vector-valued performance criteria. *Automatic Control, IEEE Transactions on*, 9, 117-118.
- KNOWLES, J. D. & CORNE, D. W. 2000. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, 8, 149-172.
- KNOWLES, J. D., THIELE, L. & ZITZLER, E. 2006. A tutorial on the performance assessment of stochastic multiobjective optimizers. *Computer Engineering and Networks Laboratory (TIK)* [Online].
- KOOPMAN, B. O. 1953. The Optimum Distribution of Effort. *Operations Research*, 1, 52-63.
- KOOPMAN, B. O. 1956. Fallacies in Operations Research. *Operations Research*, 4, 422-426.
- KOOPMANS, T. C. 1951. Analysis of production as an efficient combination of activities. *Activity analysis of production and allocation*, 13, 33-37.
- KOZA, J. R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. *Springer*, 229.
- KROESE, D. P. 2011. The cross-entropy method. *Wiley Encyclopedia of Operations Research and Management Science*.
- KROESE, D. P., POROTSKY, S. & RUBINSTEIN, R. Y. 2006. The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 8, 383-407.
- KROESE, D. P. & RUBENSTEIN, R. Y. 2005. The cross-entropy method for combinatorial optimization, rare event simulation and neural computation. *Annals of Operations Research*, 134.
- KUHN, H. W. & TUCKER, A. W. Nonlinear programming. Proceedings of the 2nd Berkeley Symposium on mathematical statistics and probability, 1951 California. 481-492.
- LAMONT, G. B. 1993. Compendium of Parallel Programs for the Intel iPSC Computers. *Wright-Patterson AFB, OH*, 45433.



- LANZI, P. L., STOLZMANN, W. & S.W., W. 2002. *Advances in Learning Classifier Systems. Lecture notes in Computer Science*. San Francisco, California: Springer.
- LAUMANN, M., THIELE, L., DEB, K. & ZITZLER, E. 2002. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10, 263-282.
- LEONARD, T. 2011. *Comparing airport apron layout designs using computer simulation and the cross-entropy method*. Master's, University of Stellenbosch.
- LIEPINS, G. E., HILLIARD, M. R., RICHARDSON, J. & PALMER, M. 1990. Genetic algorithms application to set covering and travelling salesman problems. In: BROWN, D. E. & WHITE, C. C. (eds.) *Operations research and Artificial Intelligence: The integration of problem solving strategies*. Norwell, Massachusetts: Kluwer Academic.
- LÜ, Q., BAI, Z.-H. & XIAO-YAN, X. 2008. Leader-Based Parallel Cross Entropy Algorithm for Maximum Clique Problem *Journal of Software*, 19, 2899-2907.
- LUKE, S. 2009. *Essentials of Metaheuristics*, Lulu.
- MACCRIMMON, K. R. 1973. An overview of multiple objective decision making. *Multiple Criteria Decision Making*, 18-44.
- MANNOR, S., RUBINSTEIN, R. & GAT, Y. The cross-entropy method for fast policy search. Proceedings of the 20th International Conference on Machine Learning (ICML-2003), 2003 Washington DC. 512.
- MARGLIN, S. A. 1967. *Public investment criteria*, Cambridge, Massachusetts, The MIT Press.
- MARGOLIN, L. 2002. *Application of the cross-entropy method for scheduling problems*. Master's, Technion.
- MARGOLIN, L. 2005. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134, 201-214.
- MENACHE, I., MANNOR, S. & SHIMKIN, N. 2005. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134, 215-238.
- METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. & TELLER, E. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1087.
- MICHALEWICZ, Z. 1996. *Genetic algorithms data structures*, Springer.
- MICHALEWICZ, Z. & FOGEL, D. B. 2004. *How to solve it: modern heuristics*, Springer-Verlag New York Inc.
- MIETTINEN, K. 1999. *Nonlinear multiobjective optimization*, Springer.
- MOSCATO, P. 1989. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. *Caltech Concurrent Computation Program, C3P Report*.
- MULLER, A. & LATEGAN, L. 2008. Theme 5: Developing the idea of the thesis and the protocol. *An introduction to postgraduate supervision*, 41-62.
- NASH, J. 1953. Two-person cooperative games. *Econometrica: Journal of the Econometric Society*, 128-140.
- NEAPOLITAN, R. E. & NAIMIPOUR, K. 1996. *Foundations of algorithms*, Lexington, Massachusetts, D.C. Heath and Company.
- O'MEARA, J. & SHAFFER, T. L. 1964. Obscenity in the Supreme Court: A Note on *Jacobellis v. Ohio*. *Notre Dame L.*, 40, 1.
- OOMARDATH, V. 2009. Overview of the mineral sands industry. Empangeni: Exxaro KZN Sands.
- OSYCZKA, A. 1978. An approach to multicriterion optimization problems for engineering design. *Computer Methods in Applied Mechanics and Engineering*, 15, 309-333.
- OSYCZKA, A. 1985. Multicriteria Optimization for Engineering Design. *Design Optimization*, 1, 193-227.
- OYAMA, A., OBAYASHI, S. & NAKAMURA, T. 2001. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. *Applied Soft Computing*, 1, 179-187.
- PARETO, V. 1896. *Cours D'Economie Politique*, Lausanne, F. Rouge.
- PEARL, J. 1984. *Heuristics: intelligent search strategies for computer problem solving*, Reading, MA, Addison-Wesley Pub. Co., Inc.,
- PEGDEN, C. D., SADOWSKI, R. P. & SHANNON, R. E. 1995. *Introduction to simulation using SIMAN*, McGraw-Hill, Inc.



- PENNEY, D. 2000. Carbon monoxide toxicity. *CRC Press LLC, 2000 Corporate Blvd., NW Boca Raton FL 33431 USA*. 295, 295.
- POLYA, G. 2008. *How to solve it: A new aspect of mathematical method*, Princeton University Press.
- PURSHOUSE, R. & FLEMING, P. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. *Evolutionary Multi-Criterion Optimization*, 2003. Springer, 67-67.
- PURSHOUSE, R. C. & FLEMING, P. J. 2007. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation.*, 11, 770-784.
- RAJKUMAR, K. 2012. Post Implementation Review - CO gas utilisation at URIC and Reductant. Empangeni: Tronox KZN Sands.
- RIDDER, A. 2005. Importance sampling simulations of Markovian reliability systems using cross-entropy. *Annals of Operations Research*, 134, 119-136.
- RITZEL, B. J., WAYLAND EHEART, J. & RANJITHAN, S. 1994. Using genetic algorithms to solve a multiobjective groundwater pollution containment problem. *Water Resources Research*, 30, 1589-1603.
- RUBENSTEIN, R. Y. 2002. The cross-entropy method and rare events for maximal cut and bipartition problems. *ACM Transactions on Modelling and Computer Simulation (TOMACS)*.
- RUBINSTEIN, R. Y. 1997. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99, 89-112.
- RUBINSTEIN, R. Y. 1999. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1, 127-190.
- RUBINSTEIN, R. Y. & KROESE, D. P. 2004. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, Springer Verlag.
- RUSSELL, S. & NORVIG, P. 1995. *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, New Jersey., Prentice-Hall.
- SASAKI, D. & OBAYASHI, S. 2005. Efficient search for trade-offs by adaptive range multi-objective genetic algorithms. *Journal of Aerospace Computing, Information, and Communication*, 2, 44-64.
- SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st international Conference on Genetic Algorithms*, 1985. L. Erlbaum Associates Inc., 93-100.
- SCHREINER, B. & REINHARDT, H. J. 2008. Outlook for industrial gases in the petrochemical industry. *Hydrocarbon Processing*.
- SCHUBERT, E. S. & GOTTSCHLING, R. Co-generation: A challenge for furnace off-gas cleaning systems. *In: HOED, R. T. J. P. D., ed. Southern African Pyrometallurgy*, 2011 Johannesburg. Southern African Institute of Mining and Metallurgy.
- SCHWEFEL, H. P. 1995. *Evolution and optimum seeking*, Wiley.
- SCRIBER, T. J. & BRUNNER, D. T. Inside Discrete-event Simulation Software: How it works and why it matters. *In: JAIN, S., CREASEY, R. R., HIMMELSPACH, J., WHITE, K. P. & FU, M., eds. Proceedings of the Winter Simulation Conference*, 2011.
- SELBY, J. H. 2007. The industrial uses of zircon and zirconia, and the radiological consequences of these uses.
- STADLER, W. 1979. A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960. *Journal of Optimization Theory and Applications*, 29, 1-52.
- STADLER, W. 1988. *Multicriteria Optimization in Engineering and in the Sciences*, Springer.
- STORN, R. & PRICE, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341-359.
- TAHA, H. 2003. *Operations Research*, New Jersey, Prentice Hall.
- TURKDOGAN, E. 1996. *Fundamentals of steelmaking*, Institute of Materials, Minerals and Mining.
- TYLER, R. M. & MINNITT, R. C. A. 2004. A review of sub-Saharan heavy mineral sand deposits: implications for new projects in southern Africa. *JOURNAL-SOUTH AFRICAN INSTITUTE OF MINING AND METALLURGY*, 104, 89-100.



- VAN VELDHUIZEN, D. A. 1999. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*.
- VAN VELDHUIZEN, D. A. & LAMONT, G. B. Multiobjective optimization with messy genetic algorithms. Proceedings of the 2000 ACM symposium on Applied computing-Volume 1, 2000 2000. ACM, 470-476.
- VENTER, G. 2010. Review of Optimization Techniques. In: BLOCKLEY, R. & SHYY, W. (eds.) *Encyclopedia of Aerospace Engineering*. John Wiley & Sons.
- VICINI, A. & QUAGLIARELLA, D. A multiobjective approach to transonic wing design by means of genetic algorithms. 35th AIAA Aerospace Science Meeting and Exhibit, 1997. DTIC Document.
- WALTHER, D. 2007. Chemie mit CO₂. *Nachr. aus der Chemie*, 55, 1188-1194.
- WATKINS, C. J. C. H. & DAYAN, P. 1992. Q-learning. *Machine Learning*, 8, 279-292.
- WHITE, K. P. & INGALLS, R. G. Introduction to simulation. Winter Simulation Conference (WSC), Proceedings of the 2009, 2009 2009. IEEE, 12-23.
- WOJSZNIS, W., MEHTA, A., WOJSZNIS, P., THIELE, D. & BLEVINS, T. 2007. Multi-objective optimization for model predictive control. *ISA transactions*, 46, 351-361.
- WOLPERT, D. H. & MACREADY, W. G. 1997. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1, 67-82.
- WWW.ASSE.ORG 2003. Carbon Monoxide - Odorless, Colorless, Deadly. *Professional Safety*.
- WWW.CEMETHOD.ORG. Available: <http://www.cemethod.org>.
- WWW.FREEDICTIONARY.COM. Available: www.freeditonary.com.
- WWW.MININGWEEKLY.COM. Available: <http://www.miningweekly.com/article/exxaro-and-tronox-to-combine-mineral-sands-operations-2011-09-26>.
- YANG, X. S. 2008. Firefly algorithm. *Nature-Inspired Metaheuristic Algorithms*, 79-90.
- ZADEH, L. A. 1963. Optimality and Nonscalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, AC-8, 59-60.
- ZHOU, A., QU, B. Y., LI, H., ZHAO, S. Z., SUGANTHAN, P. N. & ZHANG, Q. 2011. Multiobjective evolutionary algorithms: A survey of the state-of-the-art. *Swarm and Evolutionary Computation*.
- ZIESAK, M., BRUCHNER, A. K. & HEMM, M. 2004. Simulation technique for modelling the production chain in forestry. *European Journal of Forest Research*, 123, 239-244.
- ZIETSMAN, J. H. & PISTORIUS, P. C. 2006. Modelling of an ilmenite-smelting DC arc furnace process. *Minerals Engineering*, 19, 262-279.
- ZITZLER, E., LAUMANN, M., THIELE, L., FONSECA, C. M. & DA FONSECA, V. G. Why quality assessment of multiobjective optimizers is difficult. 2002. Morgan Kaufmann Publishers Inc., 666-674.
- ZITZLER, E. & THIELE, L. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3, 257-271.
- ZITZLER, E., THIELE, L., LAUMANN, M., FONSECA, C. M. & DA FONSECA, V. G. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7, 117-132.

Appendices

Appendix A Input data analysis

Only the analyses of three input variables are given as examples.

CO gas production from FC1 – 1 January 2010 to 30 June 2010

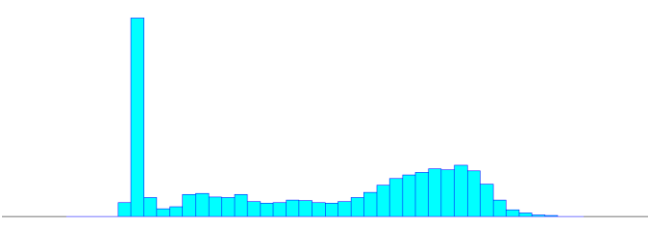
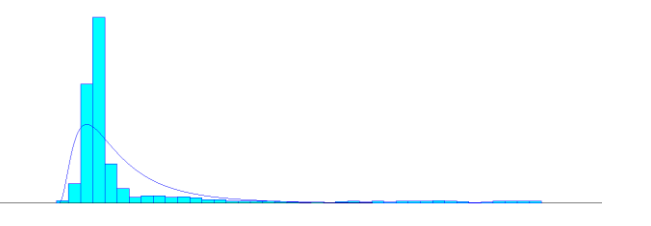
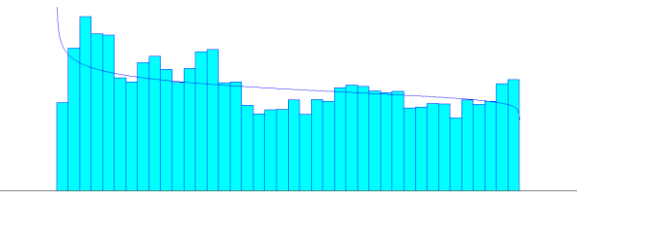
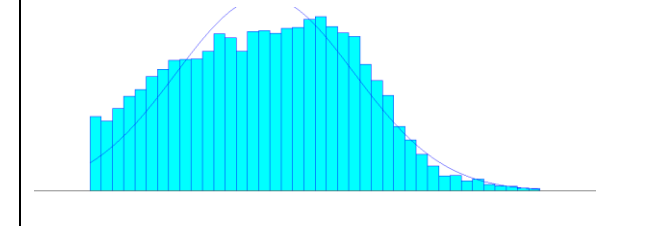
FC1 CO gas production: All data	FC1 CO gas production $\leq 15 \text{ m}^3/\text{min}$	FC 1 CO gas production $> 15 \text{ m}^3/\text{min}, \leq 65 \text{ m}^3/\text{min}$	FC1 CO gas production $> 65 \text{ m}^3/\text{min}$
 <p>Data Summary</p> <p>Number of Data Points = 25551 Min Data Value = -17.4 Max Data Value = 129 Sample Mean = 52.1 Sample Std Dev = 37.5</p> <p>Histogram Summary</p> <p>Histogram Range = -18 to 129 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Lognormal Expression: $1 + \text{LOGN}(1.96, 1.66)$ Square Error: 0.085277</p> <p>Chi Square Test¹¹</p> <p>Number of intervals = 28 Degrees of freedom = 25 Test Statistic = 9.16×10^3 Corresponding p-value < 0.005</p> <p>Kolmogorov-Smirnov Test</p> <p>Test Statistic = 0.242 Corresponding p-value < 0.01</p> <p>Data Summary</p> <p>Number of Data Points = 6469 Min Data Value = 1.29 Max Data Value = 15 Sample Mean = 3.18 Sample Std Dev = 2.69</p> <p>Histogram Summary</p> <p>Histogram Range = 1 to 15 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Beta Expression: $15 + 50 * \text{BETA}(0.91, 1.04)$ Square Error: 0.000952</p> <p>Chi Square Test</p> <p>Number of intervals = 40 Degrees of freedom = 37 Test Statistic = 227 Corresponding p-value < 0.005</p> <p>Kolmogorov-Smirnov Test</p> <p>Test Statistic = 0.0478 Corresponding p-value < 0.01</p> <p>Data Summary</p> <p>Number of Data Points = 6481 Min Data Value = 15 Max Data Value = 65 Sample Mean = 37.9 Sample Std Dev = 14.7</p> <p>Histogram Summary</p> <p>Histogram Range = 15 to 65 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Normal Expression: $\text{NORM}(87.2, 11.3)$ Square Error: 0.001314</p> <p>Chi Square Test</p> <p>Number of intervals = 39 Degrees of freedom = 36 Test Statistic = 743 Corresponding p-value < 0.005</p> <p>Data Summary</p> <p>Number of Data Points = 12198 Min Data Value = 65 Max Data Value = 122 Sample Mean = 87.2 Sample Std Dev = 11.3</p> <p>Histogram Summary</p> <p>Histogram Range = 65 to 122 Number of Intervals = 40</p>

Figure 68: Example of input data analysis – FC1 CO gas production

¹¹ In cases where hypothesis tests were rejected when fitting a distribution on input data sets, the distribution that fitted the closest were still used in the simulation. Due to the large data sets, the hypothesis tests were often rejected.

CO gas production from FC2 – 1 January 2010 to 30 June 2010

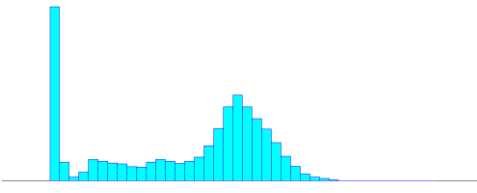
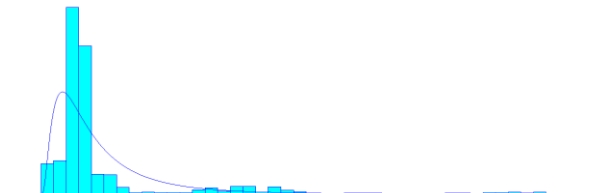
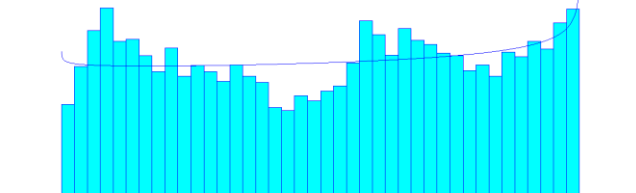
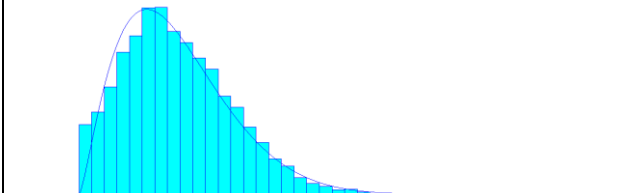
FC2 CO gas production: All data	FC2 CO gas production $\leq 15 \text{ m}^3/\text{min}$	FC 2 CO gas production $> 15 \text{ m}^3/\text{min}, \leq 65 \text{ m}^3/\text{min}$	FC2 CO gas production $> 65 \text{ m}^3/\text{min}$
 <p>Data Summary</p> <p>Number of Data Points = 25540 Min Data Value = 0.0772 Max Data Value = 164 Sample Mean = 56.6 Sample Std Dev = 34.9</p> <p>Histogram Summary</p> <p>Histogram Range = 0 to 164 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Lognormal Expression: $\text{LOGN}(1.84, 1.86)$ Square Error: 0.080991</p> <p>Chi Square Test</p> <p>Number of intervals = 26 Degrees of freedom = 23 Test Statistic = 4.55×10^3 Corresponding p-value < 0.005</p> <p>Kolmogorov-Smirnov Test</p> <p>Test Statistic = 0.211 Corresponding p-value < 0.01</p> <p>Data Summary</p> <p>Number of Data Points = 5351 Min Data Value = 0.0772 Max Data Value = 14.9 Sample Mean = 2 Sample Std Dev = 2.59</p> <p>Histogram Summary</p> <p>Histogram Range = 0 to 15 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Beta Expression: $15 + 50 * \text{BETA}(0.985, 0.925)$ Square Error: 0.000639</p> <p>Chi Square Test</p> <p>Number of intervals = 40 Degrees of freedom = 37 Test Statistic = 156 Corresponding p-value < 0.005</p> <p>Kolmogorov-Smirnov Test</p> <p>Test Statistic = 0.0269 Corresponding p-value < 0.01</p> <p>Data Summary</p> <p>Number of Data Points = 6018 Min Data Value = 15 Max Data Value = 65 Sample Mean = 40.6 Sample Std Dev = 14.9</p> <p>Histogram Summary</p> <p>Histogram Range = 15 to 65 Number of Intervals = 40</p>	 <p>Distribution Summary</p> <p>Distribution: Beta Expression: $65 + 99 * \text{BETA}(2.39, 10)$ Square Error: 0.001143</p> <p>Chi Square Test</p> <p>Number of intervals = 25 Degrees of freedom = 22 Test Statistic = 816 Corresponding p-value < 0.005</p> <p>Data Summary</p> <p>Number of Data Points = 14171 Min Data Value = 65 Max Data Value = 164 Sample Mean = 84.1 Sample Std Dev = 10.7</p> <p>Histogram Summary</p> <p>Histogram Range = 65 to 164 Number of Intervals = 40</p>

Figure 69: Example of input data analysis – FC2 CO gas production

URIC MTTF and MTTR – 1 January 2010 to 30 June 2010

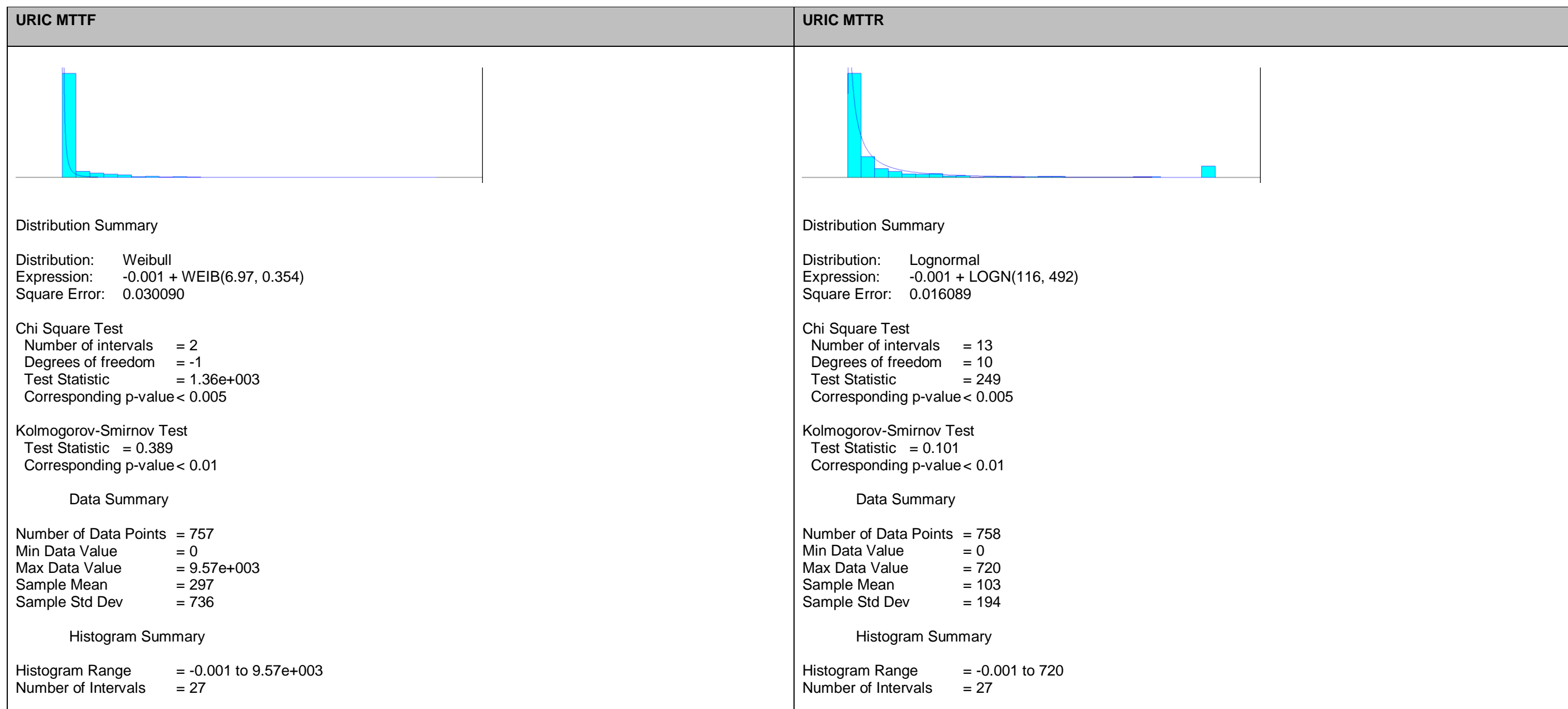


Figure 70: Example of input data analysis – URIC MTTF and MTTR

Appendix B Microsoft® Visio model of the CO gas problem

Microsoft® Visio Representation of the CO gas supply per minute from the furnaces

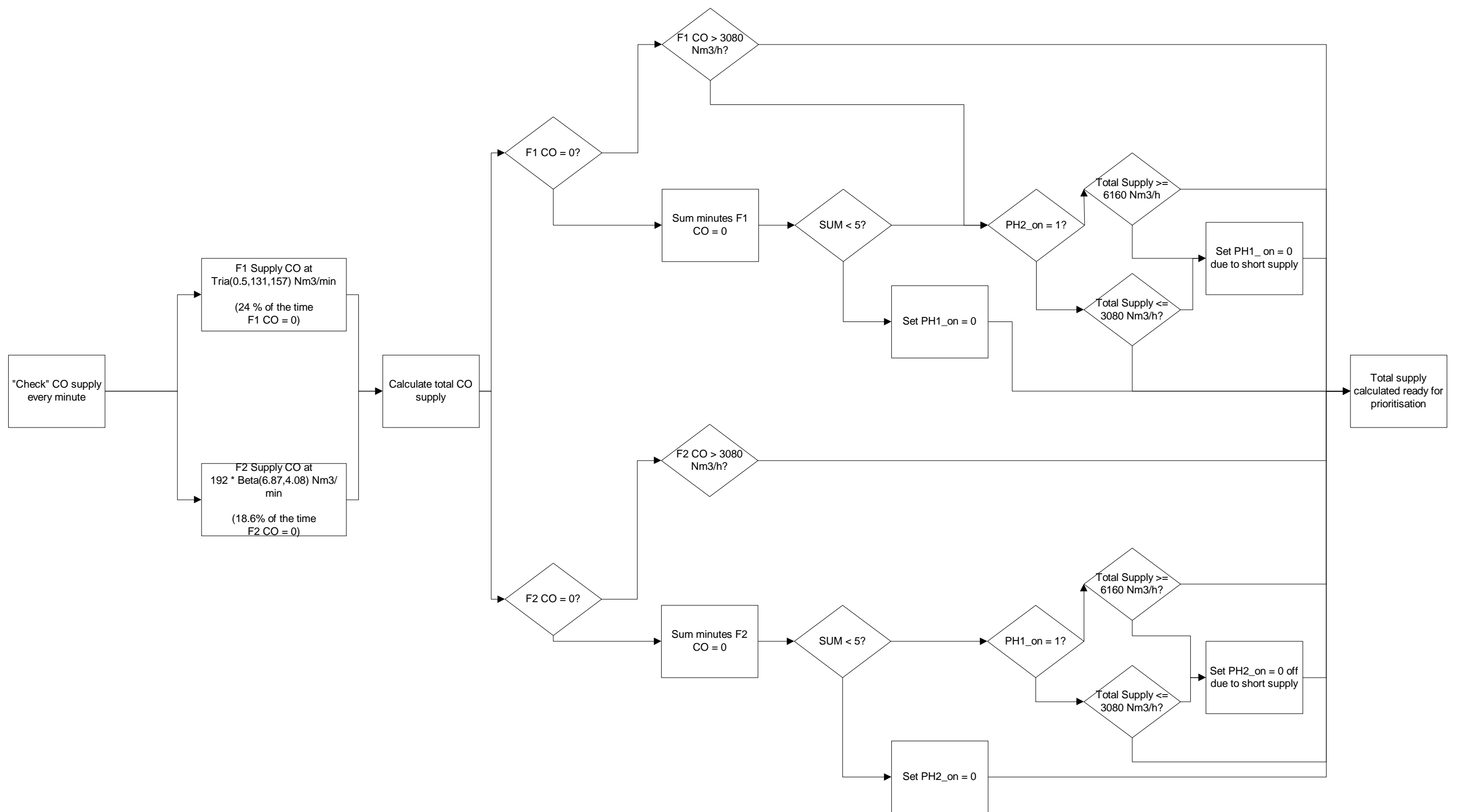


Figure 71: Microsoft® Visio depiction of CO gas supply

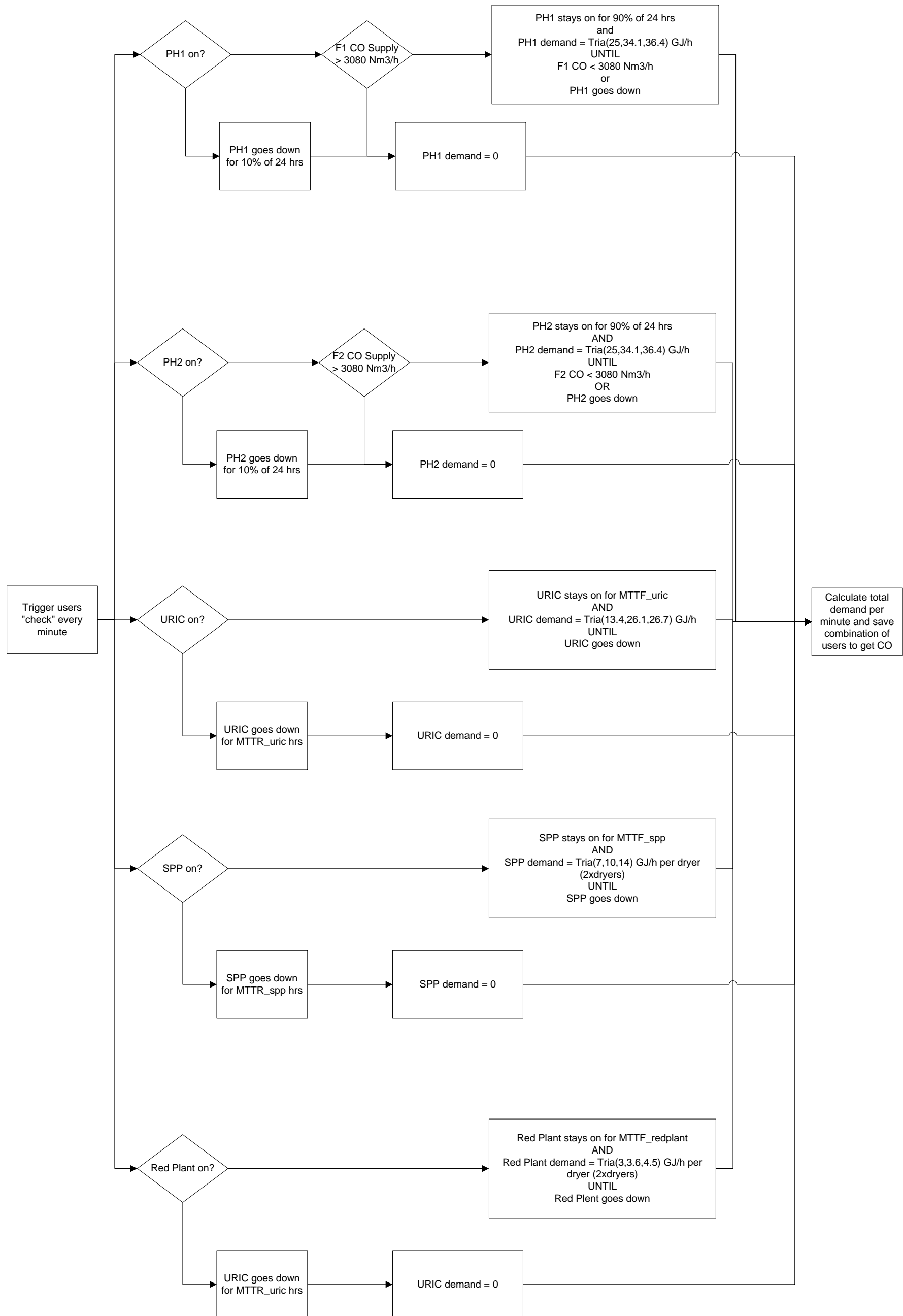


Figure 72: Microsoft® Visio depiction of establishing the demand for gas per minute

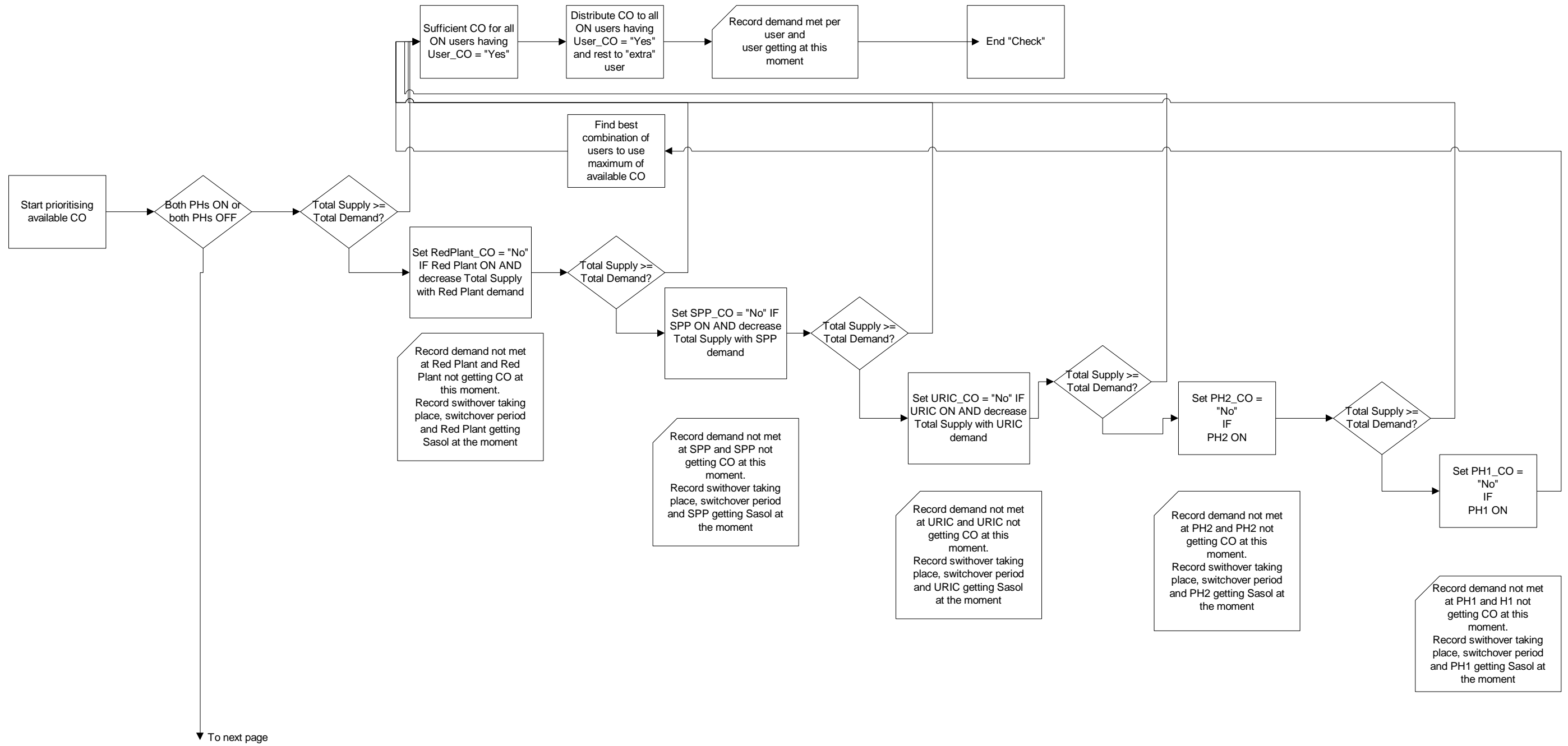


Figure 73: Microsoft® Visio depiction of dynamic prioritisation process

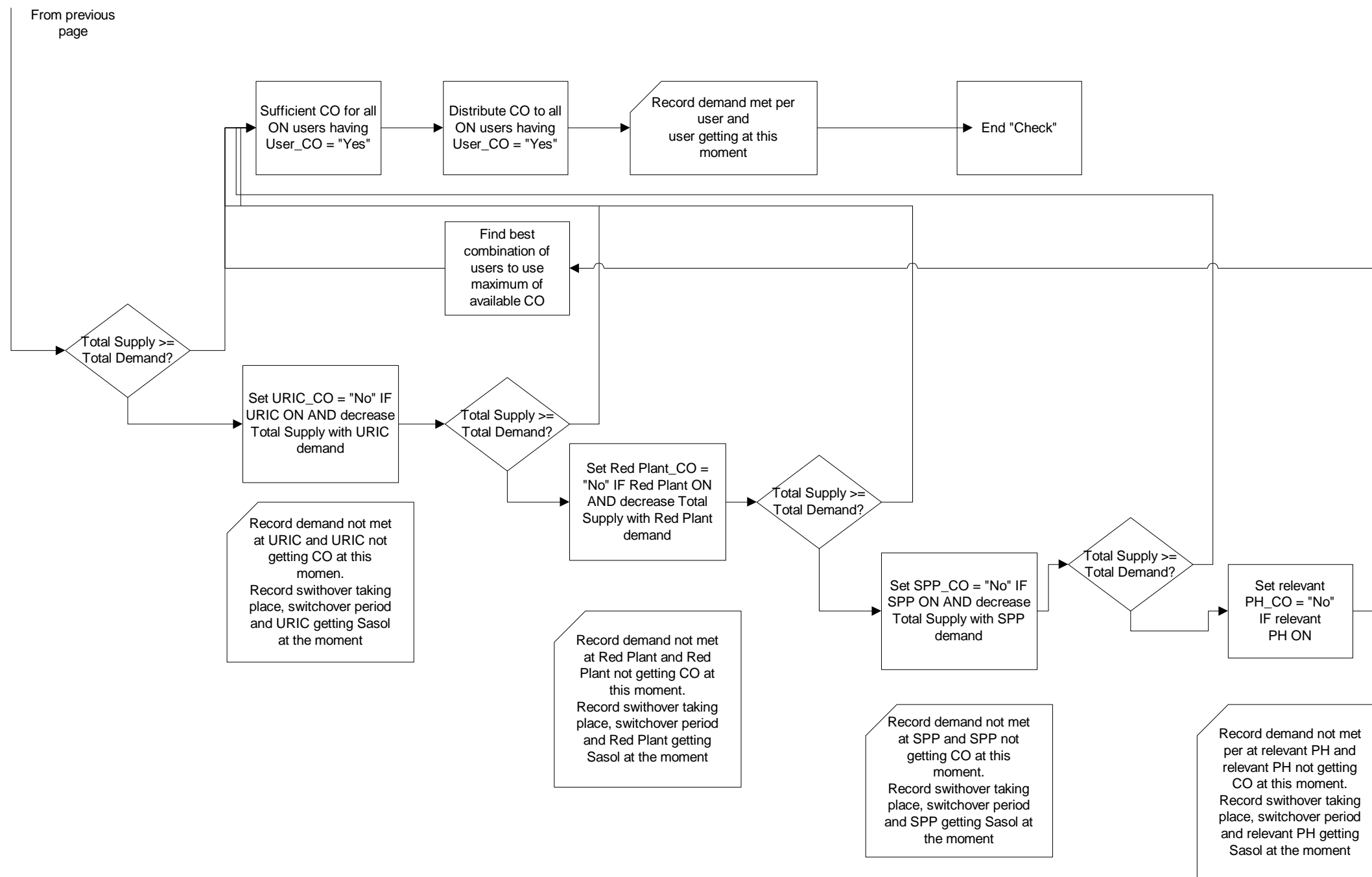


Figure 74: Microsoft® Visio depiction of dynamic prioritisation process (continued from previous page)

Appendix C Arena® model of the CO gas problem

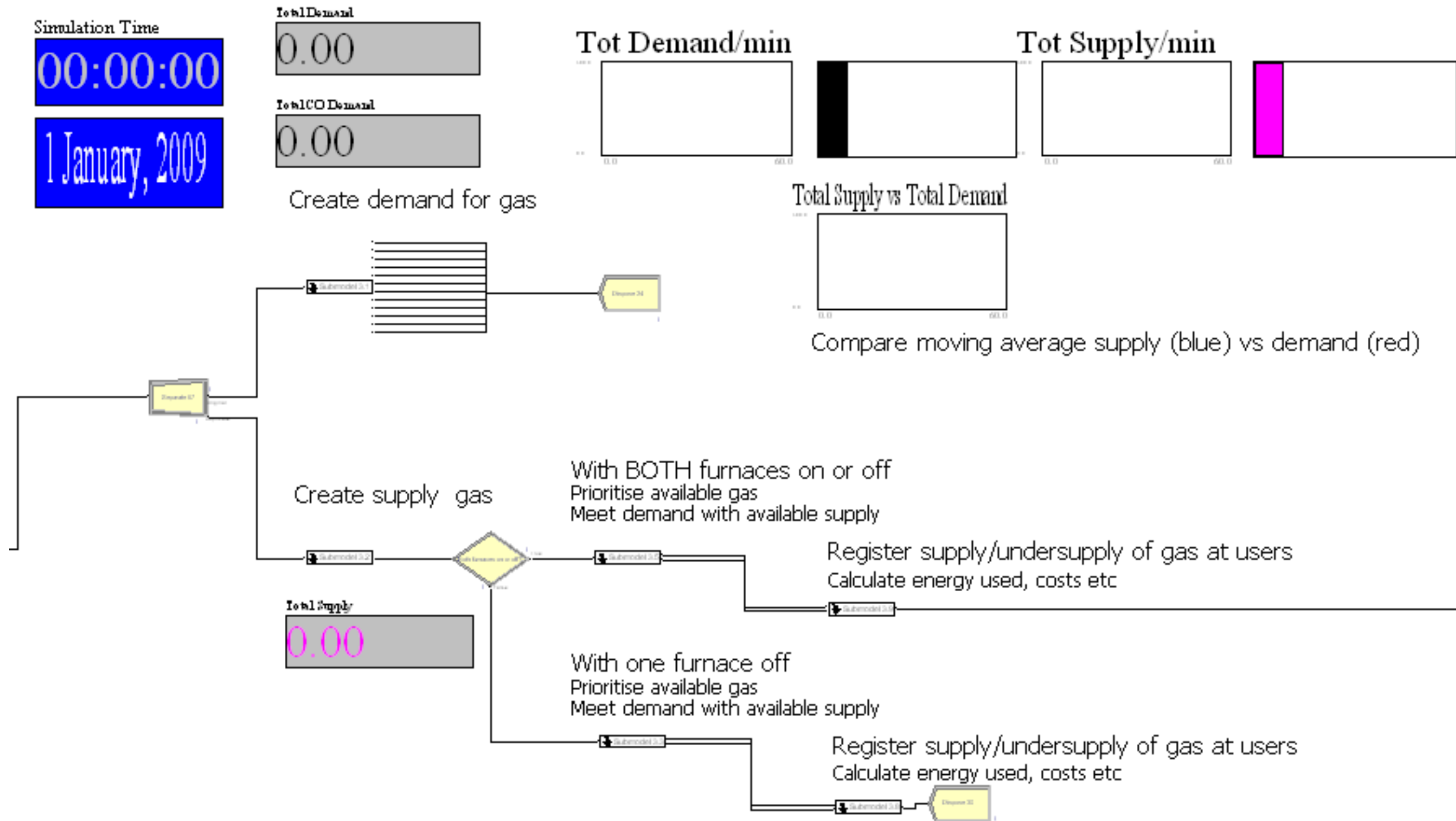


Figure 75: Arena® model. Overview of the submodels

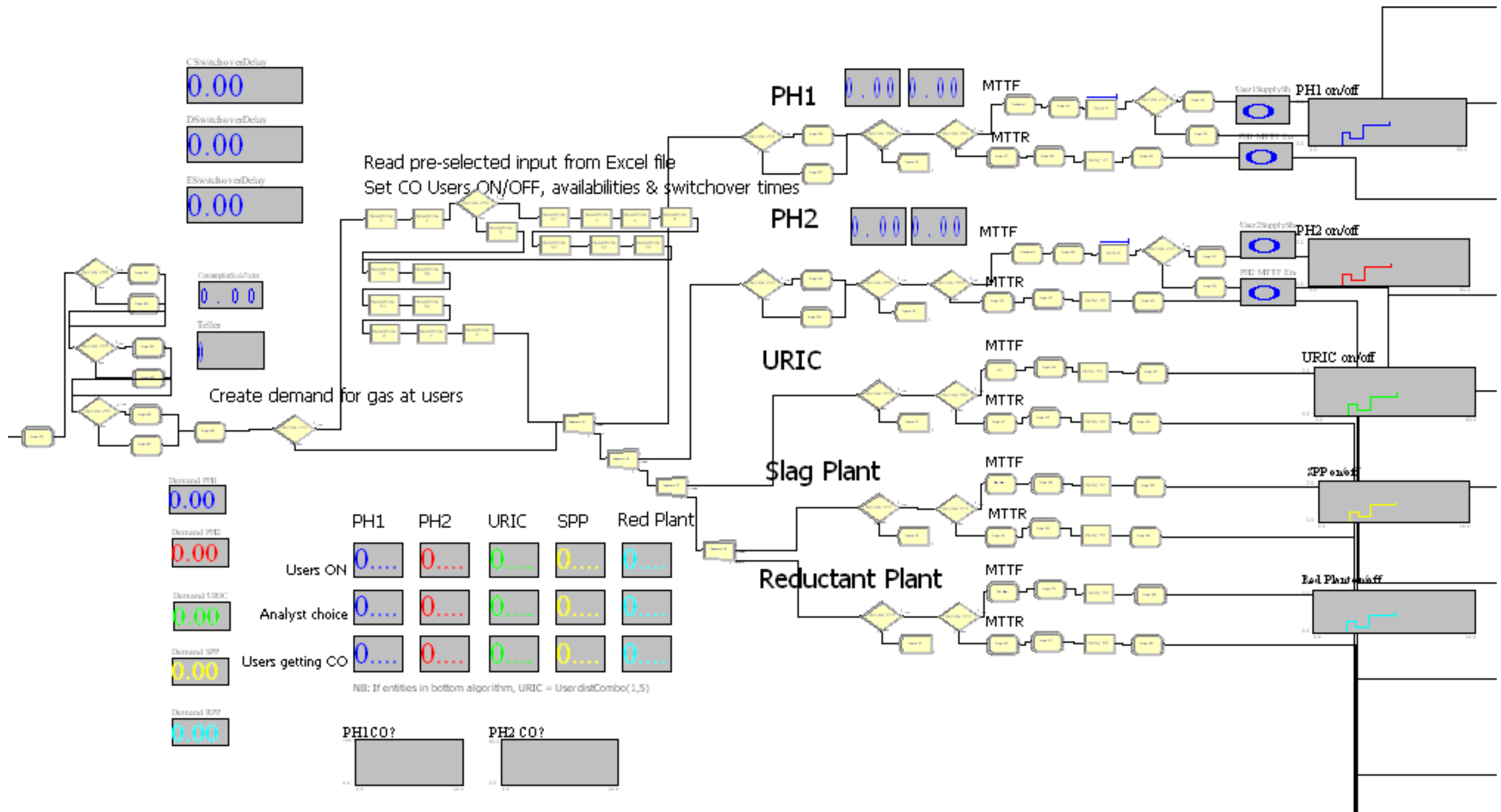


Figure 76: Arena® model. Submodel 1 – Creating the demand for gas

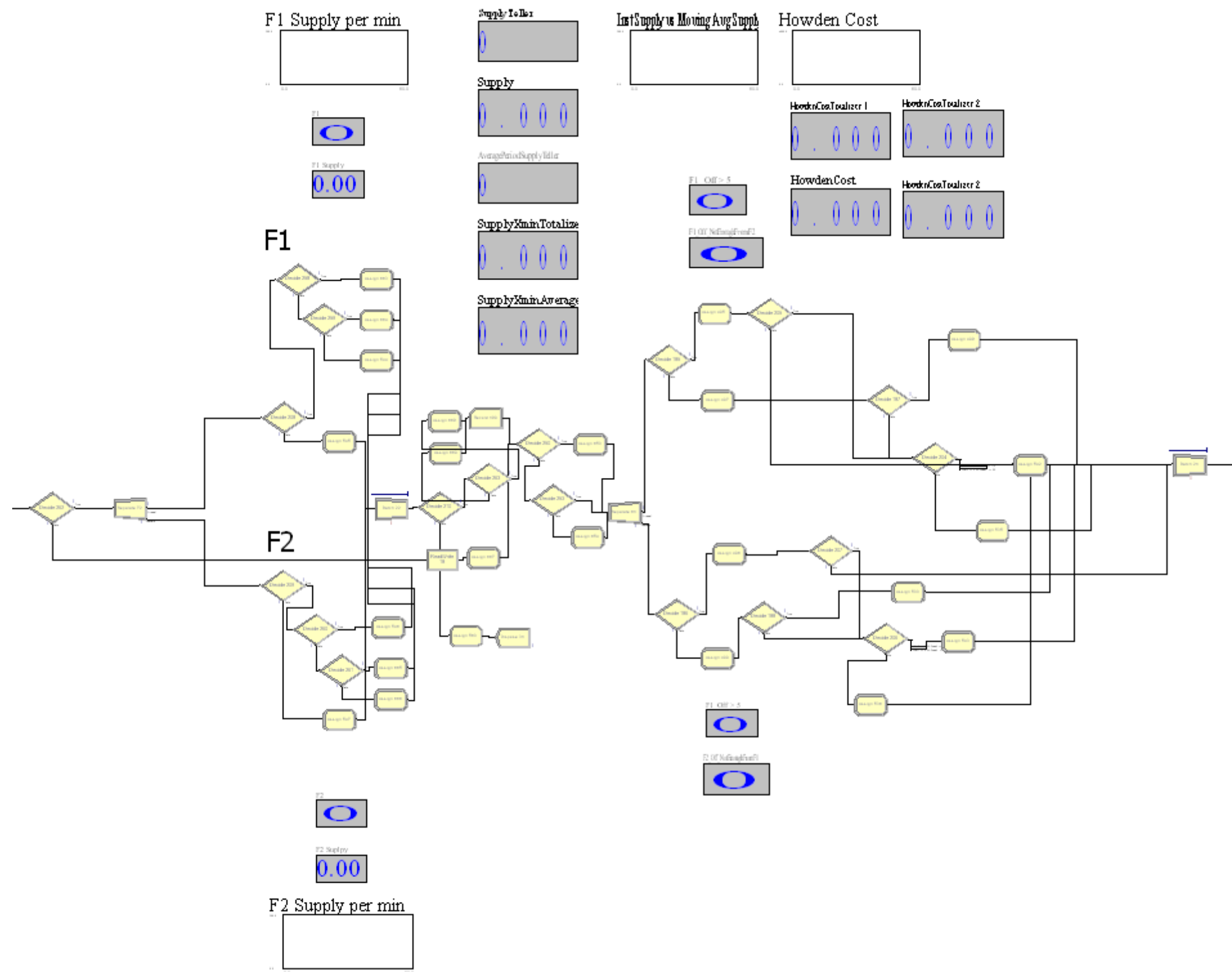


Figure 77: Arena® model. Submodel 2 – Creating the supply of gas by the two furnaces

	PH1	PH2	URIC	SPP	Red Plant
Users ON	0....	0....	0....	0....	0....
Users getting CO	0....	0....	0....	0....	0....

NB: IF entities in bottom algorithm, URIC = UserdistCombo(1,5)

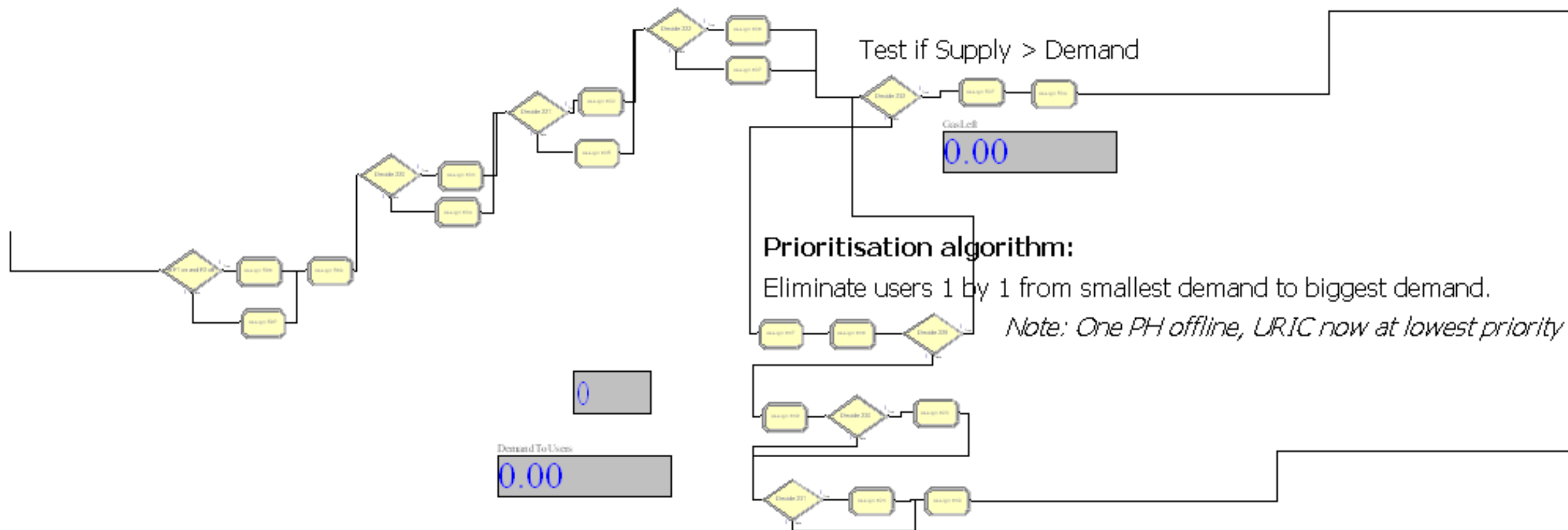


Figure 78: Arena® model. Submodels 3 and 4 combined – Prioritisation of the CO gas

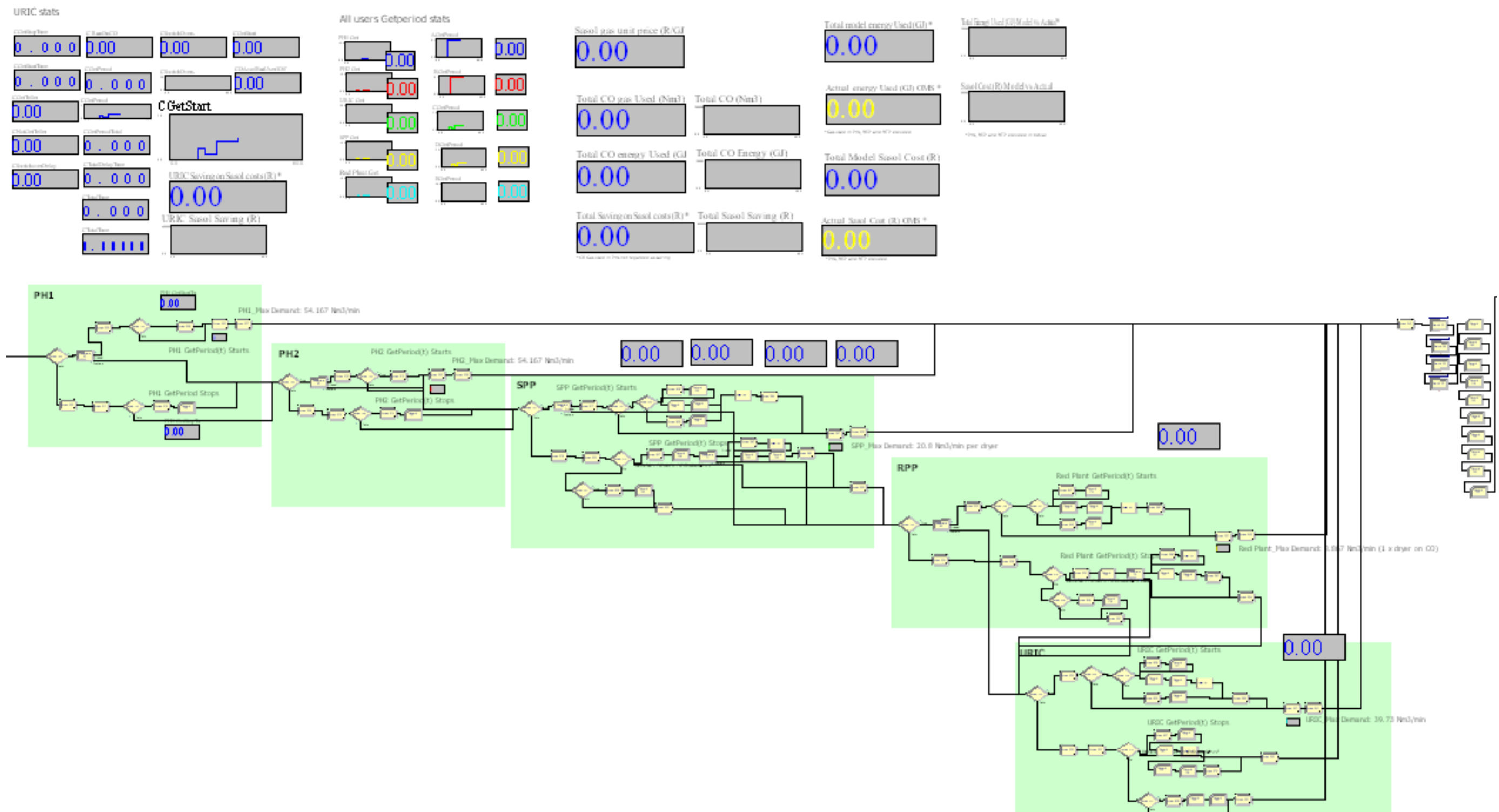


Figure 79: Arena® model. Submodels 5 and 6 combined – Registration of gas usage at the users

Appendix D Howden plant distribution priorities

Howden plant functional manual (Brouwer *et al.*, 2003) p.36: Prioritisation of CO gas users

Period 1 – No Furnace Off-Gas available (6 – 14 ton/h production Furnace 1)±3months		
Period 2 – Furnace 1 Off-Gas available (14 – 20 ton/h production Furnace 1)±5months		
One furnace producing 14 ton/h	3080m³/h	
Pre-Heater 1		
One furnace producing 18 ton/h	3960m³/h	
Pre-Heater 1		
Slag Dryer		
Period 3 – Furnace 1 Off-Gas available (20 – 28 ton/h production Furnace 1)±8months		
One furnace producing		
Pre-Heater 1		
Slag dryer		
Metal treatment		
Period 4 – Furnace 1&2 Off-Gas available (28 – 32 ton/h production Furnace 1&2)		
Two furnaces producing	One furnace producing	One furnace producing
Pre-Heater 1 & 2	Pre-Heater 1 or 2	URIC drier
URIC drier	Slag drier	Slag drier
Slag drier	Reductant drier 1 & 2	Reductant drier 1 & 2
Reductant drier 1 & 2	URIC drier	Roaster
Roaster	Roaster	
Period 5 – Furnace 1&2 Off-Gas available (>32 ton/h production Furnace 1&2)		
Two furnaces producing	One furnace producing	One furnace producing
Pre-Heater 1 & 2	Pre-Heater 1 or 2	URIC drier
URIC drier	Slag drier	Slag drier
Slag drier	Reductant drier 1 & 2	Reductant drier 1 & 2
Reductant drier 1 & 2	URIC drier	Ladle Heaters 1 2 &3
Ladle Heaters 1 2 &3	Ladle Heaters 1 2 &3	Refractory burners
Refractory burners	Refractory burners	Laundry burners
Laundry burners	Laundry burners	Roaster
Roaster	Roaster	

Figure 80: Howden plant prioritisation logic

Appendix E Project Implementation Plan

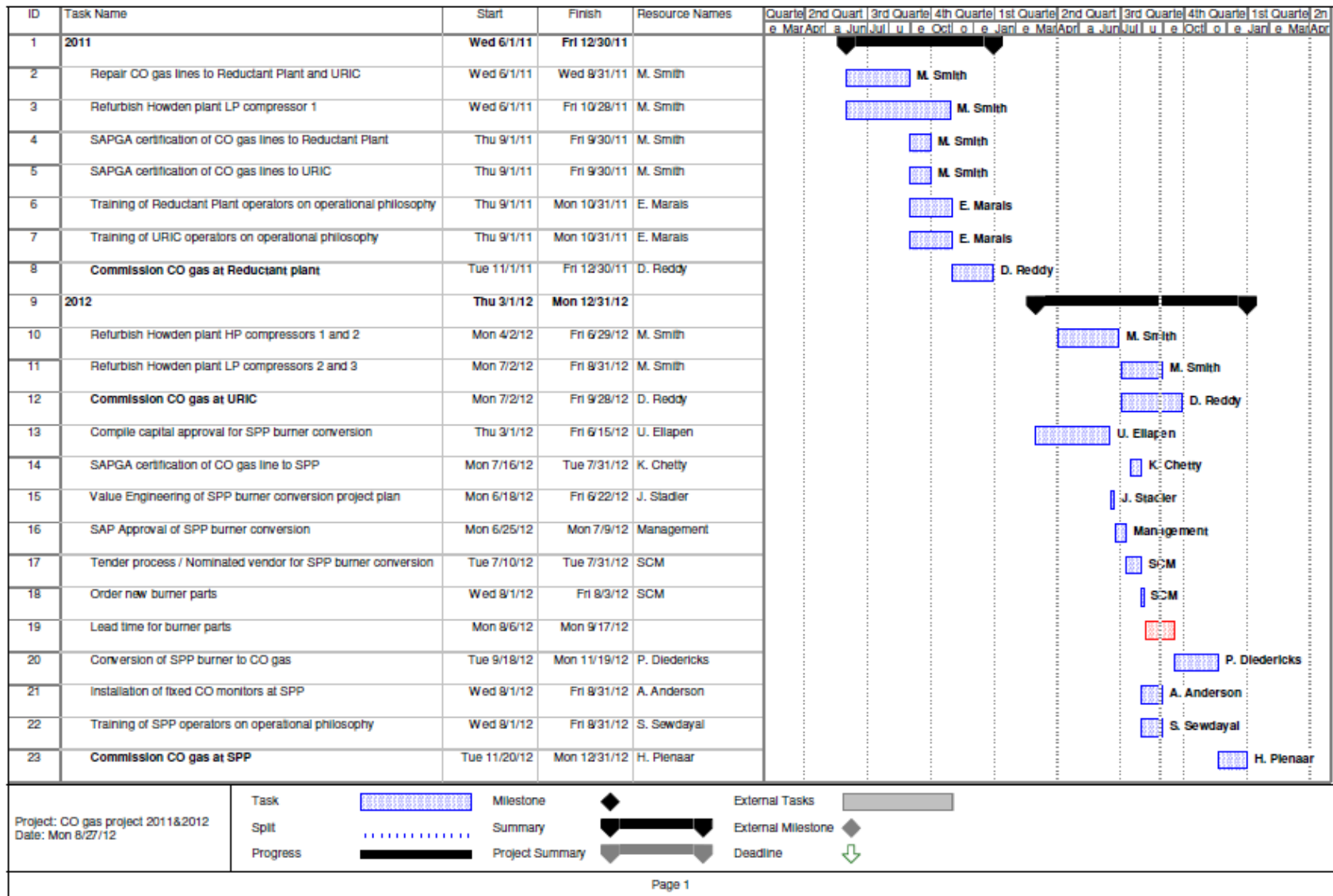


Figure 81: Microsoft® Project CO Gas Project implementation plan