Bi-objective Generator Maintenance Scheduling for a National Power Utility

Berndt Gerald Lindner

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren

March 2017

Declaration

By submitting this dissertation 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.

Date: March 2017

Abstract

One of the key focus areas for the management of a power utility in a regulated energy market is planned preventative maintenance of the power generating units in its power system. The so-called *generator maintenance scheduling* (GMS) problem refers to finding a schedule according to which the planned maintenance can be performed on the generating units in a power system. A novel bi-objective optimisation model is proposed in this dissertation for the GMS problem in which demand satisfaction reliability is maximised and electricity production cost is minimised.

The first scheduling objective is one of the most common objectives in GMS problems in the literature, namely minimising the sum of squared net reserve levels. This objective serves to create an even (reliable) margin of generating capacity over expected demand. The second scheduling objective is the (linear) production cost associated with a maintenance plan of all the generating units in a system. The latter objective is aimed at exploiting the following correlation: planning maintenance on a cost-efficient power station during a high-demand period incurs a higher fuel cost. Production cost is simply taken as fuel cost in this dissertation since it is the most prominent production cost component of power generation.

Dominance-based multi-objective simulated annealing is adopted as model solution technique. Solving the aforementioned model clearly demonstrates that maintenance schedules which minimise the sum of squared reserves are typically also associated with low production costs, but that the lowest sum of squared reserves maintenance schedule does not necessarily achieve the lowest production cost (a sentiment also reported in the literature). Hence there is a need for adopting a multi-objective modelling approach in the context of GMS problems in search of trade-off solutions rather than adopting a standard single-objective modelling paradigm. A sensitivity analysis is performed in respect of model constraint relaxations and the degree of constraint violations. In the process, certain soft constraints which sensitively influence the model objectives are identified.

A decision support system, whose working is based on the bi-objective optimisation model described above, is designed and a concept demonstrator of this system is implemented on a personal computer. This concept demonstrator may be used to find and analyse trade-off solutions to instances of the GMS model and offers interactive features which facilitate sensitivity analyses in a very natural way. The viability and practical use of the concept demonstrator is finally illustrated by applying it to two realistic GMS case studies. It is found that the decision system is capable of producing high-quality sets of trade-off maintenance schedules in each case.

Uittreksel

Een van die hooffokusareas vir die bestuur van ’n kragvoorsiener in ’n gereguleerde energiemark is beplande voorkomende onderhoud van die opwekkingseenhede in sy kragnetwerk. Die sogenaamde *opwekkingseenheid-onderhoudskeduleringsprobleem* (OOP) behels die soeke na ’n skedule waarvolgens beplande, voorkomende onderhoud van kragopwekkingseenhede in ’n kragnetwerk uitgevoer kan word. ’n Nuwe tweedoelige optimeringsprobleem word in hierdie proefskrif vir die OOP daargestel waarin vraagvoorsieningsbetroubaarheid gemaksimeer word en kragopwekkingskoste geminimeer word.

Die eerste skeduleringsdoel is een van die mees algemene doele in OOPe in die literatuur, naamlik die minimering van die som van gekwadreerde netto reserwevlakke. Hierdie doel is daarop gemik om ’n gelykmatige (betroubare) band van opwekkingskapasiteit bo en behalwe die verwagte vraag te lewer. Die tweede skeduleringsdoel is die (lineêre) produksiekoste wat met ’n onderhoudsplan vir al die opwekkingseenhede in ’n stelsel gepaard gaan. Laasgenoemde doel is daarop gemik om skedules te lewer wat die volgende korrelasie uitbuit: beplanning van onderhoud aan ’n koste-doeltreffende opwekkingseenheid gedurende ’n periode van hoë vraag het ook ’n hoë brandstofkoste tot gevolg. Produksiekoste word in hierdie proefskrif bloot as brandstofkoste geneem, aangesien dit een van die mees prominente kostekomponente van kragopwekking is.

Dominasie-gebaseerde, veeldoelige gesimuleerde tempering word as modeloplossingstegniek ingespan. Oplossing van die bogenoemde model toon duidelik dat onderhoudskedules wat die som van gekwadreerde reservevlakke minimeer ook geneig is om skedules met lae gepaardgaande produksiekostes te lewer, maar dat die beste skedules in terme van die som van gekwadreerde reservevlakke nie noodwendig die laagste gepaardgaande produksiekostes tot gevolg het nie (’n sentiment wat ook in die literatuur gerapporteer word). Gevolglik is dit in die konteks van die OOP wenslik om ’n veeldoelige optimeringsbenadering in die soeke na afruilingsoplossings te volg eerder as om die standaard, enkeldoelige optimeringsparadigma aan te hang. ’n Sensitiwiteitsanalise word ook met betrekking tot die modelbeperkings en die mate van oorskryding daarvan uitgevoer. In die proses word sommige sagte beperkings geïdentifiseer wat die modeldoelfunksies sensitief beïnvloed.

’n Besluitsteunstelsel, waarvan die werking op die bogenoemde tweedoelige optimeringsmodel gebaseer is, word ontwerp en ’n konsepdemonstrator daarvan word op ’n persoonlike rekenaar geïmplementeer. Hierdie konsepdemonstrator kan gebruik word om afruilingsoplossings vir spesifieke gevalle van die OOP te vind en te analiseer, en die demonstrator bied ook interaktiewe funksies waarmee sensitiwiteitsanalises op ’n baie natuurlike wyse uitgevoer kan word. Die haalbaarheid en praktiese werking van die konsepdemonstrator word uiteindelik gedemonstreer deur dit op twee realistiese gevallestudies toe te pas. Daar word bevind dat the stelsel in elke geval daartoe in staat is om hoë-kwaliteit afruilingsonderhoudskedules te lewer.

Acknowledgements

The author wishes to thank God for blessing him with this immense opportunity and providing for him along the difficult, but rewarding, path taken.

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- My promoter, Prof JH van Vuuren, who has truly promoted me to complete my PhD and without whom I would not have been able to do so. He has tirelessly helped, motivated, and pushed me (beyond what I thought was my limits) in respect of my academic work. I thank him for his tireless editing and improvements to this document. I would also like to thank him for imparting knowledge and appreciation for the other important aspects in life such as philosophy, art, and whisky.
- Jancke Eygelaar, my colleague, friend, sound board and much more during my studies. Thank you for all your input, as well as sharing your knowledge and work regarding this project. I wish you all the best with your studies.
- My research grant sponsor, Eskom, for their interaction and financial support.
- The Bursary and Post Graduate and International Offices at Stellenbosch University for their scholarships awarded.
- The Department of Industrial Engineering, and the SUnORE research group in particular, for all their support and services made available, including the coffee.
- My Dad, Gerald Lindner, who has always supported me, especially financially, to study as far as possible.
- My Mom, Julie Fleming, for her belief in me and always insilling a strong will of hope and perseverance.
- My fellow SUnORE members (more friends), without whom the PhD path would have been much lonelier. I cherish the shared knowledge and laughter, as well as all the good memories during and after office hours.
- Celice, Renier, and Marc, for their support, specifically their editorial help, without you the date on the front of this disseration would have read December 2017.

Table of Contents

List of Reserved Symbols

The symbols listed below are reserved for a specific use, unless specified otherwise in a localised section where its meaning is apparent. Other symbols may be used throughout the dissertation in an unreserved fashion.

Symbols in this dissertation conform to the following font conventions:

a, A	Symbol denoting a **parameter** or **variable**	(Upper or lower case)
$\boldsymbol{a}$	Symbol denoting a **vector**	(Lower case boldface)
$\boldsymbol{A}$	Symbol denoting a **matrix**	(Upper case boldface)
$\mathcal{A}$	Symbol denoting a **set**	(Upper case calligraphy)

GMS model

Symbol	Meaning
Indices	
i	The index of generating units
j	The index of time periods
k	The index of generating unit subsets
Sets	
$\mathcal{I}$	The set of generating units
$\mathcal{J}$	The set of time periods
$\mathcal{K}$	The set of generating unit subsets
$\mathcal{I}_k$	The subset of generating units in subset $k \in \mathcal{K}$
Variables	
x_i	An integer variable denoting the starting time of the maintenance of generating unit $i \in \mathcal{I}$.
X_{ij}	A binary variable taking the value 1 if maintenance of generating unit $i \in \mathcal{I}$ commences during time period $j \in \mathcal{J}$, or zero otherwise
Y_{ij}	A binary variable taking the value 1 if maintenance of generating unit $i \in \mathcal{I}$ is in maintenance at time period $j \in \mathcal{J}$, or zero otherwise
r_j	The net reserve margin during time period $j \in \mathcal{J}$, excluding the safety margin capacity
Parameters	
n	The number of generating units (*i.e.* $n = \lvert\mathcal{I}\rvert$)
m	The number of time periods in the maintenance planning horizon (*i.e.* $m = \lvert\mathcal{J}\rvert$)

e_i	The earliest time period during which maintenance of generating unit $i \in I$ may start
f_i	The latest time period during which maintenance of generating unit $i \in I$ may start
I_i	The power generating capacity of generating unit $i \in I$
d_i	The maintenance duration of generating unit $i \in I$
D_j	The electricity demand during time period $j \in J$
S	The safety margin as a proportion of the demand in the entire power system
μ_i	The maintenance crew required to perform maintenance on generating unit $i \in I$ when in maintenance
μ_i^u	The maintenance crew required to perform maintenance on generating unit $i \in I$ during its u-th period of maintenance
μ^l_{cij}	The maintenance crew required to perform maintenance on generating unit $i \in I$ when in maintenance during time period $j \in J$ if maintenance were to commence during time period $c \in J$
M_j	The maximum maintenance crew available to perform maintenance operations on the entire system during time period $j \in J$
κ_k	The maximum number of generating units within subset I_k that are allowed to be in simultaneous maintenance during any time period

Production planning model

Symbol	Meaning
Indices	
s	The index of power stations
h	The index of power station types
t	The index of time slices
Sets	
S	The set of power stations
S_h	The set of power stations of type h
S_{base}	The subset of base power stations
S_{peak}	The subset of peak power stations
S_{exact}	The subset of exact-energy-required power stations
S_{ps}	The subset of pumped storage scheme power stations
S_{psp}	The subset of pumped storage scheme power stations pumps
T	The set of time slices t
Variables	
Z_{st}	The energy production planned for power station $s \in S$ during time slice $t \in T$
A_{st}	The available generation capacity of power station $s \in S$ during time slice $t \in T$
E_s	The energy availability factor of power station $s \in S$ during the day
P_s	The unavailability proportion due to planned capability loss factors at power station $s \in S$ during the day
f_s^{max}	The maximum daily production requirements at power station $s \in S$
f_s^{min}	The minimum daily production requirements at power station $s \in S$
f_s^{req}	The exact daily production requirements at power station $s \in S_{\text{exact}}$
a_s	The online capacity committed for station $s \in S$

Parameters	
C_s	The energy production cost rate of power station $s \in S$
I_s	The installed generation capacity of power station $s \in S$
U_s	The unavailability proportion due to unplanned capability loss factors at power station $s \in S$ during the day
O_s	The unavailability proportion due to other capability loss factors at power station $s \in S$ during the day
L_t	The load demand forecast for time slice $t \in T$
$q_s^{\max}$	The maximum energy utilisation factors at power station $s \in S$
$q_s^{\min}$	The minimum energy utilisation factors at power station $s \in S$
B_a	The efficiency at pumped storage scheme station $a \in S_{\text{ps}}$
$L_{\max}$	The day's peak hourly demand

Multi-objective optimisation

Symbol	Meaning
Sets	
P_s	The set of Pareto optimal solutions
P_f	The set of objective function vectors in the Pareto front
F	The set of mutually non-dominating solutions found thus far during a search
Parameters	
α	Simulated annealing cooling parameter
β	Simulated annealing reheating parameter
τ	Simulated annealing initial acceptance ratio
γ	Constraint violation severity factor
$t_{\max}$	Budget of computation time allowed before algorithmic termination
$\Omega_{\max}$	Number of successive reheatings allowed before algorithmic termination

List of Acronyms

ACO	Ant colony optimisation
B&B	Branch-and-bound
Btu	British thermal unit
CP	Constraint programming
CSV	Comma separated values
DISCO	Distribution company
DMOSA	Dominance-based multi-objective simulated annealing
DP	Dynamic programming
DPLVC	Daily peak load variation curve
DSS	Decision support system
EAF	Energy availability factor
ED	Economic dispatch
EENS	Expected energy not served
EFS	Energy flow simulator
ENS	Energy not served
ER	Effective/equivalent reserves
EUE	Expected unserved energy
EUF	Energy utilisation factor
FOR	Forced outage rate
GA	Genetic algorithm
GMS	Generator maintenance scheduling
GUI	Graphical user interface
GENCO	Generation company
GSO	Group search optimiser
GSOMP	Group search optimiser with multiple producers
ID	Identification
IDE	Integrated development environment
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input-Output
IP	Integer programming
IPP	Independent power producer
ISO	Independent system operator
kWh	Kilowatt-hours
LDC	Load duration curve
LOL	Loss of load
LOLE	Loss of load expectation
LOLP	Loss of load probability
LP	Linear programming

MIP	Mixed integer programming
MILP	Mixed integer linear programming
MO	Multi-objective
MOEA	Multi-objective evolutionary algorithm
~~**MOO**~~	~~Multi-objective optimisation~~
MOPSO	Multi-objective particle swarm optimisation
MOSA	Multi-objective simulated annealing
MS	Maintenance scheduling
MTBF	Mean time between failures
MTTF	Mean time to failure
MTTR	Mean time to repair
MBtu	Mega British thermal unit
MW	Megawatt
MWh	Megawatt-hours
NSGA-II	Non-dominated sorting genetic algorithm-II
OCLF	Other capability loss factor
PCLF	Planned capability loss factor
PM	Planned maintenance
PPM	Planned preventative maintenance
PRM	Preventative maintenance
PSO	Particle swarm optimisation
RETAILCO	Retail company
RTS	Reliability Test System
SA	Simulated annealing
SO	Single-objective
SSR	Sum of squared reserves
TMS	Transmission maintenance scheduling
TRANSCO	Transmission company
TS	Tabu search
TWh	Terawatt-hours
UC	Unit commitment
UCLF	Unplanned capability loss factor
UI	User interface

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 Background

All things must rest and be subjected to routine maintenance in order to increase their survivability and sustainability. How much rest and/or maintenance is required typically involves a trade-off: too much of it and the entity becomes too unproductive, too little of it and the entity runs the risk of wearing out or breaking down. As important is the timing of the rest/maintenance granted. This is applicable to both natural and manmade entities. This dissertation focuses on preventative *maintenance scheduling* (MS) for one of the most important entities in electric power systems, namely power generating units.

In modern society, energy production is a prerequisite for physical and economic wellbeing. Energy is required for the production of light, heat, and transport, to name but a few examples. Energy may be produced from fuel (*e.g.* oil, gasoline, uranium, gas, coal, or wood) or natural forces (*e.g.* wind, or water) [90]. Providing energy consistently, efficiently, reliably, and sustainably is of crucial importance in modern economies. This is becoming an especially difficult task in view of the globe's growing electrical energy demand (as a result of population growth and development) and depleting resources.

Figure 1.1 contains a map of the world's electricity consumption during 2015 in *terawatt hours* (TWh). As may be seen in the figure, China and the United States of America consume by far the largest amount of electricity worldwide. Figure 1.2 illustrates the steadily increasing trend of electricity consumption around the world. As may be seen in the figure, the Asian market is growing substantially, and the trend for South Africa's electricity consumption is illustrated in Figure 1.3.

In order to meet this demand, a society will build a number of power stations to *produce* energy from different sources, usually within the borders of a country. These power stations are typically interconnected to form a power system or network. Such a power system is further responsible

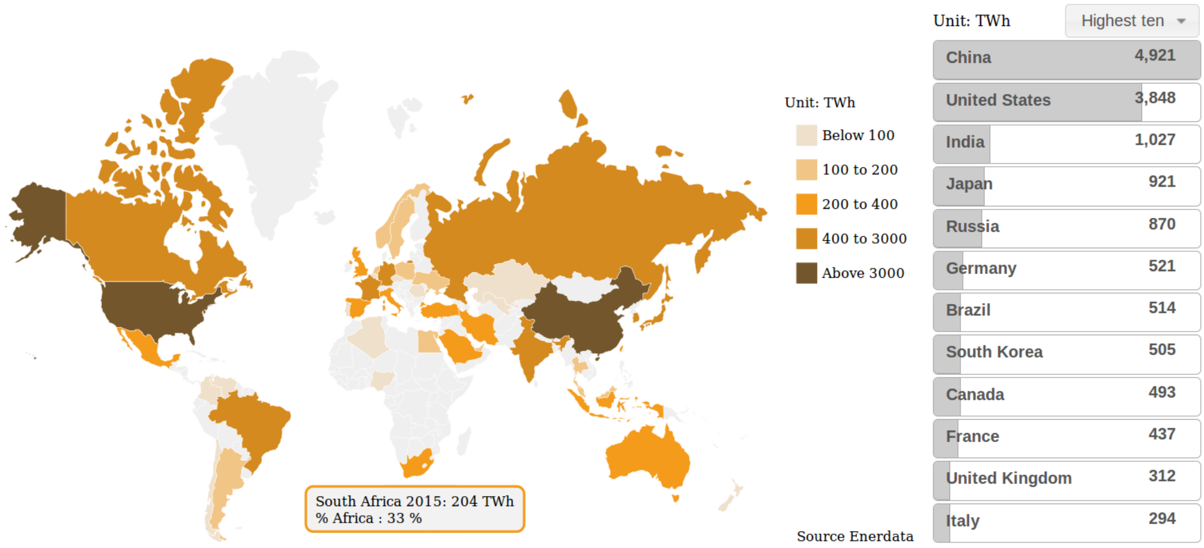


Figure 1.1: *World electricity consumption during 2015, totalling 20 568 TWh [70].*

for the *transmission* and *distribution* of the power *produced* by these stations to industrial and urban end-users.

A power station is complex in nature, requiring a large number of people and various types of equipment for its effective operations. Power stations essentially use some form of fuel (containing potential energy) to create mechanical energy. A machine, called a generator, converts this mechanical energy into electrical energy by using electromagnetic principles. The main types of equipment required to accomplish this (*e.g.* boilers, steam (water) turbines, and generators) are often treated as single entities, called power generating units, in power system reliability analyses. A power station generally contains more than one generating unit [91].

Most power stations around the world burn fossil fuels (such as coal, oil, or natural gas) to generate electricity. Others use nuclear power, but there is an increasing use of cleaner renewable sources for this purpose (such as solar, wind, wave, and hydroelectric sources) [90, 86]. Coal is, however, still the world's most abundant and widely distributed fossil fuel, and it fuels more than 40% of the world's electricity generation, although this figure is much higher in many countries, such as South Africa (93%), Poland (92%), China (79%), India (69%) and the United States of America (49%) [166].

The top five largest power stations in the world are hydroelectric stations, the largest being the Three Gorges Dam station in China with an installed capacity of 22 500 *megawatt* (MW) [205] and consisting of 32 power generating units [207]. Although currently only a proposal, the Grand Inga Dam station in the Congo will surpass all existing power stations in capacity when it comes into operation. If construction commences as planned, the design targets to top 40 000 MW in installed capacity [112, 113]. To put this into perspective, it will be capable of producing (if it were to operate 24 hours a day, 365 days a year, with enough water) 350 TWh, which is 56% of Africa's 621 TWh energy consumption in 2015 (see Figure 1.2). Another proposal, the Penzhin Tidal Power Plant, is designed for an installed capacity up to 87 100 MW [205]. Kusile power station, which is expected to become the world's largest coal-fired power plant upon completion, is currently being constructed in Mpumalanga, South Africa and will comprise six generating units, each rated at an 800 MW installed capacity for a total capacity of 4 800 MW. It is expected that the first of its units will be synchronised in 2017 [76].

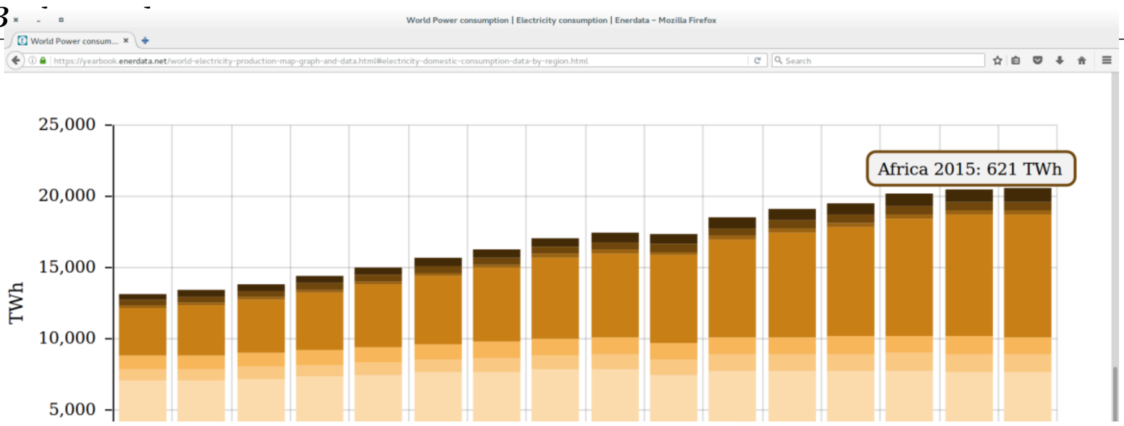


Figure 1.2: *World energy consumption during the period 2000–2015 [70].*

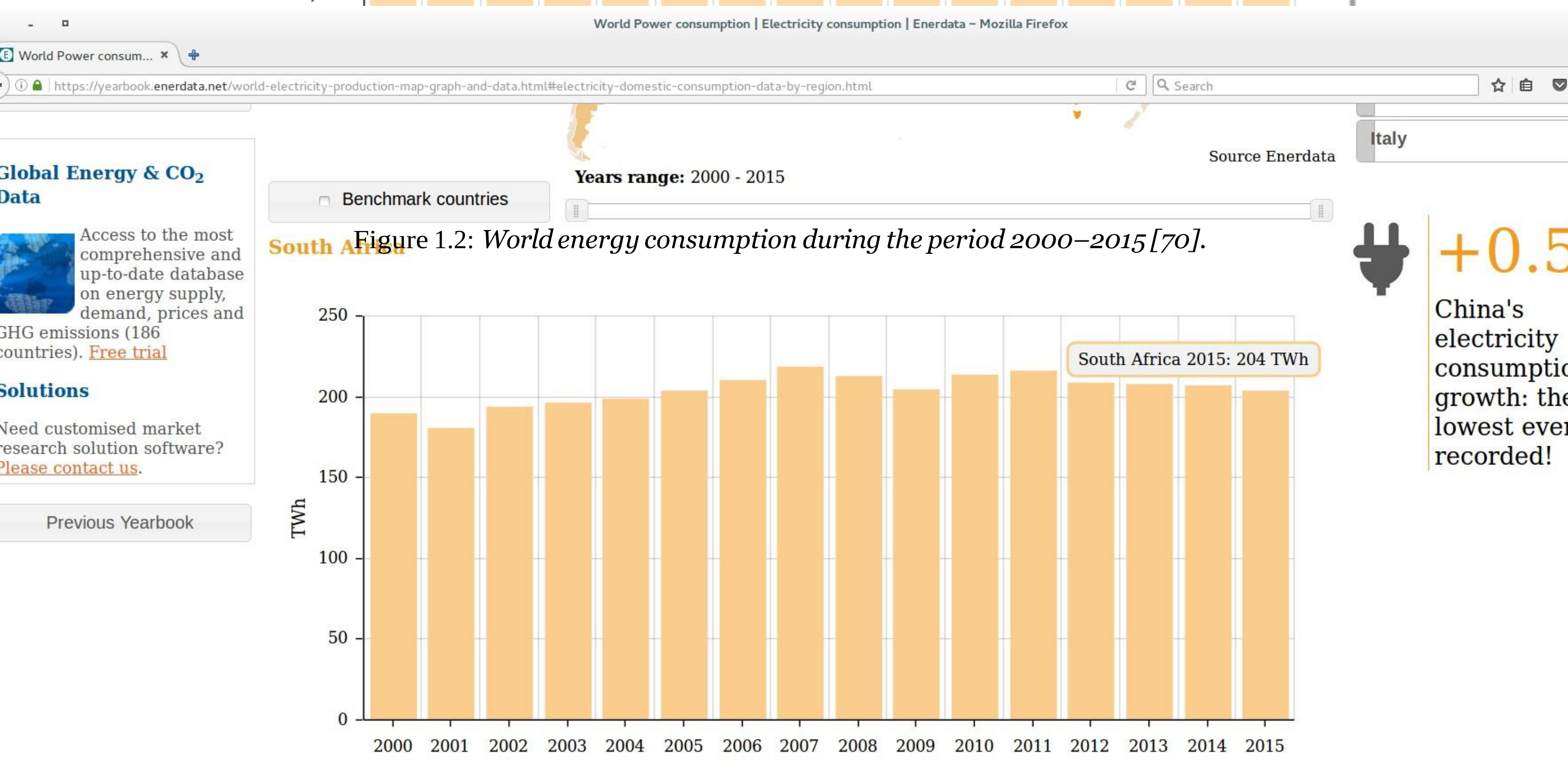
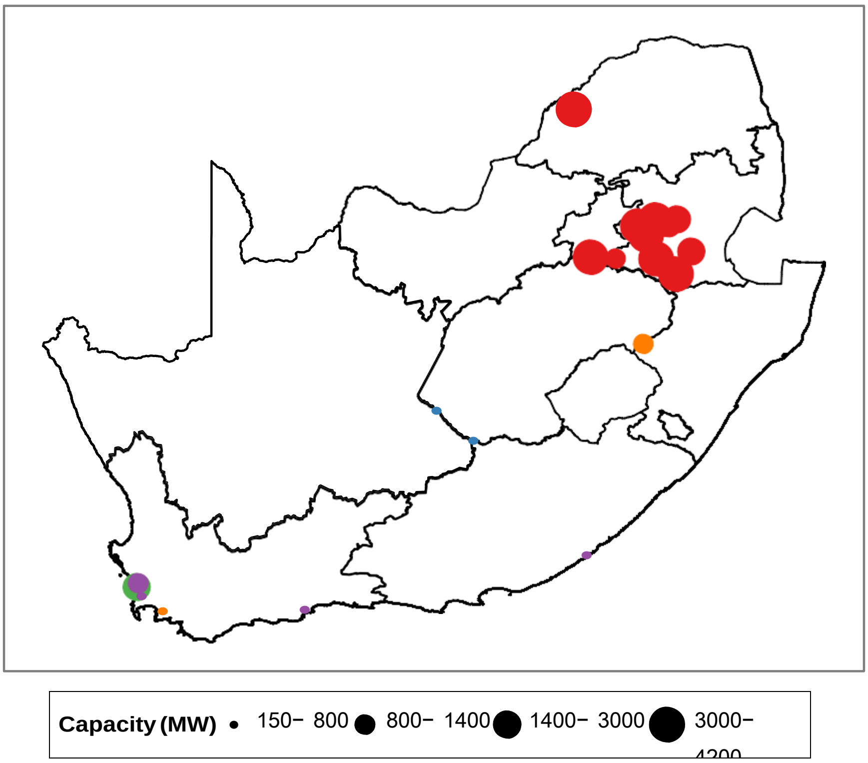


Figure 1.3: *South Africa's energy consumption during the period 2000–2015 [70].*

South Africa has experienced serious energy problems of late [180]. Eskom is the South African state-owned electricity utility which *generates*, *transmits* and *distributes* electricity in all sectors of South African society. It generates approximately 95% of the country's electricity [73]. The utility predominantly generates electricity through coal-fired power stations, as illustrated in Figure 1.4. Two coal-fired power stations in South Africa may be seen in Figure 1.5.

Power outages in South Africa are mainly brought about by higher than expected demand, infrastructure failure or vandalism, and a diminishing reserve capacity (available capacity over and above demand). The reserve margin for generating capacity has decreased in recent years from the desired 15% to less than 8% [123]. As a result, South African power stations have recently been forced to operate virtually continuously at very high load factors (how hard a plant is operating on a percentage basis). In addition, the generating units of South African power stations are relatively old, which means that they require above-average levels of maintenance. These two aspects contribute significantly to the prevalence of unplanned power outages [168, p. 5]. Appropriate *preventative maintenance* (PRM) planning is crucial to mitigate the risk of unplanned outages and is one of the key focus areas for the management of a power utility



Type	Coal	Hydroelectric	Nuclear	OCGT	Pumped storage	
Number	13 (60%)	2 (9%)	1(5%)	4 (18%)	2 (9%)	22
Capacity (MW)	37678 (86%)	600 (1%)	1800 (4%)	2426 (6%)	1400 (3%)	43904

Figure 1.4: *The distribution of Eskom's power stations in South Africa in 2015 according to Hatton [104, p.11]. OCGT is an acronym for open-cycle gas turbines.*

[46, 49, 124, 145] — especially for a power utility such as Eskom, which has been postponing maintenance plans on its already ageing stations [53]. Eskom has of late attempted to turn the tide on unplanned power outages by focussing its attention on planned maintenance during 2015 and 2016 [110], and "this has resulted in a reduction of the number of plant breakdowns over the past seven months, positively impacting plant availability" [77].

1.2 Informal problem description

As mentioned and motivated in the previous section, effectively planning PRM of the power generating units in the power system of a power utility is important, so as to satisfy demand as efficiently and effectively as possible [46, 49, 90, 124, 145]. MS plays an important role in improving the overall availability and extending the life of equipment [1] in power stations. It is also crucial in risk management [68]. As the number of generating units of a power utility is typically large and the operational constraints over the planning horizon are typically complex in nature, there is a growing need for developing new methods for planning PRM of power generating units [161].

A schedule for the planned maintenance outages of generating units in a power system is sought in the celebrated *generator maintenance scheduling* (GMS) problem [165]. A novel bi-objective

(a) *Arnot power station in Mpumalanga, consisting of six ± 390 MW units adding up to 2 352 MW*

(b) *Lethabo power station in the Free State, consisting of six 618 MW units adding up to 3 708 MW*

Figure 1.5: *Two coal-fired power stations in South Africa [79].*

approach to modelling the GMS problem is proposed in this dissertation as well as how the solution to this problem may be incorporated into a national power utility's decision support software. It is envisaged that this modelling approach may, for example, be incorporated into Eskom's existing *Energy Flow Simulator* (EFS).

Power utilities across the world often employ such large-scale energy flow simulation models of their energy supply chains to inform decisions on operational and strategic levels. Foley *et al.* [85] provide an overview of electricity systems modelling techniques and discusses a number of key proprietary electricity system models used in the United States of America and Europe. These energy system models typically interconnect the conversion and consumption of energy [172], including operations involved with primary fuel supplies (*e.g.* mining and petroleum extraction), conversion and processing (*e.g.* power plants and refineries), and end-use demand for energy services (boilers and residential space conditioning). The demand for energy is normally disaggregated by sector (*i.e.* residential, manufacturing, transportation, and commercial) and by specific functions within a sector (*e.g.* residential air conditioning, heating, lighting, and hot water) [172]. These energy flow models usually serve to facilitate the investigation of what-if scenarios by decision makers [141], with some utilising optimisation techniques [71, 172]. It is within this decision support framework that MS solutions should ideally be incorporated in a dynamic fashion.

Figure 1.6 provides an overview of the typical energy flow in South Africa. The EFS is a decision support software tool which models this energy supply chain from "fuel to fridge" [104, p. 3]. The working and various constituent components of this simulation tool are illustrated in Figure 1.7. The EFS focusses primarily on coal-generated energy (but also includes other energy sources), since coal is the predominant energy source in South Africa [104, 105, 131]. The EFS has been developed to function as a what-if analysis tool in the context of different future scenarios. It allows for the accommodation of different energy availability factors per station, different weather patterns, a variety of *gross domestic product* levels, varying coal supply levels and qualities, *etc.* The only simulation technique employed in the EFS is Monte-Carlo simulation.

A simulation replication by the EFS is initiated by calling a *consumption module* (see Figure 1.7(a)), which forecasts the national energy demand per region (central, eastern and southern) and customer type (residential, manufacturing, mining, and other) according to the selected level of gross domestic product (high, medium, or low) and weather scenario (hot, normal or cold).

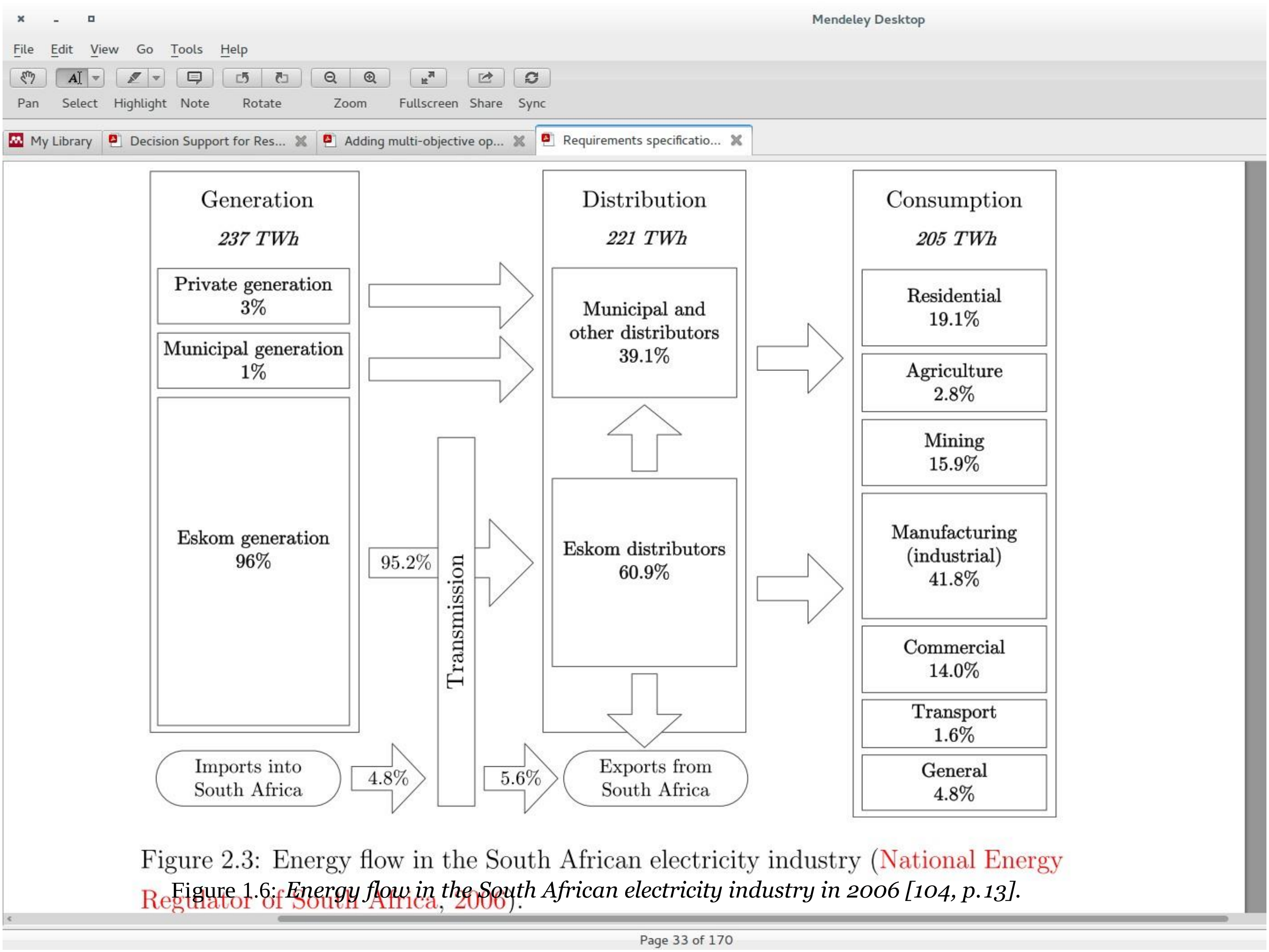


Figure 1.6: *Energy flow in the South African electricity industry in 2006 [104, p.13].*

The demand thus forecast is then used by a *production planning module* (see Figure 1.7(b)) which employs a *linear programming* (LP) model to schedule the planned energy production per power station (including coal, nuclear, gas-turbine, hydro-electric, and renewable energy units) so as to minimise energy production cost. Demand must be met whilst taking into account production capacity. The third main component of the EFS is a *primary energy module* (see Figure 1.7(c)), which deals primarily with coal, because coal-fired power plants produce most of the utility's electricity. The primary energy module facilitates what-if analyses in terms of a variety of different plans and scenarios, including unplanned power station maintenance, variation in the calorific value of coal, variation in the quantity of coal delivered, and variation in the coal burnt at each coal-fired power station. The final main component is a *generation module* (see Figure 1.7(d)). The energy production plan, supply reliability, and the quality and quantity of coal to be delivered are taken as input into the generation module which then quantifies emissions, such as sulphur and nitrogen oxides. System losses are also incorporated into the EFS [105].

Until recently, the EFS had no optimisation capacity (other than solving an LP problem in its energy production planning component (in Figure 1.7(b)). Many variable and parameter values currently employed within the EFS are known to be sub-optimal, and a need has therefore arisen to be able to optimise decision variables within the EFS [104]. Two masters projects were subsequently carried out and finalised in 2015 [104] and 2016 [25] in order to add optimisation capabilities to the EFS. The first masters project by Hatton [104] involved the design of a *single-objective* (SO) optimisation component, using the cross entropy method (a metaheuristic), to determine good coal stockpile management policies for the *primary energy module* (in Figure 1.7(c)). In this study, the model minimised the cost related to coal on hand, coal shortages,

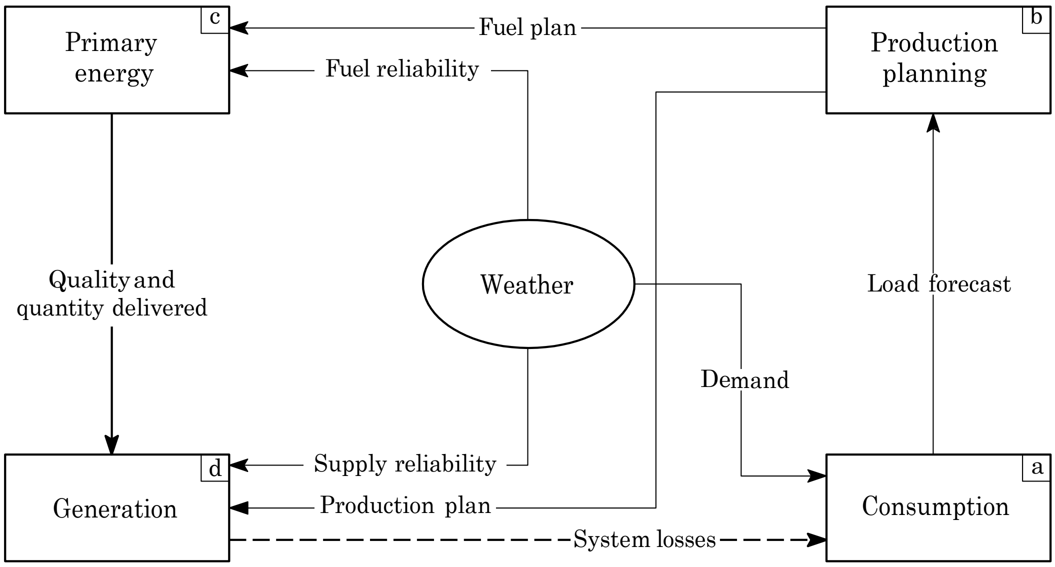


Figure 1.7: *High-level representation of the EFS (adapted from [105]).*

emergency deliveries, and delivery cancellations. In addition, Hatton investigated and proposed future areas for simulation and optimisation within the EFS. The second follow-on masters project by Brits [25] further centred around a *multi-objective* (MO) coal inventory model for Eskom's network of coal-fired power stations using the simulation outputs of the EFS in conjunction with an MO version of the cross entropy method. The objectives in the MO paradigm considered by Brits [25] included minimising the total average coal stokpile levels, the total coal transfers, and the total average coal inventory outside the warning limits. In addition, Brits [25] identified and modified some of the existing EFS architecture in order to improve its potential as a simulation and optimisation tool. These improvements included a conversion of the weekly resolution of the production planning module to a daily resolution.

In 2011, Schlünz [168] carried out a masters project which determined optimal generator maintenance schedules for a national power utility such as Eskom. In this study, only one scheduling objective was considered, namely levelisation of net reserve margins based on a maintenance schedule as input over the entire planning horizon.

The GMS modelling approach proposed in this dissertation is designed specifically to allow for its incorporation into the above-mentioned improved EFS decision support framework, by incorporating it into the improved daily production planning module (of Figure 1.7(b)) developed in [25]. Furthermore, the modelling approach proposed in this dissertation is bi-objective in nature, with the one scheduling objective seeking to levelise the net reserve margins over the entire planning horizon, very similar to the work done by Schlünz [168], while the other scheduling objective seeks to minimise the expected generation cost, based on the improved daily production planning module (of Figure 1.7(b)) developed by Brits [25].

1.3 Dissertation scope and objectives

The following nine objectives are pursued in this dissertation:

I To *conduct* a thorough survey of the literature related to:

(a) MS problems in general,

(b) related problems in the energy sector,

(c) models for MS of generating units in particular,

(d) the nature and appropriate ranges of parameters required for maintenance of generating units,

(e) *multi-objective optimisation* (MOO) studies in respect of the maintenance of generating units and the solution of related energy problems, and

(f) any other data required to generate instances of the bi-objective GMS problem described in §1.2.

II To *establish* a suitable framework for evaluating the effectiveness of a given generator maintenance schedule for a power utility in terms of the reserve margin associated with these generating units not in maintenance and the cost of producing energy by the generating units not in maintenance.

III To *formulate* a bi-objective GMS model suitable for use as a basis for decision support in respect of the MS of the generating units of a power utility. The model should take as input the parameters and data identified in Objectives I(c)–(d), and function within the context of the framework of Objective II.

IV To *design* a generic *decision support system* (DSS) capable of suggesting trade-off maintenance schedules for user-specified instances of the bi-objective GMS model of Objective III.

V To *implement* a concept demonstrator of the DSS of Objective IV in an applicable software platform. This concept demonstrator should be flexible in the sense of being able to take as input an instance of the bi-objective GMS model of Objective III via user-specification of the parameters and data mentioned in Objectives I(c)–(d) and produce as output a set of trade-off maintenance schedules for that instance.

VI To *verify* and *validate* the implementation of the concept demonstrator of Objective V according to generally accepted modelling guidelines.

VII To *apply* the concept demonstrator of Objective V to a special case study involving realistic GMS parameters.

VIII To *evaluate* the effectiveness of the DSS and associated concept demonstrator of Objectives IV–VI in terms of its capability to identify high-quality trade-off solutions to instances of the bi-objective GMS model of Objective III and to identify possible improvements achievable by relaxing model constraints in the form of a sensitivity analysis.

IX To *recommend* sensible follow-up work related to the work in this dissertation which may be pursued in future.

The scope of the dissertation shall be restricted to the GMS problem, but will take cognisance of solutions to the related *unit commitment* (UC) and *economic dispatch* (ED) problems in order to estimate the (deterministic) energy production cost associated with a maintenance schedule. Forced/unplanned outages and transmission losses will not be taken into account.

1.4 Dissertation organisation

This introduction is followed by eight additional chapters, a bibliography and a number of appendices. In Chapter 2, a background on general maintenance activities is presented. The environment in which power systems operate and the typical problems prevalent in the energy industry are discussed. General GMS modelling considerations are presented. A comprehensive literature review on GMS model formulations and model solution approaches is also presented. The final section of the chapter contains a discussion on studies related to energy-related problems utilising MOO principles and the very few GMS studies found in the literature employing truly MOO techniques.

The newly proposed GMS model is formulated in Chapter 3. The decision variables are defined in conjunction with the constraints and the two scheduling objectives. A presentation finally follows of how the proposed model may be incorporated into an EFS.

In Chapter 4, a number of basic notions in MOO are discussed. A description follows of various algorithms suited to GMS and the solution of MOO problems. Reasons are also given for the algorithm selected for implementation in this dissertation. Information is finally provided on the working of this algorithm, and its implementation is elucidated.

In Chapter 5, data pertaining to two case studies employed in this dissertation to test the effectiveness of the proposed GMS model are presented. The first case study is a GMS benchmark system commonly found in the literature, whilst the second case study is a much larger instance designed for the South African power utility, Eskom. All the data required to solve the GMS instances embodied in these two case studies are presented.

Chapter 6 contains the results of an extensive parameter optimisation experiment aimed at uncovering suitable parameter values for the SA algorithm in the context of the two case studies of Chapter 5.

In Chapter 7, numerical results are presented in respect of the verification and validation of the GMS model proposed in Chapter 3 in conjunction with the performance of the SA algorithm. These results are also compared to another common and well-performing algorithm in the MOO literature. Reasons for GMS trade-offs are presented and a sensitivity analysis is conducted in respect of the constraint right-hand sides. The chapter closes with a number of suggested improvements to the MOO algorithm, including the use of parallel computing.

The penultimate chapter of this dissertation, Chapter 8, focusses on the implementation of a DSS for solving instances of the GMS problem. The chapter opens with a discussion on the basic notions that have to be borne in mind when developing DSSs. The relatively scant work in the literature related to DSSs in the context of the GMS problem is presented, as well as an overview of DSSs currently in use in the energy industry. The design and implementation of a novel DSS for GMS is proposed, and this is followed by a description of a concept demonstrator of the proposed DSS. Feedback from Eskom after having been presented with this concept demonstrator is also reported.

The dissertation closes with a summary of the work conducted and important findings, together with an overview and appraisal of the contributions made. Suggestions for future work related to GMS is finally presented.

CHAPTER 2

Literature review

Contents

This chapter opens with a discussion on various types of maintenance strategies. This is followed by a description of the energy industry environment and typical activities carried out in this environment. General modelling considerations for the GMS problem are presented next. A more substantial literature review follows on GMS problem formulations, including typical objectives and constraints found in the literature. This is followed by a review of GMS solution approaches and techniques applied, predominantly to SO GMS models but also includes some simple MO analyses. The chapter closes with a discussion highlighting Pareto-based MOO modelling approaches found in the literature in the context of energy problems.

2.1 Maintenance

Maintenance generally involves planned and unplanned actions carried out to retain a system in order to restore it to an acceptable condition [176]. There are a variety of different (and often somewhat conflicting) types and classifications of maintenance in the literature. Some authors classify the different types of maintenance strategies or activities into two [5], three [134, 155], four [12] or five types [94], each with their own advantages and disadvantages [94, 155, 134]. Table 2.1 illustrates the different types of maintenance activities considered in the literature, while Figure 2.1 contains an example of a classification structure for the different maintenance strategies.

In the earlier literature, the only type of maintenance considered was *breakdown maintenance*, also called *unplanned maintenance*, *reactive maintenance*, *corrective maintenance* or *run to failure maintenance*, which takes place only after breakdowns have occurred [114, 134]. This type of strategy usually results in unscheduled down-time, typically at a high cost [155]. In this type of

Table 2.1: *Different classifications of maintenance types/strategies.*

Maintenance					Reference
Corrective	Preventive				[5]
Corrective	Periodic/ preventive	Predictive			[155]
Unplanned	Planned	Condition-based			[134]
Run to Failure	Preventive	Condition-based	Design improvement		[12]
Reactive	Preventive	Inspection	Backup	Upgrade	[94]

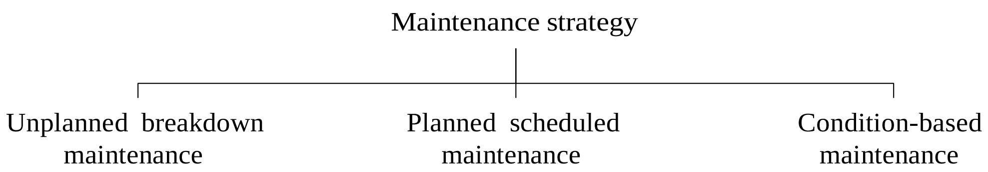


Figure 2.1: *Classification of different types of maintenance activities according to Martin [134].*

strategy the maintenance crew's duties might range from making adjustments to machinery to getting a facility back on line. Loss control is the most critical aspect of the *reactive maintenance* crew's job. In many cases, temporary repairs may be made so that the facility may return to function quickly and be repaired correctly, or permanently, at a later time [94].

Planned maintenance (PM) activities are those activities that are undertaken in order to reduce the probability of unexpected breakdowns — their expected execution times, durations and contents are predetermined [134, 153]. PRM may be defined as comprising activities for which expected execution, frequencies, job contents, durations and resource requirements do not vary too much over time [153] or maintenance activities that reduce the probability of breakdown, by replacing worn-out components on a timely basis [94]. PRM is typically carried out regardless of the health status of a physical asset [114]. Some authors classify PRM as *planned preventive maintenance* or *scheduled maintenance* [206], or as *time-based preventive maintenance* [114]. As mentioned, these schedules are usually predetermined (based on, for example, the mean time between failures of certain components) or in more critical cases, schedules may be specified by legal or governmental bodies or other policy setters, in which case the amount of usage of the equipment (such as number of cycles, total operating hours, mileage) is typically used to determine regularly scheduled PM intervals [94].

If it is possible to predict the failure of the system components sufficiently in advance, a process called *predictive maintenance*, then the performance of the components may be optimised and enhanced while simultaneously reducing the maintenance expenditure [155]. One way of predicting failure in the future involves constantly monitoring the system components for symptoms, analysing the trend frequency of these symptoms, and making decisions as to the existence, location, cause, and severity of faults [155]. Martin [134] defines *condition-based maintenance* as PM based upon measuring the condition of all machine components during the normal operation of the machine. Certain characteristics (such as temperature, vibration, or cracks) are monitored continuously or periodically using special equipment [153]. These measurements facilitate the prediction of the time to failure for all elements and thus allow for maintenance to be planned before any elements fail. According to Gallimore and Penlesky [94] *inspection* activities may alter a PRM schedule by identifying instances in which maintenance should be performed either

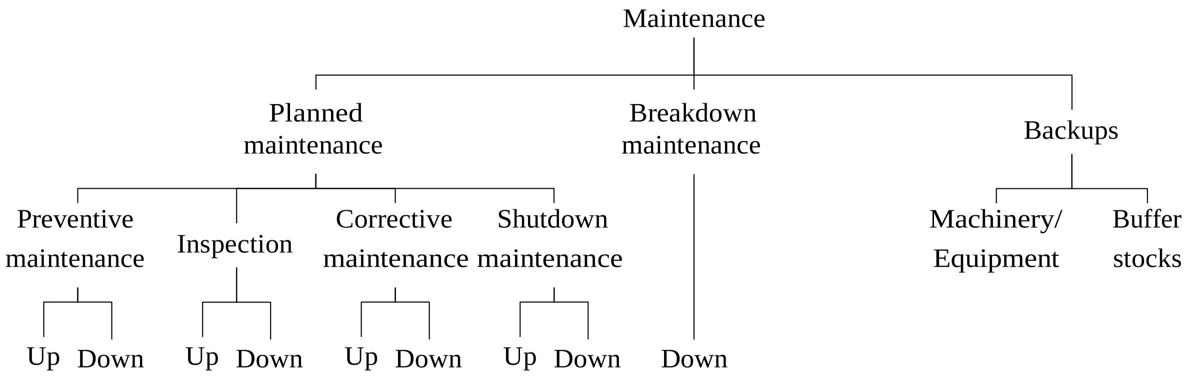


Figure 2.2: *Classification of maintenance activities according to Or [153].*

earlier or later than the regular PRM schedule would dictate.

A simple example of a PM program is maintenance for automotive vehicles, where time and distance determine fluid change requirements. Similarly, a good example of condition-based maintenance is the oil pressure warning light that provides notification that a driver should stop the vehicle, because the engine lubrication is insufficient and failure will most probably occur [206].

Backup equipment may be used as a maintenance element when either the cost or the risk of breakdown is extremely high (*e.g.* the multiple engines on a commercial aircraft). Backup is also effectively applied to operations in which time constraints do not permit proper PRM or repair procedures to be carried out [94]. Equipment may be *upgraded* when the existing equipment is redesigned or modified rather than being replaced by new machinery [94].

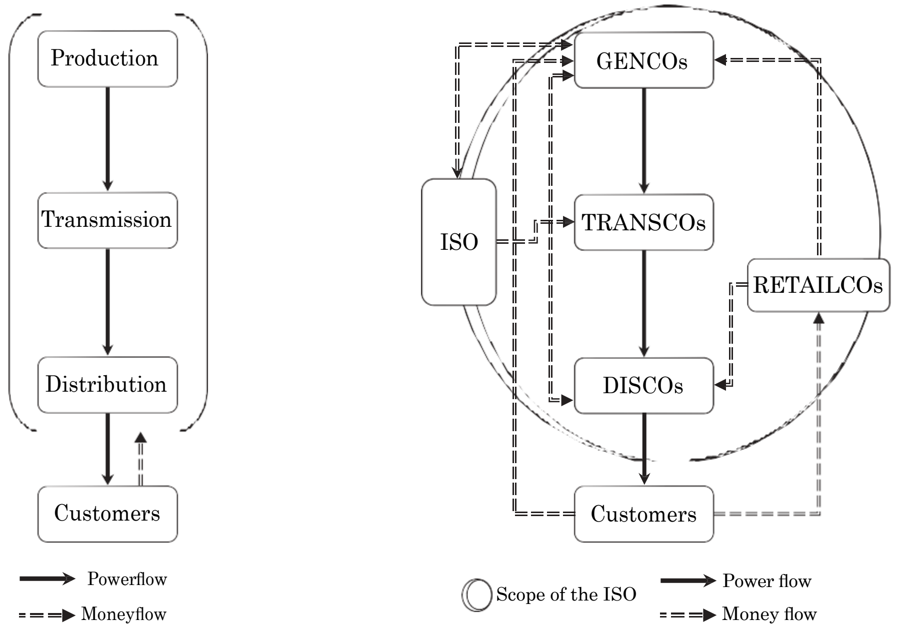
Or [153] has proposed a slightly different classification for maintenance activities, as illustrated in Figure 2.2. Notably, he distinguishes between corrective and breakdown maintenance. *Inspection* activities are similar to PRM activities, but they have shorter durations and fewer resource requirements and they trigger corrective maintenance activities. Since these two activities are so similar, Or [153] considers PRM to include *inspection* activities. *Corrective maintenance* activities are those in which expected execution times, job contents, durations and resource requirements are predetermined, based on *inspection*, *breakdown maintenance* reports, and warnings from production units. They may, therefore, vary considerably from one execution to the next. *Corrective maintenance* activities usually require the related machinery to be down during their execution. Or [153] regards *predictive maintenance* as a special type of corrective maintenance. In *shutdown maintenance*, the entire facility is shut down for an extended period of time so that maintenance may take place [153].

MS plays a very important part in power systems [68], since other planning activities are directly affected by it [125]. In power systems, the major components (including generators and transmission lines) require periodical maintenance [1] and one of the main decision variables in the problem of power plant maintenance is the MS of generating units [32]. While there are a number of different, overlapping and somewhat conflicting definitions of the various types of maintenance activities in the literature, the GMS problem is almost always described as involving the PRM of the power generating units in a power system [46, 49, 90, 124, 145]. This dissertation deals specifically with this type of *planned preventative maintenance* (PPM) in order to prevent generating unit failure or breakdown. Condition-based maintenance has, however, been applied more recently (2016), to determine optimal maintenance schedules for power generating units [160]. PM incurs a considerable expense in power systems, because it requires shop facilities, skilled labor, keeping records, and stocking of replacement parts. The cost of downtime resulting from avoidable outages may, however, amount to ten or more times the actual cost of system component repair [186].

2.2 Power systems

The energy industry is focussed on three main activities, namely *production*, *transmission*, and *distribution* [45, 90, 201]. Traditionally, the industry has been organised in the structured manner illustrated in Figure 2.3(a), where a single entity has monopoly over the entire energy system [45, 90]. Since the latter 1990s, however, deregulation of the power industry has opened up the electricity market to competition [45, 90]. This development is mainly due to the need for more efficiency in power production and delivery [154].

In restructured competitive power systems there are typically a number of independent entities, including *generation companies* (GENCOs), *transmission companies* (TRANSCOs), *distribution companies* (DISCOs), and *retail companies* (RETAILCOs), as illustrated in Figure 2.3(b) [90, 95]. An *independent service operator* (ISO) is typically responsible for the reliability and security of the power system. It dispatches all or part of the energy transactions and can decrease loads on the network in order to avoid congestion [90]. In addition to the above-mentioned entities, there are also other actors in such a competitive system, but their roles are typically minor [90].



(a) *Regulated systems* (b) *Deregulated systems*

Figure 2.3: *Interactions for regulated (a) versus deregulated (b) power systems [90].*

Many countries like England, the United States of America, Canada, Australia, New Zealand, Chile, Argentina, Peru, Colombia, and the Scandinavian region, have already deregulated their electricity industries. Although there are some disadvantages associated with deregulating the market, there are some who claim that it is better than regulated power systems [154]. South Africa still has a regulated power system that is solely controlled by the parastatal electricity company Eskom [75].

2.3 Related energy problems

There are a variety of scheduling problems that have to be solved simultaneously in the management of power systems. Due to the complexity of the different time-scales involved, uncertainties of different orders of magnitude, and dimensionality difficulties in power systems, these problems are, however, usually decomposed into scheduling subproblems which are solved separately [216]. This includes long-term scheduling (*e.g.* MS and nuclear refueling), mid-term scheduling (*e.g.* production scheduling and hydro scheduling), and short-term scheduling (thermal UC, short-term hydro, and ED) [216]. MS is an important overarching strategic planning operation in the energy sector which affects all the main activities in the energy sector, in the sense that all the above-mentioned energy-related problems all take a maintenance schedule as input [67].

2.3.1 The unit commitment problem

The UC problem seeks to determine which available generating units (*i.e.* those not scheduled for maintenance) should be connected to the power generation system, so as to contribute actively to power generation. The verb "commit" may here be interpreted as turning a power generating unit on (that is, bringing the unit into operation), synchronizing it with the system, and connecting it so that it can deliver power to the network [212]. The reason for not simply committing all the available power generating units, thus ensuring maximum expectation of satisfying demand, is the possibly exorbitant cost of keeping a power generating unit online unnecessarily; considerable cost saving is often possible if a power generating unit does not have to be committed [212].

Usually the objectives of the UC problem include minimising operating costs, minimising emissions, or maximising the demand satisfaction capability [173]. The objective of minimising operating cost typically consists of minimising production cost, maintenance cost, start-up cost and shut-down cost. The production cost is usually determined by solving the ED problem, a typical subproblem of the UC problem [173, 212]. Instances of the UC problem are usually complex combinatorial optimisation problems [197]. Methods that have been used to solve such instances (exactly or approximately) include extensive enumeration, heuristic priority list (merit order) scheduling, *dynamic programming* (DP), Lagrangian relaxation, the *branch-and-bound* (B&B) method, expert systems/artificial neural networks, *simulated annealing* (SA), *genetic algorithms* (GAs), and network programming [173].

The UC problem is very similar to the GMS problem, but differs from it in that the UC problem is typically applicable over shorter time horizons (usually days or weeks, whereas in the GMS problem the planning horizon usually spans weeks or years) resolved into shorter decision periods (usually hours or days, whereas in the GMS problem it is usually days or weeks) [91]. Instances of the UC problem also require more accurate forecasted demand than do those of the GMS problem. MS of generating units (and transmission lines) affects UC and ED schedules. The GMS problem should therefore ideally be solved in conjunction with the UC problem, but it is often solved independently, in which case the GMS problem's solutions are used as (availability) constraints for the UC problem [90, 168].

2.3.2 The economic dispatch problem

The celebrated ED problem seeks to determine the optimal output from available generating units, so as to meet the expected demand at the lowest possible cost, subject to various constraints. The ED problem is typically modelled as a subproblem of the UC problem [212]. The

UC problem deals with how many units to commit over a time period (usually incorporating integer variables) whereas the ED problem deals with how much power/energy each power generating unit that has actually been committed should deliver (usually by the incorporation of continuous variables). If the ED problem objective is formulated as a nonlinear function, it can be solved by applying the Kuhn-Tucker method, the lambda iteration (search) method, or DP (usually involving nonconvex curves) [212]. If, on the other hand, the ED problem's objective is formulated as a linear or piece-wise linear function, the ED problem can be solved by standard LP methods. The ED problem usually seeks to minimise fuel costs, which is also a common objective in the GMS problem.

2.3.3 The transmission maintenance scheduling problem

The *transmission maintenance scheduling* (TMS) problem is similar to the GMS problem, but in the TMS problem the transmission lines of a power system have to be maintained, whereas the generating units of a power system have to be maintained in the GMS problem. If generator units are in maintenance, the transmission lines connected to them are not in use and thus there may be an opportunity to perform transmission line maintenance on them. The GMS and TMS problems should therefore ideally be solved jointly as was done in [1, 68, 135, 137], although these problems are often solved independently in the literature [29, 32, 42, 90, 108, 10, 126, 147, 169].

2.4 General GMS model considerations

MS problems may be described as seeking to determine in advance a fixed preventative maintenance schedule over a certain time horizon that satisfies the system constraints and optimises some objective function [68].

Typical decisions in GMS problems include when to start maintenance on a unit, when to shut down a unit for maintenance, when to re-start it again, and how much resources (*e.g.* technicians) to assign to the maintenance of a given unit during a given period [68, 90].

The GMS problem is well known in the *operations research* literature. Although the GMS problem is related to a number of classical optimisation problems, such as the *assignment problem*, the *travelling salesman problem* and the *vehicle routing problem*, it is not one of these [168, p. 12]. Factors complicating formulations of the GMS problem result from attempts at incorporating *"the peculiarity of maintenance scheduling"* [124] as consequence of the following power system features: the fact that generated electricity cannot be stored; that the transmission network is limited and hence that a required amount of electricity must be generated at every instant; that an adequate amount of reserve capacity has to be available at all times; and the parallel nature of electricity supply within a power system (due to multiple generating units) [124].

The GMS problem is combinatorial in nature and is usually nonlinear, making it a difficult problem to solve. Furthermore, the complexity of planning preventative maintenance is due to the enormous size of typical systems to be modelled. There are usually many variables, including binary variables, in GMS models which renders resolution of the problem difficult. This usually constrains the level of detail represented in models so that computational efficiency is achieved [32]. As noted by Canto [32], "*balancing complexity, problem size, model calculating time, and reality approximation level is essential*" in GMS problems (and related power system PRM problems).

GMS problem instance solution times in the literature range from a few minutes [145, 167, 170] to days [167], depending on the problem instance, the algorithm used, and the computing resources available. In [122] it is explained that if such a schedule is worked out manually by engineers it could take days or even weeks to schedule an instance involving 19 generating units over a six-month planning horizon. GMS is usually planned for entire years and thus the typical length of algorithmic computation time (if less than a few days) is not of that much concern. A more important concern is the quality of the solutions obtained [167].

2.4.1 GMS problem decision variables

The GMS problem is typically formulated as a scheduling problem with binary decision variables representing whether or not maintenance of the various units should occur during each of a set of time periods. The number of independent variables in a GMS model is therefore determined by the number of generating units to be scheduled for maintenance and the number of time periods into which the scheduling window is discretised [147]. The number of units in GMS model formulations considered in the literature range from 5 [89], to 21 (a standard GMS benchmark system in the literature) [49] to as many as 157 (an Eskom case study) [170]. Maintenance plans usually span an annual time horizon [32, 68, 82, 89, 126], but this can vary, and planning horizons in the literature range from eight weeks [67] to five years [146]. Common time periods adopted in the GMS model formulations are one week [68, 126, 124, 169], but this also varies, with values ranging in the literature from single days [89, 170] and five-day periods [107] to monthly periods [4].

A maintenance schedule obtained for a national power utility, containing 157 generating units over 365 days is illustrated in Figure 2.4.

2.4.2 GMS problem parameters

Typically the inputs, and thus data required, to solve instances of the GMS problem include the installed capacity per unit, maintenance time window lengths (earliest and latest starting times within which maintenance of each unit should occur), the duration of the maintenance of each generating unit, the resources required to service each unit and the expected energy demand. Most of these parameters are pre-determined or estimated well in advance of the activity of determining maintenance schedules [91]. The parameters for a typical GMS benchmark system [49] in the literature is illustrated in Table 2.2, whilst the weekly peak load demands for another GMS benchmark system [49] are shown in Table 2.3.

Maintenance windows for the generating units

Each unit may have its own desirable maintenance time period or “window” [149]. This may be adjusted based on the decision maker’s preference or may be excluded/relaxed entirely from consideration. These values are typically based on the previous year’s maintenance schedule, allowing for a minimum and maximum time to occur between consecutive maintenance outages of a unit [65]. An outage that takes place too soon wastes money, since needless maintenance is performed. Waiting too long, on the other hand, may incur additional expenses, since the unit’s availability deteriorates owing to increased forced outages and repairs become more expensive. These costs may be quantified and incorporated as scheduling objective or may simply be specified as subjective maintenance window constraints for each unit [149].

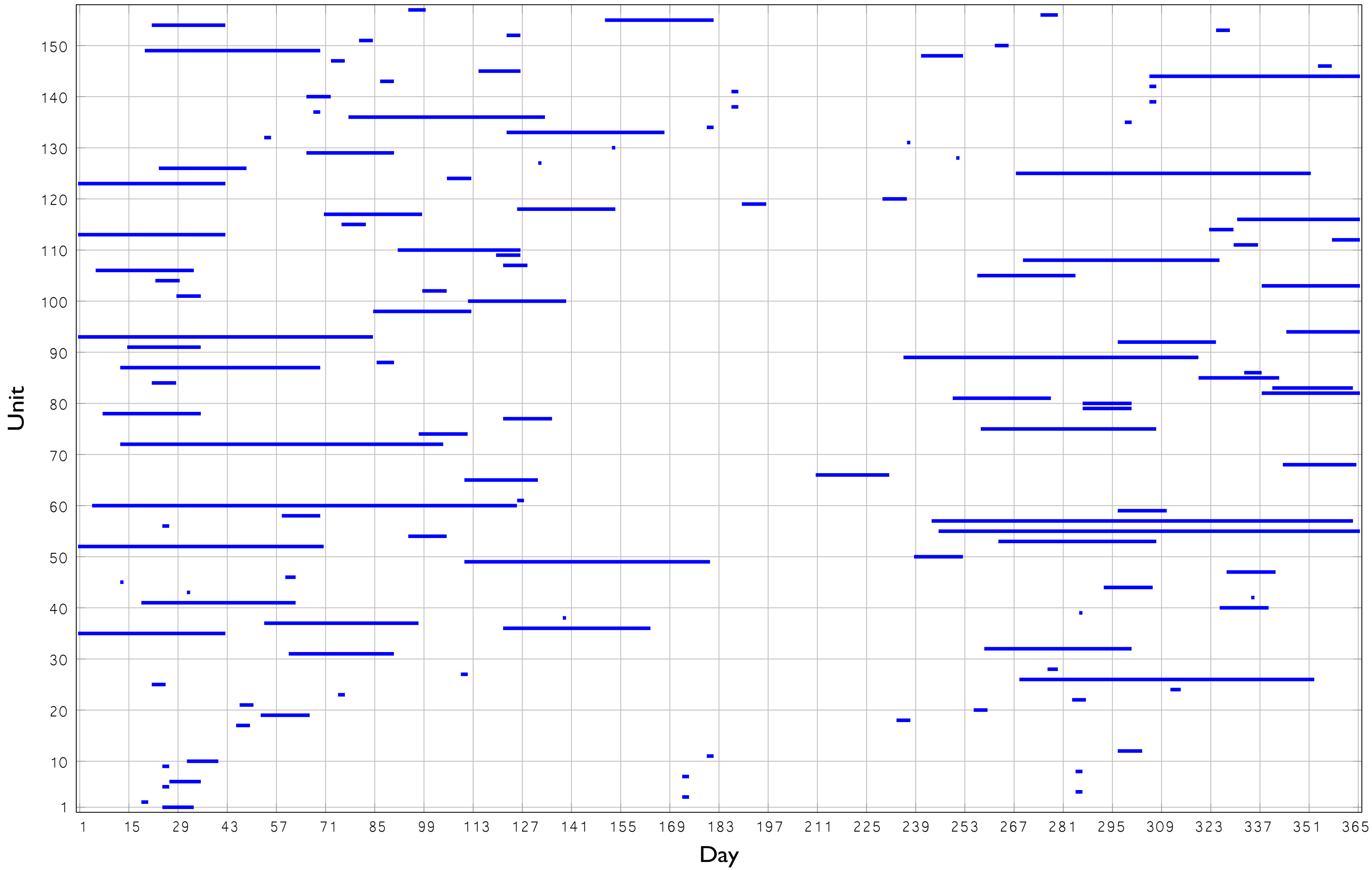


Figure 2.4: *Maintenance schedule proposed in a South African national power utility case study [170]. The Gantt chart indicates during which time periods the various units should be scheduled for maintenance.*

Figure 7.9: *The best maintenance schedule found using the sum of squares objective function for the Eskom case study.*

Table 2.2: *Input data for a 21-unit GMS benchmark system [49] commonly considered in the literature.*

Unit	Capacity (MW)	Earliest starting time (week)	Latest staring time (week)	Duration (weeks)	Manpower required during each week of maintenance
1	555	1	20	7	10,10,5,5,5,5,3
2	555	27	48	5	10,10,10,5,5
3	180	1	25	2	15,15
4	180	1	26	1	20
5	640	27	48	5	10,10,10,10,10
6	640	1	24	3	15,15,15
7	640	1	24	3	15,15,15
8	555	27	47	6	10,10,10,5,5,5
9	276	1	17	10	3,2,2,2,2,2,2,2,2,3
10	140	1	23	4	10,10,5,5
11	90	1	26	1	20
12	76	27	50	3	10,15,15
13	76	1	25	2	15,15
14	94	1	23	4	10,10,10,10
15	39	1	25	2	15,15
16	188	1	25	2	15,15
17	58	27	52	1	20
18	48	27	51	2	15,15
19	137	27	52	1	15
20	469	27	52	1	15
21	52	1	24	3	10,10,10

Generating unit maintenance durations

In GMS problems, the required maintenance duration of each generating unit is usually predetermined [91]. The expected PM duration of units depends on a number of factors. Ideally the duration should be as short as possible, but there is a limit to how much the duration may be shortened. The duration may usually be shortened, but at an additional cost. A three-week PM outage may, for example, be reduced to a two-week period by scheduling overtime maintenance. Overtime is an expensive contributor to maintenance costs [59].

Minimum maintenance durations may vary as a function of individual tasks, due to the different characteristics of maintenance tasks [88]. The expected duration of PM typically depends on the type of power plant. Canto [32] provides the following guidelines:

- Thermal power plants:
  - coal sources: 4 weeks
  - fuel oil: 3 weeks
  - natural gas sources: 3 weeks
- Nuclear power plants: 6 weeks
- Hydroelectric power plants: variable duration according to each power plant.

Table 2.3: *The weekly peak load demands a 22-unit GMS benchmark system [67].*

Week	Demand (MW)	Week	Demand (MW)	Week	Demand (MW)	Week	Demand (MW)
1	1 694	14	1 396	27	1 737	40	1 982
2	1 714	15	1 443	28	1 927	41	1 672
3	1 844	16	1 273	29	2 137	42	1 782
4	1 694	17	1 263	30	1 927	43	1 772
5	1 684	18	1 655	31	1 907	44	1 556
6	1 763	19	1 695	32	1 888	45	1 706
7	1 663	20	1 675	33	1 818	46	1 806
8	1 583	21	1 805	34	1 848	47	1 826
9	1 543	22	1 705	35	2 118	48	1 906
10	1 586	23	1 766	36	1 879	49	1 999
11	1 690	24	1 946	37	2 089	50	2 109
12	1 496	25	2 116	38	1 989	51	2 209
13	1 456	26	1 916	39	1 999	52	1 779

In his GMS model Canto [32], however, assumes the same maintenance duration, namely one month, for all the units (even in different power stations). Some authors state that maintenance of thermal units takes relatively longer [145]. Naturally the size and quantity of equipment in the units will also affect the expected maintenance duration.

Deterministic versus stochastic scheduling paradigms

An important consideration is whether to adopt a stochastic or deterministic modelling approach. Stochastic GMS models usually incorporate the expected electricity demand and forced outages (breakdowns) of power generating units in a probabilistic manner. This is typically done to predict well-known stochastic “risk” measurements, such as the *loss of load probability* (LOLP) and *expected unserved energy* (EUE).

Deterministic models, on the other hand, are generally easier and less time consuming to solve, but they naturally lead to less realistic models than those in a stochastic paradigm.

Regulated versus deregulated power systems

In deregulated markets, the nature of MS is different and slightly more complicated. In these market environments the GENCOs and TRANSCOs are usually responsible for maintaining their equipment, whilst the ISO ensures satisfactory levels of system reliability and security. This is illustrated in Figure 2.5, but the coordination procedure may vary depending on the system [90]. These three actors generally have conflicting goals. For example, the ISO will typically attempt to maximise the total system reliability throughout the entire planning period whilst the GENCOs will try to maximise profit [95] and will most likely schedule maintenance when the price of electricity is low [127], which may make it difficult for demand to be met [90]. These conflicting goals may be resolved through iterative modifications to proposed maintenance schedules or proposing a set of trade-off solutions.

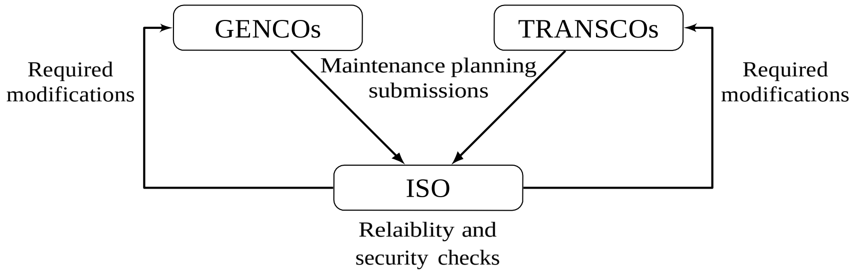


Figure 2.5: *Coordination procedure for MS in a deregulated power system [90].*

2.5 GMS problem formulations

The GMS problem is typically formulated as a scheduling problem with binary decision variables representing whether or not maintenance of each unit should occur during each of a set of discrete time periods. A maintenance schedule is usually modelled as follows [32]. Suppose there are n generating units in the power system in question and m decision time periods over the planning horizon. Let $I = \{1, \ldots, n\}$ denote the set of generating units and let $J = \{1, \ldots, m\}$ be the set of time periods. Define the binary decision variable X_{ij} to take the value 1 if maintenance of generating unit $i \in I$ commences during time period $j \in J$, or zero otherwise. Some researchers prefer to define the decision variables as integer values (for example, where X_i denotes the starting time for maintenance to occur on unit i, in which the case the range for X_i will be the set J). Finally, define the binary auxiliary variable[1] Y_{ij} to take the value 1 if generating unit $i \in I$ is in maintenance during time period $j \in J$, or zero otherwise. Then a maintenance schedule (such as that shown in Figure 2.4) is an assignment of zeros and ones to the $n \times m$ matrix $\boldsymbol{Y} = [Y_{ij}]$ of (auxiliary) decision variables satisfying a variety of constraints.

2.5.1 GMS problem constraints

The constraints included in formulations of the GMS problem may vary significantly, depending on the nature and underlying assumptions of the power utility's operations. The major constraints employed in the literature include the following [3, 90, 107, 124]:

Maintenance window constraints ensure that each unit is serviced between an earliest and latest time period. These time windows are typically dictated by annual generating unit service frequencies, as imposed either by power utility policy or by operational service levels.

Reliability constraints may be incorporated by specifying a reserve/safety margin over and above the expected peak demand for deterministic models. Stochastic models may have minimum levels of stochastic reliability measures such as LOLP and/or EUE values [216].

Load constraints ensure that the expected load demand must be exactly met during each time period. This demand must, of course, be met by the planned production of generating units that are not scheduled for maintenance during the relevant time period. These constraints are also usually included in the UC and/or ED subproblems.

Service contiguity constraints are imposed to ensure that the number of time periods required to service a particular generating unit run consecutively over time.

[1] Some authors actually define this as the primary decision variables, instead of X_{ij}.

Resource constraints specify a limit on the amount of resources available for the purpose of maintenance. These resources may involve service budgets and the availability of spare parts.

Crew/manpower constraints are, in fact, also a type of resource constraint, which focus on the availability of manpower for maintenance work.

Exclusion constraints are employed when certain generating units are not allowed to be taken out of service simultaneously (*e.g.* two units in the same power station or too many units in the same geographical region).

Transmission/network constraints have been incorporated recently and seek to ensure the transmission capabilities of the electrical network (*e.g.* maintaining voltage levels) or that a power station meets the demands of the geographic regions within its service area via the transmission network infrastructure.

These major constraints and some other constraints found in the literature (such as precedence constraints, maintenance period constraints and some other defined GMS constraints) are described below in mathematical form.

Maintenance is typically allowed only once[2] during the entire scheduling window [32, 68, 89], and a maintenance window constraint set ensures that maintenance of a generating unit occurs during pre-specified time windows. The maintenance window constraint is formulated as follows. Let e_i and f_i denote the earliest and latest sting time periods, respectively, during which maintenance of generating unit $i \in I$ may start. Then the maintenance window constraint set may be formulated [169] as

$$\sum_{j=e_i}^{R_i} X_{ij} = 1, \quad i \in I. \tag{2.1}$$

Another way of formulating the above maintenance window constraint set [46] is to require that

$$\sum_{j \in J_i} X_{ij} = 1, \quad i \in I, \tag{2.2}$$

where $J_i = \{j \in J \mid e_i \leq j \leq f_i\}$ is the set of time periods during which the maintenance of generating unit $i \in I$ may start. If integer decision variables X_i are used instead of the binary decision variables X_{ij}, then the maintenance window constraint set may be formulated [125, 202] as

$$e_i \leq X_i \leq f_i, \quad i \in I. \tag{2.3}$$

The duration of maintenance typically varies per unit (as a result of size, type, *etc.*) and so the duration constraint set

$$\sum_{j \in J} Y_{ij} = d_i, \quad i \in I \tag{2.4}$$

is usually enforced when using binary decision variables [32, 147], which also requires the constraint set

$$\begin{aligned} Y_{ij} - Y_{ij-1} &\leq X_{ij}, \quad i \in I, j \in J \\ Y_{i1} &\leq X_{i1}, \quad i \in I, \end{aligned} \tag{2.5}$$

ensuring that maintenance of a generating unit occurs over consecutive time periods. If integer decision variables X_i are used instead of binary decision variables X_{ij}, representing when

[2] If units are to undergo more than one maintenance outage during the time period J, dummy units may be added to the problem — one for each unit outage during the entire scheduling period [168, p. 149].

maintenance commences, the above contiguity (non-stop) constraint set may be formulated [68] as

$$Y_{ij} = \begin{cases} 1, & \text{for } X_i \leq j \leq X_i + d_i - 1 \\ 0, & \text{for all other } j. \end{cases} \tag{2.6}$$

In [125], the above constraint set is written more compactly, albeit nonlinearly, as

$$\prod_{j=X_i}^{X_i+d_i-1} Y_{ij} = 1, \quad i \in I. \tag{2.7}$$

One should, if possible, avoid including any nonlinear constraints or objectives in a model formulation, since it is much more difficult to solve a nonlinear optimisation problem than solving a linear optimisation problem. A more elegant way of representing the duration and contiguity constraints as a single linear constraint involves requiring that

$$\sum_{j=X_i}^{X_i+d_i-1} Y_{ij} = d_i, \quad i \in I \tag{2.8}$$

[124, 148, 169].

Reliability constraints for deterministic GMS models, usually called *reserve margin* or *demand* constraints (or less accurately *load constraints*), are usually enforced to ensure that the available capacity (the system's total installed capacity less the installed capacity lost due to maintenance) is able to satisfy the electricity demand (usually plus a certain safety margin). For stochastic models, where usually only the demand and forced outage probabilities are treated as stochastic, reliability constraints are similarly enforced by specifying minimum LOLP, EUE, and other stochastic risk measures. These stochastic measures are explained in more detail in the next subsection. The reliability constraints described here are for deterministic GMS models. Let I_{ij} denote the installed power generating capacity of unit $i \in I$ during period $j \in J$ and let D_j denote the peak load demand during time period $j \in J$ (*e.g.* the peak demand as illustrated in Table 2.3). Also define

$$r_j = \sum_{i \in I} I_{ij}(1 - Y_{ij}) - D_j, \quad j \in J,$$

as the net reserve margin during time period j. Then the simple and widely used constraint set

$$r_j \geq r_{\min}, \quad j \in J, \tag{2.9}$$

ensures that the available power generating capacity is at least as large as a specified minimum reserve level $r_{\min}$ [42, 48, 46, 89, 161, 167]. Here $r_{\min}$ can be set to zero, meaning that the available power generating capacity should be at least as large as the expected peak demand [89, 48]. Some authors [107, 169] assume that $r_{\min}$ depends on the demand, so that the reserve margin is above the demand D_j together with a specified safety margin S so that constraint set (2.9) is slightly modified to

$$r_j \geq D_j S, \quad j \in J. \tag{2.10}$$

When the output levels of the generating units are introduced as additional variables (part of the UC and ED problems) in conjunction with the GMS problem, then load demand constraints may be enforced. Let p_{ij} represent the generation output of power generating unit $i \in I$ during time period $j \in J$, and let $p_i^{\min}$ denote the minimum production level of power generating unit

$i \in I$. Usually a unit's installed capacity I_{ij} is constant over time and may be denoted by I_i. Then the load demand constraints

$$\sum_{i \in I} p_{ij} = D_j, \quad j \in J \tag{2.11}$$

and the power limit constraints

$$p_i^{\min}(1 - Y_{ij}) \leq p_{ij} \leq I_i(1 - Y_{ij}), \ i \in I, \quad j \in J, \tag{2.12}$$

ensure load/demand satisfaction and adherence to minimum and maximum energy production levels [27, 29, 66, 82, 167]. As may be seen in (2.12), the optimal production plan is influenced by the PM auxiliary variables Y_{ij}, in that the generation output variable p_{ij} must be zero if maintenance of generating unit i occurs during that time period (*i.e.* if $Y_{ij} = 1$).

Period constraints are defined in [32, 125] to limit the maximum number of maintenance outages ψ_j per period $j \in J$ as

$$\sum_{i \in I} Y_{ij} \leq \psi_j, \quad j \in J. \tag{2.13}$$

Resource constraints are generally employed to ensure that the total maintenance scheduled during each period does not exceed the total amount of resources (*e.g.* personnel, parts, special tools, *etc.* [124]) available per time period $j \in J$, defined as σ_j. If the amount of resources required to maintain unit i, denoted by R_i, is constant throughout its maintenance duration, then the simple constraint set

$$\sum_{i \in I} R_i Y_{ij} \leq \sigma_j, \quad j \in J \tag{2.14}$$

is usually adopted. In [125, 148], a simple resource constraint set is specified by requiring that

$$\sum_{i \in I} Y_{ij} I_i \leq B_j, \quad j \in J, \tag{2.15}$$

where B_j represents a set maximum amount of power allowed to be maintained in the power system during time period $j \in J$. This enforces that only a certain number of generating units, based on their installed capacity I_i, are allowed to be maintained at a time.

In [6], a more complicated formulation of the resource constraints, which seems more practical, is employed. Let $S = \{1, \ldots, p\}$ be the set of power stations, let $E = \{1, \ldots, u\}$ be the set of equipment types (*e.g.* boilers, turbines, *etc.*), and let $R = \{1, \ldots, f\}$ be the set of resource types (*e.g.* parts). Define the binary decision variable w_{siej} to take the value 1 if equipment of type $e \in E$ in power generating unit $i \in I$ of power station $s \in S$ during time period $j \in J$ is not in preventative maintenance, or 0 otherwise. Then the constraint set

$$\sum_{i \in I} (1 - w_{siej}) R_{sejr} \leq \sigma_{sejr}, \quad s \in S, \quad e \in E, \quad j \in J, \quad r \in R \tag{2.16}$$

will ensure that no more than the available amount of resources for maintenance is committed, where R_{sejr} is the amount of resources of type $r \in R$ required by equipment of type $e \in E$ in power station $s \in S$ during time period $j \in J$ and σ_{sejr} is the total amount of resources of type $r \in R$ available for equipment of type $e \in E$ in power station $s \in S$ during period $j \in J$.

In [216], a similar approach is taken to incorporate r different types of resource constraints for the different types of resources, but the model only accommodates resources required per unit

$i \in I$, and no further equipment within the unit as in (2.16). Similarly, in [6] the maintenance crew constraint set

$$\sum_{i \in I} (1 - w_{siej}) m_{sej} \le M_{sej}, \quad s \in S, \quad e \in E, \quad j \in J \tag{2.17}$$

ensures that no more than the available number of crew members required for maintenance is committed, where m_{sej} is the amount of manpower required for equipment of type $e \in E$ in power station $s \in S$ during time period $j \in J$ and M_{sej} is the total number of maintenance crew members available for equipment of type $e \in E$ in power station $s \in S$ during period $j \in J$.

For the above two (resource and maintenance crew) constraints to hold, the amount of resources of type $r \in R$ or the amount of manpower required must not change over the maintenance time period (as is the case in the 21-unit system in Table 2.2). In [6], however, the authors ignored these constraints by assuming that there is no shortage of the resources and maintenance crew required.

If, however, the required resources or manpower needed for maintenance of unit $i \in I$ changes during its duration, *e.g.* the manpower required becomes less over time (as is the case in the 21-unit system in Table 2.2), the resource constraint formulation becomes more complicated to implement. This is because the resources required R_{ij} has to be adapted depending on when maintenance starts, *i.e.* are dependent on the decision variables. One way of formulating this [46, 89] is to employ additional dependent time period sets, which change depending on when unit maintenance starts. This is, however, only possible if the formulation allows one to know when maintenance starts (which is not the case when using the binary decision variables), *i.e.* for the integer decision variables formulations this is just the value X_i, or if employing a metaheuristic model solution technique this may be overcome by encoding strategies [46, 89].

In [169], an elegant approach is followed to overcome this problem. Let m'_{cij} denote the required maintenance crew for unit $i \in I$ when in maintenance during time period $j \in J$ if maintenance were to commence during time period $c \in J$. Then

$$m'_{cij} = \begin{cases} m_i^{j-c+1}, & \text{if } j - c < d_i \\ 0, & \text{otherwise,} \end{cases} \tag{2.18}$$

where m_i^u denotes the maintenance crew required for unit $i \in I$ in its u-th period of maintenance, and the maintenance crew constraint set may be formulated as

$$\sum_{i \in I} \sum_{c=1}^{j} m'_{cij} X_{ic} \le M_j, \quad j \in J, \tag{2.19}$$

where M_j denotes the total manpower (or crew) available during time period $j \in J$. Usually M_j is constant throughout the time period and so its subscript j is omitted in most formulations.

If, however, the maintenance crew requirements m_i of generating unit $i \in I$ remains the same throughout its maintenance time period, then the manpower constraint may be expressed more simply as

$$\sum_{i \in I} m_i Y_{ij} \le M_j, \quad j \in J. \tag{2.20}$$

Exclusion constraints are sometimes incorporated into GMS models in order to prevent certain units from being in simultaneous maintenance (*e.g.* units within the same power station or class, or within the same geographical region [147, 168]). Sometimes the more specific name of a geographical constraint is applied [107] which helps to avoid larger transmission losses and lower reserve capacity in one region and to limit the number of generator units under maintenance in each region. Consider the more general exclusion constraint where at most some specified number of units, within some subset of units, are allowed to be in simultaneous maintenance. Let $\mathcal{K}$ denote the set of indices of generating unit exclusion subsets. If there are K such subsets, then $\mathcal{K} = \{1, \ldots, K\}$. Define $\mathcal{I}_k \subseteq \mathcal{I}$ as the k-th subset of generating units that form an exclusion set, with $k \in \mathcal{K}$. The exclusion constraint set may then be formulated as

$$\sum_{i \in \mathcal{I}_k} Y_{ij} \leq K_k, \quad j \in \mathcal{J}, k \in \mathcal{K}, \tag{2.21}$$

where K_k denotes the maximum number of units within subset $\mathcal{I}_k$ that are allowed to be in simultaneous maintenance during any time period [169]. In [32, 150], K_k is taken as 1, so that no two units can be in maintenance during the same period.

Precedence constraints may be incorporated into GMS models [32, 42], indicating that some generating units should be in maintenance before others (*e.g.* based on different priority levels). If maintenance of unit i_1 has to start before that of unit i_2, the pair of constraint sets

$$\sum_{p=1}^{j} X_{i_1 p} - X_{i_2 j} \geq 0, \quad j \in \mathcal{J}, \tag{2.22}$$

$$X_{i_1 j} + X_{i_2 j} \leq 1, \quad j \in \mathcal{J} \tag{2.23}$$

ensures this precedence when using binary decision variables [32]. Constraint set (2.22) ensures that unit i_2's maintenance does not start before that of unit i_1, and constraint set (2.23) prevents the simultaneous maintenance of the two units (i_1 and i_2) from occurring. When using the integer variables, the constraint

$$X_{i_1} + d_{i_1} \leq X_{i_2}, \tag{2.24}$$

ensures that maintenance of unit i_2 will only start after the maintenance of unit i_1 has been completed [125, 150].

More recently, transmission/network constraints have been included in GMS models, especially when the TMS problem is solved in conjunction with the GMS problem [1, 68, 135, 137]. This is done to limit the amount of energy sent over the power system's transmission network. For these constraints to be formulated, the amount of energy that has to be sent out from each generating unit needs to be determined. This is usually accomplished by solving the ED subproblem [128] with the addition of constraints sets (2.11) and (2.12). From these generating unit output levels p_{ij}, a load flow problem is solved to obtain the line flows (the energy sent out along the transmission lines) [128]. Let $\mathcal{L} = \{1, \ldots, L\}$ be set of indices of transmission lines (usually from one bus to another [128]). These computed line flows must adhere to the transmission constraint set

$$|f_R| \leq F_R, \quad f \in \mathcal{L}, \tag{2.25}$$

where f_R denotes the flow along transmission line f and F_R is the set maximum flow limit of transmission line f in the power network [68, 128].

Finally, the constraint sets

$$X_{ij} \in \{0,1\}, \quad i \in \mathcal{I}, j \in \mathcal{J}, \tag{2.26}$$

$$Y_{ij} \in \{0,1\}, \quad i \in \mathcal{I}, j \in \mathcal{J} \tag{2.27}$$

specify the logical nature of the decision (X_{ij}) and auxiliary (Y_{ij}) variables.

As noted in [17, 124] a rough distinction may be made between constraints that must not be violated and constraints which should be more or less satisfied, referred to as *hard* and *soft* constraints, respectively. The decision maker may specify which constraints are to be considered hard and which are to be considered soft.

Kralj and Petrović [124] classified the constraints for maintenance to occur within the planning period and achieving an adequate level of reliability (minimum deterministic or stochastic reliability measures) as hard constraints. Resource and crew constraints, on the other hand, were classified as soft constraints, since additional resources (including manpower) may be planned if required and/or overtime work may be scheduled so that more work may be accomplished [124]. This will all, however, come at an additional cost, but with more flexibility[3]. In [66], it is noted that a maintenance schedule with a high reliability but requiring additional crew is often acceptable in a power plant. The flexibility for the crew constraint in [66] was allowed to be violated by as much as 5%.

Satoh and Nara [167], who employed a metaheuristic model solution technique, further classified GMS constraints as "easy" or "difficult" to satisfy and encode, which is especially applicable when not using mathematical programming techniques. Satoh and Nara classified the window, duration, precedence, and generator output limit constraints as easy constraints, since most of these may be dealt with by a suitable encoding of solutions, whereas exclusion, load, and reliability constraints were classified as difficult constraints since it is harder to find feasible solutions satisfying these constraints and it is also harder to enforce these in a solution encoding scheme. Finding feasible solutions for these difficult constraints is often certainly not easy, and a penalty is therefore typically associated with violating these difficult constraints. These penalties may be included in an objective function to be minimised by the incorporation of subjective penalty parameters [167, 169]. Kralj and Petrović [124] noted that some constraints can be built implicitly into the model (through the adoption of judicious encoding schemes) such as requirements as to when a unit must be maintained, that maintenance must occur over consecutive time periods (*i.e.* with no interruptions) and that the duration of unit maintenance is fixed.

Baskar *et al.* [17] classified the maintenance window and contiguity constraints as soft constraints, since the maintenance starting period of any unit can easily be selected in the "preferred maintenance" interval. Furthermore, in [17], crew, demand, and a stochastic reliability constraint (involving the EUE) were treated as hard constraints, noting that it is very difficult to generate solutions satisfying these hard constraints.

To conclude, the GMS problem is often a complex problem, and it is usually difficult to satisfy all its constraints [128]. Sometimes constraint violations are allowed for specified soft constraints and minimisation of these soft constraint violations is incorporated as a separate GMS objective (called a convenience criterion) [128, 63].

2.5.2 GMS problem objectives

A wide variety of GMS objective functions may be found in the literature. These objective functions are usually based on three dominant scheduling criteria, namely economic criteria, reliability criteria and convenience criteria [124, 169, 222], with the economic and/or reliability criteria most often occurring in the GMS literature.

Economic criteria. The most common economic GMS objectives consist of minimising the total operating cost associated with a generator maintenance schedule, which includes energy production and maintenance cost [46]. Energy production cost includes fuel cost,

[3]Huang *et al.* [107] actually used fuzzy set theory to deal with this uncertainty problem.

salaries and wages, costs related to energy production and generator start-up and shutdown costs. Maintenance costs, on the other hand, include replacement part costs and salaries and wages related to unit maintenance [168]. These economic costs typically vary from generating unit to generating unit and data related to these costs are sometimes difficult to obtain [168].

Reliability criteria. Depending on the model, reliability criteria may be either stochastic or deterministic in nature [145]. The most common deterministic objectives in this class involve measures on the reserve margin, either levelising the reserve margin or ensuring a satisfactory level of available power. When levelising the reserve margin over the time period, the most common formulation is to minimise the *sum of squared reserves* (SSR) [90, 145]. Another option is to maximise the smallest reserve load during any time period. Stochastic reliability objectives generally involve common stochastic risk measures, such as the total period's LOLP and EUE [46]. Like deterministic criteria, another stochastic criterion involves levelising risk indices (such as LOLP) for each time period (this is commonly achieved by levelising the effective/equivalent reserve margins, which usually incorporates a stochastic installed capacity per unit, *forced outage rate* (FOR), and expected load) — typically by minimising the sum of squared effective/equivalent reserve margins [47]. Generally, it can be shown that optimal solutions obtained under either a deterministic or a stochastic reliability criterion are also acceptable (although not necessarily optimal) in terms of the other reliability criterion [147].

Convenience criteria. Examples of convenience criteria include minimising soft constraint violations, minimising deviations from a desired maintenance schedule, or minimising possible disruptions to the power generation schedule.

Depending on the model and objective function employed, these three categories of scheduling criteria are often conflicting, ultimately making the GMS problem MO in nature [124]. For example, Kralj and Petrović [124] noted that a compromise between several requirements must typically be achieved. Examples include ensuring adequate reliability with minimal fuel costs, ensuring minimal deviation from technologically optimal maintenance schedules, and maximising the efficiency of the available resources and manpower while respecting all other constraints. Mukerji *et al.* [149] noted that the two most important optimisation criteria are maximising reliability in some sense and minimising production cost. Schedules with high reliability scores tend to have low production costs, and *vice versa*, but the schedule that gives the highest reliability may not have the lowest production cost [149, 216]. This phenomenon may be attributed to the fact that a utility having low reserves will have to bring online its more expensive expensive units more frequently [216]. Canto [32] noted that the electricity energy demand must be supplied under an adequate reliability level whilst the associated cost of electric generator shutdown has to be the smallest possible.

SO and some MO formulations and solution approaches have been proposed for GMS problems [169], with most employing SO approaches, or including all but the one dominant criterion as constraints in model formulations [90, 124]. In SO approaches, deciding on which objective to choose for this purpose depends on the power system's goals: for some power utilities, reliability objectives are more important than economic considerations (such as in regulated electricity markets, for example [168]) whilst other utilities prefer economic considerations (those in deregulated electricity markets may only be concerned with maximising net revenue [89]).

The different types of objectives formulated within the above three criterion categories are described in more detail in the remainder of this section, and this is followed by a discussion on how SO and MO GMS problem instances are typically solved.

Economic criteria

A number of different types of economic objectives are adopted in GMS models in the literature. In regulated systems, the goal is usually to minimise total cost[4], typically focussing on production and maintenance costs [32, 42, 46, 90]. The deregulation of the electric power market in many countries has, however, shifted the focus away from operating cost and reliability more towards profitability [90, 127, 168, 201].

The total operating cost usually includes the cost of energy production and maintenance [46, 201, 222]. Energy production costs include fuel costs [201, p. 248], salaries and wages of production personnel, costs related to energy production and generator start-up and shut-down costs [168]. Maintenance costs are usually partitioned into fixed and dependent maintenance costs [201, p. 248]. Fixed maintenance cost is normally constant and is not related to whether the generating unit is often in a state of operation. Dependent maintenance costs, on the other hand, are related to how the generating unit is operated, arising from wear and damage due to switching it on and off frequently, or prolonged operation. Dependent maintenance costs may also derive from implemented maintenance and operation standards, and the cost of maintenance may change when the maintenance time is allowed to float, resulting in additional overtime payments or payments of wages to extra staff [201, p. 248]. These economic costs typically vary from generating unit to generating unit, and data related to these costs are sometimes difficult to obtain [168], especially data related to maintenance costs [201, p. 249].

Let c^p_{ij} denote the production cost associated with power generating unit $i \in I$ during time period $j \in J$ and let c^m_{ij} denote the maintenance cost incurred if power generating unit $i \in I$ is in maintenance during time period $j \in J$. Also, let p_{ij} represent the generation output of power generating unit $i \in I$ during time period $j \in J$. Then the objective is often to minimise the total operating cost (production and maintenance costs), *i.e.* to

$$\text{minimise} \quad \sum_{i\in I}\sum_{j\in J}\left(c^p_{ij}(p_{ij}) + c^m_{ij} Y_{ij}\right), \tag{2.28}$$

where $c^p_{ij}(p_{ij})$ is the production cost associated with generator output p_{ij} [27, 69, 122, 128]. In this case, the load demand constraints

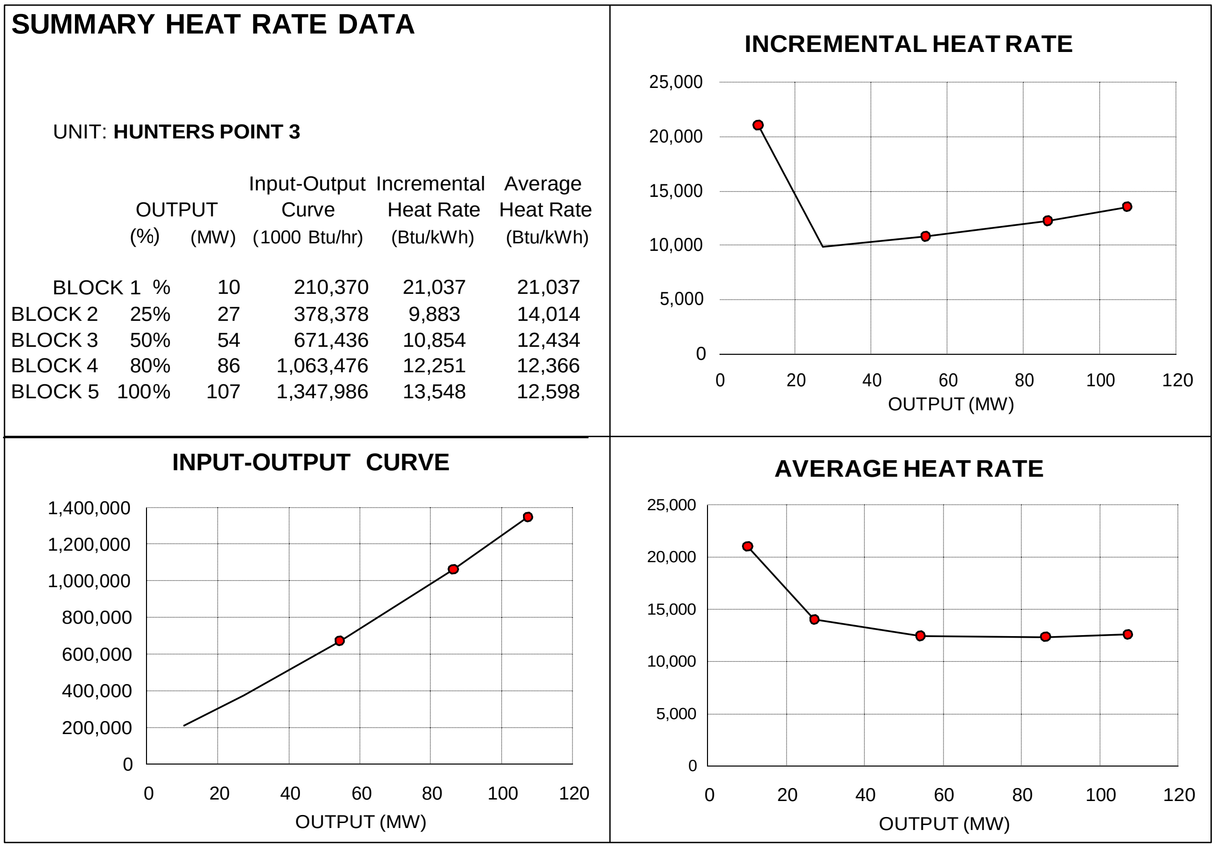
$$p_i^{\min}(1 - Y_{ij}) \leq p_{ij} \leq I_i(1 - Y_{ij}), \quad i \in I, \quad j \in J \tag{2.29}$$

and

$$\sum_{i\in I} p_{ij} = D_j, \quad j \in J \tag{2.30}$$

related to minimum and maximum production and demand satisfaction are usually enforced [27, 29, 66, 167]. As may be seen in (2.29), the optimal production plan is influenced by the PM decision variables Y_{ij}, in the sense that the generation output variable p_{ij} must be zero if maintenance of generating unit $i \in J$ occurs during that time period $j \in J$ (*i.e.* if $Y_{ij} = 1$). Usually p_{ij} is determined by solving an instance of the well-known ED problem (described in §2.3.2). El-Sharkh *et al.* [69] noted that there are, in fact, two subproblems to be solved, namely a subproblem involving determination of the MS variables Y_{ij}, called the *maintenance scheduling subproblem*, and one involving determination of the production output variables p_{ij}, called the *power system subproblem*. Satoh and Nara [167] used SA to find optimal values for the maintenance schedule variables and then, based on these Y_{ij}-values, determined the production output variables, whose production (fuel) cost was formulated as a linear function, using the

[4]Usually in regulated systems, a government regulates the system directly or indirectly. In this case the utility should not take advantage of the end consumer [90].



SUMMARY HEAT RATE DATA

UNIT: **HUNTERS POINT 3**

	OUTPUT (%)	(MW)	Input-Output Curve (1000 Btu/hr)	Incremental Heat Rate (Btu/kWh)	Average Heat Rate (Btu/kWh)
BLOCK 1	%	10	210,370	21,037	21,037
BLOCK 2	25%	27	378,378	9,883	14,014
BLOCK 3	50%	54	671,436	10,854	12,434
BLOCK 4	80%	86	1,063,476	12,251	12,366
BLOCK 5	100%	107	1,347,986	13,548	12,598

Figure 2.6: *I/O curves for one of the Pacific Gas and Electric company's power generating units (called "Hunters point 3") [121].*

equal incremental method to solve the ED problem. Methods employed to solve the ED problem depend on how the production cost $c^p_{ij}(p_{ij})$ is modelled. The formulation involves modelling each power generating unit's fuel consumption (cost) associated with producing a certain amount of power.

The fuel cost is the most significant cost associated with power generation and is sometimes merely called the production cost [201, p. 248]. To understand how the fuel cost is estimated, an understanding must first be gained of how a power generating unit's heat rate (the ratio of thermal energy in to thermal energy out) is determined. The heat rate is the inverse of a power generating unit's efficiency (*i.e.* how much energy is converted from the fuel source to output electric power). In order to determine heat rates, engineers first measure the fuel (input) required to maintain various levels of generation (output) [121, 212]. These (input, output) coordinates together constitute the so-called I/O curve of the power generating unit in question. From these values incremental and average heat rate values may be determined, as illustrated in Figure 2.6. The generation output is usually measured in MW, whilst the fuel input may be measured in units of weight per hour [*e.g.* tonnes/h], energy per hour [MBtu[5]/h], monetary expenditure per hour, calculated according to the cost of fuel in terms of monetary unit per weight ($/Tonnes) or per energy ($/MBtu) [$/h].

Depending on the data, the I/O curve (or the fuel/production cost rate) may be modelled as

[5] The *British thermal unit* (Btu) is the traditional unit of work in the power industry. One Btu is approximately 1055 joules.

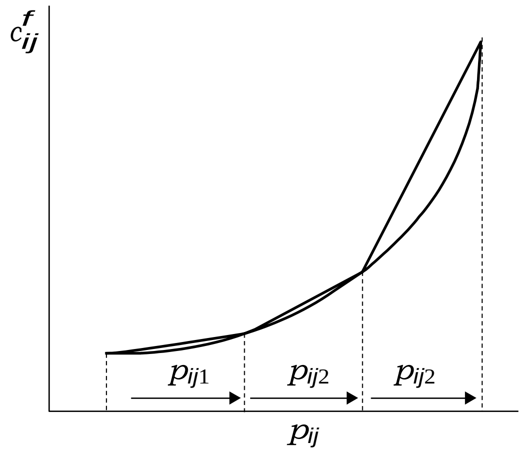


Figure 2.7: *A nonlinear cost function approximated by a piecewise-linear function involving three linear segments [197, p. 70, 213].*

approximately linear [17, 32, 167, 126, 128, 150, 216], piece-wise linear [33, 57], quadratic [67, 66, 154] or cubic with varying degrees of accuracy [212].

Many authors formulate the fuel cost as the quadratic function

$$c^f_{ij}(p_{ij}) = a_i p_{ij}^2 + b_i p_{ij} + c_i \quad [\$/h], \tag{2.31}$$

where a_i, b_i, and c_i are cost coefficients pertaining to power generating unit $i \in I$. Some authors in literature and many power utilities, however, prefer to represent the fuel cost as linear or piecewise-linear cost functions [213, p. 68]. The nonlinear fuel cost function in Figure 2.7 is, for example, approximated by a piece-wise linear function containing three linear function segments. This local linearisation creates the problem of possibly achieving less accurate cost estimates (than when adopting a nonlinear cost function). Naturally, the degree of error of estimating the cost curve decreases with the number of segments used. This is illustrated by the example quoted from [213, p. 72] in Table 2.4. The example relates to three power generating units whose production output must be optimised. The standard solution, using the lambda search method, was employed in [213, p. 68] for the three quadratic representations of the units' cost curves. Optimum scheduling results are compared to those obtained by approximating the three functions by piecewise-linear functions.

It may be seen in Table 2.4 that the total cost does not differ much — using even one linear function segment results in a cost error of under 0.5%. There is, however, a more substantial difference in the corresponding generator output values, especially for Generator 3. This type of error is worth noting if the fuel or production cost is represented as a linear function.

In the GMS problem in [165], the total fuel (called generation) cost was minimised using a merit (cheapest to most expensive) order dispatch logic.

A substantial number of authors [201, 215, 216, 221] have claimed that operational costs (specifically those based on fuel cost) are not very sensitive to variation of maintenance plans, when the same amount of maintenance is performed in each case [167], with values in the literature ranging from 0.08% to 0.3% [149, 216, 221]. In [216], the insensitivity of production cost is attributed to most probably being the result of many simplifications made because of computation difficulties. As noted in [221], however, the UC problem influences the production cost more significantly, with Johnson *et al.* [115] reporting a fuel cost saving of around 1%, and thus it is important to follow an accurate UC logic within the larger GMS problem instance. In [218], production costs (consisting of fuel cost with more elaborate formulations of start-up

Table 2.4: *Scheduling results emanating from piecewise-linear approximations of quadratic cost functions [213, Example 3C].*

# of linear function segments	Generator 1 (MW)	Generator 2 (MW)	Generator 3 (MW)	Total Cost ($/h)	Cost Error
1	400	400	50	8 227.870	0.40899 %
2	375	350	125	8 195.369	0.01236 %
3	450	300	100	8 204.105	0.11897 %
5	400	340	110	8 195.206	0.01037 %
10	385	340	125	8 194.554	0.00242 %
50	393	335	122	8 194.357	0.00001 %
Standard solution with lambda search	393.2	334.6	122.2	8 194.356	

and maintenance costs) varied as much as 6%. Importantly, GMS models involving production cost objectives also typically take much longer to solve than GMS models involving most other criteria (such as reliability criteria) [215, 216, 222] — in [216], for example, a run time increase of 64% is reported. In [149], a run time increase of 63% (166% when the model's constraints are relaxed) is reported. In [222], a run time of 7 minutes is reported when minimising the expected energy production cost, whilst a run time of 0.14–0.27 minutes is reported for various reliability criteria measures — this means that it took 50–29 times longer to compute solutions in the former case.

The maintenance cost associated with a maintenance plan represents the cost of suboptimality or lost opportunity as a result of adopting a maintenance schedule. It typically includes the cost of maintaining a power generating unit too early or too late [59, 149]. Each power generating unit usually has its own, ideal maintenance window. Maintenance of a unit that takes place too soon wastes money, because needless maintenance work is done. Waiting too long, on the other hand, can also be expensive, because the power generating unit's availability deteriorates owing to an increased probability of forced outages, often causing repairs to become more expensive. Besides this increased probability of failure risk, higher costs will be incurred where overhauls are carried out too late as these will necessitate increased maintenance (parts would have been in service longer) [148]. These costs may be quantified and incorporated collectively as a model objective or as window constraints for each power generating unit separately [149]. The maintenance cost is usually expressed as a function of time elapsed between two successive maintenance outages [124].

El-Sharkh *et al.* [69] adopted a fuzzy logic approach toward modelling maintenance cost, since this cost varies according to the changes in market prices, availability of spare parts, weather conditions and the availability of maintenance crew. In [83, 186], the scheduling objective consisted only of maintenance costs, that is, the second part of (2.28), where the maintenance cost c_{ij}^m increases linearly with time (*i.e.* as a function of j), so that maintenance of power generating units is scheduled as early as possible, subject to reserve capacity, crew availability, maintenance window and maintenance duration constraints. Similarly, in [148], the scheduling objective is to

$$\text{minimise} \sum_{i\in I}\sum_{j\in J} c_{ij}^m Y_{ij} I_i. \tag{2.32}$$

The maintenance cost coefficients c_{ij}^m in (2.32) are at a minimum when a unit is in maintenance

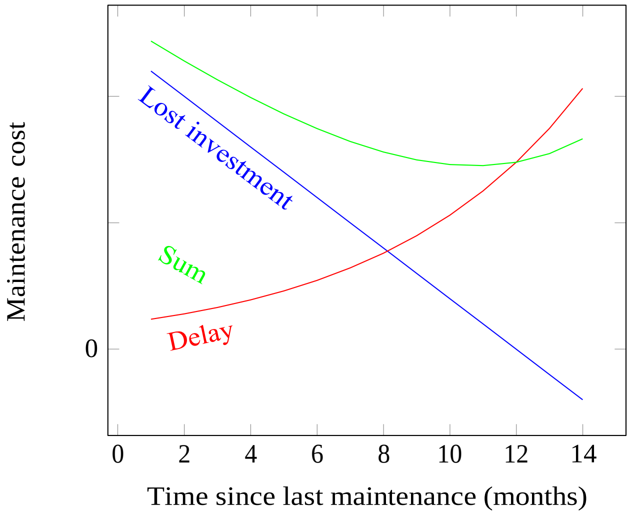


Figure 2.8: *The sum of underpreciated maintenance investment lost by maintaining a unit too early and the expected out-of-pocket (deterioration worsening maintenance work/cost and expected cost due to the likelihood of forced outages) maintenance cost in terms of time since the previous maintenance [59].*

during its preferred period of maintenance; outside the preference window these costs increase linearly. The incremental cost of performing maintenance prior to the preferred period is a result of premature exchange or repair of the power generating unit's components which might still be operating without failure [148]. Also, in (2.32) the maintenance cost is multiplied by the power generating unit's installed capacity I_i so that precedence is given to the larger units' preferred maintenance periods. The effect of including the model objective (2.32) is somewhat similar to incorporating a maintenance window constraint (*i.e.* when maintenance is allowed to occur).

As noted by Dopazo *et al.* [59] and Kuzle *et al.* [127], the maintenance cost objective is usually formulated so as to facilitate finding an optimal balance between not scheduling units too early for maintenance, since part of the investment made during the previous maintenance is lost as the unit's parts were meant to be in operation longer but are replaced prematurely, and not scheduling units too late for maintenance, which may incur additional expenses as a result of partial or full damage to machinery. This is illustrated in Figure 2.8 where the sum of underpreciated maintenance investment lost by maintaining a unit too early and the expected out-of-pocket (larger maintenance due to deterioration and expected cost due to the likelihood of forced outages) maintenance cost in terms of time since previous maintenance are plotted [59]. Curves like the one in Figure 2.8 may be developed for each generating unit of a power system. In the absence of constraints each unit would be maintained at its individual optimum point in time (the time period associated with the lowest value in the total maintenance cost). The presence of constraints will, however, probably affect where in time this optimum is [59]. Dopazo and Merrill [59] suggests minimising the overall system maintenance cost (which is the sum of all the units' maintenance costs) associated with a maintenance schedule.

If maintenance outage durations are allowed to vary (as mentioned they are usually fixed), a trade-off results between the energy production cost and the maintenance cost [46, 215]. Shorter outage durations lead to higher maintenance costs [222], since more crew (through overtime work or additional staff) are employed and parts are needed in a shorter time [215], but lower production costs are incurred since more expensive units do not have to be brought online as much during those times when cheaper units are in maintenance [46, 222]. As further noted in [59], a three-week outage could, for example, be reduced to a two-week period by scheduling overtime (at a known cost), although overtime is an expensive contributor to maintenance costs [59] and will most probably very rarely be permitted. Also, the power system's reliability (in terms of satisfying demand) will be higher as a result of shorter outage durations [222].

As noted in [201, p. 249], sometimes only the production cost is included in economic cost scheduling objectives since it is difficult to quantify maintenance cost accurately. Canto [32] furthermore noted that maintenance costs are typically insignificant compared to start-up and production costs. Maintenance costs were, in fact, found to be of the order of one thousand times smaller than generating unit start-up costs and of the order of one million times smaller than production costs [32].

Start-up costs are nevertheless sometimes included in scheduling objectives. Suppose power generating unit $i \in I$ has a start-up cost c_i^s associated with it [32, 38, 42]. The UC problem typically has to be solved, based on the values of the maintenance schedule variables Y_{ij}, so as to determine the output variable values s_{ijn}, which take the value 1 if power generating unit $i \in I$ starts up during subperiod $n \in N$ of period $j \in J$. The specification of a finer grained subperiod set $N = \{1, \ldots, o\}$ (typically hours/days) within period j (typically days/weeks) is usually aimed at more accurately determining the expected electricity demand for the UC and ED problem (*i.e.* daily or hourly demand). For instance, Canto [32] takes the maintenance period set J as a set of thirteen one-month periods, and within the each period, there are six subperiods, corresponding to two parts, namely business days and weekdays, and within these parts, three more subparts, namely peak, middle, and low demand. In [32], the economic scheduling objective is to

$$\text{minimise} \sum_{i\in I}\sum_{j\in J}\sum_{n\in N}(c_i^p p_{ijn} + c_i^s s_{ijn}). \tag{2.33}$$

In [38, 42, 218], the start-up cost component of the profit maximisation objective is also modelled as in the second part of (2.33).

The start-up cost may be modelled in a more complex manner, as noted in [197, 212], because the start-up cost for a thermal unit is actually a function of the time that the unit has been shut down. That is, it is cheaper to restart a warm power generating unit than a cold one. If such a complex model is employed, the start-up cost formulation in the second part of (2.33) is inadequate. In [197], for example, an exponential function is presented to estimate the start-up cost.

As mentioned, the deregulation of the power industry has opened up the market to competition [90]. In a deregulated system, the economic scheduling objective for the GENCOs[6] typically changes to the maximisation of profit (or revenue less costs) [38, 90, 95, 119]. The costs are similar to the ones described above (namely fuel costs, maintenance costs and start-up costs). The revenue component is, however, incorporated by including an energy (spot) price estimate λ_{jn} (typically measured in units of monetary cost per megawatt-hour) for period $j \in J$ and subperiod $n \in N$ [15, 38, 42, 95]. In general, GENCOs attempt to

$$\text{maximise} \sum_{i\in I}\sum_{j\in J}\sum_{n\in N}(\lambda_{jn} p_{ijn} - (a_i p_{ijn}^2 + b_i p_{ijn} + c_i) - c_i^s s_{ijn} - c_i^m Y_{ij} I_i) \tag{2.34}$$

[15, 38, 218]. The objective function in (2.34) represents the GENCOs' profit (expected revenue less production cost less start-up cost less maintenance cost). GENCOs will want to deliver electricity when the price is high, and will attempt to schedule maintenance of generating units when the price is low [127].

[6]As has been noted for deregulated systems, the ISO is usually concerned with demand satisfaction and congestion avoidance, whilst the GENCOs (and TRANSCOs) seek to maximise their profits [90, 95].

Reliability criteria

Reliability scheduling criteria involve satisfying demand as reliably as possible. That is, the available capacity (the installed capacity less the capacity lost due to maintenance and outages) should be able to meet the expected demand according to a certain level of reliability. These reliability criteria may be either stochastic or deterministic in nature [23, 145, 201]. Deterministic models incorporate reliability more simply as the net reserve margin

$$r_j = \sum_{i \in I} I_i(1 - Y_{ij}) - D_j, \quad j \in J \tag{2.35}$$

during period j. Sometimes forced outages are also included as a safety factor S on top of the demand.

In contrast, stochastic models are able to accommodate reliability (which is usually rather expressed in terms of inverse risk measures) more accurately by taking into account the expected FORs and the variations in expected demand. In deterministic modelling approaches, the main criterion is usually to levelise the net reserve loads[7] [90, 201]. In stochastic approaches, on the other hand, the risk of the expected available capacity not satisfying demand per period is also levelised or the sum of these risks per period is minimised. The objective functions that have been adopted to measure this type of risk include the LOLP, the *loss of load expectation* (LOLE), or the *expected energy not served* (EENS) [201]. Another technique is to levelise the effective/equivalent reserve margins (which incorporates a stochastic installed capacity per unit, FORs, and expected load), usually by minimising the sum of squared effective/equivalent reserve margins [47]. This effective/equivalent reserve margin per time period is calculated from the *effective/equivalent load carrying capacity* for each unit and the *effective/equivalent load* for each interval [47, 145].

In deterministic models, the most common objective function, when attempting to levelise the net reserve load, is to

$$\text{minimise} \sum_{j \in J} r_j^2 \tag{2.36}$$

[48, 46, 66, 89, 145, 161, 169, 170]. The scheduling objective in (2.36) penalises the deviation between the available capacity during period $j \in J$ and the expected peak demand D_j across the planning horizon. The reason for taking the square of r_j instead of r_j itself within the sum in (2.36) is that outliers are penalised more severely [168] in the former case. Since the maintenance outage duration is assumed to be constant, simply summing the reserves ($\sum_{j \in J} r_j$) will, in fact, always produce the same value for different maintenance schedule decision variables Y_{ij} [222].

In [145], the objective function (2.36) is slightly modified to

$$\text{minimise} \sum_{j \in J} (r_j - r_{\min})^2, \tag{2.37}$$

where r_{min} is a set minimum reserve requirement. The objective in (2.37) seeks to steer the reserve load as close as possible to r_{min}. Similarly, some authors include a safety margin S, so that D_j in (2.35) is replaced by $D_j(1 + S)$. Since S is a constant (usually between 8–15%), its value will not affect the decision variables in the objective function (*i.e.* the optimum maintenance schedule decision variables X_{ij}). The variable values might, however, be infeasible in the more constrained instance if they violate the widely enforced constraint that the available

[7] The net reserve is the gross reserve less the capacity lost due to maintenance. Furthermore, the gross reserve is the total installed capacity ($\sum_{i \in I} I_i$) less the expected demand for time period $j \in J$ [67, 90, 215, 201].

capacity must be at least the demand plus a safety margin $D_j(1 + S)$. A safety margin is usually included in deterministic models so as to take into account random factors, such as random unit outages [27].

Another way to levelise net reserves is to minimise the deviation of the net reserves r_j from the average net reserve $\bar{r}$, that is to

$$\text{minimise} \sum_{j \in J} (\bar{r} - r_j), \tag{2.38}$$

where

$$\bar{r} = \frac{1}{m} \sum_{j \in J} r_j \tag{2.39}$$

is the average net reserve [67]. Similarly, the scheduling objective adopted in [17] is to

$$\text{minimise} \sum_{j \in J} \left(\frac{r_j - \bar{r}_j}{\bar{r}_j} \right)^2. \tag{2.40}$$

Some authors [185], however, levelise the net reserve load according to the net reserve rate

$$r^r_j = \frac{\sum_{j \in J} r_j}{D_j + \sum_{i \in I} I_i X_{ij}} \tag{2.41}$$

and the average net reserve rate

$$\bar{r}^r_j = \frac{1}{m} \sum_{j \in J} r^r_j. \tag{2.42}$$

In particular, their scheduling objective is to

$$\text{minimise} \sum_{j \in J} \left(\frac{r^r_j - \bar{r}^r_j}{\bar{r}^r_j} \right)^2. \tag{2.43}$$

The objective function adopted in [202] is similar to (2.43), except that the calculation of the net reserve rate r^r_j (2.41) is slightly different. In [42], the scheduling objective is to maximise the sum by period of the ratio of the net reserves to the gross reserves.

Instead of the widely used levelling objectives described above, the reliability objective of maximising the minimum net reserve, that is

$$\text{maximising} \quad \min_{j \in J} \left(r_j \right), \tag{2.44}$$

has also been adopted [149, 152]. This objective is useful if a power utility wishes to focus its scheduling attention on the smallest available net reserve over the scheduling horizon. Also, this will in effect attempt to levelise the reserve margins as far as possible, but there will be more variations in the reserve compared to the case of the preferred minimisation of the SSR approach [45].

Models employing stochastic reliability measures take into account the stochastic nature of predicting energy demand and the influence of random outages, as illustrated in Figure 2.9. Once the capacity (sometimes called generation) model and the load model have been formed, they are combined (convolved) to form an appropriate risk model.

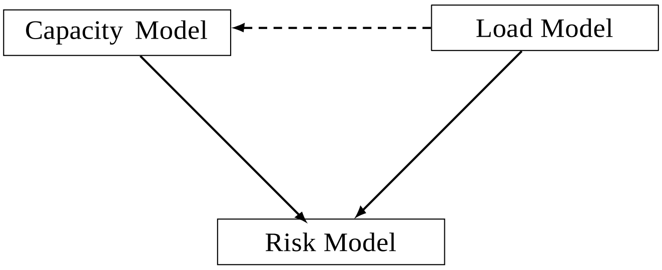


Figure 2.9: *Conceptual tasks involved in the evaluation of stochastic reliability measures for power generating systems [21, 200].*

The types of load models typically used are first described below, and this is followed by a description of the types of capacity models. A description finally follows on how these two models may be used to develop various risk models (incorporating, for example, LOLP and EENS).

The load in a power system is a stochastic process. A number of different load models may be derived from the primary data according to the extent of reliability calculations required [21, 23]. Most primary load data consist of the weekly or daily load as a percentage of the annual load, and/or the hourly peak load as a percentage of the daily peak [23, 9]. With these percentage data available and the annual peak load known, the hourly chronological load profile can be established [23]. The most widely used load model is the so-called *load duration curve* (LDC), illustrated in Figure 2.10, which represents the duration (or fraction of total) time (on the horizontal axis) during which the system load (measured in MW) is greater than or equal to a certain value (on the vertical axis). The LDC may simply be estimated without employing any mathematical formulae or algorithms by sorting the peak loads in descending order, as illustrated in Figure 2.11 for the *Institute of Electrical and Electronics Engineers reliability test system* (IEEE-RTS) established in 1979 [9].

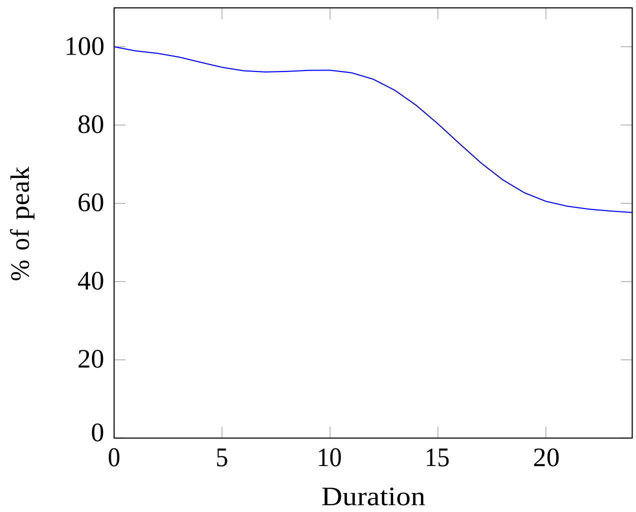
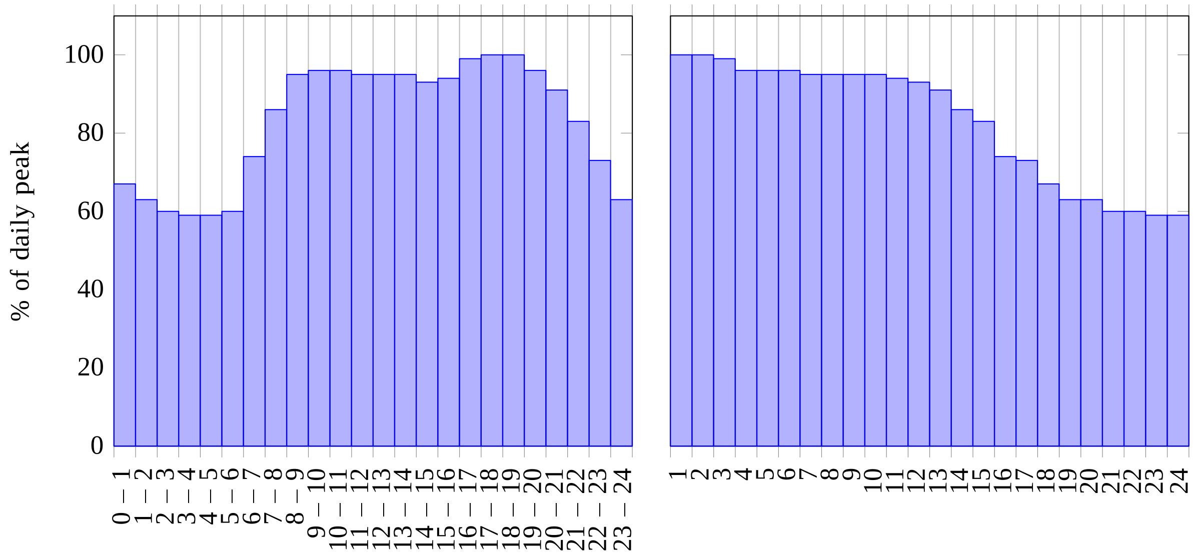


Figure 2.10: *An example of an LDC, represented as a function of time based on (hourly or daily) load curves [201, p. 114].*

The simplest and most widely used load model involves representing each day by its daily peak load. The individual daily peak loads may be arranged in descending order to form a cumulative load model, known as the *daily peak load variation curve* (DPLVC) [200], the assumption in this case being that the peak load of a day will last the entire day [23].



(a) *Load curve* (b) *Load curve sorted*

Figure 2.11: *(a) An example of peak loads per period (hour of the day) data. These data pertain to weekdays during the winter period of the IEEE-RTS benchmark system [9]. (b) A simple discrete estimation of the LDC achieved by sorting the loads in the load curve in (a) according to non-increasing order.*

A capacity model may be derived from a so-called capacity outage probability table obtained by applying techniques from reliability theory to the outage data. Such a table represents the capacity outage states of the generating system together with the probabilities of each state occurring [200]. The details pertaining to typical techniques employed to determine a power system's capacity outage probability table are described next.

In a power system, there are two fundamental types of outages, namely *scheduled outages* and *forced outages* [198]. Scheduled outages result when a component is deliberately taken out of service, usually for the purpose of preventative maintenance. Forced outages, on the other hand, result when a unit fails during service due solely to random events such as breakdown or malfunction of equipment [158, 198]. There are also two types of forced outages, namely *partial* forced outages and *full* forced outages. Partial forced outages take the form of reductions in the capability of a power generating unit, while full forced outages occur when a critical component in a power generating unit fails and the unit can no longer operate [34, p. 65]. Power equipment, such as generators, generator transformers and transmission lines are generally considered to be system components that are repairable [201, p. 104]. Boilers, steam (water) turbines and generators are often treated as single entities, called power generating units, in power system reliability analyses [201, 215].

The *failure rate* (or *hazard rate*) $\lambda(t)$ of a component is the conditional probability that the component is working before the time instant t, but develops a fault during a small interval $[t, t + \Delta t]$ thereafter. It is believed that the failure rate function for power equipment follows the well-known "bath-tub" curve in Figure 2.12 comprising three stages: an early period with a decreasing failure rate, an occasional period with a constant failure rate, and a regenerative period with an increasing failure rate [201, p. 104]. Power generating units are generally subjected to preventative maintenance service during the occasional phase, where the failure rate is a constant, *i.e.* $\lambda(t) = \lambda$, and it is assumed that the interarrival times between failure times follows an exponential distribution, so that the *mean time between failures* (MTBF) is $1/\lambda$ [201,

p. 104]. The repair rate $\mu(t)$ of a component is defined similarly to its failure rate, but there is no consensus in the literature on an acceptable distribution of a component's repair time. The distribution is often taken as an exponential distribution for convenience, in which case the *mean time to repair* (MTTR) is $1/\mu$.

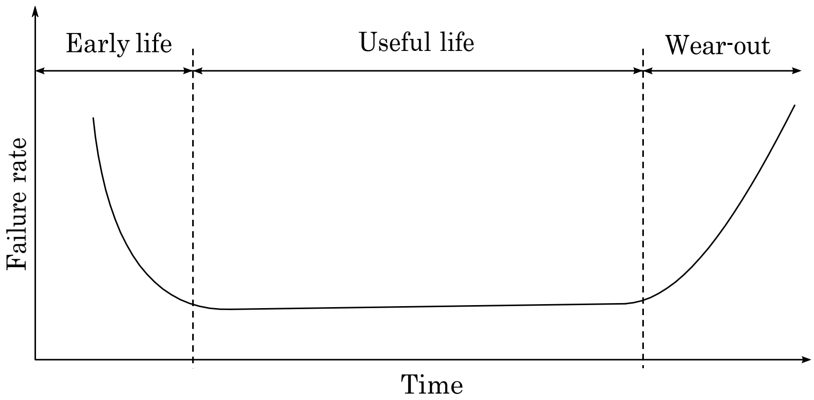


Figure 2.12: *The failure rate of a system as a function of time, represented by the well-known "bath-tub" curve [5].*

The status of a power generating unit is conventionally described as residing in one of several possible states [158]. A hierarchical representation of these possible states is shown in Figure 2.13.

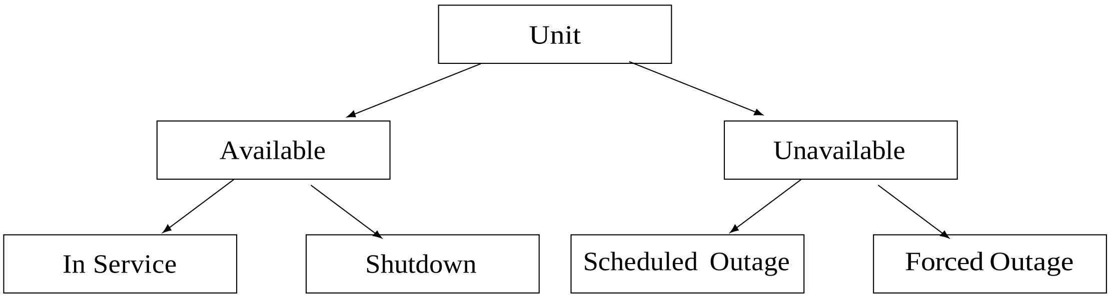


Figure 2.13: *Generating unit states [158].*

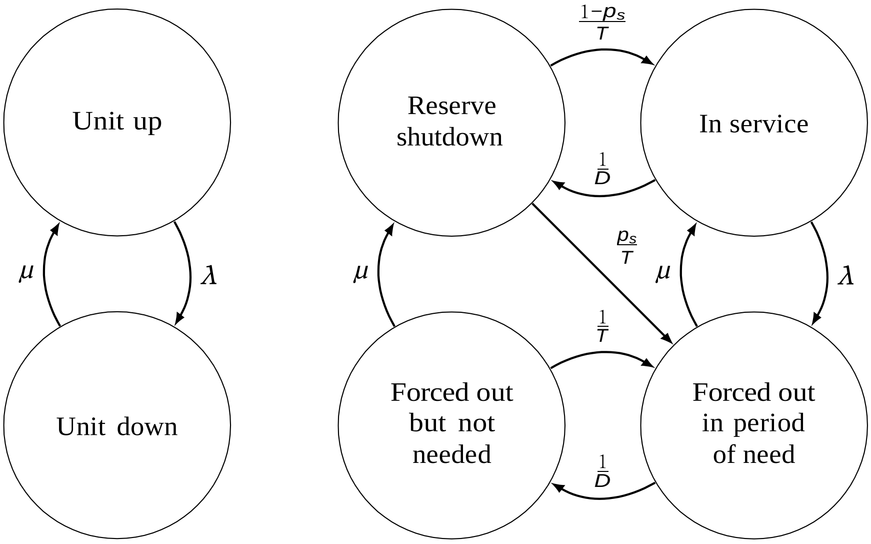
Power generating units are usually described as alternating between two states, generally applicable to base load units [8, 21, 158, 200], or between four states, generally applicable to peaking or intermittent operating units [8, 21, 200], as illustrated in Figure 2.14. In the dual-state model of Figure 2.14(a), it is assumed that a generating unit only assumes one of two states: operational (referred to as *unit up*) or in repair (referred to as *unit down*).

The FOR[8] of a power generating unit is the probability that the unit will be out of service for reasons other than PM [90]. This probability is usually estimated as the unit's unavailability [21]. For a dual-state generating unit i, the probability that the unit will be in repair is

$$q_i = \frac{\mathrm{FOD}_i}{\mathrm{FOD}_i + \mathrm{ISD}_i} = \frac{\lambda_i}{\lambda_i + \mu_i} = \frac{\mathrm{MTTR}_i}{\mathrm{MTTR}_i + \mathrm{MTTF}_i}, \tag{2.45}$$

where FOD_i is the *forced outage duration* and ISD_i is the *in service duration* of generating unit i [23, 31, 200]. Furthermore, λ_i is the expected failure rate of unit i, μ_i is the expected repair

[8]It is actually not a "rate" in reliability theoretic terms as it is the ratio of two time values [21, p. 21].



(a) *Two-state* (b) *Four-state*

Figure 2.14: *(a) A two-state model, generally applicable to base load units [8, 21, 158, 200], where μ denotes the repair rate and λ denotes the failure rate, and (b) a four-state model, generally applicable to peaking or intermittent operating units [8, 21, 200], where additionally T denotes the average reserve shut-down between periods of need, D denotes the average in service time per occasion of demand, and p_s denotes the probability of starting failure.*

rate of unit i, MTTF_i is the *mean time to failure* (MTTF) of unit i, and MTTR_i is the MTTR of unit i. The probability that unit i will be in service,

$$p_i = 1 - q_i, \tag{2.46}$$

is also called the *availability* of generating unit i. Using the values q_i and p_i in (2.46), it is possible to construct a capacity outage probability table, as illustrated for two generating units in Table 2.5. Using the binomial theorem to calculate the entries of a capacity outage probability table for many units, however, becomes too time consuming. In such a case it is preferable to employ a recurrence/convolution algorithm for this purpose [200, 201]. In the case of the dual-state model, after the i-th new unit is newly added,

$$P_i(X) = P_{i-1}(X)(1 - q_i) + P_{i-1}(X - I_i)q_i, \tag{2.47}$$

where I_i denotes the installed capacity of power generating unit i and q_i denotes the FOR of power generating unit i. Furthermore, $P_i(X)$ denotes the cumulative probability of a particular capacity outage state of X (measured in MW). This probability is obtained by means of the recurrence/convolution algorithm [21, 200, 201], where $P_{i-1}(X)$ is the cumulative probability of the capacity outage state of X before the i-th unit is added. The expression in (2.47) is initialised by setting $P_{i-1}(X) = 1$ for $X \leq I_i$. Otherwise, $P_{i-1}(X) = 0$ and $P_{i-1}(X - I_i) = 1$ for $X \leq I_i$.
The exact state probability $p_i(X)$ can be obtained similarly, changing only $P_i(X)$ and $P_{i-1}(X)$ in (2.47) to the corresponding exact state probabilities ($p_i(X)$ and $p_{i-1}(X)$, respectively). The initial conditions, however, are different [201, p. 123] in the exact state probability case. An example of a capacity outage probability table populated by means of the convolution/recurrence algorithm in (2.47) may be found in Table A.2.

Expression (2.47) may be modified to

$$P_i(X) = \sum_{s \in S} p_s P_{i-1}(X - \mathcal{C}_i) \tag{2.48}$$

Table 2.5: *The capacity outage probability table for two power generating units ($i = 1, 2$ with installed capacities I_1 and I_2), assuming that both units adhere to the dual-state model of Figure 2.14(a) [201, p. 112].*

Available capacity [MW]	Outage capacity X [MW]	Exact probability $p(X)$	Cumulative probability $P(X)$
I_1+I_2	0	p_1p_2	1
I_1	I_2	p_1q_2	$q_1q_2+q_1p_2+p_1q_2$
I_2	I_1	q_1p_2	$q_1q_2+q_1p_2$
0	I_1+I_2	q_1q_2	q_1q_2

in order to include multi-state unit representation (such as in Figure 2.14(b), for example), where $S = \{1, \ldots, s\}$ is the set of unit states, C_i^s is the capacity outage value (measured in MW) in state s for the i-th unit being added and p_s is the probability of a unit being in state s. Note that when $s = 2$, (2.48) reduces to (2.47) [21, 200].

The basic approaches adopted in the literature to develop the risk models, referred to above and illustrated in Figure 2.9, may now be described in terms of the above models for capacity outages and expected load. There are four major power generating system reliability indices, namely the LOLP/LOLE, the EENS or EUE, the *frequency and duration* and the *system-minutes* [201, p. 121]. Minimising the LOLP/LOLE is the most widely adopted technique [145, 215, 200]. The LOLP is generally defined as the probability of the available capacity (based on the expected FORs of the power generating units) not being able to meet the expected demand [145].

In order to explain the notions of the LOLP and related risk measurement calculations, consider the LDC graph in Figure 2.15. Let $L = \{1, \ldots, f\}$ denote the set of capacity outage states in the capacity outage probability table. Let O_R be the forced outage value for outage f (X in the capacity outage table), let P_R be the probability of forced outages in excess of O_R (*i.e.* the cumulative probability $P(X)$). Furthermore, let p_R be the exact probability of forced outage O_R and let Δ_R be the time interval between the intercepts on the LDC for the two successive forced outage values O_R and O_{R+1}. Then

$$\text{LOLP} = \sum_{R \in L} P_R \Delta_R \tag{2.49}$$

or

$$\text{LOLP} = \sum_{R \in L} p_R t_R \tag{2.50}$$

[21, 31].

The LOLE is equivalently the expected number of periods (hours/days/weeks) when the available capacity cannot meet the expected peak demand [201, p. 122]. Let $A = \{1, \ldots, p\}$ denote the set of periods in the LOLP calulcation. Then

$$\text{LOLE} = \text{LOLP} \times |A| \tag{2.51}$$

[201, p. 122]. Typical LOLE measurements found in the literature include days per year [185]. The United States of America and Canada stipulate that the LOLE should be less than 0.1 days/year, while Europe and Japan stipulate that it should be less than 0.3 days/year [201, p. 123].

When the *daily peak load variation curve* (DPLVC), which assumes that the peak load of the day will last the entire day [23], is used instead of the LDC, the expression

$$\text{LOLP} = \sum_{k \in A} P_k(\text{AC}_k < D_k^p) \tag{2.52}$$

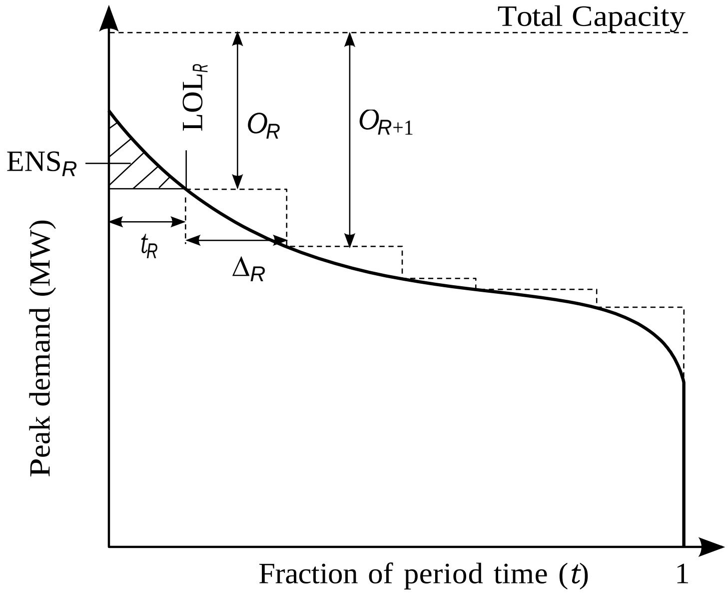


Figure 2.15: *Calculation of the loss of load (LOL) and energy not served (ENS) due to outage f on the load duration curve. Any point on the curve denotes the fraction of time during which the expected demand is greater than or equal a certain pre-determined value. Adapted from [21, 23, 31].*

is obtained [21, 200], where AC_k is the available capacity during period k, D_k^p is the expected peak demand during period k and $P_k(\mathrm{AC}_k < D_k^p)$ is the probability of loss of load during period k. Furthermore, if TC_k denotes the total capacity during period k, then

$$P_k(\mathrm{AC}_k < D_k^p) = P(\mathrm{TC}_k - D_k^p), \tag{2.53}$$

where $P(\mathrm{TC}_k - D_k^p)$ is the cumulative probability of an outage exceeding $\mathrm{TC}_k - D_k^p$, which may be obtained from the capacity outage probability table.

Since the LOLP/LOLE does not differentiate between small and large capacity outages [158] and does not indicate the frequency of loss of loads, other risk measures have also been defined in the literature. The EENS or the EUE is the expectation of the energy lost due to insufficient power supply [201]. If the ENS due to forced outage f (ENS_R) is the area under the graph in Figure 2.15, then

$$\mathrm{EENS} = \sum_{R \in L} \mathrm{ENS}_R p_R. \tag{2.54}$$

It is generally known that optimal solutions obtained under one of the risk criteria (LOLP, EENS, *etc.*) are also acceptable in terms of the others [147], but there are differences in optimal solutions resulting from the adoption of these objectives [222]. As noted in [217], the EENS criterion is preferred over the LOLP criterion, because of two potential problems with LOLP, namely that some systems having similar LOLP values may have considerably different EENS values, and systems having higher LOLP values can have lower EENS values than those with lower LOLP values.

In the above risk models, it is assumed that no scheduled outages occur. When taking into account scheduled outages, however, the risk measurements (LOLP/LOLE, EENS, *etc.*) are calculated slightly differently. This is usually achieved by partitioning the total decision period (*e.g.* one year) into constant maintenance periods (*e.g.* weeks or days) with subperiods (*e.g.* days or hours) and calculating the above risk measurements (since the capacity model has to be updated) for all the subperiods. The total risk across the entire period is the sum of these risk measurements per period [21, 22, 200, 201]. The main difference when maintenance schedules are taken into account in the calculation of the risk measurements, is that the capacity outage probability tables have to be updated. When generating units are removed from the system, typically for scheduled maintenance, they are neither available for service nor for failure. It is, therefore, not necessary to recreate the capacity outage probability table from scratch, as explained above. The original capacity outage probability table (created assuming that no maintenance is scheduled, as explained above) can merely be updated by means of a deconvolution algorithm (the reverse of (2.47) and (2.48)). That is, when a dual state unit i is removed,

$$P_{i-1}(X) = \frac{P_i(X) - P_{i-1}(X - I_i)q_i}{1 - q_i}, \tag{2.55}$$

where $P_{i-1}(X - I_i) = 1$ for $X \leq I_i$ [21, 200, 201]. When, however, a multi-state unit i is removed,

$$P_{i-1}(X) = \frac{P_i(X) - \sum_{s=2}^{S} [P_{i-1}(X - I_i)q_i]}{p_1}. \tag{2.56}$$

In [200], the LOLE_j for maintenance period j, is calculated as

$$\text{LOLE}_j = \sum_{k \in A} P_k(C_k < D^p_k) \times |A|, \tag{2.57}$$

where k is an index labelling subperiod within in maintenance period j (in [200], for example, k is the day number during the week j), C_k is the available capacity during subperiod k, and $D^p{}_k$ is the peak demand during subperiod k. In this case the scheduling objective is to

$$\text{minimise} \sum_{j \in J} \text{LOLE}_j. \tag{2.58}$$

Another popular stochastic reliability technique, very similar to the deterministic approach of levelising the reserve method, is referred to as the levelised risk method, where the LOLP for each interval is levelised (instead of the total system's LOLP or EUE) or more usually the *effective/equivalent reserves* (ER) (which incorporates a stochastic installed capacity per unit, FORs, and expected load) are levelised, normally by minimising the sum of squared ER [47]. This effective/equivalent reserve margin per time period is calculated from the *effective/equivalent load carrying capacity* for each unit and the *effective/equivalent load* for each interval [47, 145, 161]. Note that levelising this ER will essentially also levelise the LOLP for each interval [161]. Mohanta *et al.* [145] compared results obtained by adopting stochastic and deterministic reliability modelling approaches. The deterministic approach involved levelising the reserve margins by minimising the SSR over time while the stochastic approach very similarly involved levelising the "risk" by minimising the sum of squared effective reserve margins over time. Both models were solved by a GA and by a GA/SA hybrid algorithm. Results comparing which method produces the lowest LOLP values logically supported the view that stochastic reliability measures produced better schedules in terms of risk, although longer computing time were required for calculating the stochastic reliability measures, as illustrated in Table 2.6.

Note that minimising the stochastic risk measurements described above is, in principle, similar to maximising deterministic reliability measures [145]. In both cases, the aim is to ensure that there is enough available capacity to meet the expected demand. Stochastic measures, however, predict the load demands more accurately during each time period (deterministic measures merely use the peak demand per period). In addition, stochastic measures take into account the influence of random unplanned outages as well, whereas in deterministic reliability measures

Table 2.6: *Comparison of the number of generations and computing time (measured in seconds) required for two different reliability objective approaches, namely a stochastic and deterministic measure by two similar algorithms for a 6-unit test system [145].*

Objective	GA	GA/SA
Levelise reserve (deterministic)	472 gen. (503.02 sec.)	295 gen. (305.62 sec.)
Levelise risk (stochastic)	606 gen. (852.34 sec.)	297 gen. (464.12 sec.)

these unplanned outages may be less accurately incorporated by specifying a safety margin over all the time periods. Note that since the LOLP objective seeks to minimise the number of times the available capacity is less than the expected peak demand, the objective does not levelise the risk over the period k — it rather minimises instances when the available capacity is less than demand (*i.e.* there may be instances when there is considerable reserve, much more than needed), but the LOLP will not penalise these instances.

Convenience criteria

Another class of scheduling criteria sometimes employed to solve GMS model formulations is referred to as convenience criteria [124], which attempt to measure how satisfactory a maintenance schedule is (*i.e.* minimising the degree of constraint violations, minimising possible disruption to the existing schedule [124, 103], or minimising the deviation from an ideal maintenance schedule [222]). Sometimes it is hard to satisfy all the constraints in a GMS model, making it almost impossible to solve the problem instances within an acceptable computing time and using the available resources [128, 222]. The author is not aware of any work in the literature in which SO formulations of the GMS problem involve only convenience criteria. The formulations in [125, 126, 128], however, include such criteria within an MO modelling paradigm.

In [125, 126], convenience criteria, along with an economic criterion (fuel cost) and a reliability (EUE) criterion, were incorporated as the allowed number of thermal units simultaneously maintained within each plant. In [128], the degree of (reserve, maintenance crew, line flow, and maintenance duration) constraint violations were regarded as convenience criteria to be minimised (although the convenience criteria were referred to as a reliability index [128]). These criteria were incorporated into a GMS model along with economic (fuel and maintenance cost) criteria. The approaches followed to solve these and other MO GMS models are described in some detail in the following section.

Another type of convenience criteria that may be taken into account, is not the number of maximum crew available constraint violations, but the allocation of manpower [87], although this is not easily quantifiable (*i.e.* two schedules might both never require more than 25 maintenance crew during a week, but the amount of manpower allocated could vary greatly, in which case one may be more preferred than the other).

2.6 GMS solution techniques and approaches

A wide variety of solution techniques have been employed in the literature for solving instances of the GMS problem. Traditionally GMS models have been formulated as SO optimisation problems, usually incorporating the dominant scheduling criterion in the objective function, with some authors including other criteria as constraints [124]. A considerable volume of research has appeared on solving MO energy-related problems, with only a few studies involving truly MO GMS problems, as described in this section.

Solution techniques typically adopted for solving GMS problem instances include metaheuristics, mathematical programming techniques, DP, heuristic search algorithms, and fuzzy set theory [3, 90, 125, 169], the most popular techniques being metaheuristics, mathematical programming techniques and DP [90].

Mathematical programming techniques are typically used to solve SO instances of the GMS problem, and mostly include variations on the B&B method. Bender's decomposition has also been used widely due to the large dimensions of realistic GMS problem instances.

Metaheuristics are used when the dimensions of a GMS problem instance increases to the point where exact solution methodologies take too long to implement. These techniques rather often obtain very good (although not necessarily optimal) solutions within more acceptable computation time frames. Recently, metaheuristics have been used to solve GMS problem instances close to optimality within very limited computational times [168]. Typical metaheuristics applied to the GMS problem include GAs, the method of SA, *tabu searches* (TSs), *ant colony optimisation* (ACO), *particle swarm optimisation* (PSO) and hybrids of the above.

Dynamic programming[9] ideally suits the temporal nature and sequential decision process of MS problems [168, 90]. Until the 1990s, it was often used in the context of the GMS problem [90, 107, 108, 216, 221].

Fuzzy set theory is employed to address multiple objectives and uncertainties in GMS problem constraints [169], and has been used in [46, 107].

Expert systems involve development of automated solution methodology by imitating the many years of experience of experts in the field [169] and have been used in the context of GMS in [129].

Heuristic search algorithms search and improve upon the quality of solutions based on trial and error, and are comparatively seldomly used in the literature [3], which may be due to the inferior quality solutions that they often produce and the significant operator input needed [46]. According to Canto [32], however, industry operators solve instances of the GMS problem via heuristic techniques in most cases.

Constraint programming is a particularly useful method for finding exact solutions to highly constrained problem instances and has been used in the context of GMS in [91].

Game theory-based approaches have been applied to accommodate the conflicting criteria arising in deregulated power systems [90].

Some authors classify metaheuristics and fuzzy set theory as *artificial intelligence* techniques [47].

In the earlier literature on the GMS problem, instances were solved by means of mathematical programming and DP techniques, but this limits the size of the instance that can be solved due to the exponential increase of the memory requirements [46, 68] and the increasing computing time involved in the implementation of these techniques. Modern GMS solution methods therefore include metaheuristics, fuzzy set theory and expert systems.

[9] Some authors [90, 118] classify DP as a mathematical programming technique.

2.6.1 Mathematical programming techniques

Most SO GMS problems may be formulated as pure *integer programming* (IP) problems[10], 0-1 IP problems, or *mixed integer programming* (MIP) problems[11]. Generally the most common technique used to solve IP solution is the B&B method, or variations on the B&B method [209, p. 512]. Another popular IP solution technique is the cutting plane algorithm [209, p. 545]. The implicit enumeration technique, a type of B&B variation, is often used to solve 0-1 IP problems [209, p. 512]. Furthermore LP techniques, and specifically the simplex method, may be used in GMS and related energy problems when formulated as LP problems.

The simplex method

The simplex algorithm was developed by George Dantzig in 1947 and is a popular algorithm for solving LP problems. The journal *Computing in Science and Engineering* listed it as one of the top ten algorithms of the twentieth century [58]. The algorithm searches for an optimal solution along the extreme points of the feasible region of the LP problem.

The branch-and-bound method

The B&B method is a general algorithm that may be applied to solve IP (and MIP) problems. It may also be applied to solve combinatorial optimisation problems as well as nonlinear IP problems [124]. It systematically enumerates candidate solutions in the form of a tree search. The algorithm explores feasible branches of this tree, representing various disjoint subsets of the solution set. Before enumerating the candidate solutions of a branch, the quality of solutions of a branch is compared to upper and lower bounds on an optimal solution, and if it cannot produce a better solution than the best one found so far by the algorithm, the entire branch of the search tree is discarded. This method of pruning the search tree enables the algorithm not to enumerate all the possible branches (basically avoiding a brute-force enumeration), thereby eliminating the time and resources required to evaluate candidate solutions that will certainly not contain optimal solutions.

For IP problems, the B&B method is initiated by solving the LP relaxation of the IP (*i.e.* allowing the decision variables to be non-integer values), typically via the simplex algorithm. If all the decision variables solved via the simplex algorithm assume integer values, then the optimal solution to the LP relaxation is also an optimal solution to the IP problem [209, p. 513]. If, however, some of the decision variables are not integer, then those variables are branched upon by creating smaller subproblems in an attempt to find optimal integer values for these decision variables. For MIP problems, only the variables that are required to be integers are branched upon [209, p. 523] — the other continuous decision variables are treated conventionally in the simplex algorithm.

The SO MIP GMS models (based on economic criteria) in [4, 148] were solved by the B&B method. In [15], the GMS problem was solved in a deregulated environment, formulated as an MIP model. The problem was initially solved via the B&B method, without any intervention

[10]A pure IP problem is a LP problem in which all of the variables are additionally to assume non-negative integer values and is much harder to solve than an LP problem in which variables may assume real values [209, p. 375].

[11]An IP problem in which only some of the variables are required to be integers is called an MIP problem (some authors refer to such a problem as a *mixed integer linear programming* (MILP) problem).

from the ISO, after which consideration was afforded to the ISO's corrective parameters to the model (namely the maximum maintenance allowed based on installed capacity).

Benders decomposition

The direct application of MIP solution techniques is sometimes not desirable since their computation times grow exponentially with the problem size [90]. To overcome this, decomposition methods may be employed. The most popular decomposition technique is Benders decomposition [90]. Benders decomposition was first proposed by Benders in 1962 [19]. It is based on LP duality theory [118] and involves decomposing the large-scale problem into a master subproblem and several independent small-scale subproblems which are easier to solve.

The Benders decomposition method has been used to solve GMS problem instances in [32, 127]. As stated in [90], Benders decomposition applies particularly well to MS problems in power systems because of the problems' intrinsic two-stage structure, where the master problem is only concerned with the constraints related to the MS problem as well as the resources required, while load and network constraints, and fuel management constraints, may be moved to the subproblems [90].

Advantages and disadvantages of mathematical programming techniques

Solutions obtained by mathematical programming techniques are preferred over other techniques (such as metaheuristics) as they are guaranteed to produce optimal solutions. Mathematical programming techniques, however, usually take too long to implement for realistically sized GMS problem instances. Mathematical programming techniques usually struggle more to find optimal solutions to nonlinear problem formulations within reasonable computing times. This is a serious problem in the context of GMS problems where, for example, the main reliability criterion used is the SSR [90]. Mathematical programming techniques are also not applicable to problems where the objectives and/or constraints are too complex to formulate in closed form (*i.e.* where black box approaches such as simulation are the only means of evaluating functions) and are also difficult to apply to problems exhibiting multiple scheduling criteria/goals (*i.e.* multiple conflicting objectives), as is usually the case in GMS problems.

2.6.2 Metaheuristics

Due to the typical NP-hardness of GMS problems for power systems and the large dimensions of real-world instances, metaheuristics are often applied to these problems [82, 90]. Metaheuristics allow for more flexibility to deal with nonlinear and/or very complex constraints and/or objectives [90]. In fact, metaheuristics do not require the objective(s) and constraints of an optimisation problem to be mathematically formulated in closed form (*i.e.* black box approaches such as simulation may be used to evaluate a solution's objective function values and constraint violations).

Metaheuristics generally attempt to explore different areas in the search space associated with an optimisation problem in a "smart" way in order to uncover a near-optimal solution in less computation time and expend less memory resources. They evaluate only a subset of the feasible solutions, and are not guaranteed to find optimal solutions [164].

Metaheuristics may broadly be classified as trajectory-based or population-based searches. Trajectory-based metaheuristics function by manipulating and transforming a single solution at any point in time during the search process — this marks them as having more exploitation capabilities (*i.e.* they have more capability to intensify the search in local regions). Population-based metaheuristics, on the other hand, function by manipulating and transforming multiple candidate solutions (a population of solutions) simultaneously, so that an entire population of solutions evolves over time — this marks them as having more exploration capabilities (*i.e.* they have more capability for better diversification over the entire search space) [61, 187].

The four most widely known metaheuristics for general optimisation problems, according to Dréo *et al.* [61, p. V], are SA, TS, genetic and evolutionary algorithms, and ACO. In the context of the GMS problem, SA [27, 49, 82, 103, 167, 169, 165] and GAs [17, 27, 49, 200, 202] are the most commonly used metaheuristics, although TS [67, 27], ACO [89, 150] and PSO have also been applied in this context. Hybrids of the above-mentioned metaheuristics have also been applied to GMS problem instances [27, 46, 128].

Simulated annealing

SA is a trajectory-based metaheuristic. This method emerged from the independent work of Kirkpatrick *et al.* [120] in 1983 and Černỳ[35] in 1985. SA is inspired by the annealing process in metallurgy, whereby a physical system is led to a lower energy state by heating it and then allowing it to cool gradually. Similar to the physical process, the algorithm generates a neighbouring solution by perturbing the current solution. If the neighbouring solution is better then the current solution it becomes the new current solution (basically a hill-climbing algorithm), whereas if this neighbouring solution is worse than the current solution it is accepted as the new current solution with a certain probability. This probability depends on how much worse the neighbouring solution is, where a much worse solution is less likely to be accepted than a slightly worse solution. This probability also depends on a parameter called the *temperature* of the system. The higher the temperature, the more easily non-improving solutions are accepted. The reason for probabilistically accepting a worse solution is so that the algorithm can explore the search space and avoid becoming stuck in local optima. The temperature is gradually lowered over time in the hope that the SA algorithm finally reaches a *frozen* state, where a near-optimum solution is found. The SA algorithm is classified as a memoryless method, since no information extracted dynamically is used during the search [187, p. 25]. The interested reader is referred to [61, 187] for further details on the SA algorithm.

SA has been applied in the context of GMS problem instances [27, 49, 167, 169, 165, 103, 82], specifically for regulated power systems [90].

Satoh and Nara [167] illustrated the benefits associated with employing metaheuristics, specifically SA, over mathematical programming techniques in the context of three realistically sized GMS systems, as illustrated in Table 2.7. The objective was to minimise the total production and maintenance cost. As may be seen in Table 2.7, for small-scale systems the SA algorithm returned optimal results as confirmed by an IP technique, and in considerably shorter computational time. For the medium-scale system, the IP method could not obtain an optimal solution even after 310 hours (the search was terminated prior to it finding an optimal solution and the best solution found up to that point is reported). The SA algorithm, however, found a better solution (lower cost) within a much shorter time. There is, of course, no way of knowing whether this cost value obtained by the SA algorithm (namely 15 593) is the globally optimal result, but it is usually taken as a near-optimal result. For the large-scale system, IP was not even attempted due to computational limits and only the solution found by the SA algorithm is reported.

Although the study in Table 2.7 was conducted some years ago (1991), and technology has

Table 2.7: *Comparison of the results of the 1991 study by Satoh and Nara [167] applying a 0-1 IP technique, namely implicit enumeration, and the SA method to three realistically sized GMS problem instances [167].*

Size of system	Number of units $\lvert I \rvert$	Planning horizon $\lvert J \rvert$ (weeks)	Number of constraints	Simulated annealing Cost	Simulated annealing Time	Integer programming Cost	Integer programming Time
Small	15	25	5	7 768	3.5 min	7 768	41.2 min
Medium	30	40	6	15 593	272.6 min	15 602	>310.8 hrs*
Large	60	52	10	31 686	1 295.9 min	—	—

changed significantly since then, this finding generally still holds, namely that metaheuristics can typically find near-optimal solutions much quicker (and with less memory overhead) than mathematical programming (exact) techniques, especially as the problem dimensions and complexity increases. This is further highlighted by results obtained in 2014 [82], where it was found that an MIP GMS model formulated in [82] could not be solved, using GAMS 22.2 and the CPLEX solver, for real case studies because of the considerable computational burden associated with the problem in view of its NP-hardness.

Further specific advantages associated with using the SA algorithm include that it has been observed to generally achieve high-quality solutions, is a method that is very generally applicable and relatively easy to implement (encode), and offers considerable flexibility as one may add new constraints relatively easily [61, p. 44]. The main disadvantages include that the user has to specify a large number of the algorithm's parameter values, including the initial temperature, the rate of decrease of the temperature, the length of the temperature stages, and the search termination criteria. Although recommended values for the parameters of the SA algorithm may be found in the literature, these are usually problem-specific and the user generally has to employ an extensive empirical research effort to determine the best combination of values for these parameters [61, p. 44]. Another disadvantage of SA is that the algorithm requires many iterations in order to achieve high-quality results. This may cause excessive computation times for problems in which it takes relatively long to evaluate a solution's objective function values [61, p. 44]. Three specific directions are suggested in [61] to increase the effectiveness of the SA algorithm and minimise its computation time, namely to utilise/optimise adequate parameter settings, parallellise the algorithm over a number of processors, and incorporate statistical physics-based approaches to analyse and study disordered mediums [61, p. 44].

Genetic algorithms

A GA is a population-based metaheuristic. It is a stochastic search procedure, inspired by the biological processes of evolution and natural selection. The solutions to a given problem instance are the *individuals*, and each individual is associated with a *fitness* value (usually determined by the solution's objective function values). A *population* of *individuals* is randomly generated and allowed to evolve iteratively towards better solutions (based on their *fitness* values) over time until a stopping criterion is met. Operators employed by a GA include *selection*, *mutation*, and *crossover* whose parameter values have to be selected by the user. The term and usage of GAs became extremely popular after the publication of the book "*Genetic algorithms in search optimization and machine learning*" by Goldberg *et al.* [100] in 1989. The interested reader is referred to this book [61] for further details on the working of GAs.

GAs have been applied to solve GMS problem instances in [17, 27, 49, 200, 202], specifically for regulated power systems [90].

Different solution coding schemes have been proposed for GMS problems (*i.e.* to represent maintenance schedules). This mostly includes binary representation [128, 146, 185] and integer-coding [17, 47] based on the maintenance starting times. Research performed to compare these two techniques for GMS problems points to integer-coding as the superior encoding scheme [17], being more efficient [202] since it generates a smaller search space [47], reduces the probability of infeasibility during the search process, and avoids the necessary overhead of coding a solution as a binary string (since binary length is longer than integer length) [90].

Real number encoding [17] has also been employed in the context of GMS problems. For example in [200], solutions were represented as real values between 0 and 1, obtained by dividing the maintenance starting time values by the number of periods in the planning horizon ($|J|$).

Tabu searches

The TS algorithm is a trajectory-based search metaheuristic which was first proposed by Glover [98] in 1986. It is inspired by and incorporates mechanisms of human memory. It avoids revisiting solutions by recording the recent history of the search in a short-time memory called a *tabu list*.

It adopts an approach mostly opposite to the SA algorithm, which does not not utilise memory at all and thus cannot learn from lessons of the past [61, p. 7]. Like SA, a neighbourhood of solutions is generated from the current solution (in SA only one neighbour is generated). The best neighbouring solution found replaces the current solution, even if this solution is worse than the current solution. Hence when a local optimum is encountered, the search carries on by selecting a candidate worse than the current solution, in an attempt to avoid becoming stuck in a local optimum. In order to improve the performance of the TS, aspiration criteria, intensification strategies and diversification strategies are sometimes incorporated [61, p. 10].

Dréo *et al.* [61] noted that for certain optimisation problems, TS yields excellent results. In addition, in its basic form, it contains fewer parameters than does SA, which makes it easier to implement. The various additional mechanisms, however, like intensification and diversification strategies, often bring about notable complexity [61]. Talbi [187, p. 141] noted that theoretical studies carried out in respect of TS algorithms reveal that TS algorithms yield weaker results than those established by SA.

TS has only been applied to solve instances of the GMS problem in [67, 27], specifically for regulated power systems [90].

Ant colony optimisation

ACO is a population-based metaheuristic which was first proposed by Colorni *et al.* [41] in 1991. It is a member of the class of swarm intelligence algorithms, which are collectively inspired by the behaviour of biological species such as ants, bees, wasps, termites, fish, and birds [187]. The two most successful swarm intelligence algorithms are ACO and PSO [187, p. 240]. ACO is specifically inspired by the foraging behaviour of ant colonies, whose members are individually equipped with very limited visual faculties, but which are nevertheless usually able to find the shortest path between a food source and their anthill. The key to the ants' success is attributed to the indirect communication with one another by means of the dynamic modifications of the their environment, known as *stygmergy* [61, p. 13].

ACO begins with a random set of candidate solutions to an optimisation problem instance. Over the iterations of the search, "ants" deposit pheromone on the components of promising solutions. In this way, the environment of the decision space is iteratively modified and the search is gradually biased towards more desirable regions of the search space, where optimal or near-optimal solutions may be found [89].

In order for ACO to be applied to GMS problems, instances the problem first have to be mapped onto (represented as) graphs, expressed as sets of points at which decisions have to be made (decision points) and sets of options that are available at each decision point (decision paths). For the GMS problem, these decision points consist of the generating units that need to be maintained and the corresponding decision paths are the potential starting times for maintenance [89].

ACO has been applied to solve instances of the GMS problem in [83, 87, 89, 150], specifically for regulated power systems [90].

Particle swarm optimisation

The other successful swarm intelligence algorithm is PSO. PSO is also a population-based meta-heuristic which exhibits many similarites with GAs [90]. PSO is originally attributed to Kennedy, Eberhart and Shi [117, 177] in 1995. It mimics/simulates the social behaviour (specifically the movements) of organisms such as flocks of birds or schools of fish when they attempt to find a source of food by information exchange [187, p. 247].

In the basic model, a swarm consists of n particles moving around iteratively in the search space of the optimisation problem, where the position of a particle represents a candidate solution. Each particle is biased by its own position and velocity, but guided by previous individual or collective success in respect of uncovering local optima [187, p. 247].

The method was first intended only for continuous optimisation problems, but it has since been adapted to be applicable to integer and binary optimisation problems. PSO has been applied to solve instances of the GMS problem in [66, 122, 185].

Comparison of metaheuristics

The celebrated *no free lunch theorems* [211] state "*...that any two algorithms are equivalent when their performance is averaged across all possible problems*" [210]. This means that no optimisation algorithm can be claimed to perform better than another algorithm in all problem instances. These theorems are, however, subject to intense and controversial debate [62]. The situation is different when treating specific classes of real-world problems, as there are general tendencies in specific classes of problems [151]. This was illustrated in the paper by Droste *et al.* [62] entitled *perhaps not a free lunch but at least a free appetizer* in which it was shown that a particular algorithm performs better over a subset of the entire function set in their example. This means that there can be a better algorithm to solve restricted classes of optmisation problems. In this paper they explained and tied together the seemingly contradictory statement of the no free lunch theorems and results in the literature which claimed that some algorithms performed better in certain problem instances.

Choosing the correct metaheuristic (or other optimisation technique) for a given optimisation problem is not trivial. Dréo *et al.* [61, p. 17] elude to this difficulty as choosing an "efficient" method, able to produce an "optimal" solution — or a solution of acceptable quality — within a "reasonable" computation time.

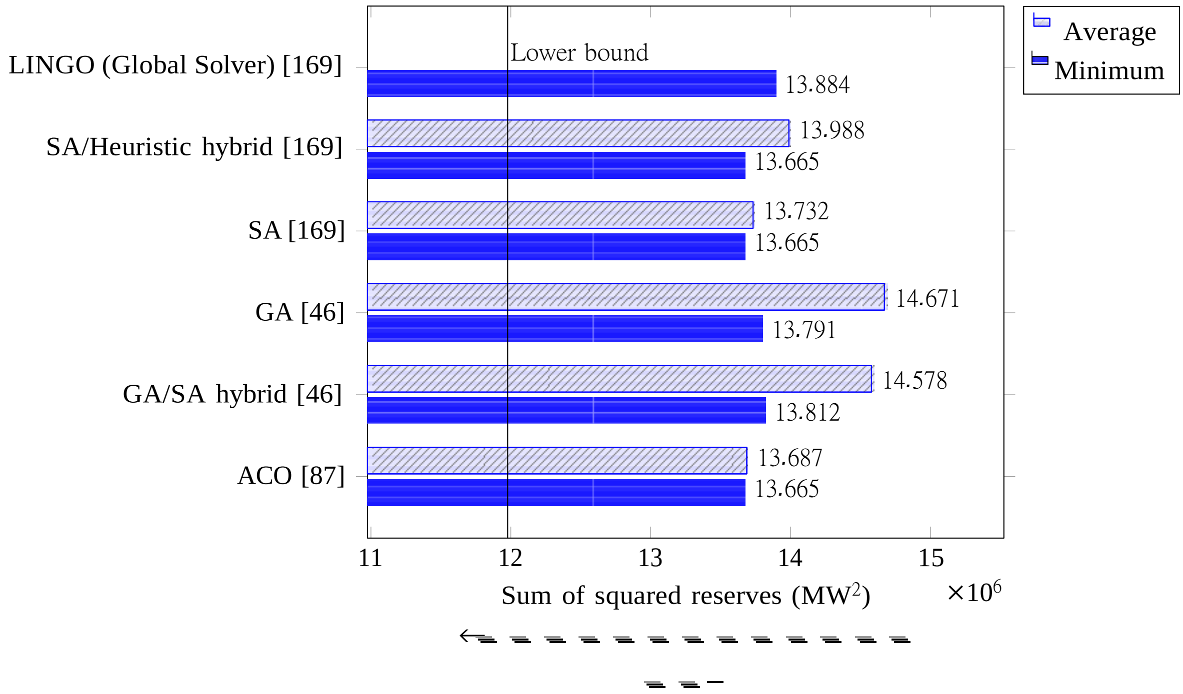


Figure 2.16: *Comparison of the relative performances of different metaheuristics, and an IP solver (LINGO), when used to solve a 21-unit system from in the literature [169]. The (reliability) objective was to minimise the SSR.*

Dréo *et al.* [61, p. 17] claim that the "optimal" adjustment of the various parameters of the metaheuristic, which can be recommended theoreticlly, is often of little help in practice, because it induces additional computation time. Generally the user must embark on extensive fine tuning experimentation (in terms of parameters, solution encoding, constraint handling, *etc.*) for any of the metaheuristics mentioned above to yield near-optimal results to a given optimisation problem instance within a reasonable computation time. It may thus be argued that any metaheuristic applied correctly to a given optimisation problem is a good choice.

The results obtained when solving a well-known GMS benchmark system, the 21-unit system [47], by means of different techniques are summarised in Figure 2.16. The relative performance of three different types of metaheuristics (SA, GA, and ACO) and two additional hybrids are compared to the IP solver LINGO (version 9.0) in the figure. The objective was to minimise the SSR (a reliability criteria). LINGO was not able to find an optimal solution within a cut-off time of 12 hours, and so the best feasible result found up to that point is reported instead. This demonstrates the difficulty of solving a GMS problem instance exactly within a reasonable computing time [169]. A lower bound on the objective function value is reported as 11 997 600 MW^2, as calculated by LINGO after 12 hours.

ACO, SA, and a SA/Heuristic hybrid found the best known objective function value of 13 665 000 MW^2, with the GA and GA/SA hybrid relatively close. Even for this relatively small GMS system (calling for maintenance scheduling of 21 generating units over a period of 52 weeks), all of the metaheuristics found better results than the IP solver (because of the quadratic objective function) after 12 hours, validating the use of approximate solution approaches towards solving instances of the GMS problem. It would be of great interest to compare the computing times required by the different metaheuristics. These computing times are, however, not reported as far as the author is aware, and even if they were it would be difficult to compare these computing times since the algorithms were implemented on different platforms.

In [82] it was furthermore found that for nine 26-unit and and nine 36-unit test systems (18 in total), ACO slightly outperformed SA by an average of 0.76% (across all 18 systems) in terms of minising a cost objective (consisting of start-up, production, and maintenance cost), with Fattahi *et al.* [82] claiming ACO to be superior in the context of these test systems [82]. SA was, however, much faster, requiring on average 394 seconds less computation time (a 39% improvement) than ACO.

2.6.3 Dynamic programming

DP, also known as dynamic optimisation, is a technique for solving complex or large temporal problems by breaking them down into a collection of simpler, smaller nested subproblems [209, p. 961] and putting the solutions to these subproblems together, working backwards in time, in order to arrive at a solution to the original problem.

It has been stated that DP ideally suits GMS problems [3, 90, 124, 216]. This is attributed to the fact that DP is especially suitable for problems where (1) a sequence of interrelated decisions has to be taken, (2) the objective function need not be a continuous function of decision and state variables, and (3) an analytical form of the objective function or constraint functions is not required (as long as one can obtain function values at a given state) [216]. Until the 1990s, DP was often used [90] to solve GMS problem instances [107, 108]. In [107, 108], DP was combined with fuzzy logic.

The "curse of dimensionality", however, limits the application of this method to small sized GMS problem instances [3, 124, 216], as in the case with mathematical programming techniques. DP with successive approximations has, however, been developed to reduce the problem dimensions [124]. The DP with successive approximations technique does converge, but does not guarantee a global optimum [124]. Like DP, the DP with successive approximations was used earlier on for solving GMS problems as in [216, 221].

2.6.4 Fuzzy set theory

Fuzzy set/logic theory is a mathematical theory in which set membership is represented by real numbers (usually between 0 and 1). The variables are considered to be fuzzy, which corresponds to the vagueness or fuzziness of their values [133, p. 4]. In essence, fuzzy sets are concerned with the degree of truth of a variable's value or membership category. This is not to be confused with a degree of likelihood, which is dealt with in probability theory [133, p. 2]. The building block of fuzzy logic is the membership function of a fuzzy set or number [168, p. 38]. The interested reader is referred to the book entitled *Fuzzy logic for planning and decision making* [133] for more information on fuzzy logic.

Fuzzy logic may be used to handle the uncertainties and multiple objectives present in GMS problems. It is typically used in conjunction with other solution techniques, such as DP and metaheuristics [168, p. 40].

In [107, 108], multiple GMS objectives (relating to reserve margin and production cost criteria) and soft constraints (manpower, time windows, and geographical/exclusion constraints) were *fuzzified* by triangular and trapezoidal membership functions. In [107], the membership functions' parameter were adjusted by a heuristic and a GMS model was solved by DP, whilst further work in [108] by the same author, similarly involved solving the same model by DP, but the membership functions' parameter were adjusted by a GA (metaheuristic), with the latter providing better results in both objectives, as illustrated in Table 2.8.

Table 2.8: *Comparison of results obtained by a conventional fuzzy DP system where the membership functions' parameters were adjusted according to a heuristic and a GA. These results are also compared to the Taiwan Power Company's (TPC's) actual schedule for 1992 [108].*

	Decrease in production cost	Reserve margin (MW) Minimum	Maximum
TPC actual schedule	0.00%	59	1 112
Fuzzy DP	0.03%	317	1 727
Fuzzy DP with GA	0.15%	320	1 740

In [146], fuzzy logic was used to take into the account the uncertainties related to FOR (by considering the uncertainties related to the MTTF and the MTTR for the generating units) and the uncertainties due to load forecasting which led to fuzzy LOLP indices.

Further usage of fuzzy logic theory in GMS problems occurred in [48, 69]. Fuzzy logic seems to be a promising approach towards solving GMS problem instances [3], especially since many of the uncertainties and MO goals present in GMS models can thus be addressed [168, p. 40]. Fuzzy DP [3] and GAs with fuzzy evaluation functions [48] are an effective and practical approach towards finding good GMS solutions. The drawback of fuzzy logic approaches is the substantial amounts of data and expert knowledge required for their implementation [90].

2.6.5 Knowledge-based/expert systems

Expert systems emulate the decision-making ability of a human expert. Most power utilities employ an expert team to plan MS work [129]. Conventional mathematical programming, however, cannot include all these conditions completely due to the complexity involved in scheduling. As noted in [129], knowledge-based expert systems can be applied as alternatives. That is, the knowledge of experts may be included in an expert system in order to generate good GMS solutions.

In [129], a knowledge-based expert system was used to solve an instance of the GMS problem for the Taiwan Power Company's network. Knowledge-based expert systems have further been used in the context of the GMS problem in [14, 92]. Expert systems have also been used in conjunction with other solution techniques, for instance in [175], where an expert system was combined with a fuzzy modelling environment.

2.6.6 Heuristics

A heuristic[12] is defined as any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal, but sufficient (or good enough) for achieving the immediate goals. A heuristic attempts to find a solution to an optimisation problem by employing loosely defined rules-of-thumb and trial-and-error approaches, and is usually used when the quest for an optimal solution is impractical (as a result of time or memory constraints). In addition, heuristics may speed up the process of finding satisfactory solutions.

In the context of the GMS problem, heuristics require significant operator input and may even fail to find feasible solutions [3, 45]. These methods are seldom used [3] to solve instances of the

[12]The word *heuristic* is derived from the Greek word *heuriskein* which means "to find." Metaheuristics are, in fact, extensions of heuristics, created by controlling and tuning basic heuristic algorithms, usually by employing memory and learning processes.

GMS problem. According to Canto [32], however, the industry operators in most cases solve instances of the GMS problem via heuristic techniques.

2.6.7 Constraint programming

Constraint programming (CP) is a modern programming paradigm in which relations between variables are stated in the form of constraints, particularly in combinatorial problems [43]. When appropriate variable propagation rules are employed, CP is especially useful for finding feasible solutions to highly constrained optimisation problems [90].

CP has been found to work well in the context of energy-related problems [24, 26, 99], but has not been used extensively in the context of GMS problems (the only case of such an application that author could find is [91]). This scarcity is attributed by Froger *et al.* [90] to the fact that CP is less suitable to problems in which the main goal of the algorithm is to find near-optimal, as opposed to exactly optimal, solutions, which is typically the case in GMS problems.

2.6.8 Game theory

Game theory deals with mathematical models of conflict and cooperation between intelligent rational decision makers [209, p. 803]. It has thus been applied more extensively in deregulated power systems, where conflicting market environments arise. Every GENCO, for example, attempts to predict its competitors' decisions so as to stay one step ahead [90]. GENCOs are also often able to manipulate market prices through capacity withdrawal [37].

In [37, 119] a game theoretic model/framework was presented for analysing strategic decisions related to MS and generator output for the GENCOs in a deregulated power system. In [144], a model was presented for the coordination procedure of an ISO and three GENCOs. The results thus obtained indicated that the GMS for a profit-oriented GECNO may be modified to satisfy the reliability requirements of the ISO. The strategies adopted by the GENCOs in the studies mentioned above were defined by a Nash equilibrium of the game [90].

2.6.9 Simple multi-objective GMS modelling approaches

Most real-world problems are MO in nature [183], including GMS problems [147, 222]. If there are trade-offs between the model objectives in question this will lead naturally to the notion of Pareto optimality where certain solutions are not dominated by other solutions and the goal is to find the smallest set of such non-dominating solutions. MOO problems are problems containing multiple objectives which have to be optimised simultaneously. A few researchers have lately shifted their focus to formulating GMS models as MOO problems and using appropriate solution techniques.

Traditionally the techniques used to solve instances of the GMS problem focussed on optimising one objective and including all but the one dominant criterion as model attainment constraints [124]. Other GMS MO modelling approaches have focussed on optimising the objectives separately [67, 149, 216], weighting and summing scheduling criteria into one objective function [128], or attempting to find objective function values that are as "close" as possible to some ideal point/value [125, 126]. These methods are described in some detail in this section. It is, however, stressed that these methods are not truly MOO techniques, since they do not iteratively attempt to find some kind of Pareto front or approximation thereof, but only return one or a handful of solutions. An important distinction is made in this dissertation, between these simple (and inferior) MO modelling approaches and Pareto-based MOO (generally known in the operations research literature just as MOO). Pareto-based MOO techniques have been applied to energy-related problems, but relatively infrequently to GMS problems (the author was able to find only two papers in which Pareto-based MOO techniques were employed).

The importance of MO approaches in MS was already pointed out in 1991 by Mukerji *et al.* [149], who discussed the difference between schedules computed by pursuing two alternative scheduling criteria separately. The objectives involved minimising the production cost (an economic criterion) and maximising the minimum reserve (a reliability criterion). The schedules were computed by means of the implicit enumeration IP algorithm. Mukerji *et al.* also pointed out the benefits that may be achieved by relaxing some of the constraints. These results are illustrated in Figure 2.17. As may be seen in the figure, the difference between the schedule maximising only the minimum reserve (● in Figure 2.17) and that minimising only the production cost (■) decreased the production cost from ⍰327.72×10^6 to ⍰327.45×10^6 (a mere 0.08% improvement), but lowered the minimum reserve from 165 to 152 MW (a 7.88% worsening). Relaxing some of the model's constraints further resulted in the improvement of both objective function values. The schedule maximising only the minimum reserves, but relaxing some of the model constraints (○), increased the minimum reserve from 165 to 193 MW (a 16.97% improvement) and decreased the production cost from ⍰327.72×10^6 to ⍰327.09×10^6 (a 0.19% improvement). On the other hand, the schedule minimising only the production cost, but relaxing the model constraints (□), increased the minimum reserve from 152 to 160 MW (a 5.26% improvement) and decreased the production cost from ⍰327.45×10^6 to ⍰327.08×10^6 (a 0.11% improvement). These results show that the optimum reliability and production cost values could be improved upon by relaxing some of the operating constraints. As noted in [149], the potential savings obtainable must be evaluated against the inconvenience and cost incurred by relaxing these constraints.

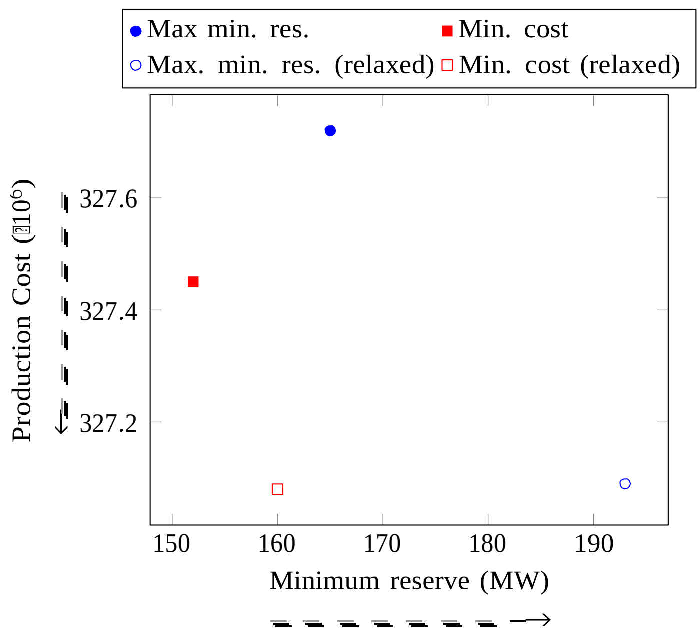


Figure 2.17: *GMS case study results in [149] comparing two different optimisation strategies, namely maximising the minimum reserve and minimising the production cost. In addition, results are also included for the same two strategies whilst relaxing the model constraints. The production cost results are expressed in a fictitious unit of currency (⍰), since the heat rate and fuel cost data were modified slightly from the true values for the case company.*

Leou [128] in 2006 combined a GA with the method of SA (a GA/SA hybrid) to solve a bi-objective GMS model where the objectives were to minimise fuel cost and maintenance cost (an economic criterion) and to minimise reliability, crew, line flow, and duration constraint violations (a convenience criterion). These two objectives were, however, weighted together to form a single objective function to be minimised.

Yamayee *et al.* [216] in 1983 used DP to minimise the production cost and optimise a reliability measure (LOLP) separately, as illustrated in Figure 2.18. The production cost per unit was an average estimated cost (a constant $/MWh coefficient). Minimising the LOLP objective (● in Figure 2.18) resulted in an improvement of 0.073 (22%) and minimising the production cost objective (■) resulted in an improvement of a mere $5 000 (0.002%). In addition, the run time required to minimise the production cost objective was was much longer, 3:43 miniseconds[13] compared to 1:21 miniseconds (175% longer). This was attributed to the production cost being calculated from the individual energies of the units whereas calculation of the reliability objective involved calculation for the entire system. From these results Yamayee *et al.* [216] concluded that optimisation of the reliability objective (LOLP) is more desirable than minimising production cost.

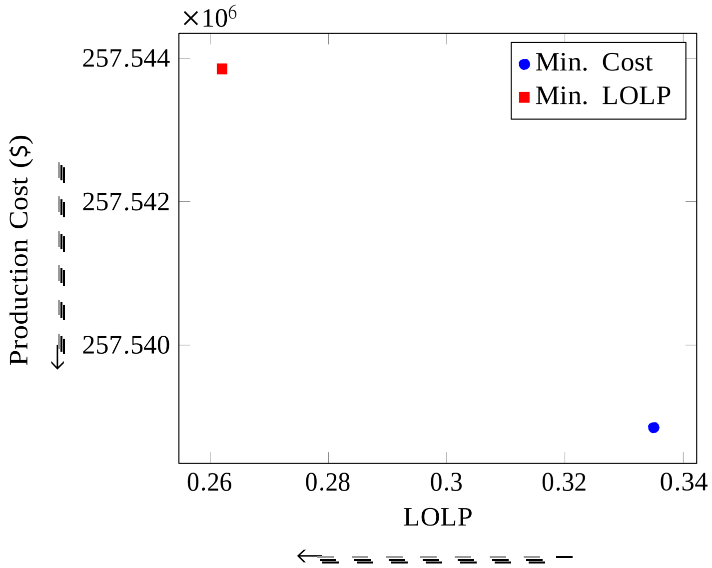


Figure 2.18: *LOLP and production cost results obtained by Yamayee* et al. *[216] when minimising each of these two objectives separately. DP was used to solve the model and the data were obtained from the GMS 21-unit benchmark system [216].*

Kralj and Rajaković [125, 126] adopted a “multi-objective” B&B algorithm in 1994 and 1995 to solve instances of a tri-objective GMS model including an economic objective (minimisation of fuel costs), a reliability-related objective (minimisation of the expected unserved energy over time) and a convenience-related objective (minimisation of constraint violations). Their results are illustrated in Figure 2.19. A solution was found by minimising the Euclidean distance between the ideal point and the objective function vector, which contained the values of the three objectives. This was an important study, because it is generally claimed to be the first time that a MO solution approach was used to solve a GMS problem.

Moro and Ramos [147] in 1999 proposed a two-stage goal programming approach, solved by the B&B method, for instances of a bi-objective GMS model involving thermal units. A case study was performed for a Spanish electric power system, consisting of 71 thermal generating units (8 nuclear units (7 401 MW), 36 coal units (10 675 MW) and the rest being oil/gas units (7 910

[13]The author assumes this to be another unit of measure, perhaps milliseconds.

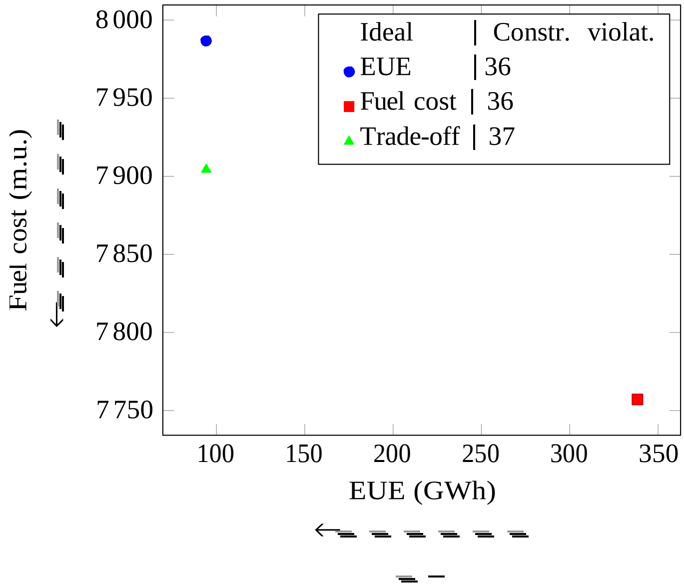


Figure 2.19: *Fuel cost (monetary unit (m.u.)), EUE and constraint violation results obtained when solving the 21-unit GMS benchmark system [125, 126].*

MW)) grouped into 16 thermal plants. The system therefore had a total installed capacity of 25 986 MW. The first objective was to minimise the total operational cost (TOC), including linear fuel costs, linear *operation and maintenance* costs , startup costs, and some penalty costs for unserved power. The second objective was to levelise the reserve margins (a reliability criterion), achieved by minimising the sum of the differences between the reserve margins percentage values of consecutive periods (defined as the available capacity divided by the period peak load demand). They first minimised the TOC objective and then optimised the reliability objective in a attempt to find the best trade-off between these two conflicting objectives. During the second stage the reliability objective was optimised whilst a constraint was added, constraining solutions to a minimum TOC value the user was willing to accept (for an improved reliability value), defined as

$$\mathrm{TOC}_2 \leq \mathrm{TOC}_1(1+\beta), \tag{2.59}$$

where TOC_1 is the minimum total operating cost found during the first stage, TOC_2 is the total variable operating cost found during the second stage (which levelises the reserve margins), and β represents the amount of cost increase the maintenance planner is willing to tolerate, so as to improve the reserve margin objective. In the case study, β was set at 3% (with the final problem solved obtaining $\mathrm{TOC}_1 = \$2\,625 \times 10^6$ and $\mathrm{TOC}_2 = \$2\,704 \times 10^6$). Moro and Ramos noted that more "reliable" solutions could have been obtained by increasing β. No further analysis was, however, reported in this respect (*i.e.* only one "optimal" solution was found, based on this single β-value).

2.7 Pareto-based optimisation in energy problems

MOO is a subfield of *multiple criteria decision making* that is concerned with optimisation problems requiring the simultaneous optimisation of more than one objective function. In this type of problem, optimal decisions usually have to be taken in the presence of trade-offs between two or more conflicting objectives. In an MOO problem there typically does not exist a single solution that simultaneously optimises each objective. In that case, the objective functions are said to be *conflicting*, and there exists a (possibly infinite) number of so-called Pareto optimal solutions. A solution is called non-dominated, *Pareto optimal*, Pareto efficient or non-inferior, if none of their objective functions can be improved in value without degrading some of the other objective values.

A critical element to the MO paradigm, is the *decision maker*, which is assumed to be a person who can provide further preference information concerning the desirability of Pareto optimal solutions. MOO methods may be classified as follows [143] with respect to the role of the decision maker:

No preference methods are methods in which the decision maker is involved. A neutral compromise solution is instead identified without preference information provided by the decision maker.

***A priori* methods** are methods in which the decision maker articulates preferences before commencement of the optimisation process. Well-known examples of *a priori* methods include the utility function method, the lexicographic method, and goal programming.

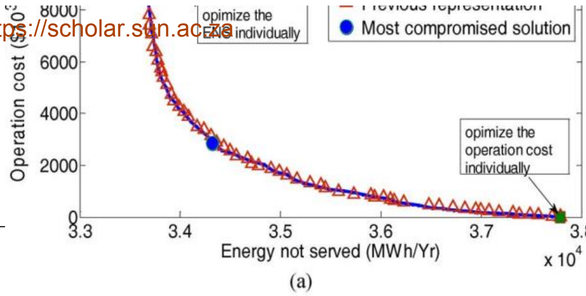
***A posteriori* methods** aim to generate a representative set of Pareto optimal solutions after which the decision maker subjectively chooses the most suitable one among them. Most *a posteriori* methods are either mathematical programming-inspired techniques (such as the *normal boundary intersection* technique, the *modified normal boundary intersection* technique, the *normal constraint* technique, the *successive Pareto optimisation* technique, and the *directed search domain* technique) where algorithmic execution is repeated and each run of the algorithm produces one Pareto optimal solution, or metaheuristics (*e.g.* evolutionary algorithms) where one run of the algorithm produces an entire set of Pareto optimal solutions.

Interactive methods allow the decision maker to guide the search by alternating optimisation and preference articulation (*i.e.* the decision maker is shown a Pareto optimal solution and asked to describe how the solution may be improved). This process is repeated until a satisfactory solution is obtained.

Recently there has been a focus on applying MO metaheuristics to solve energy-related optimisation problems. MO metaheuristics have become very popular in MOO due to their ability to (1) find multiple approximately Pareto optimal solutions in a single run, (2) work without objective function derivative information, (3) converge speedily to near-Pareto optimal solutions with a high degree of accuracy, (4) handle both continuous function and combinatorial optimisation problems with ease, and (5) be less susceptible to problems arising from the shape or continuity properties of the Pareto front. These issues are a real concern for mathematical programming techniques [183]. In addition, MO metaheuristics are not as susceptible to problems arising as a result of noncontinuous objective functions and large-scale search spaces, which are often insurmountable difficulties for traditional solution methods [217].

Most MO metaheuristics are evolutionary algorithms [61, p. 207] and are referred to as *multi-objective evolutionary algorithms* (MOEAs). The most popular MOEA is the *non-dominated sorting genetic algorithm*-II (NSGA-II) [217].

In 2009, Yang *et al.* [217] used the NSGA-II, to solve a tri-objective MS problem for regulated power systems (see Figure 2.3(a)), where the decision variables define the extent of maintenance strategies (no maintenance, minor maintenance, or major maintenance) of individual components in typical substations (very similar to the situation considered in the GMS problem). The conflicting objectives included an overall operation cost (consisting of the expected inspection cost and maintenance cost), the expected failure cost (which is a function of the failure probability and average cost of failure for the components in the power system) and a reliability



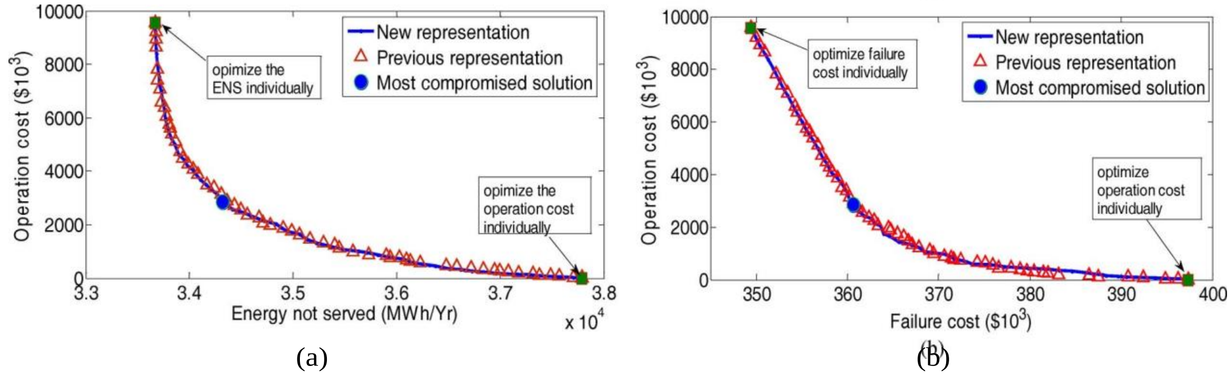


Figure 2.20: *Pareto fronts for the entire IEEE-RTS system proposed in [217]. (a) Operational cost versus energy not served, and (b) operational cost versus failure cost.*

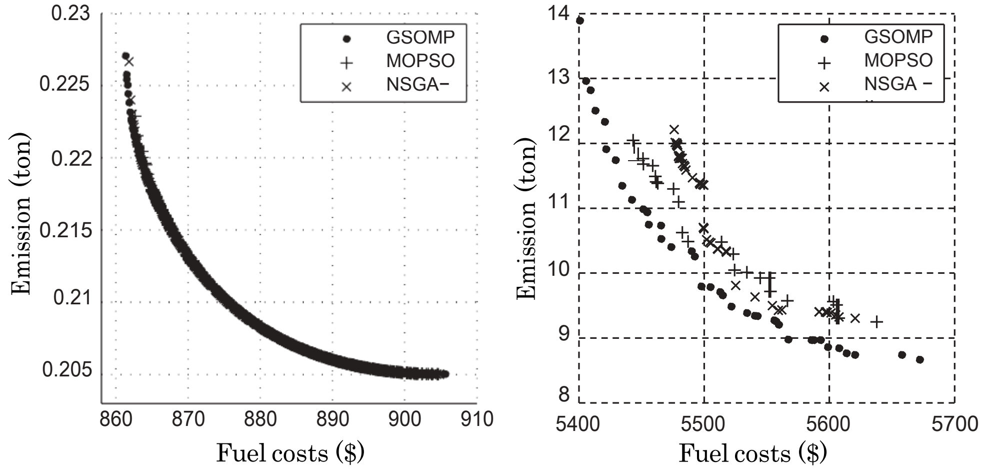
objective (energy not served). The final results obtained by Yang *et al.* [217] are shown in Figure 2.20. Naturally, the operational cost (consisting of the expected inspection cost and maintenance cost) increases as the reliability is improved (energy not served is reduced), since more frequent maintenance has to occur in this case [217], as is illustrated in Figure 2.20(a). Also, the more maintenance is done, translating into an increase in the operational cost consisting of inspection and maintenance costs, the lower the expected failure cost will be, as is illustrated in Figure 2.20(b). Furthermore, as may be seen in Figure 2.20, the solutions occupying the two extremes of each Pareto front in the figures are consistent with those obtained when Yang *et al.* [217] optimised each objective individually. This means that the elitism of the NSGA-II effectively maintained the extreme solutions so that more diverse choices could be provided.

In 2012, Guo *et al.* [102], used a novel MOEA, called the *group search optimiser with multiple producers* (GSOMP), to solve a *dynamic economic emission dispatch* problem[14] for power systems. The two conflicting objectives to be minimised were the fuel costs and emissions. The GSOMP algorithm is an extension of the SO *group search optimiser* (GSO), a newly developed population-based optimisation algorithm inspired by animal searching behaviour [106]. The GSMOP was developed by Wu *et al.* [214] to solve the MO problem of the placement of Flexible AC Transmission System devices in reactive power dispatch problems. The GSOMP algorithm performed well in the context of the dynamic economic emission dispatch problem [102] compared to the other well-known MOEAs, such as the NSGA-II and *multi-objective particle swarm optimisation* (MOPSO), as illustrated in Figure 2.21. It was also further reported that GSOMP consumed much less computation time than the NSGA-II algorithm, but required more computation time than the MOPSO algorithm. Both the NSGA-II and MOPSO algorithms have been applied to solve ED problems [102].

Pareto-based multi-objective optimisation in GMS problems

Very few GMS studies in the literature employ Pareto-based MOO techniques (the author was only able to find two papers in which Pareto-based MOO techniques were employed) and those that do are limited to deregulated power systems (see Figure 2.3(b)). This is not surprising, since MOO modelling techniques and the use of MOO metaheuristics are relatively new fields and as has been mentioned, power industries have only relatively recently shifted more towards

[14]Note that this is not a GMS problem, but is rather an advancement of the ED problem, a typical subproblem of the GMS problem, as described in §2.3.2.



(a) *IEEE 30-bus system (Case 1)* (b) *IEEE 118-bus system (Case 2)*

Figure 2.21: *Approximate Pareto fronts obtained by three MOEAs in respect of the dynamic economic emission dispatch problem solved in [102].*

deregulation of the market. In addition, the agents in deregulated power systems naturally pursue many conflicting goals.

In 2014, Zhan *et al.* [218] proposed an MO GMS problem for deregulated systems with five objectives, namely the separate profits of three producers, a system reliability objective for the ISO, and the total generation costs of all three producers. System reliability was pursued in the form of minimisation of the standard deviation of a reliability index, which guarantees similar reserve capacities in the different time periods (*i.e.* levelising the reserve margin). The total generation cost to be minimised consisted of fuel cost, start-up costs, and maintenance cost. The profit in the market environment was calculated by subtracting these generation costs from the revenue of each of the three producers. The results obtained in the context of two benchmark systems, involving the 32-unit IEEE-RTS in [218] are shown in Figure 2.22. The GMS model was solved by the GSOMP, proposed in [102]. Although the generating cost was formulated quite extensively, including sophisticated start-up cost and expected maintenance cost formulations, in addition to quadratic fuel costs, the generators' output levels were not "optimised" in any way (as in the ED problem), but were rather generated/modified as random values between allowable minimum and maximum values.

Very similar to the paper by Zhan *et al.* [218], Chen *et al.* (Zhan being a co-author) [38] solved an almost identical MO GMS model (with one objective fewer, namely the total generation cost being excluded), but using the NSGA-II algorithm instead.

The author is not aware of any further MOO research (which produce any kind of Pareto front) in GMS problems. This is confirmed in Table A1 of the GMS literature review paper [90], where only three references (out of 77 listed) were categorised as using MO approaches — one being the work reported above by Zhan *et al.* [218]. The other two references [126, 147] (described in the previous section) are not truly MOO approaches, since they only produce a very small subset (two to three solutions) along the Pareto front. This sentiment is also shared in [87, p. 10]. Kralj and Petrović [126] adopted a "multiobjective" B&B algorithm to minimise the Euclidean distance between an ideal point and the objective function vector, which contained the values of the three objectives, as was illustrated in Figure 2.19 (only three points were obtained). Moro and Ramos [147] proposed a two-stage goal programming approach (solved by

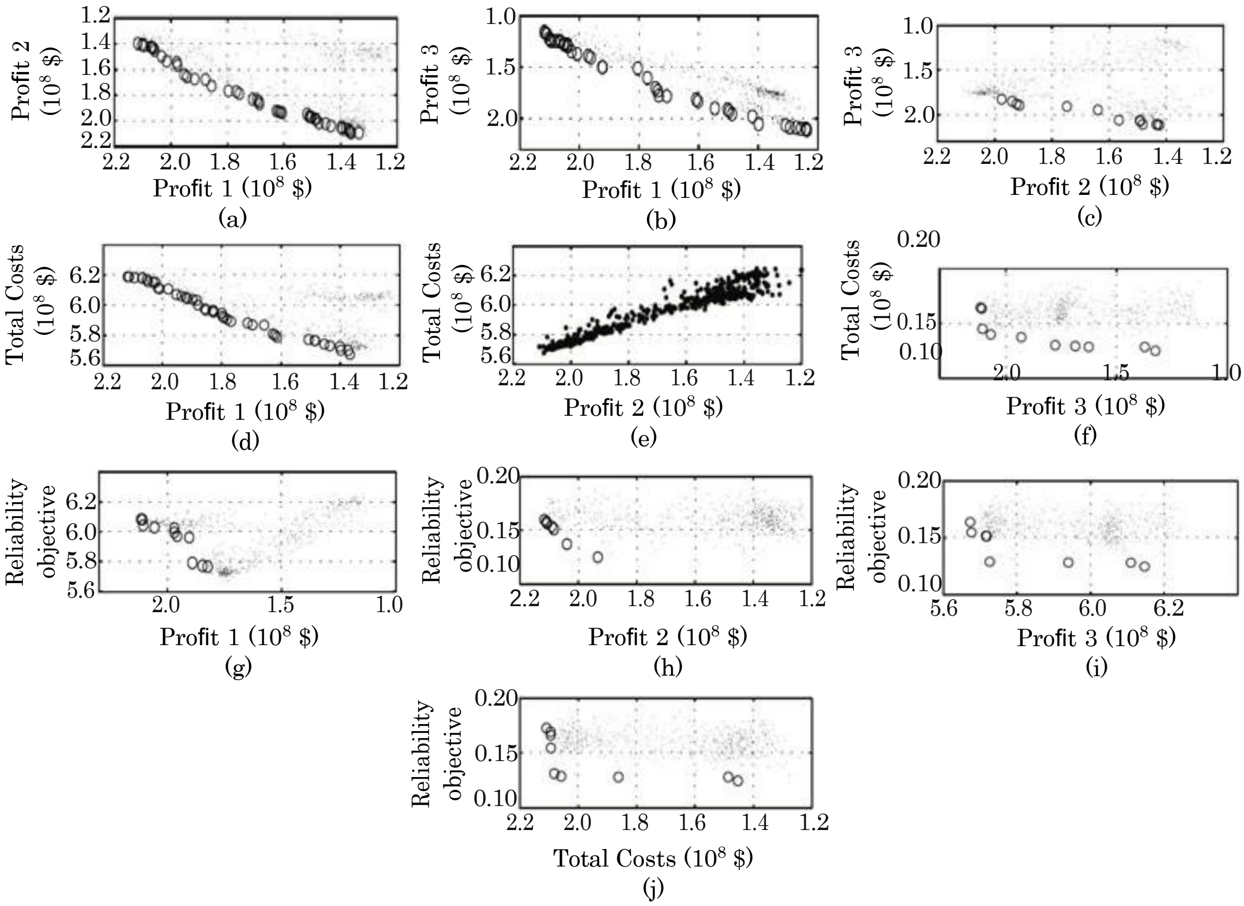


Figure 2.22: *Approximate Pareto fronts obtained by the GSOMP algorithm in the context of a 32-unit IEEE-RTS case study. The graphs exhibit trade-offs between profits for producers 1, 2 and 3, the total generation costs (consisting of fuel cost, start-up costs, and the maintenance cost) for all the producers and reliable energy provision (levelisation of the reserve margin). For all the graphs, except (e), the smaller lighter points represent the approximate Pareto front considering all the objectives. The smaller, darker points represent the approximate Pareto fronts considering only two objectives [218].*

the B&B method), in which the one objective is optimised whilst the other objective is added as a minimum constraint satisfaction requirement (2.59). This procedure produced only one “optimal” solution[15].

2.8 Chapter Summary

The aim in this chapter was to provide the reader with an introduction to the energy industry in general and to GMS problem formulations in particular. This was followed by a more in-depth review of the large variety of GMS model formulations (constraints and objectives) and associated solution techniques employed in the literature. A brief description was also provided of some preliminary work on simple MO modelling approaches and analyses applied in the context of the GMS problem. Finally, recent work in the energy sector was highlighted which involves application of MOO techniques and the relatively few genuine MOO modelling approaches found for GMS problems.

[15]A Pareto-front may, however, be traced out by this goal programming approach by iteratively changing the constraint lower bounds (*i.e.* producing a series of points), categorised as an *a priori* MOO method.

Table 2.9: *Summary of important literature reviewed on the GMS formulation of objectives and solution techniques, adapted from [82, 90]. Sorted according to reference publication years. The number of times the reference has been cited is according to Google Scholar as on 2016-09-01.*

			System		Criteria														Approach			
					Reliability					Economic								C				
					Det.			Sto.		Fuel cost												
Ref.	Year	Cited	Reg.	Der.	LR	MMR	LOLP	EUE	LER	Rev.	Lin.	P-lin.	Qua.	Sta.	Mai.	Ope.	Fix.	Conv.	SO	MO	PMO	Technique
[216]	1983	122	../				../				../								../			DP
[149]	1991	87				../							../						../			Implicit enumeration method
[167]	1991	193	../								../				../				../			SA
[10]	1991	37	../					../			../				../				../			Benders decomompoistion
[126]	1995	44	../					../			../							../		../		Adapted B&B
[47]	1997	20	../		../														../			GA
[29]	2000	175	../								../				../				../			Hybrid GA/SA, GA/TS
[91]	1998	16	../												../	../		../	../			CP
[135]	1998	30	../								../				../				../			Benders decomompoistion
[147]	1999	71	../		../						../			../	../	../						B&B with goal programming
[67]	2000	85	../		../								../		../	../			../			TS
[56]	2002	48	../								../				../				../			Multipopulation cultural algorithm
[69]	2003	44	../										../		../				../			GA with fuzzy logic
[146]	2004	64	../				../		../										../			GA/SA hybrid with fuzzy logic
[48]	1999	62	../		../														../			GA with fuzzy logic
[42]	2005	121		../	../					../	../			../	../				../			MILP (GAMS/CPLEX), ACO, SA
[119]	2005	34		../						../			../		../				../			Game theory
[128]	2006	50	../								../				../			../		../		GA/SA hybrid
[46]	2007	93	../		../														../			GA/SA hybrid
[87]	2007	30	../		../														../			ACO
[145]	2007	69	../				../		../										../			GA/SA hybrid
[32]	2008	83	../								../			../					../			Benders decomompoistion
[200]	2008	62	../				../												../			GA
[83]	2011	25	../												../				../			ACO
[103]	2011	8		../			../											../	../			SA
[165]	2011	49	../								../											SA
[66]	2012	27	../		../								../	../	../				../			PSO
[161]	2012	25	../						../										../			GA with extermal optimisation (EO)
[169]	2013	27	../		../														../			SA
[185]	2013	22	../						../										../			PSO/GA hybrid
[82]	2014	9	../								../			../	../		../		../			MILP (GAMS/CPLEX), ACO, SA
[38]	2014	1		../	../					../			../	../	../						../	NSGA-II
[218]	2014	2		../	../					../			../	../	../						../	GSMOP

A summary is provided in Table 2.9 of some of the major GMS literature, categorised into different categories. The table is adapted from similar tables in [82, 90]. The table is, however, not all-encompassing (not all the papers discussed in this literature review chapter appear in the table), but rather aims to represent the most important papers on different GMS modelling and solution techniques implemented in the literature. Each reference (Ref.) in Table 2.9 is classified according to:

- The power system it targets, regulated (Reg.) and/or deregulated (Der.).
- The criteria incorporated in objective function(s), namely reliability-based (Reliability), economic-based (Economic) and/or convenience-based (C)
  - The nature of reliability objectives: either deterministic (Det.) or stochastic (Sto.). These objectives usually take into account expected energy demand and forced outages (breakdowns) of power generating units models. Deterministic reliability measures include those based on levelising reserves (LR) or minimising the maximum reserve (MMR). Stochastic risk (as opposed to reliability) measures include the minimisation of total LOLP or EUE, or seek to levelise the equivalent/effective reserves (LER) per period.
  - Economic objectives include revenue (Rev.), fuel costs (Fuel costs), start-up costs (Sta.), maintenance costs (Mai.), other operational costs (Ope.), and other fixed costs (Fix.). Fuel costs, are modelled as linear (Lin.), or quadratic (Qua.) functions.
- The solution approaches employed, either categorised as SO[16], simple multi-objective (MO), or Pareto-based multi-objective (PMO).
- The specific solution technique(s) implemented.

The context of the work contained in the remainder of this dissertation is indicated by means of shading in the table.

[16]If more than one objective is involved, then the objectives were optimised separately, weighted together in some way, or the non-dominant criteria were included as constraints.

CHAPTER 3

GMS modelling approach

Contents

As previously suggested a MO modelling approach is required in the context of GMS because of the inherent trade-offs between the various conflicting scheduling objectives. The two objectives proposed in this dissertation for inclusion in a GMS formulation are two dominant objectives commonly found in the literature on GMS (albeit usually separately), namely levelling reserve margins and minimising production cost.

A maintenance schedule is usually defined as follows [32]. Suppose there are n generating units in a power system, indexed by the set $\mathcal{I} = \{1, \ldots, n\}$, and m decision time periods over the planning horizon, indexed by the set $\mathcal{J} = \{1, \ldots, m\}$. Define the binary decision variable X_{ij} to take the value 1 if maintenance of generating unit $i \in \mathcal{I}$ commences during time period $j \in \mathcal{J}$, or zero otherwise. Also define the binary auxiliary variable Y_{ij} to take the value 1 if generating unit $i \in \mathcal{I}$ is in maintenance during time period $j \in \mathcal{J}$, or zero otherwise. Then a maintenance schedule is an assignment of zeros and ones to the $n \times m$ matrix $\boldsymbol{Y} = [Y_{ij}]$ of auxiliary decision variables satisfying a variety of constraints.

3.1 GMS model constraints

It is generally assumed that maintenance of each generating unit is allowed only once[1] during the planning horizon [59]. Let e_i and f_i denote the earliest and latest time periods, respectively, during which maintenance of generating unit $i \in \mathcal{I}$ may start. Then the maintenance window constraints may be formulated as

$$\sum_{j=e_i}^{R_i} X_{ij} = 1, \quad i \in \mathcal{I}. \tag{3.1}$$

[1]If units are to be subjected to more than one maintenance outage during the planning horizon, dummy units may be added to the problem formulation so that this assumption is satisfied [168, p. 149].

In order to ensure that maintenance of each generating unit lasts for a prespecified duration, the maintenance duration constraints

$$\sum_{j=e_i}^{R_i+d_i-1} Y_{ij} = d_i, \quad i \in I \tag{3.2}$$

are included, where d_i denotes the maintenance duration of generating unit $i \in I$. The constraints

$$\begin{aligned} Y_{ij} - Y_{i,j-1} \leq X_{ij}, \quad & i \in I, j \in J \setminus \{1\}, \\ Y_{i1} \leq X_{i1}, \quad & i \in I \end{aligned} \tag{3.3}$$

furthermore ensure that maintenance of each generating unit occurs over consecutive (uninterrupted) time periods.

Let I_i denote the installed power generating capacity of unit $i \in I$, let D_j denote the system peak load demand during time period $j \in J$ and define

$$r_j = \sum_{i \in I} I_i(1 - Y_{ij}) - D_j$$

as the net reserve margin during time period $j \in J$. Then the demand constraint

$$r_j \geq D_j S, \quad j \in J, \tag{3.4}$$

ensures that the available power generating capacity is at least as large as the expected load demand D_j together with a pre-specified safety margin S.

Let μ^l_{cij} denote the required maintenance crew for generating unit $i \in I$ when in maintenance during time period $j \in J$ if maintenance were to commence during time period c. Define

$$\mu_{cij} = \begin{cases} \mu_i^{j-c+1}, & \text{if } j - c < d_i \\ 0, & \text{otherwise,} \end{cases} \tag{3.5}$$

where μ_i^u denotes the required maintenance crew for unit $i \in I$ in its u-th period of maintenance. The maintenance crew constraint set may then be formulated as

$$\sum_{i \in I} \sum_{c=1}^{j} \mu_{cij} X_{ic} \leq M_j, \quad j \in J, \tag{3.6}$$

where M_j denotes the manpower available to perform maintenance operations during time period $j \in J$. If a generating unit's maintenance crew requirements remains the same throughout its maintenance time period, then the manpower constraint may be expressed more simply as

$$\sum_{i \in I} \mu_i Y_{ij} \leq M_j, \quad j \in J,$$

where μ_i denotes the required maintenance crew for generating unit $i \in I$ when in maintenance.

Exclusion constraints are sometimes incorporated into GMS model formulations in order to prevent certain units form being in maintenance simultaneously. Consider the more general exclusion situation where at most some specified number of units, within some subset of units, are allowed to be in maintenance simultaneously. Let $K = \{1, \ldots, K\}$ denote the set of indices of the generating unit exclusion subsets and define $I_k \subseteq I$ as the k-th subset of generating units that form an exclusion set, with $k \in K$. Then the exclusion constraints set may be formulated as

$$\sum_{i \in I_k} Y_{ij} \leq \kappa_k, \quad j \in J, \; k \in K, \tag{3.7}$$

where κ_k denotes the maximum number of units within subset I_k that are allowed to be in simultaneous maintenance during any time period.

Finally, the constraint sets

$$\begin{aligned} X_{ij} &\in \{0,1\}, \quad i \in I, \; j \in J, \\ Y_{ij} &\in \{0,1\}, \quad i \in I, \; j \in J. \end{aligned} \tag{3.8}$$

enforce the binary nature of the decision and auxiliary variables.

3.2 First GMS objective: Levelling reserve margins

Although more elaborate stochastic risk criteria are sometimes used, including formulations based on LOLP, EUE, and effective reserve margins, it is generally known that optimal solutions under one reliability criterion are often also acceptable, though not necessarily optimal, in terms of others [147, 222]. Criteria which level the reserve margins, however, usually obtain less riskier solutions [147]. As mentioned, the most common method of levelling the reserve energy over and above the demand over all time periods may be accomplished by minimising the SSR over time [169]. The reserve margin during a particular time period is the difference between the available capacity and the expected demand. Minimising this SSR results in an even (more "reliable") band of net reserve margins, as illustrated in Figure 3.1(a). The first GMS model objective adopted in this dissertation is therefore to level the net reserve margins $r_1, \ldots, r_m$ by minimising the sum of squared r_j values over time periods $j \in J$, in other words

$$\text{minimising} \sum_{j \in J} r_j^2. \tag{3.9}$$

This first objective function (3.9) together with the constraints (3.1)–(3.8) are exactly the same GMS model formulated and solved in [168, 169]. In this dissertation, however, a second objective involving minimisation of production cost is proposed over and above (3.9), demonstrating how these two objectives may be accommodated together in an MO modelling paradigm.

3.3 Second GMS objective: Minimising production cost

As mentioned, the production planning module of an EFS (see Figure 1.7(b)) typically schedules the planned energy production by making use of available power generating units (including coal, nuclear, gas-turbine, hydro-electric and renewable generating units) with a view to minimise production cost. Power stations associated with cheaper production costs are typically scheduled first for production until demand is met, whilst taking into account production capacities of the various power stations [105].

Power stations generally contain more than one generating unit (*e.g.* a nuclear power plant, may for example, have two generating units) [91]. The production planning module proposed in this dissertation for use in estimating energy production cost is primarily an LP model whose decision variables are the planned energy production levels for the different power stations. The

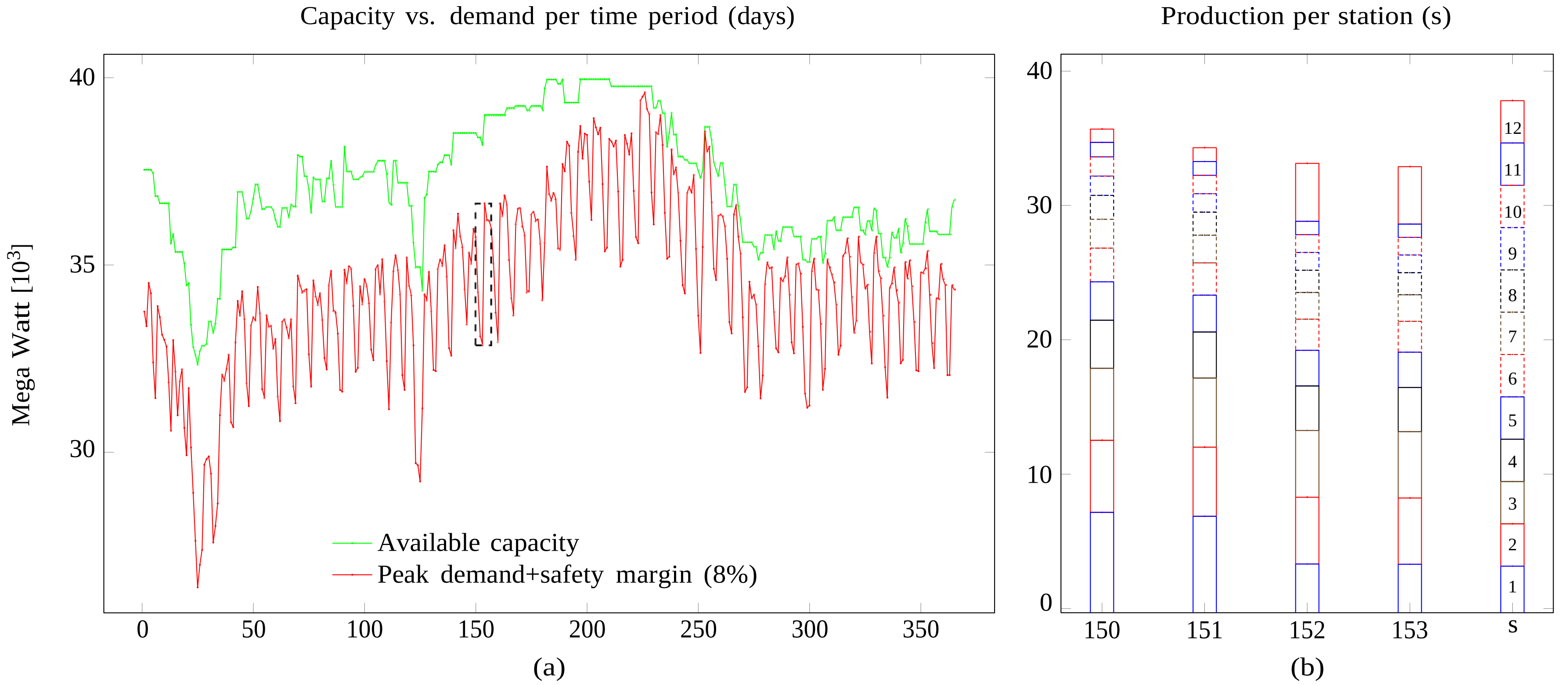


Figure 3.1: *(a) Available capacity versus peak demand (including an adequate safety margin) solved in [168, 169] for a 157-unit Eskom case study. The difference between these two is the reserve level (used in the first objective). (b) An example of the logic of an EFS's production planning module (used in the second model objective in this dissertation) for days 150, 151, 152, and 153 in an annual scheduling horizon. 1-Cheapest power station, 12-Most expensive power station (measured in \$/MWh or R/MWh).*

Table 3.1: *An example of how a maintenance schedule affects a station's PCLF during a time period j.*

Power station	Energy production cost rate ($/MWh)	Generating unit	Maintenance schedule	PCLF
$s = 1$	110	$i = 1$	$Y_{1j} = 1$	33.33%
		$i = 2$	$Y_{2j} = 0$	
		$i = 3$	$Y_{3j} = 0$	
$s = 2$	150	$i = 4$	$Y_{4j} = 0$	50%
		$i = 5$	$Y_{5j} = 1$	
		$i = 6$	$Y_{6j} = 0$	
		$i = 7$	$Y_{7j} = 1$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$s = 12$	2 300	$i = 54$	$Y_{54j} = 0$	0%
		$i = 55$	$Y_{55j} = 0$	

auxiliary variables Y_{ij} described in §3.1 and §3.2 are used to calculate availability LP parameter values in the various power stations in a power system over time as described in some detail in this section. Although a better resolution for the decision variables of the production planning module is possible (*i.e.* considering units separately instead of power stations) this will increase the memory and computing time requirements of the module (which are already high). The benefits of such an increased resolution therefore has to be weighed up against the increased burden of its adoption. It is, however, worth noting that the fuel costs associated with a power station's generating units will typically be the same, thus there might not be much difference between formulating the LP model decision variables in terms of generating units separately and formulating them in terms of entire power stations.

Generally, the production cost and other operational costs (including wear and tear due to frequency of start-ups and shut-downs, the duration and level of operation of the generating facility, *etc.*) associated with a generating unit (or power station) may be represented as a constant rate for the amount of energy produced, *i.e.* a monetary unit per unit energy produced (*e.g.* $/MWh) [215].

To illustrate how a maintenance schedule may affect production cost, an example of a production plan is shown in Figure 3.1(b). If power station 1 (cheapest) has to undergo maintenance during days 152–153, this will decrease its *planned capability loss factor* (PCLF) (as illustrated in Table 3.1), which will, in turn, decrease the station's *energy availability factor* (EAF). This latter value is an input parameter to the production planning module (in the form of capacity constraints in the linear programming model), translating into less energy production being scheduled for the station, although it is the cheapest power station, which will increase the overall production cost. It should therefore be attempted (if possible) to ensure that maintenance of cost-efficient power stations does not occur during high energy demand periods (*e.g.* winter, as seen in Figure 3.1(a)).

The production cost objective adopted in this dissertation is based on the work of Brits [25], who improved the current production planning module of the EFS in 2016 by increasing the LP model decision variable resolution from weekly to daily.

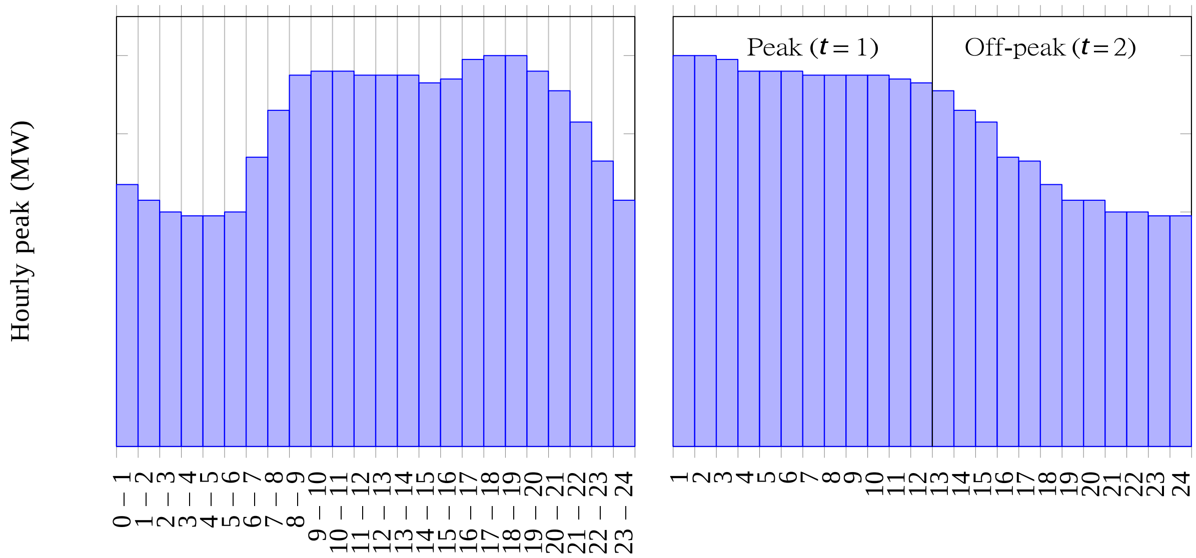
The LP proposed differentiates between peak and off-peak demand periods by sorting the hourly demand for each day in descending order. Each day of every time period within GMS planning horizon is partitioned into two time slices $t \in \mathcal{T} = \{1,2\}$ of twelve hours each. The time slice with the highest demand (denoted by $t = 1$) is treated as the peak demand slice, while the time slice with the lowest demand (denoted by $t = 2$) is treated as the off-peak demand slice, as illustrated in Figure 3.2. The decision variable of the LP model is

$$Z_{st} = \begin{cases} \text{energy production planned for power station } s \\ \text{during time slice } t \text{ (measured in MWh).} \end{cases} \tag{3.10}$$

The set of power stations is denoted by

$$S = \bigcup_{h=1}^{p} S_h, \tag{3.11}$$

where S_h denotes the set of power stations of type h. Hence p different power station types are included in the model. These types may encompass coal-fired, oil-fired, hydroelectric (conventional and pumped storage), nuclear, and gas-turbine power stations. The power stations are arranged into a set $S_{\text{base}} \subseteq S$ of *base load* stations, and a set $S_{\text{peak}} \subseteq S$ of *peak demand* power stations. It is assumed that base load stations are designed to operate continuously at a steady load and are generally only shut down for planned maintenance, in the case of emergency maintenance, or when there is very low demand. Peak demand stations, on the other hand, are responsible for generating the additional demand placed on the system during peak demand periods over and above the base demand. Each subset S_h may contain base load and/or peak demand stations. One of these subsets, however, contains a single, virtual power station that represents unmet load. A very large energy production cost rate (measured in \$/MWh) is assigned to this virtual station in order to penalise and hence discourage unmet demand. The energy production cost rate for power station s is denoted by C_s. The cost rate of the energy produced by the generating units of thermal and nuclear power stations vary depending on the stations' fuel source costs and efficiencies (typically measured as their heat rates). Hydroelectric power plants have zero fuel costs, because the resource is freely available, and their start-up costs are also zero [32].



(a) *Load curve* (b) *Load curve sorted*

Figure 3.2: *An example of how an EFS's production planning module may use each day's expected hourly electricity demand data (a) and sort it into two time slices $t \in \mathcal{T} = \{1, 2\}$ of 12 hours each (b).*

The available generation capacity of power station $s \in S$ during time slice $t \in \mathcal{T}$ is denoted by

$$A_{st} = 12 P^{\mathcal{E}}_s, \quad t \in \mathcal{T}, \quad s \in S, \tag{3.12}$$

where I^S is the installed generation capacity of power station $s \in S$ (the sum of the installed generation capacities of its generating units) and E_s is the EAF of power station $s \in S$ across both time slices $t \in T$ in the day. The factor is calculated as

$$E_s = 1 - (P_s + U_s + O_s), \quad s \in S, \tag{3.13}$$

where P_s denotes an unavailability proportion due to PCLFs at power station s during the day which represent power generation losses specifically planned by the management of the power system for maintenance purposes and other shutdowns. Similarly, U_s denotes an unavailability proportion due to *unplanned capability loss factors* (UCLFs) which represent breakdowns (often as a result of a lack of planned maintenance), while O_s denotes an unavailability proportion due to *other capability loss factors* (OCLFs) as a result of extraordinary events outside the control of the management of the power system, such as employee strikes or damage to transmission cables [104, 142]. The auxiliary decision variables Y_{ij} of the GMS model described above in the beginning of this chapter are used to calculate the unavailability proportion due to PCLFs per station $s \in S$. This is how the GMS model interacts with the production planning module as was illustrated in Table 3.1. If, for example, one out of the four units in a power station are scheduled for maintenance, then the PCLF factor value P_s is 0.25 (across both time slices $t \in T$ in the day) over the entire maintenance duration of the unit in question, provided that no other unit within the station also enters maintenance.

Let L_t be the load demand forecast for time slice $t \in T$. Then the objective in the energy production planning module is to

$$\text{minimise} \sum_{t \in T} \sum_{s \in S} C_s Z_{st} \tag{3.14a}$$

subject to

$$Z_{st} \leq A_{st}, \quad s \in S, \quad t \in T, \tag{3.14b}$$

$$\sum_{s \in S} Z_{st} = L_t, \quad t \in T, \tag{3.14c}$$

$$Z_{st} \geq 0, \quad s \in S, \quad t \in T. \tag{3.14d}$$

The objective function in (3.14a) represents the total production cost. Constraint set (3.14b) ensures that each power station's planned energy production does not exceed its available capacity during any time slice. The balance between planned energy production and forecast demand during each time slice is ensured by constraint set (3.14c), while constraint set (3.14d) is a sign restriction constraint on the energy production decision variables.

Depending on the power system's generation mix, a number of additional constraints may be included. Generating units are usually designed to operate between specified minimum and maximum power levels (measured in MW) [32] and base load power stations (especially coal fired and nuclear stations) typically have maximum and minimum daily production requirements. The two constraint sets

$$\sum_{t \in T} Z_{st} \leq {}_s, \quad s \in S, \tag{3.14e}$$

$$\sum_{t \in T} Z_{st} \geq {}_s, \quad s \in S \tag{3.14f}$$

may therefore be added to the model formulation.

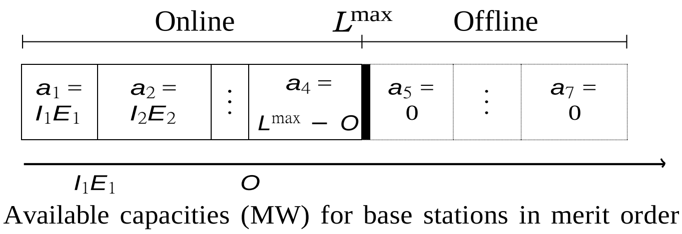


Figure 3.3: *Illustrative example of the simple UC algorithm's logic. The maximum daily peak demand (L^{max}) is used to determine which base stations are to be online and which are to be taken offline. The algorithm is executed prior to solving the ED LP problem.*

The maximum and minimum daily loads for power station s are calculated as

$$f_s^{max} = \sum_{t\in T} A_{st} q_s^{max}, \quad s \in S, \tag{3.15a}$$

$$f_s^{min} = \sum_{t\in T} A_{st} q_s^{min}, \quad s \in S, \tag{3.15b}$$

respectively where q_s^{max} and q_s^{min} represent the pre-specified maximum and minimum *energy utilisation factors* (EUFs) for stations $s \in S$, respectively. If there are no maximum and minimum daily load restrictions on a station (gas turbine stations, for example, have more freedom of production), their EUF values may be set at the extremes (0 and 100%).

Before solving the LP model (3.14a)–(3.14f), a simple algorithm is employed to find a solution to the UC problem[2]. This algorithm updates the maximum and minimum daily loads in (3.15a)–(3.15b) by performing the substitutions

$$f_s^{max} \leftarrow f_s^{max} \frac{a_s}{I_s E_s}, \quad s \in S_{base}, \tag{3.16a}$$

$$f_s^{min} \leftarrow f_s^{min} \frac{a_s}{I_s E_s}, \quad s \in S_{base}, \tag{3.16b}$$

where $a_s \in [0, I_s E_s]$ is the online capacity committed for station $s \in S_{base}$. The algorithm calculates the values a_s based on the day's peak hourly demand L^{max} and the available capacities ($I_s E_s$) for the base stations sorted in increasing order of cost coefficients C_s, as demonstrated in Figure 3.3. Alternatively, if the maximum daily peak demand L^{max} is larger than the sum of all the base power station's available capacities, then all base load stations are committed (*i.e.* $a_s = I_s E_s$). The above station commitment algorithm satisfies the constraint that nuclear generating units are always connected, except when they are in a state of maintenance [32], since nuclear plants usually appear first in the ordered list of base power stations according to the increasing order of cost coefficients C_s.

Certain power stations (base load and/or peak demand) may have an exact daily production requirement (*e.g.* hydro power plants operate according to a set amount of stored water energy). If this is the case, define the subset $S_{exact} \subseteq S$ of exact energy required power stations. Then the constraint

$$\sum_{t\in T} Z_{st} = f_s^{req}, \quad s \in S_{exact} \tag{3.17}$$

may be added to the LP formulation. Hydro stations are typically installed with around 50% more capacity than energy (water) available [9]. If, however, considerable maintenance is performed on the generating units, so much so that the capacity available from generating units

[2]In the proposed model it is actually a station commitment problem.

cannot match the energy available in the dam's water, then the constraint above may be relaxed according to the available capacity.

At pumped storage scheme stations, water is stored in an upper reservoir and, after running through a set of turbines, is discharged into a lower reservoir from where it is later pumped back to the upper reservoir. Pumping usually takes place during off-peak demand periods ($t = 2$) in order to have maximum generation capability during peak demand periods ($t = 1$). The pumps use energy while in operation and this energy requirement must therefore be included in the LP formulation as additional constraints. Let $S_{ps} \in S$ denote the set of pumped storage scheme stations and let $S_{psp} \subseteq S$ be the set of corresponding pumps. Then the load balance constraint (3.14c) becomes

$$\sum_{s \in S/=S_{psp}} Z_{st} - \sum_{s \in S_{psp}} Z_{st} = L_t, \quad t \in T \tag{3.18}$$

in the case where pumped storage scheme stations are used. In addition, the two constraint sets

$$\sum_{t \in T} Z_{at} \leq B_a \sum_{t \in T} Z_{bt}, \quad a \in S_{ps}, \quad b \in S_{psp}, \tag{3.19}$$

$$Z_{s1} \leq 0, \qquad s \in S_{psp} \tag{3.20}$$

should be added to the LP model formulation, where B_a is the efficiency of station $a \in S_{ps}$ (the pumped storage scheme stations). The production cost rate C_s of the pumps are zero, which means that they have no effect on the objective function in (3.14a). Constraint set (3.19), however, ensures that the energy required by the pumps are taken into account while (3.20) ensures that the pumps are not required during peak demand time slices ($t = 1$).

The units of measurement for all the parameters and variables in the production planning module are provided in Table 3.2.

Table 3.2: *Units of measurement for the parameters in the production planning module*

Symbols	Unit of measurement	Symbols	Unit of measurement
Z_{st}	MWh	B_a	%
C_s	\$/MWh	E_s	%
A_{st}	MWh	P_s	%
L_t	MWh	U_s	%
f_s^{max}	MWh	O_s	%
f_s^{min}	MWh	q_s^{max}	%
f_s^{req}	MWh	q_s^{min}	%
I_s	MW		

The LP model is solved for each day over the GMS planning horizon to compute the planned energy production at each power station so that the daily production cost is at a minimum. The GMS cost function is therefore the sum of these daily production costs over the entire planning horizon.

3.4 Incorporating GMS into an energy flow simulator

Figure 3.4 illustrates how the architecture of an EFS may be updated in order to incorporate the newly proposed GMS model (Figure 3.4(e)) and how this model may interact with the other components of the EFS [132]. The consumption module (Figure 3.4(a)) provides the demand per hour required for the production planning module as well as the peak demand D_j per time period $j \in J$. From these peak demand values D_j and the maintenance schedule values Y_{ij}, the SSR values may easily be determined. In conjunction with the same maintenance schedule Y_{ij}, the PCLFs for the power stations may be used by the production planning module (Figure 3.4(b)) to determine the estimated production cost associated with the maintenance schedule. These two objectives may be incorporated in an MO modelling paradigm to optimise MS as explained in the following chapter.

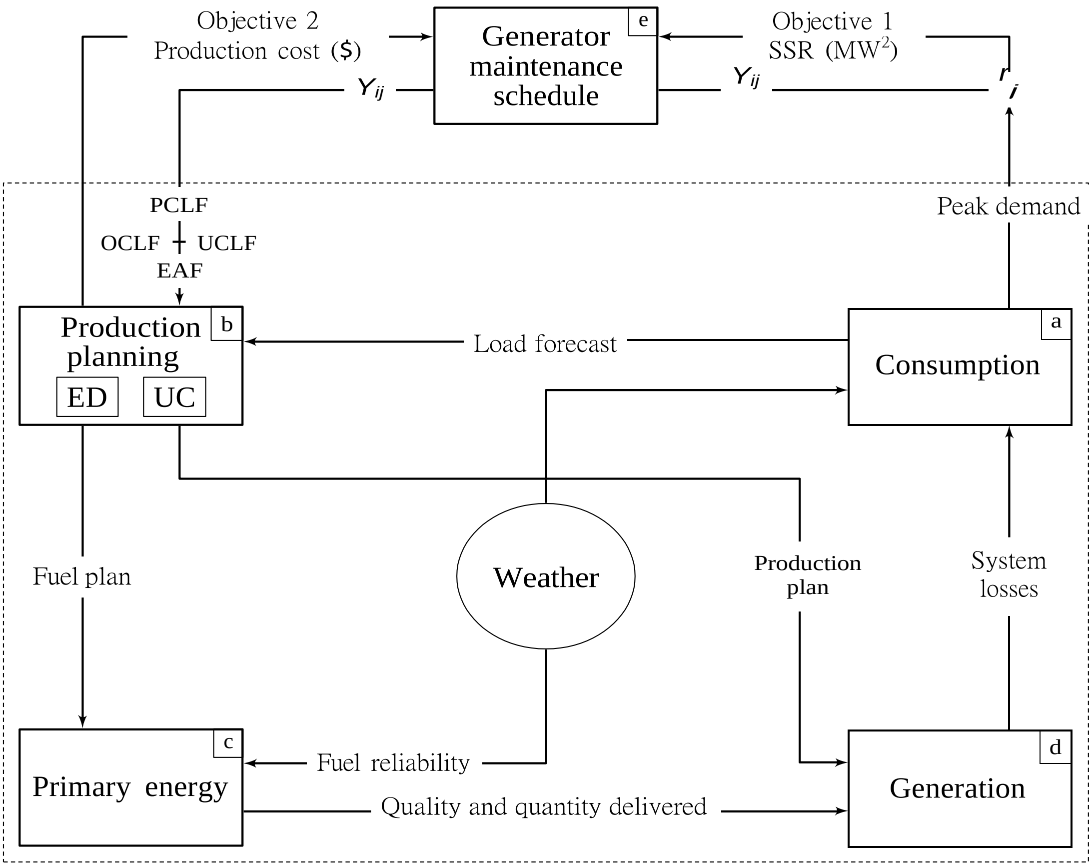


Figure 3.4: *High-level representation of an EFS (dashed area) and how it is anticipated that the proposed GMS approach may form part of it, specifically in respect of the production planning module which follows a simple UC logic and its ED problem is formulated as a linear program.*

3.5 Chapter summary

In this chapter, a novel bi-objective GMS model was proposed. First the GMS constraints most commonly adopted in the literature were formulated in §3.1. The first GMS objective function, involving the minimisation of the SSR was formulated in §3.2. A more elaborate explanation followed in §3.3 of how the second GMS objective function, involving minimisation of production cost, may be estimated. How the proposed GMS model may form part of a typical EFS architecture was finally described in §3.4.

CHAPTER 4

Solution methodology

Contents

The general underlying principles of MOO are presented in the first section of this chapter. This is followed by a motivation for the choice of algorithmic solution approach adopted to solve the GMS model proposed in Chapter 3 and the different variants of this algorithm available for MOO. A description of the constraint handling technique adopted is further provided. It is finally described how the algorithm of choice was implemented in order to produce the numerical results reported later in this dissertation.

4.1 Basic notions in multi-objective optimisation

MOO is a subdiscipline of *multiple criteria decision making* which is concerned with solving optimisation problems involving the pursuit of more than one objective simultaneously. In MOO, the goal is to simultaneously maximise or minimise o objective functions, $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_o(\boldsymbol{x})$, which are functions of a vector of decision variables $\boldsymbol{x} = [x_1, x_2, \ldots, x_a]$. Suppose, without loss of generality, that all the objective functions are to be minimised[1]. A MOO problem may then, in general, be expressed as

$$\text{minimise } \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_o(\boldsymbol{x})] \tag{4.1a}$$

subject to the constraints

$$g_i(\boldsymbol{x}) \leq G_i, \quad i = 1, \ldots, u, \tag{4.1b}$$

$$h_j(\boldsymbol{x}) = H_j, \quad j = 1, \ldots, v, \tag{4.1c}$$

$$\boldsymbol{x} \in \mathbb{R}^a, \tag{4.1d}$$

where $g_1(\boldsymbol{x}), \ldots, g_u(\boldsymbol{x})$ are the so-called inequality constraint functions and $h_1(\boldsymbol{x}), \ldots, h_v(\boldsymbol{x})$ are the equality constraint functions. Furthermore, $G_1, \ldots, G_u$ and $H_1, \ldots, H_v$ are assumed to be

[1]An objective function to be maximised can instead be minimised after multiplication by -1.

(strictly positive) limiting values for the constraint functions. The set of all feasible decision vectors form the so-called *decision space* of the problem, denoted here by X.

MOO techniques are employed in cases where the objective functions are conflicting, in which case a set of trade-off solutions is sought, which leads naturally to the notion of Pareto optimality as a result of the fact that there is typically no single solution $\boldsymbol{x}^*$ that minimises all the objective functions simultaneously. A feasible decision vector $\boldsymbol{x} \in X$ *dominates* a decision vector $\boldsymbol{y} \in X$, denoted by $\boldsymbol{x} \prec \boldsymbol{y}$, if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ for all $i \in \{1, \ldots, o\}$ and there exists at least one $i^* \in \{1, \ldots, o\}$ such that $f_{i^*}(\boldsymbol{x}) < f_{i^*}(\boldsymbol{y})$ [179].

A solution is said to be *globally non-dominated*, or *Pareto optimal*, if no other feasible solution dominates it. The solutions in the Pareto set P_S produce a set of objective function vectors, known as the *Pareto front* P_f, whose corresponding decision vectors are elements of P_S, that is $P_f = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in P_S\}$.

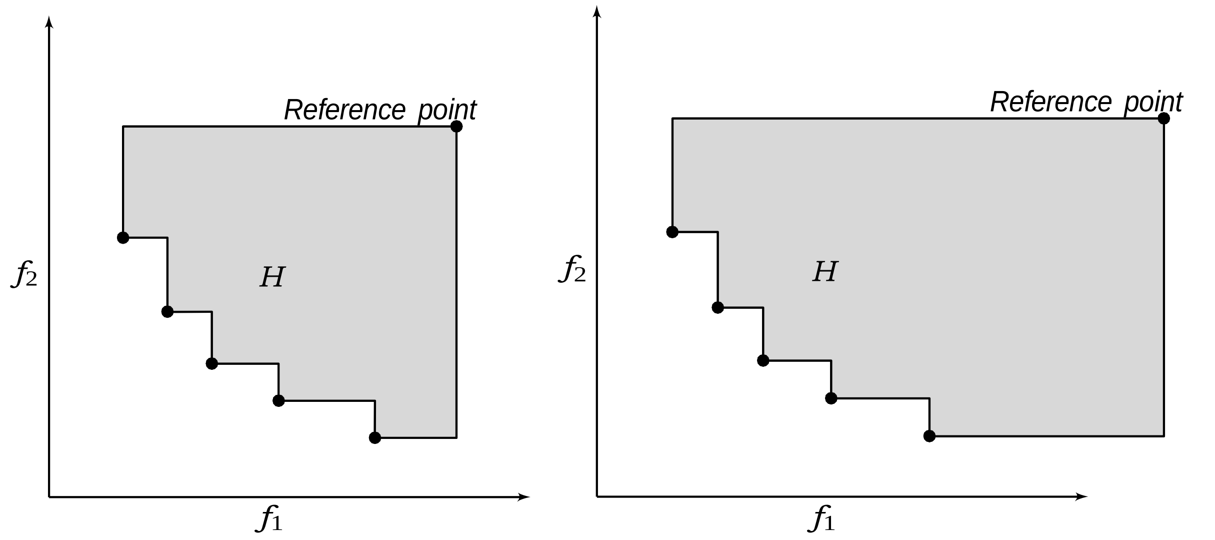
4.1.1 The hypervolume indicator

It is naturally difficult to measure and compare the quality of different Pareto front approximations. A popular measure aimed at evaluating the relative performance of one non-dominated front in respect of another is the well-known *hypervolume indicator* [203, 220]. This indicator, also known as the *hyperarea metric*, *S-metric*, or *Lebesgue measure* and denoted here by H, measures the hypervolume of the objective function space that is dominated by solutions in the non-dominated front with respect to a preselected vector in objective space, known as the *reference point*, which is dominated by all solutions in the front under consideration [220]. A large hypervolume is desirable and it attains its maximum value if and only if the non-dominated front is, in fact, the true Pareto front [203]. Hypervolume is a preferred measure of the quality of an approximated Pareto front since it attempts to capture the closeness of the approximate solutions with respect to the Pareto optimal set, as well as, to a certain extent, the spread of solutions within a non-dominated set across the objective space [203]. In the bi-objective function space, the hypervolume represents the area enclosed between the approximate Pareto front and the reference point, as illustrated in Figure 4.1. Use of the notion of hypervolume as a measure of the quality of a non-dominated front in objective space has some disadvantages associated with it, three of the main being that it is sensitive to the relative scaling of the objectives and to the presence or absence of extremal points in a front, and that it is also sensitive to the choice of the reference point [203].

In order to illstrate the challenge of choosing a good reference point so as not to bias the indicator, consider the bi-objective optimisation problem example associated with Figure 4.1, and assume that the entries in objective space have been normalised. The reference point shown in Figure 4.1(b) is a poor choice, as it is (visually) apparent that the hypervolume is significantly more sensitive to changes in values of the first objective function (f_1) than those of the second objective function (f_2). The reference point in Figure 4.1(a) would be a better, less biased choice.

4.1.2 Composite functions

A very popular approach towards converting an MO problem to a scalar optimisation problem is to minimise a linear combination of the different objectives in (4.1a) [50] into the single objective



(a) *Good reference point for calculating H* (b) *Poor reference point for calculating H*

Figure 4.1: *Hypervolume H of a non-dominated front with five points indicated by the surface area of the shaded region. Both the objective functions f_1 and f_2 are to be minimised.*

function

$$F(\boldsymbol{x}) = \sum_{i=1}^{o} w_i f_i(\boldsymbol{x}). \tag{4.2}$$

Here the single composite objective F is a weighted combination of o objectives, using the weightings $w_1, \ldots, w_o$ as indicators of the relative contributions of the values of $f_1, f_2, \ldots, f_o$ to the value of F. Certain decision vectors corresponding to points on the Pareto front of (4.1a)–(4.1d) may be uncovered by a judicious choice of the weights in (4.2) when the objective functions in (4.1a) are replaced with the single function (4.2). It is, however, unclear what these weights should be in advance.

Another problem related to scalarising an MOO problem is that even for convex Pareto fronts, an evenly distributed set of weights may fail to produce an even distribution of points along the Pareto front [50]. In fact, one of the principal advantages of MOO is that no preference is given to the objectives, rather seeking a non-dominated front from which the decision maker may choose a desired solution. Perhaps more importantly, parts of the Pareto front are inaccessible by any choice of weights $w_1, w_2, \ldots, w_o$ for non-convex Pareto fronts. A proof of this claim is provided in [50], showing that for problems where the Pareto front is non-convex, an algorithm minimising weighted sums of the objectives will be unable to converge to some regions of the Pareto front.

Despite the drawbacks associated with composite functions such as (4.2), MOO techniques utilising composite functions have been shown to perform well in some specific contexts, especially in respect of many-objective problems (those with a large number of objectives) [109]. It is further observed that many MOO algorithms utilising Pareto ranking (such as the NSGA-II) perform well when there are not too many objectives [159] — these algorithms are typically widely established and well developed for problems with two or three objectives [109]. It is noted in [109] that it is more effective to perform many SO optimisations than adopt a Pareto-based (or dominance-based) ranking optimiser in the context of many-objective problems. Lately, however, researchers have attempted to improve MOO algorithms based on rankings whose performance scales to many-objective problems. Deb and Jain [51], for example, developed an improved NSGA-III algorithm specifically for many-objective problems in 2014.

4.1.3 Desirable properties of multi-objective optimisation algorithms

In MOO, important properties of a good algorithm are [151]:

Searching precision. The algorithm must find the Pareto optimal solutions, that are global optima in MOO. When this is difficult to achieve due to problem complexity, it must find solutions near to the Pareto optimal solutions in objective function space.

Searching time. It must efficiently find the non-dominated set.

Uniform distribution over the non-dominated set. The solutions found must be widely spread, or uniformly distributed, over the entire non-dominated set instead of clusters of points in objective function space.

Information about Pareto front. The algorithm must provide as much information about the Pareto front as possible.

MOO has become an important research topic of late for scientists and researchers due to the MO nature of many real-world problems. Researchers have developed many MOO procedures [183], but it is difficult to compare results returned by one MO method to those returned by another, as there is typically not a unique optimum in MOO as in SO optimisation.

4.2 Motivation for solution methodology chosen

As eluded to in §2.6.2, algorithms perform on average equally well over all problem instances, but some algorithms do perform better in the context of restricted classes of optimisation problems. As mentioned, mathematical programming techniques, dynamic programming techniques, and metaheuristics are typically used to solve GMS problems.

A metaheuristic approach was selected to solve the GMS model proposed in Chapter 3 since mathematical programming and dynamic programming techniques struggle to find solutions to realistically sized model instances within acceptable time frames. Mathematical programming techniques are also more suitable when the objectives and constraints are linear (and not too complex). The ED problem considered in this dissertation is furthermore formulated as an LP problem, and a metaheuristic may easily be equipped with an LP solver. Most importantly, metaheuristics have recently become very popular in MOO due to their ability to find multiple non-dominated solutions within a single run. They also function without derivatives, tend to converge rapidly to Pareto optimal solutions with appropriate parameter tuning, and are able to handle both continuous and combinatorial optimisation problems with ease. Metaheuristics are also less susceptible to shape or continuity complexities of the objective functions [183].

Of the metaheuristics that have been applied to GMS problem instances in the literature, SA and GAs are the most widely used [128, 147]. The method of SA is employed in this dissertation as it has been adopted successfully a number of times in the GMS literature [46, 165, 167, 169]. The SA hybrid developed by Schlünz and van Vuuren [169], for example, has outperformed a GA and GA/SA hybrid, and has matched the best known solution found via an ACO in the context of a 21-unit GMS benchmark system [169]. More specifically, an innovative MO SA algorithm, developed by Smith *et al.* [179], is adopted in this dissertation and is described in some detail in the following section.

Like most other metaheuristics, SA is able to accommodate objective functions with any arbitrary degrees of nonlinearity, discontinuity, and stochasticity, and accommodate quite arbitrary boundary conditions and constraints imposed on these cost functions [111]. SA may furthermore be implemented quite easily with the degree of coding quite minimal relative to other nonlinear optimization algorithms and metaheuristics [111]. SA is also the only metaheuristic that is statistically guaranteed to find a globally optimal solution for SO optimisation problems [111] when the annealing schedule cools sufficiently slowly (*i.e.* run for a long time) [96, 101]. This theoretical result is, however, not of much practical benefit, since the time required to ensure a significant probability of reaching a globally optimal solution might not be less (or might even be more) than that required to perform a complete search of the solution space. SA nevertheless often yields excellent results when executed in conjunction with a faster cooling schedule [179]. Some negative features of the SA algorithm include that it may be quite time-consuming to find a near-optimal solution, that it may be difficult to fine-tune to specific problems and that if the cooling schedule is not selected appropriately the algorithm may not perform as expected [111].

4.3 The method of simulated annealing

SA is a metaheuristic first proposed by Kirkpatrick *et al.* [120] in 1983, and may be thought of as the computational analogue of slowly cooling a metal so that it adopts a low-energy, crystalline state. In metallurgy, annealing is a heat treatment employed to optimise the physical and sometimes chemical properties of a material. After initial heating, the material is then cooled slowly in stages, keeping the temperature constant during each stage (called an *epoch*) for a sufficient duration. If applied correctly, this strategy should lead to a crystallined solid state, which is stable, and corresponds to a minimum energy state [61, p. 25]. The objective function in a minimisation problem is analogous to the free energy in the physical annealing system, while a feasible solution to the problem corresponds to a certain state of the material in the physical case. The final (possibly globally optimal) solution to the minimisation problem is finally analogous to the physical system being frozen in its ground state.

In its original SO configuration, the aim of the SA algorithm is to minimise a specified objective function $f(\boldsymbol{x})$ of a vector $\boldsymbol{x}$ of decision variables. During each iteration of the algorithm, the current solution $\boldsymbol{x}$ experiences a small, random change in order to obtain a neighbouring solution $\boldsymbol{x}'$. This neighbouring solution's energy value $E(\boldsymbol{x}')$ is usually taken as its objective function value $f(\boldsymbol{x}')$ in most SO applications. The difference in energy between the neighbouring and current solution is simply defined as $\Delta E(\boldsymbol{x}', \boldsymbol{x}) = E(\boldsymbol{x}') - E(\boldsymbol{x})$. The neighbouring solution is accepted as the new current solution with probability

$$P(\boldsymbol{x}') = \min\{1, e^{-\Delta E(\boldsymbol{x}',\boldsymbol{x})/T}\}, \tag{4.3}$$

where T is a control parameter referred to as the *temperature* of the system. An improving solution $\boldsymbol{x}'$ (*i.e.* a solution with a lower energy value than that of $\boldsymbol{x}$) is therefore always accepted as the new current solution. The probability $P(\boldsymbol{x}')$ of accepting a worsening neighbouring solution, however, depends on the magnitude of the energy difference $\Delta E(\boldsymbol{x}', \boldsymbol{x})$ and the temperature T. At higher temperature values the probability $P(\boldsymbol{x}')$ is large, resulting in the SA algorithm being likely to accept more worsening solutions, thus exploring the search space. At lower T values only neighbouring solutions exhibiting small increases $\Delta E(\boldsymbol{x}', \boldsymbol{x})$ in energy are, however, likely to be accepted, leading to limited exploration (*i.e.* exploitation). For this reason the SA algorithm is initiated with a high temperature value which decreases slowly over time so as to allow the search to explore the solution space initially, but then later settle on and converge to a local minimum.

4.3.1 Early multi-objective simulated annealing algorithms

Many *multi-objective simulated annealing* (MOSA) algorithms are based upon a composite function of the form (4.2), in which case

$$E(\boldsymbol{x}) = \sum_{i=1}^{o} w_i f_i(\boldsymbol{x}). \tag{4.4}$$

The energy $E(\boldsymbol{x})$ in SA is therefore a combination of the o objective functions in (4.1a) according to relative weightings $w_1, \ldots, w_o$ in (4.2). Adopting this scheme, it follows that $E(\boldsymbol{x})$ will be minimised. An equivalent alternative is to take the energy function as the sum of the values $\log f_1(\boldsymbol{x}), \ldots, \log f_o(\boldsymbol{x})$ [72]. Other non-linear and stochastic composite energy formulations have also been considered [151, 178, 194].

Serafini [174] in 1994 proposed one of the first MOSA algorithms. Different approaches were considered for the construction of an energy function, including methods equivalent to the weighted sum (and products) method, the difference in whichever objective represents the greatest difference between solutions, the minimum difference between objectives, and composites of these functions. Results obtained in this manner within the context of travelling salesman problem instances were presented, for which the algorithm performs well, but no comparison was performed with respect of results obtained by other optimisation techniques.

Czyżak and Jaszkiewicz [44] in 1998 proposed a *Pareto* SA algorithm, employing several parallel annealing chains, each optimising a composite energy function. Results obtained by the algorithm in respect of a knapsack problem demonstrate superiority of the algorithm to Serafini's MOSA algorithm [174].

In 1999, Ulungu *et al.* [194] proposed a MOSA that is very similar to Serafini's MOSA algorithm [174]. The acceptance criterion is again calculated using a weighted sum of the objectives and an archived set of non-dominated solutions is maintained. Although results obtained by the algorithm in respect of a knapsack problem were presented, no comparison was performed in respect of the results obtained by either of the previously discussed MOSA algorithms.

Suppapitnarm *et al.* [184] proposed a MOSA algorithm in 2000 which, instead of weighting and summing the objectives to produce a composite energy difference for the acceptance criteria, uses a multiplicative function with individual temperatures for each objective. These individual temperatures are adjusted independently by the algorithm. The multiplicative energy function is equivalent to a weighted sum of logarithms of the objectives. This negates the need for *a priori* weighting of the objectives, and may thus be considered to function as a weighted composite sum approach with controlled weightings determined by the algorithm. Results obtained by this algorithm for a range of test and real problems demonstrated that the algorithm performs comparably to other optimisation techniques.

As mentioned by Smith *et al.* [178], one of the most promising MOSA algorithms is the one proposed by Nam and Park [151] in 2000, since the notion of dominance is utilised in that algorithm. If the neighbouring solution dominates the current solution, the neighbouring solution is accepted (*i.e.* representing an improving move). Otherwise, if the neighbouring solution is dominated by the current solution it is accepted based on modifications of the probability given in (4.3). Nam and Park define several schemes for calculating the energy difference controlling acceptance similar to the methods adopted by Serafini [174]. Based on a small empirical study involving bi-objective problem instances, they suggest that the best of these schemes is the average difference in objective values. If there is no superiority between the current and neighbouring solution (*i.e.* one does not dominate the other), the neighbouring solution is still accepted, because this promotes exploration of the search space and facilitates escape from local optima. As the dimensionality increases, however, so does the proportion of all moves which are accepted unconditionally according to this scheme, thus emulating a random walk through the search space when dealing with problems having many objectives.

All of the MOSA approaches mentioned above utilise composite energy functions, with some of these offering variations on the weighted sum, such as the sum of logged objectives. An assurance of convergence can be established for a MO simulated annealer adopting a composite objective function and fixed weights as in (4.2) for convex optimisation problems [50]. The MOSA algorithm described in the next section may, however, be seen as superior in this regard, since it does not utilise any composite energy functions and is applicable to both convex and non-convex MOO problems.

4.3.2 The dominance-based multi-objective SA algorithm

A relatively recent and powerful *dominance-based multi objective simulated annealing* (DMOSA) algorithm proposed by Smith *et al.* (2004) [179] is employed in this dissertation. This DMOSA algorithm [179] assumes as energy function $E(\boldsymbol{x})$ a suitable normalisation of the number[2] of non-dominated solutions uncovered thus far during the search that dominate $\boldsymbol{f}(\boldsymbol{x})$. Let F denote the set of mutually non-dominating solutions found thus far during the search (*i.e.* an archived non-dominated front), define $\tilde{F} = F \cup \{\boldsymbol{x}\} \cup \{\boldsymbol{x}'\}$ and let $\tilde{F}_{\boldsymbol{x}}$ be the elements of $\tilde{F}$ that dominate $\boldsymbol{x}$, that is $\tilde{F}_{\boldsymbol{x}} = \{\boldsymbol{y} \in \tilde{F} \mid \boldsymbol{y} \prec \boldsymbol{x}\}$. Then $\tilde{F}_{\boldsymbol{x}}$ is taken as the energy $E(\boldsymbol{x})$ of $\boldsymbol{x}$ and so

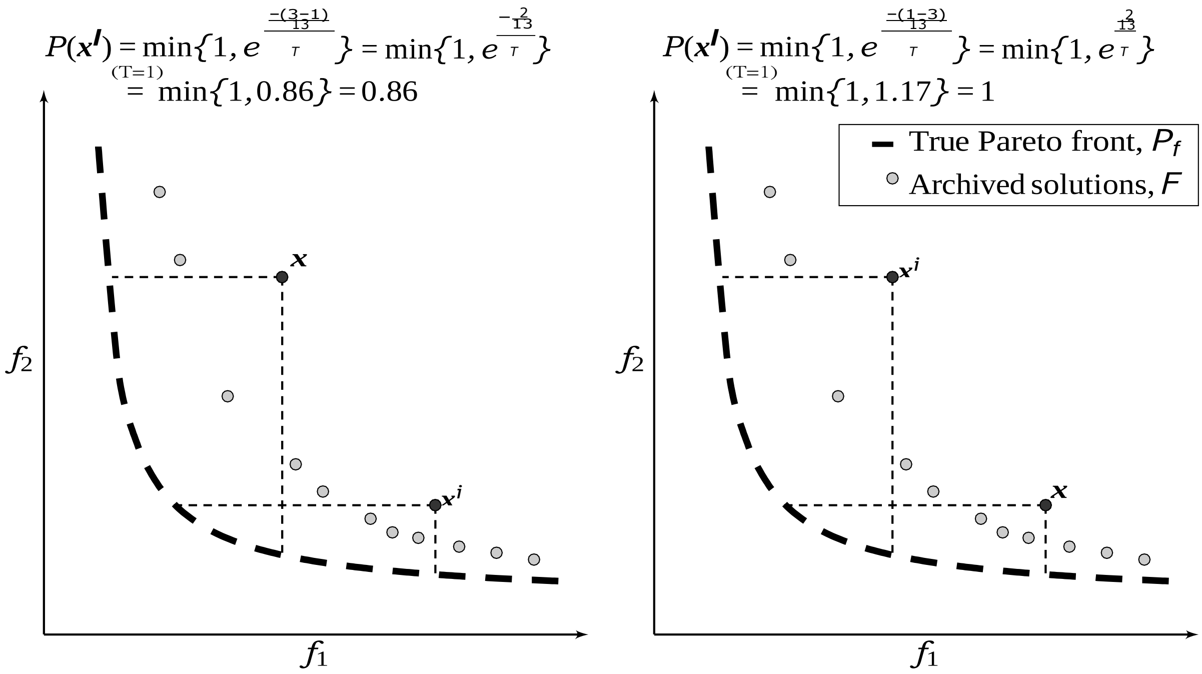
$$\Delta E(\boldsymbol{x}', \boldsymbol{x}) = \frac{|\tilde{F}_{\boldsymbol{x}'}| - |\tilde{F}_{\boldsymbol{x}}|}{|\tilde{F}|}. \tag{4.5}$$

Division by $|\tilde{F}|$ in (4.5) ensures that ΔE is always less than one and provides some level of robustness against fluctuations in the number of solutions in F. An example of the working of the acceptance mechanism employed in the DMOSA algorithm is provided in Figure 4.2.

The probability of accepting a neighbouring solution $\boldsymbol{x}'$ in the DMOSA is the same as that in SO SA, defined in (4.3). Proposals that are dominated by one or more members of the current archive are accepted with the probability in (4.3). Consider, for example, Figure 4.2(a), where the neighbouring solution $\boldsymbol{x}'$ is dominated by more archived solutions than is $\boldsymbol{x}$ and therefore has a higher energy value (*i.e.* it is a worsening move). This neighbouring solution will be accepted based upon the probability in (4.3) which depends on the energy difference between the current and neighbouring solution in (4.5) and the temperature T. Consider Figure 4.2(b) next, where the neighbouring solution $\boldsymbol{x}'$ is dominated by fewer archived solutions than $\boldsymbol{x}$ and therefore has a lower energy value. This neighbouring solution would therefore automatically be accepted as it is considered to embody an improving move. It should be noted that this probability does not depend upon metric information in objective space, *i.e.* there is no weighting of multiple criteria into a single objective and the acceptance probability is unaffected by rescaling of the objectives [179].

Whenever $\tilde{F}$ is a non-dominated set, the energy difference between any two of its elements is zero. The reason for inclusion of the current solution $\boldsymbol{x}$ and the neighbouring solution $\boldsymbol{x}'$ in the definition of $\tilde{F}$, is that $\Delta E(\boldsymbol{x}', \boldsymbol{x}) < 0$ if $\boldsymbol{x}' \prec \boldsymbol{x}$, which ensures that even if both the neighbouring and current solutions are dominated by the same number of non-dominated solutions in F, neighbouring solutions that are closer to the archived non-dominated front F are

[2] If P_f is a continuous set, the energy is taken as a Lebesgue measure (informally, the length, area or volume for problems with two, three, or four objectives, respectively).



(a) *Worse neighbouring solution* (b) *Improved neighbouring solution*

Figure 4.2: *Example of the energy measure for a current solution* $\mathbf{x}$ *and its neighbouring solution* $\mathbf{x}^i$*, used to calculate the probability* $P(\mathbf{x}^i)$ *of accepting the neighbouring solution, in a bi-objective minimisation problem.*

always accepted. Besides its simplicity and efficiency in promoting the storage of non-dominated solutions uncovered during the search process, another benefit of this energy measure is that it encourages exploration along the non-dominated front, regardless of the portion of the true Pareto front dominating a solution, as illustrated in Figure 4.2(b).

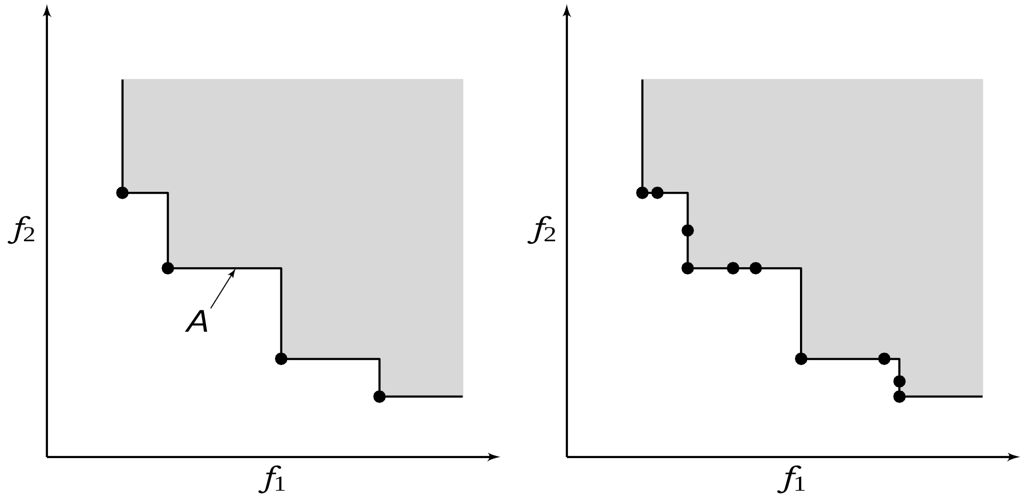
In the implementation of Smith *et al.*'s DMOSA algorithm [179], a solution to a GMS problem instance is denoted by a vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ which represents the integer scheduled maintenance starting times for each of the n generating units. From these starting (integer) values the binary decision variables X_{ij} are easily found and with the maintenance duration parametric values d_i, the auxiliary variables Y_{ij} may be determined from which all the objectives and constraint functions of the model may be determined as described in Chapter 3.

4.3.3 Interpolating the non-dominated front

When there is a small number of points in the archived non-dominated front, the estimated front may be interpolated by points on the attainment surface, as described in [179], in order to achieve better energy representations, as illustrated in Figure 4.3. If only one point is in the current front, points may be interpolated between this single point and the previous non-dominated point added to the archive, as illustrated in Figure 4.4.

If the number of points in the archived non-dominated front is below ten, then the archived front is interpolated in this dissertation by points on the attainment surface. The number of interpolated points added to the archived non-dominated front is such that the total number of archived non-dominated points found thus far by the SA algorithm, together with the interpolated points, sum up to ten, as illustrated in Figures 4.3 and 4.4.

When the algorithm initialises, the initial solution forms the current front (containing only one



(a) *Before interpolating the front* (b) *After interpolating the front*

Figure 4.3: *Interpolating the archived non-dominated front with points from the attainment surface A for a bi-objective minimisation problem.*

point) and obviously no previous non-dominated point has yet been added to the archive in order to interpolate the non-dominated front as illustrated in Figure 4.4. In this case, the initial point is also substituted as the previous non-dominated solution added to the archive, in which case nine exactly similar points are to be added (as "interpolated" points). This in effect means that the front is not interpolated at all during the first iteration of the algorithm, but will naturally be later on as the search progresses.

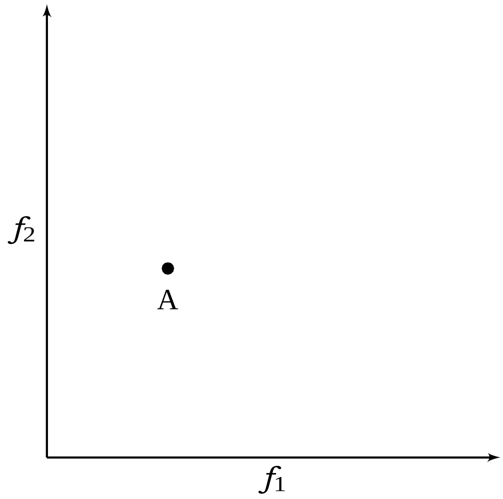
4.3.4 Constraint handling

Constraint handling techniques for metaheuristics[3] in the literature [39, 187] may be classified into the following seven classes:

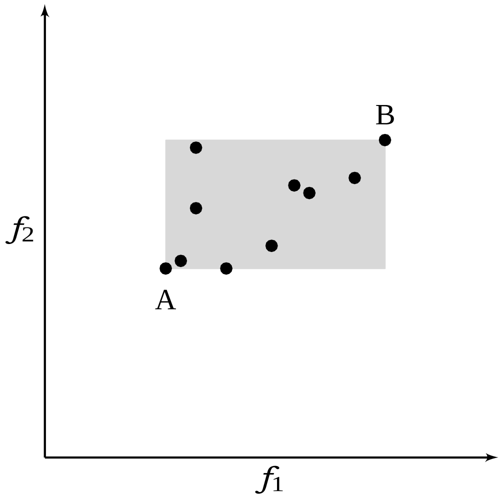
Rejecting strategies — A simply implementable approach where only feasible solutions are kept (considered) during the search process, *i.e.* infeasible solutions generated are discarded/rejected. These strategies are only effective when the proportion of feasible solutions is very large. These strategies do not, however, exploit infeasible solutions (*i.e.* guiding the search process to a feasible area by "travelling through" infeasible regions of the solution space in order to find desirable feasible solutions). A major disadvantage to these strategies is that if the proportion of feasible solutions is very small, then it may be difficult to find even feasible solutions within acceptable computation budgets.

Penalising strategies — One of the most popular techniques due to the relative ease of implementation and effectiveness, where infeasible solutions are also considered during the search process, but objective function values are penalised when the solutions are infeasible. These penalties may be linear or nonlinear, static or dynamic, or adaptive. A significant disadvantage of these strategies is that one has to choose a penalty function and parameters suited to the problem instance at hand, usually requiring a considerable amount of experimentation.

[3]Considerably more work has been done on SO metaheuristics than on MO metaheuristics [39, p. 115].



(a) *Before interpolating the front*



(b) *After interpolating the front*

Figure 4.4: *Interpolating the archived non-dominated front, if only one point (A) is in the front, with the previous point (B) removed from archive.*

Repairing strategies — Here heuristic (mostly greedy) algorithms transform infeasible solution into feasible ones. These heuristics are usually problem-specific. Repairing strategies may, however, be computationally expensive to implement.

Decoding strategies — The topology of the search space is transformed/mapped to a space consisting only of feasible solutions by these strategies.

Preserving strategies — These strategies incorporate problem-specific knowledge into the solution representation and search operators generate only feasible solutions during encoding. For some problems, however, such as the graph coloring problem, it is even difficult to find feasible initial solutions or a population of solutions to initialise the search.

Treating constraints as objectives — Constraint violation values are ranked along with the objective function values by these strategies. Constraints may be summed together into one single function or may be considered as separate functions for each of the separate, violated constraints.

Selecting for feasibility — Selection rules preferring feasible solutions over infeasible solutions are employed by these strategies. This sort of scheme can easily be extended to MOEAs by employing, for example, binary tournament selection.

GMS problem instances are usually highly constrained, rendering rejecting strategies undesirable, since it may take too long to find (or even initiate the search with) feasible solutions. For some of the GMS constraints it is impossible to avoid violation merely through correct solution encoding, such as satisfying the load demand. Preserving and decoding strategies are therefore also not desirable in the context of the GMS problem. Selecting for feasibility strategies are more reserved for MOEAs. Treating constraints as objectives is an appealing technique, but this adds more complexity to the problem and the decision maker. (Later in this dissertation, some experimentation takes place on relaxing the constraints and incorporating some of the GMS constraint violations as another objective to be minimised.)

In this dissertation, solutions are generated so that they always satisfy the hard constraints of the problem, namely earliest and latest maintenance starting times (3.1), the duration of maintenance (3.2) and the requirement that maintenance must occur over consecutive time periods (3.3), which is a preserving strategy. The other constraints of the model, namely the demand constraint (3.4), the maximum crew availability constraint (3.6), and the exclusion constraints (3.7), are too difficult to satisfy by preserving (or decoding or repairing) strategies and are thus all treated as soft constraints. Their violations are minimised by a penalising strategy, since it is the most popular constraint handling technique for single and multi-objective EAs (as well as some other metaheuristics) [39, p. 113].

A multiplicative penalty function developed by Schlünz *et al.* [171] is employed whereby any constraint violation incurs a penalty value related to the magnitude of that violation. Define, for the constrained MOO problem (4.1a)–(4.1d) the total scaled constraint violation of the inequality constraint functions $g_1(\boldsymbol{x}), \ldots, g_u(\boldsymbol{x})$ as

$$G(\boldsymbol{x}) = \sum_{i=1}^{u} \max\left(0, \frac{g_i(\boldsymbol{x}) - G_i}{G_i}\right). \tag{4.6}$$

Similarly, let

$$H(\boldsymbol{x}) = \sum_{j=1}^{v} \frac{h_j(\boldsymbol{x}) - H_j}{H_j} \tag{4.7}$$

be the total scaled constraint violation of the equality constraint functions $h_1(\boldsymbol{x}), \ldots, h_v(\boldsymbol{x})$. Given a severity factor γ as a free parameter whose value is typically determined empirically, define the multiplicative penalty function as

$$\varphi(\boldsymbol{x}) = e^{\gamma(G(\boldsymbol{x})+H(\boldsymbol{x}))}. \tag{4.8}$$

The objective functions to be minimised are multiplied by this penalty violation function value, *i.e.* the problem is now to

$$\text{minimise } \boldsymbol{f}(\boldsymbol{x}) = \varphi(\boldsymbol{x})[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_o(\boldsymbol{x})]. \tag{4.9}$$

This multiplicative penalty function only involves one parameter whose value must be chosen judiciously, namely the severity of penalty factors γ. In other penalising strategies, in contrast, each constraint set will typically have its own severity parameter which therefore require considerably more parameter fine-tuning. In addition, this multiplicative penalty function scales each constraint to the amount of violation, thus avoiding having to scale weights for the violation of different constraints' units of measure.

4.3.5 Cooling and reheating schedules

Convergence of the SA algorithm to a global minimum is guaranteed if and only if the cooling schedule is sufficiently gradual [178]. It has, however, been shown that SA may be a very effective metaheuristic optimisation technique even in conjunction with relatively rapid cooling schedules [111]. A SA algorithm is called a *simulated quenching* algorithm when the temperature schedule is too fast [111].

Although a variety of cooling schedules have been proposed for the SA algorithm, the predominant cooling schedule used for GMS problems, is the well-known geometric cooling schedule [169]. Schlünz and van Vuuren [169] compared GMS results obtained using this and three other cooling schedules. The results indicated that the geometric cooling schedule achieved the second best solution quality, behind a cooling schedule proposed by van Laarhoven and Aarts [196], but the

Table 4.1: *Solution quality and run-time results for different cooling parameters in the context of a large GMS problem instance solved by SA in [167]. The algorithm was terminated when the CPU time reached 60 minutes, in which case the time-out cost is marked by an asterisk.*

Cooling rate	Cost	Time
0.950	41 858	35.6 min.
0.955	36 733	42.0 min.
0.960	37 204	45.0 min.
0.965	31 819	53.4 min.
0.970	38 367*	60.0 min.
0.980	31 729*	60.0 min.

SA search time associated with the geometric cooling schedule was less. Since the bi-objective model proposed in Chapter 3 takes relatively long to solve, the geometric cooling schedule is adopted in this dissertation. There are also other successful *adaptive* cooling schedules [192], where feedback is received from the algorithm in order to evaluate what the next decrement in temperature should be. In such cases the cooling schedule depends on the change in objective value at the end of any given epoch. Although such a change is easily measurable for solving SO optimisation problems, it is not clear how this is applicable to MOO problems. In addition, the geometric cooling schedule is simple, effective and by far the most widely applied cooling schedule [61, p. 30], especially in the context of GMS [169]. The updating rule for this schedule is

$$T_{e+1} = \alpha T_e, \quad e = 0, 1, 2, \ldots, \tag{4.10}$$

where T_e is the temperature during epoch e of the search process and $\alpha \in (0, 1)$ is a constant called the *cooling parameter*.

A further addition to the geometric cooling scheduling, although not employed as often, is geometric reheating, which allows the SA algorithm to escape more efficiently from local minima [2, 13]. Reheating is achieved by increasing the temperature by some factor at the end of certain epochs (instead of cooling). A simple geometric reheating schedule, suggested in [2], is

$$T_{e+1} = \frac{T_e}{\beta}, \tag{4.11}$$

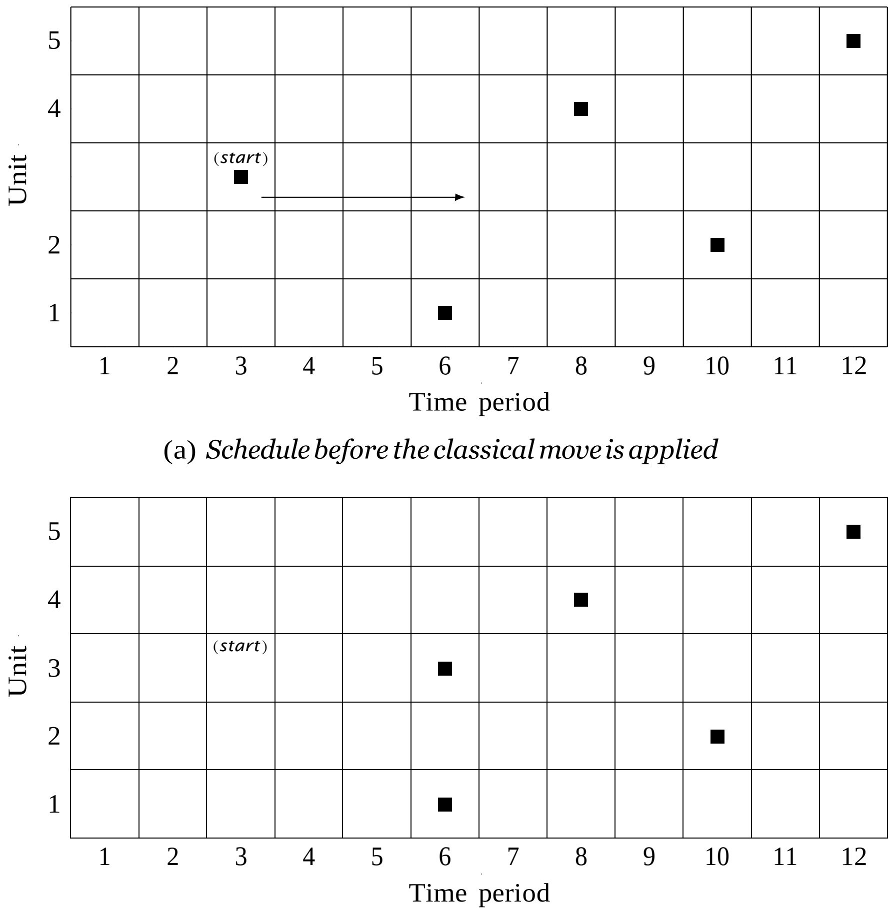
where $\beta \in (0, 1)$ is a constant called the *reheating parameter*.

Suggestions in the literature for good values of α mostly range from 0.9 to 0.99 [46, 61, 103] (or sometimes even as far down as 0.85 [82]), with suggested values for β being similar [2, 28] or being equal to 0.5 [13]. It has been suggested that a lower value of β may be employed for faster reheating than cooling [2]. For instance, in [192] it is suggested that β-values typically range from 0.25 to 0.67. As noted in [60], however, empirical evidence in the literature for general SA applications supports the need for slow cooling, where typical values are in the range [0.8, 0.99] with a bias to higher values of α (*i.e.* slower cooling). There are two conflicting factors to be taken into account when selecting the value of the cooling parameter, namely the quality of solutions uncovered and computing time, as illustrated in Table 4.1. There is, however, no general rule for choosing the value of α in (4.10) and β in (4.11). The optimal values for these parameters are problem-specific and must therefore be determined empirically through experimentation [167].

The number of iterations during each epoch (*i.e.* the length of each epoch) is determined dynamically. This is adhered by implementing different variations of the suggestion in [61, p. 45], which states that the inner Metropolis loop of the SA algorithm should terminate when one of the following two conditions is met: a maximum of $12N$ solutions are accepted, or a maximum of $100N$ solutions are attempted, where N is a measure of the number of degrees of freedom in the optimisation problem — in this case $N = n$ (the number of units).

4.3.6 GMS neighbourhood move operators

Two types of SA neighbourhood move operators (perturbation of the current solution) are predominant in the GMS literature, the one being a special case of the other [169]. According to the first move operator, called the *classical* neighbour move operator and illustrated in Figure 4.5, one unit is randomly selected according to a uniform distribution and its maintenance starting time is then randomly changed to a new value within the allowed maintenance window (again according to a uniform distribution). Another very similar operator restricts the new starting value to the two adjacent time periods (a slight change in starting time), *e.g.* if the selected unit's starting time period is 3, then its new starting time period may only be 4 or 5. These two similar moves are classified as *elementary* moves.

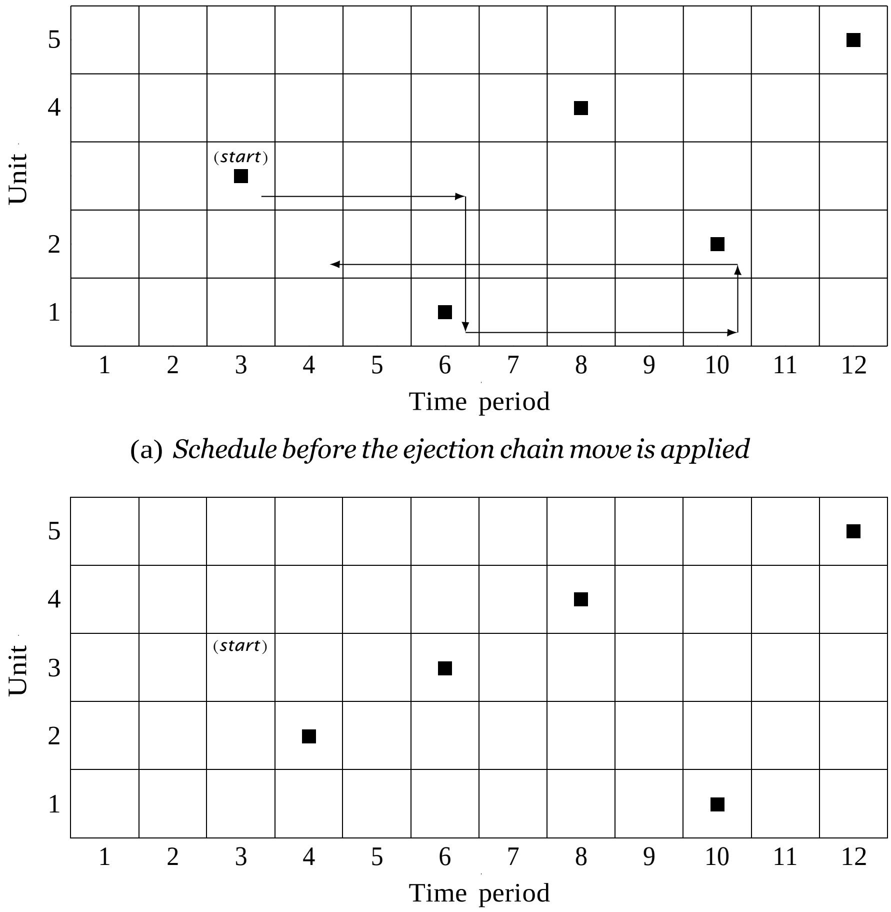


(a) *Schedule before the classical move is applied*

(b) *Schedule after the classical move is applied*

Figure 4.5: *The classical move illustrated through an example.*

A second GMS move operator, classified as a *compound* move, was proposed in [169], is called the *ejection chain* neighbour move operator and is illustrated in Figure 4.6. The ejection chain operator initiates similarly to the classical move operator by randomly selecting a unit according to a uniform distribution and perturbing its maintenance starting time (*start*) randomly to a new value (*new start*) within its allowed maintenance window. Then another unit whose maintenance starts during this newly selected starting time (*new start*) is chosen at random, and its maintenance starting time is randomly perturbed to a new value within its allowed maintenance window. This process is repeated until either the newly selected starting time for a unit corresponds to the initial unit's starting time *start*, or no unit is found for which maintenance starts during the newly selected time (see Figure 4.6). The ejection chain is a more global move operator, typically affecting large changes to a solution, whilst the classical move operator is confined to more local changes. The ejection chain neighbourhood operator was found to be superior to the classical neighbourhood move operator when comparing the best objective function values found, but at the cost of potentially requiring significantly more solution time [169]. The ejection chain neighbourhood move operator is implemented in this dissertation.



(a) *Schedule before the ejection chain move is applied*

(b) *Schedule after the ejection chain move is applied*

Figure 4.6: *The ejection chain move operator illustrated through an example.*

4.3.7 Algorithmic initialisation

An initial solution for the SA algorithm is determined as follows. For each generating unit $i \in I$, a random maintenance starting time X_i is generated uniformly between its earliest (e_i) and latest (f_i) starting time. The corresponding objective function values are determined, as described above (including incorporation of the constraint violations).

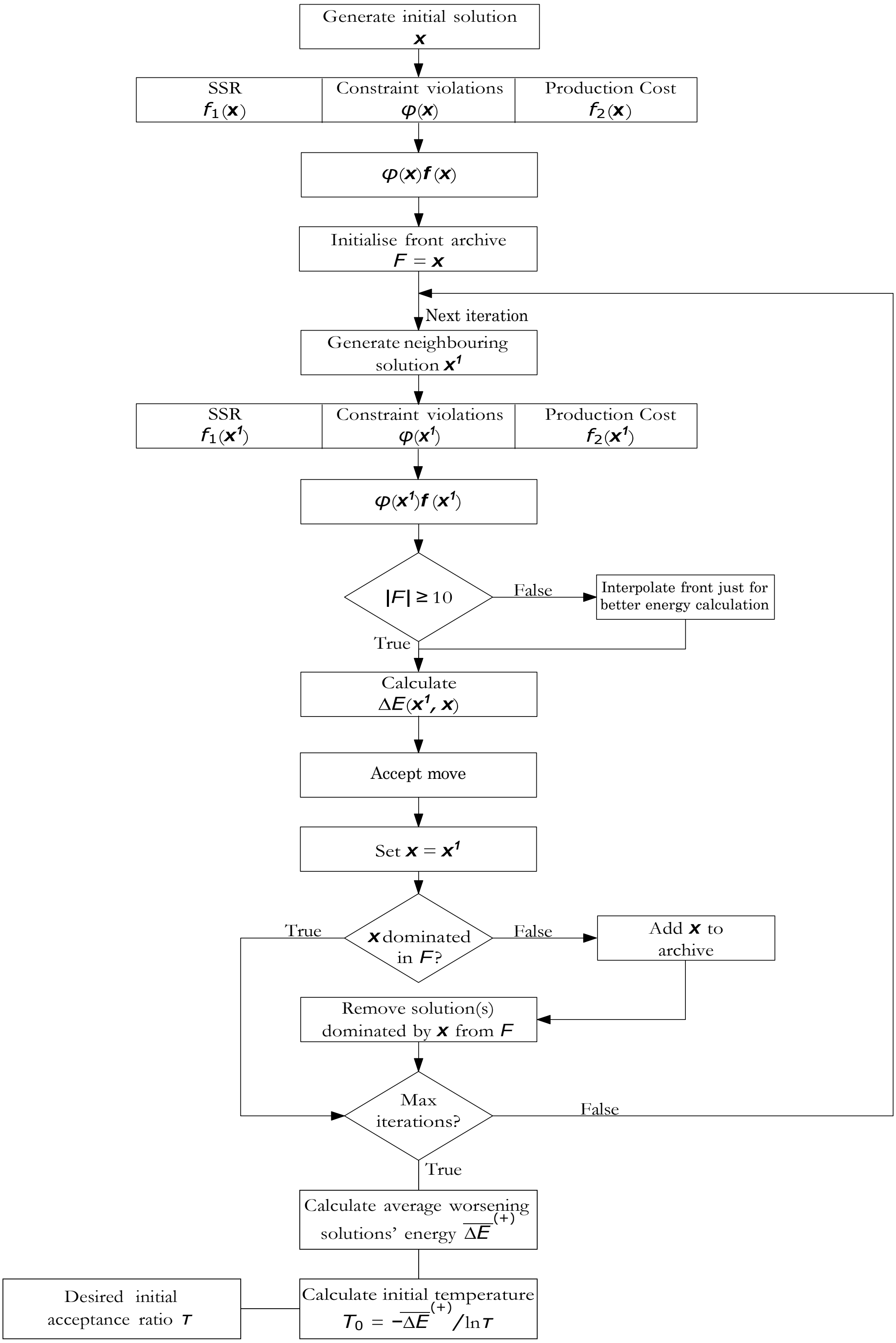


Figure 4.7: *Flow chart of the process followed to implement the initial random walk heuristic before executing the DMOSA algorithm in order to calculate the desired initial temperature.*

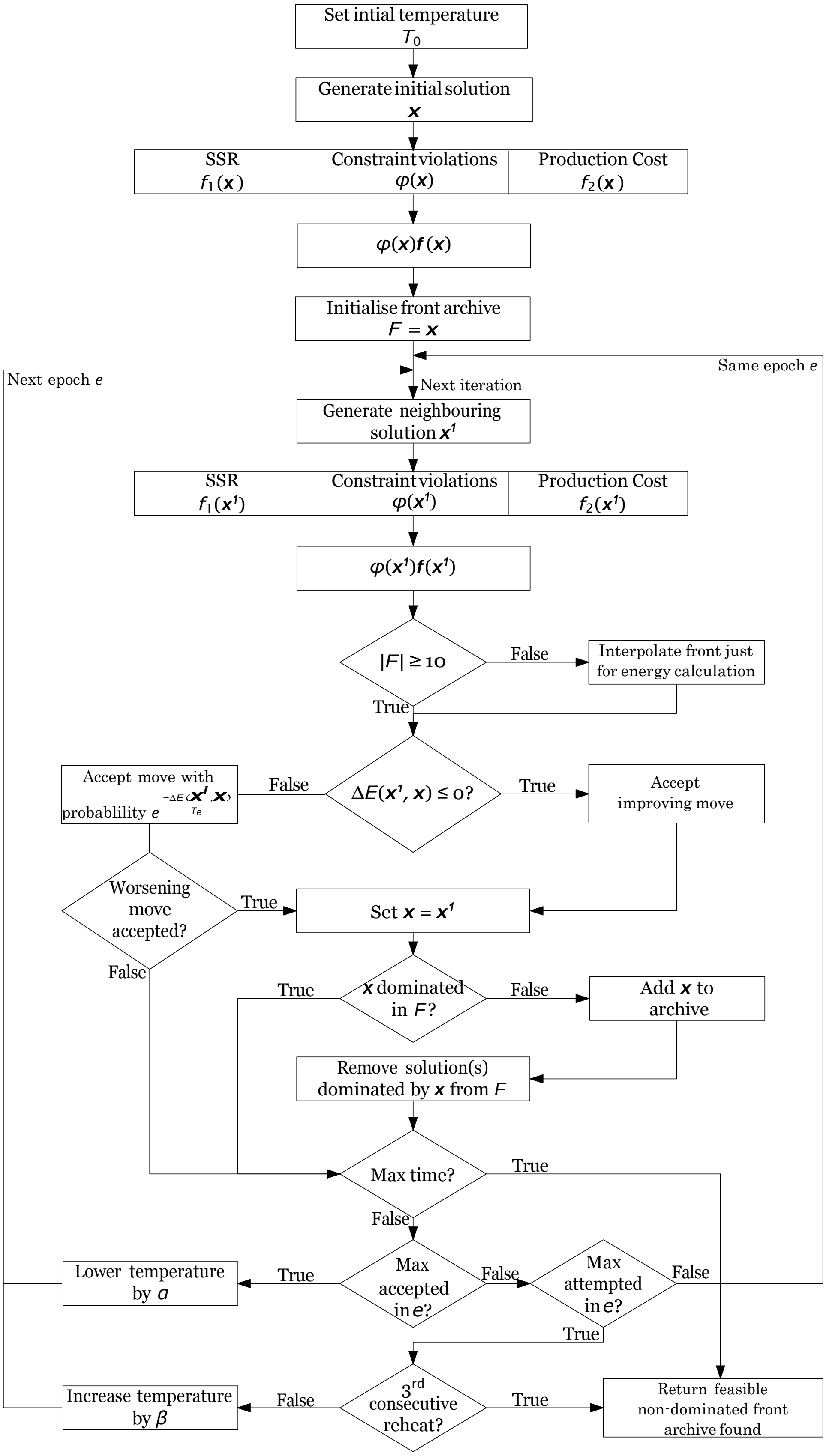


Figure 4.8: *Flow chart demonstrating the working of the DMOSA algorithm as implemented in this dissertation.*

The initial temperature T_0 is calculated according to the average increase method presented in [192], as $T_0 = -\overline{\Delta E}^{(+)} / \ln \tau$, where τ is the initial acceptance ratio (the number of accepted worsening moves to the number of attempted moves), which may be specified by the user and $\overline{\Delta E}^{(+)}$ is the average increase in energy (worsening of the objective function value). The value of $\overline{\Delta E}^{(+)}$ may be estimated by executing a random walk (in this dissertation consisting of 100 iterations) within the solution space, using the randomly generated initial solution as starting point.

4.3.8 Algorithmic termination criteria

The SA algorithm terminates when the temperature loop terminates according to pre-specified criteria. A number of such termination criteria have been proposed in the literature. In this dissertation, the following two termination criteria are implemented: termination occurs when a pre-specified number Ω_{frozen} of successive reheatings have been performed or when a pre-specified budget $t_{\max}$ of computing time has elapsed. In this dissertation $\Omega_{\text{frozen}} = 3$, as suggested in [61, p. 45], whilst $t_{\max}$ is taken as eight hours, so that the user may find a new maintenance schedule within the confines of a normal working day, or overnight. Some experimentation in terms of running the algorithm for longer is, however, performed later in this dissertation, especially for large GMS problem instances.

4.4 Algorithmic implementation

The DMOSA algorithm and all other calculations described in this chapter were implemented by the author in the programming language R (version 3.0.2) within the RStudio *integrated development environment* (IDE). The *lpSolve* package in R was used to solve the LP model for the ED problem proposed in §3.3. All the computation evaluations were performed on a personal computer with a 3.40 GHz Intel® Core™ i7-4770 CPU processor with 8.11 GB RAM and running in Ubuntu Gnome 3.10.04.

Figure 4.7 contains a flow chart of the process followed to implement the initial random walk heuristic before executing the DMOSA algorithm in order to calculate the desired initial temperature, while Figure 4.8 contains a flow chart of the DMOSA algorithm as implemented in this dissertation to compute non-dominated fronts for the GMS model proposed in Chapter 3.

4.5 Chapter summary

In this chapter, the details of the solution methodology employed in this dissertation were presented. General MOO principles were presented in §4.1 after which a motivation for the choice of the SA algorithm as a solution technique employed in this dissertation was presented in §4.2. The working of the SA algorithm and the different types of MOSA algorithms present in the literature were next described in §4.3. This discussion contained a description of how Smith *et al.*'s DMOSA algorithm [179] is implemented in this dissertation. Some technical implementation information was finally presented in §4.4.

CHAPTER 5

Case study data

Contents

Two case studies are carried out later in this dissertation in order to showcase the workability of applying the novel modelling approach proposed in Chapter 3 to existing GMS problem instances. The first case study is a 32-unit benchmark established in the literature. The second is a real-world case study involving the generating units of the South African Power Utility, Eskom, (involving slightly altered data in order not to divulge sensitive information). All the parameters and data required to solve the GMS model proposed in Chapter 3 in the contexts of these case studies are provided in this chapter.

5.1 A 32-unit IEEE-RTS inspired case study

In 2011, Schlünz and van Vuuren [168, 169] created a test system derived from the load model and generation system of the 1979 IEEE *Reliability Test System* (RTS) [9] with the addition of certain constraints and parameter values. This IEEE-RTS inspired case study consists of a GMS problem instance containing 32 generating units with a total installed capacity of 3 405 MW. The units require maintenance over a 52-week planning horizon, with the single objective of levelling reserves, by minimising the SSR over the planning period as specified in (3.9) [169]. The constraints of the system consist of the specification of maintenance windows, respecting exclusion constraints, meeting of the expected peak demand together with a safety margin, and adhering to the availability of maintenance crew as specified in (3.1)–(3.8). In particular, $S = 15\%$ and M_j=25 for all $j \in J$. The other parameters required for the specification of the constraints may be found in the first eight columns of Table 5.1 and the interested reader is referred to [169, 9] for more details on this test system.

The same data set as described above is also employed in this dissertation, but the data set employed here also includes additional information required for the added objective of minimising production cost. This additional information was inferred from the original 1979 IEEE-RTS [9], and is specified in the last five columns of Table 5.1.

Table 5.1: *Data for the 32-unit IEEE-RTS inspired case study.*

Unit i	Installed Capacity (MW) I_i	Earliest starting time (week) e_i	Earliest starting time (week) l_i	Duration (weeks) d_i	Manpower required during each period of maintenance μ_i^u	Exclusion set k	Max # of units K_k	Station Name	Heat Rate[1] $(\frac{\text{Btu}}{\text{kWh}})$	Fuel Cost[2] $(\frac{\$}{\text{MBtu}})$	Cost[2] Rate $(\frac{\$}{\text{MWh}})$ C_s	Min/Max EUF (%) $q_s^{\min}/q_s^{\max}$
1	50	1	51	2	6, 6	6	3	Hydro			0	0/100
2	50	1	51	2	6, 6	6	3					
3	50	1	51	2	6, 6	6	3					
4	50	1	51	2	6, 6	6	3					
5	50	1	51	2	6, 6	6	3					
6	50	1	51	2	6, 6	6	3					
7	400	1	21	6	15, 10, 10, 10, 10, 5			Nuclear	10 000	0.60	6.00	100/100
8	400	27	47	6	15, 10, 10, 10, 10, 5							
9	350	1	48	5	5, 10, 15, 15, 5	7	1	Coal-1	9 500	0.12	11.40	60/100
10	155	1	23	4	5, 15, 10, 10	5	3	Coal-2	9 700	0.12	11.64	60/100
11	155	27	49	4	5, 15, 10, 10							
12	155	1	23	4	12, 12, 8, 8	7	1					
13	155	1	49	4	12, 12, 8, 8	7	1					
14	76	1	24	3	12, 10, 10	2	2	Coal-3	12 000	0.12	14.40	60/100
15	76	27	50	3	12, 10, 10	2	2					
16	76	1	24	3	12, 10, 10	1	2					
17	76	27	50	3	12, 10, 10	1	2					
18	197	1	23	4	8, 10, 10, 8	4	1	Oil-1	9 600	0.23	22.08	60/100
19	197	1	23	4	8, 10, 10, 8	4	1					
20	197	27	49	4	8, 10, 10, 8	4	1					
21	100	1	50	3	10, 10, 15	3	1	Oil-2	10 000	0.23	23.00	60/100
22	100	1	50	3	10, 10, 15	3	1					
23	100	1	50	3	15, 10, 10	3	1					
24	12	1	51	2	4, 4	5	3	Oil-3	12 000	0.23	27.60	60/100
25	12	1	51	2	4, 4	5	3					
26	12	1	51	2	4, 4	5	3					
27	12	1	51	2	4, 4	5	3					
28	12	1	51	2	4, 4	5	3					
29	20	1	25	2	7, 7	1	2	Gas Turbine	14 500	0.30	43.50	4/100
30	20	1	25	2	7, 7	1	2					
31	20	1	25	2	7, 7	2	2					
32	20	27	51	2	7, 7	2	2					

[1] This is taken as 100% output, which is not exactly correct as explained in §5.1.1, since this assumes that the power plant will always run at 100%. A constant heat rate is, however, usually adopted in approximate cost representations [136, 126] so that a constant production (or fuel) cost rate ($/MWh) may be assumed, as in [126, 136, 216]. [2] Monetary value relative to 1979.

5.1.1 Generation cost LP for the 32-unit IEEE-RTS inspired case study

The generation mix of the appended IEEE-RTS inspired case study described above (hereafter referred to as the IEEE-RTS inspired case study) consists of one conventional hydroelectric power station, one nuclear power station, three coal-fired power stations, three oil-fired power stations and one gas-turbine station. The index values for the various power stations are captured in Table 5.2, employing the same notation as in Chapter 3.

Table 5.2: *Index values assigned to the power stations in the 32-unit IEEE-RTS inspired case study.*

h	S_h	Description	Type
1	$\{1\}$	Conventional hydroelectric	Peak/Exact
2	$\{2\}$	Nuclear	Base
3	$\{3, 4, 5\}$	Coal	
4	$\{6, 7, 8\}$	Oil	
5	$\{9\}$	Gas-Turbine	Peak
6	$\{10\}$	Unmet	Virtual

Referring to Table 5.2 for the numbering, the power stations in subsets S_2, S_3, and S_4 are considered base load stations, while those in S_1 and S_5 are considered peak demand stations. Each base load station has a maximum and minimum daily production requirement associated with it, while the conventional hydroelectric station has an exact daily production requirement, shown in Table 5.3.

Given this information, the objective in the special case of the general LP generation cost model described in §3.3 to be solved for the IEEE-RTS inspired case study is to

$$\text{minimise} \sum_{t\in T}\sum_{s\in S} C_s Z_{st} \tag{5.1a}$$

subject to

$$Z_{st} \leq A_{st}, \quad s \in S, \quad t \in T, \tag{5.1b}$$

$$\sum_{s\in S} Z_{st} = L_t, \quad t \in T, \tag{5.1c}$$

$$\sum_{t\in T} Z_{st} \leq \;_{s}, \quad s \in S_2 \cup S_3 \cup S_4, \tag{5.1d}$$

$$\sum_{t\in T} Z_{st} \geq \;_{s}, \quad s \in S_2 \cup S_3 \cup S_4, \tag{5.1e}$$

$$\sum_{t\in T} Z_{1t} = f_1^{\text{req}}, \tag{5.1f}$$

$$Z_{st} \geq 0, \quad s \in S, \quad t \in T. \tag{5.1g}$$

Constraint set (5.1b) ensures that the planned energy production Z_{st} of power station $s \in S$ during time slice $t \in T$ does not exceed its available capacity A_{st} during that time slice, while constraint set (5.1c) balances planned energy and the forecast demand L_t during time slice $t \in T$. The maximum production requirements, $f_s^{\max}$ and $f_s^{\min}$ respectively, for base load station $s \in S_2 \cup S_3 \cup S_4$ are ensured by constraint sets (5.1d) and (5.1e), respectively. The daily production requirement for the conventional hydroelectric station in S_1 is specified by constraint (5.1f), while constraint set (5.1g) is the usual sign restriction constraint.

Table 5.3: *Hydro capacity and energy for the 32-unit IEEE-RTS inspired case study [9]. In the original system the total energy was specified as 200 GWh. This value has been reduced 100 GWh so as to enforce a tighter reserve.*

Quarter	Capacity Available[1](%)	Energy Distribution[2](%)
1	100	35
2	100	35
3	90	10
4	90	20

[1] 100% capacity = 50 MW
[2] 100% energy = 100 GWh

The production cost rate ($/MWh) for the IEEE-RTS inspired case study was purely estimated as the fuel cost, by multiplying the maximal output (100%) heat rate values for the different power stations by the stations' corresponding fuel rate costs given in [9], as illustrated in Table 5.1. These fuel costs are, however, applicable to the 1979 value of money. In order to have more meaningful up-to-date monetary values, these fuel cost rates were inflated (for the United States of America) using an online inflation calculator [40], to represent realistic 2016 values. The cumulative rate of inflation used for this purpose was 231.5%, meaning that all original cost values were multiplied by 3.315. Although the above estimation is not entirely accurate, since power plants will not always run at 100% (their production rates will actually vary based on the production planning module described in §3.3), this cost modelling approach is nevertheless thought to embody a good representation serving the purpose of creating a constant fuel cost rate C_S ($/MWh). This approach has been used widely in the literature [32, 126, 137, 216].

The UCLF and OCLF values were set to zero, so as to only analyse how the PCLF values (which depend solely on the GMS decision variables) affect the fuel cost.

5.1.2 Hourly load data

The hourly loads were calculated from the original IEEE-RTS achieving an annual peak load of 2 850 MW and the weekly, daily and hourly peak percentages of this annual load given in [9]. This yielded a total annual energy demand of 15 297.478 GWh, which almost exactly matches the energy demand calculated in the IEEE-RTS follow-on paper in 1986 [11] of 15 297.075 GWh, thus serving as validation of the hourly load data considered in this dissertation. Table 5.4 contains the daily peak load demand (from the hourly data) for the 32-unit inspired case study. The weekly peak demand is used, as in [168, 169], to calculated the SSR.

5.2 A 157-unit Eskom case study

In 2011, Schlünz and van Vuuren [168, 170] created a larger and more realistic case study instance of the GMS problem based on the generating units of the South African electricity utility, Eskom. The Eskom case study consists of 105 generating units with a total installed capacity of 39 949 MW requiring maintenance over a 365-day period planning horizon, with the single objective of levelling reserves, by minimising the SSR over the planning period as

Table 5.4: *Daily peak demand data for the 32-unit IEEE-RTS inspired case study [9], the 52 weekly peak demands were used to calculate the SSR as in [169].*

Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)
1	2285	62	1624	123	2290	184	2152	245	1552	306	2360
2	2457	63	1582	124	2242	185	2109	246	1868	307	1933
3	2408	64	1953	125	1836	186	2066	247	2009	308	1883
4	2359	65	2100	126	1789	187	2023	248	1969	309	2345
5	2310	66	2058	127	2306	188	1657	249	1929	310	2522
6	1892	67	2016	128	2480	189	1614	250	1888	311	2472
7	1843	68	1974	129	2430	190	2163	251	1547	312	2421
8	2385	69	1617	130	2381	191	2326	252	1507	313	2371
9	2565	70	1575	131	2331	192	2279	253	2067	314	1942
10	2514	71	1895	132	1910	193	2233	254	2223	315	1892
11	2462	72	2038	133	1860	194	2186	255	2179	316	2410
12	2411	73	1997	134	2332	195	1791	256	2134	317	2591
13	1975	74	1956	135	2508	196	1745	257	2090	318	2539
14	1924	75	1916	136	2458	197	2123	258	1712	319	2487
15	2327	76	1569	137	2408	198	2283	259	1667	320	2436
16	2502	77	1529	138	2358	199	2237	260	1842	321	1995
17	2452	78	1927	139	1931	200	2192	261	1981	322	1943
18	2402	79	2072	140	1881	201	2146	262	1941	323	2491
19	2352	80	2031	141	2269	202	1758	263	1902	324	2679
20	1927	81	1989	142	2440	203	1712	264	1862	325	2625
21	1877	82	1948	143	2391	204	2332	265	1525	326	2572
22	2211	83	1595	144	2342	205	2508	266	1486	327	2518
23	2377	84	1554	145	2294	206	2458	267	1919	328	2063
24	2329	85	1866	146	1879	207	2408	268	2063	329	2009
25	2282	86	2006	147	1830	208	2358	269	2022	330	2359
26	2234	87	1966	148	2149	209	1931	270	1980	331	2537
27	1830	88	1926	149	2311	210	1881	271	1939	332	2486
28	1783	89	1886	150	2265	211	1914	272	1589	333	2436
29	2332	90	1545	151	2219	212	2058	273	1547	334	2385
30	2508	91	1505	152	2172	213	2017	274	1919	335	1953
31	2458	92	1988	153	1779	214	1976	275	2063	336	1903
32	2408	93	2138	154	1733	215	1935	276	2022	337	2497
33	2358	94	2095	155	2385	216	1585	277	1980	338	2685
34	1931	95	2052	156	2565	217	1544	278	1939	339	2631
35	1881	96	2010	157	2514	218	2057	279	1589	340	2578
36	2229	97	1646	158	2462	219	2212	280	1547	341	2524
37	2397	98	1604	159	2411	220	2168	281	1970	342	2067
38	2349	99	1911	160	1975	221	2124	282	2118	343	2014
39	2301	100	2055	161	1924	222	2079	283	2076	344	2571
40	2253	101	2014	162	2351	223	1703	284	2033	345	2765
41	1846	102	1973	163	2528	224	1659	285	1991	346	2710
42	1798	103	1932	164	2477	225	2120	286	1631	347	2654
43	2205	104	1582	165	2427	226	2280	287	1589	348	2599
44	2371	105	1541	166	2376	227	2234	288	1972	349	2129
45	2324	106	2120	167	1947	228	2189	289	2120	350	2074
46	2276	107	2280	168	1896	229	2143	290	2078	351	2651
47	2229	108	2234	169	2375	230	1756	291	2035	352	2850
48	1826	109	2189	170	2554	231	1710	292	1993	353	2793
49	1778	110	2143	171	2503	232	1933	293	1632	354	2736
50	2136	111	1756	172	2452	233	2078	294	1590	355	2679
51	2297	112	1710	173	2401	234	2036	295	2120	356	2195
52	2251	113	1999	174	1967	235	1995	296	2280	357	2138
53	2205	114	2149	175	1916	236	1953	297	2234	358	2523
54	2159	115	2106	176	2282	237	1600	298	2189	359	2713
55	1769	116	2063	177	2454	238	1559	299	2143	360	2659
56	1723	117	2020	178	2405	239	1924	300	1756	361	2604
57	1961	118	1655	179	2356	240	2069	301	1710	362	2550
58	2109	119	1612	180	2307	241	2028	302	2335	363	2089
59	2067	120	2218	181	1890	242	1986	303	2511	364	2035
60	2025	121	2385	182	1841	243	1945	304	2461		
61	1982	122	2337	183	2001	244	1593	305	2411		

specified in (3.9) [170]. The data included in their GMS problem instance do not, however, represent the exact Eskom generation system, due to confidentiality concerns, but the case study nevertheless represents a realistic GMS scenario. Constraints in the Eskom case study instance are restricted to the adherence to maintenance windows, the system meeting the expected peak demand together with a safety margin, and respecting certain exclusion constraints. A safety margin of 8% was set (S = 8%) so as to ensure a minimum capacity of 2 000 MW over and above demand throughout the planning period.

Some units require more than one maintenance outage during the planning horizon, but since the GMS model of Chapter 3 requires each unit to experience a single maintenance outage, dummy units were added to the problem instance — one for each unit's additional outage during the year — thereby increasing the number of units in the system to 157. The resulting additional capacity that these dummy units provide to the system was subtracted from the new total capacity in order to render the system capacity unaffected by the addition of dummy units. Furthermore, the exclusion sets imposed on the system are only introduced to prevent dummy units from being in simultaneous maintenance with their corresponding (real) units. The parameters and data required for the specification of the model constraints may be found in the first eight columns of Table 5.5 and the interested reader is referred to [168] for more details on this test system. The dimensions of this GMS problem instance are considerably larger than those of other test systems in the literature.

Table 5.5: *Data for the Eskom case study sorted according to the merit order defined in Table 5.6.*

Unit i	Real unit	Unit name [168]	Cap. (MW) I_i	Earliest starting (day) e_i	Latest starting (day) l_i	Duration (days) d_i	Excl-usion k K_k	Station (Table 5.5) s	Production cost rate (R/MWh) C_s	Min/Max EUF (%) $\underline{q}_s^{\min}/\overline{q}_s^{\max}$
1	1	L1	900	225	282	84		3	0	100/100
2	2	L2	900	0	0	0				
3	3	P1	615	1	50	42		4	84	60/95
4	3	P1	615	302	323	7				
5	4	P2	615	71	92	7				
6	5	P3	615	323	331	35				
7	6	P4	615	43	85	28				
8	7	P5	615	85	127	28				
9	8	P6	615	190	211	7				
10	9	N1	593	43	85	28		6	96	60/95
11	10	N2	593	0	0	0				
12	11	N3	593	99	141	28				
13	12	N4	593	29	50	7				
14	13	N5	593	78	99	7				
15	13	N5	593	316	338	28				
16	14	N6	593	15	36	7				
17	14	N6	593	218	260	28				
18	15	F1	575	0	0	0		7	114	60/95
19	16	F2	575	71	148	70				
20	17	F3	575	211	239	14				
21	18	F4	575	0	0	0				
22	19	F5	575	1	78	70				
23	20	F6	575	232	288	45				

Table 5.5 (continued): Data for the Eskom case study sorted according to the merit order defined in Table 5.6.

Unit i	Real unit	Unit name [168]	Cap. (MW) I_i	Earliest starting (day) e_i	Latest starting (day) l_i	Duration (days) d_i	Excl-usion k K_k	Station (Table 5.5) s	Production cost rate (R/MWh) C_s	Min/Max EUF (%) q_s^{min}/q_s^{max}
24	21	J1	190	0	0	0		8	118	60/95
25	22	J2	185	1	78	92				
26	23	J3	190	0	0	0				
27	24	J4	190	92	120	14				
28	25	J5	190	204	267	50				
29	26	J6	190	0	0	0				
30	27	J7	190	113	141	14				
31	28	J8	190	1	43	28				
32	28	J8	190	260	288	14				
33	29	J9	190	281	309	14				
34	30	J10	190	218	260	28				
35	30	J10	190	302	338	28				
36	31	Q1	575	218	239	7		9	128	60/95
37	32	Q2	575	0	0	0				
38	33	Q3	575	0	0	0				
39	34	Q4	575	1	50	42				
40	35	Q5	575	85	106	7				
41	36	Q6	575	239	282	84				
42	37	M1	475	1	36	21		10	130	60/95
43	37	M1	475	288	330	28				
44	38	M2	475	1	78	84				
45	39	M3	475	323	345	21				
46	40	M4	475	0	0	0		11	130	60/95
47	41	M5	475	0	0	0				
48	42	M6	475	0	0	0				
49	43	K1	640	330	343	23		13	170	60/95
50	44	K2	640	1	22	7				
51	45	K3	640	295	330	23				
52	46	K4	640	323	344	5				
53	47	K5	640	1	71	57				
54	48	K6	640	85	106	5				
55	49	O1	612	1	43	28		14	180	60/95
56	50	O2	612	106	127	7				
57	50	O2	612	260	310	56				
58	51	O3	612	99	120	7				
59	52	O4	669	64	113	35				
60	53	O5	669	330	351	7				
61	54	O6	669	351	358	8				
62	55	T1	585	85	99	4		15	198	60/95
63	55	T1	585	274	306	60				
64	56	T2	585	92	120	12				
65	56	T2	585	344	358	4				
66	57	T3	585	64	78	4				

Table 5.5 (continued): Data for the Eskom case study sorted according to the merit order defined in Table 5.6.

Unit i	Real unit	Unit name [168]	Cap. (MW) I_i	Earliest starting (day) e_i	Latest starting (day) l_i	Duration (days) d_i	Excl-usion k	K_k	Station (Table 5.5) s	Production cost rate (R/MWh) C_s	Min/Max EUF (%) q_s^{min}/q_s^{max}
67	57	T3	585	218	246	12					
68	58	T4	585	1	64	50					
69	58	T4	585	253	267	4			15	198	60/95
70	59	T5	585	71	85	4					
71	60	T6	585	120	134	4					
72	60	T6	585	323	337	4					
73	61	D1	190	0	0	0					
74	62	D2	190	0	0	0					
75	63	D3	185	50	92	30					
76	64	D4	180	218	267	42			16	211	55/95
77	65	D5	180	0	0	0					
78	66	D6	160	0	0	0					
79	67	D7	170	1	50	42					
80	68	D8	180	85	134	42					
81	69	C1	330	36	50	4					
82	69	C1	330	225	239	4					
83	70	C2	350	29	57	14					
84	70	C2	350	253	267	4					
85	71	C3	380	36	50	4					
86	71	C3	380	281	295	4			17	231	55/95
87	72	C4	350	71	85	2					
88	72	C4	350	302	316	3					
89	73	C5	350	8	22	4					
90	73	C5	350	239	282	84					
91	74	C6	350	106	120	2					
92	74	C6	350	274	288	3					
93	75	I1	190	78	113	21					
94	76	I2	190	211	246	21					
95	77	I3	190	0	0	0			19	308	55/95
96	78	I4	190	309	344	21					
97	79	I5	190	0	0	0					
98	80	I6	190	0	0	0					
99	81	B1	148	0	0	0					
100	82	B2	148	0	0	0			20	2400	5/100
101	83	B3	148	0	0	0					
102	84	B4	148	0	0	0					
103	85	H1	148	0	0	0					
104	86	H2	148	0	0	0			21	2401	4/100
105	87	H3	148	0	0	0					
106	88	A1	57	8	29	9	1	1			
107	88	A1	57	15	29	2	1	1	22	2650	0/100
108	88	A1	57	169	183	2					
109	88	A1	57	281	295	2					

Table 5.5 (continued): Data for the Eskom case study sorted according to the merit order defined in Table 5.6.

Unit i	Real unit	Unit name [168]	Cap. (MW) I_i	Earliest starting (day) e_i	Latest starting (day) l_i	Duration (days) d_i	Exclusion k	K_k	Station (Table 5.5) s	Production cost rate (R/MWh) C_s	Min/Max EUF (%) $q_s^{\min}/q_s^{\max}$
110	89	A2	57	15	29	2	2	1	22	2650	0/100
111	89	A2	57	15	36	9	2	1			
112	89	A2	57	169	183	2					
113	89	A2	57	281	295	2					
114	90	A3	57	15	29	2	3	1			
115	90	A3	57	29	50	9	3	1			
116	90	A3	57	176	190	2					
117	90	A3	57	288	309	7					
118	91	S1	57	50	64	2			23	2651	0/100
119	91	S1	57	85	141	45	5	1			
120	91	S1	57	176	190	2	5	1			
121	91	S1	57	295	309	2					
122	92	S2	57	15	85	56	6	1			
123	92	S2	57	57	71	2	6	1			
124	92	S2	57	183	197	2					
125	92	S2	57	302	316	2					
126	93	S3	57	57	78	7					
127	93	S3	57	183	197	2					
128	93	S3	57	302	316	2					
129	94	G1	90	71	99	11			25	0	0/100
130	94	G1	90	246	246	120					
131	95	G2	90	15	29	2					
132	95	G2	90	232	246	120					
133	96	G3	90	43	71	11					
134	96	G3	90	295	323	14					
135	97	G4	90	1	85	121	4	1			
136	97	G4	90	113	127	2	4	1			
137	98	U1	120	22	57	21			26	0	0/100
138	98	U1	120	120	162	31					
139	98	U1	120	267	288	5					
140	99	U2	120	85	106	5					
141	100	E1	250	8	64	44			27	0	6/100
142	100	E1	250	134	148	1					
143	100	E1	250	281	295	1					
144	100	E1	250	316	344	14					
145	101	E2	250	8	64	44					
146	101	E2	250	330	344	1					
147	102	E3	250	29	43	1					
148	102	E3	250	267	295	14					
149	103	E4	250	8	22	1					
150	103	E4	250	50	64	3					
151	103	E4	250	309	337	14					
152	104	R1	200	22	57	25			28	0	8/100

Table 5.5 (continued): Data for the Eskom case study sorted according to the merit order defined in Table 5.6.

Unit i	Real unit	Unit name [168]	Cap. (MW) I_i	Earliest starting (day) e_i	Latest starting (day) l_i	Duration (days) d_i	Excl-usion k K_k	Station (Table 5.5) s	Production cost rate (R/MWh) C_s	Min/Max EUF (%) $q_s^{\min}/q_s^{\max}$
153	104	R1	200	127	141	1				
154	104	R1	200	239	253	1				
155	105	R2	200	36	71	25		28	0	8/100
156	105	R2	200	141	155	1				
157	105	R2	200	232	246	1				

5.2.1 Generation cost LP for the 157-unit Eskom case study

Eskom's generation mix consists of sixteen coal-fired power stations, one nuclear station, four gas-turbine stations, two conventional hydroelectric stations and three pumped storage schemes. In addition, Eskom imports electricity from a power station in Mozambique. The utility also relies on *independent power producers* (IPPs) to feed power into the South African national transmission grid. The index values adopted for the various power stations are provided in Table 5.6.

Table 5.6: *Index values for the power stations in the 157-unit Eskom case study.*

h	S_h	Description
1	$\{1\}$	Import
2	$\{2\}$	IPPs
3	$\{3\}$	Nuclear
4	$\{4, 5, \ldots, 14\}$	Coal subset 1
5	$\{15, 16, \ldots, 19\}$	Coal subset 2
6	$\{20\}$	Unmet (virtual)
7	$\{21, 22, 23, 24\}$	Gas-turbine
8	$\{25, 26\}$	Conventional hydroelectric
9	$\{27, 28, 29\}$	Pumped storage schemes
10	$\{30, 31, 32\}$	Pumped storage schemes (pumps)

Referring to Table 5.6 for the numbering, the power stations in subsets S_1, S_2, S_3, S_4 and S_5 are considered base load stations. The power stations in subsets S_5, S_6, S_7, S_8 and S_9 are considered peak demand stations. Each base load station has a minimum and maximum daily production requirement associated with it while the two conventional hydroelectric stations each have an exact daily production requirement shown in Table 5.7.

Given this information, the objective in the special case of the general LP generation cost model of §3.3 to be solved in the context of the Eskom case study is to

$$\text{minimise} \quad \sum_{t\in T}\sum_{s\in S} C_s Z_{st} \tag{5.2a}$$

subject to

$$Z_{st} \leq A_{st}, \qquad s \in S, \quad t \in T, \tag{5.2b}$$

$$\sum_{s\in S/=S_{10}} Z_{st} - \sum_{s\in S_{10}} Z_{st} = L_t, \qquad t\in T, \tag{5.2c}$$

$$\sum_{t\in T} Z_{st} \leq {}_{s}, \qquad s\in S_1\cup S_2\cup S_3\cup S_4\cup S_5, \tag{5.2d}$$

$$\sum_{t\in T} Z_{st} \geq f_s^{\min}, \qquad s\in S_1\cup S_2\cup S_3\cup S_4, \tag{5.2e}$$

$$Z_{st} \geq \frac{1}{2} f_s^{\min}, \qquad s\in S_5,\quad t\in T, \tag{5.2f}$$

$$\sum_{t\in T} Z_{st} = f_s^{\mathrm{req}}, \qquad s\in S_8, \tag{5.2g}$$

$$\sum_{t\in T} Z_{at} \leq B_a \sum_{t\in T} Z_{bt}, \qquad a\in S_9,\quad b\in S_{10}, \tag{5.2h}$$

$$Z_{s1} \leq 0, \qquad s\in S_{10}, \tag{5.2i}$$

$$Z_{st} \geq 0, \qquad s\in S,\quad t\in T, \tag{5.2j}$$

where the various symbols have the same meanings as before. Constraint set (5.2b) ensures that each power station's planned energy production during each time slice does not exceed its available capacity during that time slice, while constraint set (5.2c) balances the planned energy and forecast demand during each time slice. The maximum production requirements for the base load stations in $S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$ are ensured by constraint set (5.2d), while the minimum production requirements for stations in $S_1 \cup S_2 \cup S_3 \cup S_4$ are ensured by constraint set (5.2e). For the stations in S_5, at least half of the minimum daily production must occur during the peak demand period and at least half must occur during the offpeak demand period. These additional requirements, which are ensured by constraint set (5.2f), were incorporated into the model in an attempt to keep these five power stations online during the night. The daily production requirements for the conventional hydroelectric stations are ensured by constraint set (5.2g). The load balance constraints for the three pumped storage schemes are enforced by constraint set (5.2h). The efficiency values B_{27}, B_{28}, and B_{29} of the three pumped storage scheme stations are 0.72, 0.75, and 0.745, respectively. Constraint set (5.2i) ensures that the pumps are not used during peak demand periods, while constraint set (5.2j) is the usual sign restriction constraint [25].

Table 5.7: *The daily hydro energy available for the 157-unit Eskom case study.*

Month	$s = 25$ (MWh)	$s = 26$ (MWh)
Jan	1 293	2 846
Feb	1 681	2 393
Mar	2 199	2 264
Apr	2 587	1 229
May	2 393	970
Jun	1 552	1 035
Jul	1 035	1 358
Aug	1 358	1 876
Sep	1 811	2 458
Oct	1 681	2 782
Nov	1 164	2 846
Dec	1 035	2 976

Table 5.8: *Daily peak demand data for the 157-unit Eskom Case study [168].*

Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)	Day	Demand (MW)
1	31 252	62	28 546	123	27 505	184	34 008	245	34 179	306	29 321
2	30 890	63	31 000	124	27 454	185	34 197	246	34 333	307	29 839
3	31 962	64	31 062	125	27 056	186	34 043	247	34 187	308	32 542
4	31 704	65	30 857	126	28 862	187	32 829	248	34 625	309	32 344
5	29 997	66	30 594	127	31 694	188	32 798	249	32 802	310	32 173
6	29 114	67	31 061	128	31 554	189	34 902	250	31 153	311	31 974
7	31 389	68	29 416	129	32 251	190	34 722	251	30 232	312	31 424
8	31 116	69	29 007	130	31 283	191	35 453	252	32 849	313	30 186
9	30 684	70	32 158	131	29 811	192	35 352	253	35 701	314	30 414
10	30 558	71	31 917	132	29 779	193	33 711	254	35 199	315	32 626
11	30 390	72	31 747	133	32 324	194	33 140	255	35 326	316	32 706
12	29 515	73	31 780	134	32 555	195	32 558	256	33 948	317	33 061
13	28 311	74	31 820	135	32 404	196	35 203	257	32 314	318	32 610
14	30 548	75	30 198	136	32 907	197	35 841	258	32 033	319	31 614
15	29 778	76	29 399	137	32 138	198	35 034	259	33 618	320	30 730
16	28 690	77	32 039	138	30 351	199	35 654	260	33 651	321	31 031
17	29 537	78	31 650	139	30 164	200	35 621	261	33 605	322	33 101
18	29 835	79	31 440	140	33 279	201	34 464	262	33 359	323	32 462
19	28 380	80	31 722	141	32 842	202	33 522	263	32 557	324	32 416
20	27 703	81	31 073	142	33 674	203	36 036	264	30 996	325	31 818
21	29 374	82	30 108	143	33 133	204	35 806	265	30 715	326	31 919
22	27 893	83	29 823	144	32 864	205	35 636	266	33 649	327	30 761
23	26 775	84	31 917	145	31 830	206	35 799	267	33 884	328	29 970
24	25 588	85	32 275	146	30 963	207	34 406	268	33 102	329	32 701
25	24 438	86	31 295	147	32 722	208	32 743	269	32 686	330	33 105
26	24 992	87	31 249	148	32 455	209	32 834	270	31 114	331	32 253
27	25 364	88	30 709	149	33 312	210	35 522	271	29 267	332	32 076
28	27 468	89	29 318	150	33 114	211	35 457	272	29 380	333	31 155
29	27 609	90	29 275	151	31 759	212	35 334	273	31 994	334	29 880
30	27 674	91	32 305	152	30 672	213	35 478	274	31 586	335	29 126
31	27 249	92	31 975	153	30 451	214	34 220	275	31 670	336	31 840
32	25 544	93	32 390	154	33 942	215	32 366	276	31 427	337	31 946
33	25 955	94	32 321	155	33 531	216	32 535	277	30 396	338	32 344
34	26 510	95	31 425	156	33 509	217	35 619	278	29 107	339	31 781
35	28 693	96	29 767	157	33 316	218	35 400	279	29 673	340	31 472
36	29 705	97	29 864	158	32 459	219	35 136	280	31 929	341	29 975
37	29 552	98	31 895	159	31 256	220	35 659	281	32 468	342	30 062
38	29 846	99	31 456	160	30 527	221	34 236	282	32 321	343	32 479
39	30 191	100	32 071	161	33 938	222	33 097	283	32 348	344	32 070
40	28 515	101	31 894	162	33 627	223	32 942	284	31 243	345	32 521
41	28 397	102	31 483	163	34 135	224	36 463	285	30 344	346	31 878
42	30 494	103	30 314	164	33 903	225	36 559	286	30 245	347	30 996
43	31 515	104	30 051	165	32 549	226	36 664	287	32 083	348	29 800
44	31 149	105	32 312	166	31 609	227	36 256	288	32 003	349	29 781
45	31 757	106	32 413	167	31 188	228	36 127	289	32 129	350	32 225
46	31 064	107	31 702	168	33 431	229	34 199	290	32 594	351	32 198
47	29 494	108	32 563	169	33 806	230	33 408	291	31 676	352	32 321
48	28 916	109	31 525	170	33 822	231	35 680	292	30 495	353	32 751
49	30 910	110	30 029	171	33 373	232	35 632	293	30 223	354	31 669
50	31 111	111	28 841	172	33 151	233	36 104	294	32 420	355	30 478
51	31 030	112	31 007	173	31 752	234	35 364	295	32 446	356	29 857
52	31 862	113	32 273	174	31 774	235	33 695	296	32 189	357	31 587
53	31 204	114	32 663	175	33 665	236	32 559	297	30 878	358	31 564
54	29 352	115	32 301	176	33 731	237	32 611	298	29 227	359	32 430
55	29 132	116	31 699	177	33 504	238	35 252	299	28 879	360	32 055
56	31 156	117	29 685	178	33 543	239	34 638	300	28 932	361	31 912
57	30 873	118	29 318	179	32 954	240	34 811	301	32 250	362	29 682
58	30 899	119	32 605	180	31 560	241	34 183	302	32 562	363	29 684
59	30 338	120	31 914	181	32 974	242	33 005	303	31 798	364	31 908
60	30 577	121	31 675	182	34 836	243	31 906	304	31 785	365	31 798
61	29 170	122	30 420	183	34 173	244	31 701	305	30 952		

The production cost rates C_S (R/MWh) and the EUF values for the power stations ($q_S^{\min}$ and $q_S^{\max}$) were made available by the existing architecture of the EFS's production planning in 2014 [195]. As in the 32-unit IEEE-RTS case study, the UCLF and OCLF values were set to zero, so as to only analyse how the PCLF values (which depend solely on the GMS decision variables) affect the production cost.

5.2.2 Hourly load data

The hourly energy demand data for the annual planning horizon is presented on the compact disc described in Appendix B. Table 5.8 contains the daily peak load demand (from the hourly data) of the power system. These data were, in fact, represented in Figure 3.1(a).

5.3 Chapter summary

The data pertaining to two case studies carried out later in this dissertation were described in this chapter. These case studies are referred to as the 32-unit IEEE-RTS inspired case study and the 157-unit Eskom case study. All the parameters and data required to specify these GMS problem instances were presented in tabular form. The GMS model proposed in Chapter 3 is solved in Chapters 6 and 7 in the contexts of the case studies reviewed here.

CHAPTER 6

Algorithmic parameter evaluation

Contents

This chapter is devoted to determining good parameter values for the DMOSA algorithm described in Chapter 4. These values are determined more extensively for the 32-unit IEEE-RTS inspired case study than for the much larger 157-unit Eskom case study. In the latter case, the focus is rather to determine appropriate epoch lengths and, in relation, extending the proposed stopping criterion for the DMOSA algorithm.

6.1 Algorithmic parameters

The parameters that may be adjusted (fine tuned) for the DMOSA algorithm described in Chapter 4, include the epoch lengths (over how many iterations the temperature is kept constant), the cooling parameters, namely the geometric decrease rate and increase magnitude of temperature (α and β), the initial acceptance ratio (τ), and the multiplicative constraint violation severity (γ).

As discussed in some detail in §4.3.5, suggestions in the literature as to good α-values range from 0.80 to 0.99 [46, 60, 82, 61, p. 45] with β-values being similar [2, 28] or being selected slightly lower between 0.25 and 0.67 [13, 192] for faster reheating than cooling [2].

Suggestions in the literature for good τ-values range from 0.2 to 0.5 [179, 61, p. 45] whilst in [30] it is suggested that τ be taken as 0.8. In the relatively new multiplicative penalty function developed in [171], the penalty factor severity γ was set to 1. Based on these suggestions, three (low, medium and high) values are considered in this dissertation for each of the four SA parameters, as shown in Table 6.1. Thus 3^4 = 81 cases are considered with a view to find a combination of parameter values that perform best. In each of these combinations, sixteen runs of the DMOSA algorithm are executed, resulting in $81 \times 16 = 1\,296$ non-dominated fronts.

The performance of the different parameter combinations are measured according to the *hypervolume indicator* H described in §4.1.1. In order to determine H for the non-dominated front returned by each run of the DMOSA algorithm, the objectives are normalised (so that they

Table 6.1: *Parameter values considered in the algorithmic parameter evaluation experiment.*

Parameter	Low	Medium	High	Comments
α	0.85	0.90	0.95	Geometric decrease (cooling)
β	0.55	0.75	0.95	Geometric increase (reheating)
τ	0.50	0.65	0.80	Initial rate of acceptance
γ	0.50	1.00	1.50	Multiplicative constraint violation severity

achieve values between 0 and 1). In addition, an adequate reference point had to be chosen. In this case, this is trivial [203], since the reference point chosen should not allow certain objectives to contribute more than others towards the hypervolume measures. A rule-of-thumb for the choice of a good reference point is to take the "corner point" of the objective space (*i.e.* either the bounds of the objective functions being maximised or minimised) [203]. All hypervolume indicators are calculated using the *dominatedHypervolume* function in the *mco* (abbreviation for *multiple criteria optimization*) package [140] in R.

6.2 Parameter evaluation for the 32-unit IEEE-RTS case study

In the case of the 32-unit IEEE-RTS case study, no strong trade-off values could be found between the two objectives proposed for the GMS model of Chapter 3, as illustrated in Figure 6.1(a). This means that the schedule maximising the reliability value (minimising the SSR) determined in the SO study in [169] (× in Figure 6.1) would also produce minimum or very close to minimum fuel cost schedules.

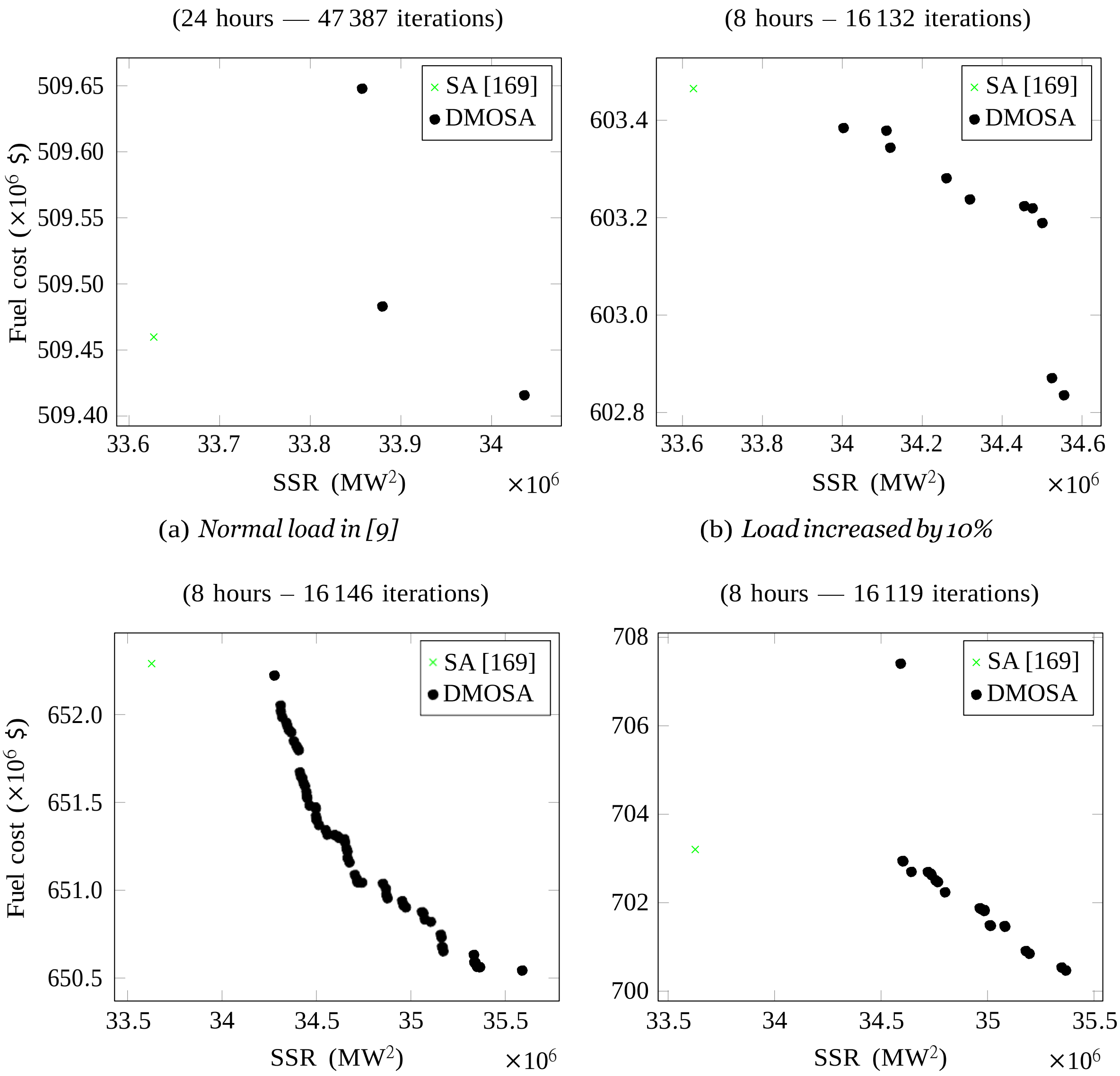
This may be attributed to the fact that there is sufficient reserve capacity (even with large amounts of maintenance) to satisfy the load demand at minimal fuel cost, and therefore the amount of energy production planned for the more expensive power stations (if any) are not significant enough to cause large fuel cost differences. It is, however, interesting that when the load is increased[1] more trade-offs are possible between schedules attempting to minimise the SSR (MW^2) and the fuel cost ($), as illustrated in Figure 6.1(c).

The IEEE-RTS's load was increased by 15% in this dissertation in order to facilitate a comparison of results with those of Schlünz *et al.* [168, 169], who adopted a safety margin of S = 15% in (3.4). This increase ensures that the maintenance solutions proposed in this dissertation also still satisfy the demand constraint in [169].

For the 32-unit IEEE-RTS case study, the upper and lower bounds for normalising the objective function values and determining a good reference point for hypervolume calculations were estimated by evaluating all the non-dominated fronts found for the parameter tests (81×16=1 296 runs). These bounds are illustrated in Table 6.7. With the objective functions normalised in respect of these bounds (see Table 6.7), the reference point was chosen slightly larger at (1.05, 1.05).

Experiments were initially conducted to determine a good epoch length criterion (*i.e.* for how many iterations the SA search should remain in a temperature stage), assuming the medium values in Table 6.1 for the other parameters. These values were considered in four multiples of the suggested scheme in [61, p. 45] which states that inner Metropolis loop should terminate when one of the following two conditions is met: a maximum of $12N$ solutions are accepted, or

[1]Importantly, the peak demand remains the same for similar SSR value comparisons.



(a) *Normal load in [9]* (b) *Load increased by 10%*

(c) *Load increased by 15%* (d) *Load increased by 20%*

Figure 6.1: *Non-dominated fronts found in respect of the IEEE-RTS case study for the different load increases. The increase in loads are only applicable to the ED LP model (the second objective). The peak demand used to calculate the SSR value remains the same so as to be able to compare results with those reported in [169].*

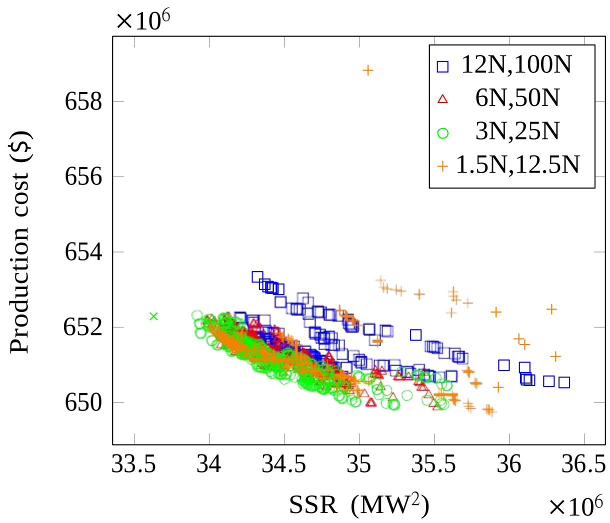
a maximum of $100N$ solutions are attempted. Here N is a measure of the number of degrees of freedom in the optimisation problem — in this case $N = n$ (the number of units). In order to measure the performance of each of these four epoch length stopping criteria, eight runs of the DMOSA were executed. The non-dominated front hypervolume averages and standard deviations achieved by employing these four epoch length stopping criteria are shown in Figures 6.2(a) and 6.2(c). Figures 6.2(b) and 6.2(d) contain similar graphs and values for the case where the (combined) attainment front is considered. As may be seen in the figures, the $3N$, $25N$ stopping criterion performed the best, on average, for the eight runs tested. These values are therefore adopted in the remainder of this dissertation when applying the DMOSA algorithm to the 32-unit IEEE-RTS inspired case study. These values also produced the best (combined)

Table 6.2: *Objective function upper and lower bounds for the 32-unit IEEE-RTS inspired case study (with a 15% increased load). These bounds are used to normalise the objectives and to determine an adequate hypervolume reference point.*

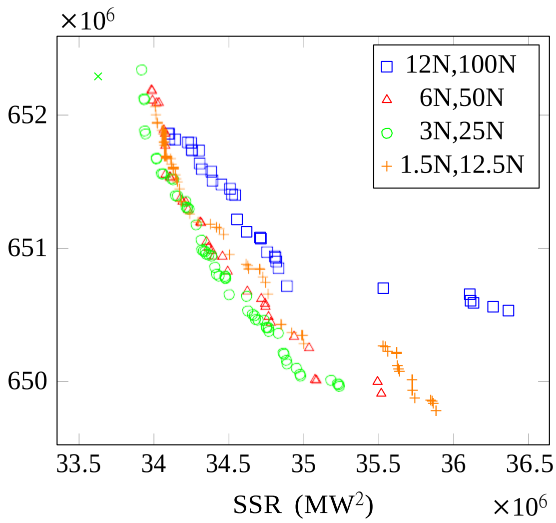
Objective	Lower Bound	Upper Bound
SSR (MW^2)	33 627 292[1]	37 354 828
Production Cost ($)	649 516 204	664 202 101

[1] Best solution found for SO study in [168, 169]

attainment front, as is evident in Figure 6.2(b). This attainment is beneficial in the paradigm of parallel computing, which is ideally suited to (MO) metaheuristics. These benefits are explored further in the next chapter.



(a) *Non-dominated fronts for all 8 runs*



(b) *Non-dominated attainment front from 8 runs*

Epoch lengths sotpping criteria	Ave. H	SD H
12N,100N	0.773	0.077
6N,50N	0.901	0.026
3N,25N	0.920	0.033
1.5N,12.5N	0.753	0.188

(c) *Non-dominated fronts for all 8 runs*

Epoch lengths stopping criteria	H	Plot
12N,100N	0.890	D
6N,50N	0.955	D.
3N,25N	0.972	O
1.5N,12.5N	0.949	+

(d) *Attainment front from 8 runs*

Figure 6.2: *Results pertaining to different epoch length stopping criteria for the 32-unit IEEE-RTS inspired case study.*

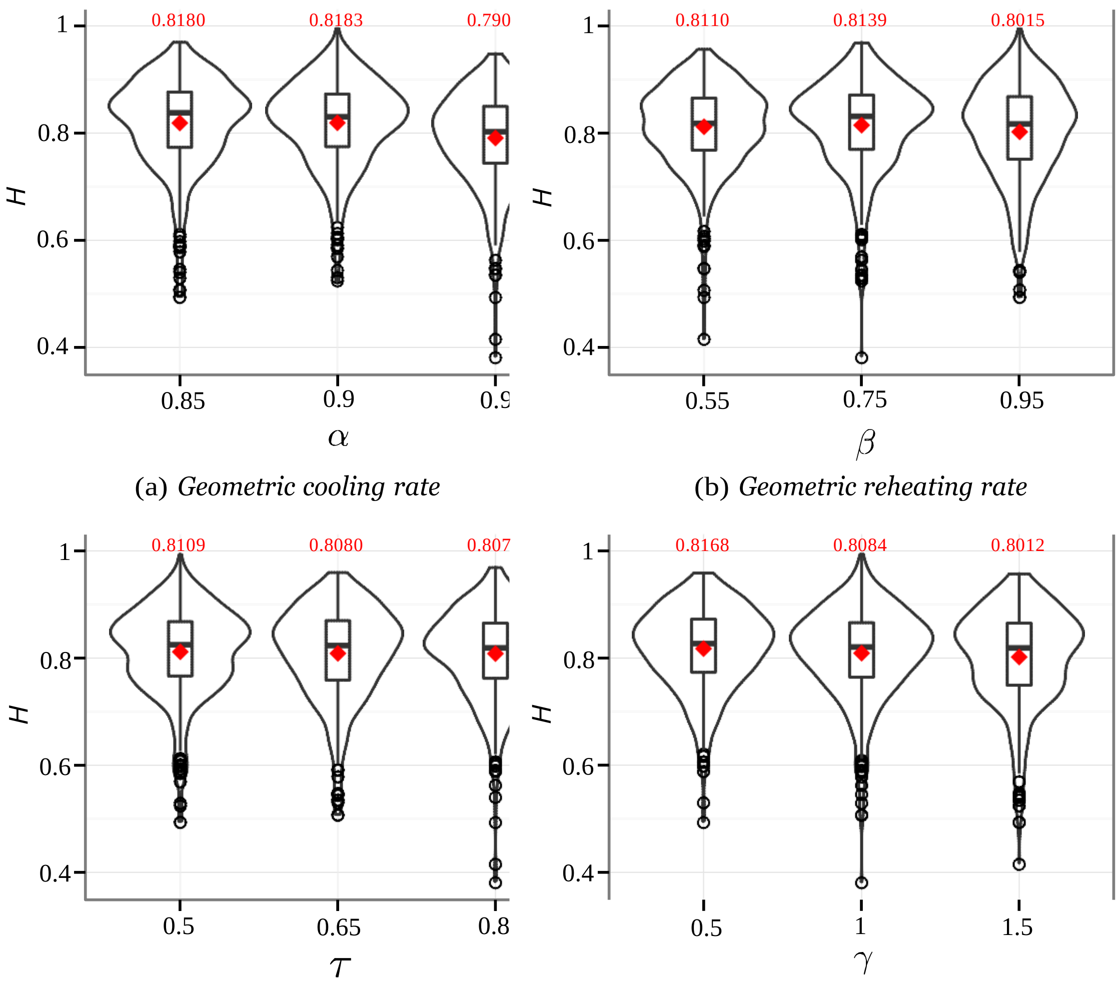
Adopting this $3N$, $25N$ epoch length stopping criterion, sixteen runs of the DMOSA algorithm were subsequently executed for each of the 81 combination of parameters (shown in Table 6.1). The two best objective function pairs (according to average and maximum hypervolume values) are presented in Table 6.3.

Table 6.4 summarises all the average hypervolume results for the 81 different parameter combinations, along with the average numbers of solutions in the final non-dominated front and average numbers of iterations required to achieve the three parameter values (low, medium, and

Table 6.3: *The best average and absolute maximum results found for the 32-unit IEEE-RTS inspired case study over all 81 different parameter combinations in Table 6.1. Sixteen runs were performed for each parameter combination.*

Measurement	H	α	β	τ	γ
Average	**0.861**	0.85	0.75	0.65	0.5
Maximum	0.995	0.90	0.95	0.5	1

high). These results are conditionally shaded in the table and the results are sorted in order of non-decreasing average hypervolume values for each of the 81 different parameter combinations. Figure 6.3 summarises all the hypervolume results thus obtained. This is represented statistically in the form of a box and whisker plot with outliers. The mean is denoted by ◆ in the Figure 6.3, with the corresponding value presented at the top of graph. In addition, the variance is represented by means of a *violin* trace. Each box and violin plot in Figure 6.3 is based on 432 (27 × 16) hypervolume values.



(a) *Geometric cooling rate* (b) *Geometric reheating rate*

(c) *Initial acceptance ratio* (d) *Constraint penalty severity*

Figure 6.3: *The distribution of hypervolume indicators for the 32-unit IEEE-RTS inspired case study, represented as box and violin plots, as well as the mean for each of the four different parameters in respect of their low, medium and high values.*

Table 6.4: *Parameter test results for the 32-unit IEEE-RTS inspired case study, including the average (Ave.) hypervolume H, the average number of solutions in the final non-dominated front |F|, the average number iterations (Iter.) and the corresponding H ranking. These values are sorted according to non-decreasing average hypervolume values.*

Ave. H	α	β	τ	γ	Ave. $\|F\|$	Ave. Iter.	H Rank	H	α	β	τ	γ	Ave. $\|F\|$	Ave. Iter.	H Rank
0.861	0.85	0.75	0.65	0.50	34	12 243	1	0.812	0.95	0.75	0.50	1.50	32	10 801	42
0.858	0.85	0.55	0.50	0.50	43	12 723	2	0.811	0.85	0.75	0.50	1.00	37	11 570	43
0.857	0.90	0.55	0.65	0.50	33	12 936	3	0.811	0.95	0.95	0.65	1.00	41	12 826	44
0.855	0.85	0.75	0.80	0.50	23	12 274	4	0.810	0.90	0.95	0.50	0.50	30	11 106	45
0.853	0.85	0.55	0.50	1.00	32	12 970	5	0.809	0.85	0.75	0.80	1.50	34	10 048	46
0.853	0.90	0.95	0.80	0.50	36	12 281	6	0.808	0.95	0.75	0.50	1.00	39	12 679	47
0.852	0.85	0.55	0.50	1.50	28	12 727	7	0.807	0.95	0.55	0.50	1.50	35	13 086	48
0.848	0.90	0.75	0.80	1.00	39	12 899	8	0.806	0.90	0.95	0.50	1.00	32	10 223	49
0.848	0.85	0.55	0.80	1.50	34	12 613	9	0.806	0.90	0.75	0.65	1.00	35	12 753	50
0.847	0.90	0.95	0.65	1.00	30	11 816	10	0.804	0.85	0.95	0.80	1.50	24	8 670	51
0.839	0.90	0.75	0.80	1.50	34	12 602	11	0.803	0.85	0.75	0.50	1.50	34	11 416	52
0.839	0.90	0.55	0.65	1.00	31	12 909	12	0.802	0.95	0.95	0.50	0.50	30	11 769	53
0.834	0.90	0.55	0.50	1.50	24	13 051	13	0.801	0.95	0.55	0.50	0.50	36	12 890	54
0.833	0.85	0.95	0.50	0.50	35	9 020	14	0.800	0.90	0.95	0.80	1.50	40	11 654	55
0.832	0.85	0.75	0.65	1.50	31	12 487	15	0.799	0.85	0.55	0.80	1.00	37	10 371	56
0.831	0.85	0.55	0.80	0.50	25	11 853	16	0.798	0.85	0.95	0.65	0.50	30	6 541	57
0.830	0.85	0.55	0.65	1.50	35	13 035	17	0.798	0.90	0.75	0.65	0.50	27	11 991	58
0.829	0.90	0.75	0.80	0.50	34	12 652	18	0.797	0.95	0.75	0.80	0.50	34	12 836	59
0.828	0.95	0.75	0.65	0.50	25	12 927	19	0.795	0.85	0.95	0.65	1.00	32	9 029	60
0.827	0.90	0.75	0.50	0.50	31	11 992	20	0.794	0.85	0.95	0.65	1.50	34	7 961	61
0.827	0.95	0.95	0.65	0.50	40	12 982	21	0.790	0.95	0.55	0.65	1.00	37	12 850	62
0.827	0.85	0.95	0.80	1.00	23	10 442	22	0.790	0.95	0.75	0.80	1.50	34	12 935	63
0.826	0.90	0.55	0.80	0.50	30	12 937	23	0.788	0.90	0.75	0.50	1.50	35	11 834	64
0.826	0.95	0.75	0.50	0.50	35	12 787	24	0.788	0.85	0.95	0.50	1.00	30	8 038	65
0.826	0.85	0.55	0.65	1.00	30	12 526	25	0.785	0.95	0.75	0.65	1.00	31	12 782	66
0.826	0.85	0.55	0.65	0.50	39	12 901	26	0.782	0.95	0.55	0.80	1.00	38	12 998	67
0.825	0.85	0.75	0.50	0.50	30	12 676	27	0.777	0.90	0.95	0.50	1.50	38	10 869	68
0.824	0.90	0.55	0.80	1.50	22	13 055	28	0.777	0.95	0.95	0.80	0.50	28	12 838	69
0.823	0.90	0.75	0.65	1.50	26	12 286	29	0.775	0.85	0.95	0.80	0.50	24	8 032	70
0.822	0.90	0.55	0.50	1.00	35	12 756	30	0.774	0.95	0.95	0.50	1.50	27	12 702	71
0.822	0.90	0.95	0.65	0.50	26	11 086	31	0.774	0.85	0.75	0.65	1.00	30	12 926	72
0.821	0.90	0.55	0.65	1.50	30	13 089	32	0.769	0.95	0.55	0.65	0.50	24	12 070	73
0.818	0.95	0.75	0.65	1.50	21	13 078	33	0.768	0.95	0.55	0.50	1.00	34	12 911	74
0.816	0.90	0.75	0.50	1.00	24	12 314	34	0.765	0.85	0.95	0.50	1.50	34	8 878	75
0.815	0.95	0.95	0.80	1.00	32	11 307	35	0.757	0.95	0.75	0.80	1.00	21	12 980	76
0.815	0.95	0.95	0.80	1.50	29	12 913	36	0.754	0.95	0.95	0.65	1.50	32	10 806	77
0.815	0.90	0.95	0.80	1.00	36	11 216	37	0.743	0.95	0.55	0.65	1.50	34	13 008	78
0.814	0.90	0.55	0.50	0.50	36	12 470	38	0.742	0.90	0.95	0.65	1.50	30	8 878	79
0.814	0.95	0.95	0.50	1.00	36	12 242	39	0.735	0.95	0.55	0.80	1.50	33	12 893	80
0.812	0.85	0.75	0.80	1.00	25	12 219	40	0.728	0.95	0.55	0.80	0.50	34	13 098	81
0.812	0.90	0.55	0.80	1.00	31	12 879	41								

The only clear trend in Table 6.4 is that a slower cooling rate α (*e.g.* 0.95) produces a smaller hypervolume value. This is also evident in Figure 6.3(a). This observation is attributed to the algorithm spending more time exploring whilst at the higher temperatures. The observation is also because a timeout (of 8 hours) is enforced on the computation time expended by the algorithm — if this were to be increased, a different result may possibly have been achieved. This is evident in the literature related to SA, which states that there is a trade-off between solution quality and computing time expended (see Table 4.1, for example). As may be seen in the three other graphs in Figure 6.3, there is no clearly discernible trend for the other three parameters in relation to the hypervolumes attained. Naturally, the combination of the parameter values is, however, important as evident in Table 6.4. For example, the two worst performing parameters in terms of the hypervolume indicator (the last two rows in Table 6.4) are achieved when the SA algorithm cools slowly ($\alpha = 0.95$) and has a large temperature increase ($\beta = 0.55$), attributed to the algorithm spending more time exploring at the higher temperatures, as explained above.

A slight trend is visible in Table 6.4 for the constraint violation severity factor γ, where a lower value of 0.50 seems to perform well (the first rows in Table 6.4) in terms of the hypervolume indicator, while a higher constraint penalty violation severity value of 1.50 does not perform so well in this respect. This is attributed to the objective functions being penalised too severely and the algorithm not being allowed to search through the infeasible regions of the solution space in order to reach high-quality feasible solutions.

No real clear trend is visible in Table 6.4 in respect of the influence that the initial acceptance ratio τ has on the average hypervolume H.

Table 6.5 summarises the average number of iterations (accepted and attempted moves) required in each case by the algorithm, along with the average number of reheats and the average number of consecutive reheats, as well as the performance ranking according to the average hypervolume in Table 6.4. These results are sorted in non-descending order according to the average number of iterations required for each of the 81 different parameter combinations. Naturally, the larger the average number of reheats and specifically the larger the average number of consecutive reheats, the smaller the average number of iterations the algorithm requires, since the algorithm invariably terminates before the eight-hour computation time budget $t_{\max}$ has elapsed, as is evident in Table 6.5.

As may be seen in the first rows of Table 6.5 (for fast cooling rates of $\alpha = 0.85$ and small incremental reheating values of $\beta = 0.95$), the algorithm quickly decreases the temperature and with small increases in temperature (reheating by $1/(\beta = 0.95)$) the algorithm therefore terminates due to the occurrence of three consecutive reheatings ($\Omega_{\text{frozen}} = 3$) before the eight-hour computation time budget $t_{\max}$ has elapsed (hence the average number of iterations is much smaller). Reheating occurs when a maximum allowable number of solutions have been attempted ($25N$ in this case). The reason why the algorithm reheats more frequently when assuming these parameter values ($\alpha = 0.95$ and $\beta = 0.95$) is that at lower temperatures the probability of accepting a worsening solution is smaller and hence the algorithm less frequently accepts neighbouring solutions. The opposite response is observed for the reheating parameter, as may be seen the last rows of Table 6.5, where a large increase in temperature (reheating by $1/(\beta = 0.55)$) means that the algorithm explores more again in the next epoch and hence again starts accepting more neighbouring solutions. The algorithm typically terminates after eight hours, and so the number of iterations required is larger.

No clear trend is visible in Table 6.5 with respect to the influence that the initial acceptance ratio τ has on the number of iterations required by the algorithm before termination (resulting in a larger number of search iterations).

Table 6.5: *Parameter test results for the 32-unit IEEE-RTS inspired case study, including the average number of iterations (Ave. Iter.), the average number of times reheating occured (Reh.) and the average number of consecutive reheatings that occurred (CR) across the sixteen runs. The results are sorted according to non-decreasing average numbers of iterations.*

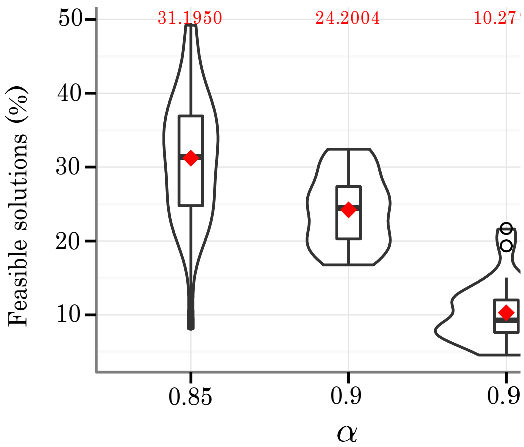
Ave. Iter.	α	β	τ	γ	Ave. Reh.	Ave. CR	H Rank
6 541	0.85	0.95	0.65	0.50	3.2	3.0	57
7 961	0.85	0.95	0.65	1.50	3.9	3.0	61
8 032	0.85	0.95	0.80	0.50	4.2	2.9	70
8 038	0.85	0.95	0.50	1.00	4.2	2.9	65
8 670	0.85	0.95	0.80	1.50	3.9	3.0	51
8 878	0.85	0.95	0.50	1.50	4.8	3.0	75
8 878	0.90	0.95	0.65	1.50	3.7	2.8	79
9 020	0.85	0.95	0.50	0.50	5.0	2.9	14
9 029	0.85	0.95	0.65	1.00	4.7	3.0	60
10 048	0.85	0.75	0.80	1.50	4.2	2.6	46
10 223	0.90	0.95	0.50	1.00	4.4	2.9	49
10 371	0.85	0.55	0.80	1.00	3.2	1.6	56
10 442	0.85	0.95	0.80	1.00	5.1	3.0	22
10 801	0.95	0.75	0.50	1.50	2.9	2.3	42
10 806	0.95	0.95	0.65	1.50	2.1	1.9	77
10 869	0.90	0.95	0.50	1.50	4.1	2.6	68
11 086	0.90	0.95	0.65	0.50	4.5	2.9	31
11 106	0.90	0.95	0.50	0.50	4.3	2.6	45
11 216	0.90	0.95	0.80	1.00	3.9	2.8	37
11 307	0.95	0.95	0.80	1.00	4.4	2.8	35
11 416	0.85	0.75	0.50	1.50	4.8	2.6	52
11 570	0.85	0.75	0.50	1.00	4.8	2.4	43
11 654	0.90	0.95	0.80	1.50	4.2	2.7	55
11 769	0.95	0.95	0.50	0.50	3.8	2.9	53
11 816	0.90	0.95	0.65	1.00	4.0	2.5	10
11 834	0.90	0.75	0.50	1.50	3.8	2.4	64
11 853	0.85	0.55	0.80	0.50	3.6	2.3	16
11 991	0.90	0.75	0.65	0.50	3.9	2.4	58
11 992	0.90	0.75	0.50	0.50	3.8	2.3	20
12 070	0.95	0.55	0.65	0.50	1.6	1.4	73
12 219	0.85	0.75	0.80	1.00	3.9	2.0	40
12 242	0.95	0.95	0.50	1.00	2.9	2.3	39
12 243	0.85	0.75	0.65	0.50	4.6	2.4	1
12 274	0.85	0.75	0.80	0.50	4.2	2.1	4
12 281	0.90	0.95	0.80	0.50	4.2	2.2	6
12 286	0.90	0.75	0.65	1.50	3.5	2.0	29
12 314	0.90	0.75	0.50	1.00	4.0	2.3	34
12 470	0.90	0.55	0.50	0.50	3.2	1.9	38
12 487	0.85	0.75	0.65	1.50	4.6	2.0	15
12 526	0.85	0.55	0.65	1.00	4.2	2.1	25
12 602	0.90	0.75	0.80	1.50	3.6	2.2	11
12 613	0.85	0.55	0.80	1.50	3.8	2.1	9
12 652	0.90	0.75	0.80	0.50	3.2	1.9	18
12 676	0.85	0.75	0.50	0.50	3.2	1.8	27
12 679	0.95	0.75	0.50	1.00	2.3	2.1	47
12 702	0.95	0.95	0.50	1.50	2.1	1.8	71
12 723	0.85	0.55	0.50	0.50	4.1	1.9	2
12 727	0.85	0.55	0.50	1.50	3.9	2.0	7
12 753	0.90	0.75	0.65	1.00	3.0	1.7	50
12 756	0.90	0.55	0.50	1.00	3.4	1.9	30
12 782	0.95	0.75	0.65	1.00	1.6	1.4	66
12 787	0.95	0.75	0.50	0.50	1.9	1.5	24
12 826	0.95	0.95	0.65	1.00	2.3	1.7	44
12 836	0.95	0.75	0.80	0.50	1.3	1.1	59
12 838	0.95	0.95	0.80	0.50	2.1	1.8	69
12 850	0.95	0.55	0.65	1.00	1.4	1.4	62
12 879	0.90	0.55	0.80	1.00	2.3	1.5	41
12 890	0.95	0.55	0.50	0.50	1.8	1.3	54
12 893	0.95	0.55	0.80	1.50	0.9	0.9	80
12 899	0.90	0.75	0.80	1.00	3.3	1.9	8
12 901	0.85	0.55	0.65	0.50	4.1	2.1	26
12 909	0.90	0.55	0.65	1.00	2.9	1.6	12
12 911	0.95	0.55	0.50	1.00	1.7	1.4	74
12 913	0.95	0.95	0.80	1.50	2.3	2.1	36
12 926	0.85	0.75	0.65	1.00	1.6	1.2	72
12 927	0.95	0.75	0.65	0.50	2.1	1.6	19
12 935	0.95	0.75	0.80	1.50	1.7	1.4	63
12 936	0.90	0.55	0.65	0.50	2.8	1.6	3
12 937	0.90	0.55	0.80	0.50	2.6	1.4	23
12 970	0.85	0.55	0.50	1.00	3.6	1.8	5
12 980	0.95	0.75	0.80	1.00	1.5	1.3	76
12 982	0.95	0.95	0.65	0.50	2.4	2.1	21
12 998	0.95	0.55	0.80	1.00	1.4	1.3	67
13 008	0.95	0.55	0.65	1.50	1.6	1.5	78
13 035	0.85	0.55	0.65	1.50	3.6	1.6	17
13 051	0.90	0.55	0.50	1.50	2.8	1.6	13
13 055	0.90	0.55	0.80	1.50	2.4	1.4	28
13 078	0.95	0.75	0.65	1.50	1.8	1.4	33
13 086	0.95	0.55	0.50	1.50	1.6	1.3	48
13 089	0.90	0.55	0.65	1.50	2.8	1.8	32
13 098	0.95	0.55	0.80	0.50	1.1	1.1	81

Table 6.6: *Parameter test results for the IEEE-RTS case study sorted according to non-decreasing average numbers of feasible solutions accepted by the DMOSA algorithm. The table also contains the average hypervolume (Ave. H) and average hypervolume rank (H Rank).*

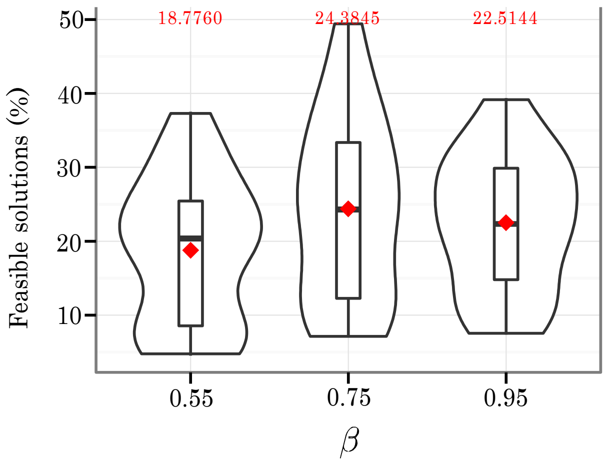
Feas. (%)	α	β	τ	γ	Ave. H	H Rank	Feas. (%)	α	β	τ	γ	Ave. H	H Rank
49.47	0.85	0.75	0.50	0.50	0.825	27	21.67	0.95	0.95	0.80	1.00	0.815	35
46.59	0.85	0.75	0.50	1.00	0.811	43	21.54	0.85	0.95	0.65	0.50	0.798	57
41.78	0.85	0.75	0.50	1.50	0.803	52	21.07	0.85	0.55	0.80	1.00	0.799	56
39.37	0.85	0.95	0.50	0.50	0.833	14	21.05	0.90	0.95	0.80	1.50	0.800	55
38.61	0.85	0.75	0.65	0.50	0.861	1	20.62	0.90	0.55	0.50	1.00	0.822	30
37.54	0.85	0.55	0.50	1.00	0.853	5	20.36	0.90	0.55	0.80	0.50	0.826	23
37.04	0.85	0.95	0.50	1.50	0.765	75	20.20	0.90	0.75	0.80	1.50	0.839	11
36.82	0.85	0.75	0.80	1.00	0.812	40	19.34	0.95	0.75	0.50	0.50	0.826	24
35.80	0.85	0.75	0.65	1.50	0.832	15	19.01	0.90	0.95	0.65	1.50	0.742	79
35.20	0.85	0.75	0.80	0.50	0.855	4	18.90	0.90	0.55	0.80	1.50	0.824	28
34.03	0.85	0.95	0.65	1.00	0.795	60	18.52	0.90	0.95	0.80	1.00	0.815	37
33.50	0.85	0.55	0.50	1.50	0.852	7	17.80	0.90	0.55	0.65	1.50	0.821	32
33.48	0.85	0.55	0.50	0.50	0.858	2	17.77	0.90	0.55	0.50	0.50	0.814	38
32.47	0.90	0.95	0.65	1.00	0.847	10	16.73	0.90	0.55	0.80	1.00	0.812	41
31.52	0.90	0.75	0.50	1.00	0.816	34	15.04	0.95	0.95	0.50	1.50	0.774	71
31.42	0.85	0.95	0.80	1.00	0.827	22	14.54	0.95	0.95	0.65	0.50	0.827	21
30.88	0.90	0.75	0.50	1.50	0.788	64	13.25	0.95	0.75	0.50	1.00	0.807	47
30.74	0.90	0.95	0.50	0.50	0.810	45	12.48	0.95	0.75	0.65	0.50	0.828	19
30.72	0.90	0.75	0.50	0.50	0.827	20	12.05	0.95	0.75	0.50	1.50	0.812	42
30.26	0.85	0.95	0.65	1.50	0.794	61	11.99	0.95	0.75	0.65	1.50	0.818	33
30.01	0.85	0.55	0.65	1.00	0.826	25	11.98	0.95	0.75	0.65	1.00	0.785	66
29.79	0.85	0.55	0.65	1.50	0.830	17	11.96	0.95	0.95	0.65	1.00	0.811	44
29.52	0.90	0.95	0.50	1.00	0.806	49	11.59	0.95	0.95	0.50	1.00	0.814	39
27.79	0.85	0.95	0.50	1.00	0.788	65	11.20	0.95	0.95	0.50	0.50	0.802	53
27.59	0.90	0.95	0.65	0.50	0.822	31	9.33	0.95	0.55	0.50	0.50	0.801	54
27.13	0.90	0.95	0.50	1.50	0.777	68	9.25	0.95	0.55	0.50	1.50	0.807	48
26.15	0.90	0.75	0.65	1.00	0.806	50	8.90	0.95	0.95	0.80	1.50	0.815	36
25.97	0.90	0.75	0.80	1.00	0.848	8	8.07	0.95	0.75	0.80	1.00	0.757	76
25.85	0.90	0.55	0.65	1.00	0.839	12	8.04	0.95	0.75	0.80	1.50	0.790	63
25.55	0.90	0.75	0.80	0.50	0.829	18	7.94	0.95	0.95	0.80	0.50	0.777	69
25.45	0.85	0.55	0.65	0.50	0.826	26	7.86	0.85	0.75	0.65	1.00	0.774	72
25.42	0.85	0.55	0.80	1.50	0.848	9	7.83	0.95	0.55	0.65	1.00	0.790	62
25.00	0.90	0.55	0.50	1.50	0.834	13	7.63	0.95	0.55	0.50	1.00	0.768	74
24.42	0.90	0.95	0.80	0.50	0.853	6	7.61	0.95	0.55	0.65	1.50	0.743	78
24.27	0.90	0.75	0.65	0.50	0.798	58	7.35	0.95	0.95	0.65	1.50	0.754	77
24.13	0.85	0.75	0.80	1.50	0.809	46	7.08	0.95	0.75	0.80	0.50	0.797	59
23.48	0.85	0.95	0.80	1.50	0.804	51	6.09	0.95	0.55	0.80	1.00	0.782	67
22.60	0.90	0.75	0.65	1.50	0.823	29	5.76	0.95	0.55	0.65	0.50	0.769	73
22.48	0.85	0.55	0.80	0.50	0.831	16	5.07	0.95	0.55	0.80	1.50	0.735	80
22.33	0.85	0.95	0.80	0.50	0.775	70	4.51	0.95	0.55	0.80	0.50	0.728	81
22.08	0.90	0.55	0.65	0.50	0.857	3							

For the constraint violation severity factor γ, it may be seen in the last rows of Table 6.5 that a larger value (1.5) causes the algorithm not to reheat as often. The algorithm therefore tends not to terminate before the eight-hour computation time budget t_{max} has elapsed.

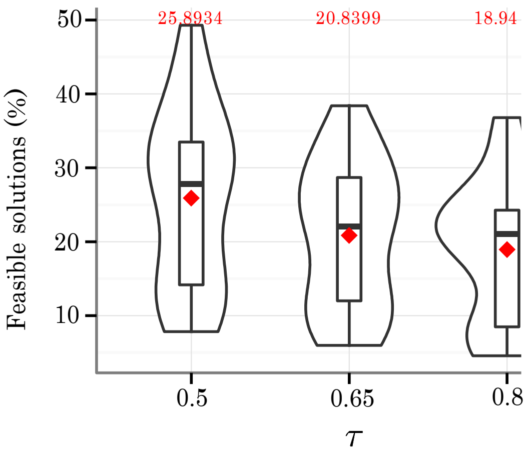
Table 6.6 summarises the total percentage of feasible solutions accepted (all sixteen runs' data combined) along with the average hypervolume and the corresponding rank. The feasibility percentage in the table represents the number of times the DMOSA algorithm found and accepted feasible solutions over the total number of solutions accepted. As may be seen in Table 6.6, a fast geometric cooling rate (α = 0.85) produces a larger feasibility percentage, since the algorithm does not explore as long at high temperatures and therefore accepts fewer infeasible solutions (due to the multiplicative penalty constraint handling technique employed). The converse is also true, as illustrated in the last rows of Table 6.6, namely that a slow geometric cooling rate (α = 0.95) translates into more infeasible solutions being accepted (a low feasibility percentage). This is further evident in Figure 6.4(a) in which it is clear that the reheating parameter β also



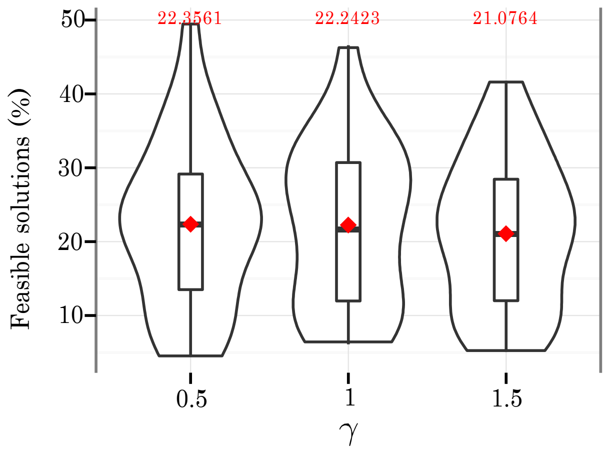
(a) *Geometric cooling rate*



(b) *Geometric reheating rate*



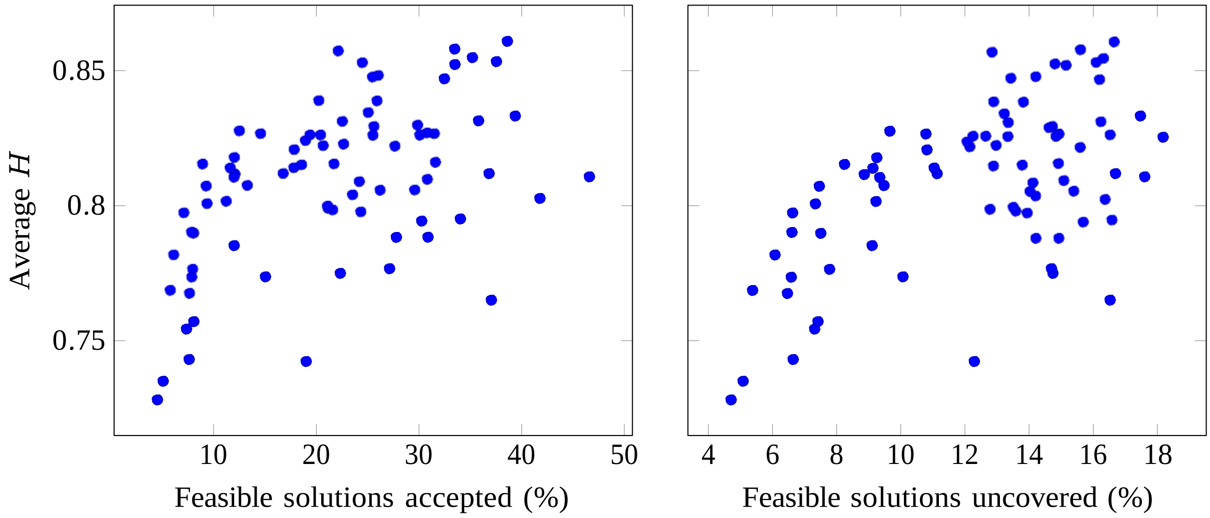
(c) *Initial acceptance ration*



(d) *Constraint penalty severity*

Figure 6.4: *The distribution of feasible solutions accepted in the 32-unit IEEE-RTS case study, represented as box and violin plots, as well as the mean for each of the four different cooling parameters and their low, medium and high settings.*

influences the feasibility percentage (the last rows of Table 6.6, where a reheating parameter value of 0.55 results in the algorithm exploring longer at higher temperatures). Naturally, the



(a)

(b)

Figure 6.5: *Scatter plot for the 32-unit IEEE-RTS inspired case study of the average hypervolume as a function of (a) the percentage of feasible solutions accepted (data from Table 6.6) and (b) the percentage of feasible solutions uncovered (attempted and accepted).*

combination of slow cooling ($\alpha = 0.95$) and fast reheating ($\alpha = 0.55$) means that the algorithm spends a considerable amount of time exploring at high temperatures, resulting in a large number of infeasible solutions being accepted (due to the multiplicative penalty constraint handling technique employed) as is evident in the last rows of Table 6.6 and in Figure 6.4(b).

As may further be seen in Table 6.6 and Figure 6.4(c), the initial acceptance ratio τ affects the number of feasible solutions accepted. This is intuitive, since at higher values ($\tau = 0.8$) the algorithm begins probabilistically accepting worsening and probably more infeasible solutions (due to the multiplicative penalty constraint handling technique employed). At lower values ($\tau = 0.5$), on the other hand, the algorithm initially does not accept as many worsening solutions.

For the multiplicative constraint penalty severity factor γ, no clear trend was found in respect of the influence on the percentage of feasible solutions accepted, as shown in Table 6.6 and Figure 6.4(d). A larger value ($\gamma = 1.5$), however, on average produces a smaller percentage of feasible solutions accepted.

There is a correlation between the number of feasible solutions accepted (column 1 in Table 6.6) and the average hypervolume (column 3 in Table 6.6), as illustrated in Figure 6.5(a), in the sense that the more feasible solutions the DMOSA algorithm accepts, the better the performance of the final non-dominated front in terms of hypervolume. This is attributed to the search progressing more quickly towards the feasible space and finding the final non-dominated front quicker. Of further interest is Figure 6.5(b), which illustrates a correlation between the average hypervolume achieved and the feasible solutions uncovered (those accepted and attempted). Naturally, the percentage of feasible solutions uncovered is far lower than the percentage of feasible solutions accepted (see Figure 6.5), since not all the worse solutions (and most probably more infeasible solutions) are accepted. This further demonstrates that the GMS problem is usually a relatively tightly constrained problem.

6.3 Parameter evaluation for the 157-unit Eskom case study

For the 157-unit Eskom case study, the upper and lower bounds employed for normalising the objective function values and for hypervolume reference point selection were estimated by studying all the non-dominated fronts found during the course of the research reported in this dissertation. These bounds are provided in Table 6.7. As in the case of the 32-unit IEEE-RTS inspired case study, the objective functions were normalised with respect to the bounds provided in Table 6.7. The reference point was again chosen slightly larger at (1.05, 1.05).

Table 6.7: *Objective function upper and lower bounds for the 157-unit Eskom case study. These values are used to normalise the objectives and to determine an adequate hypervolume reference point.*

Objective	Lower Bound	Upper Bound
SSR (MW^2)	11 101 712 702[1]	11 442 963 630
Production Cost (R)	26 654 977 043	26 678 744 511

[1] Best solution found for the SO study in [168, 170]

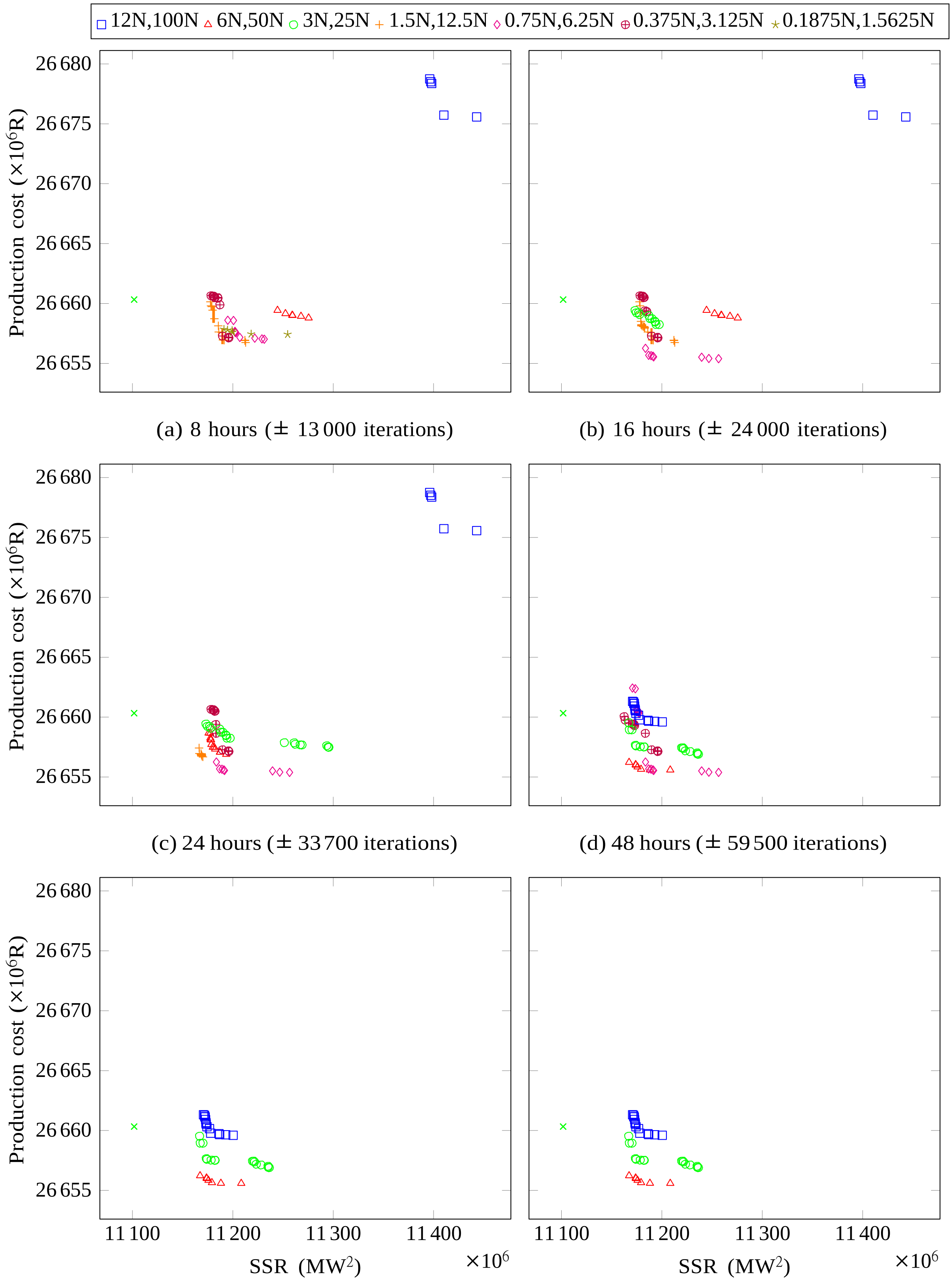
The 157-unit Eskom case study problem instance is considerably larger than the 32-unit IEEE-RTS inspired case study (a 157 unit × 365 day schedule *versus* a 32 unit × 52 week schedule). It therefore takes considerably longer to find non-dominated fronts of reasonable quality. It was found empirically that the epoch length criterion was more influential in respect of the performance of the DMOSA algorithm than the other four SA parameters (α, β, τ and γ). A less extensive parameter evaluation was subsequently performed in respect of the Eskom case study in the sense that the best α, β, τ and γ values uncovered in the context of the IEEE-RTS inspired case study (see Table 6.3) were employed in order to find a suitable epoch length stopping criterion. In addition, due to the larger problem instance size, the amount of computation time allowed (8 hours in the 32-unit IEEE-RTS inspired case study) naturally had an influence on the performance of the epoch length criteria, as is evident in Figure 6.6. Table 6.8 contains the hypervolume values corresponding to the non-dominated fronts of Figure 6.6.

Table 6.8: *Hypervolumes of the non-dominated fronts for the 157-unit Eskom case study (where $N = n = 157$), according to different epoch length criteria over different computation time budgets, as illustrated in Figure 6.6.*

	D 12N,100N	D. 6N,50N	O 3N,25N	+ 1.5N,12.5N	0.75N,6.25N	⊕ 0.375N,3.125N	* 0.1875N,1.5265N
8	0.029	0.560	*	**0.804**	0.746	0.787	0.745
16	0.029	0.560	0.765	0.805	**0.835**	0.788	—
24	0.029	0.806	0.783	**0.842**	0.835	0.788	—
48	0.724	**0.878**	0.826	—	0.865	0.829	—
56	0.724	**0.878**	0.826	—	—	—	—
64	0.724	**0.878**	0.826	—	—	—	—

* No feasible non-dominated front found
— Algorithm terminated by reheating stopping criteria

As may be seen in Figure 6.6 and Table 6.8, the shorter the epoch length stopping criterion (*e.g.* * in Figure 6.6 and Table 6.8), the faster algorithm converges to a non-dominated front (since it dwells shorter at higher temperatures, thus accepting fewer worsening moves). It also therefore terminates much quicker due to experiencing three consecutive reheatings occurring



(a) 8 hours (± 13 000 iterations)

(b) 16 hours (± 24 000 iterations)

(c) 24 hours (± 33 700 iterations)

(d) 48 hours (± 59 500 iterations)

(e) 56 hours (± 74 000 iterations)

(f) 64 hours (± 78 400 iterations)

Figure 6.6: *Non-dominated fronts uncovered during different epoch duration tests for the 157-unit Eskom case study.*

earlier. This may be viewed as a case of *simulated quenching*, rather than simulated annealing.

Alternatively, the larger the epoch stopping criterion (*e.g.* D in Figure 6.6 and Table 6.8), the longer the algorithm takes to converge to a non-dominated front (see Figure 6.6(a)), spending more time exploring the solution space (accepting worsening moves). As time increases, however, it converges to better non-dominated fronts more slowly, and is expected to overtake the algorithm having shorter epoch length stopping criteria in terms of resulting solution quality as computing time continues. This is evident in Figure 6.6 and Table 6.8, again reconfirming the common notion for SA schedules and, in fact, most metaheuristics that there is a trade-off between solution quality and computing time.

As may be seen in Figure 6.6 and Table 6.8, the epoch length stopping criterion should be altered based on the amount of time available to compute GMS solutions. Another interesting point is that one may instead employ shorter epoch lengths, yet lower reheating values β, or extend the algorithm's run time by increasing the number of consecutive reheatings allowed before termination (currently $\Omega_{\text{frozen}} = 3$) so as to extend the performance of the shorter epoch length stopping criterion over time.

6.4 Chapter summary

The purpose of the chapter was to describe an empirical experiment performed in search of adequate parameters for the DMOSA algorithm (reviewed in Chapter 4) for both case studies discussed in Chapter 5, namely the 32-unit IEEE-RTS inspired case study (a 32×52 schedule) and the 157-unit Eskom case study (a 157×365 schedule). In the former case study, all parameters were extensively analysed. As the decision space is, however, far larger in the latter case study (and the system is more tightly constrained), more focus was placed on determining an adequate epoch length criterion and a suitable stopping criterion.

In the following chapter, numerical results are presented in respect of the two case studies mentioned above, using the best-performing parameter values uncovered in this chapter. Additionally, verification and validation, as well as an analysis, of the results is documented.

CHAPTER 7

Numerical results

Contents

This chapter is devoted to a presentation of the results obtained by the DMOSA algorithm when solving the bi-objective GMS model of Chapter 4 in the context of the two test systems of Chapter 5 and adopting the algorithmic parameter values identified in Chapter 5. The model proposed in Chapter 3 is verified and validated, and an indication is given of the performance of the SA algorithm. A graphical representation of the bi-objective solution space is presented of the 32-unit IEEE-RTS inspired case study. Analyses are also performed of the trade-off effects achieved by generator maintenance schedules aimed at optimising the two model objective separately. Furthermore, a sensitivity analysis is performed in respect of the effects of the GMS model constraints, as well as the effect of considering these constraints as another (composite) objective to be optimised within the MO GMS paradigm. Finally, a number of algorithmic improvements and alterations are tested in an attempt to improve the performance of the DMOSA algorithm.

7.1 Verification and validation of the proposed model

Validation of a model is concerned with how accurately the model represents the real-world process. In layman's terms, the following question is therefore considered during the model validation process: *Is it the right model*? Verification of a model, on the other hand, is concerned with the accuracy of the model and whether it performs as expected; in layman's terms, the following question is therefore considered during the model validation process: *Is the model right*? In studies utilising simulation and algorithmic techniques, verification primarily entails ensuring correct coding. Verification and validation overlap in many cases and hence they are dealt with jointly in this section.

In this section, the energy production planned by the generation cost minimisation LP-module of Brits [25], as described in §3.3, is compared with an energy schedule plan in the literature for the 32-unit IEEE-RTS inspired case study. In addition, the DMOSA algorithm's performance is measured against those of two NSGA-II *off-the-shelf* packages available in R. Further validation of the GMS model of Chapter 3 is considered in the next chapter regarding its implementation and feedback received from Eskom with respect to a DSS in which the model is embedded.

7.1.1 Verification and validation of the ED LP model

The output of the production planning module for the 32-unit IEEE-RTS inspired case study proposed in §5.1.1 was compared with the production plan determined in the IEEE-RTS follow-on paper by Allan *et al.* in 1986 [11], as illustrated in Figure 7.1. Allan *et al.* determined the energy supplied (Table VIII in [11]) by each unit using an energy-based evaluation method described in [11]. This method is based on a merit order (similar to Table 5.1) of the units and the IEEE-RTS's FORs for each unit provided in the 1979 IEEE-RTS [9]. It is assumed that Allan *et al.* also took into account the planned generator maintenance schedule provided in Table XI in [11]. This generator maintenance schedule was determined by Billinton and El-Sheikhi [20] who adopted a levelised risk scheduling criterion. Allan *et al.* [11], furthermore, assumed that there are no capacity limitations associated with the hydro units and no energy limitations (*i.e.* rendering constraint (5.1f) and Table (5.3) inapplicable in the IEEE-RTS). Thus the EUF values for the power stations ($q_s^{\min}$ and $q_s^{\max}$) were set at the extremes, namely 0% and 100%. This maintenance schedule was provided as input to the model proposed in §3.3, and the corresponding planned energy production for each unit was determined, as shown in Figure 7.1.

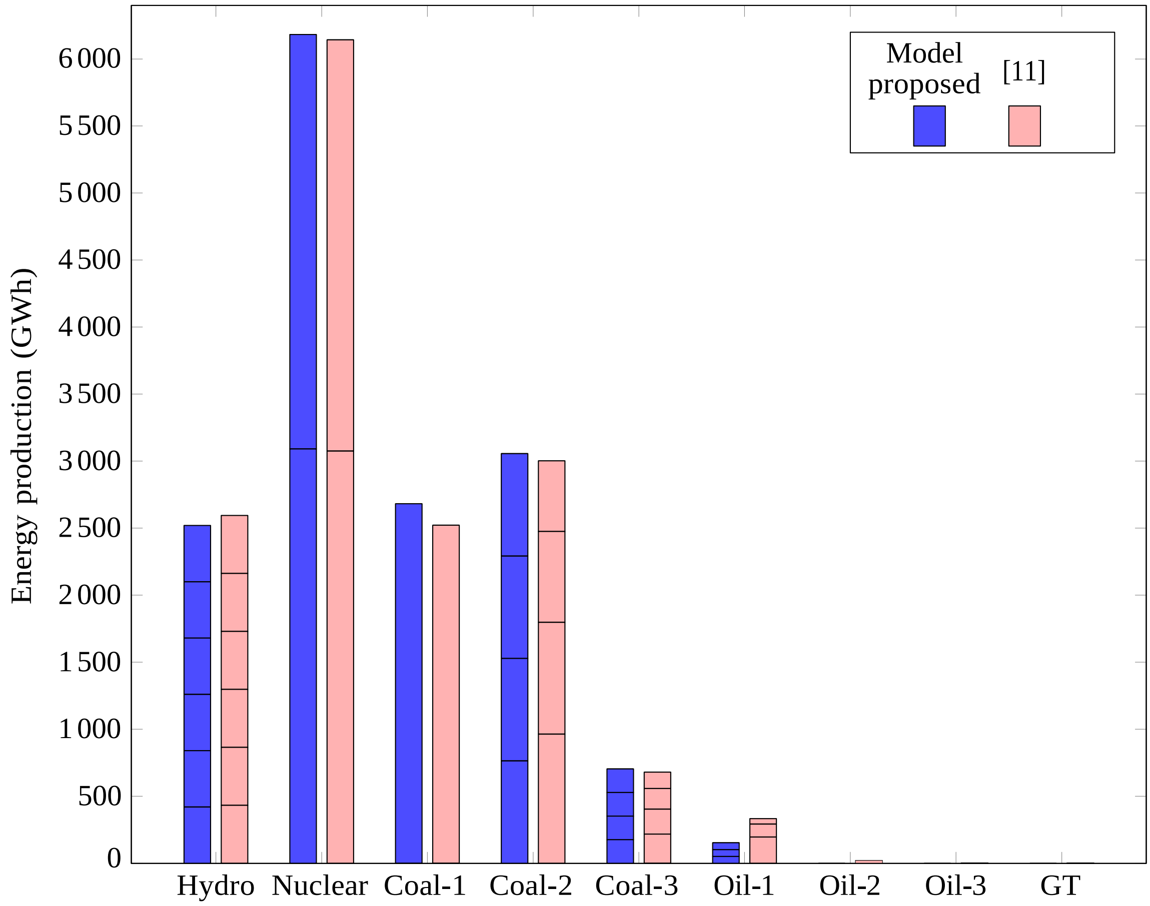


Figure 7.1: *Comparison of the energy production scheduled for each generating unit in each power station of the 32-unit IEEE-RTS inspired case study according to the model proposed in §3.3 and the modelling approach in [11].*

Table 7.1: *Comparison between the energy production scheduled according to the model proposed in §3.3 and the modelling approach in [11] for the 32-unit IEEE-RTS inspired case study. Columns three and four are represented graphically in Figure 7.1.*

Station Name	Unit	Model (GWh)	[11] (GWh)	Difference (Gwh)	Cost rate[1] ($/MWh)	Model ($)	[11] ($)	Difference ($)
Hydro	1	420	432	− 12	0	0	0	0
	2	420	432	− 12	0	0	0	0
	3	420	432	− 12	0	0	0	0
	4	420	432	− 12	0	0	0	0
	5	420	432	− 12	0	0	0	0
	6	420	432	− 12	0	0	0	0
Nuclear	7	3 091	3 075	16	20	61 483 968	61 163 182	320 786
	8	3 091	3 068	24	20	61 483 968	61 016 195	467 773
Coal-1	9	2 682	2 522	160	38	101 346 251	95 298 963	6 047 288
Coal-2	10	764	964	− 200	38	29 369 302	37 059 735	− 7 690 433
	11	764	834	− 70	38	29 369 302	32 056 523	− 2 687 222
	12	764	678	86	38	29 369 302	26 061 468	3 307 834
	13	764	527	236	38	29 369 302	20 276 602	9 092 700
Coal-3	14	176	218	− 42	48	8 401 113	10 385 301	− 1 984 188
	15	176	186	− 10	48	8 401 113	8 883 717	− 482 605
	16	176	154	22	48	8 401 113	7 345 807	1 055 306
	17	176	123	53	48	8 401 113	5 867 327	2 533 785
Oil-1	18	51	196	− 145	73	3 765 907	14 346 479	− 10 580 572
	19	51	97	− 45	73	3 765 907	7 073 511	− 3 307 604
	20	51	41	11	73	3 765 907	2 975 019	790 888
Oil-2	21	0	10	− 10	76	0	751 699	− 751 699
	22	0	6	− 6	76	0	431 623	− 431 623
	23	0	3	− 3	76	0	237 808	− 237 808
Oil-3	24	0	0	0	91	0	24 520	− 24 520
	25	0	0	0	91	0	22 691	− 22 691
	26	0	0	0	91	0	20 952	− 20 952
	27	0	0	0	91	0	19 214	− 19 214
	28	0	0	0	91	0	17 750	− 17 750
GT	29	0	0	0	144	0	38 214	− 38 214
	30	0	0	0	144	0	33 743	− 33 743
	31	0	0	0	144	0	29 562	− 29 562
	32	0	0	0	144	0	26 101	− 26 101
		15 297	15 296	2		386 693 566	391 463 705	− 4 770 139

[1] Rate in Table 5.1 inflated from 1979 to 2016 (× 3.315) .

Table 7.1 further contains a comparison of the energy production planned for the generation cost LP model proposed in this dissertation with the energy production planned by Allan *et al.* [11] in 1986, as well as the production difference in energy (GWh) and cost ($). As may be seen in Figure 7.1 and Table 7.1, although the LP model of §3.3 planned less production for the hydro stations, it planned more production for the nuclear units and the next stations in the merit order of Table 5.1, resulting in a $4 770 139 ($391 463 705 − $386 693 566) cheaper production plan (representing a 1.22% improvement). Per station, the LP model of §3.3 scheduled similar values to those in [11], as is evident in Figure 7.1. The LP model of §3.3, however, only plans per

Table 7.2: *Capabilities and parameter settings, together with their default values, in two off-the-shelf NSGA-II packages available in R.*

Settings	*nsga2* function in *mco* package	*nsga2R* package
Constraints	../	
Upper & lower bounds	../	../
Generations (g)	100	20
Population size (p)	100	100
Crossover probability	0.7	0.7
Crossover distribution index	5	5
Mutation probability	0.2	0.2
Mutation distribution index	10	10
Tournament size		2

station and so the energy production per station is equally spread across the generating units, whilst the energy production plan in [11] plans per unit, and thus the energy production per unit is much more varied (see Table 7.1 and Figure 7.1). This validates the LP model of §3.3 and adds verification that the LP production model by Brits [25] was coded correctly in R and implemented correctly by the author.

7.1.2 Comparison of results with those obtained by an off-the-shelf NSGA-II implementation

Most MO metaheuristics reside within the class of evolutionary algorithms [61, p. 207] and as such are typically referred to as MOEAs. The most popular MOEA is the NSGA-II [217]. More information on the working of the NSGA-II may be found in [52], a paper that has been cited a staggering 17 735 times according to Google scholar.

There are two NSGA-II implementations available as packages in R, namely the *nsga2* function in the *mco* (standing for multiple criteria optimization) package [140] and the *nsga2R* package [193]. Both packages are *off-the-shelf* algorithmic implementations which may be used relatively easily to find Pareto-optimal solutions to MOO problems. No additional coding implementation for the NSGA-II is required in these packages, which means that the packages automatically execute routines such as fast non-dominated sorting, crowding distance measurement, tournament selection, simulated binary crossover, and polynomial mutation. The user must specify the number and range of the decision variables, the objective functions, constraints (not available for the *nsga2R* package which is applicable to box-constrained MOO problems only), and important parameters of the algorithm. These parameters include the population size, tournament size, crossover probability, crossover distribution index, mutation probability, mutation distribution index, and the number of generations. The recommended/default algorithmic parameter settings for the packages, as presented in Table 7.2, were adopted by the author, whilst a slightly different population size and number of generation were employed. The NSGA-II packages in R are only capable of handling continuous decision variables. That said, a gross simplification for converting the discrete starting times for each unit (the decision variables X_i in the model of Chapter 3) was achieved by rounding the continuous decision variables produced by the NSGA-II, whereupon all the GMS objective and constraint functions could be further determined.

The non-dominated front obtained by a single eight-hour DMOSA run with the best parameter settings uncovered in Chapter 6 (according to average H values in Table 6.3) is compared with results obtained by the affiliated NSGA-II packages available in R for different paramter settings in Figure 7.2 (also allowing eight hours of computation time). As may be seen in Figure 7.2, the DMOSA algorithm found non-dominated solutions relatively close to the best-known SO solution found by Schlünz and van Vuuren [169] which only minimises the SSR objective. As may be further seen in Figure 7.2, the DMOSA algorithm implemented by the author outperformed the two different NSGA-II packages available in R for all the different parameter combinations. The NSGA-II's performance could naturally be improved if it were to compute longer, with possibly larger population sizes and if rounding of its continuous variables did not take place.

It was not unexpected that the DMOSA algorithm outperformed these off-the-shelf packages since the NSGA-II implementation have not been tailored specifically to the GMS problem's combinatorial nature and form. The above comparison should therefore not be considered to imply that the DMOSA algorithm is superior to the NSGA-II in respect of solving the model of Chapter 3. The purpose of the comparison was merely to validate whether the DMOSA algorithm was an appropriate choice and to verify whether the non-dominated fronts returned by the DMOSA algorithm are acceptable in terms of quality. In addition, it is a concern of the author that the DMOSA algorithm takes long to uncover non-dominated fronts due to the manner of its encoding. The NSGA-II R packages, were also observed to take relatively long to find adequate non-dominated fronts, which may be attributed to the relatively long computation time associated with the production planning LP model of Brits [25] employed in both cases to estimate the production cost objective of the model of Chapter 3.

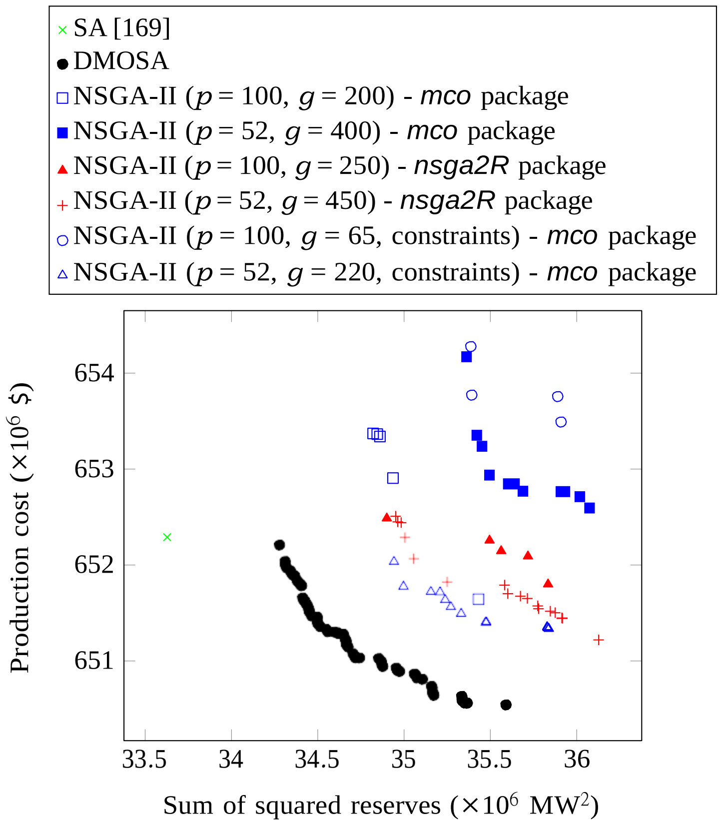
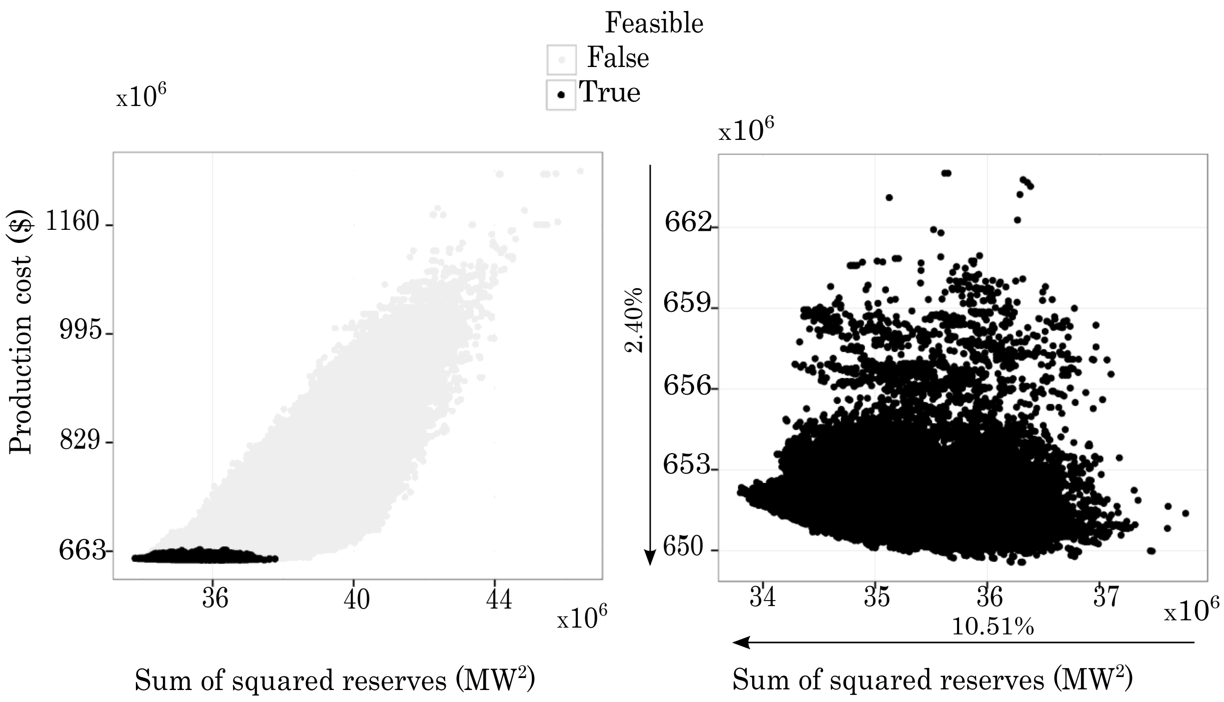


Figure 7.2: *Comparison of non-dominated fronts found by the DMOSA algorithm and the NSGA-II packages in R, for the 32-unit IEEE-RTS case study. All search runs took approximately 8 hours to complete.*

7.2 Non-dominated GMS solution uncovered

Figure 7.3(a) contains a representation of the objective space of the model of Chapter 3 discovered for the 32-unit IEEE-RTS inspired case study (both feasible and infeasible solutions) during the extensive parameter evaluations of Chapter 6. Figure 7.3(b) contains only the feasible GMS solutions in the objective space, showing that there is no strong trade-off present between levelising reserve margins (minimising the SSR) and minimising production cost based on the generation cost minimisation LP model derived by Brits [25]. This means that minimising the SSR objective will generally lead to minimum or close-to-minimum production costs, which is attributed to the fact that if there are sufficient net reserves (more achievable in GMS problem instances with low gross reserve levels[1] when levelising the net reserves), then there are fewer instances where energy has to be produced by more expensive units. A solution with the lowest SSR value does, however, not necessarily achieve the smallest production cost. This is confirmed in the GMS literature [149, 216], as mentioned before.

As may be seen in Figure 7.3(b), different feasible GMS solutions may exhibit more variation in respect of SSR objective function values (changes as large as 10.51%, representing $\pm\ 3.5 \times 10^6$ MW^2) whilst the production cost objective is not as sensitive (changes as large as 2.40%, representing $\pm\ \$16 \times 10^6$).



(a) (b)

Figure 7.3: *(a) Objective space representation of the feasible and infeasible GMS solutions uncovered by the DMOSA algorithm for the 32-unit IEEE-RTS inspired case study. (b) Objective space representation (zoomed in) of only the feasible solutions in part (a) of the figure.*

Figure 7.4 further illustrates the degrees of specific constraint violations embodied in the GMS solutions of Figure 7.3(a). As may be seen in Figure 7.4(b), the main reason for so many infeasible solutions achieving larger (worse) values in both objectives (see Figure 7.4(a)) is that these solutions predominantly violate the demand constraints. Intuitively, a GMS solution

[1]The net reserve is the gross reserve less the capacity lost due to maintenance. Furthermore, the gross reserve is the total installed capacity ($\sum_{i\in I} I_i$) less the expected demand during time period $j \in J$ [67, 90, 215, 201].

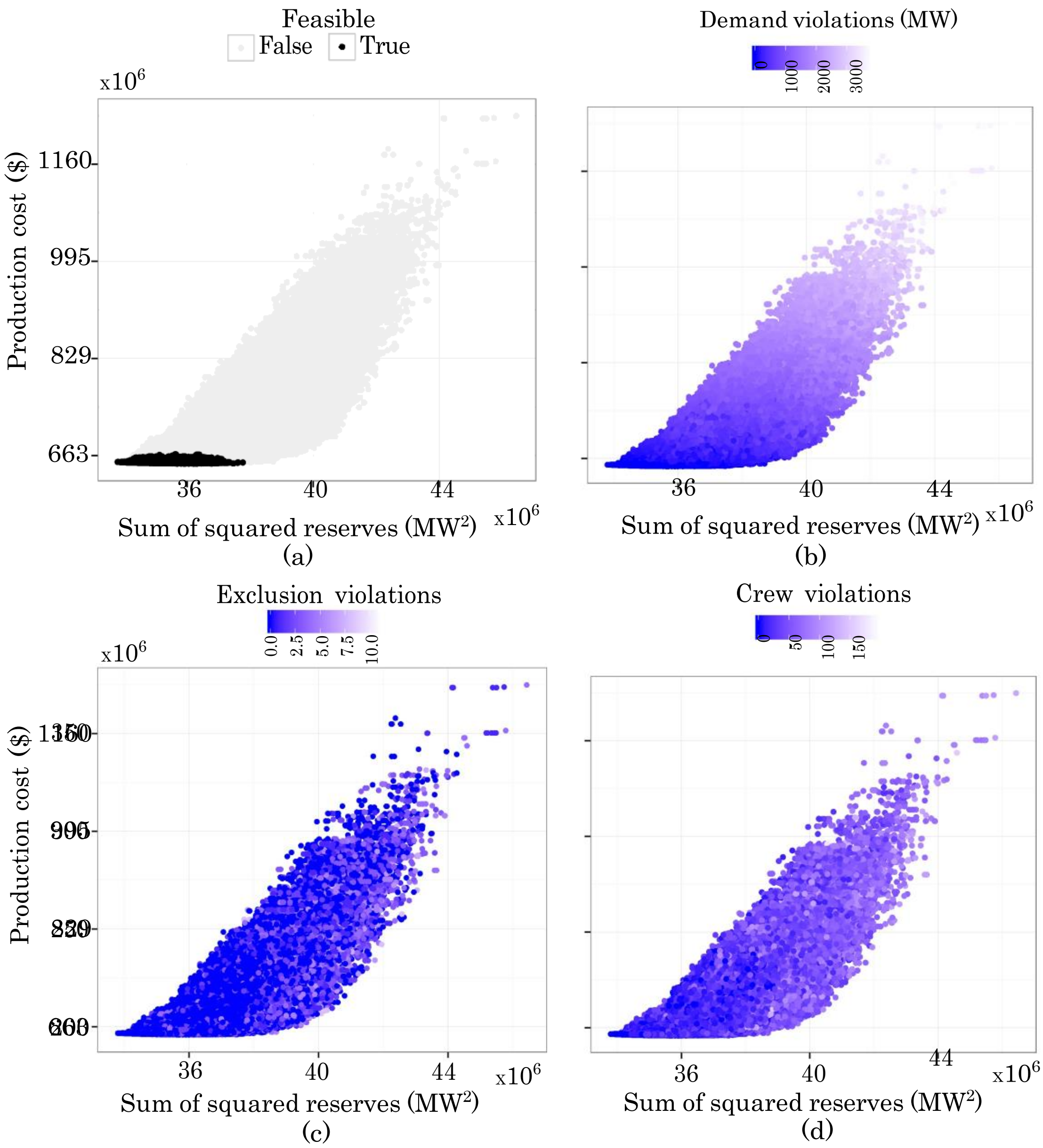


Figure 7.4: *Objective space representation of the feasible and infeasible GMS solutions returned by the DMOSA algorithm, for the 32-unit IEEE-RTS inspired case study. The objective function values are shown in (a) for all solutions. The same representation is shown in (b) [(c) and (d), respectively] together with an indication of the degree of load constraint variations [exclusion constraint violations and crew availability constraint violations, respectively].*

violating the demand constraints will achieve larger (worse) SSR and production cost values. There is, however, no clear trend in this respect between the exclusion constraint violations and the two GMS objective function values (*i.e.* the violations are more spread out in Figure 7.4(c)). GMS solutions that violate the crew availability constraints more tend to also achieve larger (worse) SSR-values (and sometimes larger production cost values), as may be seen in Figure 7.4(d). The reason for this is that GMS solutions achieving fewer crew constraint violations will generally also be more spread out (*i.e.* not too many units will be scheduled for maintenance during any one period), and will thus, depending on the expected demand profiles, exhibit more levelised reserves (lower SSR-values). In the next section, a further analysis is conducted in respect of the sensitivity of possible trade-offs uncovered when relaxing some of the GMS model's constraints.

The final attained front, obtained from the non-dominated fronts from all the experiments conducted in this dissertation for the 32-unit IEEE-RTS inspired case study is shown in Figure 7.5(a). In addition, the best solution obtained by the SO SA algorithm minimising the SSR in [169] is also plotted in the figure for comparison purposes. Figures 7.5(b), 7.5(c) and 7.6 contain comparisons of the daily available capacity and daily fuel costs for the two extremal scenarios in Figure 7.5(a), whilst Figure 7.7 contains a graphical comparison of their corresponding maintenance schedules for the 32 generating units.

As may be seen in Figures 7.5(b) and 7.5(c), if the net reserves are smaller, *i.e.* more maintenance is performed, the fuel cost is higher on any day. This increased cost is attributed to the more expensive units having to be bought online. This is especially evident when the two large and cheap nuclear units are scheduled for maintenance (see Figure 7.7). In scenario B, for example, the first nuclear unit was scheduled for maintenance earlier, starting in week 4 (day 28), than in Scenario A (see Figure 7.7) which forced the more fuel-expensive units to pick up the load that the nuclear unit would have supplied (since it is the cheapest), as may be seen in Figure 7.6.

More variation is exhibited in the SSR dimension of objective space than in the fuel cost dimension of objective space, as may be seen in Figure 7.5(a). It may be further of interest to the decision maker to note the daily and total variation in energy production per station (see Figure 7.6). A maintenance schedule may be preferred not just in terms of fuel cost, but other reasons. Schedules according to which less production scheduled at older and/or more environmentally harmful stations may, for example, be preferred over other schedules.

The attainment front for the 157-unit Eskom case study parameter tests conducted in §6.3 (see Figure 6.6) is illustrated in Figure 7.8(a). In addition, the best solution obtained by the SO SA algorithm minimising the SSR in [168, 170] is also plotted in the figure for comparison purposes.

Figures 7.8(b), 7.8(c) and 7.9 contain a graphical comparison of daily available capacity and daily fuel costs for the two extremal scenarios in Figure 7.8(a), whilst Figure 7.10 contains a similar comparison of their corresponding maintenance schedules for the 157 generating units.

Only the power stations that achieve a difference in fuel costs (according to the two schedules in the two extremal scenarios) are represented in Figures 7.6 and 7.9. Thus some stations are not included in these figures due to either their fuel cost rate being zero, their difference in daily energy produced being zero across the entire planning horizon, or no energy being produced at these power stations on any day of any of the two maintenance schedules scenarios (probably because all the cheaper stations were able to meet the load). As may be seen in Figure 7.9, only the first twelve coal-fired power stations (in the merit order) make up for the relatively small R4 946 483 fuel cost difference between maintenance schedules in Scenarios A and B.

As may be seen in Figure 7.8(a) the number and spread of points in the non-dominated front is substantially less than in the 32-unit IEEE-RTS inspired case study (compare the figure with Figure 7.5(a)). For the two extremal scenarios A and B in Figure 7.8, there is not as much variation in the generator maintenance schedules (see Figure 7.10), compared to the extremal schedules obtained for the 32-unit IEEE-RTS inspired case study (see Figure 7.7). This is attributed to the Eskom case study exhibiting tighter constraints in terms of the earliest (e_i) and latest (f_i) starting times allowed (*i.e.* narrower maintenance windows, which translates to restricted freedom of variation for the decision variables) than in the IEEE-RTS case study, as described in Tables 5.1 and 5.5, respectively. In the next section, an analysis is performed in respect of relaxing these maintenance window constraints.

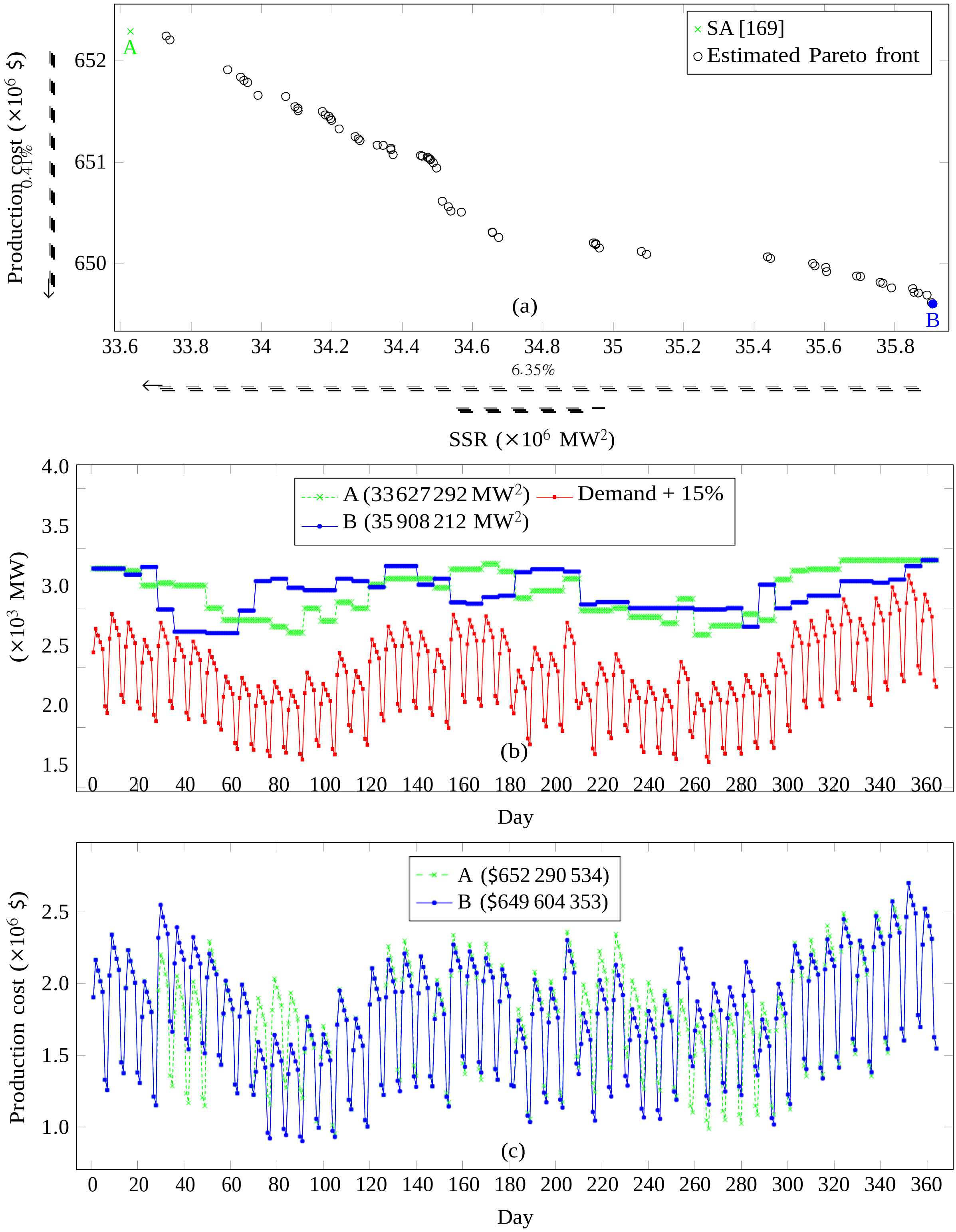


Figure 7.5: *(a) The estimated Pareto front (the attainment front obtained from all the non-dominated fronts produced during all search runs in this dissertation) for the 32-unit IEEE-RTS inspired case study, with the minimum cost solution in the front (B) compared to the SO SA solution (A) found in [169], which only minimises the SSR. (b) The daily available capacity for these two extremal scenarios (A and B in part (a) of the Figure). (c) The daily fuel cost for the two extremal scenarios.*

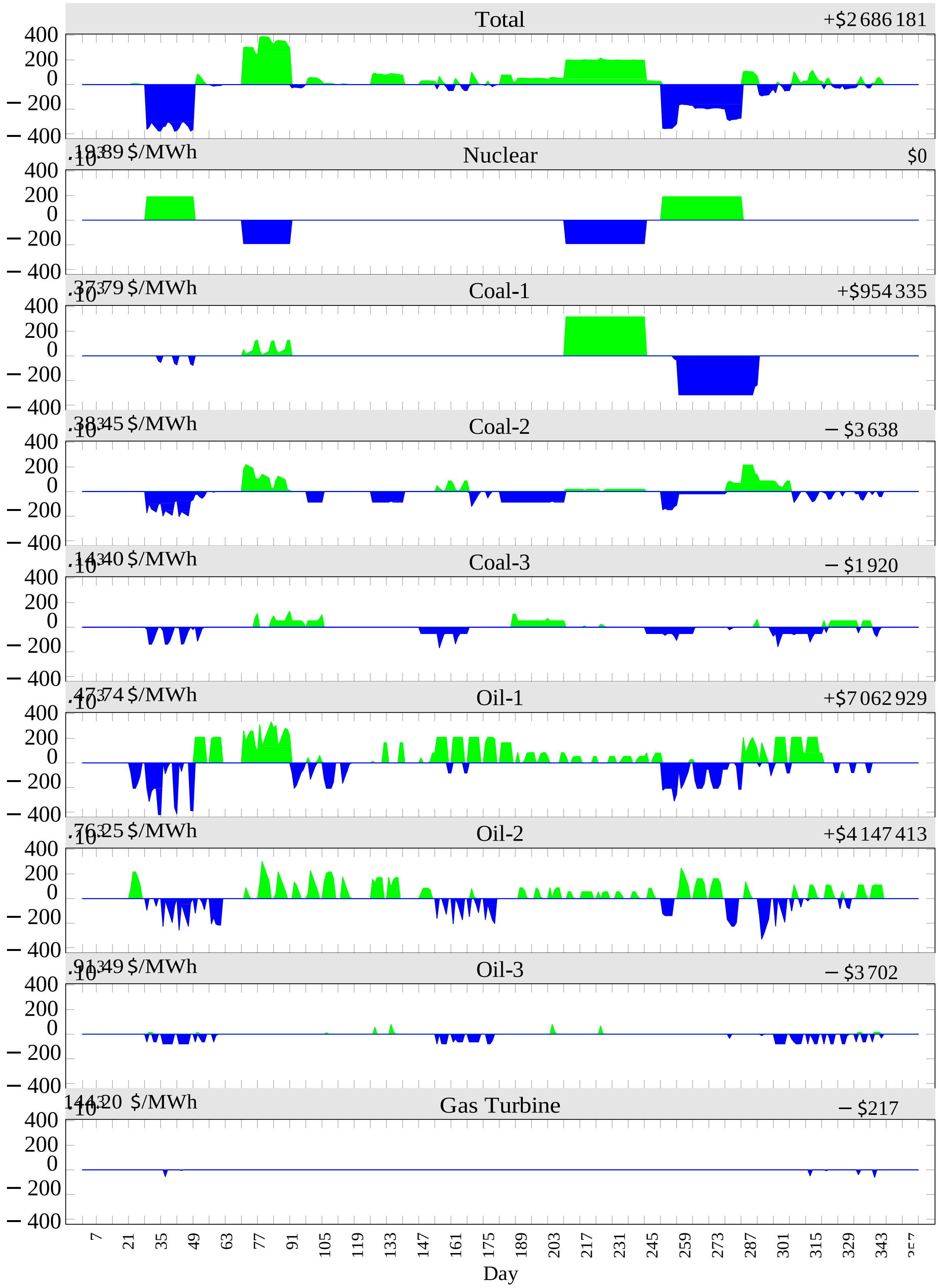


Figure 7.6: *The daily fuel cost difference between the two extremal scenarios (A minus B in Figure 7.5), for the 32-unit IEEE-RTS inspired case study. The vertical axis is measured in units of* $\$10^3$.

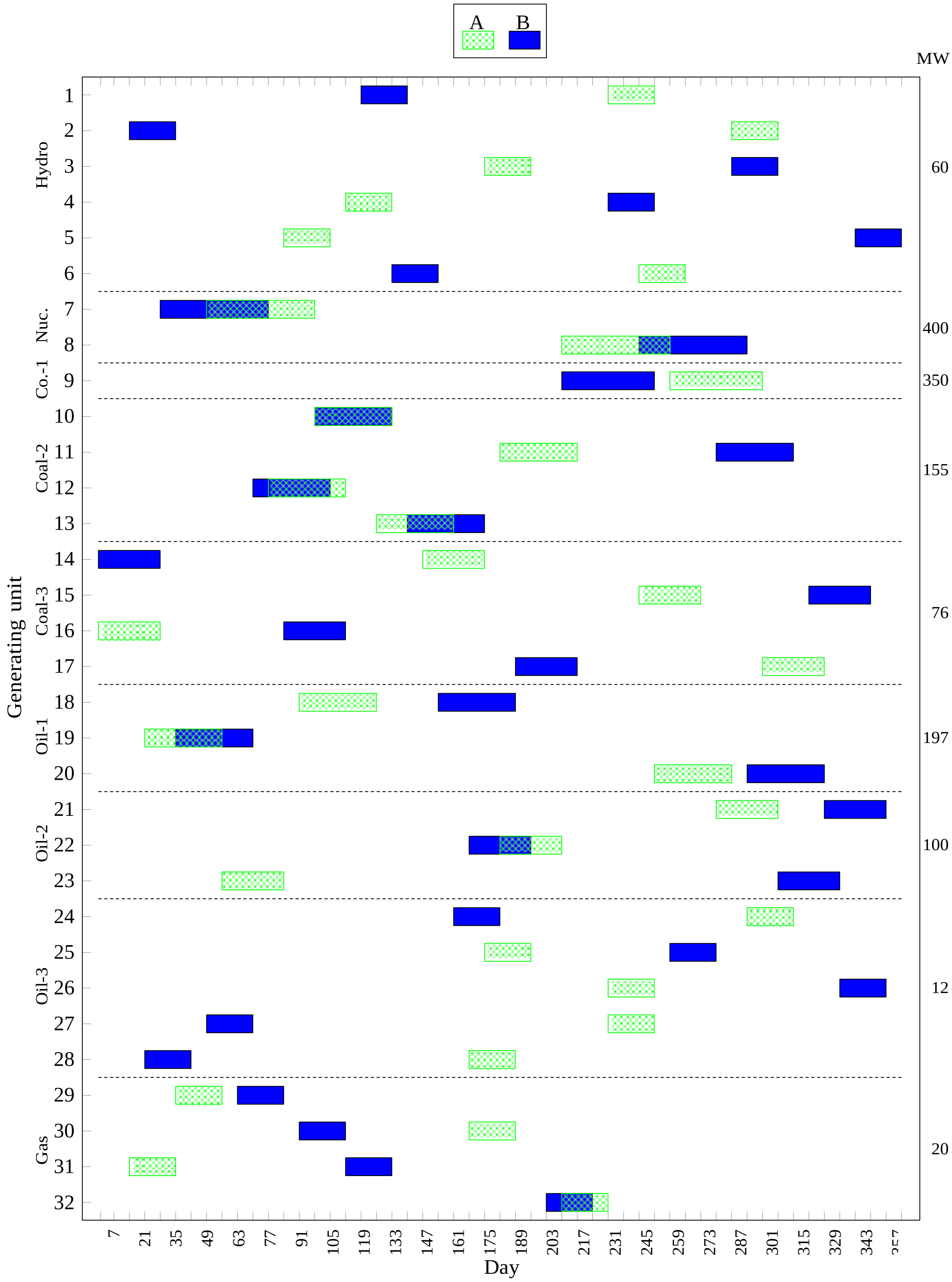


Figure 7.7: *Comparison of the maintenance schedules corresponding to the two extremal scenarios (A and B in Figure 7.5), for the 32-unit IEEE-RTS inspired case study. The bars represent unit values of the binary GMS auxiliary decision variables Y_{ij}. Scenario A represents the best solution found by Schlünz and van Vuuren [168, 169]. The units are arranged according to the merit order given in Table 5.1.*

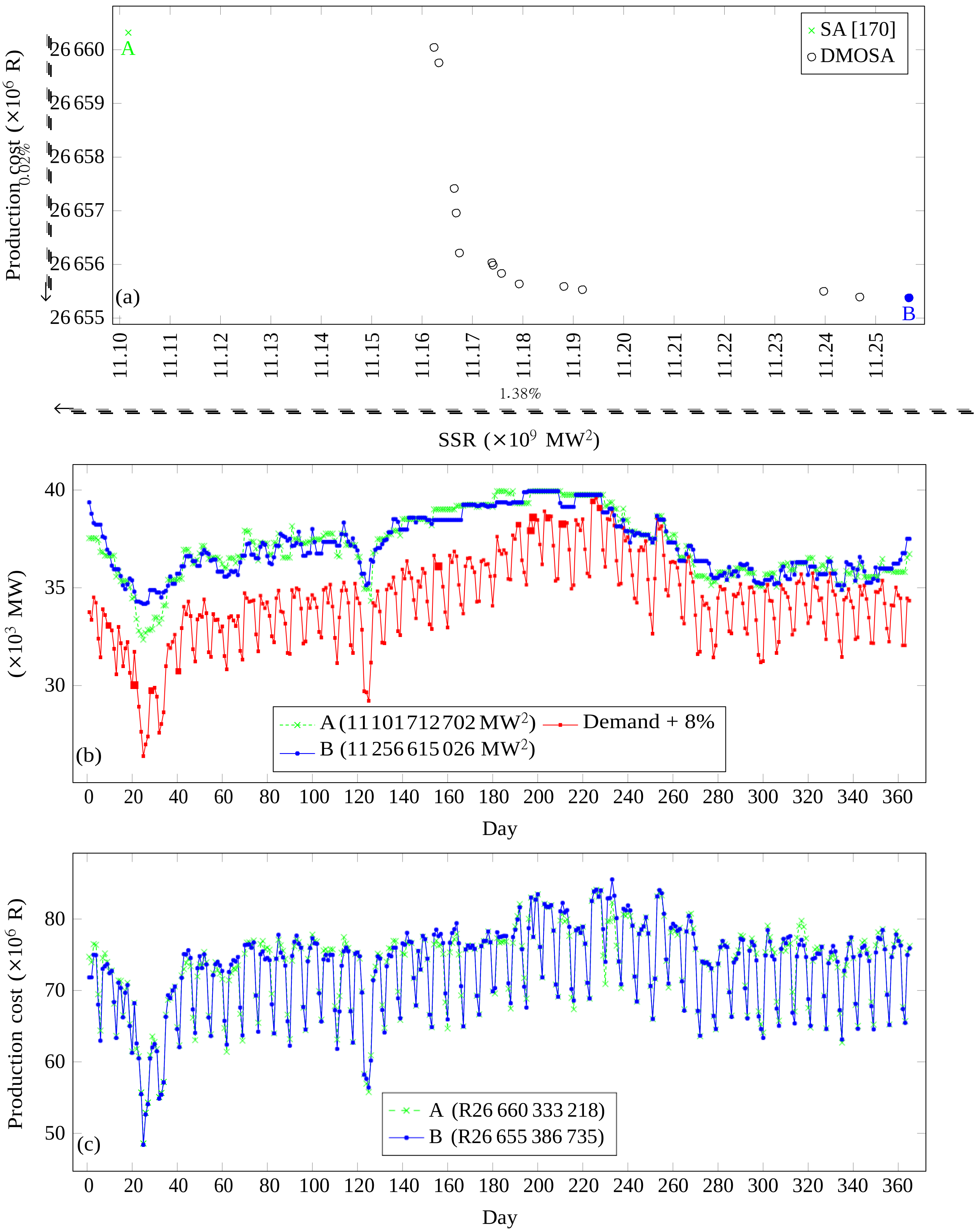


Figure 7.8: *(a) The attainment front obtained from all the non-dominated fronts produced during the search runs reported in Chapter 6 for the 157-unit Eskom case study, with the minimum cost solution in the front (B) compared to the SO SA solution (A) found in [168, 170], which only minimises the SSR. (b) The daily available capacity for these two extremal scenarios (A and B in part (a) of the figure). (c) The daily fuel cost for the two extremal scenarios.*

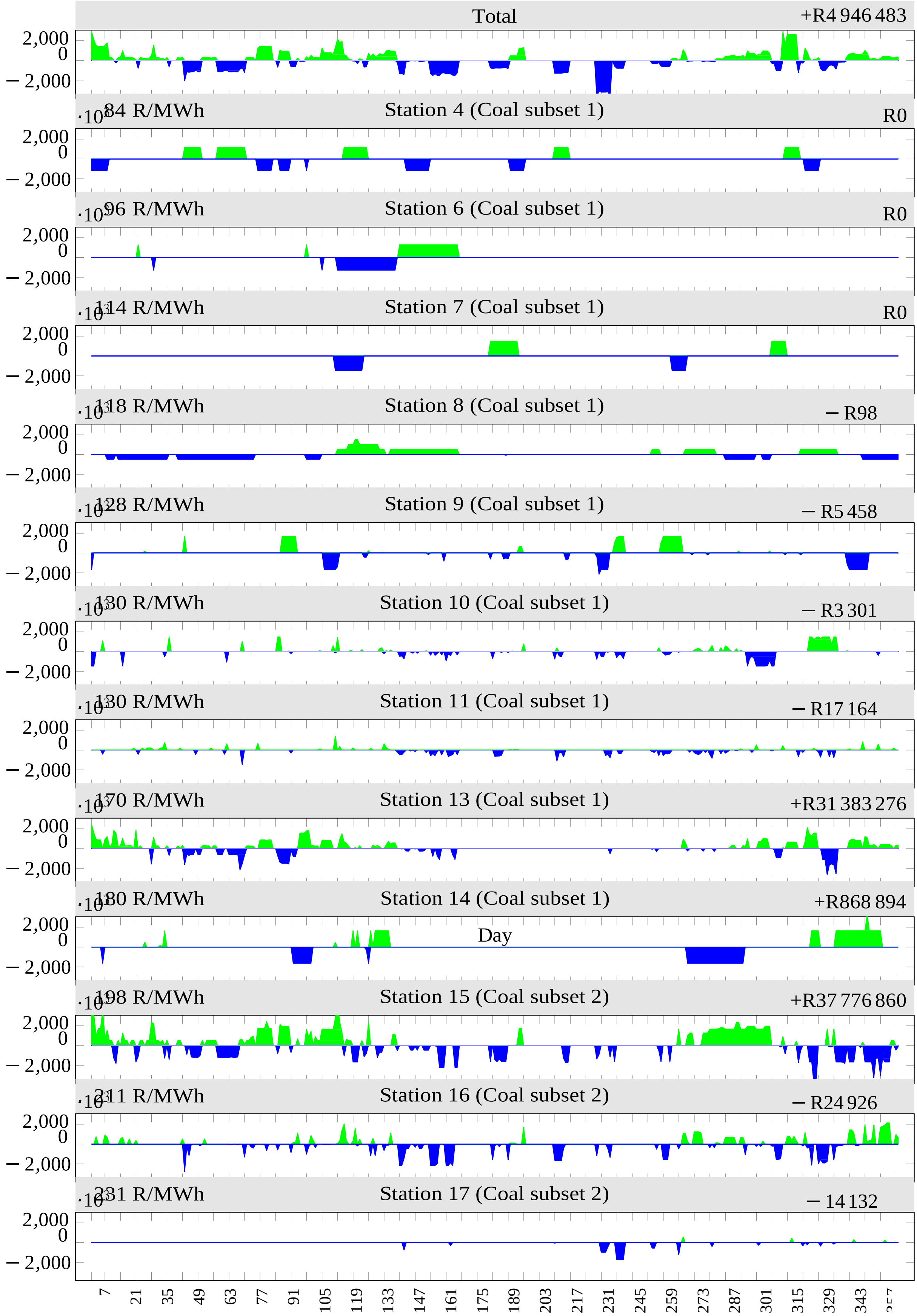


Figure 7.9: *The daily fuel cost difference between the two extremal scenarios (A minus B in Figure 7.8), for the 157-unit Eskom case study. The vertical axis is measured in units of R*10^3.

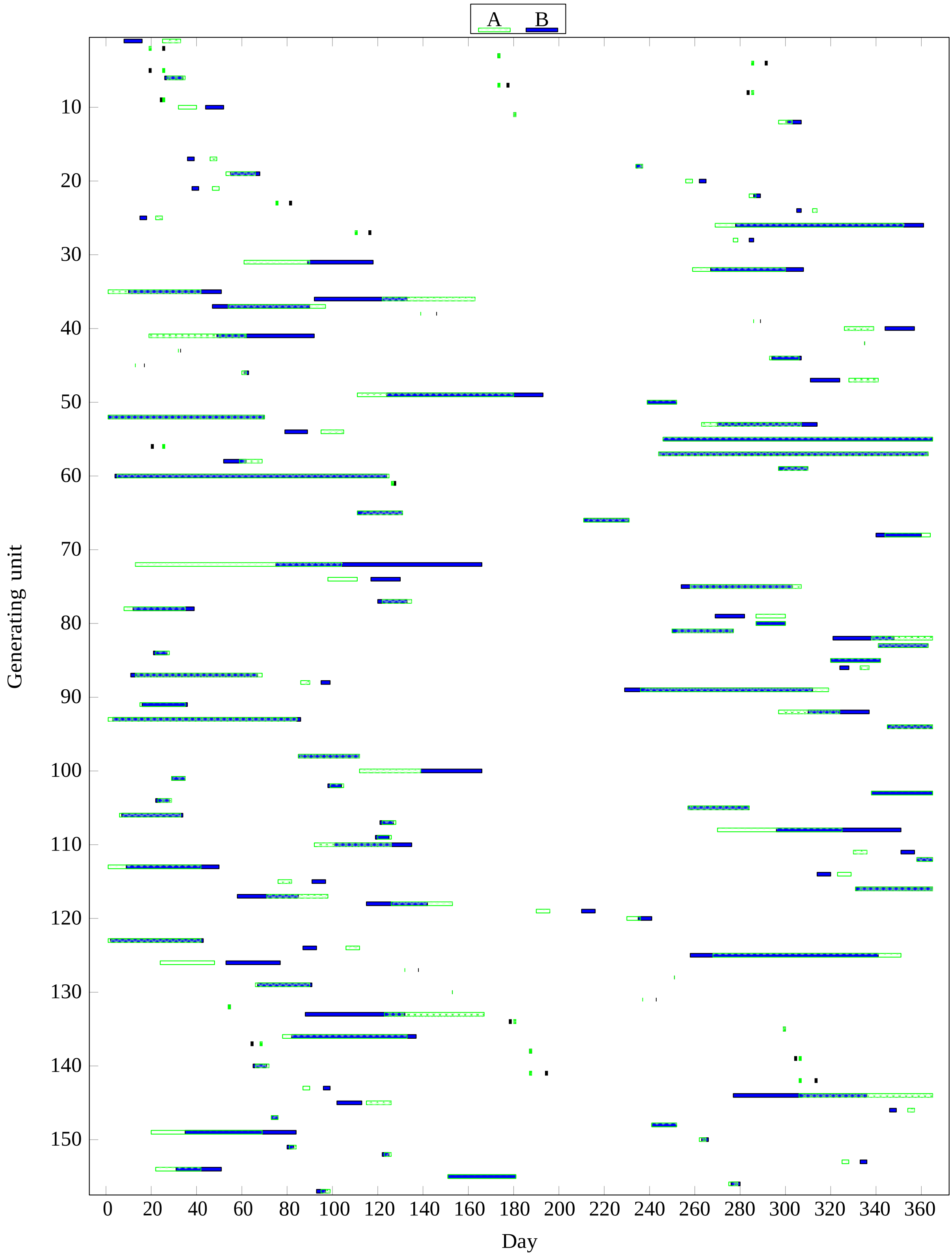


Figure 7.10: *Comparison of the maintenance schedules corresponding to the two extremal scenarios (A and B in Figure 7.8) for the 157-unit Eskom case study. The bars represent unit values of the binary GMS auxiliary decision variables Y_{ij}. Scenario A represents the best solution found by Schlünz and van Vuuren [168, 170] and was also presented in Figure 2.4. The units are arranged according to the order given in [168], not in the merit order of Table 5.5.*

7.3 Sensitivity analysis in respect of constraint relaxations

As mentioned in §2.5.1, a rough distinction may be made between constraints that must not be violated and constraints that are only required to be more or less satisfied [17, 124]. Some constraints (such as the amount of crew available) may be relaxed in order to find better GMS solutions [66]. This section is therefore dedicated to analyses in respect of how relaxing some of the GMS constraints may affect the objective function values of GMS solutions and at what cost, by including certain GMS constraints as another (composite) objective to be optimised within an MO paradigm.

7.3.1 Relaxing the soft constraints entirely

Figure 7.11 contains the non-dominated front returned found by the DMOSA algorithm before and after the manpower constraints in the 32-unit IEEE-RTS inspired case study were relaxed. This was easily achieved as a result of the elegant multiplicative constraint penalty function employed, where the multiplicative penalty severity factor γ may be set to 0. This results in the exponential multiplicative penalty factor taking the value 1, and thus rendering the objective function values unaffected by the amount of soft constraint violation (see §4.3.4).

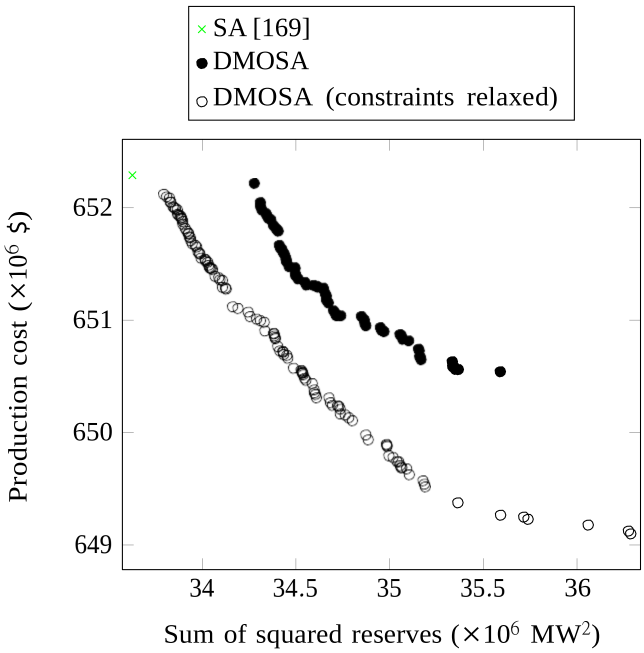


Figure 7.11: *The non-dominated fronts returned by the DMOSA algorithm within 8 hours before and after relaxing the soft constraints in the 32-unit IEEE-RTS inspired case study.*

Figure 7.12 contains a representation of the same data as those in Figure 7.11, but specifically illustrates how much each of the two constraint sets (exclusion and crew constraints) were violated, as well as the maximum number of crew required to implement the maintenance schedule. No demand violation penalties are therefore present in the results of in Figure 7.12. This absence is attributed to the fact that intuitively those GMS solutions that violate demand constraints exhibit larger (worse) SSR-values and production cost values as discussed above and illustrated in Figure 7.4. In addition, the gross reserves for the 32-unit IEEE-RTS inspired case study are sufficiently high and so not many instances of demand constraint violation occur.

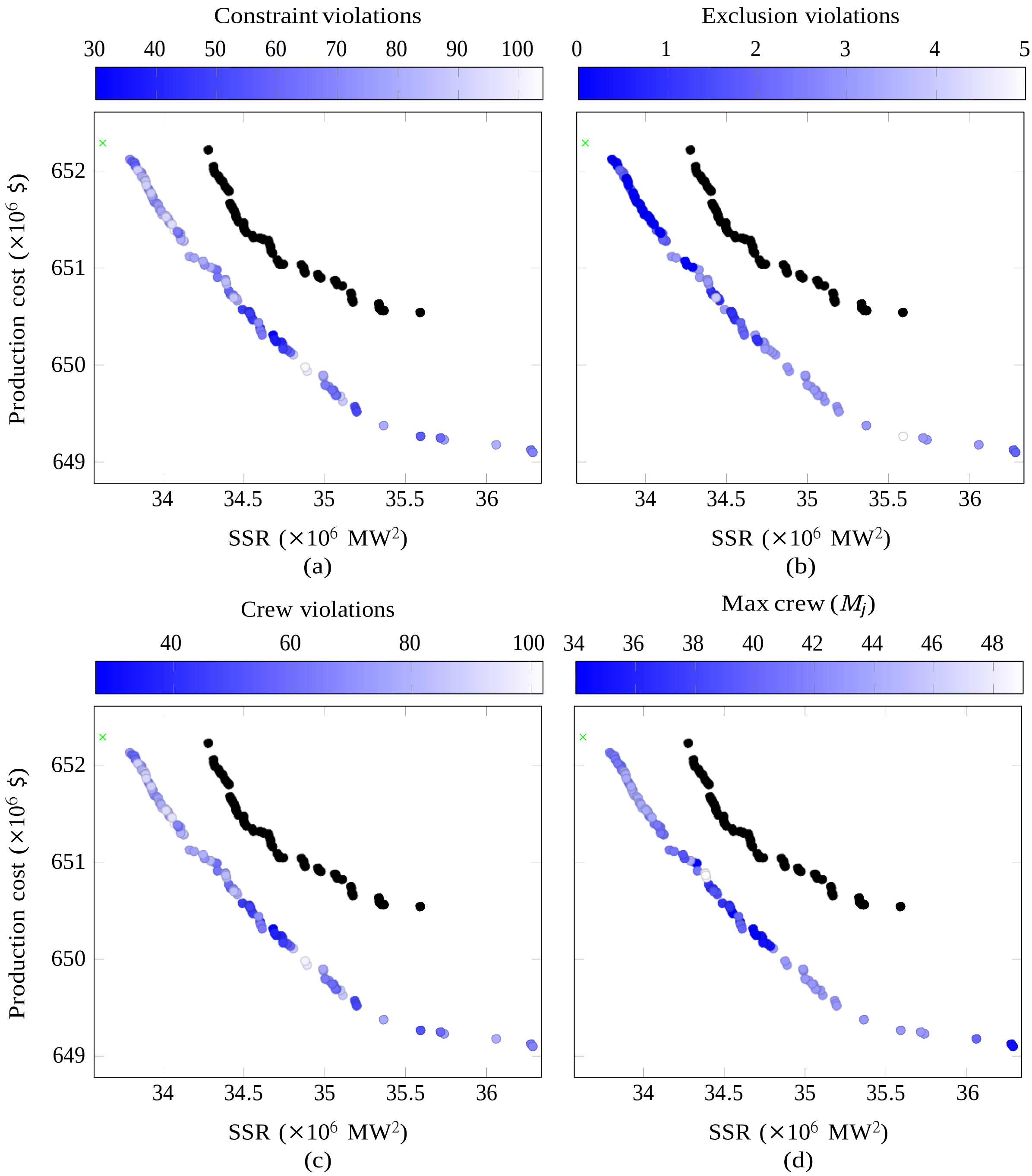


Figure 7.12: *Improved non-dominated front obtained by relaxing the soft constraints in the 32-unit IEEE-RTS inspired case study.*

In [66], the crew constraint was allowed to be violated by as much as 5%. The exclusion constraints of the 32-unit IEEE-RTS inspired case study would usually not be allowed to be violated, whilst efforts may of course be made, to a certain extent, to employ more crew members in the case of crew availability constraint violations. Figure 7.13 represents the non-dominated fronts before and after relaxation of the crew availability constraint set. Only the number of crew available was altered — all the other constraints sets (exclusion and demand constraints) were satisfied. As may be seen in Figure 7.13, further improvements in both (the SSR and production cost) objectives are achievable by increasing the number of maximum crew available. This will naturally come at an additional expense (requiring employment of more full-time crew). As may further be seen in Figure 7.13, there is a stronger trade-off between satisfying the crew availability constraint and minimising the production cost objective than between satisfying the crew available constraints and minimising the SSR. This strong trade-off is attributed to the fact that usually schedules which have lower SSR-values, will usually involve maintenance that is typically more spread out during the year (as dictated strongly by the expected peak demand profile), and so the amount of crew required will also be more spread out, resulting in a situation where a smaller number of crew members is required. The presence of the production cost objective, however, results in scheduling most of the cheaper units during low energy demand periods so that cheaper units are online during times of high demand, which will translate into less spread-out maintenance schedules and possibly more maintenance crew required during certain periods — hence the stronger trade-off. These trade-offs are explored further in an MO paradigm in the next section.

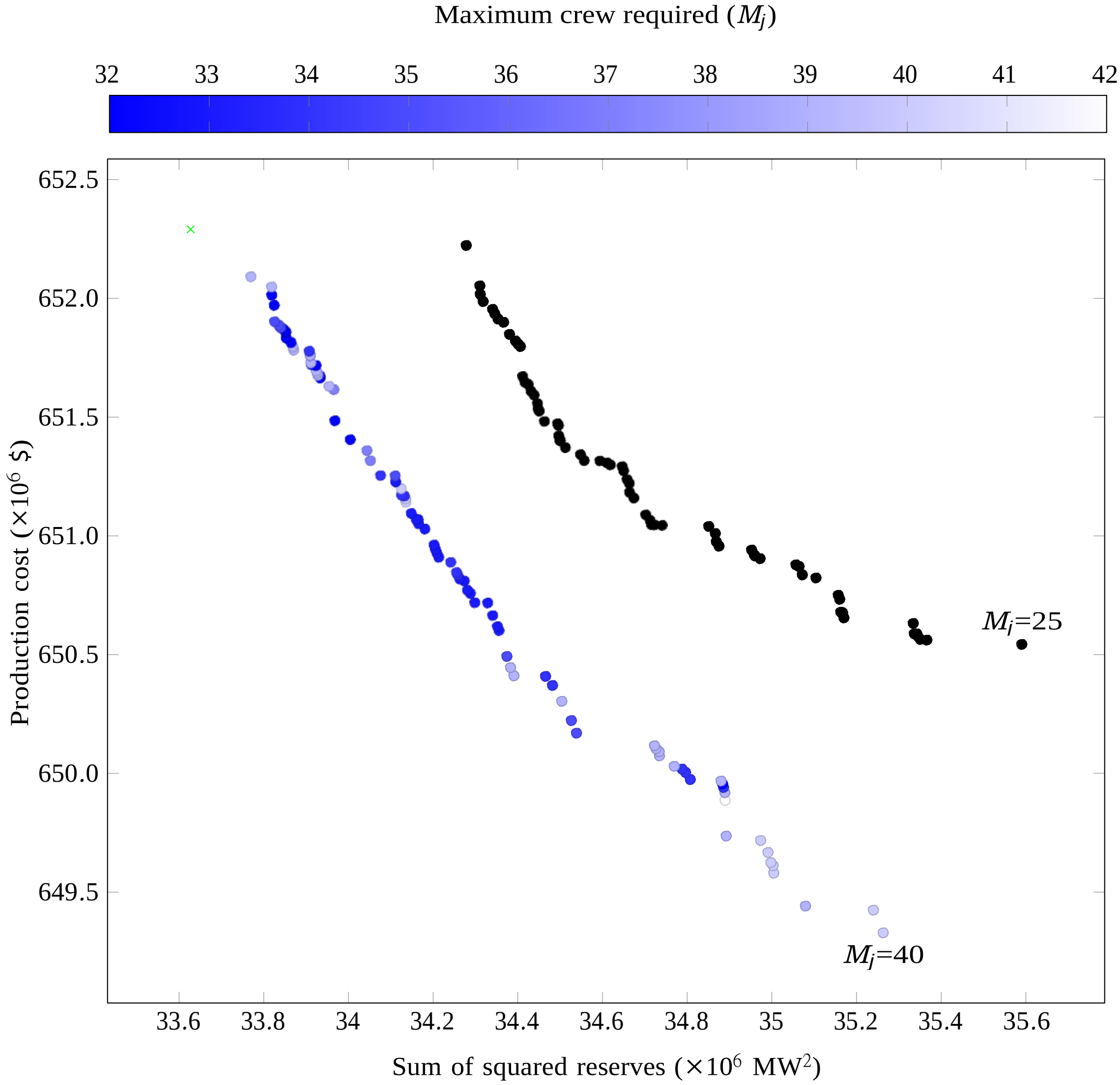


Figure 7.13: *Improved non-dominated front obtained by relaxing only the maximum crew availability constraint in the 32-unit IEEE-RTS inspired case study.*

The 157-unit Eskom case study does not have any crew constraints, but has demand and exclusion constraints. The exclusion constraints may not be violated since they are, in fact, only present to avoid that real units and their virtual copies are in simultaneous maintenance (which is not physically possible). The demand constraints ensure that the available capacity during any period j is at least as large as the expected peak demand D_j plus a pre-specified safety margin (S = 8% in this case). The gross (and subsequently the net) reserves are much tighter in the Eskom case study than in the IEEE-RTS inspired case study — compare Figure 7.5(b) to 7.8(b). Since the demand constraints are tighter in the 157-unit Eskom case study, relaxing them will have a larger impact for the Eskom case study. This is illustrated in Figure 7.14, where decreasing the safety margin from 8% (● in Figure 7.14) to 0% (○) allowed the DMOSA algorithm to return considerably better non-dominated fronts.

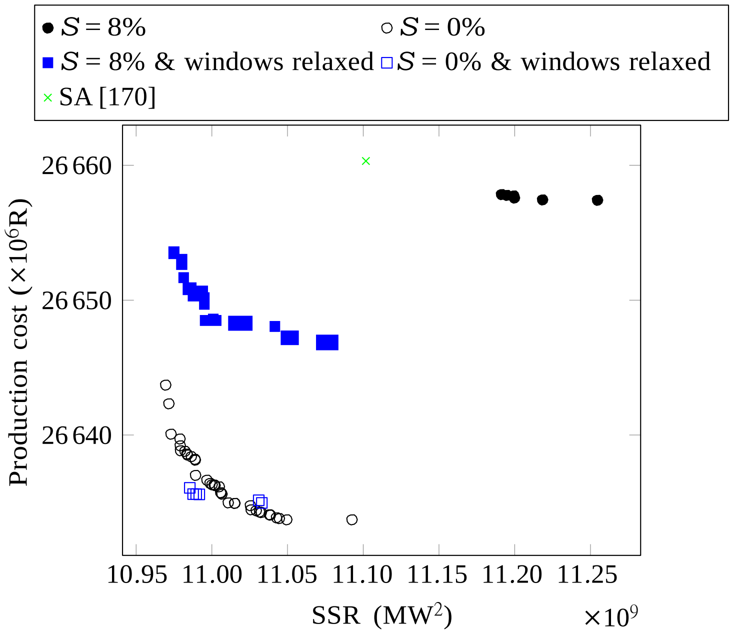


Figure 7.14: *Improved non-dominated fronts obtained by the DMOSA algorithm for the 157-unit Eskom case study within 8 hours*[2] *by relaxing the right-hand sides of some of the model's constraints, namely the safety margin S in the demand constraints, as well as the earliest and latest maintenance starting times (making the windows as wide possible).*

Figure 7.14 illustrates the improved non-dominated fronts achievable if the earliest and latest maintenance starting times are relaxed to the extreme values in the maintenance window. This maintenance window constraint relaxation was tested for both the normal safety margin of 8% (■) and when relaxing this value to 0% (□). The improvements achievable are attributed to more GMS solutions being available to choose from (*i.e.* more freedom of variation for the decision variables). When both the demand and window constraints' right-hand side values were relaxed (□) and the DMOSA algorithm was allowed to run considerably longer than eight hours, very few, but encouragingly small, SSR and production cost value combinations were found. It is assumed that if the DMOSA algorithm were to be allowed more computation time, further improved fronts (than □) would be returned. Since the domain of the decision variables is much larger (as a result of wider maintenance windows), there are many more maintenance scheduling possibilities available. Since the earliest and latest maintenance starting times specified in the Eskom case study naturally force most maintenance to occur during low demand periods, the

[2] Except the run relaxing both the safety margin value to 0% and relaxing the maintenance window earliest and latest times to the scheduling window extremes (□). This combination required a much longer computation time.

algorithm therefore finds good SSR-values (and associated low production cost values) more quickly (compare ○ with □ in Figure 7.14). This points to the fact that the maintenance window and demand constraints of the 157-unit Eskom case study are such that they restrict GMS solutions to seemingly good solutions in terms of the two GMS objectives proposed in Chapter 3.

The earliest and latest maintenance starting times are tighter in the Eskom case study than in the IEEE-RTS inspired case study — compare these values in Tables 5.1 and Table 5.5. This is also assumed to be one of the reasons why fewer non-dominated solutions are available in the Eskom case study (see Figure 7.5) than in the IEEE-RTS case study (see Figure 7.8), but this could also occur if the demand and window constraints were not as restricting.

Figure 7.14 illustrates the fact that the safety margin (and thus the demand constraint) is more restricting on the GMS model objective functions than the maintenance window starting times since, as may be seen in Figure 7.14, the non-dominated fronts in the former case (◐) are better than those in the latter (■). This observation is, however, specific to the degree of constraint relaxation — in this instance this relaxation is from 8% to 0% for the safety margin and for the maintenance window the extreme points (the scheduling window start and end).

7.3.2 Maximum crew trade-offs

Instead of only considering the maximum crew available as a constraint to be satisfied, one could rather interpret the maximum number of crew members required over the entire scheduling window to carry out a maintenance schedule as another objective to be minimised within an MO paradigm. This was done separately for the two objectives (the SSR and production cost). The resulting model was also solved using the same DMOSA algorithm, and the results are reported in Figures 7.15 and 7.16. As has been stated in §7.3.1, and may further be seen in Figures 7.15 and 7.16, there is a stronger trade-off between the maximum crew required and the minimisation of production cost than between the maximunm crew required and minimisation of the SSR.

In addition many more iterations were possible (44 038) within the same computation time budget of eight hours for the objectives involving calculating the maximum crew required and minimising the SSR (see Figure 7.16) than (15 291) for the objective involving calculating the maximum crew required and minimising the production cost (see Figure 7.16). This is mainly due to the fact that the production cost takes much longer to calculate.

An important point to stress is that although a schedule may require a smaller maximum number of crew members across the entire scheduling window, the crew required per time period might not be that well spread out over the scheduling window (*i.e.* large variations may occur between the number of crew members required over the scheduling window), meaning that some of the crew might be idle during certain time periods of the planning horizon. It may therefore be of interest to develop another GMS objective which levelises the crew required over the planning period, possibly borrowing from the formulations described in §2.5.2, in respect of levelising the reserve margins over the planning horizon.

7.3.3 Reducing the maintenance duration

If maintenance outage durations are allowed to vary (as mentioned they are usually fixed), a trade-off results between the energy production cost and the maintenance cost [46, 215]. Shorter outage durations generally lead to higher maintenance costs [222], since more crew and parts are typically required in a shorter time [215], but lower production costs are incurred since more

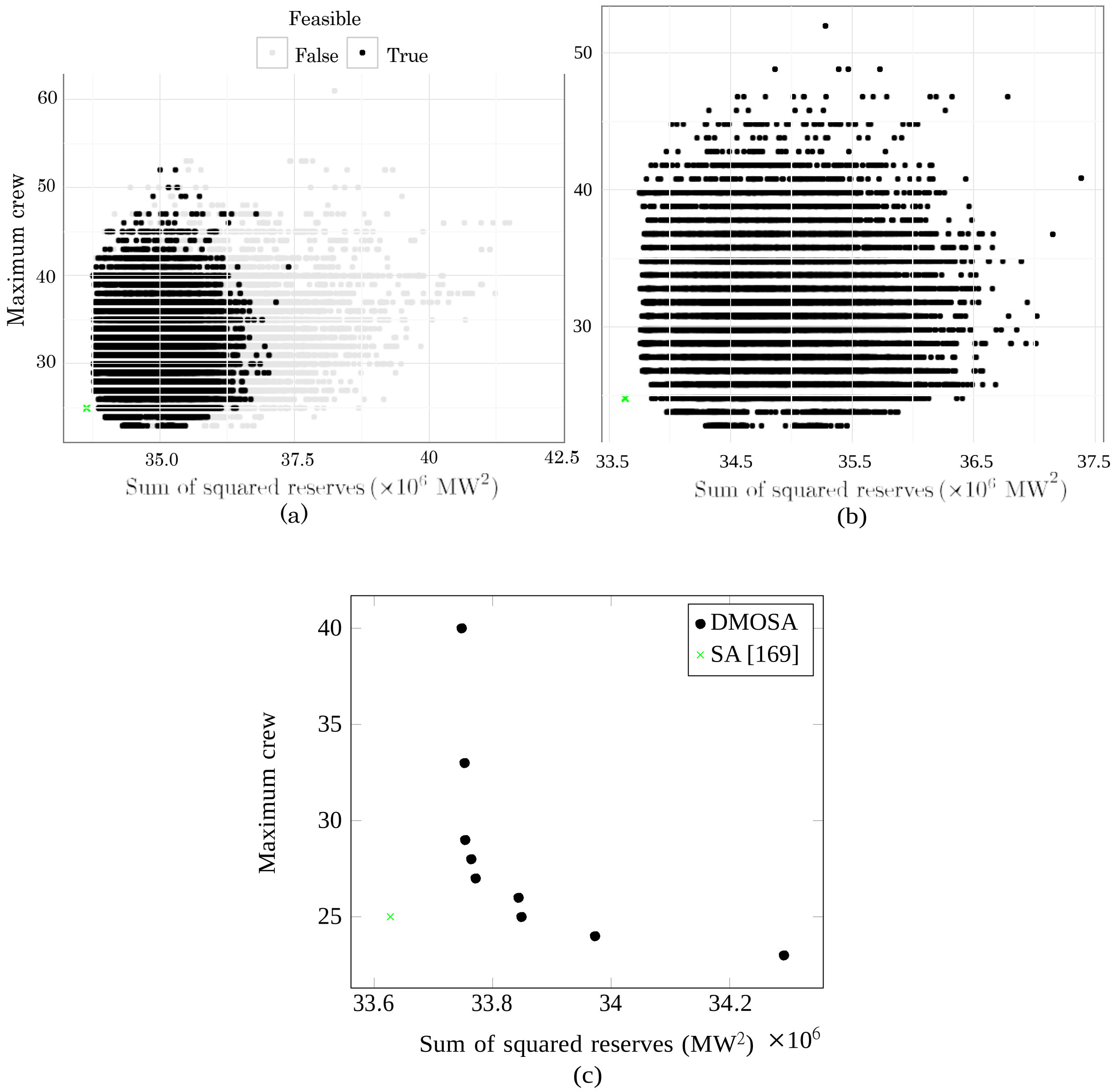


Figure 7.15: *Trade-offs uncovered by the DMOSA algorithm within 44 038 iterations (8 hours) for the 32-unit IEEE-RTS inspired case study between minimisation of the SSR and the maximum crew required.*

expensive units do not have to be online as much during those times when cheaper units are in maintenance [46, 222]. As further noted in [59], a three-week outage could, for example, be reduced to two weeks by securing overtime crew (at a known cost), although overtime may be an expensive contributor to maintenance costs [59] and will most probably very rarely be permitted. The power system's reliability (in terms of satisfying demand) will, however, be higher as a result of shorter outage durations.

Figure 7.17 contains a representation of the objective space for the different GMS model requiring the normal durations of maintenance (as in Figure 7.3(a)), halving the times required for maintenance and dividing the times required for maintenance by four (rounding up in both cases), and the extreme where no maintenance occurs ($\triangle$ in Figure 7.17). Naturally, the SSR-value will be larger as maintenance duration decreases since the available capacity will be higher. Similarly, the production cost will be lower as maintenance duration decreases since there will

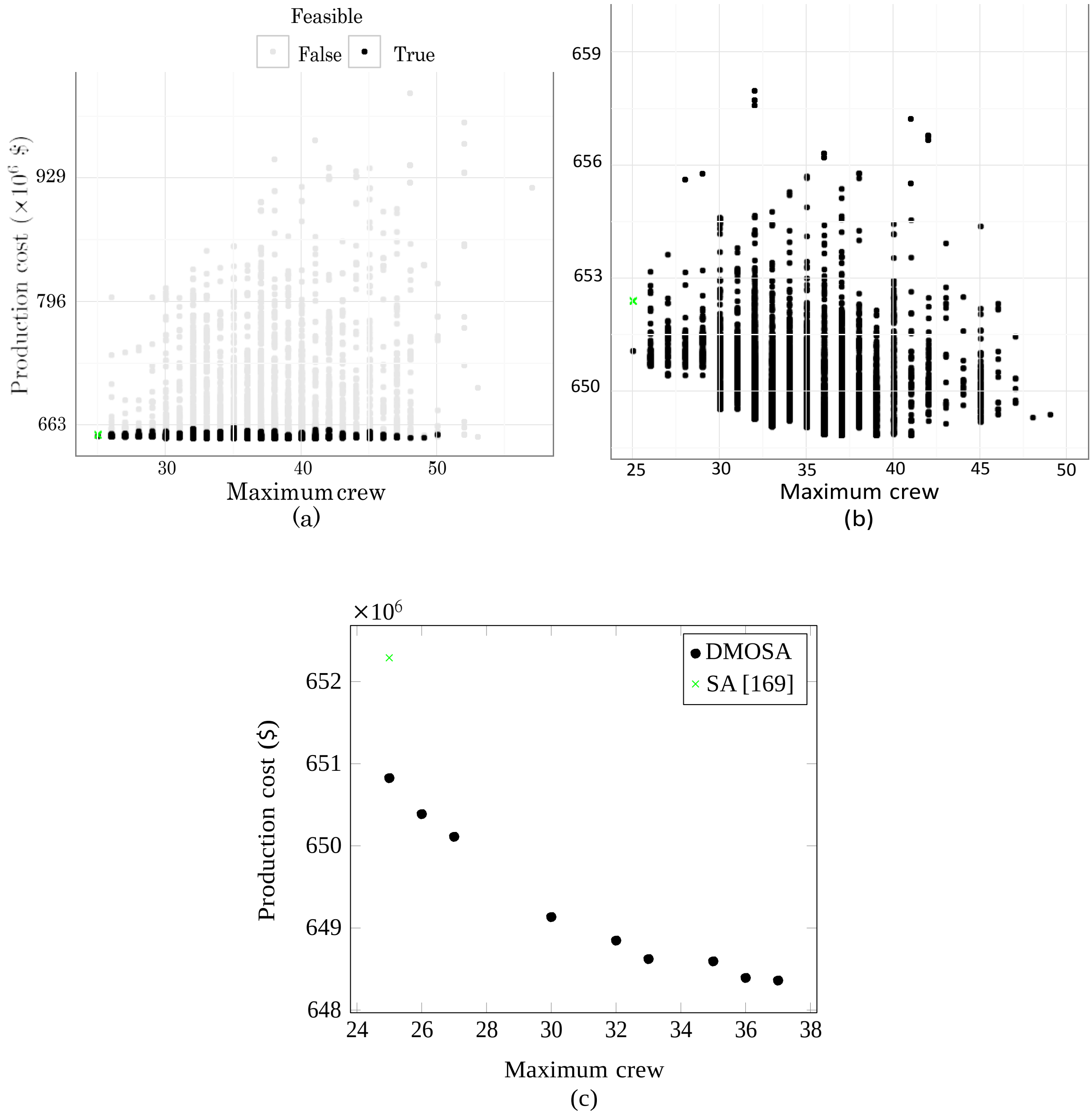
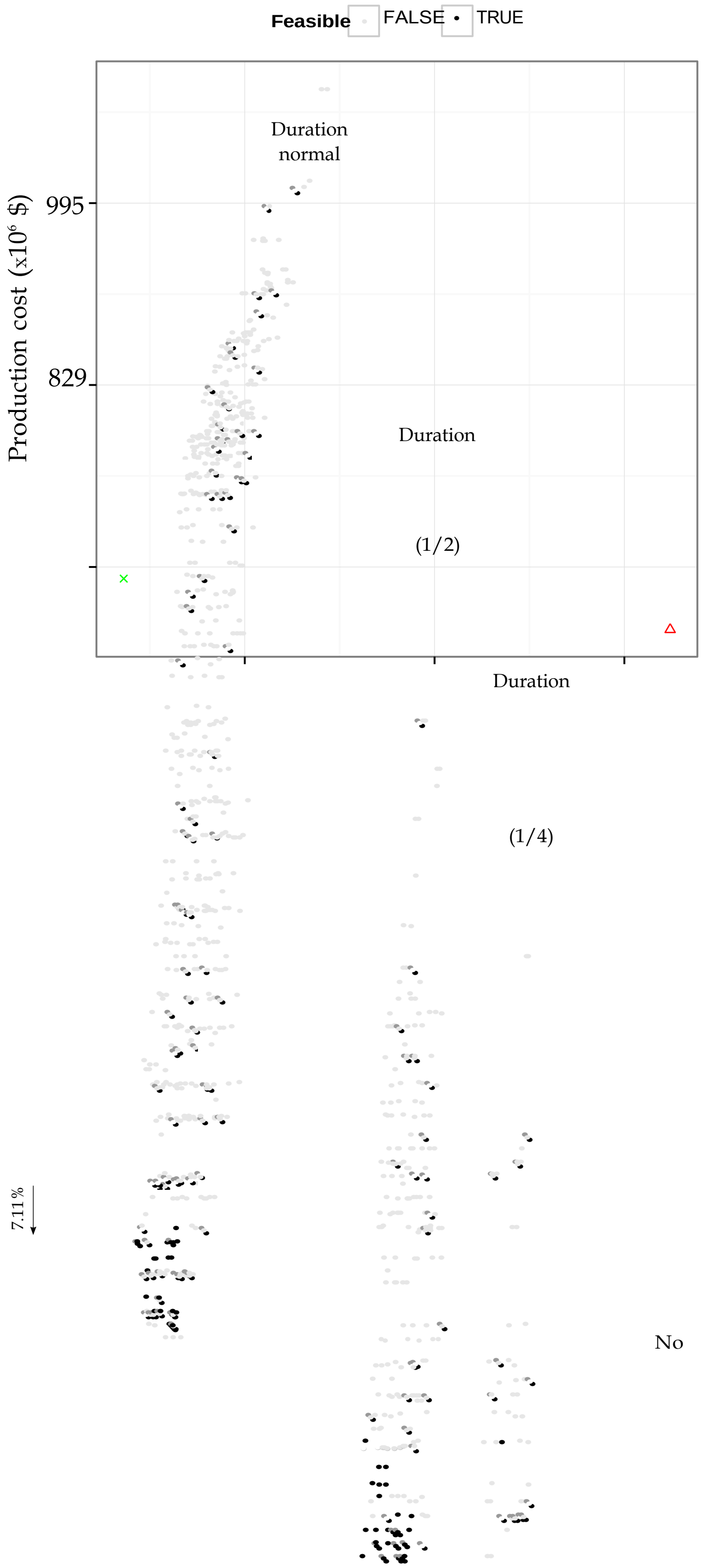


Figure 7.16: *Trade-offs between the production cost objective and the maximum crew required uncovered by the DMOSA algorithm for the 32-unit IEEE-RTS inspired case study within 15 291 iterations (8 hours).*

be more net reserves available, ensuring that the more expensive units do not have to be brought online. As mentioned in §2.4.2, the maintenance durations should ideally be as short as possible, but there is a limit to how much the duration can be shortened. An analysis may be performed to estimate the benefit to cost ratios achievable when shortening the maintenance durations, in terms of the energy production cost savings and higher reserves possible. The SSR and fuel cost values associated with the situation in which no maintenance is performed throughout the year was determined as 62 415 010 MW^2 (constant available capacity) and \$605 928 459, respectively (△), which represents a potential \$46 362 075 fuel cost saving against that of Scenario A's (× in

Figure 7.17) cost ($652 290 534) — a substantial 7.11% saving. This, however, will be at the infeasible expense of scheduling absolutely no maintenance throughout the year. Similar observations are presented in [221], where the basic production cost, which represents the cost if no maintenance were to occur, is estimated at $96 909 400 (in 1975), while the maintenance schedule incurs the larger cost of $100 475 600, thus corresponding to a $3 566 200 (3%) production cost saving.



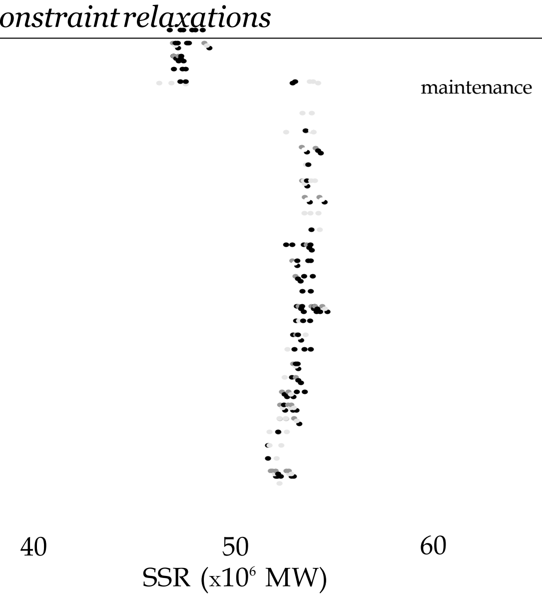


Figure 7.17: *Comparing the GMS objective space of the 32-unit IEEE-RTS inspired case study when shortening the maintenance duration time.*

7.4 Algorithmic improvements

Two alterations are suggested in this section to improve the performance of the DMOSA algorithm. The first suggestion is that the algorithm should be initialised from a good starting point, namely a minimum SSR-value as determined by an SO SA algorithm (minimising only the SSR objective function). The other improvement involves the use of parallel computing for different parallel runs of the DMOSA algorithm.

7.4.1 Minimising the SSR first

Maintenance schedules achieving high reliability tend to incur a low production cost, and *vice versa*, but a schedule corresponding to the highest reliability may not result in the lowest production cost [149, 216]. This phenomenon may be attributed to the fact that a power utility with low net reserves will have to bring its more expensive units online more frequently [216]. This is evident in Figure 7.3. Importantly, production cost-related objectives also typically

take much longer to minimise than optimising other GMS objectives (such as reliability-related objectives) [216]. In this dissertation it was found that the time taken to minimise the SSR objective (around 0.001 seconds for the IEEE-RTS related case study) was much less than the time required to minimise the fuel cost (around 1 second for the IEEE-RTS related case study) in the R programming language. Similarly, though not as extreme, it was also reported in [216] that a computation time increase of 64% was observed for the production cost objective over a reliability objective. Furthermore, a similar computation time increase of 63% was reported in [149] (166% when the model's constraints are relaxed).

Based on these observations, it is anticipated that it might be possible to find (possibly better) non-dominated fronts faster if an SO SA algorithm initially only minimises the less computationally expensive SSR objective function for a certain time period and uses its result as an initial solution for the DMOSA algorithm to further uncover non-dominated fronts based on both objective functions (*i.e.* having the MO algorithm start at a good initial solution found by an SO algorithm, instead of a random initial solution). A motivation for this suggestion is that it is an attempt to bypass the long time the DMOSA algorithm spends moving from the top right-hand corner in the objective space of Figure 7.5(a) to the desired bottom left (due to the computationally expensive production cost estimate). This suggestion will essentially cause the algorithm to spend more time computing production costs for more desirable maintenance schedules and may be very beneficial in cases where the computation time is subjected to a shorter budget. Also, this alteration may help the algorithm to navigate more quickly towards (and possibly initialising with) feasible solutions due to the nature of the GMS problem. There are, however, also drawbacks to this suggestion. For example, it may be necessary for the metaheuristic to start at a random (or even poor) solution in order to allow it to find good solutions.

Exploratory experimentation was carried out employing the above-mentioned suggestion. An SO SA algorithm was allowed 0.5 hours to find a GMS solution associated with a small SSR-value. As may be seen in Figure 7.18, the algorithm uncovered a maintenance schedule associated with an SSR-value that is very close to the best-known solution in the literature for the IEEE-RTS inspired benchmark, as reported in [169]. From this initial solution, either the DMOSA algorithm was further deployed, initialising from this good SSR-value solution (■ in Figure 7.18) or the SO SA algorithm was further employed but switched to minimising only the production cost (▲), *i.e.* running from the top left-hand corner to bottom right in Figure 7.18. This alteration to the model solution approach produced a better non-dominated front in one test case for the IEEE-RTS inspired case study than the standard model solution approach via the DMOSA algorithm, as reported in Figure 7.18.

As may be seen in Figure 7.18, the SO SA algorithm further only minimising the production cost from a good SSR-value starting solution (▲) did manage to find lower production cost solutions, but these solutions are still dominated by the solutions uncovered by the DMOSA algorithm. This result highlights the importance of employing MOO techniques, because although a spread of non-dominated solutions was found by the SO SA algorithm employed to minimise the two objectives sequentially, no notion of dominance is used to monitor and encourage the search progression. Note that the good parameter values for the DMOSA algorithm determined in Chapter 6 were also rather suboptimal for the different SO SA algorithm.

The above mentioned alteration for the initialisation of the DMOSA algorithm was further tested in respect of the 157-unit Eskom case study and the result is shown in Figure 7.19 (which is the same plot presented in Figure 6.6 for the standard DMOSA algorithm, but with the addition of the solutions obtained by this altered DMOSA algorithm). This alteration did, in fact, manage to find better non-dominated fronts than the standard DMOSA algorithm (initialised with a

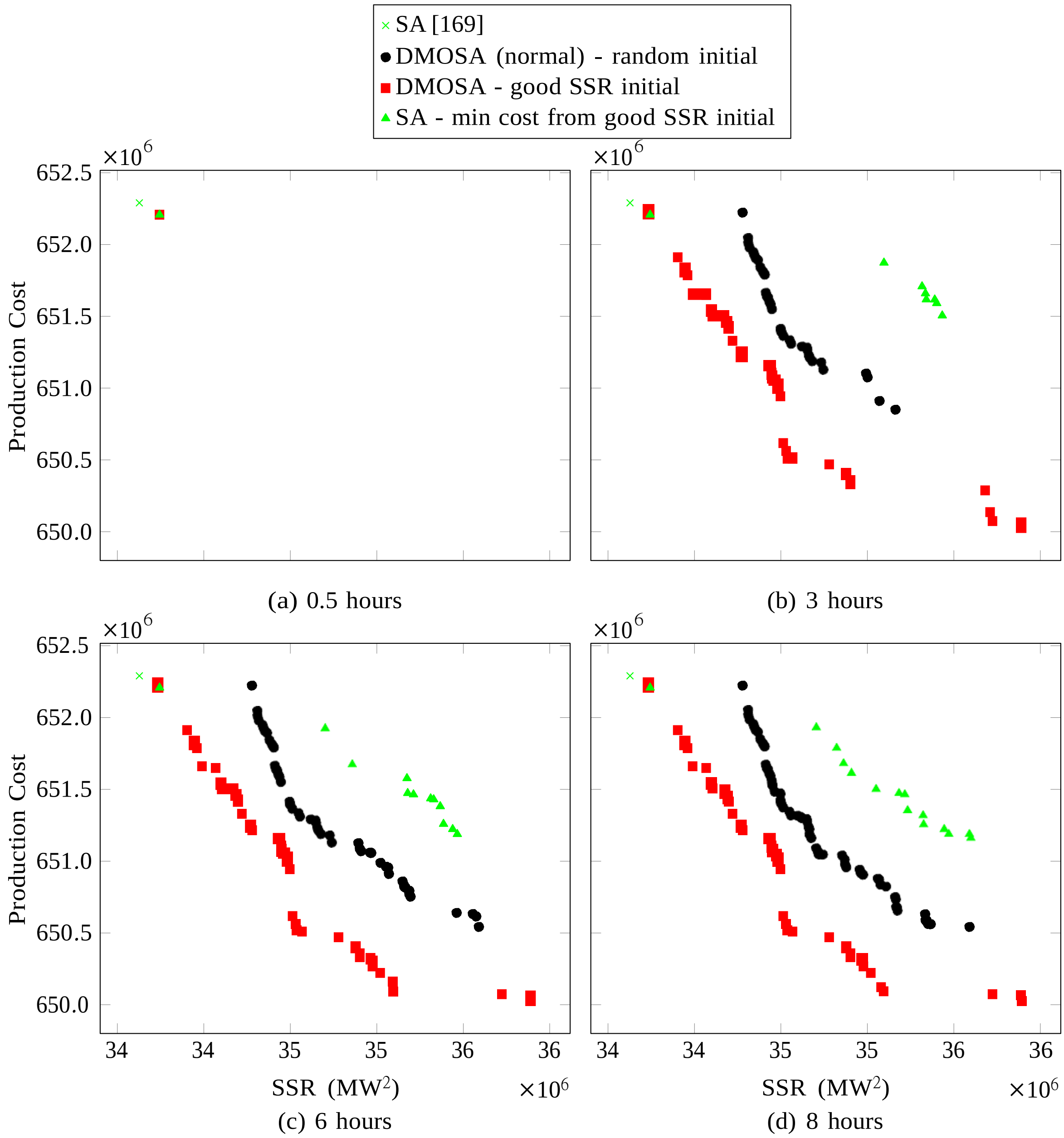
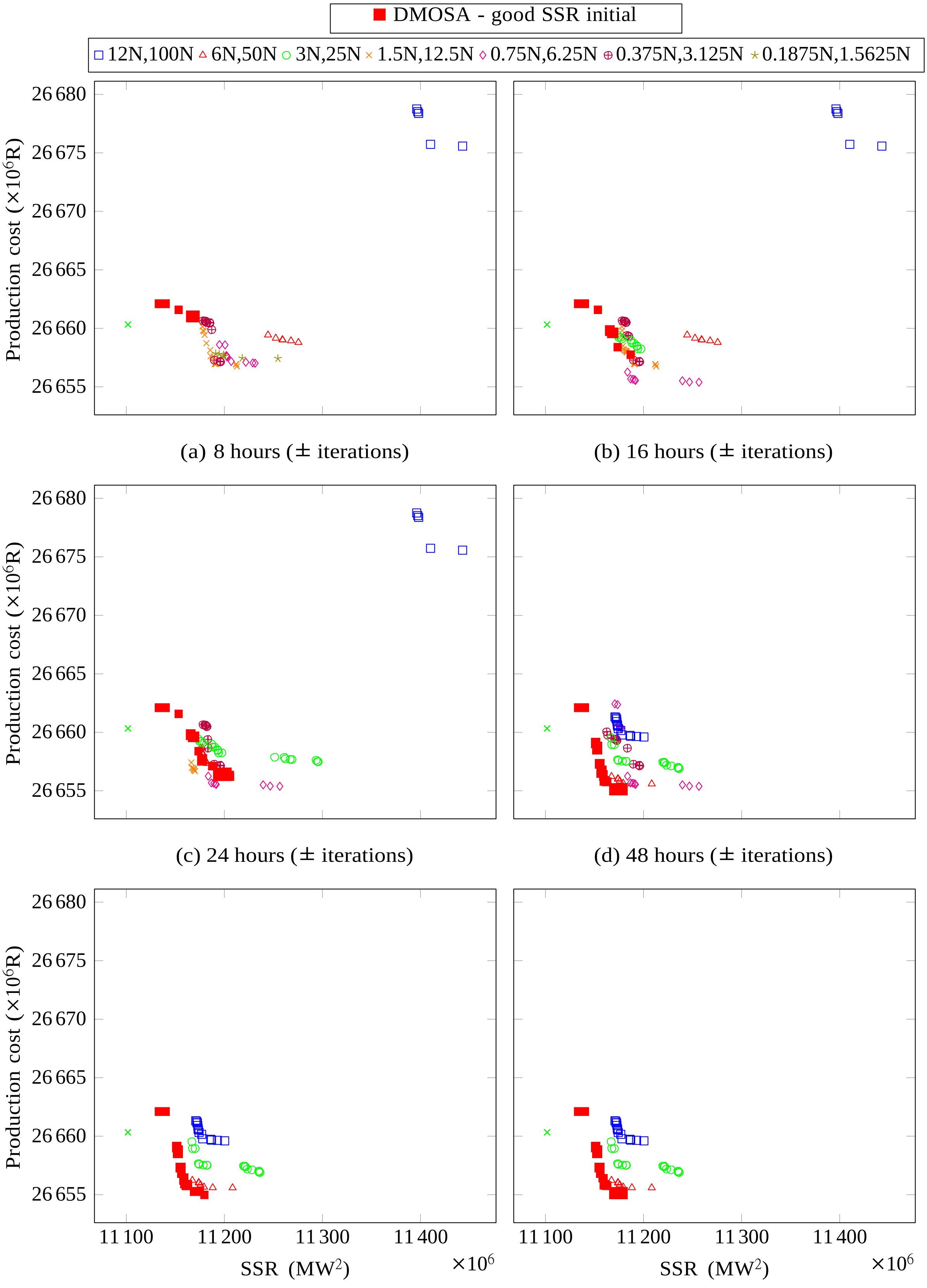


Figure 7.18: *Non-dominated fronts achieved for the IEEE-RTS inspired case study by the standard DMOSA algorithm and the improvement suggested, namely incorporating an initial SO SA algorithm to minimise the SSR whereupon either the standard DMOSA algorithm initialising with a solution achieving a good SSR-value and explores further or the SO SA algorithm switches to minimising the production cost.*

random initial solution) for some of the different epoch length stopping criteria after eight hours of computation time (see Figure 7.19(a)). The epoch length stopping criterion for this DMOSA algorithmic variation was taken as $3N$, $25N$. It was found that the shorter epoch length stopping criteria in Figure 7.19 for the standard DMOSA algorithm yield solutions that dominate the non-dominated fronts uncovered by the DMOSA algorithmic variation relatively early on during the search (see Figure 7.19(a)) but, as for the standard DMOSA algorithm (whose initial solution is generated randomly), the longer-length epoch stopping criteria eventually render results that dominate solutions returned *via* their shorter counterparts as time progresses, as may be seen in Figure 7.19.



(a) 8 hours (± iterations)

(b) 16 hours (± iterations)

(c) 24 hours (± iterations)

(d) 48 hours (± iterations)

(e) 56 hours (± iterations)

(f) 64 hours (± iterations)

Figure 7.19: *Non-dominated fronts achieved for the Eskom case study by the standard DMOSA algorithm and by the DMOSA algorithm altered according to the suggestion in §7.4.1.*

7.4.2 The use of parallel computing

Another possible avenue available for improving the DMOSA algorithm involves the use of parallel computing. Parallel metaheuristics utilise techniques of parallel programming to execute multiple metaheuristic searches in parallel so as to guide the search process more effectively. For trajectory-based metaheuristics, such as SA, three models commonly employed in the literature include [7]:

Parallel moves. This model is also called the *parallel exploration and evaluation of the neighbourhood* model and involves a low-level master-slave implementation that does not alter the behaviour of the technique. A sequential search would compute a result of the same quality, but slower. At the beginning of each iteration, the master duplicates the current solution between distributed computing nodes. Each node separately manages its candidate solutions and the results are returned to the master.

Parallel multi-starts. This model involves simultaneously launching several trajectory-based methods aimed at computing better and robust solutions. These parallel methods may be heterogeneous or homogeneous, independent or cooperative, start from the same or different solution(s), and may be configured with the same or with different parameters.

Move accelerations. In this model the quality of each metaheuristic move is evaluated in a parallel centralised way. The model is particularly interesting when performing function evaluations is CPU time-consuming and/or I/O intensive and the (constraint or objective) functions may be parallelised. In that case, the function may be viewed as an aggregation of a certain number of partial functions that may be evaluated in parallel.

Figure 7.20 illustrates the working of these different models graphically. Only the parallel multi-start model is suggested for use in conjunction with the DMOSA algorithm. Several DMOSA algorithmic instances are, in effect, simultaneously launched from different starting/initial solutions. The search runs operate independently according to their own archived fronts uncovered. A final attainment front of all the non-dominated fronts uncovered during all of the parallel runs is finally computed.

R's *foreach* [163] and *doParallel* [162] parallel computing packages were used to implement independent parallel SA algorithms with different starting solutions according to this suggestion. The number of pre-specified parallel runs may be specified by the user.

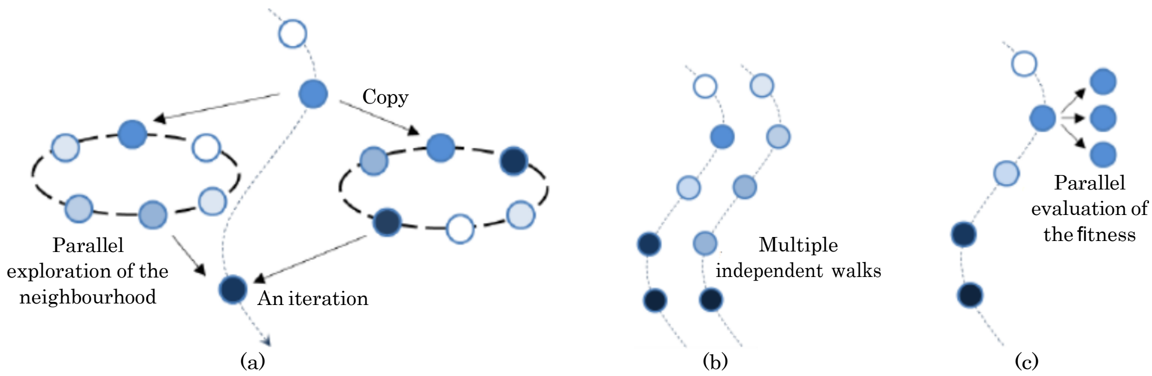


Figure 7.20: *The three classical parallel models for implementing trajectory-based metaheuristics: (a) parallel exploration of the neighbourhood (or the "parallel moves model"), (b) the parallel multistart model, and (c) parallel evaluation of fitness (or the "move acceleration model") [7].*

Figure 7.21 illustrates the improvements that are achievable by means of parallel computing, whilst Table 7.3 contains a summary of the efficiency improvements (how many iterations may be computed in total within eight hours). As may be seen in Figure 7.21, improved non-dominated fronts may be achieved by employing more parallel search runs as opposed to just one single run. Naturally, the more iterations that may be performed within the set computation time budget $t_{\max}$ (when employing parallel computing), the better the final attainment front obtained (compare □ to ● in Figure 7.21). In addition, intuitively having several stochastic DMOSA algorithms initialising from different starting solutions is likely to result in a wider search yielding a more spread-out non-dominated front. In one instance it was found that the attainment front of two parallel DMOSA algorithmic runs (+) outperformed the attainment front obtained when four parallel runs were executed (○). This is because the DMOSA search algorithm is a stochastic algorithm. A more comprehensive study is expected to alleviate this anomaly in support of the notion that the more parallel computing resources are available, the better the expected algorithmic performance.

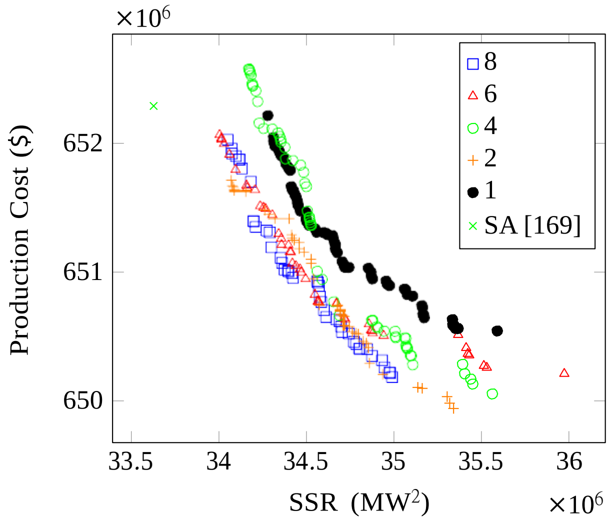


Figure 7.21: *The attainment fronts obtained by parallel DMOSA runs for the 32-unit IEEE-RTS inspired case study.*

Table 7.3: *Comparison of the speed-ups possible by employing R's doParallel and foreach packages.*

Processes	Total iterations	Total time	Time per total iterations	Speed-up
8	87 479	8 hours	0.33 sec.	5.06
6	72 146	8 hours	0.40 sec.	4.18
4	61 129	8 hours	0.47 sec.	3.55
2	32 271	8 hours	0.89 sec.	1.88
1	17 266	8 hours	1.67 sec.	1

Table 7.3 illustrates the amount of time saved (or the increase in the number of iterations possible within a fixed computation time budget) when using the parallel computing packages in R. The "total iterations" in the table represent the number of iterations performed across all the cores in the parallel search. It is observed that as the number of parallel runs increases, the speed increase is approximately linear initially, but is slightly lower than the theoretically expected value later on (*e.g.* 1.88 is slightly lower than the theoretically expected value of 2, *etc.*). Once the number

or parallel processes exceeds the number of physical cores (four physical cores in this case), the increases in performance diminishes on account of the fact that operations are performed in random access memory. This means that the hyper-threading capability is underutilised as there are fewer CPU cycles that are otherwise waiting on disk access operations [182]. This finding agrees with results obtained by others adopting a parallel computing paradigm in R [18]. The performance intricacies of parallel computing are complex and lie deep within the field of computer science.

7.5 Chapter summary

This chapter was devoted to a presentation of the results obtained by the DMOSA algorithm for the bi-objective GMS model. This included a verification and validation of the energy production plan model described in §3.3 in conjunction with the performance of the DMOSA algorithm with two *off-the-shelf* NSGA-II packages. A graphical representation of the bi-objective solution space was presented for the 32-unit IEEE-RTS inspired case study. An analysis was further performed in respect of two extremal schedules obtained by optimising the two scheduling objectives separately in the model of Chapter 3, for both the 32-unit IEEE-RTS inspired case study and the Eskom case study, illustrating some of the reasons for conflicting schedules. Furthermore, a sensitivity analysis was performed in respect of the effects of the GMS model constraints, as well as the effect of considering these constraints as another (composite) objective to be optimised within the MO GMS paradigm. The chapter closed with two suggested improvements to the DMOSA algorithm, namely including an initial SO SA algorithm aimed at finding good initial solutions for the DMOSA algorithm and adopting a parallel computing approach.

CHAPTER 8

Decision support system

Contents

In this chapter, a computerised DSS is presented for solving instances of the GMS problem in an MO paradigm. The chapter opens with a discussion on the basic notions that have to be borne in mind when developing DSSs. The relatively scant work in the literature related to DSSs in the context of the GMS problem is presented, as well as an overview of DSSs currently in use in the energy industry. The design and implementation of a novel DSS for GMS is proposed, and this is followed by a description of a concept demonstrator of the proposed DSS. Feedback from Eskom after having been presented with this concept demonstrator is also reported.

8.1 Basic notions in decision support systems

A DSS may be defined as any computer-based *information system* that supports decision-making activities [157]. Power [157] describes a DSS as an "interactive computer-based system intended to help managers make decisions." Some authors have extended the definition of a DSS to encompass any system that makes some contribution to decision making [181]. A DSS departs from the traditional management information system in that it emphasises the support of decision making in all its steps, although the decision is still finally made by the decision maker [116, p. 31]. Sprague Jr. [181] chose rather to define a DSS by its characteristics. These characteristics are that it

- is more applicable to the less well-structured, underspecified problems that managers often face,
- will combine traditional data access and retrieval functions with the use of models or analytic techniques,
- incorporate features which make it easy to be operated by noncomputer users in an interactive mode, and

- focus on flexibility and adaptability in order to allow for changes in the environment and the user's decision making approach.

Power [156, p. 6] also defined three major characteristics of a DSS, namely that DSSs

- are especially designed to assist in operations,
- should support decision making, rather than automate it, and
- should be capable of responding quickly to the changing needs and environment of the decision maker(s).

Power [156, p. 16] further defined the following five broad categories of DSSs:

Communication-driven DSSs. This type of DSS supports multiple users working on a shared task. Examples include tools like Google Docs or Groove music.

Data-driven DSSs. This type of DSS (also called a data-orientated DSS) emphasises access to and manipulation of a time series of internal (and sometimes external) data. These systems include file drawer and management reporting systems, data warehousing and analytical systems, executive information systems, and spatial DSSs.

Document-driven DSSs. This is a new type of DSS, which retrieves, manages, and manipulates unstructured information (examples including policies and procedures, catalogs, product specifications, corporate records, important correspondence, and corporate historical documents such as minutes of meetings) in a variety of electronic formats. Some authors call this type of system a knowledge management system. A search engine is, for example, a powerful decision-making tool associated with a document-driven DSS.

Knowledge-driven DSSs. This type of DSS imparts specialised problem solving expertise, usually stored as facts, procedures, and rules to the user. A related concept is *data mining*.

Model-driven DSSs. This type of DSS emphasises access to and manipulation of a financial, statistical, optimisation, or simulation model. Model-driven DSSs use data and parameters provided by users to aid decision makers in analysing a scenario — they are not necessarily data-intensive. Dicodess [55, 93] is an example of an (open source) model-driven DSS generator.

Operations researchers primarily focus on optimisation and simulation models as the "real" DSSs [157]. Power [157] further differentiates between enterprise-wide DSSs and desktop DSSs. An enterprise-wide DSS is connected to large data warehouses and serves many users. Alternatively, a desktop (or single-user) DSS is a small system that is implemented and runs on an individual user's PC [157].

Usually optimisation software packages and DSSs built around them are commonly implemented as single-user desktop packages [157]. The DSS proposed in this dissertation is a desktop model-driven DSS for the GMS problem.

The three fundamental components of a DSS architecture are: the *database* (or *knowledge base*), the *model* (*i.e.* the user criteria and decision factors), and the *interface* to the user. The users themselves are also critical components in the DSS architecture [156].

The *user interface* (UI), sometimes referred to as the *human-computer interface* or the *man-machine interface*, is the space where interactions between humans and machines occur [208]. This occurs especially in the industrial design field of *human-machine interaction* [208], which is also sometimes referred to as *human-computer interaction*, *man-machine interaction* or *computer-human interaction* [204]. The tools used for including human factors in the interface design are typically developed based on a knowledge of computer science, such as computer graphics, operating systems, and programming languages [208].

There any many types of UIs, one of the most important being a *graphical user interface* (GUI). Nowadays, the expression GUI is attributed to *human-machine interfaces* on computers, as nearly all of them now use graphics. Furthermore, due to the increased use of personal computers and society's decreasing awareness of heavy machinery, the term UI is also just generally assumed to mean the GUI, whilst in the industrial control panel and machinery control design environment, for example, the term *human-machine interface* is preferred [208].

8.2 DSSs in energy and maintenance problems

The only literature found on a DSS designed specifically for the GMS problem is the work by Schlünz and van Vuuren [168, 170], on which this dissertation builds. The solution approach employed in the DSS in [168, 170] was the single-objective SA algorithm attempting to minimise the SSR. A patent was, however, also found by the author entitled "A decision support system (dss) for maintenance of a plurality of renewable energy generators in a renewable power plant" [219].

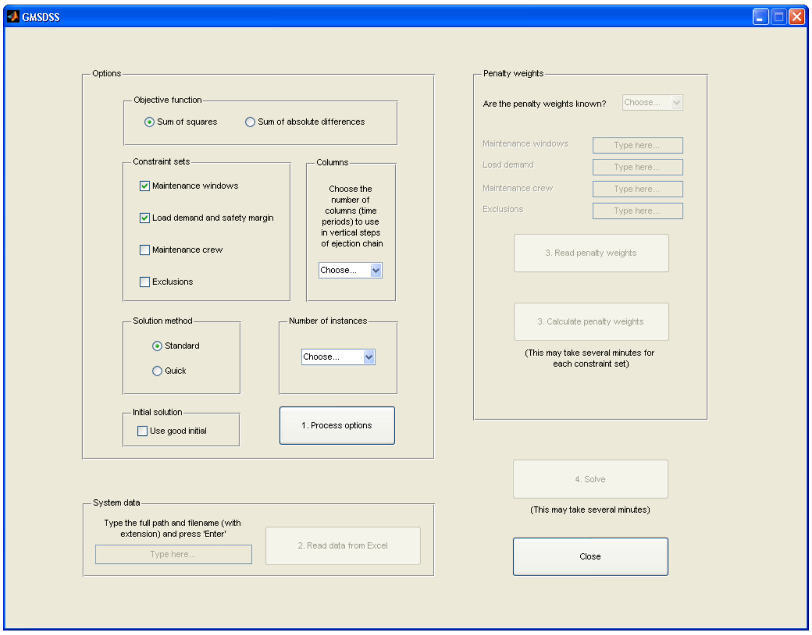
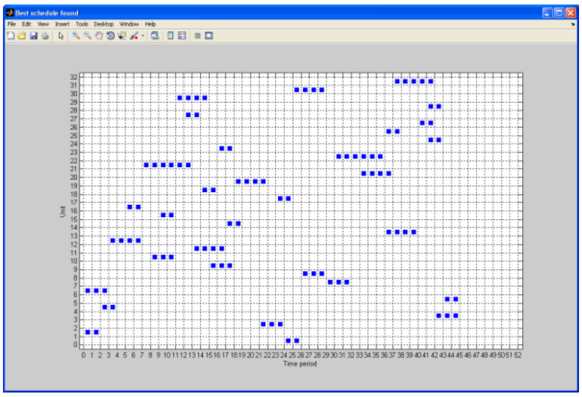


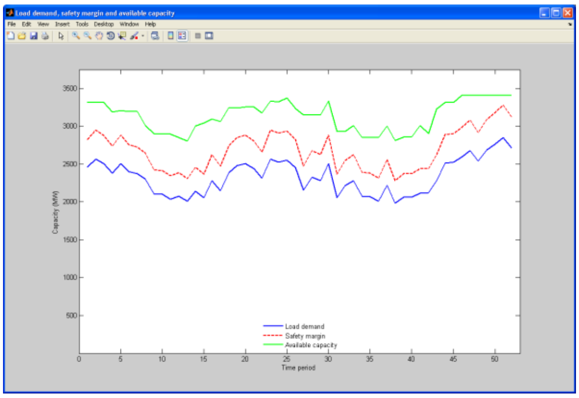
Figure 8.1: *Screenshot of the GUI presented to the user by the GMS DSS in [168, 170] upon launching the system, developed in MATLAB.*

The DSS proposed by Schlünz and van Vuuren [168, 170] was implemented in the MathWorks software suite MATLAB (version R2009a) [188]. The DSS in essence consisted of a collection of MATLAB script files with a corresponding GUI. A screenshot of the GUI presented to the user upon launching the GMS DSS is presented in Figure 8.1. Results related to the generator maintenance schedule returned by the SA algorithm (in terms of minimum SSR-values) is presented in Figure 8.2. These results include the actual maintenance schedule (see Figure 8.2 (a)), together with its corresponding available capacity versus the load's peak demand (see Figure

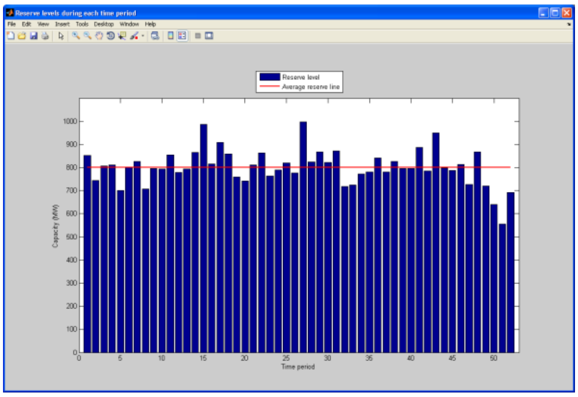
8.2 (b)), as well as its net reserve levels (see Figure 8.2 (c)), the maintenance crew utilised (see Figure 8.2 (d)), and the number of units in simultaneous maintenance within each exclusion subset (see Figure 8.2 (e)).



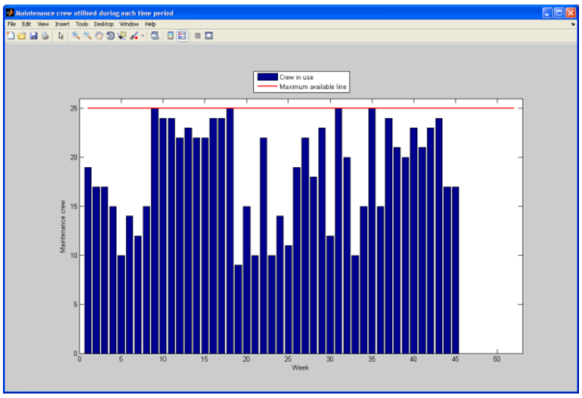
(a) Best schedule found



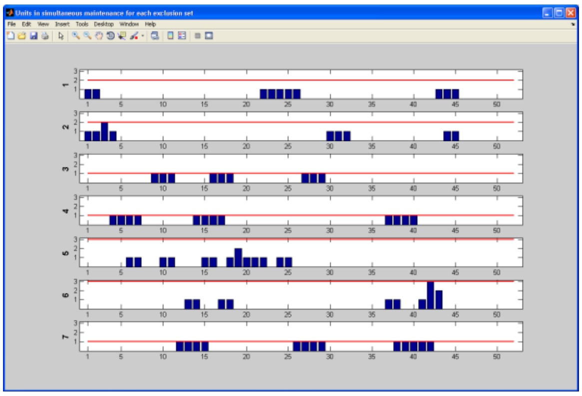
(b) Demand and available capacity levels



(c) Reserve levels



(d) Maintenance crew utilised



(e) Number of units in simultaneous maintenance

Figure 8.2: *Examples of the output figures generated by the DSS in [168].*

There has been much more activity related to the development of DSSs for general maintenance scheduling applications including, for example, ROBODOC, a diagnosis and maintenance consultant, which is an off-line decision support utility that helps maintenance staff diagnose and correct robotic failures [155]. Or [153] proposed a simple DSS for maintenance planning in the context of a large foundry satisfying casting demand for the automotive industry.

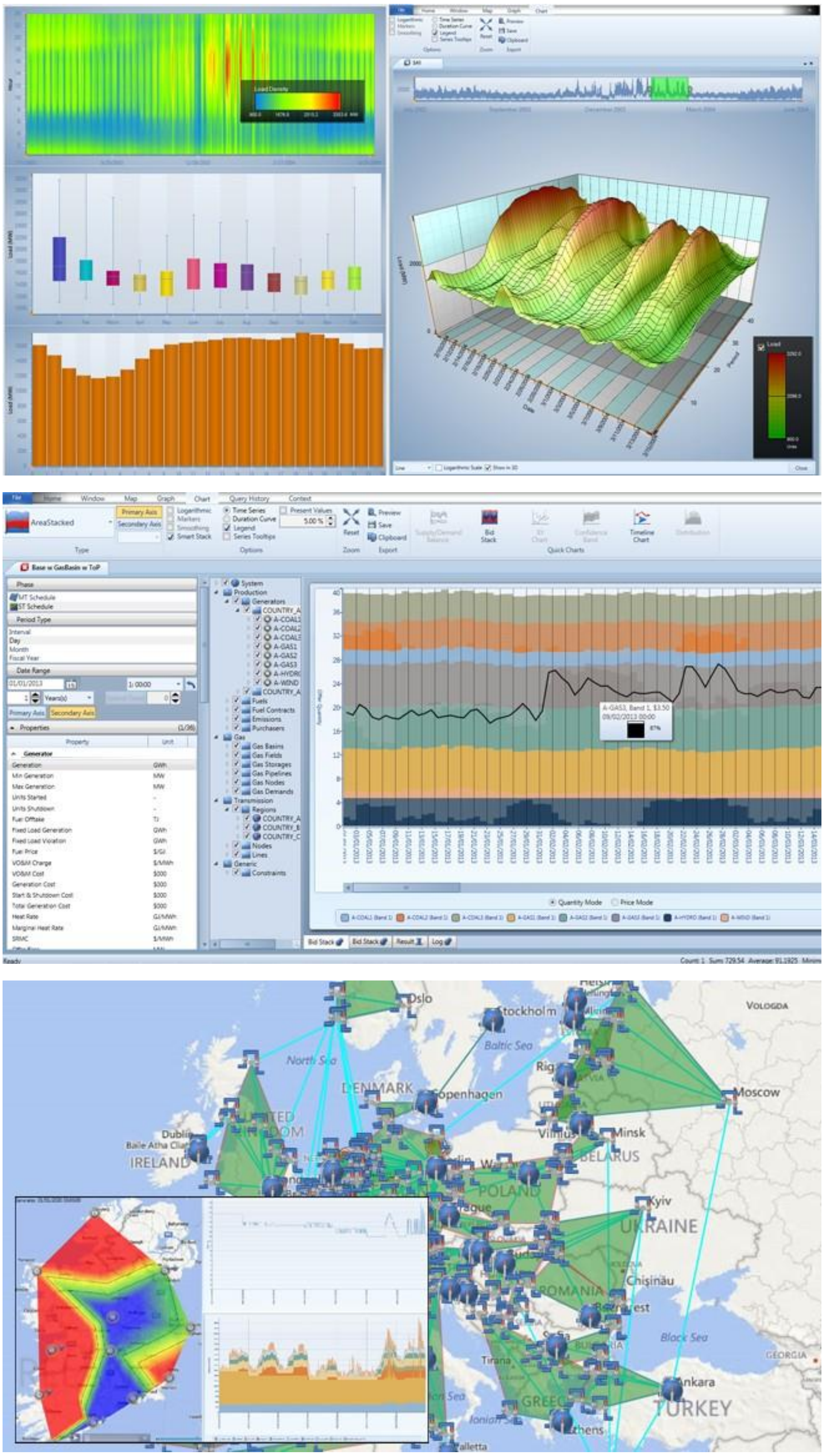


Figure 8.3: *Screenshots of the PLEXOS® software's GUI [81].*

Naturally there are also DSSs for GMS and energy-related problems in industry. Foley *et al.* [85] provided an overview of electricity systems modelling techniques and discussed a number of key proprietary electricity systems models used in the USA and Europe. These systems include AURORAxmp, EMCAS, GTMax, PLEXOS, UPLAN, WASP IV and WILMAR. The most notable of these is the PLEXOS software tool, which is claimed to be the *"the most widely used commercial integrated energy market software in the world"* [80]. It has also been applied in the South African and Sub-Saharan African contexts [80, 54].

The PLEXOS® Integrated Energy Model software for power systems was developed and is supported by Energy Exemplar®. It is an integrated electric power/water/gas simulation software tool which models, optimises (mainly through the use of mathematical programming techniques) and integrates many energy-related problems. Stochastic simulation techniques, such as Monte Carlo simulation, are also utilised. The software is equipped with a modern GUI. Examples of screenshots of this GUI are shown in Figure 8.3.

The features of the model include capacity expansion planning, power generation planning, hydro-thermal co-optimisation and maintenance optimisation amongst many others [81]. It is claimed that the maintenance optimisation feature of the PLEXOS software seeks *value-based reliability* by optimising the timing of maintenance events with respect to all system costs and accounting for constraints such as crew limits [81]. It is mentioned that this may be achieved by equalising capacity reserves across all peak periods: hourly, daily or weekly [199]. The author could find no further public information on how the maintenance planning optimisation of the PLEXOS software works.

8.3 Eskom's Tetris maintenance planning tool

Eskom uses their own *"flexible outage schedule optimisation tool,"* called the *Tetris* maintenance planning tool, for determining power generating plant maintenance [78, 84, 74]. This tool is aptly named since it mimics the game Tetris, in which a player is required to fit differently shaped building blocks into a constrained space, ensuring they are correctly orientated, with the aim of preventing the tower of blocks from growing too high. Maintenance planners may adopt this Tetris analogy, associating the blocks in Tetris with planned shutdown periods [78]. If the "tower" of planned shutdowns grows larger than a pre-defined constrained space then load shedding will have to be implemented. An overview of Eskom's proposed maintenance plan (sourced form [138]) for the first quarter of 2016 is shown in Figure 8.4. This figure illustrates the analogy of planned maintenance of generating units to the game of Tetris.

According to Eskom, the critical benefit of this Tetris planning tool over the previously used method of planning maintenance (the author is not aware of the nature of this previous method) is that it is more visual, user-friendly and operates in real-time. The system is an automated online system that planners may immediately consult should any of the variables in the system change. In February 2016, Eskom stated that "*The power system has been stable throughout the last quarter resulting in almost 6 months of no load shedding. This is due to the Tetris maintenance planning tool — supporting the execution of more planned maintenance without load shedding*" [78].

The author is not aware of what optimisation strategies are employed in the Tetris planning tool (if any). The tool is used more as a graphical tool for visualisation of proposed maintenance schedules. As may be understood from the description of this maintenance planning tool and the illustration in Figure 8.4, it visually highlights how well the capacity lost due to a planned maintenance schedule corresponds to the expected peak demand over the planning horizon (the

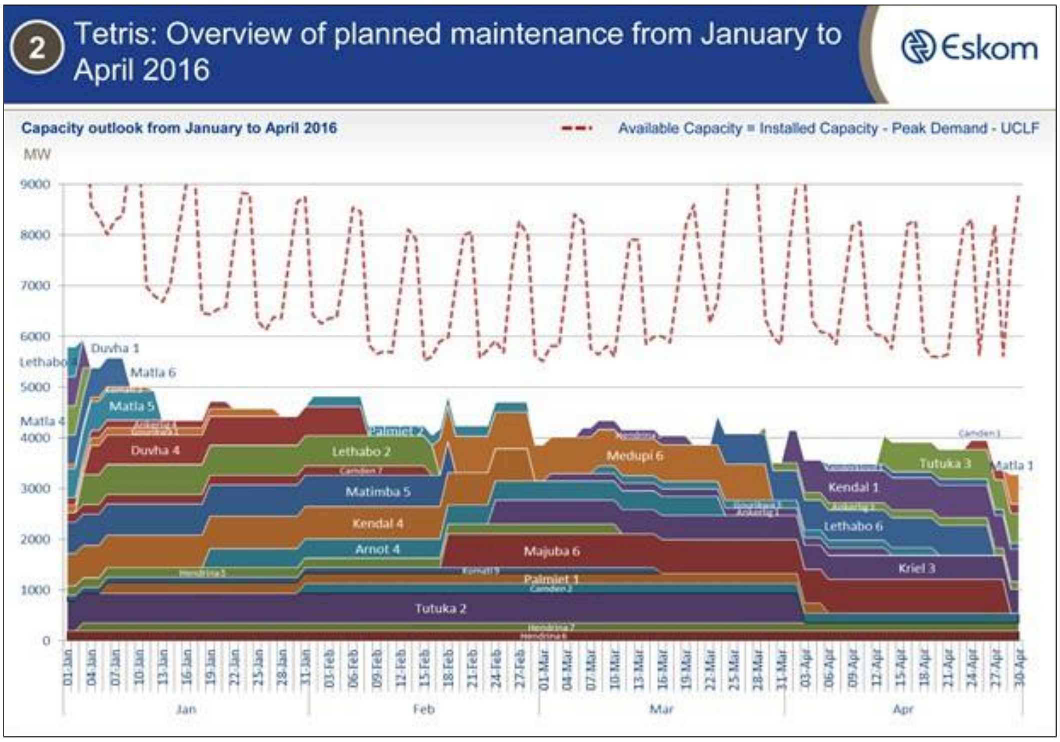


Figure 8.4: *A Maintenance plan generated by Eskom, illustrating the visual analogy between the software and the game of Tetris [138].*

SSR minimisation objective seeks to spread out this quantity evenly over time), whilst satisfying the demand constraint that there should be enough available capacity (thus avoiding a necessity for load shedding).

8.4 Proposed decision support system

The DSS concept demonstrator proposed in this dissertation is designed so that the user may solve the GMS model described in Chapter 3 by means of the DMOSA algorithm reviewed in Chapter 4. The user may vary the many different parameters of the GMS and ED models in a fashion similar to that in the Eskom EFS's current weekly production planning DSS. In addition, the user may also vary some key parameters of the DMOSA algorithm.

8.4.1 System development

The RStudio project (the most popular IDE for the R programming language) [189] developed and supports the package *Shiny* [36] for R [190]. *Shiny* is an elegant and powerful web framework for R, typically used to building interactive reports and visualisations in R — with or without web development skills [189]. The package may be used to create a UI that changes dynamically, based on R script files, and was adopted in the GMS DSS concept demonstrator. Eskom's current EFS structure also utilises R's *Shiny* package.

Furthermore, the relatively recent *googleVis* R package [97] is used to interface with the Google Charts API (abbreviation for *application programming interface*), allowing for the easy implementation of interactive charts (based on data frames). This interactive chart capability is employed so that the user may interactively and visually analyse the results returned by the SA algorithm embedded within the DSS. The current Eskom EFS structure does not have any interactive chart capabilities. A demonstration[1] of the proposed GMS DSS is available online [130].

	i		I_i	d_i	e_i	ℓ_i			k	K_k	μ_i^1	μ_i^2	μ_i^3	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	UnitNumber	UnitName	Capacity	Duration	Earliest	Latest	Station_Name	Station_ Number	Exclusion _Subset	Allowed_in _Subset	Crew_1	Crew_2	Crew_3	Crew_4
2	1	1	50	14	1	350	Hydro	25	6	3	6	6	6	
3	2	2	50	14	1	350	Hydro	25	6	3	6	6	6	
4	3	3	50	14	1	350	Hydro	25	6	3	6	6	6	
5	4	4	50	14	1	350	Hydro	25	6	3	6	6	6	
6	5	5	50	14	1	350	Hydro	25	6	3	6	6	6	
7	6	6	50	14	1	350	Hydro	25	6	3	6	6	6	
8	7	7	400	42	1	140	Nuclear	3			15	15	15	1
9	8	8	400	42	183	322	Nuclear	3			15	15	15	1
10	9	9	350	35	1	329	Coal_1	4	7	1	5	5	5	
11	10	10	155	28	1	154	Coal_2	5	5	3	5	5	5	
12	11	11	155	28	183	336	Coal_2	5			5	5	5	
13	12	12	155	28	1	154	Coal_2	5	7	1	12	12	12	1
14	13	13	155	28	1	336	Coal_2	5	7	1	12	12	12	1
15	14	14	76	21	1	161	Coal_3	6	2	2	12	12	12	1
16	15	15	76	21	183	343	Coal_3	6	2	2	12	12	12	1
17	16	16	76	21	1	161	Coal_3	6	1	2	12	12	12	1
18	17	17	76	21	183	343	Coal_3	6	1	2	12	12	12	1
19	18	18	197	28	1	154	Oil_1	7	4	1	8	8	8	
20	19	19	197	28	1	154	Oil_1	7	4	1	8	8	8	
21	20	20	197	28	183	336	Oil_1	7	4	1	8	8	8	
22	21	21	100	21	1	343	Oil_2	8	3	1	10	10	10	1
23	22	22	100	21	1	343	Oil_2	8	3	1	10	10	10	1
24	23	23	100	21	1	343	Oil_2	8	3	1	10	10	10	1
25	24	24	12	14	1	350	Oil_3	9	5	3	4	4	4	
26	25	25	12	14	1	350	Oil_3	9	5	3	4	4	4	
27	26	26	12	14	1	350	Oil_3	9	5	3	4	4	4	
28	27	27	12	14	1	350	Oil_3	9	5	3	4	4	4	
29	28	28	12	14	1	350	Oil_3	9	5	3	4	4	4	
30	29	29	20	14	1	168	Gas_Turbine	21	1	2	7	7	7	
31	30	30	20	14	1	168	Gas_Turbine	21	1	2	7	7	7	
32	31	31	20	14	1	168	Gas_Turbine	21	2	2	7	7	7	
33	32	32	20	14	183	350	Gas_Turbine	21	2	2	7	7	7	
34														

(a) Unit data file

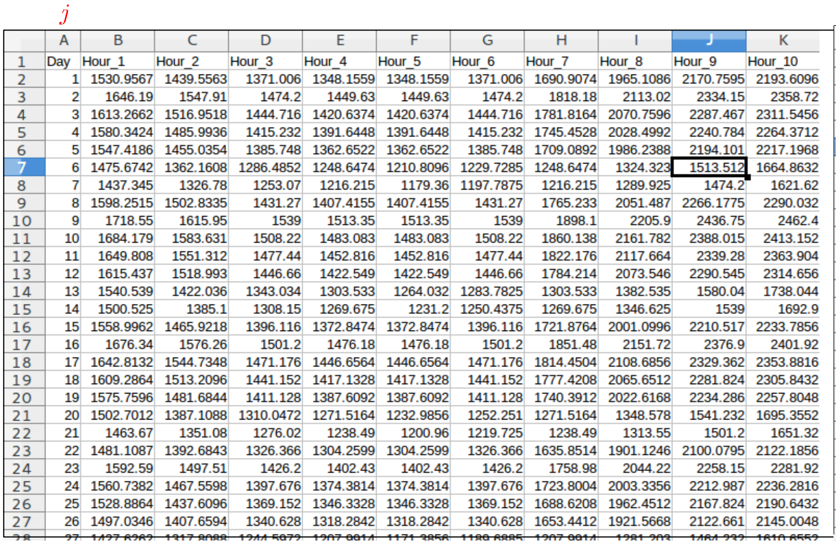
	s	h					C_s	$q_s^{\min}$	$q_s^{\max}$
	A	B	C	D	E	F	G	H	I
1	STN_NO	TYPE	NAME	NO_UNITS	UNITCAP	INSTCAP	COST	MINEUF	MAXEUF
2	1	Import	Empty	0	0	0	0	0	100
3	2	IPPS	Empty	0	0	0	0	0	100
4	3	Nuclear	Nuclear	2	400	800	6	100	100
5	4	Coal 1	Coal_1	1	350	350	11.4	60	100
6	5	Coal 1	Coal_2	4	155	620	11.64	60	100
7	6	Coal 1	Coal_3	4	76	304	14.4	60	100
8	7	Coal 1	Oil_1	3	197	591	22.08	60	100
9	8	Coal 1	Oil_2	3	100	300	23	60	100
10	9	Coal 1	Oil_3	5	12	60	27.6	60	100
11	10	Coal 1	Empty	0	0	0	0	0	100
12	11	Coal 1	Empty	0	0	0	0	0	100
13	12	Coal 1	Empty	0	0	0	0	0	100
14	13	Coal 1	Empty	0	0	0	0	0	100
15	14	Coal 1	Empty	0	0	0	0	0	100
16	15	Coal 2	Empty	0	0	0	0	0	100
17	16	Coal 2	Empty	0	0	0	0	0	100
18	17	Coal 2	Empty	0	0	0	0	0	100
19	18	Coal 2	Empty	0	0	0	0	0	100
20	19	Coal 2	Empty	0	0	0	0	0	100
21	20	Gas	Empty	0	0	0	0	0	100
22	21	Gas	GT	4	20	80	43.5	4	100
23	22	Gas	Empty	0	0	0	0	0	100
24	23	Gas	Empty	0	0	0	0	0	100
25	24	Unmet	Unmet	1	10000	10000	750	0	100
26	25	Hydro	Hydro	6	50	300	0	0	100
27	26	Hydro	Empty	0	0	0	0	0	100
28	27	Pump G	Empty	0	0	0	0	0	100
29	28	Pump G	Empty	0	0	0	0	0	100
30	29	Pump G	Empty	0	0	0	0	0	100
31	30	Pump S	Empty	0	0	0	0	0	100
32	31	Pump S	Empty	0	0	0	0	0	100
33	32	Pump S	Empty	0	0	0	0	0	100
34									

(b) Station data file

Figure 8.5: *Format of the input template files required by the GMS DSS concept demonstrator in respect of the maintenance and generating unit parameters. The values shown are those for the 32-unit IEEE-RTS inspired case study.*

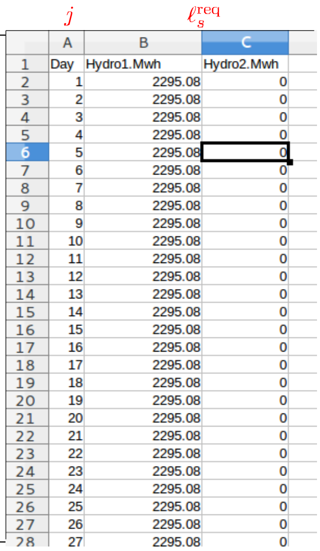
[1]No algorithmic computations may be performed. Instead, the results reported in §7.3, as illustrated in Figures 7.11 and 7.12, are presented to the user for analysis purposes.

In order to standardise procedures in the DSS, the planning resolution is taken as daily. If, however, the parameters of the model are to be based on a weekly planning resolution (as in the 32-unit IEEE-RTS inspired case study) then these values must be adapted to a daily resolution. Only *comma separated values* (CSV) files are allowed to be uploaded to the DSS concept demonstrator. Four `.csv` template files should be provided to the DSS (in an appropriate folder). Examples of the input template files are provided in Figures 8.5 and 8.6.



Day	Hour_1	Hour_2	Hour_3	Hour_4	Hour_5	Hour_6	Hour_7	Hour_8	Hour_9	Hour_10
1	1530.9567	1439.5563	1371.006	1348.1559	1348.1559	1371.006	1690.9074	1965.1086	2170.7595	2193.6096
2	1646.19	1547.91	1474.2	1449.63	1449.63	1474.2	1818.18	2113.02	2334.15	2358.72
3	1613.2662	1516.9518	1444.716	1420.6374	1420.6374	1444.716	1781.8164	2070.7596	2287.467	2311.5456
4	1580.3424	1485.9936	1415.232	1391.6448	1391.6448	1415.232	1745.4528	2028.4992	2240.784	2264.3712
5	1547.4186	1455.0354	1385.748	1362.6522	1362.6522	1385.748	1709.0892	1986.2388	2194.101	2217.1968
6	1475.6742	1362.1608	1286.4852	1248.6474	1210.8096	1229.7285	1248.6474	1324.323	1513.512	1664.8632
7	1437.345	1326.78	1253.07	1216.215	1179.36	1197.7875	1216.215	1289.925	1474.2	1621.62
8	1598.2515	1502.8335	1431.27	1407.4155	1407.4155	1431.27	1765.233	2051.487	2266.1775	2290.032
9	1718.55	1615.95	1539	1513.35	1513.35	1539	1898.1	2205.9	2436.75	2462.4
10	1684.179	1583.631	1508.22	1483.083	1483.083	1508.22	1860.138	2161.782	2388.015	2413.152
11	1649.808	1551.312	1477.44	1452.816	1452.816	1477.44	1822.176	2117.664	2339.28	2363.904
12	1615.437	1518.993	1446.66	1422.549	1422.549	1446.66	1784.214	2073.546	2290.545	2314.656
13	1540.539	1422.036	1343.034	1303.533	1264.032	1283.7825	1303.533	1382.535	1580.04	1738.044
14	1500.525	1385.1	1308.15	1269.675	1231.2	1250.4375	1269.675	1346.625	1539	1692.9
15	1558.9962	1465.9218	1396.116	1372.8474	1372.8474	1396.116	1721.8764	2001.0996	2210.517	2233.7856
16	1676.34	1576.26	1501.2	1476.18	1476.18	1501.2	1851.48	2151.72	2376.9	2401.92
17	1642.8132	1544.7348	1471.176	1446.6564	1446.6564	1471.176	1814.4504	2108.6856	2329.362	2353.8816
18	1609.2864	1513.2096	1441.152	1417.1328	1417.1328	1441.152	1777.4208	2065.6512	2281.824	2305.8432
19	1575.7596	1481.6844	1411.128	1387.6092	1387.6092	1411.128	1740.3912	2022.6168	2234.286	2257.8048
20	1502.7012	1387.1088	1310.0472	1271.5164	1232.9856	1252.251	1271.5164	1348.578	1541.232	1695.3552
21	1463.67	1351.08	1276.02	1238.49	1200.96	1219.725	1238.49	1313.55	1501.2	1651.32
22	1481.1087	1392.6843	1326.366	1304.2599	1304.2599	1326.366	1635.8514	1901.1246	2100.0795	2122.1856
23	1592.59	1497.51	1426.2	1402.43	1402.43	1426.2	1758.98	2044.22	2258.15	2281.92
24	1560.7382	1467.5598	1397.676	1374.3814	1374.3814	1397.676	1723.8004	2003.3356	2212.987	2236.2816
25	1528.8864	1437.6096	1369.152	1346.3328	1346.3328	1369.152	1688.6208	1962.4512	2167.824	2190.6432
26	1497.0346	1407.6594	1340.628	1318.2842	1318.2842	1340.628	1653.4412	1921.5668	2122.661	2145.0048

(a) *Expected hourly load demand data*



Day	Hydro1.Mwh	Hydro2.Mwh
1	2295.08	0
2	2295.08	0
3	2295.08	0
4	2295.08	0
5	2295.08	0
6	2295.08	0
7	2295.08	0
8	2295.08	0
9	2295.08	0
10	2295.08	0
11	2295.08	0
12	2295.08	0
13	2295.08	0
14	2295.08	0
15	2295.08	0
16	2295.08	0
17	2295.08	0
18	2295.08	0
19	2295.08	0
20	2295.08	0
21	2295.08	0
22	2295.08	0
23	2295.08	0
24	2295.08	0
25	2295.08	0
26	2295.08	0
27	2295.08	0

(b) *Expected daily energy (water) available for the hydroelectric stations*

Figure 8.6: *Format of the input template files required by the GMS DSS concept demonstrator in respect of the expected hourly demand and daily energy available at hydroelectric stations. The values shown are those for the 32-unit IEEE-RTS inspired case study.*

The generating unit data file for the 32-unit IEEE-RTS inspired case study is shown in Figure 8.5(a). The file should contain the GMS parameters, including each generating unit's installed capacity, the fixed duration of maintenance required, the earliest and latest starting times for planned maintenance of each generating unit, exclusion sets and parameters, and crew member parameters. The appended symbols shown in red in Figures 8.5 and 8.6 illustrate how the columns in the input `.csv` file relate to the GMS model parameters defined in Chapter 3. The data required for the fixed 32 power stations are illustrated in Figure 8.5(b). Importantly, the user has to modify the `Station_Number` column in the generating unit data file (see Figure 8.5(a)) so as to match each unit with the correct power station (a number between 1 and 32). There are a maximum of 32 stations in the Eskom EFS production planning module. If there are fewer stations in the GMS problem instance under consideration (as, for example, in the case of the 32-unit IEEE-RTS inspired case study), then the non-applicable (`Empty`) power stations' total installed capacities (`INSTCAP` column) and number of units (`NO_UNITS` column) should be set to zero (as in Figure 8.5(b)). Other parameters that may be changed in this input file include each station's cost rate and the maximum and minimum EUF values (see Figure 8.5(b)).

The expected hourly loads (MWh or MW) for each day must be uploaded to the GMS DSS in the format of the input template file provided (see Figure 8.6(a)). The DSS computes the daily (maximum) peak demand to be used in SSR calculations. The expected daily energy (water) available (MWh) at hydroelectric stations must be uploaded to the GMS DSS in the format of the input template file shown in Figure 8.6(b).

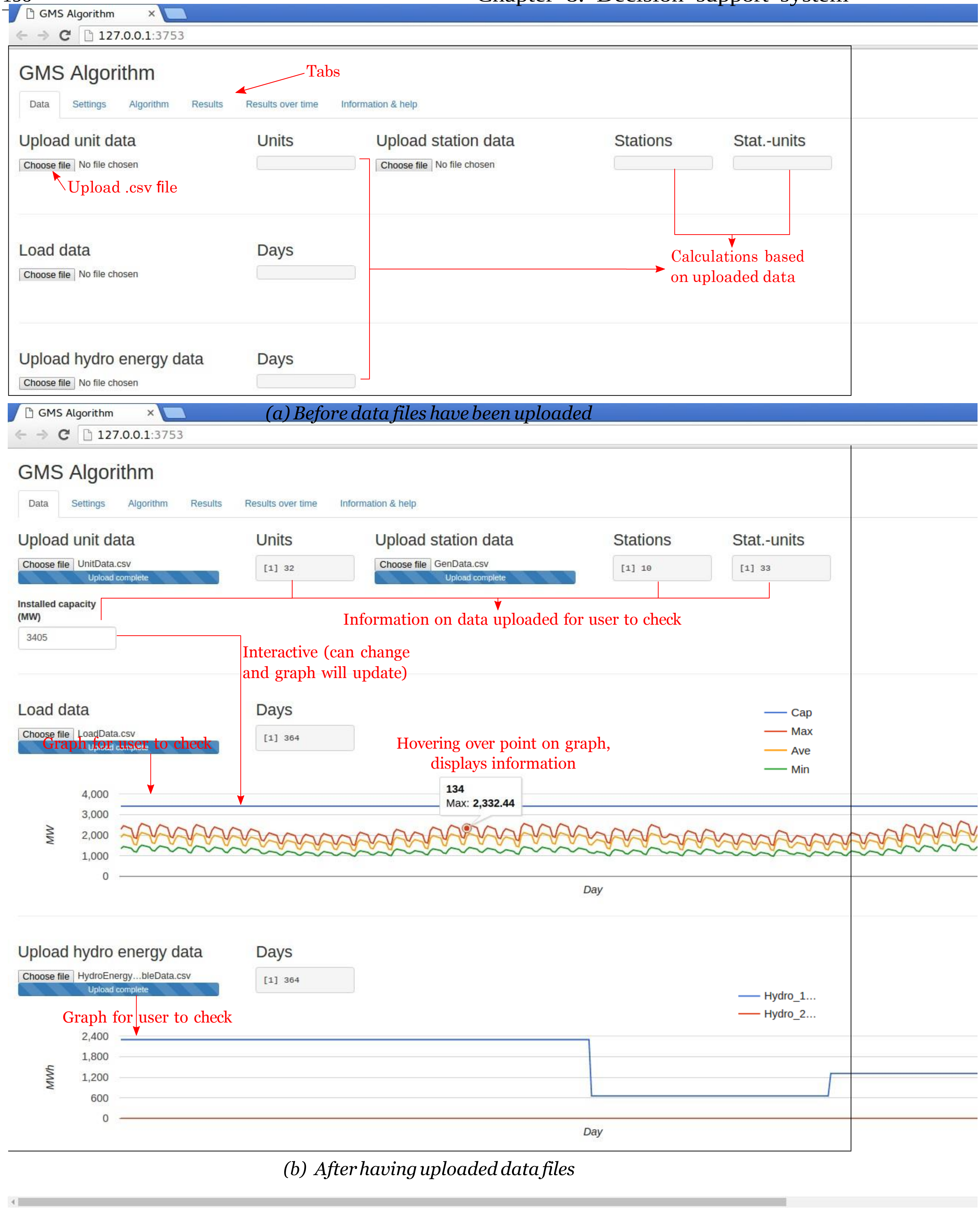


Figure 8.7: *Screenshots (with appended descriptions in red) of the proposed GMS DSS concept demonstrator's page for uploading the necessary input data (.csv files). The page contains interactive cells and graphs of the data uploaded to help the user verify the data contained and interpreted in the uploaded .csv files.*

These four sheets are conventionally uploaded to the GMS DSS (and used for further calculations) via the use of *Shiny*'s `fileInput` button, as illustrated in Figure 8.7(a). Once the files have been uploaded (as shown in Figure 8.7(b)) various output cells display key information on the files uploaded to the user, which the user must verify before continuing. For example, there are 32 rows (units) in the corresponding `.csv` files (see Figure 8.5(a)) and there are 364 rows of data in the uploaded file containing the expected hourly loads.

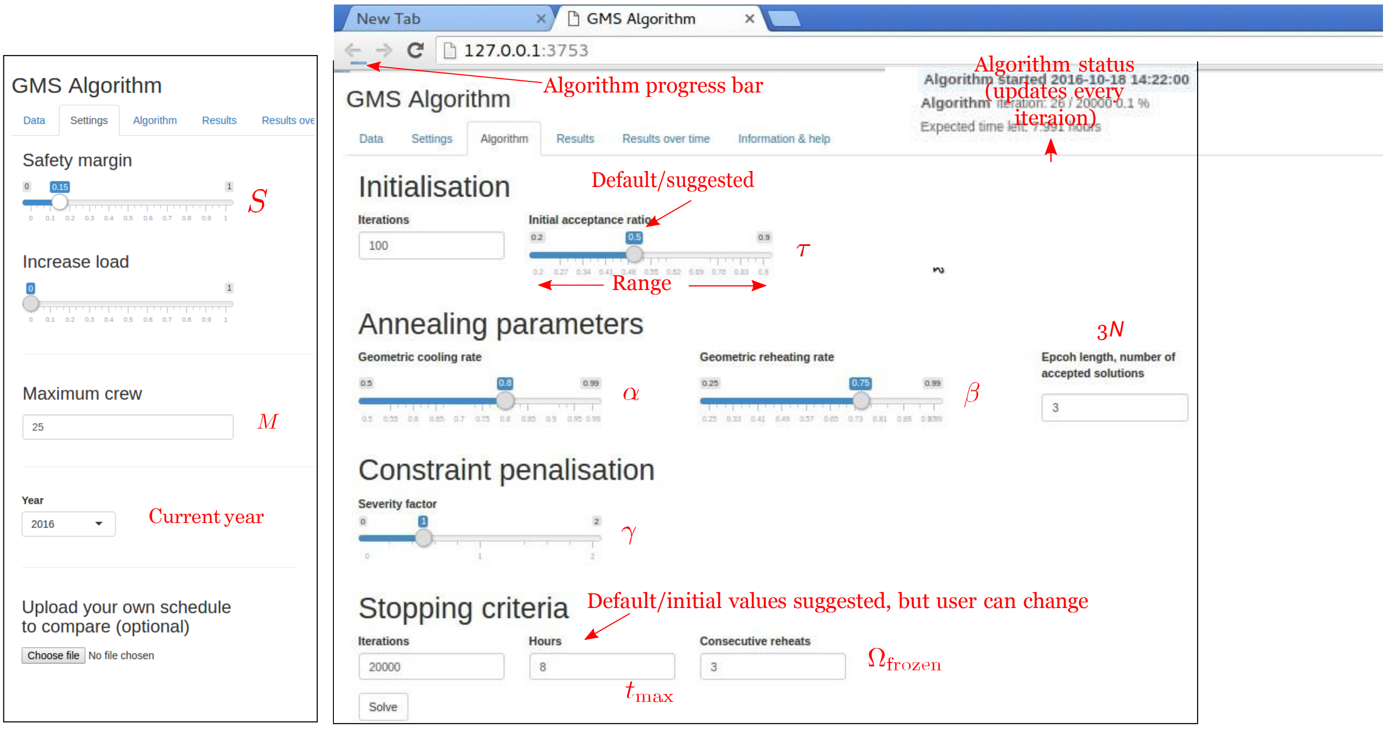
In addition, two interactive *googleVis* charts are plotted based on the load and hydroelectric data files uploaded (see Figure 8.7(b)) by the user. The first plot contains the maximum, minimum, and average hourly load demand per day. If there were any missing (*i.e.* a value of zero) or incorrect data values, this would most probably be discernible in the plot. This first plot also contains the system's total installed capacity cell value, which is initially calculated by summing the `Capacity` column in the unit data file (see Figure 8.5(a)), but may subsequently be changed by the user (in the case of dummy units or when additional power is available from other sources). All the (*googleVis*) charts in the DSS have interactive capabilities, meaning that the user may hover (move the computer's cursor over) and/or click on a chart upon which further information on the selected value is displayed.

Once the user is satisfied with the GMS data uploaded, he or she may navigate to the next tab/page which contains general and algorithmic settings (see Figure 8.8) of the GMS model and the SA algorithm. As may be seen in Figures 8.7 and 8.8, two types of *Shiny* `widgets` are used to change parameter values — either via *Shiny*'s `numericInput` element (*e.g.* the maximum crew allowed (M) value in Figure 8.8(a)) or via *Shiny*'s `slidernput` element (*e.g.* the safety margin (S) value in Figure 8.8(a)). The `slidernput` element is conveniently used when the parameter setting in question has a feasible or desired range (minimum and maximum allowed/suggested values) which the user may not violate. Initial/default values are populated in these user input fields, based on expected better-performing parameter values (such as those achieved in Chapter 5), which the user may leave unchanged or change as desired. The user may additionally (optionally) upload his or her own maintenance schedule in order to compare its performance with those of the solutions proposed by the GMS DSS (see Figure 8.8(a)).

The user may then navigate to the next tab/page which contains the settings of the DMOSA algorithm (Figure 8.8(b)). Once again the user is initially (by default) presented with suggested values for the GMS parameters, which may remain as-is or may be changed, as desired. Finally, the user may modify the three SA stopping criterion values, namely that the algorithm should terminate when either (i) the maximum number of iterations has occurred (20 000 in Figure 8.8(b)), (ii) the specified budget $t_{\max}$ of computing time has elapsed (8 hours in Figure 8.8(b)), or (iii) the specified number Ω_{frozen} of successive reheatings have been performed (3 in Figure 8.8(b)). Once any one of the three criteria is satisfied, the algorithm will terminate.

The `Solve` button (an `actionButton` element in *Shiny*) runs an `R` script file containing the DMOSA algorithm. A progress bar of the algorithmic search is displayed and updated at every iteration of the algorithm (top left of Figure 8.8(b)). In addition, more specific information on the progress of the algorithm is displayed (see top right corner of Figure 8.8(b)), including the date and time at which the user clicked `Solve`, the number of iterations performed by the algorithm, and the maximum time until completion of the algorithmic search.

Once the algorithm has terminated, all the information obtained regarding the final non-dominated solutions is stored and may be further analysed by the user in the results tab, as may be seen in Figure 8.9. To illustrate the capabilities of the interactive *googleVis* charts and tables available to the user, the non-dominated front returned by the DMOSA algorithm when all the constraints are relaxed for the 32-unit IEEE-RTS inspired case study in §7.3 (illustrated in Figures 7.11 and 7.12) is used as an example in Figure 8.9. As may be seen in Figure 8.9,



(a) *General settings* (b) *SA parameters*

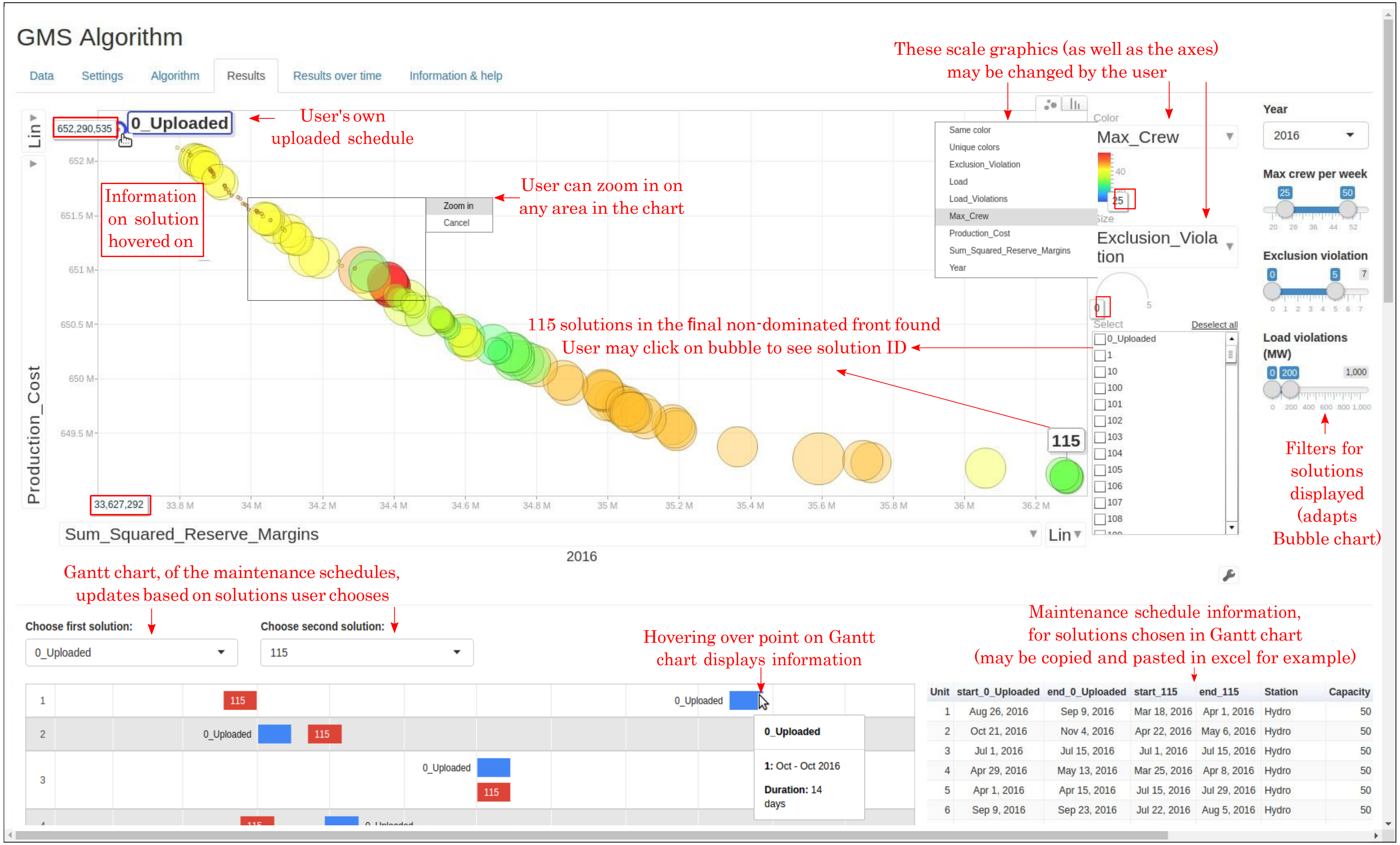
Figure 8.8: *Screenshots (with appended descriptions in red of the proposed GMS DSS's pages) of the settings the user may change. Initial/default values are suggested which the user may or may not choose to change. In some instances, a range of suggested or allowed values are represented in slider format.*

the final non-dominated front is plotted using the `gvisMotionChart` function in the R *googleVis* package. In this *Motion*[2] chart, the user may change the horizontal and vertical axes (by default the SSR and production cost values are associated with these axes, as may be seen in Figure 8.9), as well as the scale graphic values, *i.e.* the colors and sizes of the (bubble) points (in this case the maximum crew required and the exclusion violation values, respectively), as illustrated in Figure 8.9. If any solution in the front is hovered over, its values on the four axes (horizontal, vertical, color scale, and size scale) are displayed as may be seen by the four -values in Figure 8.9. Another very convenient feature of the *googleVis* Motion chart is that the user may zoom in and out with ease on a desired portion of the non-dominated front in order to further inspect certain areas of the non-dominated front (see Figure 8.9).

Using *Shiny*'s `sliderInput` element, the user may filter the GMS constraint violations or values allowed (top right of Figure 8.9) which will update the data frames from which the *googleVis* Motion chart is created, thereby interactively removing/adding solutions based on the user's preferred constraint violations or values allowed. In addition, each solution is provided with a unique *identification* (ID), in this case a number (115 non-dominated solutions were returned by the algorithm for this example). Furthermore, if the user uploads a maintenance schedule, its corresponding GMS constraint and objective function values are also plotted for comparison purposes. The solution uploaded in Figure 8.9 (ID is `0_Uploaded`) is, in fact, the solution obtained by Schlünz and van Vuuren [169] who used a SO SA algorithm to minimise only the SSR objective.

Referring to the unique ID attached to each solution in the non-dominated front, the user may

[2]The reason why it is called a Motion chart is because the chart incorporates a time dimension, whose values are slider inputs that the user may drag (or press play) upon which the chart will update.



Unit	start_0_Uploaded	end_0_Uploaded	start_115	end_115	Station	Capacity
1	Aug 26, 2016	Sep 9, 2016	Mar 18, 2016	Apr 1, 2016	Hydro	50
2	Oct 21, 2016	Nov 4, 2016	Apr 22, 2016	May 6, 2016	Hydro	50
3	Jul 1, 2016	Jul 15, 2016	Jul 1, 2016	Jul 15, 2016	Hydro	50
4	Apr 29, 2016	May 13, 2016	Mar 25, 2016	Apr 8, 2016	Hydro	50
5	Apr 1, 2016	Apr 15, 2016	Jul 15, 2016	Jul 29, 2016	Hydro	50
6	Sep 9, 2016	Sep 23, 2016	Jul 22, 2016	Aug 5, 2016	Hydro	50

Figure 8.9: *Screenshot (with appended descriptions in red) of the 32-unit IEEE-RTS inspired case study results reformed by the DMOSA algorithm in pursuit of minimising both the SSR and the production cost, with the GMS constraints relaxed as was discussed in §7.3 and illustrated in Figures 7.11 and 7.12. The Motion chart, Gantt chart and the Table are all created using the googleVis package in R. These charts have a variety of interactive capabilities.*

compare any two maintenance schedules (from all the non-dominated solutions returned and uploaded) in the form of a *googleVis* `gvisTimeline` chart (bottom left of Figure 8.9 — note that this figure has been cropped, and so it does not show the 32 generating units). This chart is interactive; the user may hover over or click on a bar and further information will be provided (as seen in Figure 8.9). More specific information on the two solutions selected by the user (which are plotted in Gantt chart format) are provided in a *googleVis* `gvisTable` table (bottom right of Figure 8.9), whose only interactive capability is that it is able to sort by columns. The values in this table may be copied, and pasted into an Excel sheet, for example, for further reporting and analysis purposes.

8.4.2 Feedback received from Eskom

In September 2016, the proposed GMS model (with its assumptions, parameters, and approach), along with key results found for the Eskom case study (the results in Chapter 7), as well as a concept demonstrator of the GMS DSS was presented to a panel of experts at Eskom. This panel included the current production assurance manager [139] (who is a key decision maker in terms of the planned maintenance of generating units), other decision makers at Eskom involved in maintenance and production planning, and some stakeholders of the EFS project described in §1.2.

The panel was generally pleased with the GMS model proposed as well as with the working of the GMS DSS concept demonstrator. Some key discussion points raised by the panel included that the decision makers had not thought of the notion that maintenance of cheap (large) generating units should preferably not be planned during high electricity demand periods, which the model seeks to avoid by minimising the production cost, subject to the constraints described in Chapter 3.

It was also noted that when a power system (such as Eskom) is under severe pressure due to low gross reserves (low installed capacity compared to expected demand), the reliability of the system is usually more important (so to avoid load shedding) than other objectives, such as minimising cost. As the gross reserves, however, become larger (a situation that Eskom is seeking to achieve with its planned increase of installed capacity), the reliability of the system becomes less pressing and other objectives (such as minimising cost) may become more important. MOO is desirable in this sense as it seeks to provide a best possible set of trade-off solutions. It was furthermore stated that it would be beneficial to include another GMS objective that measures the risk of unit failure (the LOLP, for example) within the MO paradigm proposed in this dissertation. The stakeholders mentioned that they would prefer the algorithm not to run too long — preferably overnight (12–16 hours) or perhaps during the day (± 8 hours). Another important point raised was that, as the gross reserves become larger, the cost of "dumping[3]" electricity will become more prominent and should also be included in the production planning module. This may be achieved by including another virtual station (opposite in nature to the unmet virtual station) which has a certain cost rate (measured $/MWh) associated with dumping electricity and whose energy cost is included in the cost objective to be minimised.

The author is scheduled to further test, implement, and demonstrate the working of the GMS DSS in the form of a two-day workshop in November 2016 for relevant decision makers and stakeholders at Eskom interested in further using the GMS DSS. The author also plans to add further improvements to the DSS, such as a viewing capability for the available capacity versus

[3]Dumping is the general term used when a power system cannot suddenly reduce its output power (especially for nuclear units). If, for example, demand drops and this electricity must be dumped somewhere until the reactor output is reduced.

peak demand associated with a maintenance schedule (similar to Figure 7.5(b)), as well as the daily total and station production costs associated with a maintenance schedule (similar to Figures 7.5(c) and 7.6).

8.4.3 DSS deployment and maintenance

There are a few avenues available along which to deploy the GMS DSS developed in R's *Shiny* environment so that users can run it on their computers [191]. These include having the *Shiny application* (app):

Accessible over the web. In this case, the users are not required to have R and *Shiny* installed on their computers; they only need a web browser. This may be of considerable benefit to nontechnical computer users. Two possibilities are available, namely to host the app developed through the *Shiny Server* program or to use *Shinyapps.io*, which is RStudio's hosting service. The app demonstration developed by the author [130] which is available for viewing online is, in fact, hosted through the *Shinyapps.io* hosting service. Usually these hosting services are free of charge up to a certain extent after which a fee is charged.

Run locally. In this case, the user will have to have R and *Shiny* installed on his or her computer. The code required for the app may either be hosted online (for example, on a GitHub repository) which the user may download through R, or as a (.zip) file made available to the user to personally install on his or her computer. The former is more desirable in that maintenance and updates to the code may be performed more structurally in a remote fashion by the developer, which the user may then download (seen as an "update") more easily. For the latter, *i.e.* a (.zip) file (emailed or shared), maintenance and updates of the app must be sent manually to the user, which the user then has to unzip and whose files must be placed in the current apps directory or in another appropriate location. In addition, the use of a repository (such as GitHub) is of great benefit if other developers are interested in maintaining and updating the DSS.

The author plans to analyse and test the above deployment methods before implementing the DSS at Eskom in November 2016.

8.5 Chapter summary

Various basic notions related to DSSs were reviewed in this chapter. The very few studies pertaining to DSSs in the GMS literature were discussed as well as some of the more abundant DSS references available for general maintenance scheduling problems and proprietary DSS software available in the energy industry. Some basic aspects of Eskom's Tetris maintenance planning tool were also described. The proposed GMS DSS concept was finally demonstrated and industry feedback received from stakeholders at Eskom on the proposed GMS model and the DSS concept demonstrator was relayed.

CHAPTER 9

Conclusion

Contents

A summary of the research reported in this dissertation is presented in the first section of this chapter. This is followed by a brief appraisal of the dissertation contributions in §9.2, and the chapter finally closes in §9.3 with a number of suggestions for future work.

9.1 Dissertation summary

The first chapter of this dissertation provided the reader with some background to the energy industry, including electricity consumption statistics (both worldwide and in the South African context). Background information on the types and sizes of power generating units required to supply this demand was also provided. The importance of maintenance planning for generating units was highlighted, especially in South Africa. An EFS designed for a national power utility such as Eskom, was introduced. This was followed by a motivation for the real-life requirement to optimise decision variables in the EFS. This optimisation requirement provided the context for the informal description of the problem considered in this dissertation, which is to add a DSS GMS optimisation capability to such an EFS within an MO paradigm. The dissertation scope and objectives were also outlined in Chapter 1, and this was followed by a brief description of the structure of this document.

Chapter 2 opened with an introduction to the various types of maintenance strategies considered in the literature. This was followed by a description of the energy industry. General modelling considerations were then presented for the GMS problem. The remainder of the chapter was devoted to an extensive review of traditional and current state-of-the-art modelling approaches in the GMS literature. This included a survey of the limited and fairly recent MOO GMS studies available in the literature. The contents of the chapter stand in fulfilment of Dissertation Objective I of §1.3. Finally, a summary table was presented of the major GMS literature reviewed in the chapter in terms of the different GMS modelling and solution techniques adopted and highlighting the context and nature of the work carried out in this dissertation.

The GMS model adopted in this dissertation was derived in Chapter 3. This included mathematical formulations of the GMS constraints taken into consideration in addition to describing how the two GMS objectives were formulated. The chapter closed with a presentation of how the proposed model may be incorporated into an EFS, in fulfilment of Dissertation Objectives II and III.

In Chapter 4, a number of basic notions in the theory of MOO were discussed, and this was followed by a description of the various algorithms suited to GMS and the solution of MOO problems. Information on the working and implementation of the MOO metaheuristic selected for implementation in this dissertation was presented, including how the GMS constraints are handled, in partial fulfilment of Dissertation Objectives IV and V.

Chapter 5 contained a comprehensive description of the data pertaining to the two case studies conducted in this dissertation in order to test the effectiveness of the proposed GMS model, in partial fulfilment of Dissertation Objective VII.

An extensive parameter optimisation experiment was performed in Chapter 6 so as to determine suitable parameter values for the SA algorithm in the contexts of the two case studies of Chapter 5. The best parameter values thus uncovered were employed as suggested values in the DSS proposed in this dissertation, in partial fulfilment of Dissertation Objective IV.

In Chapter 7, the GMS model of Chapter 3 was verified and validated, and the performance of the DMOSA algorithm described in Chapter 4 was compared to results obtained by an *off-the-shelf* implementation of the NSGA-II, in partial fulfilment of Dissertation Objective VI. Reasons for the GMS trade-offs between the minimisation of SRR and production cost objectives were presented. A sensitivity analysis was performed in respect of the GMS constraint right-hand sides, showing to what extent the constraints affect the objective function values attainable. The chapter concluded with suggested improvements to the MOO algorithm, including the use of parallel computing. The results of this chapter formed the foundation for the design of the DSS proposed in fulfilling part the Dissertation Objectives IV and VIII.

The penultimate chapter of this dissertation, Chapter 8, contained a description and demonstration of the design and implementation of a novel DSS for solving instances for the GMS problem, in fulfilment of Dissertation Objective IV. The chapter contained a discussion on other DSSs in the literature and in industry, as well as a tool currently employed by Eskom for maintenance planning. The working of a concept demonstrator of the DSS proposed in this dissertation was presented and the method according to which the DSS suggests trade-off solutions to the user and identifies possible improvements achievable by relaxing model constraints were described, in fulfilment of Dissertation Objective V. Feedback received from Eskom after having presented them with the GMS model and DSS concept demonstrator was also reported, in partial fulfilment of Dissertation Objectives IV–VIII. The chapter closed with a discussion on how it is envisaged that the DSS will run on a decision maker's computer.

9.2 Appraisal of dissertation contributions

Six main contributions were made in this dissertation. These contributions are described and elucidated in this section.

Contribution 1 *An extensive review on the literature relating to MO GMS problems.*

Although the recent (2016) literature review paper by Froger *et al.* [90] presents the most important literature on maintenance in the electricity industry (including the GMS, TMS and fuel management problems), the author similarly detailed the major GMS literature in Chapter 2, but in addition discussed, analysed and provided illustrations of the current state of MO GMS models in far more depth. This and additional aspects for the major GMS literature was also summarised in an easily interpretable table (Table 2.9) highlighting the difference between simple MO GMS modelling approaches[1] and Pareto-based MOO approaches, which Froger *et al.* [90] (and other authors) do not recognise.

Contribution 2 *A novel multi-objective model for the GMS and ED problems.*

Another contribution of this dissertation is the proposed MOO GMS problem formulation incorporating two of the most common GMS criteria, namely reliability and cost [90] (which are usually considered separately). As mentioned in §2.7, the author could only find two references[2] in which a Pareto-based MOO paradigm was adopted in a GMS model. In 2014, Zhan *et al.* [218] proposed an MO GMS problem for deregulated systems with five objectives, namely maximising the profits of three different energy producers, maximising system reliability, and minimising the total generation cost of all three producers. The system reliability objective was formulated in terms of the standard deviation of a reliability index which pursues similar reserve capacities in all periods (*i.e.* leveling the reserve margin). The total generation cost to be minimised consisted of the fuel cost (formulated as a quadratic function), unit start-up costs, and maintenance cost. Profit in the market environment was calculated by subtracting these generation costs from the revenue for each of the three producers. One of the benchmark systems solved for demonstration purposes in [218] involved a 32-unit IEEE RTS case study similar to the one considered in this dissertation. The GMS model was solved by means of the GSOMP algorithm. Similar to the approach adopted by Zhan *et al.* [218], Chen *et al.* [38] (Zhan being a co-author) solved an almost identical MO GMS model (with one objective fewer, namely the total generation cost being excluded), but by employing the NSGA-II algorithm instead.

Although the generating cost adopted by Zhan *et al.* [218] was formulated quite extensively, including sophisticated start-up cost and expected maintenance cost formulations (in addition to quadratic fuel costs) the generators' production output levels were not optimised in any way. They were instead treated as variables assuming random values between specified minimum and maximum allowable values. The GMS model proposed in this dissertation diverges from this sophisticated MO GMS approach in that it determines unit output levels through optimisation *via* a production planning module which involves solving an LP model for the ED problem in conjunction with a simple UC logic. In addition, the model in this dissertation utilises the constraints in the generation cost minimisation LP by Brits [25] to model the amount of water available during each time period at hydro power stations and the logic for power systems containing the pumped storage scheme. The reliability scheduling objective adopted in this dissertation also involves the pursuit of levelised reserve margins, but instead employs the more popular minimisation of the SSRs.

Furthermore, varying maintenance durations were not allowed in [218] (all the units' maintenance duration values were taken as two-week periods) whereas in this dissertation these values may vary (based on the different types and sizes of units). Moreover, no manpower constraints were incorporated in [218] and there are no earliest and latest maintenance starting time constraints in [218] (these values are set at the extremes of the planning horizon).

Contribution 3 *Analysing the reason for the occurrence of trade-offs between two very common GMS objectives.*

In §7.2, the author analysed the two extremal schedules obtained by optimising the two schedul-

[1]The models contain multiple objectives but these are either optimised separately, by including the less dominant objective(s) as constraint(s), by weighting and summing together scheduling criteria into one objective function, by attempting to find objective function values that are as close as possible to some ideal point/value, or by adopting a goal programming solution approach, producing one to three trade-off solutions, as discussed in §2.6.9.

[2]Also confirmed in the literature review in [90], as discussed in §2.7.

ing objectives in the model of Chapter 3 separately, thus motivating the desirability of pursuing trade-offs between these objectives instead of adopting one of them. The author compared the changes in these two very common GMS objective functions (namely the SSR and production cost) obtained for the two extremal scenarios in the non-dominated front. These extremal schedules and their corresponding effects on the reserves and the total and station-specific production costs per time period were also analysed. This type of analysis has not been caried out in the literature within an MOO paradigm in the GMS context. It was found that the SSR objective produces solutions that also obtain a low production cost (in much less computing time). The general sentiment in literature is that schedules with high reliability tend to achieve low production costs, and *vice versa*, but the schedule that gives the highest reliability may not be one that achieves the lowest production cost [149, 216]. This may be attributed to the fact that a power utility having low net reserves will have to bring its more expensive units online more frequently [216], as was shown in an MO modelling paradigm.

Contribution 4 *Development and implementation of an MO GMS DSS.*

A further contribution of this dissertation is a computerised DSS capable of computing GMS solutions within an MO paradigm. Based on the user's input requirements, this DSS has been designed to be used in conjunction with an EFS, and specifically in conjunction with a power consumption module (for determining the expected loads) and the improved production planning LP by Brits [25], as described in §3.4 and elucidated in Figure 3.4. This dissertation also marks the first research project adding an optimisation capability to an EFS in which a DSS was developed (the two previous projects by Brits [25] and Hatton [104] did not propose a DSS concept demonstrator).

Contribution 5 *A sensitivity analysis in respect of GMS constraints within an MO modelling paradigm.*

Sensitivity analyses were performed in respect of the relaxation of some of the model constraints in §7.3 (specifically the maximum crew constraint) as well as incorporating the constraints to be satisfied as an objective to be minimised within an MO paradigm. This analysis illustrated the effect of shortening the fixed maintenance duration required in respect of both the reliability and cost objectives. In addition, it was shown in §7.3 that some constraints, such as the earliest and latest maintenance starting times, limit the range of solutions available and thus ultimately limit the density and spread of non-dominated solutions attainable in objective space. The author is not aware of any GMS literature demonstrating how these GMS constraints affect GMS solution quality and efficiency in an MO paradigm. This analysis may also be used in future to help distinguish which GMS criteria should be included as scheduling objectives and which should be defined as fixed constraints to be satisfied.

Contribution 6 *Appending data pertaining to two GMS case studies.*

Data were appended to the 32-unit IEEE-RTS inspired case study, a benchmark system in the literature, as well as to the 157-unit Eskom case study. These appended test instances now represent more realistic studies. The data appendices include linear production cost rates and minimum and maximum EUF values required to solve the ED problem.

9.3 Suggestions for future work

Eight suggestions are made in this final section in respect of possible future work following on the research reported on this dissertation.

Suggestion 1 *Incorporate stochastic reliability (or risk) measures in a GMS model formulation.*

An important aspect not included in the scope of this dissertation was the risk of unit failure (except through requiring a deterministic safety margin over and above the peak demand). Including a more explicit measure of the maintenance schedule in terms of its risk of failure to the power system as a scheduling criterion in a GMS model should be an interesting avenue of future investigation. Many such measures exist in the GMS literature, such as the LOLP or EUE. These measures also take into account the stochastic nature of the load demand, something not done in this dissertation.

It would also be of interest to incorporate the risk of unit failure (unplanned outages) either as a GMS objective in an MO modelling paradigm, or as a constraint to be satisfied. This notion was also expressed during the panel discussion with Eskom described in §8.4.2. Furthermore, it is suggested that the risk of unit failure influences the right-hand sides of the earliest and latest starting times for maintenance to occur. A proposal for the inclusion of such an objective or constraint to the GMS model and its interaction with an EFS is provided in Figure 9.1.

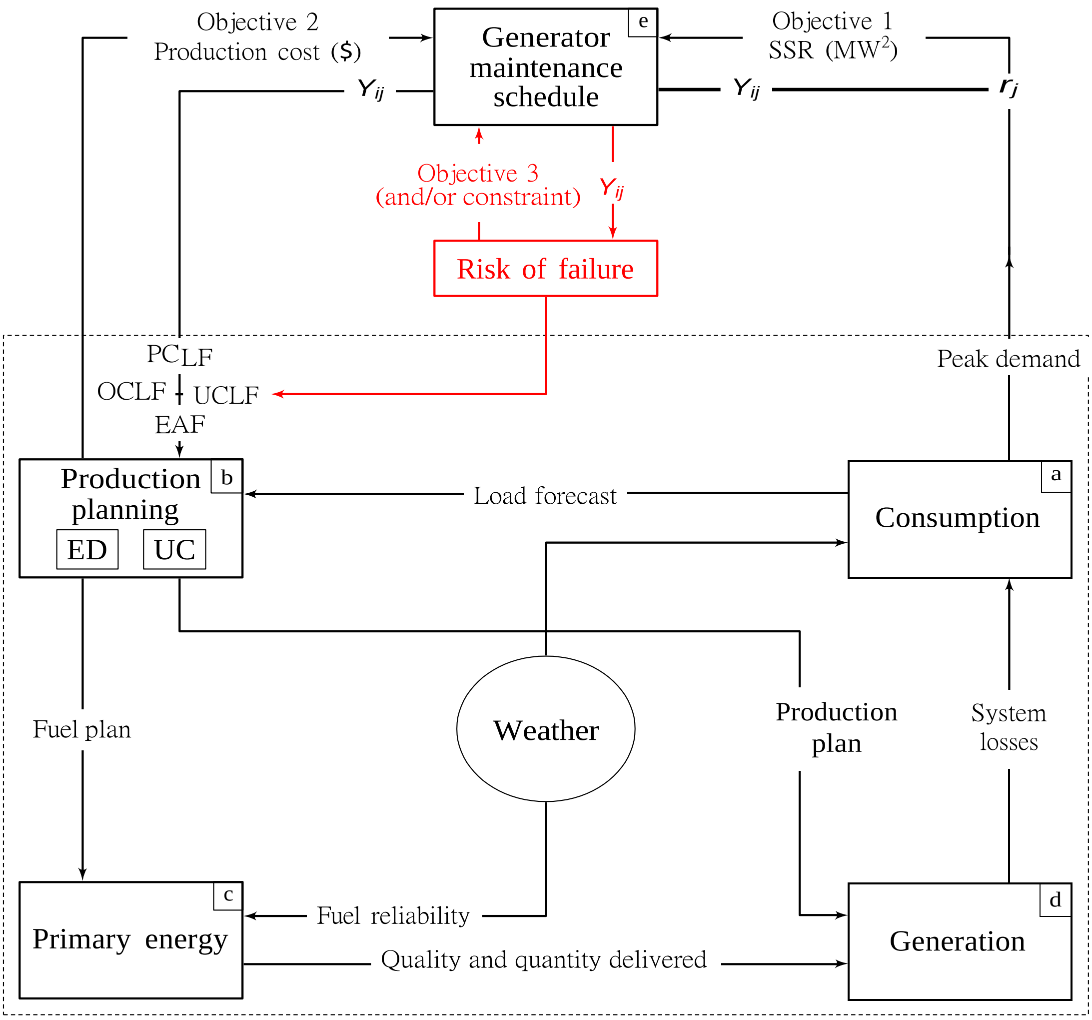


Figure 9.1: *High-level representation of the working of an EFS (dashed area) and how the GMS approach proposed in this dissertation should form part of it. This structure is appended (in red) in order to demonstrate how Suggestion 1 may be incorporated into the simulation framework if the risk measure adopted is the risk of unit failure.*

Suggestion 2 *Consider including costs other than operating costs in the GMS model proposed in this dissertation.*

Production cost was taken to mean fuel cost in this dissertation (although in the DSS the user may increase this linear cost rate accordingly, so as to simulate the inclusion of other (linear) production costs), since it is the most significant cost associated with power generation.

Other common GMS operating costs, however, include generator start-up and shut-down costs (heavily affected by the UC problem), as well as fixed and variable maintenance costs, as discussed in §2.5.2. These costs may be incorporated in the GMS cost function in addition to the current production cost (focussing on fuel cost). Furthermore, the focus in this dissertation was on regulated power systems. As more countries around the world are, however, opening up their power systems to deregulation, the need increasingly arises to include the price of electricity so as to maximise the profit resulting from a GMS programme, as described in §2.5.2. Including this price aspect in the GMS planning process will also offer more trade-off possibilities, since cheap generating units (such as hydro or nuclear units) will, for example, be scheduled for maintenance during high price periods in order to maximise profit [88].

Suggestion 3 *Consider incorporating a nonlinear fuel cost formulation in the GMS model.*

The energy production cost in this dissertation was treated as a linear increase in fuel cost per generation output ($/MWh). As mentioned in §2.5.2, however, there are also many GMS model formulations that incorporate a nonlinear (usually quadratic) fuel cost. The benefits of this increase in realism in respect of estimating fuel cost should be weighed up against the considerable increase in computing time required to solve nonlinear optimisation problems as opposed to linear problems. The computing time required to solve the existing linear production cost model is currently already rather long.

Suggestion 4 *Speed up the computing time of the DMOSA algorithm.*

The main reason why the GMS model proposed in this dissertation takes relatively long to solve is the fact that the generating cost minimisation LP-module by Brits [25] (implemented in the programming language R) is solved during each iteration of the SA algorithm so as to estimate the production cost. Possible options to improve this computational burden include using a faster programming language. An appealing example is the *Rcpp* package [64] in R. It offers a seamless integration of R and C++, meaning that certain slow-running parts of the R script may be coded in C++ (if the user is proficient in the language), which holds the potential of drastically speeding up the solution process, perhaps even by orders of magnitude. This would mean that the entire coding of the algorithm and the DSS may still take place in R, but that the certain parts of the algorithm that take particularly long to execute, may be improved by coding the relevant section in C++ syntax. This suggestion is applicable to an EFS structure which is completely coded in the R programming language. As R is a high-level language, it reduces the time spent developing code, at the cost of increased computational time.

Suggestion 5 *Consider including resources other than available maintenance crew in the GMS model formulation.*

The only resource requirement considered in the GMS model proposed in this dissertation was to satisfy the maximum maintenance crew availability constraint. Other types of GMS resource constraints, involving available tools and parts may similarly be included in the model formulation, as described in §2.5.1. In addition, different crew types may be incorporated into the model formulation (*i.e.* engineers, technicians, welders, *etc.*). This sentiment was also shared by an Eskom employee [16].

In §7.3.2, an analysis was performed in respect of alternatively minimising the maximum crew availablity constraint's right-hand side, which showed that increasing/decreasing this value brings about slight trade-offs with the SSR objective, but more significant trade-offs with the production cost objective. In addition, the current constraint specifying a maximum amount of crew available does not strongly support levelised workload of the maintenance crew. Adopting this constraint may lead to instances (especially during periods of high demand) when no crew is dispatched for maintenance, and the crew might be idle during such periods. A GMS objective that levelises the crew required across the planning horizon may be of benefit and may be incorporated in a GMS MO paradigm.

Suggestion 6 *Consider employing other types of model solution techniques.*

Other solution techniques might perform better than the DMOSA algorithm implemented in this dissertation. Hence, further analysis is required in respect of the relative performances of other MOO algorithms in the context of the GMS problem. It may, for example, be of interest to tailor and compare the performance of a population-based MOO metaheuristic, such as the NSGA-II, for GMS.

Another promising technique that has not been applied frequently to GMS is CP. Froger *et al.* [90] attribute this scarcity to the fact that CP is less suitable to problems in which the main goal of the algorithm is to find near-optimal, as opposed to exactly optimal, solutions, which is typically the case in GMS problems. CP might, however, uncover adequate feasible (though not necessarily optimal) GMS solutions in the presence of multiple, tight constraints (objectives may also be included by setting minimum/maximum constraint satisfaction values) in a shorter solution time than achievable by conventional optimisation techniques employed in the GMS literature.

Suggestion 7 *Consider solving the transmission maintenance scheduling problem in conjunction with the GMS problem.*

As was mentioned in §2.3.3, the GMS and TMS problems are interdependent and should therefore ideally be solved jointly as has been done in some instances in the literature. These problems are nevertheless most often solved independently in the literature. The TMS problem may also hold considerable benefit in the context of an EFS.

Suggestion 8 *Consider solving the GMS problem in conjunction with a more sophisticated UC logic than that adopted in this dissertation.*

As mentioned in §2.3.1, the UC problem seeks to determine which available generating units (*i.e.* those not scheduled for maintenance) should be connected to the power generation system, so as to contribute actively to power generation. Usually the objectives of the UC problem include minimising operating costs, minimising emissions, or maximising the demand satisfaction capability [173]. The objective of minimising operating cost typically consists of minimising production cost, maintenance cost, start-up cost and shut-down cost. The production cost is usually determined by solving the ED problem, a typical subproblem of the UC problem [173, 212].

A simple UC logic was employed in the generation cost minimisation LP-module of Brits [25], as discussed in §3.3 and illustrated in Figure 3.3. The UC and GMS problems are very much interlinked and thus an improved algorithm may be used to either solve these two problem together or sequentially.

References

[1] Abirami M, Ganesan S, Subramanian S & Anandhakumar R, 2014, *Source and transmission line maintenance outage scheduling in a power system using teaching learning based optimization algorithm*, Applied Soft Computing, **21**, pp. 72–83.

[2] Abramson D, Amoorthy MK & Dang H, 1999, *Simulated annealing cooling schedules for the school timetabling problem*, Asia-Pacific Journal of Operational Research, **16(1)**, pp. 1–16.

[3] Ahmad A & Kothari DP, 1998, *A review of recent advances in generator maintenance scheduling*, Electric Machines & Power Systems, **26(4)**, pp. 373–387.

[4] Ahmad DP & Kothari A, 2000, *A practical model for generator maintenance scheduling with transmission constraints*, Electric Machines & Power Systems, **28(6)**, pp. 501–513.

[5] Ahmad R & Kamaruddin S, 2012, *An overview of time-based and condition-based maintenance in industrial application*, Computers and Industrial Engineering, **63(1)**, pp. 135–149.

[6] Alardhi M & Labib A, 2008, *Preventive maintenance scheduling of multi-cogeneration plants using integer programming* , Journal of the Operational Research Society, **60**, pp. 503–509.

[7] Alba E, Luque G & Nesmachnow S, 2013, *Parallel metaheuristics: Recent advances and new trends*, International Transactions in Operational Research, **20(1)**, pp. 1–48.

[8] Albrecht PF, 1972, *A four-state model for estimate of outage risk for units in peaking service*, IEEE Transactions on Power Apparatus and Systems, **PAS-91(2)**, pp. 618–627.

[9] Albrecht PF, Bhavaraju MP, Biggerstaff BE, Billington R, Jorgensen GE, Reppen ND & Shortley PB, 1979, *IEEE reliability test system*, IEEE Transactions on Power Apparatus and Systems, **PAS-98(6)**, pp. 2047–2054.

[10] Al-Khamis T, Vemuri S, Lemonidis L & Yellen J, 1991, *Unit maintenance scheduling with fuel constraints*, Proceedings of the Power Industry Computer Application Conference, Baltimore (MD), pp. 113–119.

[11] Allan RN, Billington R & Abdel-Gawad NMK, 1986, *The IEEE Reliability test system — Extensions to and evaluation of the generating system*, IEEE Transactions on Power Systems, **PWRS-1(4)**, pp. 1–7.

[12] Al-Turki U, 2011, *A framework for strategic planning in maintenance*, Journal of Quality in Maintenance Engineering, **17(2)**, pp. 150–162.

[13] Anagnostopoulos A, Michel L, Van Hentenryck P & Vergados Y, 2006, *A simulated annealing approach to the traveling tournament problem*, Journal of Scheduling, **9(2)**, pp. 177–193.

[14] Arueti S & Okrent D, 1990, *A knowledge-based prototype for optimization of preventive maintenance scheduling* , Reliability Engineering & System Safety, **30(1)**, pp. 93–114.

[15] Barot H & Bhattacharya K, 2008, *Security coordinated maintenance scheduling in deregulation based on genco contribution to unserved energy* , IEEE Transactions on Power Systems, **23(4)**, pp. 1871–1882.

[16] Bashe M, 2016, EFS activity leader at *Eskom*, [Personal Communication], Contactable at BasheM@eskom.co.za.

[17] Baskar S, Subbaraj P, Rao M & Tamilselvi S, 2003, *Genetic algorithms solution to generator maintenance scheduling with modified genetic operators*, **150(37)**, pp. 56–60.

[18] Beck WM, 2016, *R is my friend: A brief foray into parallel processing with R*, [Online], [Cited October 12th, 2016], Available from https://beckmw.wordpress.com/2014/01/21/a-brief-foray-into-parallel-processing-with-r/#comments.

[19] Benders JF, 1962, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, **4(1)**, pp. 238–252.

[20] Billinton R & El-Sheikhi FA, 1983, *Preventive maintenance scheduling of generating units in interconnected systems*, Proceedings of the International RAM Conference, pp. 364–370.

[21] Billinton R & Allan RN, 1996, *Reliability evaluation of power systems*, 2nd Edition, Springer, New York (NY).

[22] Billinton R & Harrington PG, 1978, *Reliability evaluation in energy limited generating capacity studies*, IEEE Transactions on Power Apparatus and Systems, **PAS-97(6)**, pp. 2076–2085.

[23] Boroujeni HF, Eghtedari M, Abdollahi M & Behzadipour E, 2012, *Calculation of generation system reliability index: Loss of load probability* , Life Science Journal, **9(4)**, pp. 4903–4908.

[24] Brandt F, Bauer R, Völker M & Cardeneo A, 2013, *A constraint programming-based approach to a large-scale energy management problem with varied constraints*, Journal of Scheduling, **16(6)**, pp. 629–648.

[25] Brits RO, 2016, *Adding multi-objective optimisation capability to an electricity utility energy flow simulator* , MEng Thesis, Stellenbosch University, Stellenbosch.

[26] Buljubasi M & Gavranović H, 2012, *Orchestrating constrained programming and local search to solve a large scale energy management problem*, Proceedings of the 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, pp. 371–378.

[27] Burke EK, Clarke JA & Smith AJ, 1998, *Four methods for maintenance scheduling* , Proceedings of the Artificial neural nets and genetic algorithms conference, Norwich, pp. 264–269.

[28] Burke EK, Kendall G & Whitwell G, 2009, *A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem*, INFORMS Journal on Computing, **21(3)**, pp. 505–516.

[29] Burke E & Smith A, 2000, *Hybrid evolutionary techniques for the maintenance scheduling problem*, IEEE Transactions on Power Systems, **15(1)**, pp. 122–128.

[30] Busetti F, 2003, *Simulated annealing overview* , [Online], [Cited July 11th, 2016], Available from http://www.aiinfinance.com/saweb.pdf.

[31] Calabrese G, 1947, *Generating reserve capacity determined by the probability method*, American Institute of Electrical Engineers, **66(1)**, pp. 1439–1450.

[32] Canto SP, 2008, *Application of Benders' decomposition to power plant preventive maintenance scheduling*, European Journal of Operational Research, **184(2)**, pp. 759–777.

[33] Carrión M & Arroyo JM, 2006, *A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem*, IEEE Transactions on Power Systems, **21(3)**, pp. 1371–1378.

[34] Casazza J, Casazza J & Delea F, 2003, *Understanding electric power systems: An overview of the technology and the marketplace*, John Wiley & Sons, Piscataway (NJ).

[35] Černý V, 1985, *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*, Journal of Optimization Theory and Applications, **45(1)**, pp. 41–51.

[36] Chang W, Cheng J, Allaire JJ, Xie Y & McPherson J, 2016, *Shiny: Web application framework for R*, R package version 0.13.2, [Online], [Cited October 23rd, 2016], Available from `http://CRAN.R-project.org/package=shiny`.

[37] Chattopadhyay D, 2004, *A game theoretic model for strategic maintenance and dispatch decisions*, IEEE Transactions on Power Systems, **19(4)**, pp. 2014–2021.

[38] Chen XD, Zhan J, Wu QH & Guo C, 2014, *Multi-objective optimization of generation maintenance scheduling*, Proceedings of the 2014 IEEE PES General Meeting — Conference & Exposition, National Harbour (MD), pp. 1–5.

[39] Coello CC, Lamont GB & Van Veldhuizen DA, 2007, *Evolutionary algorithms for solving multi-objective problems*, 2nd Edition, Springer, Secaucus (NJ).

[40] Coinnews Media Group, 2016, *US inflation calculator*, [Online], [Cited June 16th, 2016], Available from `http://www.usinflationcalculator.com`.

[41] Colorni A, Dorigo M & Maniezzo V, 1991, *Distributed optimization by ant colonies*, Proceedings of the first European conference on artificial life, Paris, pp. 134–142.

[42] Conejo AJ, García-Bertrand R & Díaz-Salazar M, 2005, *Generation maintenance scheduling in restructured power systems*, IEEE Transactions on Power Systems, **20(2)**, pp. 984–992.

[43] Crawford B, Soto R, Castro C & Monfroy E, 2011, *An extensible autonomous search framework for constraint programming*, International Journal of Physical Sciences, **6(14)**, pp. 3369–3376.

[44] Czyżak P & Jaszkiewicz A, 1998, *Pareto simulated annealing — a metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis, **7(1)**, pp. 34–47.

[45] Dahal KP, 2004, *A review of maintenance scheduling approaches in deregulated power systems*, Proceedings of the International Conference in Power Systems (ICPS 2004), Kathmandu, pp. 565–570.

[46] Dahal KP & Chakpitak N, 2007, *Generator maintenance scheduling in power systems using metaheuristic-based hybrid approaches*, Electric Power Systems Research, **77(7)**, pp. 771–779.

[47] Dahal KP & McDonald J, 1997, *A review of generator maintenance scheduling using artificial intelligence techniques*, Proceedings of the 32nd Universities Power Engineering Conference (UPEC '97), Manchester, pp. 10–12.

[48] Dahal K, Aldridge C & McDonald J, 1999, *Generator maintenance scheduling using a genetic algorithm with a fuzzy evaluation function*, Fuzzy Sets and Systems, **102**, pp. 21–29.

[49] Dahal K, McDonald J & Burt G, 2000, *Modern heuristic techniques for scheduling generator maintenance in power systems*, Transactions of the Institute of Measurement and Control, **22(2)**, pp. 179–194.

[50] Das I & Dennis JE, 1997, *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*, Structural Optimization, **14(1)**, pp. 63–69.

[51] Deb K & Jain H, 2014, *An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints*, IEEE Transactions on Evolutionary Computation, **18(4)**, pp. 577–601.

[52] Deb K, Pratap A, Agarwal S & Meyarivan T, 2002, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, **6(2)**, pp. 182–197.

[53] Department of Energy, 2011, *Integrated resource plan for electricity (IRP) 2010–2030*, [Government Report], Pretoria.

[54] Department of Energy, 2013, *Integrated resource plan for electricity (IRP) 2010–2030 (Update)*, [Government Report], Pretoria.

[55] Dicodess, 2009, *Dicodess: A software framework for developing distributed cooperative decision support systems*, [Online], [Cited July 16th, 2016], Available from `http://dicodess.sourceforge.net/`.

[56] Digalakis JG & Margaritis KG, 2002, *A multipopulation cultural algorithm for the electrical generator scheduling problem*, Mathematics and Computers in Simulation, **60(3)**, pp. 293–301.

[57] Dillon TS, Edwin KW, Kochs HD & Taud RJ, 1978, *Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination*, IEEE Transactions on Power Apparatus and Systems, **PAS-97(6)**, pp. 2154–2166.

[58] Dongarra J & Sullivan F, 2000, *Guest editors' introduction: The top 10 algorithms*, Computing in Science & Engineering, **2(1)**, pp. 22–23.

[59] Dopazo JF & Merrill HM, 1975, *Optimal generator maintenance scheduling using integer programming*, IEEE Transactions on Power Apparatus and Systems, **94(5)**, pp. 1537–1545.

[60] Dowsland KA & Thompson JM, 2012, *Simulated annealing*, pp. 1623–1655 in Rozenberg G, Bäck T & Kok JN (Eds), *Handbook of natural computing*, Springer, Berlin.

[61] Dréo J, Petrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimization: Methods and case studies*, Springer, Berlin.

[62] Droste S, Jansen T & Wegener I, 1999, *Perhaps not a free lunch but at least a free appetizer*, Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, Orlando (FL), pp. 833–839.

[63] Duval P & Poilpot R, 1983, *Determining maintenance schedules for thermal production units: The Kapila model*, IEEE Transactions on Power Apparatus and Systems, **(8)**, pp. 2509–2520.

[64] Eddelbuettel D & François R, 2011, *Rcpp: Seamless R and C++ Integration*, Journal of Statistical Software, **40(8)**, pp. 1–18.

[65] Edwin K & Curtius F, 1990, *New maintenance-scheduling method with production cost minimization via integer linear programming*, International Journal of Electrical Power & Energy Systems, **1(3)**, pp. 2–7.

[66] Ekpenyong UE, Zhang J & Xia X, 2012, *An improved robust model for generator maintenance scheduling*, Electric Power Systems Research, **92**, pp. 29–36.

[67] El-Amin I, Duffuaa S & Abbas M, 2000, *A tabu search algorithm for maintenance scheduling of generating units*, Electric Power Systems Research, **54(2)**, pp. 91–99.

[68] El-Sharkh MY & El-Keib AA, 2003, *An evolutionary programming-based solution methodology for power generation and transmission maintenance scheduling*, Electric Power Systems Research, **65(1)**, pp. 35–40.

[69] El-Sharkh M, El-Keib A & Chen H, 2003, *A fuzzy evolutionary programming-based solution methodology for security-constrained generation maintenance scheduling*, Electric Power Systems Research, **67(1)**, pp. 67–72.

[70] Enerdata, 2016, *Electricity consumption — World Power consumption*, [Online], [Cited August 14th, 2016], Available from https://yearbook.enerdata.net/electricity-domestic-consumption-data-by-region.html.

[71] Energy Research Centre, 2014, *Assumptions and methodologies in the South African TIMES (SATIM) energy model*, url: http://www.erc.uct.ac.za/Research/Otherdocs/Satim/SATIM%20Methodology-v2.1.pdf.

[72] Engrand P, 1997, *A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management*, Proceedings of the 5th Conference on Nuclear Engineering, Nice, pp. 1–8.

[73] Eskom Holdings Limited, 2016, *Company information*, [Online], [Cited June 11th, 2016], Available from http://www.eskom.co.za/OurCompany/CompanyInformation/Pages/Company_Information.aspx.

[74] Eskom Holdings Limited, 2016, *Eskom Holdings Limited: Integrated Report 2016*, (Unpublished) Technical Report, Eskom Holdings Limited.

[75] Eskom Holdings Limited, 2016, *Eskom Home*, [Online], [Cited October 17th, 2016], Available from http://www.eskom.co.za/Pages/Landing.aspx.

[76] Eskom Holdings Limited, 2016, *Eskom — Kusile Power Station Project*, [Online], [Cited August 15th, 2016], Available from http://www.eskom.co.za/Whatweredoing/NewBuild/Pages/Kusile_Power_Station.aspx.

[77] Eskom Holdings Limited, 2016, *Eskom to step up maintenance in preparation for the winter demand*, [Online], [Cited August 15th, 2016], Available from http://www.eskom.co.za/news/Pages/Mar24.aspx.

[78] Eskom Holdings Limited, 2016, *Eskom's maintenance tool*, [Online], [Cited February 06th, 2016], Available from http://www.eskom.co.za/news/Pages/Feb6.aspx.

[79] Eskom Holdings Limited, 2015, *Fact sheets with additional information*, (Unpublished) Technical Report, Eskom Holdings Limited.

[80] Exemplar E, 2016, *Energy Exemplar — Energy Market Modelling — Our Company*, [Online], [Cited August 4th, 2016], Available from http://energyexemplar.com/our-company/.

[81] Exemplar E, 2016, *Energy Exemplar — Energy Market Modelling — PLEXOS Integrated Energy Model*, [Online], [Cited August 04th, 2016], Available from http://energyexemplar.com/software/plexos-desktop-edition/.

[82] Fattahi M, Mahootchi M, Mosadegh H & Fallahi F, 2014, *A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours*, Computers & Operations Research, **50**, pp. 61–79.

[83] Fetanat A & Shafipour G, 2011, *Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0–1 integer programming*, Expert Systems with Applications, **38(8)**, pp. 9729–9735.

[84] Fin24, 2015, *Fin24 — WATCH — How classic game Tetris saved Eskom*, [Online], [Cited October 24th, 2016], Available from http://www.fin24.com/Economy/Eskom/watch-how-classic-game-tetris-saved-eskom-20151124.

[85] Foley A, Gallachóir BÓ, Hur J, Baldick R & McKeogh E, 2010, *A strategic review of electricity systems models*, Energy, **35(12)**, pp. 4522–4530.

[86] Foley A & Lobera ID, 2013, *Impacts of compressed air energy storage plant on an electricity market with a large renewable energy portfolio*, Energy, **57**, pp. 85–94.

[87] Foong WK, 2007, *Ant colony optimisation for power plant maintenance scheduling*, PhD Thesis, The Univeristy of Adelaide, Adelaide.

[88] Foong WK, Maier HR & Simpson AR, 2008, *Power plant maintenance scheduling using ant colony optimization: An improved formulation*, Engineering Optimization, **40(4)**, pp. 309–329.

[89] Foong WK, Simpson AR, Maier HR & Stolp S, 2008, *Ant colony optimization for power plant maintenance scheduling optimization — A five-station hydropower system*, Annals of Operations Research, **159(1)**, pp. 433–450.

[90] Froger A, Gendreau M, Mendoza JE, Pinson É & Rousseau L, 2016, *Maintenance scheduling in the electricity industry: A literature review*, European Journal of Operational Research, **251(3)**, pp. 695–706.

[91] Frost D & Dechter R, 1998, *Optimizing with constraints: A case study in scheduling maintenance of electric power units*, Lecture Notes in Computer Science, **1520**, pp. 469–488.

[92] Fuštar S & Jelavić B, 1992, *A knowledge-based system for power system weekly scheduling*, International Journal of Electrical Power & Energy Systems, **14(2)**, pp. 206–211.

[93] Gachet A, 2004, *Building model driven decision support systems with Dicodess*, VDF, Zurich.

[94] Gallimore KF & Penlesky RJ, 1988, *A framework for developing maintenance strategies*, Production & Inventory Management Journal, **29(1)**, pp. 16–22.

[95] Geetha T & Swarup KS, 2009, *Coordinated preventive maintenance scheduling of GENCO and TRANSCO in restructured power systems*, International Journal of Electrical Power & Energy Systems, **31(10)**, pp. 626–638.

[96] Geman S & Geman D, 1984, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **6(6)**, pp. 721–741.

[97] Gesmann M & De Castillo D, 2011, *googleVis: Interface between R and the Google Visualisation API*, The R Journal, **3(2)**, pp. 40–44.

[98] Glover F, 1986, *Future paths for integer programming and links to artificial intelligence*, Computers & Operations Research, **13(5)**, pp. 533–549.

[99] Godskesen S, Jensen TS, Kjeldsen N & Larsen R, 2013, *Solving a real-life, large-scale energy management problem*, Journal of Scheduling, **16(6)**, pp. 567–583.

[100] Goldberg DE, 1989, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Longman Publishing, Boston (MA).

[101] Granville V, Krivánek M & Rasson J, 1994, *Simulated annealing: A proof of convergence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **16(6)**, pp. 652–656.

[102] Guo CX, Zhan JP & Wu QH, 2012, *Dynamic economic emission dispatch based on group search optimizer with multiple producers*, Electric Power Systems Research, **86**, pp. 8–16.

[103] Han SM, Chung KH & Kim BH, 2011, *ISO coordination of generator maintenance scheduling in competitive electricity markets using simulated annealing*, Journal of Electrical Engineering and Technology, **6(4)**, pp. 431–438.

[104] Hatton M, 2015, *Requirements specification for the optimisation function of an electric utility's energy flow simulator*, MEng Thesis, Stellenbosch University, Stellenbosch.

[105] Hatton M & Bekker J, 2014, *Development of an optimiser for a simulator of an electric utility: Challenges and approach*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, Parys, pp. 18–26.

[106] He S, Wu QH & Saunders J, 2009, *Group search optimizer: An optimization algorithm inspired by animal searching behavior*, IEEE Transactions on Evolutionary Computation, **13(5)**, pp. 973–990.

[107] Huang CJ, Lin CE & Huang CL, 1992, *Fuzzy approach for generator maintenance scheduling*, Electric Power Systems Research, **24(1)**, pp. 31–38.

[108] Huang SJ, 1997, *Generator maintenance scheduling: A fuzzy system approach with genetic enhancement*, Electric Power Systems Research, **41(3)**, pp. 233–239.

[109] Hughes EJ, 2005, *Evolutionary many-objective optimisation: many once or one many?*, Proceedings of the 2005 IEEE congress on evolutionary computation, Edinburgh, pp. 222–227.

[110] Infrastructure news, 2016, *Infrastructure news — Eskom steps up maintenance plans*, [Online], [Cited August 15th, 2016], Available from `http://www.infrastructurene.ws/2016/03/24/eskom-steps-up-maintenance-plans/`.

[111] Ingber L, 1993, *Simulated annealing: Practice versus theory*, Mathematical and Computer Modelling, **18(11)**, pp. 29–57.

[112] International Rivers, 2015, *Grand Inga Dam, DR Congo*, [Online], [Cited August 14th, 2016], Available from `https://www.internationalrivers.org/campaigns/grand-inga-dam-dr-congo`.

[113] International Rivers, 2005, *Grand Inga, Grand Illusions?*, [Online], [Cited August 14th, 2016], Available from `https://www.internationalrivers.org/resources/grand-inga-grand-illusions-1949`.

[114] Jardine AKS, Lin D & Banjevic D, 2006, *A review on machinery diagnostics and prognostics implementing condition-based maintenance*, Mechanical Systems and Signal Processing, **20(7)**, pp. 1483–1510.

[115] Johnson R, Happ H & Wright W, 1971, *Large scale hydro-thermal unit commitment-method and results*, IEEE Transactions on Power Apparatus and Systems, **(3)**, pp. 1373–1384.

[116] Kendall KE & E KJ, 2011, *Systems analysis and design*, 8th Edition, Pearson, Upper Saddle River (NJ).

[117] Kennedy J & Eberhart R, 1995, *Particle swarm optimization*, Proceedings of the IEEE International Conference on Neural Network, Perth, pp. 1942–1948.

[118] Khalid A & Ioannis K, 2012, *A survey of generator maintenance scheduling techniques*, Global Journal of Researches in Engineering, **12(1)**, pp. 10–17.

[119] Kim JH, Park JB, Park JK & Chun YH, 2005, *Generating unit maintenance scheduling under competitive market environments*, International Journal of Electrical Power & Energy Systems, **27(3)**, pp. 189–194.

[120] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimization by simulated annealing*, Science, **220(4598)**, pp. 671–680.

[121] Klein JB, 1998, *The use of heat rates in production cost modeling and market modeling*, Report, Electricity Analysis Office, California Energy Commission.

[122] Koay CA & Srinivasan D, 2003, *Particle swarm optimization-based approach for generator maintenance scheduling*, Proceedings of the 2003 IEEE Symposium on Swarm Intelligence (SIS '03), Indianapolis (IN), pp. 167–173.

[123] Kolb J, 2009, *Eskom 2000–2008: Our recent past*, [Online], [Cited May 15th, 2015], Available from http://heritage.eskom.co.za/eskom2000.htm.

[124] Kralj BL & Petrović R, 1988, *Optimal preventive maintenance scheduling of thermal generating units in power systems — A survey of problem formulations and solution methods*, European Journal of Operational Research, **35(1)**, pp. 1–15.

[125] Kralj BL & Rajaković N, 1994, *Multiobjective programming in power system optimization: New approach to generator maintenance scheduling*, International Journal of Electrical Power & Energy Systems, **16(4)**, pp. 211–220.

[126] Kralj BL & Petrović R, 1995, *A multiobjective optimization approach to thermal generating units maintenance scheduling*, European Journal of Operational Research, **84(2)**, pp. 481–493.

[127] Kuzle I, Pandžić H & Brezovec M, 2007, *Implementation of the Benders decomposition in hydro generating units maintenance scheduling*, Proceedings of the Hydro Conference 2007: New Approaches for a New Era, Granada, pp. 1–8.

[128] Leou RC, 2006, *A new method for unit maintenance scheduling considering reliability and operation expense*, International Journal of Electrical Power & Energy Systems, **28(7)**, pp. 471–481.

[129] Lin CE, Huang CJ, Huang CL, Liang CC & Lee SY, 1992, *An expert system for generator maintenance scheduling using operation index*, IEEE Transactions on Power Systems, **7(3)**, pp. 1141–1148.

[130] Lindner, B G , 2016, *GMS Shiny*, [Online], [Cited October 19th, 2016], Available from https://berndtlindner.shinyapps.io/GMS_Shiny/.

[131] Lindner BG & Van Vuuren JH, 2014, *Maintenance scheduling for the generating units of a national power utility*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, Parys, pp. 36–44.

[132] Lindner BG & Van Vuuren JH, *Tri-objective generator maintenance scheduling for a national power utility*, Proceedings of the 44th Annual Conference of the Operations Research Society of South Africa, Hartbeespoort, pp. 112–122.

[133] Lootsma FA, 2013, *Fuzzy logic for planning and decision making*, 1st Edition, Springer, Boston (MA).

[134] Martin KF, 1994, *A review by discussion of condition monitoring and fault diagnosis in machine tools*, International Journal of Machine Tools & Manufacture, **34(4)**, pp. 527–551.

[135] Marwali MKC & Shahidehpour SM, 1998, *A deterministic approach to generation and transmission maintenance scheduling with network constraints*, Electric Power Systems Research, **47(2)**, pp. 101–113.

[136] Marwali MKC & Shahidehpour SM, 1999, *A probabilistic approach to generation maintenance scheduler with network constraints*, International Journal of Electrical Power & Energy Systems, **21(8)**, pp. 533–545.

[137] Marwali MKC & Shahidehpour SM, 1998, *Integrated generation and transmission maintenance scheduling with network constraints*, IEEE Transactions on Power Systems, **13(3)**, pp. 1063–1068.

[138] Matthew le Cordeur, 2015, *Eskom does not have demand crisis — as it happened* , [Online], [Cited October 16th, 2016], Available from `http://www.fin24.com/Economy/Eskom/live-eskom-state-of-the-system-update-20151116`.

[139] Mdunge B, 2016, Production Assurance Manager at *Eskom*, [Personal Communication], Contactable at `mdungeba@eskom.co.za`.

[140] Mersmann O, 2014, *mco: Multiple criteria optimization algorithms and related functions*, R package version 1.0-15.1, [Online], [Cited October 23rd, 2016], Available from `http://CRAN.R-project.org/package=mco`.

[141] Micali V & Heunis S, 2011, *Coal stockpile simulator* , Proceedings of the 8th IEEE Industrial and Commercial Use of Energy Conference (ICUE), Cape Town, pp. 198–203.

[142] Micali V, 2012, *Prediction of availability for new power plant in the absence of data*, Proceedings of the 9th IEEE Industrial and Commercial Use of Energy Conference (ICUE), Stellenbosch, pp. 1–8.

[143] Miettinen K, 2012, *Nonlinear Multiobjective Optimization*, Springer, New York (NY).

[144] Min CG, Kim MK, Park JK & Yoon YT, 2013, *Game-theory-based generation maintenance scheduling in electricity markets*, Energy, **55**, pp. 310–318.

[145] Mohanta DK, Sadhu PK & Chakrabarti R, 2007, *Deterministic and stochastic approach for safety and reliability optimization of captive power plant maintenance scheduling using GA/SA-based hybrid techniques: A comparison of results*, Reliability Engineering and System Safety, **92(2)**, pp. 187–199.

[146] Mohanta DK, Sadhu PK & Chakrabarti R, 2004, *Fuzzy reliability evaluation of captive power plant maintenance scheduling incorporating uncertain forced outage rate and load representation*, Electric Power Systems Research, **72(1)**, pp. 73–84.

[147] Moro LM & Ramos A, 1999, *Goal programming approach to maintenance scheduling of generating units in large scale power systems*, IEEE Transactions on Power Systems, **14(3)**, pp. 1021–1028.

[148] Mromlinski LR, 1985, *Transportation problem as a model for optimal schedule of maintenance outages in power systems*, International Journal of Electrical Power & Energy Systems, **7(3)**, pp. 161–164.

[149] Mukerji R, Merrill HM, Erickson BW, Parker JH & Friedman RE, 1991, *Power plant maintenance scheduling: Optimizing economics and reliability*, IEEE Transactions on Power Systems, **6(2)**, pp. 476–483.

[150] Mytakidis T & Vlachos A, 2008, *Maintenance scheduling by using the bi-criterion algorithm of preferential anti-pheromone*, Leonardo Journal of Sciences, **12(16)**, pp. 143–164.

[151] Nam D & Park CH, 2000, *Multiobjective simulated annealing: A comparative study to evolutionary algorithms*, International Journal of Fuzzy Systems, **2(2)**, pp. 87–97.

[152] Negnevitsky M & Kelareva G, 1999, *Genetic algorithms for maintenance scheduling in power systems*, Proceedings of the the Australasian Universities Power Engineering Conference and IEAust Electric Energy Conference, Darwin, pp. 184–189.

[153] Or İ, 1993, *Development of a Decision Support System for maintenance planning* , pp. 505–520 in Holsapple CW & Whinston AB (Eds), *Recent Developments in Decision Support Systems*, Springer, Istanbul.

[154] Padhy NP, 2004, *Unit commitment — a bibliographical survey*, IEEE Transactions on Power Systems, **19(2)**, pp. 1196–1205.

[155] Patel SA & Kamrani AK, 1996, *Intelligent decision support system for diagnosis and maintenance of automated systems*, Computers & Industrial Engineering, **30(2)**, pp. 297–319.

[156] Power DJ, 2002, *Decision support systems: Concepts and resources for managers*, Quorum Books, Westport (CT).

[157] Power DJ, 1997, *What is a DSS* , The On-Line Executive Journal for Data-Intensive Decision Support, **1(3)**, pp. 223–232.

[158] Prada JF, 1999, *The value of reliability in power systems-pricing operating reserves*, (Unpublished) Technical Report, MIT Laboratory for the Energy and the Environment, Working Paper EL 99-005 WP.

[159] Purshouse RC & Fleming PJ, 2003, *Evolutionary many-objective optimisation: An exploratory analysis*, Proceedings of the IEEE Congress on Evolutionary Computation, Canberra, pp. 2066–2073.

[160] Qian X & Wu Y, 2016, *An electricity price-dependent control-limit policy for condition-based maintenance optimization for power generating unit*, Eksploatacja i Niezawodnosc - Maintenance and Reliability, **18(2)**, pp. 245–253.

[161] Reihani E, Sarikhani A, Davodi M & Davodi M, 2012, *Reliability based generator maintenance scheduling using hybrid evolutionary approach*, International Journal of Electrical Power & Energy Systems, **42(1)**, pp. 434–439.

[162] Revolution Analytics & Weston S, 2015, *doParallel: Foreach parallel adaptor for the 'parallel' package*, R package version 1.0.10, [Online], [Cited October 23rd, 2016], Available from `http://CRAN.R-project.org/package=doParallel`.

[163] Revolution Analytics & Weston S, 2015, *foreach: Provides foreach looping construct for R*, R package version 1.4.3, [Online], [Cited October 23rd, 2016], Available from `http://CRAN.R-project.org/package=foreach`.

[164] Said GAE.-NA, Mahmoud AM & El-Horbaty E.-SM, 2014, *A comparative study of meta-heuristic algorithms for solving quadratic assignment problem*, International Journal of Advanced Computer Science and Applications, **5(1)**, pp. 1–6.

[165] Saraiva JT, Pereira ML, Mendes VT & Sousa JC, 2011, *A simulated annealing based approach to solve the generator maintenance scheduling problem*, Electric Power Systems Research, **81(7)**, pp. 1283–1291.

[166] Sarker M, Chowdhury D & Hossain I, 2014, *Power generation from coal: A review*, Journal of Chemical Engineering, **27(2)**, pp. 50–54.

[167] Satoh T & Nara K, 1991, *Maintenance scheduling by using simulated annealing method [for power plants]*, IEEE Transactions on Power Systems, **6(2)**, pp. 850–857.

[168] Schlünz EB, 2011, *Decision support for generator maintenance scheduling in the energy sector*, MSc Thesis, Stellenbosch University, Stellenbosch.

[169] Schlünz EB & Van Vuuren JH, 2013, *An investigation into the effectiveness of simulated annealing as a solution approach for the generator maintenance scheduling problem*, International Journal of Electrical Power & Energy Systems, **53**, pp. 166–174.

[170] Schlünz EB & Van Vuuren JH, 2012, *The application of a computerised decision support system for generator maintenance scheduling: A South African case study*, South African Journal of Industrial Engineering, **23(3)**, pp. 169–179.

[171] Schlünz EB, Bokov PM & Van Vuuren JH, 2014, *Research reactor in-core fuel management optimisation using the multiobjective cross-entropy method*, Proceedings of the 2014 International Conference on Reactor Physics (PHYSOR 2014), Kyoto.

[172] Seebregts AJ, Goldstein GA & Smekens K, 2002, *Energy/environmental modeling with the MARKAL family of models*, pp. 75–82 in Chamoni P, Leisten R, Martin A, Minnemann J & Stadtler H (Eds), *Operations Research Proceedings 2001: Selected Papers of the International Conference on Operations Research (OR 2001) Duisburg, September 3–5, 2001*, Springer, Berlin.

[173] Sen S & Kothari D, 1998, *Optimal thermal generating unit commitment: A review*, International Journal of Electrical Power & Energy Systems, **20(7)**, pp. 443–451.

[174] Serafini P, 1994, *Simulated annealing for multi objective optimization problems*, pp. 283–292 in Tzeng GH, Wang HF, Wen UP & Yu PL (Eds), *Multiple Criteria Decision Making: Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, Springer, New York (NY).

[175] Sergaki A & Kalaitzakis K, 2002, *A fuzzy knowledge based method for maintenance planning in a power system*, Reliability Engineering & System Safety, **77(1)**, pp. 19–30.

[176] Sherif YS & Smith ML, 1981, *Optimal maintenance models for systems subject to failure—a review*, Naval Research Logistics Quarterly, **28(1)**, pp. 47–74.

[177] Shi Y & Eberhart R, 1998, *A modified particle swarm optimizer*, IEEE International Conference on Evolutionary Computation, pp. 69–73.

[178] Smith KI, Everson RM, Fieldsend JE, Murphy C & Misra R, 2008, *Dominance-based multi-objective simulated annealing*, IEEE Transactions on Evolutionary Computation, **12(3)**, pp. 323–342.

[179] Smith KI, Everson R & Fieldsend J, 2004, *Dominance measures for multi-objective simulated annealing*, Proceedings of the 3rd Congress on Evolutionary Computation, Piscataway (NJ), pp. 23–30.

[180] South African Government, 2016, *Energy challenge*, [Online], [Cited October 28th, 2015], Available from `http://www.gov.za/issues/energy-challenge`.

[181] Sprague Jr. RH, 1980, *A framework for the development of decision support systems*, Management Information System Quarterly, pp. 1–26.

[182] Stack Overflow, 2016, *Stack Overflow: — 8 logical threads at 4 cores will at a maximum run 4 times faster in parallel?— multithreading*, [Online], [Cited October 12th, 2016], Available from http://stackoverflow.com/questions/10403201/8-logical-threads-at-4-cores-will-at-a-maximum-run-4-times-faster-in-parallel.

[183] Suman B & Kumar P, 2006, *A survey of simulated annealing as a tool for single and multiobjective optimization*, Journal of the Operational Research Society, **57(10)**, pp. 1143–1160.

[184] Suppapitnarm A, Seffen K, Parks G & Clarkson P, 2000, *A simulated annealing algorithm for multiobjective optimization*, Engineering Optimization, **33(1)**, pp. 59–85.

[185] Suresh K & Kumarappan N, 2013, *Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem*, Swarm and Evolutionary Computation, **9**, pp. 69–89.

[186] Tabari NM, Pirmoradian M & Hassanpour SB, 2008, *Implicit enumeration based 0–1 integer programming for generation maintenance scheduling*, Proceedings of the IEEE Region 8 International Conference on Computational Technologies in Electrical & Electronics Engineering, Novosibirsk, pp. 151–154.

[187] Talbi E, 2009, *Metaheuristics: From design to implementation*, 1st Edition, John Wiley & Sons, Hoboken (NJ).

[188] The MathWorks Inc, 2009, *MATLAB R2009a — The language of technical computing*, [Online], [Cited October 21st, 2016], Available from http://www.mathworks.com..

[189] The RStudio project, 2016, *RStudio — Open source and enterpise-ready professional software for R*, [Online], [Cited August 1st, 2016], Available from https://www.rstudio.com/.

[190] The RStudio project, 2016, *Shiny*, [Online], [Cited July 14th, 2016], Available from http://shiny.rstudio.com/.

[191] The RStudio project, 2014, *Shiny — Sharing apps to run locally*, [Online], [Cited January 6th, 2014], Available from http://shiny.rstudio.com/articles/deployment-local.html.

[192] Triki E, Collette Y & Siarry P, 2005, *A theoretical study on the behavior of simulated annealing leading to a new cooling schedule*, European Journal of Operational Research, **166(1)**, pp. 77–92.

[193] Tsou CS, 2013, *nsga2R: Elitist Non-dominated Sorting Genetic Algorithm based on R, R package version 1.0*, [Online], [Cited October 16th, 2016], Available from http://CRAN.R-project.org/package=nsga2R.

[194] Ulungu E, Teghem J, Fortemps P & Tuyttens D, 1999, *MOSA method: A tool for solving multiobjective combinatorial optimization problems*, Journal of Multi-criteria Decision Analysis, **8(4)**, pp. 221–236.

[195] Van Harmelen G, 2014, Utility Analytics Business Area Manager at *Enerweb*, [Personal Communication], Contactable at gerard.van.harmelen@enerweb.co.za.

[196] Van Laarhoven PJ & Aarts EH, 1987, *Simulated annealing: Theory and applications*, Springer, Dordrecht.

[197] Vazquez MAO, 2006, *Optimizing the spinning reserve requirements*, PhD thesis, University of Manchester, Manchester.

[198] Vijayamohanan Pillai N, 2014, *Loss of Load Probability of a Power System*, Journal of Fundamentals of Renewable Energy and Applications, **5(149)**, pp. 1–9.

[199] Vlachodimitropoulos M, 2008, *White paper on PLEXOS and the Sub-Saharan Africa power market*, (Unpublished) Technical Report, Energy Exemplar.

[200] Volkanovski A, Mavko B, Boševski T, Čauševski A & Čepin M, 2008, *Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system*, Reliability Engineering and System Safety, **93(6)**, pp. 779–789.

[201] Wang X & McDonald JR, 1994, *Modern power system planning* , McGraw-Hill, London.

[202] Wang Y & Handschin E, 2000, *A new genetic algorithm for preventive unit maintenance scheduling of power systems*, International Journal of Electrical Power & Energy Systems, **22(5)**, pp. 343–348.

[203] While L, Hingston P, Barone L & Huband S, 2006, *A faster algorithm for calculating hypervolume*, IEEE Transactions on Evolutionary Computation, **10(1)**, pp. 29–38.

[204] Wikipedia The Free Encyclopedia, 2016, *Human-computer interaction*, [Online], [Cited July 25th, 2016], Available from https://en.wikipedia.org/wiki/Human-computer_interaction.

[205] Wikipedia The Free Encyclopedia, 2016, *List of largest power stations in the world* , [Online], [Cited August 12th, 2016], Available from https://en.wikipedia.org/wiki/List_of_largest_power_stations_in_the_world.

[206] Wikipedia The Free Encyclopedia, 2016, *Planned maintenance*, [Online], [Cited July 20th, 2016], Available from https://en.wikipedia.org/wiki/Planned_maintenance.

[207] Wikipedia The Free Encyclopedia, 2016, *Three Gorges Dam*, [Online], [Cited August 14th, 2016], Available from https://en.wikipedia.org/wiki/Three_Gorges_Dam.

[208] Wikipedia The Free Encyclopedia, 2016, *User interface*, [Online], [Cited July 14th, 2016], Available from https://en.wikipedia.org/wiki/User_interface.

[209] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th Edition, Brooks/Coles, Belmont (CA).

[210] Wolpert DH & Macready WG, 2005, *Coevolutionary free lunches*, IEEE Transactions on Evolutionary Computation, **9(6)**, pp. 721–735.

[211] Wolpert DH & Macready WG, 1997, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation, **1(1)**, pp. 67–82.

[212] Wood AJ & Wollenberg BF, 2006, *Power generation, operation, and control* , 2nd Edition, John Wiley & Sons, New York (NY).

[213] Wood AJ, Wollenberg BF & Sheble G, 2013, *Power generation, operation, and control*, 3rd Edition, John Wiley & Sons, New York (NY).

[214] Wu Q, Lu Z, Li M & Ji T, 2008, *Optimal placement of FACTS devices by a group search optimizer with multiple producer* , Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1033–1039.

[215] Yamayee ZA, 1982, *Maintenance scheduling: description, literature survey, and interface with overall operations scheduling* , IEEE Transactions on Power Apparatus and Systems, **101(8)**, pp. 2770–2779.

[216] Yamayee ZA, Sidenblad K & Yoshimura M, 1983, *A computationally efficient optimal maintenance scheduling method* , IEEE Transactions on Power Apparatus and Systems, **102(2)**, pp. 330–338.

[217] Yang F & Chang CS, 2009, *Multiobjective evolutionary optimization of maintenance schedules and extents for composite power systems*, IEEE Transactions on Power Systems, **24(4)**, pp. 1694–1702.

[218] Zhan JP, Guo CX, Wu QH, Zhang LL & Fu HJ, 2014, *Generation maintenance scheduling based on multiple objectives and their relationship analysis*, Journal of Zhejiang University SCIENCE C, **15(11)**, pp. 1035–1047.

[219] Zhou Y, Mohamed S, Lim K & Vinther R, 2013, *A decision support system (dss) for maintenance of a plurality of renewable energy generators in a renewable power plant*, [Patent], WO Patent Application number PCT/DK2012/050,453.

[220] Zitzler E, Brockhoff D & Thiele L, 2007, *The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration*, Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Matsushima, pp. 862–876.

[221] Zurn HH & Quintana VH, 1975, *Generator maintenance scheduling via successive approximations dynamic programming*, IEEE Transactions on Power Apparatus and Systems, **94(2)**, pp. 665–671.

[222] Zurn HH & Quintana VP, 1977, *Several objective criteria for generator preventive maintenance scheduling*, IEEE Transactions on Power Apparatus and Systems, **96(3)**, pp. 984–992.

IEEE-RTS reliability and outage data

The IEEE-RTS power generating system's capacity outage probability values in Table A.2 of this appendix have been prepared by means of the recursive/convolution algorithm (2.47) from the units' forced outage rates in Table A.1, assuming that all 32 units adhere to the dual-state model in Figure 2.14(a).

Table A.1: *The original IEEE-RTS power generating system's unit reliability data [9].*

Type of Unit	Fuel	Unit Size (MW)	# of Units	FOR	MTTF (hours)	MTTR (hours)	Scheduled Maintenance (weeks/year)
Fossil Steam	#6 Oil	12	5	0.02	2 940	60	2
Combustion Turbine	#2 Oil	20	4	0.10	450	50	2
Hydro		50	6	0.01	1 980	20	2
Fossil Steam	Coal	76	4	0.02	1 960	40	3
Fossil Steam	#6 Oil	100	3	0.04	1 200	50	3
Fossil Steam	Coal	155	4	0.04	960	40	4
Fossil Steam	#6 Oil	197	3	0.05	950	50	4
Fossil Steam	Coal	350	1	0.08	1 150	100	5
Nuclear	LWR	350	1	0.12	1 100	150	6

APPENDIX A

Table A.2: *The capacity outage probability for the IEEE-RTS 32 unit system, prepared by means of the recursive/convolution algorithm (2.47). The values for* $0 \leq X \leq 2\,450$ *are taken from [9], while the values for* $2\,600 \leq X \leq 3\,405$ *are taken from [201, p. 128].*

X	$P(X)$	X	$P(X)$	X	$P(X)$	X	$P(X)$
0	1.000 000	420	0.186 964	1 020	0.003 624	1 650	0.407×10^{-5}
12	0.763 604	440	0.151 403	1 040	0.003 257	1 700	0.158×10^{-5}
20	0.739 482	460	0.137 219	1 060	0.002 857	1 750	0.721×10^{-6}
24	0.634 418	480	0.126 819	1 080	0.002 564	1 800	0.291×10^{-6}
32	0.633 433	500	0.122 516	1 100	0.002 353	1 850	0.152×10^{-6}
36	0.622 712	520	0.108 057	1 120	0.002 042	1 900	0.469×10^{-7}
40	0.622 692	540	0.101 214	1 140	0.001 889	1 950	0.215×10^{-7}
44	0.605 182	560	0.084 166	1 160	0.001 274	2 000	0.724×10^{-8}
48	0.604 744	580	0.075 030	1 180	0.000 925	2 050	0.295×10^{-8}
50	0.604 744	600	0.062 113	1 200	0.000 791	2 100	0.843×10^{-9}
52	0.590 417	620	0.054 317	1 220	0.000 690	2 150	0.305×10^{-9}
56	0.588 630	640	0.050 955	1 240	0.000 603	2 200	0.927×10^{-10}
60	0.588 621	660	0.047 384	1 260	0.000 490	2 250	0.232×10^{-10}
80	0.559 930	680	0.044 769	1 280	0.000 430	2 300	0.797×10^{-11}
100	0.547 601	700	0.042 461	1 300	0.000 401	2 350	0.166×10^{-11}
120	0.512 059	720	0.040 081	1 320	0.000 305	2 400	0.469×10^{-12}
140	0.495 694	740	0.038 942	1 340	0.000 257	2 450	0.104×10^{-12}
160	0.450 812	760	0.030 935	1 360	0.000 164	2 600	0.625×10^{-15}
180	0.425 072	780	0.026 443	1 380	0.000 122	2 800	0.142×10^{-18}
200	0.381 328	800	0.024 719	1 400	0.000 102	3 000	0.982×10^{-24}
220	0.355 990	820	0.018 716	1 420	0.000 084	3 200	0.332×10^{-31}
240	0.346 093	840	0.015 467	1 440	0.000 071	3 400	0.120×10^{-47}
260	0.335 747	860	0.013 416	1 460	0.000 056	3 405	0.120×10^{-47}
280	0.328 185	880	0.012 136	1 480	0.000 046		
300	0.320 654	900	0.011 608	1 500	0.000 040		
320	0.314 581	920	0.009 621	1 520	0.000 027		
340	0.311 752	940	0.008 655	1 540	0.000 020		
360	0.283 619	960	0.006 495	1 560	0.000 013		
380	0.267 902	980	0.005 433	1 580	0.000 010		
400	0.261 870	1 000	0.004 341	1 600	0.000 008		

APPENDIX B

Contents of the accompanying disc

The compact disc included in this dissertation contains the hourly load data (for a year) for the two case studies presented in this dissertation, namely the 32-unit IEEE-RTS inspired case study and the 157-unit Eskom case study. The compact disc contains two `.csv` files. The first one of the files contains the hourly load data used in the 32-unit IEEE-RTS inspired case study and the other contains the hourly load data used in the 157-unit Eskom case study. These files are also in the correct format for the input files as required by the DSS (and as shown in Figure 8.6).