Multiobjective in-core fuel management optimisation for nuclear research reactors

Evert Barend Schlünz

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

Promoter: Prof JH van Vuuren
Co-promoter: Dr PM Bokov

December 2016

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: December 1, 2016

Abstract

The efficiency and effectiveness of fuel usage in a typical nuclear reactor is influenced by the specific arrangement of available fuel assemblies in the reactor core positions. This arrangement of assemblies is referred to as a *fuel reload configuration* and usually has to be determined anew for each operational cycle of a reactor. Very often, multiple objectives are pursued simultaneously when designing a reload configuration, especially in the context of nuclear research reactors. In the *multiobjective in-core fuel management optimisation* (MICFMO) problem, the aim is to identify a Pareto optimal set of compromise or trade-off reload configurations. Such a set may then be presented to a decision maker (*i.e.* a nuclear reactor operator) for consideration so as to select a preferred configuration.

In the first part of this dissertation, a scalarisation-based methodology for MICFMO is proposed in order to address several shortcomings associated with the popular *weighting method* often employed in the literature for solving the MICFMO problem. The proposed methodology has been implemented in a reactor simulation code, called the OSCAR-4 system. In order to demonstrate its practical applicability, the methodology is applied to solve several MICFMO problem instances in the context of two research reactors.

In the second part of the dissertation, an extensive investigation is conducted into the suitability of several multiobjective optimisation algorithms for solving the constrained MICFMO problem. The computation time required to perform the investigation is reduced through the usage of several *artificial neural networks* constructed in the dissertation for objective and constraint function evaluations. Eight *multiobjective metaheuristics* are compared in the context of a test suite of several MICFMO problem instances, based on the SAFARI-1 research reactor in South Africa. The investigation reveals that the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing metaheuristics across the problem instances in the test suite, while the MOVNS algorithm also performs well in the context of bi-objective problem instances. As part of this investigation, a *multiplicative penalty function* (MPF) constraint handling technique is also proposed and compared to an existing constraint handling technique, called *constrained-domination*. The comparison reveals that the MPF technique is a competitive alternative to constrained-domination.

In an attempt to raise the level of generality at which MICFMO may be performed and potentially improve the quality of optimisation results, a *multiobjective hyperheuristic*, called the AMALGAM method, is also considered in this dissertation. This hyperheuristic incorporates multiple metaheuristic sub-algorithms simultaneously for optimisation. Testing reveals that the AMALGAM method yields superior results in the majority of problem instances in the test suite, thus achieving the dual goal of raising the level of generality and of yielding improved optimisation results. The method has also been implemented in the OSCAR-4 system and is applied to solve several MICFMO case study problem instances, based on two research reactors, in order to demonstrate its practical applicability.

Finally, in the third part of this dissertation, a conceptual framework is proposed for an *optimisation-based personal decision support system*, dedicated to MICFM. This framework may serve as the basis for developing a computerised tool to aid nuclear reactor operators in designing suitable reload configurations.

Uittreksel

Die doeltreffendheid en doelmatigheid van brandstofverbruik in 'n tipiese kernreaktor word deur die spesifieke rangskikking van beskikbare brandstofelemente in die laaiposisies van die reaktor beïnvloed. Hierdie rangskikking staan bekend as 'n *brandstof herlaaikonfigurasie* en word gewoonlik opnuut bepaal vir elke operasionele siklus van 'n reaktor. Die gelyktydige optimering van veelvuldige doele word dikwels tydens die ontwerp van 'n herlaaikonfigurasie nagestreef, veral binne die konteks van navorsingsreaktore. Die doelwit van *meerdoelige binne-kern brandstofbeheeroptimering* (MBKBBO) is om 'n Pareto optimale versameling van herlaaikonfigurasieafruilings te identifiseer. Só 'n versameling mag dan vir oorweging (deur byvoorbeeld 'n kernreaktoroperateur) voorgelê word sodat 'n voorkeurkonfigurasie gekies kan word.

In die eerste gedeelte van hierdie proefskrif word 'n skalariseringsgebaseerde metodologie vir MBKBBO voorgestel om verskeie tekortkominge in die gewilde *gewigverswaringsmetode* aan te spreek. Laasgenoemde metode word gereeld in die literatuur gebruik om die MBKBBO probleem op te los. Die voorgestelde metodologie is in 'n reaktorsimulasiestelsel, bekend as die OSCAR-4 stelsel, geïmplementeer. Om die praktiese toepasbaarheid daarvan te demonstreer, word die metodologie gebruik om 'n aantal MBKBBO probleemgevalle binne die konteks van twee navorsingsreaktore op te los.

In die tweede gedeelte van die proefskrif word 'n uitgebreide ondersoek ingestel om die geskiktheid van verskeie meerdoelige optimeringsalgoritmes vir die oplos van die beperkte MBKBBO probleem te bepaal. Die berekeningstyd wat vir die ondersoek benodig word, word verminder deur die gebruik van *kunsmatige neurale netwerke*, wat in die proefskrif gekonstrueer word, om doelfunksies en beperkings te evalueer. Agt *meerdoelige metaheuristieke* word binne die konteks van verskeie MBKBBO toetsprobleemgevalle vergelyk wat op die SAFARI-1 navorsingsreaktor in Suid-Afrika gebaseer is. Toetse dui daarop dat die NSGA-II, die P-ACO algoritme en die MOOCEM oor die algemeen die beste oor al die toetsprobleemgevalle presteer. Die MOVNS algoritme presteer ook goed in die konteks van tweedoelige probleemgevalle. 'n *Vermenigvuldigende boetefunksie* (VBF) beperkinghanteringstegniek word ook voorgestel en vergelyk met 'n bestaande tegniek bekend as *beperkte dominasie*. Daar word bevind dat the VBF tegniek 'n mededingende alternatief tot beperkte dominasie is.

'n Poging word aangewend om die vlak van algemeenheid waarmee MBKBBO uitgevoer word, te verhoog, asook om potensieel die kwaliteit van die optimeringsresultate te verbeter. 'n *Meerdoelige hiperheuristiek*, bekend as die AMALGAM metode, word in die nastreef van hierdie twee doelwitte oorweeg. Die metode funksioneer deur middel van die gelyktydige insluiting van 'n aantal metaheuristieke deel-algoritmes. Toetse dui daarop dat the AMALGAM metode beter resultate vir die meerderheid van toetsprobleme lewer, en dus word die bogenoemde twee doelwitte bereik. Die metode is ook in the OSCAR-4 stelsel geïmplementeer en word gebruik om 'n aantal MBKBBO gevallestudie probleemgevalle (binne die konteks van twee navorsingsreaktore) op te los. Sodoende word die praktiese toepasbaarheid van die metode gedemonstreer.

In die derde deel van die proefskrif word 'n konseptuele raamwerk laastens vir 'n *optimeringsgebaseerde persoonlike besluitsteunstelsel* gemik op MBKBB, voorgestel. Hierdie raamwerk mag as grondslag dien vir die ontwikkeling van 'n gerekenariseerde hulpmiddel vir kernreaktoroperateurs om aanvaarbare herlaaikonfigurasies te ontwerp.

Acknowledgements

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

- My promotor, Prof Jan van Vuuren, for his guidance and support throughout the duration of this project. I appreciate his time, dedication and hard work in ensuring that work of a high standard is delivered.
- My co-promotor, Dr Pavel Bokov, also for his guidance and support throughout the duration of this project. I appreciate his dedication, valuable insight and attention to detail, ensuring that work of a high standard is delivered.
- This work was financially supported in part by Stellenbosch University via a postgraduate merit bursary, by the *National Research Foundation* (NRF) of South Africa (under grant number 88003), and by Necsa via their study assistance scheme and co-contributions to the NRF grant. Any opinion, finding and conclusion or recommendation expressed in this material is that of author and the NRF does not accept any liability in this regard.
- Dr Rian Prinsloo, for his technical assistance in all matters pertaining to the OSCAR-4 code system.
- My fellow postgraduate students within the *Stellenbosch Unit of Operations Research in Engineering* (SUnORE) and my colleagues at Necsa, for their support, encouragement and the interest they showed towards my studies.
- My friends and family, for their moral support and encouragement during the past four years.
- Finally, my wife, Michelle, for her love and unwavering support, patience and encouragement, especially during the more challenging times of the past four years. None of my achievements would have been possible without her.

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:		
$\boldsymbol{a}$	Symbol denoting a **vector**	(Bold-face lower case letters)
$\boldsymbol{A}$	Symbol denoting a **matrix**	(Bold-face capitals)
$\mathcal{A}$	Symbol denoting a **set** or a **list**	(Calligraphic capitals)

Symbol	**Meaning**
Reactor analysis	
ϕ	Neutron flux
k_{eff}	The effective neutron multiplication factor
ρ_{ex}	Excess reactivity
ρ_{sdm}	Shutdown margin
ρ_{cbw}	Total control bank worth
ψ_{ppf}	Power peaking factor
ϕ_{B12}	Average thermal neutron flux over the faces of beam tubes 1 and 2
ϕ_{B5}	Average thermal neutron flux over the face of beam tube 5
ϕ_{S}	Average thermal neutron flux over the silicon doping facility
ϕ_{I1}	Maximum axial thermal neutron flux in the first IPR facility
ϕ_{I2}	Maximum axial thermal neutron flux in the second IPR facility
$\psi_{\text{Mo}}^{\text{tot}}$	Assembly-averaged power levels in all molybdenum rigs
$\psi_{\text{Mo}}^{\text{min}}$	Assembly-averaged power level in the molybdenum rig with the minimum power
$k_{\text{eff}}^{\text{sm}}$	The effective neutron multiplication factor of the reactor when the two most reactive control rods are fully extracted from the core
Sets	
$\mathbb{R}^n$	The set of all real numbers in n-space
$\mathbb{Z}^n$	The set of all integers in n-space
$\mathcal{X}$	The set of all possible fuel reload configurations for an ICFMO problem
$\mathcal{S}$	The feasible region within the decision space of an optimisation problem
$\mathcal{Z}$	The feasible objective space of an optimisation problem
$\mathcal{P}_S$	The Pareto set of an MOP
$\mathcal{P}_F$	The Pareto front of an MOP

$\widetilde{\mathcal{P}}_S$	An approximate Pareto set for an MOP
$\widetilde{\mathcal{P}}_F$	An approximate Pareto front for an MOP
$\mathcal{S}_S$	The feasible region in the decision space of an ICFMO problem for the SAFARI-1 reactor
$\mathcal{S}_H$	The feasible region in the decision space of an ICFMO problem for the HOR reactor
Vectors	
$\boldsymbol{f}(\boldsymbol{x})$	A vector of objective function values in an MOP, also referred to as an objective vector
$\boldsymbol{z}$	An alternate notation for an objective vector
$\boldsymbol{z}^\star$	The ideal objective vector associated with an MOP
$\boldsymbol{z}^{\star\star}$	A utopian objective vector associated with an MOP
$\bar{\boldsymbol{z}}$	An aspiration vector
Parameters	
q	The number of objective functions in an MOP
$z_i^\star$	The i-th component in the ideal objective vector $\boldsymbol{z}^\star$
$z_i^{\star\star}$	The i-th component in a utopian objective vector $\boldsymbol{z}^{\star\star}$
$\bar{z}_i$	An aspiration level associated with the i-th objective
$g_i^{\lim}$	A non-zero limiting value associated with the i-th inequality constraint function $g_i(\boldsymbol{x})$
$h_i^{\lim}$	A non-zero limiting value associated with the i-th equality constraint function $h_i(\boldsymbol{x})$
w_i	The i-th weighting coefficient associated with the i-th objective function
μ	A sufficiently small positive scalar employed in the augmentation term of a scalarising objective function
γ	A strictly positive severity factor employed in the penalty functions $P_a(\boldsymbol{x})$ and $P_m(\boldsymbol{x})$
ρ	The Pareto rank of a solution
Functions	
$f_i(\boldsymbol{x})$	The i-th objective function in an MOP
z_i	An alternate notation for the i-th objective function
$g_i(\boldsymbol{x})$	The i-th inequality constraint function in an optimisation problem
$h_i(\boldsymbol{x})$	The i-th equality constraint function in an optimisation problem
U	A utility function
$\tilde{F}_q(\boldsymbol{x})$	A scalarising objective function for q different objectives in which the augmented Chebyshev metric is employed
$G(\boldsymbol{x})$	The total scaled constraint violation associated with the inequality constraints in an optimisation problem
$H(\boldsymbol{x})$	The total scaled constraint violation associated with the equality constraints in an optimisation problem
$P_a(\boldsymbol{x})$	The penalty function employed in the APF constraint handling technique
$F_q(\boldsymbol{x})$	A scalarising objective function for q different objectives in which the augmented Chebyshev metric and a penalty function are employed
$P_m(\boldsymbol{x})$	The penalty function employed in the MPF constraint handling technique

Relational symbols	
$\succ$	A binary relational symbol denoting Pareto dominance between two vectors
Multiobjective optimisation algorithms	
p_{hm}	The harmony memory consideration rate
p_{par}	The pitch adjustment rate
b_{par}	The bandwidth parameter
p_c	The crossover probability
p_m	The mutation probability
$\boldsymbol{\tau}^k$	The pheromone matrix associated with objective k
τ_{ij}^k	The pheromone level associated with objective k for edge (i, j)
$\boldsymbol{\eta}$	A heuristic information matrix
η_{ij}	The heuristic information for edge (i, j)
α	A bias parameter which favours pheromones
β	A bias parameter which favours heuristic information
r_0	The probability that the next vertex should be selected as the one having the highest aggregate value of pheromone and heuristic information
τ_0	The initial pheromone level
ϱ	The pheromone evaporation rate
$T_{\max}$	The maximum temperature
$\overline{N}_{\text{HL}}$	The hard limit for the archive size
$\overline{N}_{\text{SL}}$	The soft limit for the archive size
φ	The geometric cooling rate
ω	The smoothing parameter
ρ_E	The ranking threshold
S_{t+1}^i	The number of solutions created by sub-algorithm i as a contribution to the parent population during generation $t + 1$
N_{t+1}^i	The number of offspring solutions that sub-algorithm i should create during generation $t + 1$
Performance analysis and statistical testing	
I_{HVD}	The hypervolume difference to reference indicator
I_{R2}	The $R2$ indicator
H_0	The null hypothesis
H_1	The alternative hypothesis
$\tilde{\alpha}$	The significance level
R_{avg}	The average rank value

List of Acronyms

AbYSS	Archive-based Hybrid Scatter Search
ACO	Ant Colony Optimisation
ACS	Ant Colony System
APF	Additive Penalty Function
AS	Ant System
AM-1	Atom Mirny 1
AMALGAM	A Multi-algorithm, Genetically Adaptive Multiobjective
AMOSA	Archived Multiobjective Simulated Annealing
ANN	Artificial Neural Network
AOS	Auxiliary Optimisation System
BAP	Buffer Allocation Problem
BWR	Boiling Water Reactor
BOC	Beginning-of-cycle
CDP	Constrained-domination Principle
CEM	Cross-entropy Method
CIGARO	Code Independent Genetic Algorithm Reactor Optimization
CP-1	Chicago Pile Number One
CPU	Central Processing Unit
CX	Cycle Crossover
DBMS	Database Management System
DSS	Decision Support System
eV	Electron Volt
FE	Function Evaluator
FNSA	Fast Nondominated Sorting Algorithm
FORMOSA	Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing
GA	Genetic Algorithm
GARCO	Genetic Algorithm Reactor Code Optimization
GB	Gigabyte
GHz	Gigahertz
HEU	Highly Enriched Uranium
HHRC	Historical HOR Reload Configuration
HM	Harmony Memory
HMCR	Harmony Memory Consideration Rate
HMI	Human Machine Interface
HMLF	Highest-mass to Lowest-flux
HOR	Hoger Onderwijs Reactor
HS	Harmony Search

HSRC	Historical SAFARI-1 Reload Configuration
IAEA	International Atomic Energy Agency
ICFM	In-core Fuel Management
ICFMO	In-core Fuel Management Optimisation
IPR	Isotope Production Rig
L^{3}P	Low Leakage Loading Pattern
LEU	Low Enriched Uranium
MCDA	Multiple Criteria Decision Analysis
MeV	Mega-electron Volt
MFNN	Multilayer Feedforward Neural Network
MICFM	Multiobjective In-core Fuel Management
MICFMO	Multiobjective In-core Fuel Management Optimisation
MMEA	Probabilistic Model-based Multiobjective Evolutionary Algorithm
MOA	Multiobjective Optimisation Algorithm
MOEA	Multiobjective Evolutionary Algorithm
MOEA/D	Multiobjective Evolutionary Algorithm Based on Decomposition
MOGA	Multiobjective Genetic Algorithm
MOHS	Multiobjective Harmony Search
MOO	Multiobjective Optimisation
MOOCEM	Multiobjective Optimisation using the Cross-entropy Method
MOP	Multiobjective Optimisation Problem
MOSA	Multiobjective Simulated Annealing
MOTGA	Multiobjective Tchebycheff-based Genetic Algorithm
MOVNS	Multiobjective Variable Neighbourhood Search
MPF	Multiplicative Penalty Function
MSE	Mean Squared Error
MW	Megawatt
MWd	Megawatt-days
NRF	National Research Foundation
NEA	Nuclear Energy Agency
Necsa	The South African Nuclear Energy Corporation SOC Limited
NFL	No Free Lunch
NRG	Nuclear Research and consultancy Group
NSGA-II	Nondominated Sorting Genetic Algorithm II
NWWM	Nemenyi, Wilcoxon-Wilcox, Miller
OCFM	Out-of-core Fuel Management
OMOPSO	Optimised Multiobjective Particle Swarm Optimisation
OE	Optimisation Engine
OSCAR-4	Overall System for the Calculation of Reactors, Version 4
P-ACO	Pareto Ant Colony Optimisation
PAR	Pitch Adjustment Rate
ParEGO	Pareto Efficient Global Optimization
PMX	Partially Matched/Mapped Crossover
POS	Position-based Crossover
PG	Problem Generator
PSO	Particle Swarm Optimisation
PWR	Pressurised Water Reactor
R2-IBEA	R2 Indicator-based Evolutionary Algorithm
RAM	Random Access Memory
ROSA	Reload Optimization by Simulated Annealing

SA	Simulated Annealing
SAFARI-1	South Africa Fundamental Atomic Research Installation One
SOP	Single-objective Optimisation Problem
SPEA2	Strength Pareto Evolutionary Algorithm 2
TRIGA	Training, Research, Isotopes, General Atomics
VEPSO	Vector Evaluated Particle Swarm Optimisation
VRPSTW	Vehicle Routing Problem With Soft Time Windows
VVER	Water-Water Energetic Reactor

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

In 1920, Ernest Rutherford first theorised the existence of a neutrally-charged atomic particle which was in time to be called the *neutron* [178]. It was not until 1932, however, that the neutron was finally discovered [50]. Following this discovery, scientists investigated the nuclear reactions produced by bombarding various elements with this neutrally-charged particle. One such experiment proved that bombarding uranium (a heavy element with an atomic number of 92) with neutrons produced, amongst others, isotopes of barium (a much lighter element with an atomic number of 56) [50]. An explanation of this phenomenon suggested that the uranium nucleus, after capturing a neutron, "*divided itself into two nuclei of roughly equal size*" [135]. This new nuclear reaction was referred to as *fission* and it was calculated that a large amount of energy is emitted by it. Not long thereafter it was proved experimentally that a fission reaction also emitted additional neutrons which could induce fission in other uranium nuclei [50]. This discovery revealed that a self-sustaining chain reaction, in which significant amounts of energy may be released, was possible.

1.1 Background

At 15:25 on 2 December 1942, the first man-made, controlled, self-sustaining nuclear chain reaction was produced by Enrico Fermi and his colleagues [107]. The reaction was produced in the world's first nuclear reactor[1], the *Chicago Pile Number One* (CP-1), which was built on a squash court at the University of Chicago. The CP-1 reactor was literally a pile (hence the name) of graphite blocks that enclosed pieces of uranium metal and uranium oxide as fuel [107]. An artist's rendition of the CP-1 reactor and the events of 2 December 1942 may be found in Figure 1.1.

[1] A *nuclear reactor* is a device in which a controlled nuclear fission chain reaction can be maintained [42].

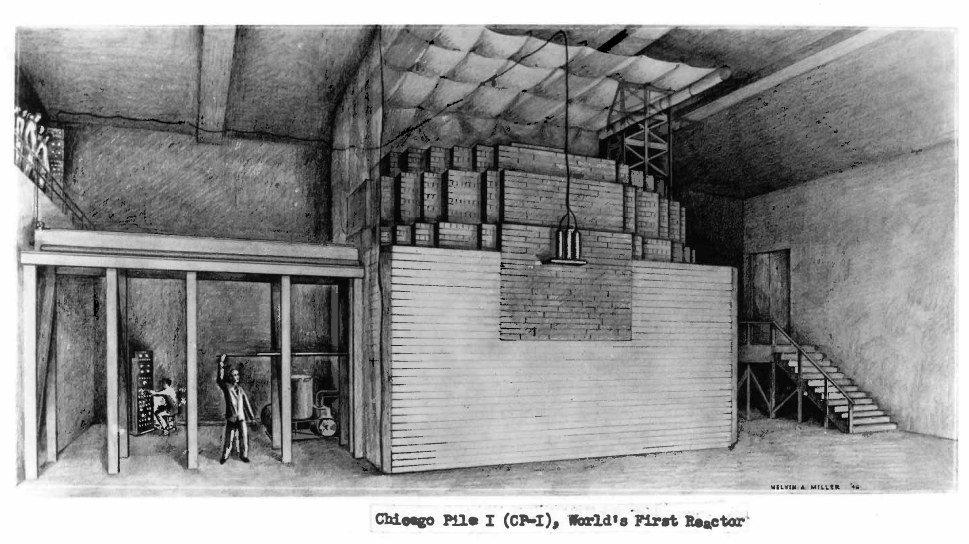


FIGURE 1.1: *Artist's rendition of the CP-1 reactor and the events of 2 December 1942 [6].*

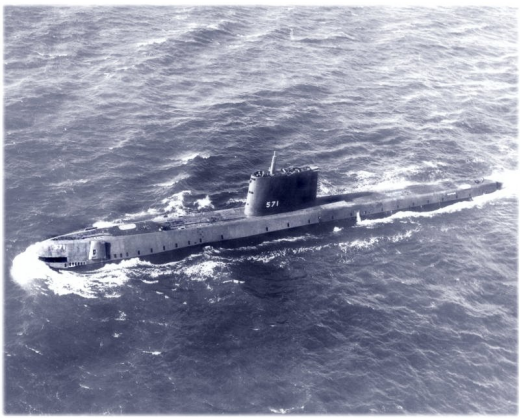
The successful experiment of the CP-1 reactor was a major step in the *Manhattan Project*, the American research project during World War II that saw the development of the first atomic (nuclear fission) bomb [49, 107]. Following the experiment, so-called *production reactors* were quickly designed and built for the purpose of producing plutonium[2] to be used in atomic bombs. The X-10 reactor at Oak Ridge, Tennessee, was the first of these reactors and was built as an interim measure until a large-scale production reactor at Hanford, Washington, could be completed [51]. By 1945, sufficient amounts of plutonium had been produced for a nuclear explosive device, and the corresponding weapon development and design were advanced enough to proceed with a field test. Under the code name *Trinity*, the first atomic bomb successfully exploded at Alamogordo, New Mexico at 05:30 on 16 July 1945 [49].

America was not the only country that designed and built nuclear reactors during those early years. The first Canadian reactor was completed in 1945, whereas the Soviet Union achieved their first controlled fission chain reaction in 1946 using the F-1 pile. Europe followed shortly thereafter, with the first British and French nuclear reactors starting up in 1947 and 1948, respectively [51].

After World War II, research efforts could also be focussed toward nuclear power generation, although weapons development still continued. Significant efforts went into developing small, compact reactors for maritime applications and it culminated in the launch of the first nuclear-powered submarine in 1954, called the *Nautilus* (see Figure 1.2(a)). It was only in 1959, however, that the first nuclear-powered surface vessel was put into service, namely the icebreaker *Lenin* shown in Figure 1.2(b).

Countries also invested heavily into nuclear-powered electricity generation research, shifting the focus away from plutonium production reactors towards commercial power reactors [51]. Several power reactor types were developed during this research phase, a listing of which may be found in [51]. Of these reactor types, the *pressurised water reactor* (PWR) and the *boiling water reactor* (BWR) are arguably the most prominent. In a PWR, the reactor core is cooled by water maintained at a high pressure of approximately 15 megapascal (so as to prevent boiling) [111]. In a BWR, however, the reactor core is cooled by water which is then allowed to boil, thus producing steam for a turbine [111]. Both reactor types are examples of *thermal light water* nuclear

[2]Plutonium was discovered in 1941 during an experiment in which uranium was bombarded by deuteron [52].



(a) *The submarine Nautilus*

(b) *The icebreaker Lenin*

FIGURE 1.2: *The first nuclear-powered submarine and surface vessel (the Nautilus [207] and the icebreaker Lenin [5], respectively).*

reactors because they primarily employ low-energy neutrons to induce fission, and utilise H_2O as coolant. The first nuclear reactor that produced electricity, in 1951, was the Experimental Breeder Reactor Number 1, although it was not designed for electricity production. In that respect, the world's first nuclear-powered electricity generator was the Atom Mirny 1 (AM-1) reactor at the Obnisk Nuclear Power Station and it produced electricity from 1954 until 1959. Several other prototype nuclear power reactors were built all over the world during that time, followed by large-scale commercial nuclear power plants in the decades thereafter. According to the *International Atomic Energy Agency* (IAEA) there are 438 nuclear power reactors in operation world-wide, as of 31 December 2014 [151].

Apart from power reactors, there are also numerous nuclear research reactors the world over. The primary use of a research reactor is to serve as a neutron source for research and other applications, as well as for training and education purposes. The research component generally revolves around the study of material properties, whereas fields of application include medicine, biology, agriculture, chemistry and new technologies [173]. Research reactors are much smaller and simpler than power reactors, operating at lower temperatures and using less fuel. According to the research reactor database of the IAEA [88], there are 246 research reactors currently operational in the world (as of November 2015). A variety of different research reactor designs prevail, with a common design being the pool-type reactor. In such a reactor, the core consists of a cluster of fuel assemblies residing within a large open pool of water. There are typically control rods and empty spaces (for experiments) between the fuel assemblies [173]. In Figure 1.3, a picture of a TRIGA reactor[3] core is presented as an example of a popular pool-type reactor.

Nuclear reactors are not without controversy, primarily due to safety concerns and public fears following nuclear accidents. The two worst nuclear accidents in the world, according to the *international nuclear event scale* introduced by the IAEA, were those of Chernobyl and Fukushima. In 1986, a combination of human error and a flawed reactor design led to the accident at the Chernobyl nuclear power plant in Ukraine. A steam explosion and several fires thereafter caused a significant release of radioactive material into the environment. The accident resulted in a number of human deaths, an increased prevalence of thyriod cancer among the affected communities in the area, and the evacuation of more than 100 000 people [234]. More recently, in 2011, a tsunami (caused by a major earthquake off the coast of Japan) hit the Fukushima Daiichi

[3]The acronym TRIGA stands for *Training, Research, Isotopes, General Atomics* [173].

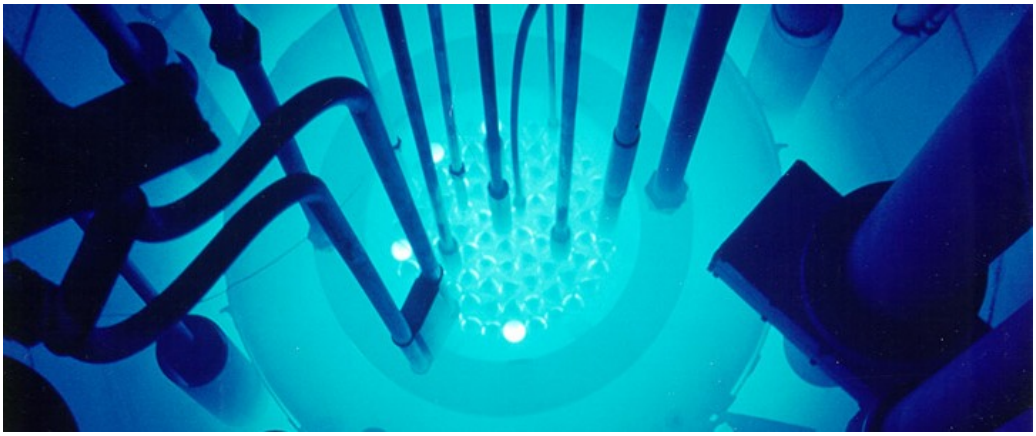


FIGURE 1.3: *The core of a TRIGA nuclear reactor [66].*

nuclear power plant with a wave height of approximately 15 meters. Although the reactors at the plant withstood the earthquake and the hit from the wave, subsequent flooding disabled almost all of the cooling systems. Three of the reactor cores ultimately melted as a result, and radioactive material was released into the ocean and atmosphere. Fortunately, no nuclear-related deaths have been recorded from the accident [234]. Despite the safety concerns, an analysis by the *Nuclear Energy Agency* (NEA) in 2010 revealed that nuclear energy had the safest energy chain[4] during the period 1969–2000 when compared to fossil and hydro energy chains, measured in terms of fatalities [149].

According to the IAEA, world-wide electrical generating capacity is estimated to almost double by 2030 (from its 2014 level), with nuclear generating capacity likely accounting for approximately 5% of the total [54]. With seventy power reactors [151] and six research reactors [88] under construction across the world as of 31 December 2014, the figures show that the future holds much promise for nuclear energy.

1.2 Informal problem description

During the operation of a typical nuclear reactor, a fraction of fuel assemblies are periodically replaced with fresh ones in order to sustain the fission chain reaction occurring in the reactor core. This time period between the reloading of fuel is referred to as an *operating cycle* for the reactor. The efficiency and effectiveness of fuel usage in a reactor is influenced by the specific arrangement of available fuel assemblies in the reactor core positions. This arrangement of assemblies is referred to as a *fuel reload configuration* (or fuel loading pattern). In general terms, a reload configuration should be designed with the aim to optimise some reactor performance objective(s), subject to operational and/or safety constraints. The problem of finding such an optimal fuel reload configuration for a nuclear reactor core is known as the *in-core fuel management optimisation* (ICFMO) problem.

A topic closely related to ICFMO is known as *out-of-core fuel management*, and it generally entails making long-term (*i.e.* multi-cycle) fuel management decisions. These include, for example, determination of the operational cycle length, as well as the number and type of fresh fuel assemblies to consider [217]. This topic, however, is not considered in this dissertation.

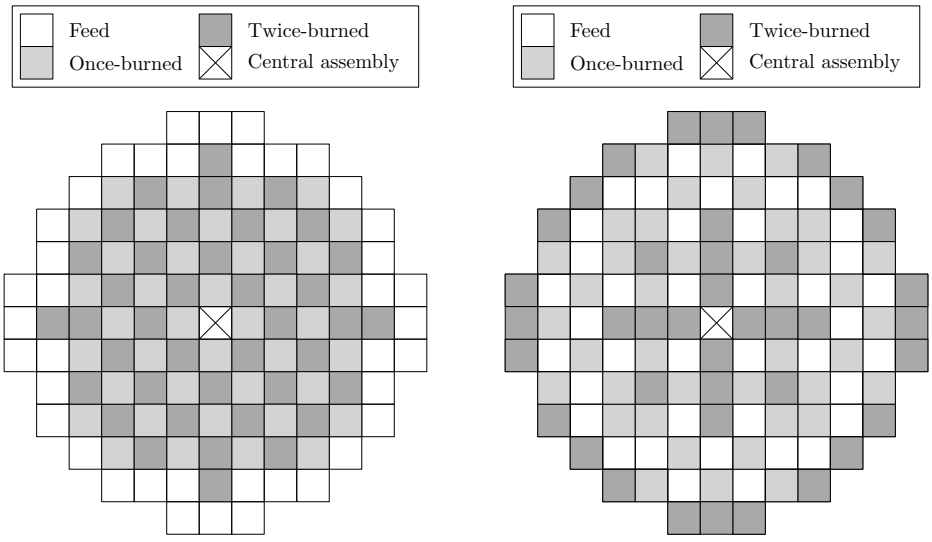
[4] According to the NEA report, "*an energy chain comprises exploration, extraction, transport, storage, power and/or heat generation, transmission, local distribution, waste treatment and disposal*" [149].

The ICFMO problem is a difficult nonlinear combinatorial optimisation problem which requires time-consuming reactor core simulation calculations in order to evaluate the quality of potential solutions [217, 218]. ICFMO is also very context-specific given the various types of reactors currently in operation across the world (differing, for example, in their design, size, geometry, fuel type, utilisation requirements, and so forth).

1.2.1 ICFMO within the context of power reactors and research reactors

In the context of power reactors, safety and the economical usage of fuel are two of the primary objectives pursued in the design of a reload configuration. Consider, for example, a typical PWR with three fuel regions[5]. These regions are generally classified according to the number of operating cycles during which the assemblies have remained in the core *e.g. once-burned* or *twice-burned*. Otherwise, if the fuel assemblies are fresh, the region is classified as having *feed* assemblies.

A very popular reload configuration for PWR cores is known as an *out-in checkerboard* design. In this design, feed assemblies are loaded into the peripheral core positions. In order to flatten the (radial) power profile across the core and prevent a power peak from occurring, the burned regions' assemblies are then loaded according to a "checkerboard" pattern in the interior positions of the core. As such, this design is motivated from the point of view of a safety-related objective. An example of a three-region *out-in checkerboard* design is presented in Figure 1.4(a). Note that the central fuel assembly is treated separately. According to Turinsky [217], this reload configuration was the preferred design for many years due to its effectiveness in respect of minimising power peaking within a reactor core. A major drawback of the design, however, is the fact that large numbers of neutrons leak out of the core. Less energy is therefore extracted from the fuel, resulting in a detrimental effect in the economical usage thereof.



(a) *An* out-in checkerboard *reload configuration (reproduced from [195])*

(b) *An* L^3P *reload configuration (adapted from [217])*

FIGURE 1.4: *Examples of PWR reload configurations.*

[5] A subset of fuel assemblies that are loaded into the core together is known as a *fuel region* [217].

More recently, a reduction in neutron leakage out of the core has been pursued in an attempt to improve the economical usage of fuel, as well as to reduce the radiation damage inflicted to the reactor vessel. This may be achieved by loading the feed assemblies away from the peripheral core positions. A compromise, however, has to be struck such that power peaking considerations are still adhered to in the interior of the core. The resulting reload configuration is referred to as a *low leakage loading pattern* (L^3P) design. In such a design, some of the most-burned fuel assemblies are loaded into the peripheral core positions (*i.e.* the high leakage positions). The majority of feed assemblies are then loaded adjacent to the aforementioned burned ones. Often, these feed assemblies are referred to as a *ring of fire* since they tend to operate at the highest power levels [217]. The remaining assemblies are finally loaded according to a checkerboard pattern in the interior positions of the core. Preference is, however, given to loading a most-burned assembly adjacent to a feed assembly. An example of a three-region L^3P design is presented in Figure 1.4(b).

In the context of research reactors, other objectives are often pursued since these reactors serve primarily as neutron sources for research and experimental purposes, as well as other irradiation applications. Accordingly, the design approaches adopted in power reactor reload configurations are not necessarily transferable to research reactors. Further complicating this observation is the fact that research reactor core geometries are often asymmetric, with additional (non-fuel) components also present throughout the core. This results in core layouts that are much more "heterogeneous" than those of power reactors, thus contributing to the difficulty of designing good reload configurations. The McMaster Nuclear Reactor's reference core configuration, as specified in an IAEA benchmarking database [175], is presented in Figure 1.5 as an example of such an heterogeneous core layout.

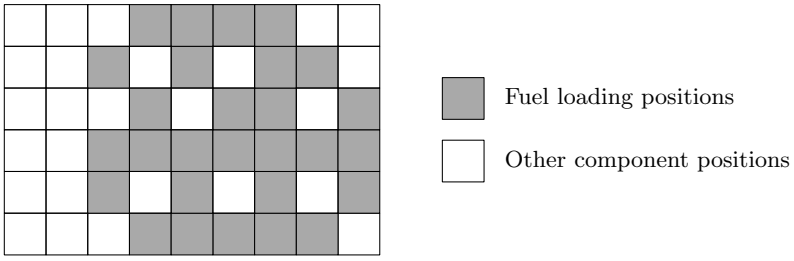


FIGURE 1.5: *The McMaster Nuclear Reactor reference core configuration (adapted from [175]).*

1.2.2 The South African context

There are three nuclear reactors currently in operation in South Africa. The Koeberg Nuclear Power Station houses two power reactors of PWR type, and is operated by the state-owned utility *Eskom*. The third is a research reactor (of tank-in-pool type) called SAFARI-1, which is an acronym for *South Africa Fundamental Atomic Research Installation One*. The reactor is owned and operated by the *South African Nuclear Energy Corporation SOC Limited* (Necsa) at their Pelindaba site. Photographs of the Koeberg power station and the Pelindaba site are presented in Figure 1.6.

SAFARI-1 is widely regarded as one of the highest utilised research reactors in the world, achieving an average operational availability of more than 300 days per year [145]. The reactor's utilisation is strongly focussed towards commercial irradiation services, as well as nuclear research, training and materials testing activities. These different goals are often conflicting and, in conjunction with safety and operational constraints, they present a particular challenge to a

(a) *The Koeberg Nuclear Power Station [161]*

(b) *The Pelindaba site, home to the SAFARI-1 reactor [146]*

FIGURE 1.6: *South African nuclear reactors.*

SAFARI-1 operator tasked with designing an appropriate reload configuration. This challenge is, however, not restricted to SAFARI-1 as there are other research reactors in the rest of the world operating under similar conditions (*i.e.* with multipurpose utilisation goals). Accordingly, finding an "optimal" reload configuration for SAFARI-1 and similar-type reactors requires consideration of multiple objectives simultaneously. These multiple objectives may also change from one operational cycle to the next, when a multipurpose research reactor has to respond to a change in its research or commercial demands. Flexibility in the reload configuration design, on a cycle-to-cycle basis, is therefore essential for the successful operation of such a reactor so as to accommodate its utilisation requirements.

This dissertation is concerned with the topic of constrained *multiobjective in-core fuel management optimisation* (MICFMO) for nuclear research reactors, with a particular emphasis on the SAFARI-1 reactor in South Africa. In *multiobjective optimisation* (MOO), the aim is to identify a set of compromise or trade-off solutions, based on the notion of *Pareto optimality*. Once such a set of solutions has been identified, it may be presented to a decision maker (*i.e.* a nuclear reactor operator) for consideration so as to select a preferred solution. Accordingly, optimisation-based decision support may be rendered to a reactor operator tasked with designing a reload configuration by solving the particular MICFMO problem instance.

1.3 Prelude to this dissertation

One of the tools employed at Necsa to provide calculation support to the SAFARI-1 reactor is the in house-developed reactor core calculation code system OSCAR-4. The name is an acronym for *Overall System for the Calculation of Reactors, version 4*. There was an initial effort during 2009 and 2010 by the developers of OSCAR-4 to include MICFMO capabilities in the system as a decision support feature. That effort was, however, put on hold because the optimisation methodology employed exhibited several shortcomings. The first of these was the use of a rudimentary scalarising objective function in which minimisation objectives and the reciprocals of maximisation objectives were aggregated together by means of a linear weighted sum. Furthermore, the notion of Pareto optimality was not considered in the methodology and, as a result, it did not yield a set of trade-off solutions. The final shortcoming was the lack of a constraint handling technique. A metaheuristic, called *harmony search* [64], was employed as the solution technique in the methodology.

There was renewed interest in 2012 to implement a working version of the MICFMO decision support feature in OSCAR-4 [182]. Some of the shortcomings in the original optimisation methodology were therefore addressed. In particular, the scalarising objective function was altered slightly and an additive penalty function was incorporated as a constraint handling technique. In the altered scalarising objective function, a maximisation objective was transformed into a minimisation objective simply by multiplying it by -1 (instead of taking its reciprocal). Scaling factors were further introduced for all objectives in order to create dimensionless function values that are equally scaled by order of magnitude. These scaled objectives were aggregated together, as before, by means of a linear weighted sum. In the additive penalty function, penalty values related to the magnitude of the constraint violations were calculated and subsequently added to the scalarising objective function. The harmony search algorithm was employed without modification, again, as the solution technique in the methodology. The notion of Pareto optimality, however, was still not incorporated during optimisation.

The author of this dissertation inherited the aforementioned working version of the MICFMO decision support feature in the OSCAR-4 system. The feature was, however, recognised to be inadequate for rendering advanced MICFMO decision support to users of the system. Two reasons for this are the naive scalarising objective function and the lack of incorporating the notion of Pareto optimality during optimisation. A priority in this dissertation is therefore to address the remaining shortcomings present in this inherited methodology. Another priority is to investigate alternative multiobjective computational methods in terms of their suitability in finding sets of high-quality trade-off solutions to the MICFMO problem. In doing so, state-of-the-art methods may be incorporated into the OSCAR-4 feature, thus enabling it to render advanced MICFMO decision support.

1.4 Dissertation scope and objectives

The following twelve objectives are pursued in this dissertation:

I To *provide* an introduction to nuclear reactor analysis and relevant information pertaining to nuclear reactors so as to garner a basic understanding of the concepts and terminology employed in the field.

II To *perform* a literature survey on the topic of ICFMO, focussing on:

(a) Popular objective functions and constraints adopted in existing model formulations.

(b) Typical solution techniques previously employed to solve the problem (in the contexts of both single-objective and multiobjective optimisation).

(c) Approaches adopted to reduce the computational burden associated with ICFMO.

III To *formulate* a suitable model for the MICFMO problem.

IV To *propose* a scalarisation-based approach to MICFMO in order to address the shortcomings of the inherited methodology in the OSCAR-4 system, to *implement* this approach within the system, and to *compare* its effectiveness in solving MICFMO problem instances to those of current approaches employed at different research reactors.

V To *construct* surrogate calculation models for predicting SAFARI-1 reactor core parameters by using artificial neural networks.

VI To *establish* a test suite of several constrained MICFMO problem instances, based on the SAFARI-1 reactor.

VII To *compare* several multiobjective metaheuristics for solving the constrained MICFMO problem in a structured and statistically sound manner based on the test suite of problem instances established in pursuit of Objective VI.

VIII To *propose* a new (multiplicative) penalty function constraint handling technique for MOO and *compare* it to an existing technique called *constrained-domination* [34] in the context of the test suite established in pursuit of Objective VI.

IX To *investigate* the suitability of several variations of a multiobjective *hyperheuristic* known as the AMALGAM[6] method [226] in terms of solving the constrained MICFMO problem by incorporating the findings of Objective VII into the method.

X To *implement* the preferred variation of the AMALGAM method within the OSCAR-4 system and to *apply* it to carry out different research reactor MICFMO case studies.

XI To *propose* a generic optimisation-based decision support system framework for multiobjective in-core fuel management.

XII To *propose* follow-up work related to the contents of this dissertation which may be pursued in the future.

The scope of this dissertation is restricted to the MICFMO problem found within the broader context of nuclear fuel management, and therefore excludes out-of-core fuel management optimisation, as well as any fuel cycle[7] optimisation. Only single-cycle optimisation is considered within the context of thermal light water nuclear research reactors.

Utilisation of a reactor core calculation code system in this dissertation is restricted to that of the OSCAR-4 system, involving only neutronic calculations. A full understanding, and an evaluation of the accuracy and appropriateness of the system (*i.e.* the mathematical models and their associated numerical solution schemes) are considered to fall outside the scope of this dissertation. Accordingly, the OSCAR-4 system is employed throughout the dissertation as a "black-box" function evaluator during MICFMO.

The specific values of operational targets and/or limits for the nuclear reactors considered in this dissertation are proprietary knowledge and are therefore not divulged. All computational results reported in this dissertation are, however, scaled with respect to these values. In order to allow for the recovery of true values from the computational results, the scaling constants are documented in a confidential report to be kept at Necsa [181].

Finally, only a conceptual framework of a decision support system for multiobjective in-core fuel management is considered in this dissertation. Populating the components of the system, as well as the subsequent implementation of the system on a personal computer, falls outside the scope of this dissertation.

[6] AMALGAM is an acronym for *a multi-algorithm, genetically adaptive multiobjective*.

[7] The *fuel cycle* mentioned here refers to the overall life-cycle of nuclear fuel, *e.g.* from the mining of raw materials through to nuclear waste disposal [217].

1.5 Dissertation organisation

This dissertation comprises eleven further chapters, five appendices and a bibliography, following this introductory chapter. In Chapter 2, the reader is introduced to some of the fundamental concepts and terminology found in nuclear reactor analysis and theory. The most important nuclear reactions are briefly described, followed by a more detailed discussion on nuclear fission and its chain reaction. A brief overview of the basic components of a nuclear reactor core is also presented. The chapter closes with a description of the primary neutronic aspects of interest within nuclear reactor analysis, as well as a discussion on the necessity of reactor core calculation code systems.

Chapter 3 contains a survey of the relevant literature pertaining to ICFMO. The aim is to provide the reader with the necessary background knowledge required to proceed with new research towards ICFMO in an informed manner. General ICFMO modelling considerations and an historical overview of early research in the field are presented first. Thereafter, the most popular objective functions and constraints adopted in model formulations for the ICFMO problem are mentioned. The complexity in respect of the ICFMO problem's decision space is also elaborated upon. In the next section, a brief description of solution techniques that are typically applied to solve the problem is presented, with a particular emphasis placed on metaheuristics. The most prominent single- and multiobjective metaheuristics are described in some detail. Finally, a brief discussion follows on different approaches considered in the literature for reducing the computational burden associated with solving instances of the ICFMO problem.

Chapter 4 opens with a brief discussion on the topic of nuclear fuel management, thus placing ICFMO in context within this broader topic. Several problem assumptions are elucidated before the optimisation model adopted in this dissertation for the MICFMO problem is formulated. The reactor core calculation code system, OSCAR-4, which is utilised in this dissertation is also briefly described. In the next sections, two nuclear research reactors, namely the SAFARI-1 and HOR reactors, are discussed in detail since they are considered as case studies later in this dissertation. Typical objectives and constraints associated with each reactor for the MICFMO problem are specified, while a description of the current reload configuration design approach followed at each reactor is also presented.

In the first section of Chapter 5, the reader is provided with a detailed description of the notion of Pareto optimality and other related concepts in order to gain a better understanding of the MOO modelling process and solution techniques employed in the dissertation. The popularly-employed weighting method of incorporating multiple objectives into a single function is described next and its shortcomings are pointed out. Thereafter, a scalarisation-based methodology for MICFMO is proposed in order to address the shortcomings present in the existing optimisation methodology within the OSCAR-4 system. The applicability of the proposed methodology is then demonstrated on problem instances within the context of the SAFARI-1 and HOR reactors.

In Chapter 6, several artificial neural networks are constructed for the prediction of SAFARI-1 core parameters corresponding to various ICFMO objectives and constraints. The chapter opens with a motivation of the necessity of these neural networks. Thereafter, general concepts pertaining to artificial neural networks are presented before moving on to a more comprehensive description of multilayer feedforward neural networks. Details are provided on the construction of a suite of neural networks for the SAFARI-1 core parameters, before the chapter closes with numerical results obtained during the training and application of the networks, as well as a discussion of these results.

In Chapter 7, two constraint handling techniques (of which one is newly-proposed in this dissertation) and eight multiobjective metaheuristics, to be employed in a comparative study later in the dissertation, are discussed. A pseudo-code listing of each metaheuristic is also provided. The chapter closes with a discussion on the topic of performance assessment of multiobjective optimisation algorithms and two performance indicators are identified for use in this dissertation.

In the first section of Chapter 8, a test suite of constrained MICFMO problem instances, based on the SAFARI-1 reactor, is created for use in the algorithmic comparative studies to be conducted in the chapter. The experimental design and the nonparametric statistical testing procedure followed in these studies are then described. Finally, the numerical results of the constraint handling technique comparison and that of the multiobjective metaheuristic comparisons (in the context of the MICFMO test suite) are presented and discussed at the close of the chapter.

In Chapter 9, the AMALGAM method, which is a multiobjective hyperheuristic that incorporates multiple sub-algorithms simultaneously, is investigated in terms of its ability to conduct constrained MICFMO in the context of the aforementioned test suite. The chapter opens with a discussion on the general working of the AMALGAM method, after which four variants thereof are considered for comprehensive investigation in a two-stage comparative study. During the first stage, the variants of the AMALGAM method are compared to one another in order to select a preferred variant. Then, during the second stage, that preferred variant is compared with its constituent sub-algorithms. The numerical results of these studies are presented and discussed at the close of the chapter.

The preferred variant of the AMALGAM method identified in Chapter 9 has been implemented in the OSCAR-4 system. In order to demonstrate the practical applicability of the method, it is employed in Chapter 10 to solve a variety of case study problem instances in the context of the SAFARI-1 and HOR research reactors. The numerical results thus obtained are presented in this chapter.

In Chapter 11, a conceptual framework is proposed for an optimisation-based personal decision support system, dedicated to MICFM. The chapter opens with a motivation for the proposal, along with some basic background information relating to decision support systems. Thereafter, the proposed framework is presented and each constituent component of the decision support system is discussed in some detail. Several suggestions are also put forward for populating the components of the system.

Finally, the dissertation closes in Chapter 12 with a summary of the work contained therein, an appraisal of the contributions of the dissertation, and suggestions for related future work.

CHAPTER 2

A brief introduction to nuclear reactor analysis

Contents

In this chapter, the reader is introduced to some of the fundamental concepts and terminology found in nuclear reactor analysis and theory. The necessary background information pertaining to nuclear reactors is provided so as to gain a better understanding of the characteristics of ICFMO and its underlying processes. This information is crucial for the interpretation of objectives, constraints and solutions to the ICFMO problem.

2.1 Introductory concepts

An atom contains a positively-charged *nucleus* composed of protons and neutrons, collectively known as *nucleons*. The nucleus is surrounded by a cloud of negatively-charged electrons [115]. Following standard notation, the number of protons in a nucleus, which is called the *atomic number*, is denoted by Z. Similarly, the total number of nucleons, called the *atomic mass number*, is denoted by A [115]. Finally, the number of neutrons is denoted by N.

The various species of nuclei containing particular numbers of protons and neutrons are referred to as *nuclides* [202]. A specific nuclide is denoted by the symbol ${}^{A}_{Z}\mathrm{X}$, where X is the chemical symbol from the periodic table for the element in question. For example, the notations ${}^{1}_{1}\mathrm{H}$ and ${}^{238}_{92}\mathrm{U}$ refer to hydrogen and uranium nuclides, respectively. Since the chemical symbol already specifies the value of Z, it is often omitted (*i.e.* ${}^{1}\mathrm{H}$ and ${}^{238}\mathrm{U}$ are written instead of ${}^{1}_{1}\mathrm{H}$ and

$^{238}_{92}$U, respectively). Atoms whose nuclei consist of the same atomic number but different atomic mass numbers (*i.e.* the same Z but a different A and thus a different number of neutrons N) are called *isotopes*. For example, the naturally occuring isotopes of uranium are ^{234}U, ^{235}U and ^{238}U (corresponding to $Z = 92$ as well as $N = 142, 143$ and 146, respectively).

There are essentially two types of nuclear reactions that are important in the study of nuclear reactors [42]. The first is spontaneous disintegrations of nuclei whereas the second is reactions that are due to collisions between nuclei and/or atomic particles. These two types of reactions are further elaborated upon in the remainder of this section.

2.1.1 Radioactive decay

Certain nuclei have too many or too few neutrons for a given number of protons and may spontaneously undergo a transformation into another nucleus [111]. Such a transformation is usually accompanied by energetic particle emissions. These nuclei are referred to as being unstable, and the spontaneous nuclear transformation is called *radioactive decay*.

The decay process of any nucleus is governed by a single fundamental law based on experimental observation. According to this law, the probability that a nucleus will decay within a given time period is essentially a constant, independent of time (*i.e.* the age of the nucleus) and only dependent on the type of nucleus itself [42]. Consider a sample of a radioactive material that decays. The total number of transformations/disintegrations occurring within the sample during a unit time is called the *activity* of the sample. Activity is usually measured in units of *curies*, where one curie is defined as 3.7×10^{10} disintegrations per second. The activity of a radioactive sample decreases exponentially over time. The period of time during which the activity of a sample is reduced by a factor 2 is called the *half-life* of the sample.

2.1.2 Nuclear collision reactions

The focus of this section is on collisions between neutrons and nuclei. Since neutrons are neutrally charged particles, they are neither affected by the positively charged nucleus of an atom nor the negatively charged electrons surrounding the nucleus. Neutrons therefore pass through the electron cloud of an atom and (potentially) interact directly with the nucleus. The interactions include, but are not limited to, the following:

Radiative capture. In this type of reaction, a neutron is captured/absorbed by the nucleus during a collision and a new nucleus of atomic mass number $A+1$ is formed. The reaction is also accompanied by the emission of *gamma rays*, which are high-energy photons.

Scattering. A neutron simply scatters off from the nucleus during a collision in this type of reaction. When the energy state of the nucleus remains the same after the collision, the reaction is referred to as *elastic* scattering. If, however, the energy state of nucleus is excited, the reaction is referred to as *inelastic* scattering. A gamma ray will then be emitted by such a nucleus.

Nuclear fission. As mentioned in Chapter 1, when a neutron interacts with the nucleus of certain nuclides, it may cause the nucleus to split apart into lighter nuclei, thus undergoing fission. These fission products/fragments are also accompanied by the release of several neutrons, typically two or three, and a significant amount of energy (approximately 200 MeV). The additional neutrons that are released may then potentially induce more fission reactions, thus leading to a chain reaction. The energy released during such chain reactions may be utilised for practical applications.

2.1.3 Nuclear cross-sections

The probability of a neutron interacting with some nucleus is characterised by a quantity known as a *cross-section*. The quantity is typically defined by the following experiment[1]:

Consider a beam of neutrons travelling at the same speed (*i.e.* with the same energy) in the same direction. Suppose this beam strikes a very thin target of material (assumed to be sufficiently thin such that no nuclei shield other nuclei from the neutron beam) uniformly. Then, a certain number of neutrons will pass straight through the target of material without interacting with the nuclei, whereas some neutrons will interact.

Let I be the beam intensity, defined as the number of neutrons per unit area per unit time striking the target. Furthermore, let N_A be the number of nuclei per unit area in the target of material. Then, the rate R at which neutrons interact with the target is proportional to I and N_A. This rate, along with the corresponding units of measurement, may be written as

$$\underset{\left[\frac{\#}{\mathrm{cm}^2\cdot\mathrm{s}}\right]}{R} = \underset{[\mathrm{cm}^2]}{\sigma} \underset{\left[\frac{\#}{\mathrm{cm}^2\cdot\mathrm{s}}\right]}{I} \underset{\left[\frac{\#}{\mathrm{cm}^2}\right]}{N_A},$$

where σ is the proportionality constant, and is referred to as the cross-section. It is also known as the *microscopic cross-section*. Each of reactions described in §2.1.2, as well as other nuclear reactions not specified there, is associated with its own characteristic cross-section. Therefore, fission is characterised by a fission cross-section σ_f, scattering by a scattering cross-section σ_s, and so forth. There is a strong dependence between these cross-sections on the incident neutron energy [202].

Let N_V denote the number density of a substance (*i.e.* the number of atoms or molecules contained within a unit volume of some substance). The product between a number density and a cross-section is referred to as a *macroscopic cross-section*, and is denoted by Σ (in units of cm^{-1}) such that $\Sigma = N_V\sigma$. As stated in Duderstadt and Hamilton [42], "*it is natural to interpret* Σ *as the probability per unit path length travelled that the neutron will undergo a reaction with a nucleus in the sample.*" The macroscopic cross-section occurs frequently in nuclear reactor physics and is used, for example, in the neutron transport equation, which is described later in this chapter.

2.2 Nuclear fission and its chain reaction

The nuclear fission reaction may be viewed as the absorption of a neutron by a nucleus, resulting in a compound nucleus having an increased energy state. When this increased energy is sufficient to split the nucleus apart, fission is induced. In certain heavy nuclei, such as ^{233}U, ^{235}U, ^{239}Pu and ^{241}Pu, the incident neutron may even have zero kinetic energy and induce fission [111]. Such nuclides are called *fissile* nuclides and they represent the primary fuel used in nuclear reactors because they can sustain a fission chain reaction. For most heavy nuclides, the increased energy state following the absorption of a neutron is insufficient to induce fission. If, however, the neutron itself has sufficient kinetic energy before it is absorbed, certain nuclei may be induced to fission. Nuclides like ^{232}Th, ^{238}U and ^{240}Pu, which can be fissioned by such energetic neutrons, are called *fissionable* nuclides [111]. Fissionable nuclides cannot sustain a stable fission chain reaction alone, and must be used in conjunction with fissile nuclides to act as nuclear fuel.

[1] This description to follow has been reproduced from Duderstadt and Hamilton [42].

The only fissile nuclide found in nature is ^{235}U and it only occurs as approximately 0.711% of natural uranium, which is primarily composed of ^{238}U [42]. Although certain nuclear reactors can operate using natural uranium as fuel, the majority of present-day reactors are fueled with *enriched* uranium. This means that the percentage of ^{235}U has been increased (by artificial means) above its natural value. If the enriched uranium contains less than 20% of ^{235}U it is classified as *low enriched uranium* (LEU), whereas *highly enriched uranium* (HEU) contains more than 20% of ^{235}U. Fissile nuclides can also be obtained from so-called *fertile* nuclides which, after the absorbtion of a neutron, are transformed through a series of radioactive disintegrations (*i.e.* decays) into a fissile nuclide [42].

The fission chain reactions maintained in nuclear reactors are made possible by the release of neutrons during a fission reaction, thus allowing these *fission neutrons* to induce more fission reactions. A simple diagrammatical representation of a fission chain reaction is presented in Figure 2.1.

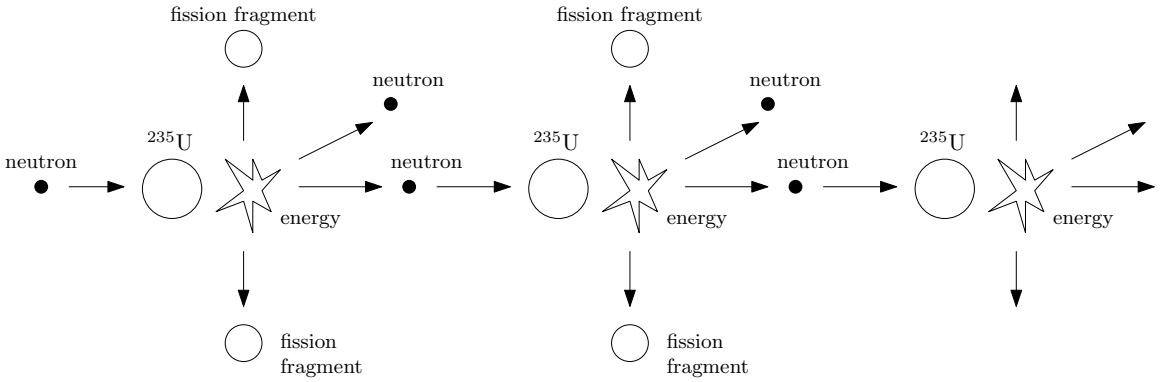


FIGURE 2.1: *A simple diagrammatical representation of a fission chain reaction.*

In order to maintain a stable fission chain reaction, a nuclear reactor has to be configured so that exactly one neutron emitted during each fission will induce another fission reaction. The remaining neutrons emitted during fission should either be absorbed in capture reactions, or should simply be allowed to leak out of the system. A nuclear chain reaction can be characterised quantitatively in terms of its *multiplication factor*, denoted by k. This factor is defined as the ratio of the number of fission neutrons in one generation to the number of fission neutrons in the preceding generation [42].

According to the above definition, if $k = 1$, the number of neutrons in any two consecutive fission generations remains constant and the chain reaction will therefore be time-independent. Such a system, in which the chain reaction is characterised by $k = 1$, is said to be *critical*. If, however, $k > 1$, the number of neutrons increases from generation to generation, thus leading to a chain reaction that grows without bound. In this case, the system is said to be *supercritical*. Similarly, if $k < 1$, the number of neutrons decreases in each successive generation and the chain reaction will ultimately die out. Such a system is said to be *subcritical*.

In a nuclear reactor, the value of k can be adjusted in order to control the chain reaction. Suppose, for example, the neutron population in the reactor has to be increased (*i.e.* the reactor power has to be increased). The value of k can temporarily be increased to above unity, thus making the reactor supercritical. When the desired neutron population/power level has been achieved, the value of k may be adjusted back to unity so that the reactor becomes critical again, at the new population/power level. This adjustment of the value of k is known as *nuclear reactor control* and is typically achieved by inserting neutron absorbing material into or removing such material from the reactor.

A quantity closely related to the multiplication factor is the core *reactivity* which is denoted by ρ and defined as

$$\rho = \frac{k-1}{k}.$$

Essentially, reactivity measures the deviation of the core multiplication from unity (*i.e.* from its critical value).

2.3 Basic components of a nuclear reactor core

As mentioned in §2.1.3, cross-sections depend on the incident neutron energy. In a nuclear reactor core, the energies of neutrons may vary from as much as 10 MeV (typically the maximum energy of a fission neutron) down to 0.001 eV after having been subjected to several scattering collisions with nuclei [115]. It is known that the fission cross-section σ_f is typically largest at low energies. As such, a fission chain reaction is easier to maintain using "slow" low-energy neutrons as opposed to "fast" energetic neutrons. These low-energy neutrons are also known as *thermal neutrons*, because they are in thermal equilibrium with the surrounding material. Nuclear reactors that primarily employ thermal neutrons to induce fission are referred to as *thermal reactors*.

In general, the basic components of a thermal nuclear reactor core are: fuel assemblies, control rods, a moderator, coolant and a reflector [111]. These components are usually located within a *reactor vessel* made of a steel alloy, for example, although some reactor designs do not include a vessel. In order to maintain the core geometry, some of these components are placed on a support structure such as a grid plate. As an example, the top view of a VVER-1000 nuclear reactor core is shown in Figure 2.2. The figure shows several fuel assemblies in the middle, the grid plate at the bottom and the reflector on the periphery, all within the reactor vessel.

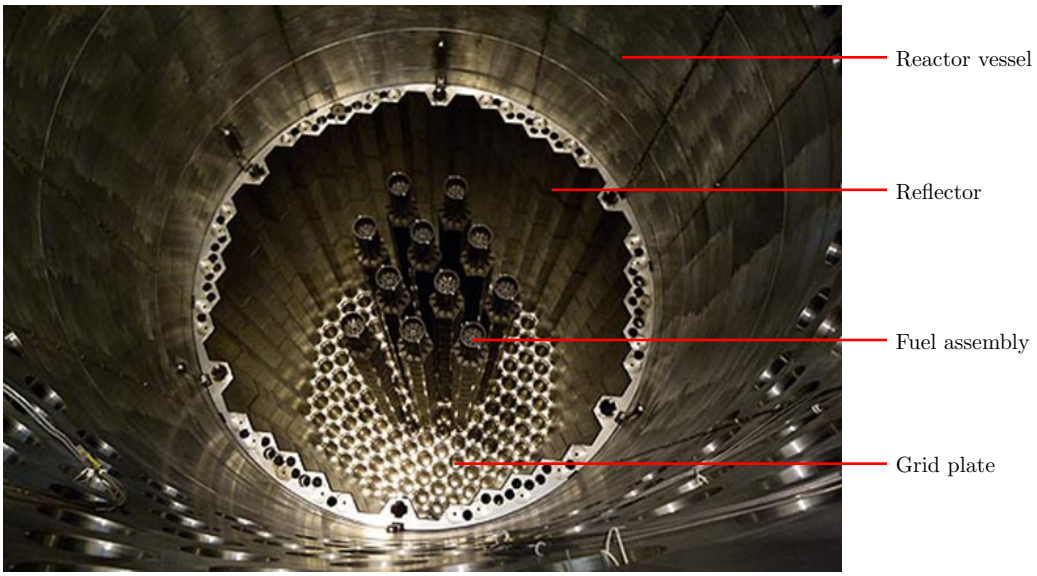


FIGURE 2.2: *Top view of a VVER-1000 nuclear reactor core [150].*

A *fuel assembly* contains the fissile/fissionable/fertile material and is typically composed of smaller, sealed units of fuel (such as cylindrical tubes or plates) that are bundled together. *Control rods* are usually movable pieces of neutron absorbing material such as boron, cadmium or gadolinium. As mentioned earlier, these rods are used to control the chain reaction (*i.e.* the

value of k) in a reactor. Since high-energy neutrons are emitted during fission, they have to be slowed down (or moderated) to the thermal energy range by means of scattering collisions. A *moderator* is introduced into a thermal reactor for this purpose, and is typically a material of low atomic mass number such as light water (H_2O), heavy water (D_2O), graphite or beryllium [111]. Given that a large amount of heat is also generated during fission, a reactor has to be cooled continuously. Accordingly, a *coolant* is circulated through the reactor to remove this heat. Examples of liquid coolant are water or sodium, while helium or carbon dioxide may be employed as gaseous coolants. A *reflector* may finally be placed on the periphery of a core in order to reflect (or scatter) neutrons back towards the core, instead of allowing them to leak out and being lost to the system. Note that the same material may fulfill more than one purpose in a reactor. Water may, for example, be used as both moderator and coolant, while beryllium may be used as reflector and an additional moderator.

It should always be remembered that a nuclear reactor is a significant source of radiation. Therefore, a reactor core will always have shielding components and/or material in order to protect operating personnel, as well as other reactor components, from this harmful radiation.

2.4 Neutron transport

According to Duderstadt and Hamilton [42], the central problem of nuclear reactor theory is to determine the distribution of neutrons in a given reactor. The reason for this is simply that the rate at which various nuclear reactions occur inside a nuclear reactor is dictated by the neutron distribution.

In a nuclear reactor, the distribution of neutrons is governed by the process of *neutron transport*, *i.e.* the motion of neutrons and their interaction with matter [202]. This movement of any given neutron is rather complex since it undergoes repeated collisions with other nuclei, before being absorbed or leaking out of the reactor. In many reactor studies, however, neutron motion is approximated by modelling it as a *diffusion* process (*i.e.* it is assumed that neutrons diffuse from regions of high neutron concentration to regions of low neutron concentration). While the derivation and analysis of the characteristic equations associated with neutron transport and diffusion fall beyond the scope of this dissertation, a brief overview of these equations is nevertheless presented here for the sake of completeness[2].

The behaviour or state of a neutron is characterised by its position $\boldsymbol{r}$, energy E, direction of motion $\boldsymbol{\Omega}$ and the observed time t. A population of neutrons may thus be represented by a distribution, called the *angular neutron density*, which is denoted by $n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$. Accordingly, the expected number of neutrons at time t, in a volume $\mathrm{d}^3\boldsymbol{r}$ about $\boldsymbol{r}$, having energy $\mathrm{d}E$ about E, and moving in a solid angle $\mathrm{d}^2\boldsymbol{\Omega}$ about $\boldsymbol{\Omega}$ is given by $n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)\,\mathrm{d}^3\boldsymbol{r}\,\mathrm{d}E\,\mathrm{d}^2\boldsymbol{\Omega}$.

The *angular neutron flux*, denoted by $\phi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$, is defined as the product of the neutron speed v_n and the angular neutron density. Therefore,

$$\phi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = v_n n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t). \tag{2.1}$$

The maximum information about the population of neutrons is given by this angular flux [73]. Integrating over all angles of $\boldsymbol{\Omega}$ in (2.1) yields the *scalar neutron flux*, denoted by $\phi(\boldsymbol{r}, E, t)$. If steady-state (*i.e.* time-independent) conditions are assumed, then the dependence on t in each of the aforementioned flux quantities simply drops away.

[2] This overview is sourced from Duderstadt and Hamilton [42], Lamarsh and Baratta [111], Stacey [202], and Hébert [73].

There are several different forms in which the neutron transport equation can be expressed. The integro-differential form of the steady-state neutron transport equation is given by

$$\overbrace{\boldsymbol{\Omega}\cdot\nabla\phi(\boldsymbol{r},E,\boldsymbol{\Omega})}^{\text{nett leakage}}+\overbrace{\Sigma_t(\boldsymbol{r},E)\phi(\boldsymbol{r},E,\boldsymbol{\Omega})}^{\text{collision loss}} = \underbrace{\int_0^\infty \mathrm{d}E'\int_{4\pi}\Sigma_s(\boldsymbol{r},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},E'\to E)\phi(\boldsymbol{r},E',\boldsymbol{\Omega}')\,\mathrm{d}\boldsymbol{\Omega}'}_{\text{inscattering}}+\underbrace{S(\boldsymbol{r},E,\boldsymbol{\Omega})}_{\text{source term}}, \quad (2.2)$$

where Σ_t and Σ_s are the macroscopic total and scattering cross-sections, respectively, and $S(\boldsymbol{r},E,\boldsymbol{\Omega})$ is a source term. The unknown variable to solve for is therefore the angular neutron flux. Appropriate boundary and initial conditions for (2.2) complete the specification of a problem instance.

The derivation of the neutron transport equation is based on the principle of neutron conservation (*i.e.* balancing the gain and loss of neutrons in an arbitrary region $\mathrm{d}^3\boldsymbol{r}\,\mathrm{d}E\,\mathrm{d}^2\boldsymbol{\Omega}$ of so-called *phase space* $(\boldsymbol{r},E,\boldsymbol{\Omega})$ in some system). Such a region of phase space can gain neutrons by means of any neutron source in the region. It can also gain neutrons when they flow into the region through the surface of $\mathrm{d}^3\boldsymbol{r}$. Finally, when neutrons, having a different energy E' and/or direction of motion $\boldsymbol{\Omega}'$, undergo a scattering collision in $\mathrm{d}^3\boldsymbol{r}$ such that $E'\to E$ and/or $\boldsymbol{\Omega}'\to\boldsymbol{\Omega}$, then the region also gains neutrons. This process is known as *inscattering*. Similarly, neutrons can be lost to the region of phase space when they leak out of it through the surface of $\mathrm{d}^3\boldsymbol{r}$, as well as when they undergo collision reactions. A collision either leads to the absorption of a neutron, or it results in a scattering collision, in which case $E\to E'$ and/or $\boldsymbol{\Omega}\to\boldsymbol{\Omega}'$, and the neutron is lost to the region (*i.e.* the opposite of inscattering).

Accordingly, the rate of change of the number of neutrons in a region of phase space is equal to the rate at which neutrons enter the region from a source, together with the rate at which neutrons flow into the region through the surface of the volume, less the rate at which neutrons leak out of the region, together with the rate at which neutrons enter the region by means of inscattering, less the rate at which neutrons are absorbed or scattered into energies or directions not part of the region of phase space. This neutron balance relation is captured in (2.2).

In the diffusion approximation of neutron transport, *Fick's law* is employed during its derivation. It is therefore assumed that the movement of neutrons is similar to the movement of gas particles. According to Fick's law, then, neutrons will "flow" from a region of high neutron concentration to a region of low concentration at a rate which is proportional to the gradient of the concentration. There exists a rigorous mathematical derivation of the neutron diffusion model that starts from the transport equation, as well as a more "heuristic physical derivation." The reader is referred to [42] for details on these derivations and only a final result is provided here.

The integro-differential form of the steady-state, energy-dependent, neutron diffusion equation is given by

$$-\nabla\cdot\big(D(\boldsymbol{r},E)\nabla\phi(\boldsymbol{r},E)\big)+\Sigma_t(\boldsymbol{r},E)\phi(\boldsymbol{r},E)=\int_0^\infty\Sigma_s(\boldsymbol{r},E'\to E)\phi(\boldsymbol{r},E')\,\mathrm{d}E'+S(\boldsymbol{r},E), \quad (2.3)$$

with all notation as previously defined. Notice that there is no angular dependency in the diffusion equation. The unknown variable to solve for is therefore the scalar neutron flux. As before, appropriate boundary and initial conditions for (2.3) complete the specification of a problem instance.

Although details have not been provided above in respect of the source terms in (2.2) and (2.3), it is important to consider the following. It is typically assumed that a neutron source density

consists only of neutrons produced by fission. Then, if the neutron leakage and absorption rate is not equal to the production rate of fission neutrons, the steady-state condition is lost. In order to maintain balance, the fission source term is divided by an unknown constant called the *effective multiplication factor*, which is denoted by k_{eff}. Accordingly, solution of the transport/diffusion equation becomes an eigenvalue problem, with k_{eff} as the eigenvalue and ϕ as the eigenvector.

2.5 Nuclear reactor analysis

The study of a nuclear reactor involves not only the neutronic analysis of the core, but also the thermal-hydraulic analysis of cooling the core, the mechanical/structural analysis of core components, and so forth. This adds further computational complexity to the modelling of a nuclear reactor, and inclusion of these aspects in modelling approaches is often referred to as *multiphysics* approaches. The scope of the dissertation is, however, limited to the (steady-state) neutronic evaluation of any given core, as is typically the case in the ICFMO literature [219]. Accordingly, this section contains only a discription of the relevant aspects within neutronics.

Duderstadt and Hamilton [42] classify the primary neutronic aspects of interest into three categories. The first category is concerned with the calculation of the core multiplication (k_{eff}) and neutron flux distribution (ϕ), *i.e.* the solution of the transport or diffusion equation. Then, the neutron flux is used to calculate the power distribution throughout the core. A very important parameter of interest is the maximum peak-to-average power ratio in the core, often referred to as the core *power peaking factor*. It is a safety parameter used to determine whether the thermal limitations of the core are adhered to. As such, the parameter is closely linked to the thermal-hydraulic analysis of a reactor core.

The second category is concerned with core reactivity and control analysis. In order to allow a nuclear reactor to operate at power for some extended period of time, the core is initially loaded with a surplus of fuel well in excess of what would be required simply to achieve criticality. Mechanisms such as neutron absorbing control rods (see §2.3) are then required to compensate for the excess reactivity contained in the initial fuel loading, in addition to controlling the fission chain reaction during the entire period of reactor operation (maintaining criticality, adjusting power levels, *etc.*). There are several important parameters of interest in this category, and a subset of these are:

Excess reactivity, denoted by ρ_{ex}, is the core reactivity present when all the control rods are fully withdrawn from the core. In general, larger values of ρ_{ex} imply longer periods of time during which the reactor may operate.

Shutdown margin, denoted by ρ_{sdm}, is a safety parameter corresponding to the amount of reactivity by which a reactor is subcritical when all the control rods have been inserted fully into the core. Usually, a limit for the shutdown margin is chosen such that the core must be subcritical even when the most-reactive control rod is fully extracted from the core (the so-called "stuck-rod criterion").

Control bank worth (or *total control rod worth*), denoted by ρ_{cbw}, is the difference between the core reactivity present when all the control rods are fully withdrawn from the core, and when all the control rods are fully inserted into the core. Therefore, $\rho_{\text{cbw}} = \rho_{\text{ex}} + \rho_{\text{sdm}}$ and this parameter measures the total negative reactivity that can be inserted into the core by all the control rods.

The third and last category is concerned with core composition changes and depletion analysis. During reactor operation, the number densities of various isotopes within the core are continually changing as a result of the nuclear reactions taking place. For example, fission reactions reduce (deplete) the amount of fuel (*i.e.* the fissile/fissionable nuclide concentration) in the core while also producing large quantities of fission products, which may be neutron absorbers. Neutron capture, on the other hand, reduces the concentration of fertile nuclides while also increasing the concentration of fissile nuclides (see §2.2). These composition changes affect the rate at which nuclear reactions occur in the core. Accordingly, monitoring the isotopic composition in the core during reactor operation is very important, because the production and/or depletion of nuclides affect the neutron flux distribution, as well as the multiplication factor.

Such an analysis of isotopic composition changes is, however, complicated because the temporal and spatial variation in isotopic composition depend on the neutron flux distribution — which itself depends on the isotopic composition in the core. A standard approach may be taken in which this time-dependent phenomenon is considered as a sequence of static criticality calculations (*i.e.* steady-state neutron transport/diffusion calculations) for each different core composition state [42]. This is made possible because of the relatively slow manner in which core composition changes occur (of the order of hours or days). In order to determine the new isotopic number densities at each of these static points in time, isotopic depletion equations are solved for each isotope that forms part of the analysis [217]. It is generally assumed that neutron flux in these differential equations is known. Furthermore, in many cases, the equations for different isotopes are coupled and therefore have to be solved accordingly.

2.6 Reactor core calculation code systems

Due to the complexities associated with nuclear reactor analysis, particularly for modern and practical reactor applications, it is required that the various mathematical theories describing the nuclear behaviour in a reactor core be implemented within computer codes. These codes are employed as computational aid tools for the modelling of a reactor core and the numerical solution of the various governing equations. An overview of a basic computational model of such a code system is presented in Figure 2.3 and is loosely based on the one found in [42].

Arguably the most important part of a reactor core calculation code system (also referred to as a reactor core simulator) is its neutronics solver. This part of the code system numerically solves the appropriate set of transport/diffusion equations so as to yield the effective multiplication factor and neutron flux throughout the core. It should be mentioned that the heterogeneity of a nuclear reactor (as a result of the different core components having various material compositions) adds to the difficulty in solving the relevant equations, because the computational reactor model becomes extremely intricate.

The aforementioned equations may be solved by deterministic techniques (which employ various numerical discretisation schemes for the variables) or stochastic techniques (which typically employ Monte Carlo simulation methods). Deterministic approaches are generally much faster than stochastic approaches and, as such, they are employed in the majority of code systems capable of three-dimensional, full-core reactor analysis calculations. In particular, so-called *nodal diffusion* methods are often employed [219].

During the early years of nuclear reactor analysis, very simplified mathematical models were employed in calculations due to the limited capacity of computers at that time. As the computing capability improved over the years, however, more realistic (and thus more complicated) calculations became possible. Therefore, there is always a trade-off between the fidelity of a reactor model, and the computational resources required/available to calculate its solution.

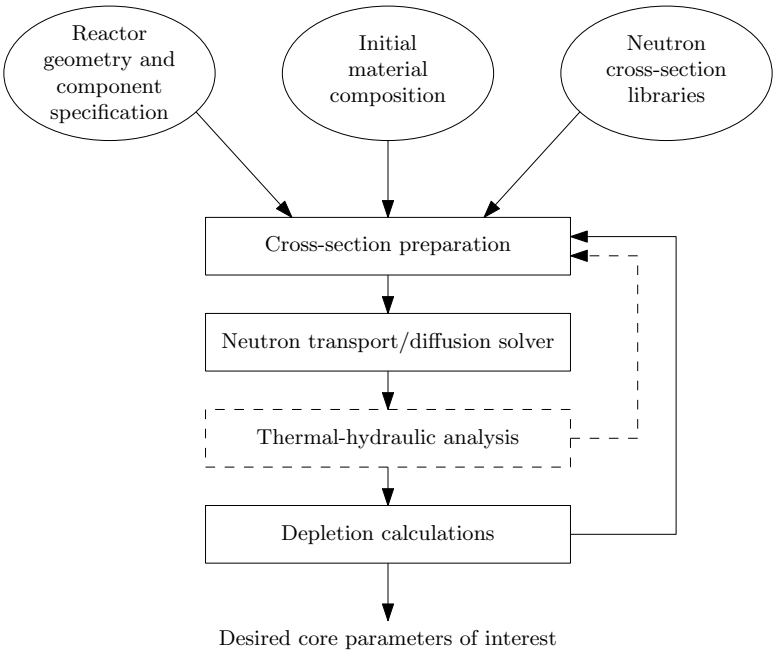


FIGURE 2.3: *A basic computational model of a reactor code system.*

2.7 Chapter summary

In this chapter, the reader was introduced to some of the fundamental concepts and terminology in nuclear reactor analysis and theory. The most important nuclear reactions in the context of reactor physics were described in §2.1. This was followed by a more detailed discussion on nuclear fission and its chain reaction (characterised by the multiplication factor) in §2.2. A brief overview of the basic components of a nuclear reactor core was also presented in §2.3.

In §2.4 the process of neutron transport, which governs the distribution of neutrons in a reactor, was discussed along with its diffusion approximation. In particular, the reader was introduced to the concept of (angular) neutron flux — the unknown variable to solve for in the neutron transport/diffusion equation. A description of the primary neutronic aspects of interest within nuclear reactor analysis was presented in §2.5. It comprises three categories, namely the calculation of the core multiplication and neutron flux distribution, core reactivity and control analysis, and depletion analysis. Finally, in 2.6 the necessity of reactor core calculation code systems was discussed, noting that there is a trade-off between the fidelity of a reactor model, and the computational resources required to solve it.

CHAPTER 3

Literature survey

Contents

This chapter contains a survey of relevant literature pertaining to ICFMO that falls within the scope of this dissertation. Background information on the early years of ICFMO research is presented first, setting the stage for an overview of the most popular objective functions and constraints adopted in model formulations for the ICFMO problem. Thereafter, a brief description of typical solution techniques that have been employed in the literature to solve the problem is presented (in the context of both single-objective and multiobjective optimisation). Since function evaluations during ICFMO are, in general, computationally expensive, a number of approaches adopted in the literature for reducing this computational burden are finally touched upon at the end of the chapter.

3.1 General modelling considerations and background

As with any optimisation problem, the components of a mathematical model for the ICFMO problem include one or more objective functions and a suite of constraints in terms of decision variables. In general, objective functions and most constraints associated with the ICFMO problem are not available in closed form, *i.e.* they cannot be expressed as elementary mathematical

functions of the decision variables. This is because of the complicated modelling requirements for nuclear reactors, as described in Chapter 2. The majority of modern approaches toward ICFMO, therefore, involve the evaluation of objective functions and the verification of constraint satisfaction in terms of reactor core parameters returned by a core simulator. In doing so, an optimisation technique may be coupled with a core simulator in order to solve an instance of the ICFMO problem. As noted in §2.6, the fidelity of the reactor model within a core simulator comes at a trade-off to the computational resources required during calculations. The quality of any solution to an ICFMO problem instance computed by the above-mentioned coupling therefore depends on the fidelity of the core simulator.

During the early years of ICFMO research, however, there was a heavy reliance on simplified reactor models and idealised problems due to the limited capacity of computers, poor fidelity and inefficient reactor core simulators, and relatively few available optimisation methods at that time [218]. Accordingly, fuel management optimisation problems were formulated as linear programming problems [28, 71, 210] in order for them to be solved using traditional linear programming techniques (*e.g.* the simplex algorithm[1] or the branch-and-bound method[2]). In some cases, nonlinear programming model formulations were also put forward which could be solved by the method of successive linear programming[3] [43, 206]. Dynamic programming was also applied during those early years [227].

Apart from the above-mentioned solution approaches, there was also a heavy reliance on *heuristics* at that time and these were largely based on the expert knowledge/experience of reload design engineers [218]. According to Burke and Kendall [24], the Oxford Dictionary of Computing [154] defines a heuristic as "*A 'rule-of-thumb' based on domain knowledge from a particular application, that gives guidance in the solution of a problem.*" The aim of employing heuristics was to restrict the decision space of ICFMO problem instances, thereby reducing the computational complexity associated with solving such instances. Naft and Sesonske [143], for example, designed a direct search[4] scheme in which trial solutions are generated by means of binary exchanges between fuel assemblies. They incorporated heuristic rules (based on symmetry considerations in the core, the age of an assembly and its depletion profile) into the scheme so as to restrict certain assemblies from being exchanged. Similar approaches, in which heuristics restricted certain binary exchanges, were also adopted in [77, 78, 102].

Turinsky and Parks [219] claimed in 1999 that the most widely used optimisation technique for nuclear fuel management, at that time, was the reload design engineer. According to Downar and Sesonske [40], as well as Turinsky [218], the early research on ICFMO failed to meet the needs of practical reload design and, in fact, led to skepticism about the usage of automated tools for optimisation. The measures taken to reduce the computational burden of ICFMO simply introduced too much inaccuracy in the reactor models and subsequent objective functions and constraints. Accordingly, the survey presented in the remainder of this chapter is focussed on more recent literature — specifically publications during the last 25–30 years.

[1] The *simplex algorithm*, developed by George Dantzig in 1947, is generally an efficient method for solving linear programming problems. See [232] for more details.

[2] The *branch-and-bound* method was first proposed by Land and Doig [112] and it is a general technique that may be applied in conjunction with the simplex algorithm to integer programming problems. An advantage of the method is that it is guaranteed to find a globally optimal solution.

[3] In the method of *successive linear programming*, a sequence of linear approximations of solutions to nonlinear programming problems is solved iteratively using traditional linear programming techniques. Each solution obtained in the new approximation is closer to the (locally) optimal solution of the nonlinear program [155].

[4] *Direct search*, proposed by Hooke and Jeeves [84], is a general method for solving optimisation problems. The method is described in [84] as a "*sequential examination of trial solutions involving comparison of each trial solution with the best obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be.*"

3.2 Problem formulations in the literature

In general, the ICFMO problem is a nonlinear assignment problem in which fuel assemblies from an available set have to be assigned in an optimal manner to fixed loading positions within a reactor core. The desirability of such an assignment is determined by the associated objective function(s), subject to the constraints imposed.

Since the scope of this dissertation is restricted to single-cycle ICFMO, the literature presented in this section comprises only formulations within this restricted context.

3.2.1 Objective function formulations

As noted by Turinsky and Parks [219], many different objectives have been adopted for the purpose of ICFMO, and they are often conflicting. This ultimately makes the ICFMO problem multiobjective in nature, which means that a trade-off solution must typically be obtained. It is also important to note that the type of reactor and the purpose of its operation influence the objective(s) pursued during ICFMO. Power reactors are, for example, intended to generate electricity whereas research reactors are generally intended for scientific experiments, and these different applications call for the pursuit of different ICFMO objectives.

The primary ICFMO objective adopted in the literature is the economically-motivated minimisation of the fuel cycle cost [218, 219]. This comes as no surprise given that the everwhelming majority of ICFMO research is orientated towards power reactors, whose aim within a utility is generally to produce electricity at the lowest cost possible. Fuel cycle costs, however, require multicycle calculations due to the fact that any fuel assembly will usually reside in a reactor core for more than one cycle [218]. In order to translate the fuel cycle cost objective into a single-cycle optimisation paradigm, surrogate objectives have typically been formulated instead. Several of these objectives are discussed below.

The most common surrogate objective for fuel cycle cost is the maximisation of the cycle length (or, equivalently, the cycle energy production) of a reactor which is typically measured in effective full-power days[5]. In practice, the reactor is shut down when its planned cycle length is reached. If it could have operated for longer, the amount of fresh fuel loaded into the core, *i.e.* the amount of ^{235}U, could be reduced accordingly, thus saving on fuel costs [109]. This reduction may be achieved by adjusting the feed enrichment[6] or limiting the number of fresh fuel assemblies loaded into the core.

In some cases, the cycle length is calculated explicitly as the objective function [60, 245]. For the most part, however, the objective function is expressed as a reactor core parameter returned by a core simulator. Several parameters have been adopted for this purpose. The most popular choices in the literature are the effective multiplication factor [9, 35, 56, 100, 123, 168, 222, 243] and, applicable only to PWRs, the soluble boron concentration[7] in the reactor [1, 25, 76, 137, 138], with both parameters to be maximised. Other parameters that have been adopted include excess reactivity [74] and core-average burnup over the cycle [238]. *Burnup* is a measure of the amount of energy generated by nuclear fuel. It is typically measured as the energy released per unit mass of initial fuel.

[5] *Effective full-power days* correspond to the number of days during which a reactor can produce power at its full-rated level. See [217] for its calculation.

[6] *Feed enrichment* refers to the percentage of ^{235}U present in fresh (new) fuel assemblies.

[7] In PWRs, boric acid (which is a neutron absorber) is added to the water, which acts as coolant and moderator. By varying its concentration, it may be used as an effective reactivity control measure during reactor operation.

Another common surrogate objective for fuel cycle cost is the maximisation of the discharge burnup, *i.e.* the average burnup of all the fuel assemblies to be discharged from the core at the end of the operational cycle. By maximising this objective, the burnup of the fuel remaining in the core during future cycles is minimised. According to Turinsky [218], this implies that more reactivity is carried forward into the next cycle, which means that the amount of fresh fuel loaded into the core may be reduced. As before, a saving on fuel costs may then be achieved. In general, core simulators are capable of calculating the discharge burnup, which has therefore been used directly as the objective function without the need of a proxy [40, 109, 194, 218, 219]. It has been argued that, in the context of single-cycle optimisation, the objective of maximising discharge burnup is preferable to the objective of maximising cycle length [218, 219]. The argument is paraphrased here as follows. Maximisation of cycle length tends to yield reload configurations in which fresh fuel assemblies are assigned to inner positions within the core, while the older assemblies are assigned to outer positions. This causes the fresh assemblies to operate at higher powers, thus attaining higher burnup and potentially leading to future cycles requiring a larger amount of fresh fuel. Maximisation of discharge burnup, however, involves a combination of extending the cycle length and shifting the core power to fuel assemblies that are to be discharged. Accordingly, it tends to yield reload configurations with "*less aggressive burning of the fresh fuel assemblies*" [219].

Secondary objectives employed in the literature primarily involve the operational safety of a reactor or, as Turinsky [218] calls it, the minimisation of risk. It should be noted, however, that the safety of a reactor is generally taken into account by the imposition of constraints. That being said, the objectives are typically concerned with maximising thermal margins from their limits (*e.g.* to ensure that fuel assemblies are adequately cooled and their structural integrity is maintained as effectively as possible). One such objective frequently adopted is the minimisation of the power peaking factor of a reactor [16, 40, 57, 62, 83]. It is recognised that minimising the power peaking factor will also lead to a flattening of the power distribution in the core, and *vice versa*. Accordingly, another objective adopted in the literature is to flatten the core power distribution and this is typically achieved by minimising the sum of squares of the difference between the normalised/relative power in each fuel position and unity [98, 165].

In research reactors, alternative objectives to the economically-motivated and safety-related objectives described above may be pursued during ICFMO. A common thread between the different utilisation purposes of a research reactor is that enhanced neutron flux levels (in some specific energy range) are sought in various locations throughout the reactor core, *e.g.* in irradiation positions or at beam tubes [180]. A typical objective employed in the literature is therefore to maximise the (thermal) neutron flux in irradiation positions within the core [74, 75, 128, 192]. Another objective also considered is the maximisation of neutron leakage from the core at beam tube locations (according to some leakage measure, usually not specified by authors) [222].

The above-mentioned objectives are considered predominantly within the context of single-objective optimisation in the ICFMO literature. There are, however several instances in which the objectives are combined to form a multiobjective formulation for the ICFMO problem. Examples of objective combinations are: cycle length and peaking factor [9, 123, 156], discharge burnup and peaking factor [95], and cycle length and irradiation position neutron flux [75]. A more detailed discussion on multiobjective formulations of the ICFMO problem is presented later in this dissertation.

3.2.2 Constraint formulations

There are many requirements that have to be met when designing a reload configuration for a nuclear reactor core. During ICFMO, these requirements may be imposed as constraints and, as

mentioned above, most of these are related to safety. There are, of course, also several physical constraints that are typically satisfied implicitly, such as assigning one fuel assembly to one loading position, and maintaining symmetry when it has been assumed.

The overwhelming majority of constraints adopted in the literature, however, involve power peaking thermal limits in the reactor core [1, 35, 56, 100, 121, 131, 137, 238, 245]. These thermal limits typically ensure that fuel assemblies can be cooled adequately and that their structural integrity can be maintained. Other important constraints include ensuring sufficient excess reactivity in the core so as to meet the cycle energy requirement (*i.e.* cycle length) [121], adhering to the maximum burnup of a fuel assembly [238], and ensuring there is sufficient shutdown margin (*i.e.* reactivity control) [74]. All these constraints, however, by no means constitute the full spectrum of those employed in the literature. A more comprehensive listing of other typical constraints may be found in [218, 219, 223].

3.2.3 The decision space

Since the ICFMO problem may be considered as a nonlinear assignment problem, its decision space comprises a finite set of combinatorial solutions. The cardinality of this set (*i.e.* the size of the decision space) depends on the number of available fuel assemblies, denoted here by m, as well as the number of loading positions in the reactor core, denoted by n, with $m \geq n$. A solution to the ICFMO problem may then be represented by a partial permutation[8] decision vector $\boldsymbol{x} = [x_1, \ldots, x_n]$ where $x_i = j$ denotes that fuel assembly $j \in \{1, \ldots, m\}$ is assigned to loading position $i \in \{1, \ldots, n\}$. The size of the ICFMO problem decision space is therefore the number of distinct partial permutations formed when using n fuel assemblies from the total set of m assemblies, namely $m!/(m-n)!$.

Consider, for example, a typical Westinghouse 4-loop PWR core. As illustrated in Figure 3.1, such a core consists of 193 fuel loading positions. If there are exactly 193 fuel assemblies available

[8] A *partial permutation*, also known as a *sequence without repetition*, or a *k-permutation of n*, is an ordered arrangement of a subset of k elements selected from a set of n elements.

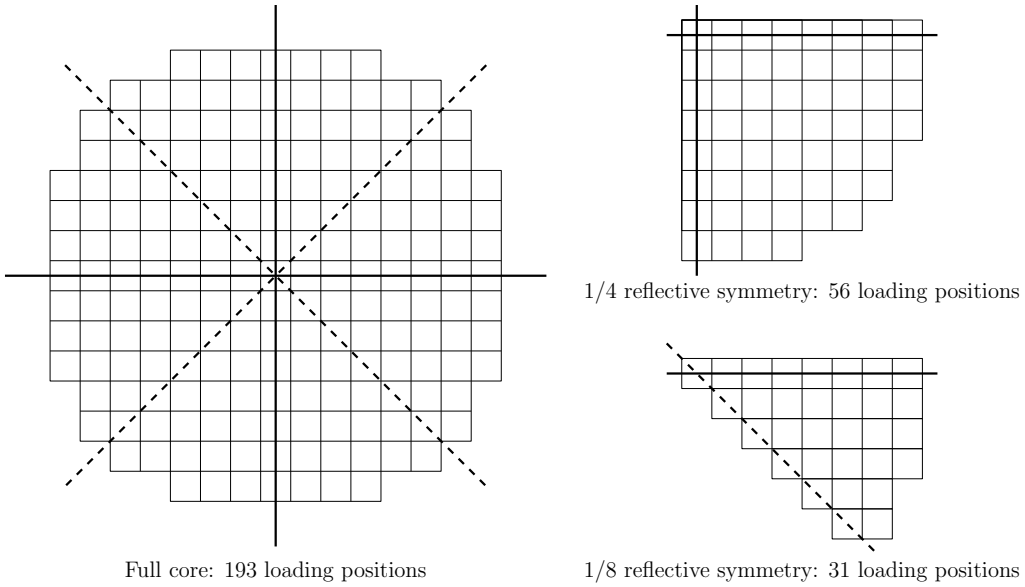


FIGURE 3.1: *Top view of the core layout of a Westinghouse 4-loop PWR with supporting illustrations of its* 1/4 *sector and* 1/8 *sector reflective symmetries.*

for ICFMO, then the size of this decision space is $193! \approx 6.85 \times 10^{358}$. This immense size may, however, be reduced significantly if it is assumed that the core will be loaded with fuel assemblies according to a symmetrical design. By assuming a 1/4 sector reflective symmetry, the number of loading positions in the problem instance is reduced to 56, as shown in Figure 3.1. If there are also exactly 56 fuel assemblies available for ICFMO in this problem instance, the size of the decision space becomes $56! \approx 7.11 \times 10^{74}$, which is a significant reduction compared to the full core instance. Similarly, by assuming a 1/8 sector reflective symmetry, the number of loading positions in the problem instance is reduced even further to 31 (see Figure 3.1). With 31 fuel assemblies available for ICFMO, the size of the decision space is reduced to $31! \approx 8.22 \times 10^{33}$.

As a second example, consider a typical VVER-1000 PWR core which consists of 163 hexagonal fuel loading positions, as shown in Figure 3.2. If there are exactly 163 fuel assemblies available for ICFMO, then the size of this decision space is $163! \approx 2.00 \times 10^{291}$. By assuming a 30° sector rotational symmetry, however, the number of loading positions in the problem instance is reduced to 28, as shown in Figure 3.2. With exactly 28 fuel assemblies available for ICFMO, the size of the decision space becomes only $28! \approx 3.05 \times 10^{29}$.

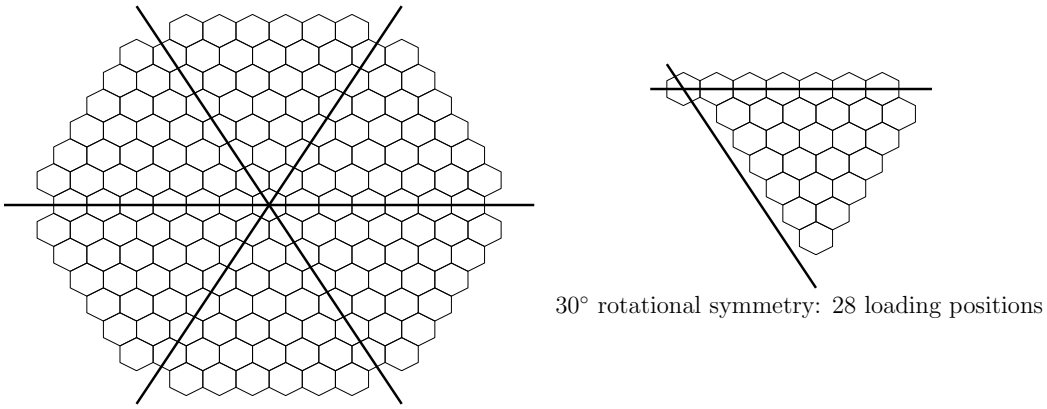


FIGURE 3.2: *Top view of the core layout of a VVER-1000 PWR with a supporting illustration of its 30° rotational symmetry.*

Unlike for typical power reactors, whose core geometries easily lend themselves to symmetry, it is not always possible to assume any form of symmetrical fuel loading in research reactors. Consider, for example, the SAFARI-1 materials testing reactor. Its core consists of 26 fuel loading positions in an asymmetric geometry, as illustrated in Figure 3.3. Clearly, it is not possible to assume any symmetry and therefore the full core has to be taken into account during ICFMO. If there are exactly 26 fuel assemblies available for ICFMO, then the size of this decision space is $26! \approx 4.03 \times 10^{26}$.

It is also worth mentioning that asymmetry in a research reactor is not restricted to the fuel loading positions in the core. Very often, other positions are filled with various (and different) materials having different physical properties affecting the neutron flux in the core. Similarly, reflector placement also has a significant effect on the flux distribution in the core which may lead to asymmetry in the problem. These factors are generally not found in power reactors.

Based on the examples given above, it is clear that the decision space of a typical ICFMO problem instance is much too large to consider solving the optimisation problem by total enumeration (*i.e.* an exhaustive search).

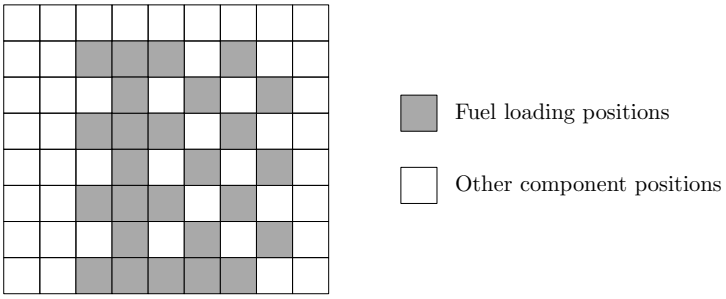


FIGURE 3.3: *Top view of the core layout of the SAFARI-1 materials testing reactor.*

3.3 Typical solution techniques in the literature

It has been established in §3.2.3 that the ICFMO problem is a very large combinatorial optimisation problem. Suitable modern solution techniques may, however, be able to obtain good solutions to such large problem instances within a reasonable computation time. Many solution techniques have been employed in the context of ICFMO and several of the popular ones are presented in this section. These include knowledge-based/expert systems, mathematical programming techniques, and metaheuristics.

3.3.1 Knowledge-based/expert systems

Nuclear reactors have been operated in various places around the world for several decades and a vast amount of experience has been accumulated by field experts charged with designing reload configurations. Their invaluable knowledge may be included in *expert systems* in order to attempt solving instances of the ICFMO problem. An expert system is, however, only as effective as the rule sets contained in its knowledge base. Examples of expert systems in the literature include those designed by Kim *et al.* [100], Lin and Lin [121], and Tahara *et al.* [212].

The FUELCON system is arguably the most prominent expert system for ICFMO available in the literature [61, 62, 63, 147]. A brief discussion on its working is presented as an example. FUELCON contains a knowledge base in the form of "IF-THEN" rule sets that may generally be classified as *forbidden-type rules* or *preference rules*. The difference between these two classes is the following. Forbidden-type rules correspond to strict requirements that aim to exclude potentially infeasible solutions (*i.e.* they are constraint-related) whereas preference rules are enforced only when possible so as to guide the search towards preferable solutions (*i.e.* they are objective-related). In FUELCON, solutions are constructed one fuel assembly at a time according to a predefined loading sequence. Using the rule sets within an exhaustive enumeration search mechanism results in the identification of families of potential solutions. The system is interactive, thereby allowing the reload design engineer to expand or refine the rule sets according to the feedback he/she receives. In a later version of FUELCON, this manual revision of the rule sets may be performed by an automated procedure employing neural network learning algorithms [148].

3.3.2 Mathematical programming techniques

It was mentioned in §3.1 that, during the early years of ICFMO research, simplified reactor models were employed so as to formulate fuel management optimisation problems as linear and

nonlinear programming problems. This was done in order to take advantage of the traditional mathematical programming solution techniques available at that time. Over the years, significant improvements have been made in computing power and in the efficiency of mathematical programming techniques. Accordingly, less restrictive simplifications may be assumed in reactor models, allowing for more realistic (albeit still simplified) ICFMO problem formulations to be considered in a linear or nonlinear programming context.

The renowned simplex algorithm still forms the basis of many linear programming solvers used today and was employed in the context of ICFMO, for example, in [208]. Similarly, branch-and-bound methods are generally used for solving integer optimisation problems since they are still the most viable solution methods available for such problems. The integer program in [194], the mixed-integer linear program in [101] and the mixed-integer nonlinear program in [168] were, for example, all solved by algorithms that employ a branch-and-bound method. Finally, two popular techniques for solving nonlinear programming problems are the method of successive linear programming and the generalised reduced gradient[9] algorithm. These two techniques were employed in [128, 206] and in [31, 168] within the context of ICFMO to solve a mixed-integer nonlinear program.

3.3.3 Single-objective metaheuristics

The design and application of metaheuristic[10] techniques for solving hard optimisation problems is a very active field of research. Although there is no universally accepted definition of what a metaheuristic is, the following definition was recently proposed by Sörensen and Glover [199]:

> "*A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework.*"

Metaheuristics are *approximate* solution techniques designed specifically to find high-quality solutions to hard optimisation problems within acceptable computation times. As a result, there is no guarantee that a solution obtained by a metaheuristic is optimal. Therefore, the term "approximate solution" is adopted in this dissertation when referring to a solution yielded by a metaheuristic. In contrast, *exact* solution techniques are designed in such a way that an optimal solution is guaranteed to be found within a finite amount of time (although this time may be prohibitively long).

The application of several metaheuristics to the ICFMO problem in recent years has garnered much attention in the literature. An overview of the most widely-used metaheuristics in the context of ICFMO is presented in this section, namely simulated annealing, genetic algorithms, particle swarm optimisation and ant colony optimisation. A number of additional metaheuristics, attracting less attention in the literature, are also briefly mentioned.

[9] The reduced gradient method was developed for solving linearly constrained nonlinear programming problems. It involves the partitioning of variables into two groups (independent and dependent) and, at each stage of the method, the problem is considered only in terms of the independent variables. The *generalised reduced gradient* algorithm, in turn, extends the reduced gradient method by allowing nonlinear constraints and arbitrary bounds on variables. See [124] for more details.

[10] The term *metaheuristic*, first used by Fred Glover in 1986, is derived from the Greek prefix *meta-* (μετα-), which means *beyond, above*, or *at a higher level*, and the Greek word *heuriskein* (ευρισκειν), which means *to find*.

Simulated annealing

Simulated annealing (SA), developed by Kirkpatrick *et al.* [103], is a local search[11] metaheuristic based on an analogy with the physical phenomenon of annealing. During the process of annealing, a physical system is led to a low energy state by carefully controlling its temperature [41]. In the context of strengthening metals, for example, the annealing process starts by heating the metal to bestow a high energy to it. Thereafter, the metal is slowly cooled in stages, with the temperature being kept constant during each stage for a sufficient duration. Such a controlled decrease in temperature eventually leads to the metal attaining a stable solid state which corresponds to an absolute minimum energy configuration. Metals are typically very strong in such a state, containing few defects.

The SA algorithm may be used to solve an optimisation problem in a manner analogous to the process of annealing described above. In the analogy, a solution corresponds to a certain *state* of the metal whereas the objective function (in a minimisation problem) corresponds to the free *energy* in the system. A *temperature* control parameter determines the number of accessible energy states and, when lowered, it should lead to an optimal state. The final solution corresponds to the system attaining its absolute minimum energy, *i.e.* being *frozen* in its ground state.

The following two principles from the field of statistical physics are utilised in the SA algorithm [41]. When a system reaches *thermodynamic equilibrium*, the probability of the system attaining an energy E is proportional to the Boltzmann factor, $\exp(-E/k_BT)$, where k_B denotes the Boltzmann constant, and T is the temperature. The energy states then follow the Boltzmann distribution at the given temperature. Furthermore, the so-called *Metropolis algorithm* is usually employed for simulating the state changes in a system during its progress towards thermodynamic equilibrium (at a given temperature). Starting from an initial state, the system is subjected to a perturbation. If the perturbation causes a decrease in energy, the perturbed state is accepted with certainty; otherwise, the perturbed state, with an increased energy of ΔE, is only accepted with probability $\exp(-\Delta E/T)$. Repeated iterations of the Metropolis algorithm results in the system eventually reaching its equilibrium state for a given temperature. At that point, the temperature may be decreased in order for the system to reach a new equilibrium state, and so forth. The effect of the temperature is that at high temperatures, the factor $\exp(-\Delta E/T)$ is close to 1, thus causing the Metropolis algorithm to accept the majority of perturbed states. At low temperatures, however, the factor $\exp(-\Delta E/T)$ is close to 0, thus causing the Metropolis algorithm to reject the majority of increasing energy perturbations.

In the analogy, therefore, some initial solution is slightly modified and, if it results in an improved objective function value, the modified solution is accepted as the new solution; otherwise, it is only accepted with a certain probability related to the magnitude of objective function value difference. This occational worsening in objective function value prevents the system (or search) from becoming trapped in a local minimum. Also, by starting at a high temperature, the SA algorithm considers as many solutions as possible in an attempt to explore the decision space. Thereafter, the temperature is decreased according to a specific cooling schedule in order for the algorithm to converge to a solution which yields a minimum objective function value. The basic working of the SA algorithm is illustrated in Figure 3.4 by means of a flow diagram.

In 1991, the SA algorithm was, to the best knowledge of the author, the first application of any metaheuristic to the ICFMO problem [109]. That same work led to the development of a suite of computer codes for fuel management optimisation called *Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing* (FORMOSA) [129, 142]. Stevens *et al.* [204] implemented the

[11] A *local search* seeks good solutions by iteratively making small modifications to a single solution [68].

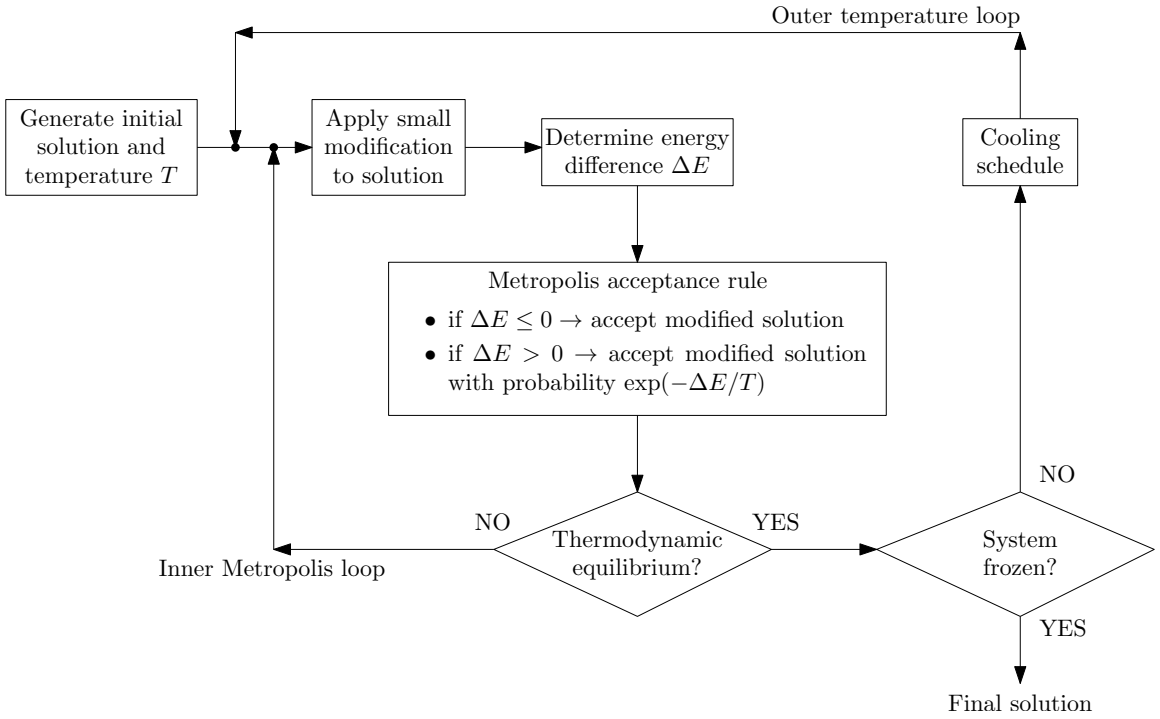


FIGURE 3.4: *Flow diagram of the simulated annealing algorithm in the context of a minimisation problem.*

SA algorithm as solution technique within the SIMAN optimisation module of the commercial software package XIMAGE[12]. Another example of commercial software employing SA as solution technique is the ROSA[13] package [223]. Notable applications of the SA algorithm to ICFMO, in addition to those mentioned above, include [74, 114, 131, 197, 237].

Genetic algorithms

A *genetic algorithm* (GA) is a population-based[14] metaheuristic inspired by the biological theory of evolution by natural selection proposed by Charles Darwin. Paraphrased from [41], these theories together state that evolution in a biological species occurs due to the competition in which the best-adapted individuals are selected for survival, while ensuring the continuation of the species through the transmission of useful characteristics from these individuals to their offspring. This transmission (*i.e.* inheritance) is based on a form of cooperative sexual reproduction. Genetic algorithms reside within a broader class of metaheuristics known as *evolutionary algorithms*. According to [41], GAs were first proposed Holland [81] in 1975, although they only became popular after the seminal work of Goldberg [69] in 1989.

In an analogy to Darwin's theory, solutions to an optimisation problem may be thought of as corresponding to the *individuals* of a species, while a subset of individuals considered simultanously at any point in time is referred to as a *population*. Each individual is associated with a *fitness* level which specifies the desirability of the individual being selected for reproduction or replacement. Fitness levels are determined by a *fitness function* which naturally depends on the

[12]XIMAGE/SIMAN-PWR is a graphical fuel management and loading pattern optimisation suite developed by Studsvik Scandpower, Inc. [235].

[13]ROSA is an acronym for *Reload Optimization by Simulated Annealing* and is developed by the *Nuclear Research and consultancy Group* (NRG) in the Netherlands.

[14]A *population-based* method seeks good solutions by iteratively selecting and then combining solutions from an existing set, usually referred to as the population [68].

objective function associated with the optimisation problem. A population iteratively evolves over time, with each iteration being referred to as a *generation*. During each generation, specific evolutionary operators act on the individuals of the current population in order to create a new population for the next generation. This new population is created by individuals reproducing, surviving or disappearing from the current population. The existing individuals that are used during reproduction are called *parents*, while the new individuals created during reproduction are called *offspring*.

The evolutionary operators are partitioned into two categories. The first category consists of two *selection operators*, namely: 1) selection for *reproduction*, which determines the likelihood that an individual will be chosen to reproduce; and 2) selection for *replacement*, which determines the specific individuals that will have to be discarded (or kept) so as to maintain a fixed population size. The second category consists of *variation operators* which are further partitioned into *mutation* operators and *crossover* operators. Mutation operators modify a single individual to form another one, whereas crossover operators fulfil the role of sexual reproduction by creating one or more offspring from a combination of two or more parents.

The above-mentioned description encompasses the basic working of a generic GA. This working is also illustrated by means of a flow diagram in Figure 3.5. The oval shapes indicate the application of the selection operators whereas the hexagonal shapes correspond to variation operators.

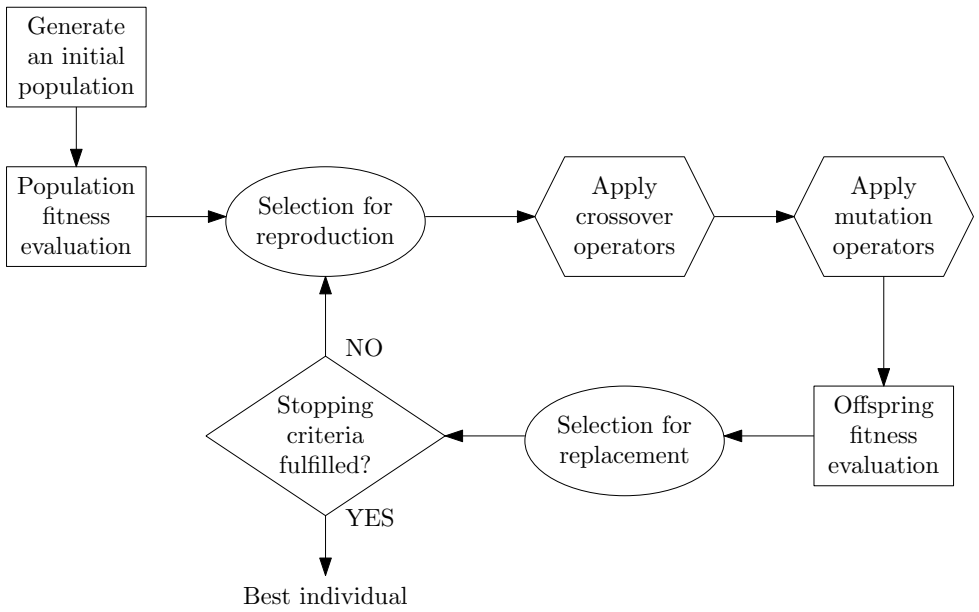


FIGURE 3.5: *Flow diagram of a generic genetic algorithm.*

According to Jayalal *et al.* [92], the first application of a GA to the ICFMO problem was by Poon an Parks [163] in 1992, only one year after SA was first applied. Examples of fuel management optimisation software packages employing GAs as their solution techniques are the *code independent genetic algorithm reactor optimization* (CIGARO) system [35], and the more extensive *genetic algorithm reactor code optimization* (GARCO) package [1], both of which were developed at Pennsylvania State University over many years. Other notable applications of GAs to the ICFMO problem may be found in [27, 56, 83, 242, 245]. A fairly extensive survey on GA applications to fuel management optimisation may be found in [92].

Particle swarm optimisation

Proposed initially by Kennedy and Eberhart [96] in 1995, *particle swarm optimisation* (PSO) is a population-based metaheuristic inspired by the flocking of birds. A particular source of inspiration was computer simulation studies in which simple rules for information sharing between individuals were employed to model the coordinated search for food by a flock of birds. PSO resides within the broader class of metaheuristics known as *swarm intelligence* methods, which consist of techniques inspired by the collective behaviour of swarms and social insect colonies. These techniques are often characterised by principles of *self-organisation* and local/indirect *information exchange* [24, 41].

In PSO, a population of *particles* (referred to as the *swarm*) is maintained, with each particle representing a location in the decision space of an optimisation problem. Particles search for an optimal location by moving through the decision space, analogous to a flock of birds flying over some region foraging for food. The flight path that an individual bird decides to take is based on cognitive aspects (modelled by the influence of its own location history) and social aspects (modelled by the influence of other birds' location histories) [24]. A bird, therefore, may fly towards a location containing food that it knows about, or towards a location containing food that the rest of the flock knows about. Accordingly, in PSO, the movements of any particle are governed by its current *velocity*, and the *positions* of good locations already found by the particle itself (cognitive aspect) or by other particles in the swarm (social aspect). Typically, the social aspects are considered in the context of a neighbourhood topology (*i.e.* the social network within the swarm). This topology defines, for any particle, with whom it can communicate and is often chosen as a fully-connected network topology, meaning that every particle can communicate with every other particle.

The basic PSO algorithm proceeds as follows. A swarm of particles is initialised with certain positions and velocities. Thereafter, the *personal best* position of each particle is tracked, as well as the *global best* position of the swarm (or the neighbourhood best solution, if a limited topology is chosen). The velocity of each particle is then iteratively adjusted towards a combination of its personal best and the global best solutions, along with a random component. The working of the PSO algorithm is illustrated by means of a flow diagram in Figure 3.6.

PSO was first applied in the context of ICFMO by Meneses *et al.* [137] in 2009. Since then, different versions of the PSO algorithm have been applied on multiple occasions for solving instances of the ICFMO problem [9, 90, 123, 136].

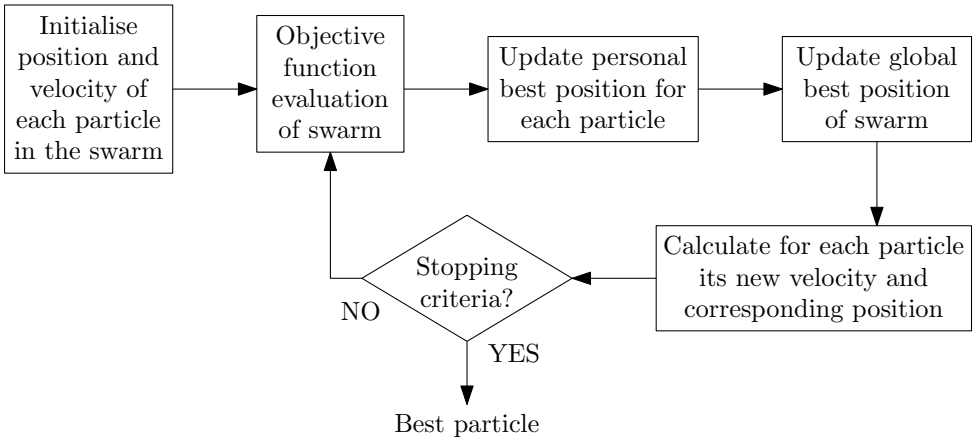


FIGURE 3.6: *Flow diagram of the basic particle swarm optimisation algorithm.*

Ant colony optimisation

The term *ant colony optimisation* (ACO) refers to a class of constructive[15] metaheuristics that are inspired by the foraging behaviour of ants, always seemingly to be able to find a shortest path between their nest and a food source. The success of the ants is driven by their ability to communicate indirectly with one another by dynamically modifying their environment — a concept known as *stygmergy* [41]. ACO also resides within the class of swarm intelligence metaheuristics.

Ants use volatile substances, called *pheromones*, to communicate with one another. These pheromones are deposited onto the ground, leaving an odorous trail for other ants to follow. Over time, however, the pheromones *evaporate* unless more ants deposit new pheromones along the same path. This is exactly where a natural optimisation of the routes occurs. An ant typically follows a path along which higher pheromone levels have been deposited with a larger probability. Since a shorter path may be traversed faster, the quantity of pheromone is slightly more significant than on a longer path due to evaporation. Ants therefore travel more frequently along these shorter paths, which results in more pheromone being deposited along those paths. In contrast, pheromone levels along longer paths decrease over time. The end result is a system reinforcing itself until all the ants follow a single (shortest) path.

The first ant algorithm, proposed by Dorigo *et al.* [39] and called the *Ant System* (AS), was specifically designed for solving the travelling salesman problem[16]. The analogy is simple: cities correspond to different food sources and the ants have to find the shortest path connecting all the cities (visiting each city exactly once). Numerous improvements and variations have been made to the AS algorithm since then, leading to the variety of ACO algorithms available today. The AS algorithm is presented here as an example of a basic ACO algorithm.

Consider a complete graph in which the vertices correspond to the cities of the travelling salesman problem and the edges correspond to the roads between the cities. During each iteration of the AS algorithm, each ant traverses the graph and constructs a complete path that translates into a solution. Suppose an ant is at a given vertex during the construction of its path. The probability that the ant moves to a next vertex is determined by a so-called "rule of displacement" (or the *transition rule*) [41]. This probability depends on: 1) the list of vertices not yet visited by the ant; 2) the *visibility* of each vertex, determined by heuristic information; and 3) the quantity of pheromone deposited on the edge joining the two vertices, called the *trail intensity*.

At the end of every iteration, each ant deposits a quantity of pheromone along its entire path. Such a pheromone deposit, which depends on the quality of a solution, promotes the search for good solutions. The evaporation of existing pheromones should, however, also receive attention at the end of every iteration in order for the system to move beyond poor solutions obtained during previous iterations [41]. This may be achieved by a constant decrease in trail intensity to counterbalance the additive effects of pheromone deposits by ants. In Figure 3.7, a flow diagram is presented illustrating the working of the AS algorithm.

As mentioned earlier, there are a variety of ACO algorithms available in the literature. Specific examples of ACO applied to the ICFMO problem include: the *Ant-Q* algorithm [126], the *ant colony system* [32], the *max-min ant system* [79] and the *rank-based ant system* [119]. In the ant model adopted by Machado and Schirru [126], for example, vertices in the graph correspond to the fuel assemblies of the ICFMO problem whereas the edges in the graph now depend on the

[15] A *constructive* method seeks good solutions by constructing a solution from its constituting elements, *i.e.* by adding elements one at a time to a partial solution [68].

[16] Given a number of *cities* and the *distances* between all pairs of cities, the travelling salesman problem asks for a shortest closed tour in which each city is visited exactly once [130].

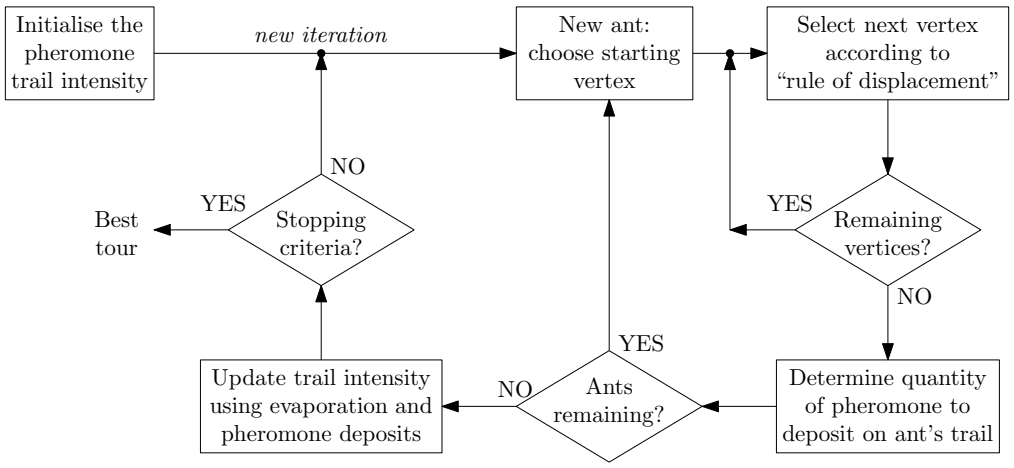


FIGURE 3.7: *Flow diagram of the ant system algorithm.*

loading positions. The authors created a so-called "ordering map" which assigns a fuel assembly selected by an ant traversing the graph to a specific loading position (so that a complete path may translate into a valid reload configuration). Accordingly, an edge represents the possibility of a fuel assembly being selected and assigned to a specific loading position.

Other metaheuristics

Apart from the more prominent metaheuristic applications discussed above, a number of other metaheuristics have also been considered in the context of ICFMO. Since they have attracted less attention in the literature, only a brief mention is made of each in this section.

Tabu search is a local search metaheuristic that incorporates a memory structure to prohibit the revisitation of recent solutions or moves. Escapes from local optima are possible, because non-improving solutions are allowed when no improving solution is found within the neighbourhood[17] of the current solution. Tabu search was one of the earliest metaheuristics applied to ICFMO [120], but re-emerged only recently in the literature [76].

Harmony search is a constructive metaheuristic based on an analogy to an improvisation jazz band playing different sounds on their instruments (different values for decision variables) in order to find a pleasing combination of sound (good objective function values). Applications of harmony search to the ICFMO problem may be found in [165, 182].

The *artificial bee colony* algorithm and the *firefly algorithm* are two swarm intelligence metaheuristics closely related to PSO. Application of these techniques to the ICFMO problem may be found in [33] and [166], respectively.

Finally, a number of probabilistic model-based metaheuristics have also been proposed in the context of ICFMO. In these metaheuristics, a probability model is constructed, taking into account the performance of previous solutions. New solutions are then generated (sampled) according to this distribution. Examples of ICFMO applications include *population-based incremental learning* [25] and several *estimation of distribution* algorithms [93].

[17]The *neighbourhood* of a solution consists of several solutions that are "close" to it and may be generated by applying a neighbourhood move operator, or move set, to the solution in question. This operator/set defines how elementary modification may be affected to a solution.

3.3.4 Multiobjective metaheuristics

The metaheuristics described in §3.3.3 were all designed for solving single-objective optimisation problems. In such instances, the aim of a metaheuristic is to find a solution that optimises one objective function. In multiobjective optimisation problems, however, several objective functions have to be optimised simultaneously. There is typically no single solution that is optimal with respect to every objective function simultaneously. Instead, a set of trade-off solutions, also known as nondominated solutions, may be sought. A solution is said to be *dominated* if another solution exists that is better than the original solution with respect to at least one objective, and no worse than that solution with respect to all of the other objectives. A solution is then *nondominated* if no other solution dominates it. The aim of a multiobjective metaheuristic is therefore to find a set of nondominated solutions.

A very limited amount of literature is available on MICFMO, especially in respect of solution techniques. In this section, an overview is presented of the multiobjective metaheuristics that have been applied to the ICFMO problem.

Several of these metaheuristics employ a quantity known as the *nondominated rank* of a solution as a measure of its quality. These ranks may be calculated as follows. Given some set of solutions, all nondominated solutions contained therein are identified, and allocated a rank of 0. Then, those solutions are removed from future consideration. The nondominated solutions contained in the remainder of the set are then identified, and allocated a rank of 1. This procedure is repeated until all the solutions have been ranked.

In addition, some of the metaheuristics employ an external (or secondary) set of solutions during their execution. This set is often referred to as an *archive* and it generally contains the nondominated solutions to the optimisation problem found thus far.

Multiobjective genetic algorithm approaches

Parks [157] designed a *multiobjective genetic algorithm* (MOGA) that utilises nondominated ranks and an archive of nondominated solutions. In this algorithm, each solution is assigned a fitness value equal to its nondominated rank within the population. Accordingly, solutions with a lower rank have a better fitness and are therefore more likely to be selected for reproduction and retained in the next generation. During the execution of the MOGA, an archive of nondominated solutions is also maintained. After a new solution has been evaluated, it may be considered for inclusion in the archive by means of domination testing[18]. The archive in this MOGA has a fixed size. When it is full, a new nondominated solution is only inserted if the solution is sufficiently dissimilar (in decision space) to an existing archive member. A dissimilarity measure based on reactivity distributions is defined in [157] for this purpose. Finally, a form of multiobjective elitism[19] is introduced into the MOGA by selecting every solution in the archive (up to a maximum of one quarter the size of the population) for reproduction.

Do and Nguyen [37] borrowed the notion of multiobjective elitism employed in the MOGA by Parks [157] in their GA for the ICFMO problem. In this algorithm, a linear weighted sum aggregation of multiple objectives is employed as the fitness function. An archive of nondominated

[18] Assume that the nondominated archive may have an unlimited size. Then, if any members in the archive are dominated by the new solution, those are removed and the new solution is inserted. Similarly, if the new solution is nondominated with respect to the archive, it is inserted. Otherwise, if the new solution is dominated by any member in the archive, it is not inserted.

[19] *Elitism*, or an *elitist strategy*, refers to the preservation of (at least) the best solution in the population, from one generation to the next one [41].

solutions is also maintained by the algorithm. There is, however, no dissimilarity measure to differentiate between two nondominated solutions. Instead, only the fitness function value is considered in this regard. As was the case in the algorithm of Parks [157], every solution in the archive is selected for reproduction in this GA as a form of multiobjective elitism.

Finally, in 2009, Hedayat *et al.* [75] applied the highly-popular *nondominated sorting genetic algorithm II* (NSGA-II) to the ICFMO problem. The algorithm was initially developed by Deb *et al.* [34] as an improvement over its predecessor. Within the NSGA-II, the fitness of a solution is determined by its nondominated rank, as well as its crowding distance, which is a measure of the density of solutions surrounding a particular solution. A so-called crowded comparison operator is used to differentiate between two nondominated solutions. In contrast to the two algorithms described above, the NSGA-II does not employ an archive of nondominated solutions. Instead, elitism is introduced by combining the offspring and parent populations into a single set, sorting (ordering) the solutions according to their nondominated ranks and crowding distances, and then truncating this combined set of solutions to the required population size. A more detailed discussion on the working of the NSGA-II is presented in a later chapter since it is employed as a solution technique in this dissertation.

Multiobjective simulated annealing approaches

A *multiobjective simulated annealing* (MOSA) algorithm was designed by Engrand [55] for application to the ICFMO problem. In the algorithm, an aggregating function G is defined as $G(\boldsymbol{x}) = \sum_{k=1}^{q} \ln f_k(\boldsymbol{x})$, where f_k denotes objective function $k \in \{1, \ldots, q\}$. This function represents the energy of the system, and is used within the Metropolis algorithm to calculate the acceptance probability of a modified solution. In the context of minimisation, when $\Delta G \leq 0$, the modified solution improves the objectives "on average" and it is accepted as the new solution. Otherwise, when $\Delta G > 0$, the modified solution is only accepted with probability $\exp(-\Delta G/T)$. During the execution of the MOSA algorithm, an archive of nondominated solutions is again maintained. Only *after* a modified solution has been accepted as the new solution, however, may it be considered for inclusion in the archive. Finally, the MOSA algorithm may periodically reset the current solution to one that is present in the archive.

Improvements on Engrand's MOSA algorithm were suggested by Parks and Suppapitnarm [158] after they identified two weaknesses in the original algorithm. The first of these weaknesses relate to the aggregating function G, which may favour some objectives over others, whereas the second relates to the acceptance criterion and subsequent archiving of a solution, which may exclude certain nondominated solutions. In their improved MOSA algorithm, a modified solution is considered for inclusion in the archive *before* the acceptance test is performed. If this solution is archived, it is also automatically accepted as the new solution. Otherwise, an overall acceptance probability is calculated as the product of the individual acceptance probabilities corresponding to each objective, *i.e.* it is given by $\prod_{k=1}^{q} \exp(-\Delta E_k/T_k)$, where the pursuit of objective k is equipped with its own temperature T_k. As such, there is no longer an aggregating energy function in this improved MOSA algorithm. Also, instead of choosing any solution in the archive during the periodic resetting of the current solution, only a subset of archived solutions is considered in the modified algorithm. This subset should always include those solutions attaining the best objective function value in each objective, along with a few randomly chosen solutions. Parks and Suppapitnarm mention that constraints may be handled according to the discretion of the user. They may, for example, be treated as hard constraints such that any solution violating a constraint is never accepted (or archived), or the magnitude of the constraint violation may be considered as an additional objective to be minimised.

Further refinements on Parks and Suppapitnarm's improvement of Engrand's MOSA algorithm were suggested by Kellar [95]. Most of these refinements were technical in nature, such as an equivalent expression for the acceptance probability which avoids arithmetic over/underflow, and modifications to the cooling schedule. More importantly, however, Keller introduced a temperature weighted aggregating function in order to differentiate between two nondominated solutions (for when a new solution potentially replaces an existing member in the archive). In addition, only those archived solutions attaining the best objective function value in each objective are contained in the subset to choose from during the periodic resetting of the current solution. Keller considered constraint violations as an additional objective in the refined MOSA algorithm. He suggested, however, that the nondominated archive should be limited only to solutions that do not violate any constraints.

Park *et al.* [156] proposed a MOSA algorithm similar to the one designed by Engrand [55] for solving the ICFMO problem. In their algorithm, objectives and constraints are aggregated into a so-called discontinuous penalty function J to be minimised. The minimum value that J can attain is zero and corresponds to a feasible solution having been found. During the execution of this MOSA algorithm, an archive of nondominated solutions is also maintained. In the algorithm, the current solution $\boldsymbol{x}_c$ is modified to create solution $\boldsymbol{x}_m$, which is then tested for feasibility, *i.e.* whether or not $J(\boldsymbol{x}_m) = 0$. If $\boldsymbol{x}_m$ is feasible, it is accepted as the new solution whenever the current solution is infeasible, or when both solutions are feasible and $\boldsymbol{x}_m$ dominates the current solution. If both solutions are feasible, but $\boldsymbol{x}_m$ does not dominate the current solution, then $\boldsymbol{x}_m$ may still be accepted, provided that it is not dominated by any member in the archive. Any feasible solution is also considered for inclusion in the nondominated archive. When $\boldsymbol{x}_m$ is infeasible, however, it is accepted as the new solution if its discontinuous penalty function value is less than that of the current solution. Otherwise, if this value is worse, it is only accepted with probability $\exp(-\Delta J/T)$. This acceptance procedure is illustrated in Figure 3.8.

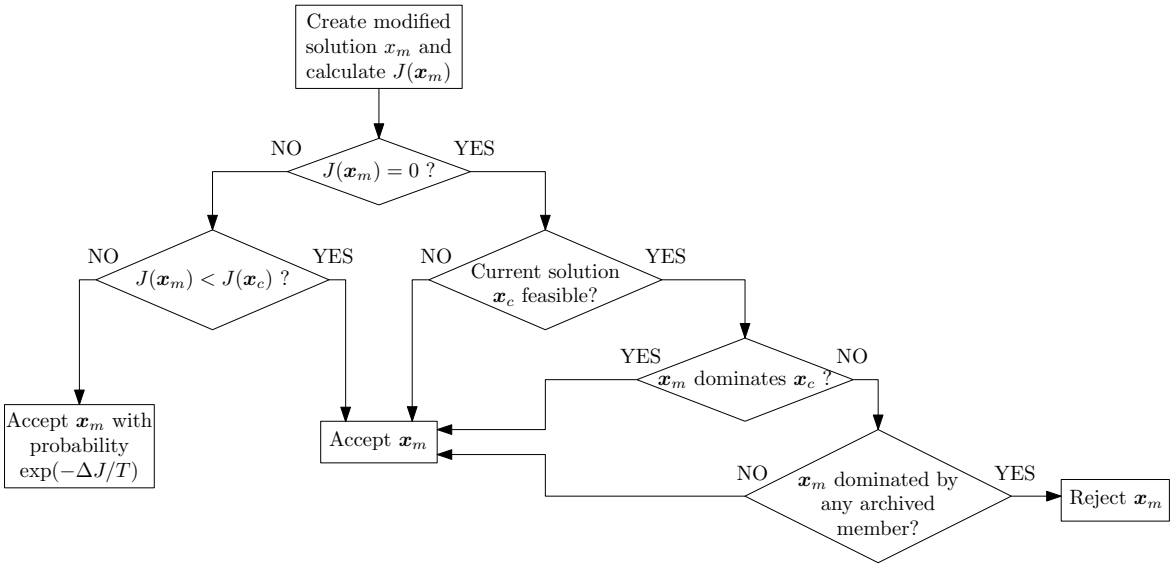


FIGURE 3.8: *New solution acceptance procedure in the MOSA algorithm proposed by Park et al. [156].*

Multiobjective particle swarm optimisation

Babazadeh *et al.* [9] applied the *vector evaluated particle swarm optimisation* (VEPSO) algorithm to the ICFMO problem. The algorithm was initially proposed by Parsopoulos and Vrahatis [159] and is based on using a separate swarm for each objective, along with some form

of information exchange between the swarms. In the VEPSO algorithm, each swarm is evaluated according to its own associated objective function. When the velocity of a particle in one swarm is updated, however, it uses information obtained from another swarm to do so (*e.g.* the global best solution from another swarm). Alternatively, it may also use a combination of personal best solutions and the global best solution from another swarm. In the implementation of the VEPSO algorithm by Babazadeh *et al.* [9], only the global best solutions within different swarms are considered for information exchange. Accordingly, the velocity of each particle is iteratively adjusted towards its own personal best solution, and the global best solution obtained within another swarm, along with a random component.

3.4 Approaches to reduce the ICFMO computational burden

It has been established that the ICFMO problem suffers from computationally expensive function evaluations. Several approaches have been adopted in the literature attempting to reduce this computational burden. Three such examples are touched upon in this section.

3.4.1 Simplified reactor models

As mentioned earlier, the use of *simplified reactor models* is one approach to overcoming this dilemma and was applied extensively during the early years of ICFMO research. The inaccuracy in reactor models thus introduced may, however, result in objective functions and constraints whose values are of unacceptable accuracy. Given the advancements made in computer technology over the years, simplified models with improved accuracy have been suggested for ICFMO again (see §3.3.2). These improved models have been employed successfully in the literature to reduce the computational burden associated with ICFMO [168, 194, 223].

3.4.2 Perturbation theory

An alternative approach that has received attention in the literature is to use *perturbation theory*[20] for estimating function values. One of the advantages associated with perturbation theory is the ability to control/determine the exact perturbation errors that arise from its usage. Perturbation theory is particularly useful (*i.e.* accurate) when utilised in conjunction with a local search metaheuristic, since new solutions are created by making small modifications/perturbations to an existing solution. The FORMOSA suite of computer codes (see §3.3.3) is a prominent example where perturbation theory has been employed in the context of ICFMO [109, 129]. Note that the SA method is employed as solution technique in FORMOSA.

3.4.3 Surrogate computational models

Finally, the computational cost associated with ICFMO function evaluations may also be reduced by replacing the core simulator with a computationally cheaper surrogate model. A popular approach adopted in the literature is to employ an *artificial neural network*[21] (ANN) for the prediction of reactor core parameters corresponding to objectives and constraints. In

[20] In *perturbation theory*, an approximate solution to a problem instance is sought by starting from the exact solution of a related problem.

[21] According to Fausett [58], "*an artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural networks.*"

this context, an ANN is used for function approximation and therefore has to follow a training process, called supervised training. During this process, training is performed by presenting the ANN with a set of known input-output pairs (*i.e.* solutions that have been evaluated by a core simulator, and their corresponding function values). A training algorithm then adjusts the network in such a way that the predicted and known outputs are close to one another. Unlike in perturbation theory, however, the prediction errors that arise from using ANNs cannot be controlled/determined exactly since the networks are trained on a limited number of solutions. Fortunately, the accuracy associated with ANNs in the context of ICFMO has been found to be acceptable. Examples in the literature of their usage may be found in [56, 131, 236]. A more detailed discussion on ANNs is presented in a later chapter since it is employed in this dissertation for reducing the computation time of ICFMO function evaluations.

3.5 Chapter summary

In this chapter, a survey of the most relevant literature pertaining to the ICFMO problem was presented. The aim thereof was to provide the reader with the necessary background knowledge required to proceed with new research towards ICFMO in an informed manner.

General ICFMO modelling considerations and an historical overview of early research in the field were mentioned in §3.1. This was followed, in §3.2, by discussions on the most popular objective function formulations in the literature, as well as typical constraints. The complexity in respect of the ICFMO problem's decision space was also elaborated on in §3.2.3.

An assortment of typical solution techniques applicable to the ICFMO problem was reviewed in §3.3, with a particular emphasis on metaheuristics. The most prominent single- and multiobjective metaheuristics were described in moderate detail in §3.3.3 and §3.3.4, respectively. Furthermore, several examples of their application in the ICFMO literature were also provided.

The chapter closed with a brief discussion in §3.4 on three different approaches considered in the literature for reducing the computational burden associated with solving instances of the ICFMO problem.

CHAPTER 4

The MICFMO problem in context

Contents

This chapter opens with a brief discussion on the topic of nuclear fuel management, thus placing ICFMO in context within this broader topic. This discussion is followed by an elucidation of several problem assumptions that are required in order to formulate a model for the MICFMO problem. The reactor core calculation code system utilised in this dissertation is also briefly described. Thereafter, the two nuclear research reactors considered as case studies in this dissertation are discussed in detail, along with the objectives and constraints associated with each reactor for MICFMO. A short description of the current reload configuration design approach employed at each reactor is also presented.

4.1 Nuclear fuel management

It was mentioned in Chapter 1 that ICFMO is contained within the broader topic of nuclear fuel management, which is one of the most important aspects of nuclear reactor operation. Silvennoinen [195] states that nuclear fuel management encompasses the fuel composition and loading aspects (be they related to physical, engineering or economic considerations) that influence optimal fuel utilisation in a reactor core, subject to the design limits imposed on it. In the context of light water reactors, nuclear fuel management is typically partitioned into two topics, namely *out-of-core fuel management* (OCFM) and *in-core fuel management* (ICFM) [219]. Due to its convenience in terms of reducing the complexity of nuclear fuel management and its widespread utilisation in the field of nuclear engineering, [40, 217, 219, 221], the above partitioning is also adopted in this dissertation.

The topic of OCFM is essentially aimed at answering the questions "*What to purchase?*" and "*What to reload?*" over a planning horizon that spans multiple operational cycles [219]. Once a decision has been made as to the fuel assemblies available for loading during a specific operational cycle, the topic of ICFM is aimed at answering the question "Where to position?" these available assemblies [219]. A listing of specific decisions related to light water reactor nuclear fuel management may be found in [217]. These decisions, and how they relate to the above partitioning of nuclear fuel management, are presented diagrammatically in Figure 4.1.

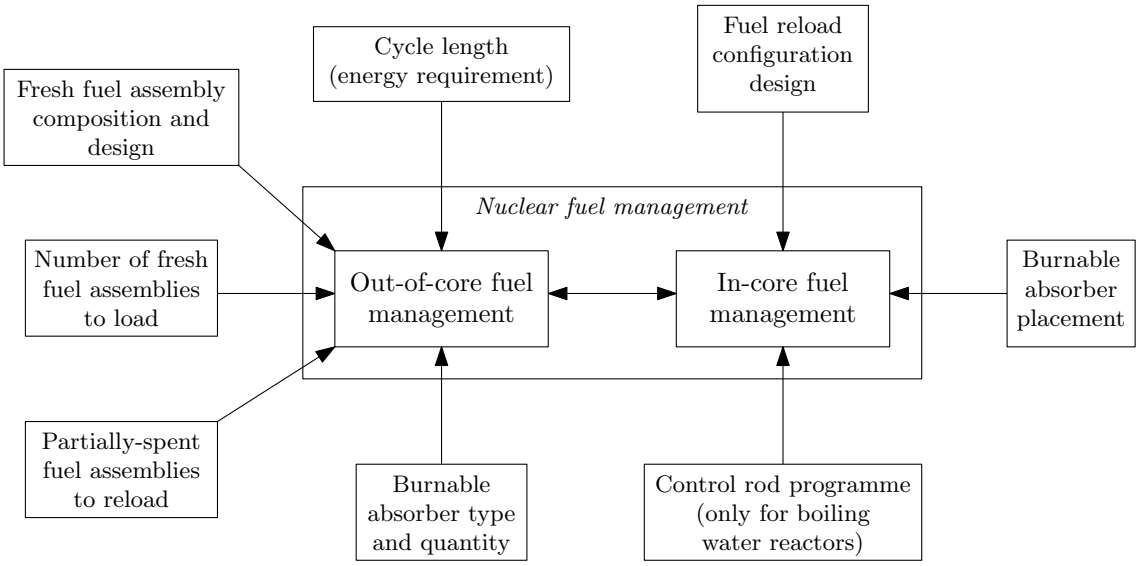


FIGURE 4.1: *Nuclear fuel management in the context of light water reactors.*

Cycle length is influenced by the utilisation requirements specific to a reactor (*e.g.* for a utility to meet its seasonal electricity generation demand) as well as the fuel cycle costs. The composition and design of fresh fuel assemblies, the type and quantity of burnable absorbers[1], the number of fresh fuel assemblies to load, and which of the partially-spent fuel assemblies to reload in the core, are highly-coupled decisions [217]. They are influenced by the cycle energy requirements, fuel cycle costs and reactor safety requirements. The fuel reload configuration design, as already mentioned, involves decision making on where to locate the available fuel assemblies in the reactor core. In some reactors, especially in PWRs, the locations of burnable absorbers in

[1] A *burnable absorber*, also known as a *burnable poison*, is a neutron-absorbing material whose effectiveness decreases over time when introduced in an active reactor core. It is generally used to control the power peaking in a reactor during the initial stages of its operating cycle.

the fresh fuel assemblies across the core have to be determined because of their significant effect on the power distribution [217]. Finally, for BWRs, a control rod programme[2] has to be determined for a specific fuel reload configuration. This programme affects the power and burnup distribution across the core, while also ensuring adequate reactivity control in the reactor.

4.2 Problem assumptions

In this section, several assumptions are made in respect of the ICFMO problem, residing within the broader context of nuclear fuel management, in order to reduce its complexity to manageable levels without significantly compromising its practical applicability.

4.2.1 Out-of-core fuel management

The partitioning of nuclear fuel management into OCFM (as a multicycle decision process) and ICFM (as a single-cycle decision process) is convenient, but artificial — the decisions in these two contexts clearly affect one another [217].

The ICFMO problem may therefore be modelled together with OCFM decisions as a single optimisation problem, with its solution being applicable to the entire multicycle planning horizon. The dimensionality of the optimisation problem, however, increases significantly for each cycle added to the planning horizon. Ultimately, the computational burden associated with a multicycle may escalate beyond practicality. A multicycle problem also requires additional burnup calculations to be performed by a core simulator in order to estimate fuel assembly isotopic compositions over the entire planning horizon. These additional computations may be excessively time-consuming, thus also potentially rendering the problem impractical. Furthermore, deviations from a reactor's planned operating schedule may occur due to new operational requirements or unforeseen events, such as a change in cycle length or power output, a reactor scram[3], the replacement of defective fuel assemblies, *etc.* [221]. These deviations will cause discrepancies between the estimated and true fuel assembly isotopic compositions over the planning horizon. In this context, "true" refers to the fact that the true operating history is utilised for calculating isotopic compositions. It does not refer to physically measured isotopic composition. Accordingly, reload configurations in the multicycle solution may be too inaccurate (*e.g.* far from optimal, or even infeasible) during later operational cycles.

Alternatively, the ICFMO problem may be modelled separately from OCFM. In such a segregated modelling approach, it is generally assumed that ICFM has little to no influence on OCFM [40]. Accordingly, the OCFM optimisation problem is considered first and its solution is typically translated into fixed input requirements (*e.g.* in the form of parameters) for the ICFMO problem to be considered thereafter. The ICFMO problem is not significantly affected in such a segregated modelling approach because reasonably accurate OCFM decisions that impact on the problem are still incorporated by means of the aforementioned fixed input requirements.

It is therefore assumed that the ICFMO problem is modelled separately from OCFM, to the extent that only a set of input requirements obtained during OCFM is necessary for ICFMO. Furthermore, since OCFM falls outside the scope of this dissertation, the input requirements are

[2]The insertion patterns of the control rods/blades in a BWR core as a function of the operating cycle exposure is referred to as a *control rod programme* [217].

[3]The United States Nuclear Regulatory Commission defines a reactor *scram* as "*The sudden shutting down of a nuclear reactor, usually by rapid insertion of control rods, either automatically or manually by the reactor operator. Also known as a* reactor trip." [220].

assumed to be readily available and no knowledge is necessary as to the manner in which it was obtained. In accordance with the partitioning of nuclear fuel management, it is also assumed that the ICFMO problem is applicable in the context of single-cycle optimisation only.

4.2.2 Fuel assemblies

In principle, more fuel assemblies may be available for the design of a reload configuration than there are loading positions in a reactor core to fill. In order to reduce the dimensionality of the optimisation problem, however, it is assumed that the number of fuel assemblies and the number of loading positions are equal. In accordance with the nuclear fuel management decision problem partitioning, the specific fuel assemblies that are available have been selected during the OCFM decision process. These assemblies form part of the fixed set of input requirements necessary for ICFMO (see §4.2.1). This assumption corresponds, for example, to what occurs in practice at the SAFARI-1 reactor — a subset of assemblies are selected from a larger available pool before designing the reload configuration. It is reasonable to assume that reactors other than SAFARI-1 may also employ it in practice.

All available fuel assemblies are also considered to be distinct from one another. This assumption reflects reality because each fuel assembly accrues its own burnup history, and associated isotopic composition with an axial distribution, as it moves through a reactor core along a unique path during its lifetime.

Finally, given an instance of the ICFMO problem for some operational cycle of a reactor, it is implicitly assumed that a subset of the available fuel assemblies for the problem instance resides in the reactor core during the preceding cycle. This assumption is also realistic because it is based on the typical manner in which a light water reactor core is reloaded.

4.2.3 The reactor type

Since it is assumed that the ICFMO problem is separate from OCFM, decisions related to the fuel reload configuration design, the burnable absorber placement, and the control rod programme remain for consideration (see Figure 4.1). The latter decision regarding a control rod programme is, however, only applicable to a BWR — a reactor type that does not fall within the research reactor scope of this dissertation. As such, control rod programming is considered to be a separate problem, not influencing ICFMO, and is excluded from the optimisation model adopted in this dissertation.

Burnable absorbers are widely used in PWRs, necessitating a decision on their placement across a reactor core. It is far less common, however, to use such absorbers in research reactors [224]. Given this low prevalence in research reactors and the fact that a PWR is a reactor type that also falls outside the scope of this dissertation, the decision on burnable absorber placement is not considered part of the ICFMO problem, and is therefore excluded from the optimisation model that follows. This assumption does not, however, preclude the usage of burnable absorbers in a reactor in the ICFMO problem. Since fresh fuel assemblies may contain burnable absorbers, they may be specified as such within the set of available fuel assemblies selected as part of the OCFM decision process (see §4.2.1).

4.2.4 Objectives and constraints

Recall from §3.1 that objective functions and constraints associated with the ICFMO problem are, in general, not available in closed form. It is therefore assumed that objectives and con-

straints should be expressible as parameters that may be obtained from a reactor core calculation code system. Furthermore, only neutronic calculations are assumed to be performed by the code system. Unless specified otherwise, it is assumed here that the parameters are to be calculated at *beginning-of-cycle* (BOC) core conditions so as to reduce the computational time required for an evaluation.

4.2.5 Parallel computing

In accordance with the scope of this dissertation, the OSCAR-4 system is utilised as the reactor core calculation code system during ICFMO. The OSCAR-4 system does not, however, easily lend itself to perform different core calculations in a parallel computing environment. This is primarily due to the inherent directory structure and text-based input/output processing employed by the system. It is therefore assumed that a parallel-computing solution technique to the ICFMO problem is not a viable option to pursue in this dissertation.

4.2.6 Computational budget

The computational budget available for solving an instance of the ICFMO problem is directly related to the point in time at which the problem is considered in practice. There are generally only two such points in time within the context of research reactors.

First, the problem may be considered *during* the operational cycle that precedes the one of the current problem instance. Since it is assumed that a subset of the available fuel assemblies resides in the core during this preceding cycle, the fuel assembly isotopic compositions to be used in the problem instance are not known yet. Accordingly, the isotopic compositions have to be estimated first utilising a code system and assuming the planned operating schedule. An advantage of this approach is that the computational budget for solving the problem instance is not severely restricted. A disadvantage, however, is the risk that discrepancies between the estimated and true fuel assembly isotopic compositions may arise due to deviations from the planned operating schedule, as mentioned in §4.2.1. A solution to the problem instance may then potentially be far from optimal, or even infeasible. Under such circumstance, it may be necessary to resolve the problem, using accurate isotopic compositions, but now within a computational budget which is very restricted.

Secondly, the problem may be considered immediately *after* completion of the preceding cycle. A so-called *core-follow* calculation is performed first in which the true operating history of the cycle is captured, and accurate isotopic compositions are determined for the relevant fuel assemblies to be used in the problem instance. An advantage of this approach is that no isotopic composition discrepancies (between estimated and true calculations) can occur, and hence that a solution to the problem instance may be considered accurate. A disadvantage, however, is that the computational budget for solving the problem instance is now very restricted because of the limited shutdown period between the two cycles.

For the purposes of this dissertation, it is assumed that a very restricted computational budget is allowed for solving an instance of the ICFMO problem. For ease of reference, however, it will be referred to as a *limited* computational budget throughout this dissertation. This assumption is conservative since it encompasses the worst-case scenario of the first approach (*i.e.* resolving the problem instance) as well as any scenario within the second approach. The reality that access to high-performance computing facilities for solving an ICFMO problem instance is not necessarily available is a further motivation for this assumption — a nuclear reactor operator must often utilise a personal computer.

4.2.7 Summary of assumptions

The assumptions in this section may be summarised as follows. The ICFMO problem is considered in this dissertation within the context of research reactors, for a single operational cycle, and involves the decision making on a fuel reload configuration design for a reactor core. The number of available fuel assemblies to be used in this design is equal to the number of fixed loading positions in the core, and is fully determined as part of OCFM. These fuel assemblies are also considered distinct from one another. Objectives and constraints associated with the ICFMO problem are expressible as parameters that may be obtained from a code system performing neutronic calculations, and a limited computational budget is allowed for solving a problem instance.

4.3 The MICFMO model

The assumptions presented in the previous section for the ICFMO problem are naturally also directly applicable to the multiobjective problem. In this section, those assumptions are employed to formulate an appropriate model for the MICFMO problem.

Let n be the number of fixed loading positions, labelled $1, \ldots, n$, in a reactor core. Let n also be the number of available fuel assemblies, labelled $1, \ldots, n$. A reload configuration may then be represented by a permutation decision vector $\boldsymbol{x} = [x_1, \ldots, x_n]$ where $x_i = j$ denotes the assignment of fuel assembly $j \in \{1, \ldots, n\}$ to loading position $i \in \{1, \ldots, n\}$. This representation is depicted in Figure 4.2. By adopting a permutation representation, the decision vector satisfies the physical requirement that a fuel assembly can only be assigned once and to one position.

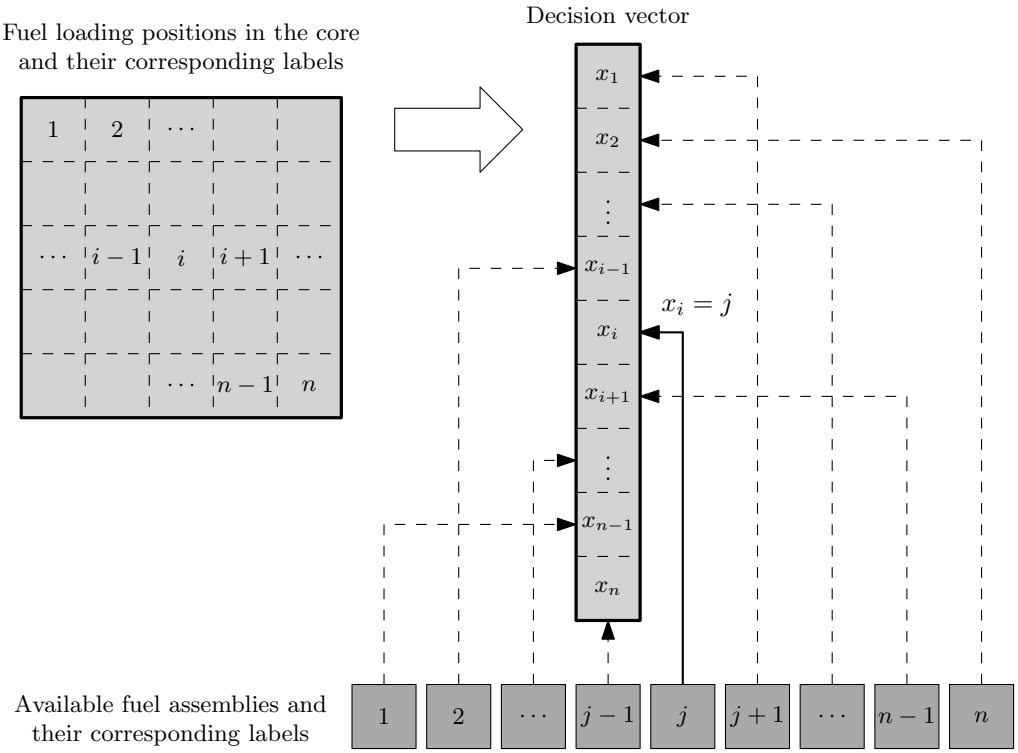


FIGURE 4.2: *Assignment of fuel assemblies to loading positions.*

Let $\mathcal{X}$ be the set of all possible reload configurations (*i.e.* permutation decision vectors) and suppose, without loss of generality, that all objective functions are to be maximised. Then, the general model of the constrained MICFMO problem with q objective functions $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})$ may be formulated as

$$\left.\begin{aligned} \text{maximise} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})], \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq g_i^{\lim}, \qquad i = 1, \ldots, r, \\ & h_j(\boldsymbol{x}) = h_j^{\lim}, \qquad j = 1, \ldots, s, \\ & \boldsymbol{x} \in \mathcal{X}, \end{aligned}\right\} \tag{4.1}$$

where $g_i(\boldsymbol{x})$ and $g_i^{\lim}$ for $i = 1, \ldots, r$ are the inequality constraint functions and their corresponding (non-zero) limiting values, respectively. Similarly, $h_j(\boldsymbol{x})$ and $h_j^{\lim}$ for $j = 1, \ldots, s$ are the equality constraint functions and their corresponding (non-zero) limiting values, respectively. The vector $\boldsymbol{f}(\boldsymbol{x})$ of objective function values corresponding to a decision vector $\boldsymbol{x} \in \mathcal{X}$ is referred to as the *objective vector*.

4.4 The OSCAR-4 code system

As already mentioned, the reactor core calculation code system utilised in this dissertation is the OSCAR-4 system. Although the system is employed as a "black-box" function evaluator during MICFMO, a brief overview of the system is presented here. The interested reader is referred to Stander *et al.* [203] for further details.

Developed in-house at Necsa, the OSCAR system is a state-of-the-art, advanced reactor simulation code providing calculation support to several research reactors across the world [153]. It is a deterministic core calculation code system, consisting of a two-dimensional transport code, a three-dimensional nodal[4] diffusion core simulator, and several related service codes. The latest production version of the system, OSCAR-4 (see Figure 4.3), was released in 2009.

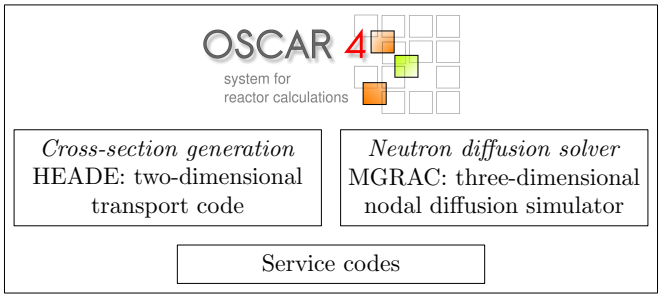


FIGURE 4.3: *The OSCAR-4 code system.*

In the two-dimensional transport code, called HEADE, few-group cross-sections are generated according to a low-order response matrix[5] formalism, while collision probability[6] methods are applied in the generation of the response matrices. These cross-sections are then employed during the solution of the three-dimensional problem.

[4] In *nodal methods*, the reactor core is decomposed into "*relatively large subregions or node cells in which the material composition and flux are assumed uniform (or at least treated in an average sense)*" [42].

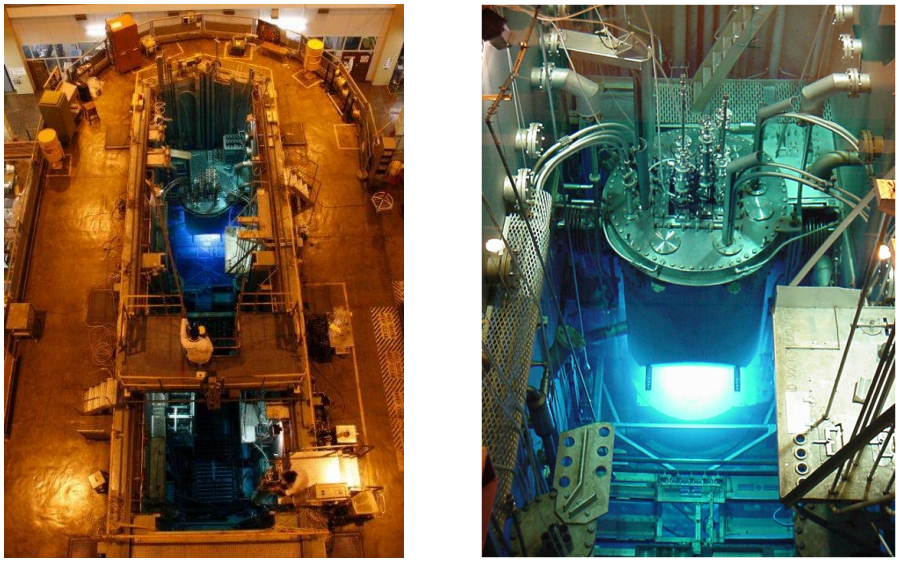
[5] The general idea behind the *response matrix method* is that "*the solution of a particle transport problem in a composite (large) domain ... is constructed from precomputed particular solutions to the local problems associated with each of the subdomains*" [122].

[6] The *collision probability method* is a popular numerical solution technique applicable to the integral form of the transport equation [179].

Full-core calculations are performed by the three-dimensional nodal diffusion simulator, called MGRAC, which employs a technique called the *multi-group analytic nodal method* [225]. This technique solves an initial-boundary value problem involving the three-dimensional, multi-energy group, time-independent diffusion equation very efficiently to determine the neutron flux distribution throughout a reactor core [153]. Finally, isotopic depletion calculations, cross-section parameterisation, and automated core-follow and reload calculations are performed by the related service codes.

4.5 The SAFARI-1 nuclear research reactor

The SAFARI-1 reactor, introduced in §1.2.2, is a 20 MW tank-in-pool type materials testing reactor in South Africa. Photographs of the reactor are presented in Figure 4.4. SAFARI-1 is primarily utilised for commercial irradiation services, as well as nuclear research, training and materials testing activities.



(a) (b)

FIGURE 4.4: *The SAFARI-1 nuclear research reactor [146].*

The commercial services rendered by SAFARI-1 revolve around the production of radioisotopes and the neutron transmutation doping of silicon[7]. Many of the radioisotopes are used for medical diagnostic purposes and the therapeutic treatment of cancer, whereas the silicon doping is performed to produce silicon semiconductors for use in electronic equipment [145]. In order to render these services, the reactor contains several in-core irradiation positions that are utilised for radioisotope production — primarily *molybdenum-99* (^{99}Mo) — while an ex-core facility is utilised for silicon transmutation doping. In respect of the nuclear and materials research conducted at SAFARI-1, neutron scattering, radiography and diffraction experiments are performed, utilising a number of neutron beam tubes[8] located around the core. These experiments find application in, for example, archaeology and palaeontology, civil and mechanical engineering, and the geosciences.

[7] *Neutron transmutation doping of silicon* is the intentional introduction of small quantities of phosphorus in pure silicon by means of a nuclear reaction between neutrons and silicon nuclides [145].

[8] A *beam tube* is essentially an "opening" on the core periphery that may allow a stream of neutrons to escape from the core into exterior experimental facilities.

The SAFARI-1 reactor operates in cycles typically lasting from 21 to 30 days, with 5-day shutdown periods in between. Usually, between two and four fresh fuel assemblies are loaded into the core during each cycle; the remaining fuel assemblies are reused from previous cycles. An individual fuel assembly, as shown in Figure 4.5(a), remains in the core for approximately seven or eight operational cycles, typically in different positions, before being discharged. The core itself consists of a 9×8 lattice, called a grid plate (see Figure 4.5(b)), which houses twenty-six LEU fuel assemblies, six control rods of fuel-follower type, seven dedicated ^{99}Mo production rig facilities, two general-purpose *isotope production rig* (IPR) facilities, as well as other core components (mostly reflectors) which are not specified here in detail.

(a) *A sample of a standard fuel assembly for SAFARI-1 [146]*

(b) *Top view of the grid plate in the SAFARI-1 core [146]*

FIGURE 4.5: *A sample of a standard fuel assembly for SAFARI-1 and a top view of the core grid plate.*

The core layout of the reactor, depicting all these components, is presented in Figure 4.6. It corresponds to a top view of the three-dimensional SAFARI-1 model used in the OSCAR-4 system. The locations of the beam tubes and the silicon doping facility are also indicated in Figure 4.6. These locations are, however, for indentification and tally purposes only — the tubes and the facility are not explicitly modelled in the OSCAR-4 model of SAFARI-1.

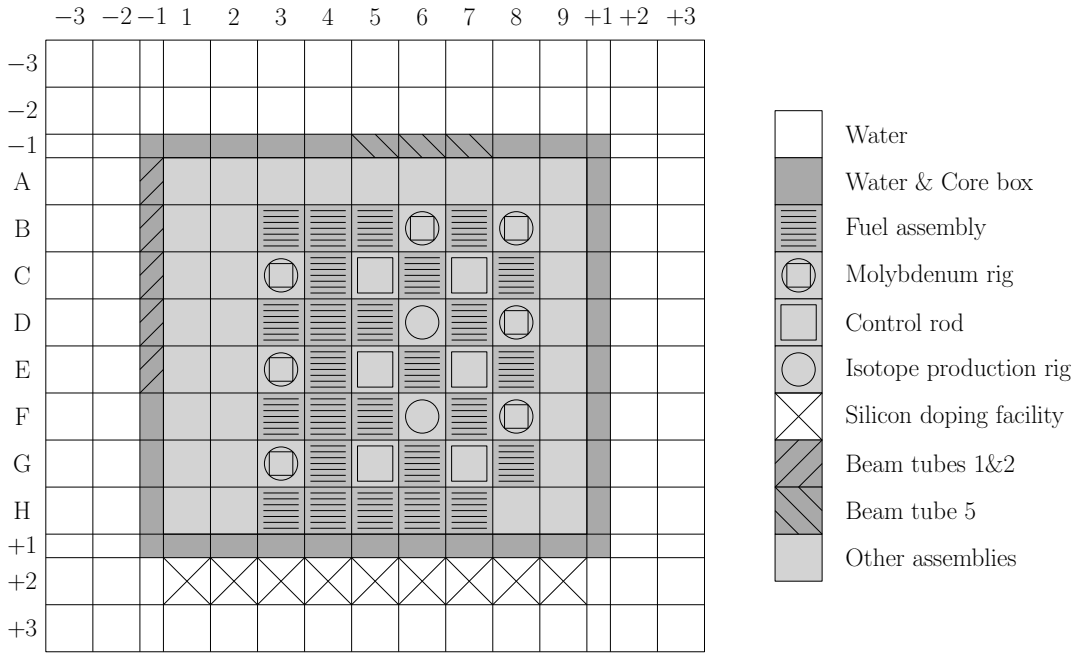


FIGURE 4.6: *Top view of the core layout of the SAFARI-1 model used in OSCAR-4.*

4.5.1 Objectives for MICFMO of SAFARI-1

A total of eight typical objectives associated with the SAFARI-1 reactor are considered for MICFMO and are presented next. The translation of these objectives into core parameters that may be returned by the OSCAR-4 system are also described.

The first objective is the popular maximisation of the operational cycle length of the reactor. The core parameter adopted as proxy for cycle length is the excess reactivity, ρ_{ex}, of the core. In the second objective, the power peaking factor of the reactor, which is a safety-related core parameter, is to be minimised.

The commercially-driven third and fourth objectives are to maximise the total production of ^{99}Mo and to maximise the utilisation of the silicon transmutation doping facility, respectively. The ^{99}Mo production objective may be translated to maximising the assembly-averaged power levels in all the molybdenum rigs (see Figure 4.6) because of a relation between the power level and the molybdenum yield. The silicon doping objective, on the other hand, may be translated to maximising the average thermal neutron flux level over the silicon doping facility (see Figure 4.6). This translation is based on the assumption that a silicon ingot requires a shorter exposure time when the neutron flux is higher. More ingots can therefore be exposed over a given period of time, thus improving the utilisation of the doping facility.

Beam line research and experiments are provisioned for in the next two objectives. Although the neutron scattering, radiography and diffraction facilities are each serviced by its own beam tube, the proximity of beam tubes 1 and 2 result in these tubes being considered together, whereas beam tube 5 is considered separately. Accordingly, the fifth and sixth objectives are to maximise the research capability at beam tubes 1 & 2 and beam tube 5 (see Figure 4.6). Since the beam line structures are not explicitly modelled in OSCAR-4, the two objectives may be translated to maximising the average thermal neutron flux over the beam tube faces at the core periphery. This translation is based on the assumption that more thermal neutrons yield improved experimental outcomes.

Finally, the seventh and eighth objectives are to maximise the production of other isotopes, *i.e.* those produced in the two IPR facilities. The two objectives may be translated to maximising the maximum axial thermal neutron flux level in the IPR facilities at positions D6 and F6 (see Figure 4.6), respectively. For this translation, it is assumed that an irradiation sample may be placed in an IPR at the axial position where the thermal flux is at its peak.

The aforementioned objectives for the SAFARI-1 reactor and the core parameters to which they are translated are summarised in Table 4.1 for ease of reference. Given the prevalence of thermal neutron flux in the objectives discussed above, it is mentioned here, for the sake of interest, that the thermal neutron energy range adopted in the OSCAR-4 model for SAFARI-1 is from 0 to 0.625 eV.

4.5.2 Constraints associated with MICFMO of SAFARI-1

The objectives in the previous section are all subject to a constraint set consisting of several safety and utilisation requirements for the SAFARI-1 reactor. The specific limiting values of these requirements are proprietary knowledge and are therefore not divulged here (see §1.4 for details). The set of inequality constraints is as follows:

- The total ^{99}Mo production must be greater than a specified limit;
- The ^{99}Mo yield for each molybdenum rig must be above a specified limit;

- The peak axial production capability in each IPR facility must be above a specified limit;
- The core power peaking factor must be below the safety limit;
- The total control bank worth, ρ_{cbw}, must be above the safety limit; and
- The shutdown margin, ρ_{sdm}, must be above the safety limit.

These constraints are translated to core parameters in the same manner as in the objectives. Accordingly, ^{99}Mo production/yield in the first two constraints translate to assembly-averaged power levels in the molybdenum rigs, while the peak axial production capability in an IPR facility in the third constraint translates to the maximum axial thermal neutron flux level in the facility. The remaining constraints already correspond to core parameters.

Identifier	Goal	Original objective	Core parameter
S1	Maximise	Cycle length	Excess reactivity (ρ_{ex})
S2	Minimise	Power peaking factor	Power peaking factor
S3	Maximise	Total ^{99}Mo production	Assembly-averaged power levels in all molybdenum rigs
S4	Maximise	Utilisation of the silicon doping facility	Average thermal neutron flux over the silicon doping facility
S5	Maximise	Research capability at beam tubes 1 & 2	Average thermal neutron flux over the beam tube faces
S6	Maximise	Research capability at beam tube 5	Average thermal neutron flux over the beam tube face
S7	Maximise	Isotope production in the first IPR facility	Maximum axial thermal neutron flux in the IPR facility
S8	Maximise	Isotope production in the second IPR facility	Maximum axial thermal neutron flux in the IPR facility

TABLE 4.1: *The objectives for MICFMO of SAFARI-1 and their corresponding core parameters.*

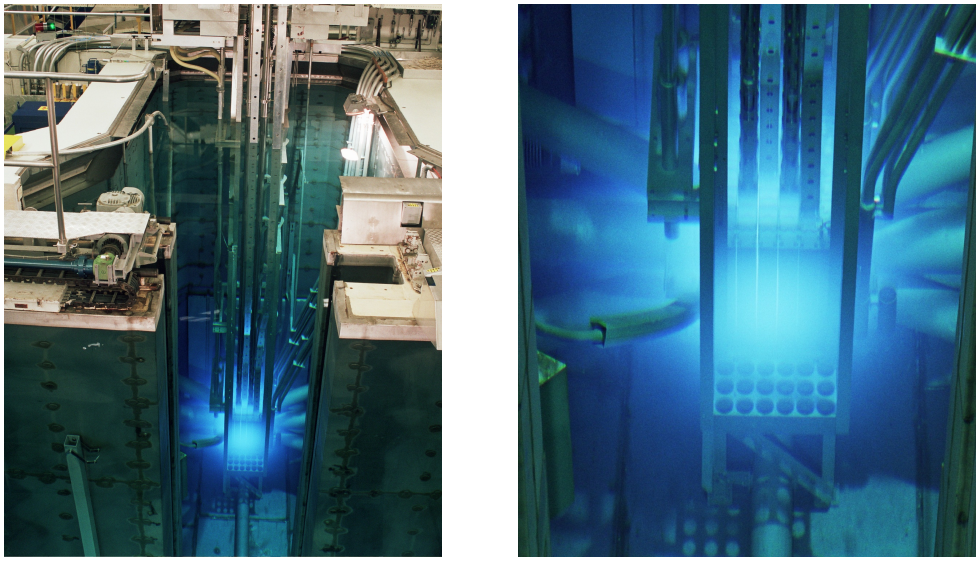
4.5.3 The current reload configuration design approach at SAFARI-1

The SAFARI-1 reactor is currently operated according to a reload configuration design approach based on a combination of several years of operating experience and the well-known *highest-mass to lowest-flux* (HMLF) heuristic. This heuristic attempts to flatten the power distribution over the reactor core by assigning more reactive fuel assemblies (highest-mass) to the less reactive loading positions (lowest-flux), and *vice versa*. Generally speaking, this means that fresh fuel assemblies are loaded in the peripheral positions of the SAFARI-1 core whereas the most-burnt assemblies are loaded in the central positions.

This design approach typically yields a safe reload configuration for any given operational cycle, while generally meeting the utilisation requirements of the reactor, especially in terms of ^{99}Mo production. A disadvantage of the approach, however, is its inflexibility to any change in reactor utilisation requirements. It is likely also the least favourable approach in terms of cycle length.

4.6 The HOR nuclear research reactor

The *Hoger Onderwijs Reactor* (HOR) is a 2 MW open-pool type research reactor located at the Delft University of Technology in the Netherlands. Photographs of the reactor are presented in Figure 4.7. The HOR reactor is operated by the Reactor Institute Delft and is primarily utilised as a source of neutrons and positrons[9] in support of the various research activities conducted at the university. These research activities find application in studies of biotissues, metals, colloids, polymers and thin layers, and geomaterials [20]. The OSCAR code system is also used to provide calculation support to the reactor.



(a) (b)

FIGURE 4.7: *The HOR nuclear research reactor [19].*

The duration of an operating cycle for HOR may vary, although it generally operates for two or three cycles per annum. Only one fresh fuel assembly is usually loaded into the core during a cycle. The reactor core consists of a 6×7 lattice which houses sixteen LEU fuel assemblies, four intra-assembly control rods, two in-core irradiation rigs, and several beryllium (Be) reflector assemblies. A number of neutron beam tubes and irradiation facilities also surround the core.

The core layout of the reactor, depicting these components, is presented in Figure 4.8. It corresponds to a top view of the three-dimensional HOR model used in OSCAR-4. Only the location of the primary beam tube, designated as R2, is indicated in Figure 4.8. As before, this location is for identification and tally purposes only — the beam tube is not explicitly modelled in the OSCAR-4 HOR model. The two in-core irradiation rig facilities are colloquially referred to as "Small BeBe" and "Big BeBe" in reference to their beryllium composition.

4.6.1 Objectives for MICFMO of HOR

A total of four typical objectives associated with the HOR reactor are considered for MICFMO and are presented next. As before, the translation of these objectives into core parameters that may be returned by the OSCAR-4 system are also described.

[9] A *positron* is a subatomic particle that has the same mass and magnitude of charge as an electron, but is positively charged. It is also sometimes called a *positive electron* and constitutes the antiparticle of an electron [53].

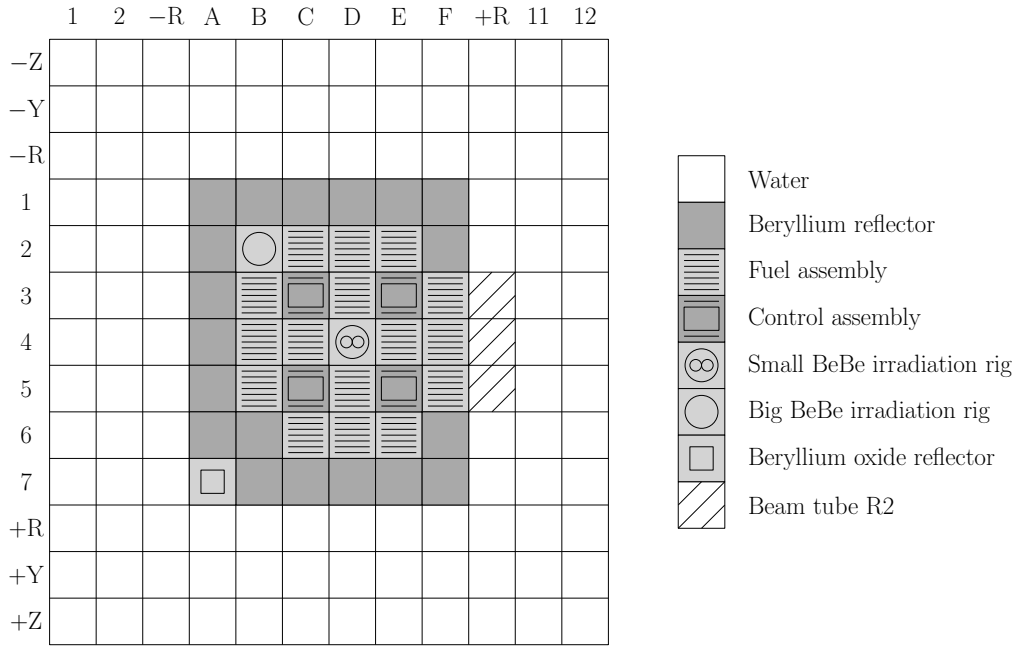


FIGURE 4.8: *Top view of the core layout of the HOR model used in OSCAR-4.*

For the HOR reactor, the first objective is also to maximise its operational cycle length which may again be translated to maximising the excess reactivity of the core. The second objective is to maximise the research capability at beam tube R2. Under the assumption that more thermal neutrons yield improved experimental outcomes, this objective may again be translated to maximising the average thermal neutron flux over the beam tube face (since the beam line structure is not explicitly modelled in OSCAR-4). The third and fourth objectives are to maximise the utilisation of the Small BeBe and Big BeBe irradiation rigs, respectively. These two objectives may be translated to maximising the average thermal neutron flux in the respective rigs. It is also assumed here that more thermal neutrons yield improved experimental outcomes. Finally, the fifth objective is to minimise the criticality of the reactor when the two most reactive control rods are fully extracted from the core. This safety-related objective is associated with the so-called *stop margin* requirement which is discussed in greater detail later in this section. The objective may be translated to minimising the effective multiplication factor of the reactor $k_{\text{eff}}^{\text{sm}}$ when the two most reactive control rods are fully extracted from the core.

The aforementioned objectives for the HOR reactor and the core parameters to which they are translated are summarised in Table 4.2 for ease of reference. As before, the thermal neutron energy range adopted in OSCAR-4 model for HOR is from 0 to 0.625 eV.

4.6.2 Constraints associated with MICFMO of HOR

The objectives in the previous section are all subject to a safety-related constraint associated with the HOR reactor. This constraint is referred to as the *stop margin* constraint and it is a slightly modified shutdown margin requirement for the reactor. According to this requirement, the reactor should remain subcritical when the two most reactive control rods are fully extracted from the core. In the OSCAR-4 system, this constraint is implemented as follows. A reactor core state is deemed subcritical (*i.e.* it satisfies the constraint) if its calculated effective multiplication factor, k_{eff}, is less than the estimated k_{eff} value of a critical core state by some prescribed safety margin. The value of this "critical k_{eff}" may be different from 1 since any given code system and model combination will exhibit a computational offset.

Identifier	Goal	Original objective	Translated core parameter
H1	Maximise	Cycle length	Excess reactivity (ρ_{ex})
H2	Maximise	Research capability at beam tube R2	Average thermal neutron flux over the beam tube face
H3	Maximise	Utilisation of the Small BeBe rig	Average thermal neutron flux in the Small BeBe rig
H4	Maximise	Utilisation of the Big BeBe rig	Average thermal neutron flux in the Big BeBe rig
H5	Minimise	Criticality of the reactor during stop margin requirement	$k_{\text{eff}}^{\text{sm}}$

TABLE 4.2: *The objectives for MICFMO of HOR and their corresponding core parameters.*

4.6.3 The current reload configuration design approach at HOR

The HOR reactor is currently operated according to a reload configuration design approach based on a combination of operating experience and trial-and-error. The design approach aims to satisfy the stop margin constraint while maximising the cycle length. In this regard, the heaviest-massed fuel assemblies, in terms of ^{235}U content, are typically assigned to the central positions in the core so as to increase the excess reactivity. The remaining fuel assemblies are then distributed approximately symmetrically about row 4 (see Figure 4.8) according to their ^{235}U masses. This is an initial attempt at satisfying the stop margin constraint. Modifications are then made according to a trial-and-error approach until a satisfactory reload configuration design is obtained.

4.7 Chapter summary

In this chapter, a general model for the MICFMO problem was presented along with detailed discussions on the two nuclear research reactors considered in this dissertation as case studies. In §4.1, the topic of nuclear fuel management was briefly described in order to illustrate the context of ICFMO within this broader topic. Several necessary problem assumptions were then presented in §4.2 before the optimisation model adopted in this dissertation for the MICFMO problem was presented in §4.3.

A short description of the OSCAR-4 code system was given in §4.4 since the objective functions and constraints in the MICFMO problem have to be calculated by such a system. The SAFARI-1 and HOR reactors were then described in some detail in §4.5 and §4.6, respectively. Typical objectives and constraints associated with each reactor for the MICFMO problem were specified, and descriptions of the current reload configuration design approaches employed at the reactors were presented.

CHAPTER 5

A scalarisation-based methodology for MICFMO

Contents

As mentioned in §1.3, one of the priorities in this dissertation is to address the shortcomings present in the optimisation methodology within the core calculation code system, OSCAR-4, as it was inherited by this author. A scalarisation-based methodology for MICFMO is therefore proposed in this chapter to address those shortcomings. First, the notion of Pareto optimality and other related concepts are discussed in some detail. This supplies the reader with the necessary background knowledge to better understand the MOO modelling process and solution techniques discussed in this chapter, and later in the dissertation. The scalarisation approach adopted in the inherited methodology is described next, followed by an elucidation of the proposed MICFMO methodology. The applicability of this methodology is finally demonstrated on problems instances within the context of the SAFARI-1 and HOR reactors introduced in Chapter 4.

5.1 Introduction

The notion of Pareto optimality in MOO was only touched upon very briefly in the dissertation thus far. In this section, a more comprehensive discussion on Pareto optimality and other related concepts are presented. This allows the reader to gain a better understanding of the MOO modelling approaches and solution techniques employed throughout this dissertation. The definitions and descriptions presented in this section follow closely those of Miettinen [139].

5.1.1 Multiobjective optimisation preliminaries

In order to present formal definitions of Pareto optimality and its related concepts, consider first a general model for a *multiobjective optimisation problem* (MOP) with $q \geq 2$ objective functions $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})$. Assume, without loss of generality, that all objective functions are to be maximised. Then an MOP may be formulated as

$$\left.\begin{array}{rl} \text{maximise} & [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}, \end{array}\right\} \tag{5.1}$$

where $\boldsymbol{x} = [x_1, \ldots, x_n]$ denotes the decision vector and $\mathcal{S}$ the *feasible region*, which is here assumed to be a subset of the *decision space* $\mathbb{R}^n$. Let $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})]$ denote the *objective vector*. For the purpose of brevity, the objective vector may also be denoted by $\boldsymbol{z} = [z_1, z_2, \ldots, z_q]$, where $z_i = f_i(\boldsymbol{x})$ for all $i = 1, \ldots, q$. These two notations are used interchangeably in the discussions that follow. Lastly, the image of the feasible region, $\mathcal{Z} = \{\boldsymbol{f}(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{S}\}$, is called the *feasible objective space* and is a subset of the *objective space* $\mathbb{R}^q$.

Although a *single-objective optimisation problem* (SOP) may have a single solution (not necessarily unique) that maximises/minimises its objective function, it is reasonable to assume that no single solution exists that is optimal with respect to every objective function simultaneously in an MOP [139]. A set of compromise or trade-off solutions may, however, be identified for the MOP (5.1) by employing a concept known as *Pareto dominance*. Formally, a decision vector $\boldsymbol{x}^* \in \mathcal{S}$ is said to *dominate* another decision vector $\boldsymbol{x} \in \mathcal{S}$ (denoted by $\boldsymbol{x}^* \succ \boldsymbol{x}$) if $f_i(\boldsymbol{x}^*) \geq f_i(\boldsymbol{x})$ for all $i = 1, \ldots, q$ and $f_j(\boldsymbol{x}^*) > f_j(\boldsymbol{x})$ for at least one $j \in \{1, \ldots, q\}$. Furthermore, an objective vector $\boldsymbol{z}^*$ dominates another objective vector $\boldsymbol{z}$ (*i.e.* $\boldsymbol{z}^* \succ \boldsymbol{z}$) if its corresponding decision vector $\boldsymbol{x}^*$ dominates $\boldsymbol{x}$.

Using this definition of dominance, a decision vector $\boldsymbol{x}^*$ in some subset $\mathcal{Q} \subseteq \mathcal{S}$ is said to be *nondominated in* $\mathcal{Q}$ if there exists no other decision vector $\boldsymbol{x} \in \mathcal{Q}$ which dominates $\boldsymbol{x}^*$. Similarly, an objective vector $\boldsymbol{z}^*$ is *nondominated* if its corresponding decision vector is nondominated. The special case in which the subset $\mathcal{Q}$ is, in fact, the entire feasible region $\mathcal{S}$ itself corresponds to a definition of Pareto optimality — a decision vector $\boldsymbol{x}^* \in \mathcal{S}$ is *Pareto optimal* if it is nondominated in $\mathcal{S}$. Also, an objective vector $\boldsymbol{z}^*$ is Pareto optimal if its corresponding decision vector is Pareto optimal. The set containing all the Pareto optimal decision vectors of (5.1) is called the *Pareto set*, and is denoted by $\mathcal{P}_S$. Similarly, the corresponding set of all Pareto optimal objective vectors is called the *Pareto front*, and is denoted by $\mathcal{P}_F$.

If the objective functions are bounded over the feasible region, then upper bounds on the Pareto front give rise to a so-called *ideal objective vector*, denoted by $\boldsymbol{z}^\star \in \mathbb{R}^q$. The components of this vector, $z_i^\star$, are obtained by maximising each of the objective functions individually, subject to the feasible region $\mathcal{S}$. Sometimes, for theoretical reasons, a vector is needed that is strictly better than the ideal objective vector. Accordingly, a *utopian objective vector*, denoted by $\boldsymbol{z}^{\star\star} \in \mathbb{R}^q$, is defined as an infeasible objective vector whose components are determined by $z_i^{\star\star} = z_i^\star + \epsilon$ for all $i = 1, \ldots, q$, where $\epsilon > 0$ is a relatively small but computationally significant scalar [139].

Apart from Pareto optimality, there are also the concepts of weak and proper Pareto optimality. Formally, a decision vector $\boldsymbol{x}^* \in \mathcal{S}$ is *weakly Pareto optimal* if there does not exist another decision vector $\boldsymbol{x} \in \mathcal{S}$ such that $f_i(\boldsymbol{x}) > f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, q$, *i.e.* for which *all* the components are better than those of $\boldsymbol{x}$. Furthermore, an objective vector is weakly Pareto optimal if its corresponding decision vector is weakly Pareto optimal. The idea behind the notion of proper Pareto optimality is that unbounded trade-offs between objectives are not allowed. Although several definitions exist for proper Pareto optimality, the one by Geoffrion [67], as described in [139], is adopted here. A decision vector $\boldsymbol{x}^* \in \mathcal{S}$ is *properly Pareto optimal (in the sense of Geoffrion)* if it is Pareto optimal and if there is some real number $M > 0$ such that, for each $i = 1, \ldots, q$ and each $\boldsymbol{x} \in \mathcal{S}$ satisfying $f_i(\boldsymbol{x}) > f_i(\boldsymbol{x}^*)$, there exists at least one $j \in \{1, \ldots, q\}$ such that $f_j(\boldsymbol{x}^*) > f_j(\boldsymbol{x})$ and

$$\frac{f_i(\boldsymbol{x}^*) - f_i(\boldsymbol{x})}{f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}^*)} \leq M.$$

Essentially then, a decision vector is properly Pareto optimal if there exists at least one pair of objectives for which a finite improvement in one objective is possible only at the expense of some reasonable deterioration in the other objective [139]. Furthermore, an objective vector is properly Pareto optimal if its corresponding decision vector is properly Pareto optimal.

These three concepts of optimality are illustrated by means of a graphical example in Figure 5.1. The example is similar to the one found in [22]. In the figure, the set of weakly Pareto optimal objective vectors is denoted by a thick black line on the edge of the feasible objective space. The endpoints of the set of Pareto optimal vectors are denoted by two circles, while the endpoints of the set of properly Pareto optimal vectors are denoted by short lines.

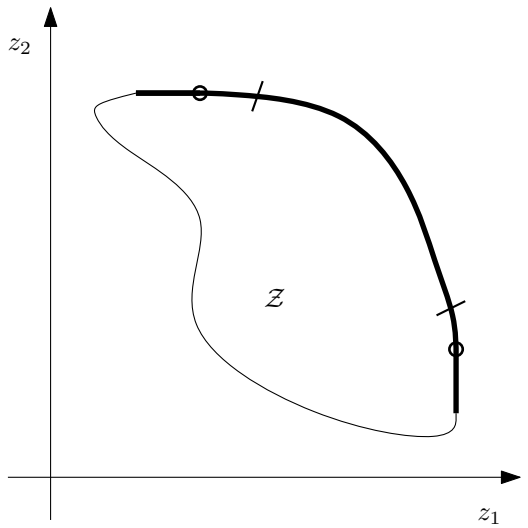


FIGURE 5.1: *Example of Pareto optimality, weak Pareto optimality and proper Pareto optimality. The set of weakly Pareto optimal objective vectors is denoted by a thick black line. The endpoints of the set of Pareto optimal vectors are denoted by two circles, while the endpoints of the set of properly Pareto optimal vectors are denoted by short lines.*

5.1.2 Classification of MOO solution methods

Since Pareto optimal solutions to an MOP are regarded as equally desirable in a mathematical sense, additional information not contained in the objective functions is typically required in order to select one solution as the final solution for implementation purposes [139]. This selection

requires a *decision maker*, which is understood to be a person who can express preference relations between different Pareto optimal solutions. Methods for solving MOPs may be classified according to the role that the decision maker assumes during the solution process. The following four classes of solution methods have been proposed by Hwang and Masud [87] according to Miettinen [139], and a short description of each is provided.

- *No-preference methods* — a decision maker, along with his preference information is not available, and the task is to find some neutral compromise solution based on reasonable assumptions.
- *A priori methods* — a decision maker specifies his preference information, and the task is to find a Pareto optimal solution that satisfies it as best as possible.
- *A posteriori methods* — a representation of the set of Pareto optimal solutions is determined, and then a decision maker chooses his most preferred solution.
- *Interactive methods* — an iterative solution procedure is developed in which solution information and decision maker preferences are exchanged after each iteration, until a solution is reached that satisfes a decision maker the most.

In the context of the MICFMO problem, it is assumed that a decision maker is available and so no-preference methods are therefore excluded from consideration. Interactive methods are also excluded for the sake of simplicity. Accordingly, only *a priori* and *a posteriori* methods are considered in this dissertation.

A popular approach for solving MOPs is by *scalarisation*. This approach is adopted in both the *a priori* and *a posteriori* classes of methods. In scalarisation, an MOP is converted into a single, or a family of SOPs [139]. This is usually achieved by aggregating the multiple objectives into a single scalarising objective function. Once an MOP is scalarised, methods in single-objective optimisation may be employed to solve the problem.

5.2 The weighting method

Recall from §1.3 that the author inherited a basic working version of an MICFMO decision support feature in the OSCAR-4 system [182]. In this inherited feature, a problem instance is solved by means of a scalarisation approach known as the *weighting method*. This well-known method is the most popular scalarisation approach adopted in the ICFMO literature [8, 9, 91, 94, 97, 123, 164, 166, 182, 191].

According to the weighting method, each of the q objective functions in an MOP is associated with a weighting coefficient and the linear weighted sum of these objective functions then form a single scalarising objective function to be optimised. More formally, an MOP of the form (5.1) is transformed into a *weighting problem*

$$\left.\begin{aligned} \text{maximise} \quad & \sum_{i=1}^{q} w_i f_i(\boldsymbol{x}), \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}, \end{aligned}\right\} \tag{5.2}$$

where $w_i \geq 0$ for all $i = 1, \ldots, q$ and generally $\sum_{i=1}^{q} w_i = 1$ [139]. If all the weighting coefficients are strictly positive, a solution of (5.2) is Pareto optimal. A serious weakness of the method,

however, is that it cannot uncover all the Pareto optimal solutions if the optimisation problem is nonconvex[1] [139].

If the weighting method is employed as an *a posteriori* method, which it is generally intended for, a set of Pareto optimal solutions can be obtained through repeated solution of (5.2) using different combinations of strictly positive weighting coefficients [139]. Since none of the above-mentioned works in the ICFMO literature employ the method in this manner, the MICFMO problem has not yet been solved in terms of finding a set of Pareto optimal solutions.

The weighting method may also be employed as an *a priori* method, as is done in the MICFMO literature and the inherited feature within OSCAR-4. The decision maker has to specify a weighting vector that represents his preference information, while the scalarising objective function in (5.2) is considered as the decision maker's *utility function*[2] [139]. It is often said that the coefficients reflect to the relative importance of the objective functions. What exactly underlies this notion, however, is not clear at all [139]. It has also been remarked that the coefficients "*should represent the rate at which the decision maker is willing to trade off values of the objective function*" [139]. It is also known that unexpected results may be obtained when weighting coefficients are modified, especially when objective functions are correlated, or when their objective values are not scaled/normalised [139]. Weighting coefficients in such an *a priori* context are therefore not easy to interpret or understand. This is also evident based on the values selected for the weighting coefficients in the ICFMO literature. For example, coefficient values in [8, 91, 123, 182] were arbitrarily selected as 1, while the values in [9, 94, 164, 166] were determined "empirically" without any mention as to methodology involved.

Although the weighting method is simple to use, it is arguably not an appropriate scalarisation approach for MOPs. The method's general inability to obtain all Pareto optimal solutions, the subsequent risk of generating misleading results in respect of the Pareto set, the misleading role of weighting coefficients, and the unexpected results from changes in these coefficients all contribute to the widely accepted recommendation in the operations research community that a different scalarisation approach should be employed for solving MOPs.

5.3 The proposed scalarisation-based methodology for MICFMO

Given the shortcomings of the weighting method, the MICFMO feature in the OSCAR-4 system inherited by the author was recognised to be inadequate for rendering advanced MICFMO decision support to users of the system. As mentioned in §1.3, one of the priorities in this dissertation is therefore to address the shortcomings present in the inherited optimisation methodology. Accordingly, an alternative scalarisation approach is proposed in this section with a view to replace the weighting method. Furthermore, adaptations to the harmony search algorithm, which is employed as the solution technique in the methodology, are also proposed in aid of rendering advanced decision support.

[1] An optimisation problem is nonconvex if any of its objective functions or the feasible region is not convex. As per the definition presented in [139], a function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if, for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$, it holds that $f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) \leq \lambda f(\boldsymbol{c}_1) + (1-\lambda) f(\boldsymbol{x}_2)$ for all $0 \leq \lambda \leq 1$. Similarly, a set $\mathcal{S} \in \mathbb{R}^n$ is *convex* if, for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$, it holds that $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \mathcal{S}$ for all $0 \leq \lambda \leq 1$.

[2] A *utility function* is a function $U : \mathbb{R}^q \to \mathbb{R}$ representing the preferences of a decision maker among the objective vectors by which a complete ordering in objective space may be achieved [139].

5.3.1 An alternative scalarisation approach

The alternative scalarisation approach proposed in this dissertation is based on the *method of weighted metrics* and the inclusion of *aspiration levels* during the solution process. According to the method of weighted metrics, the weighted distance between some reference point and the feasible objective space is minimised [139]. An aspiration level is, in turn, defined as an objective function value that is desirable or would be satisfactory to the decision maker if that value were to be achieved [139].

In the method of weighted metrics, the L_p-metrics are usually employed for measuring distance and the ideal objective vector $\boldsymbol{z}^\star$ is selected as reference point [139]. More formally, an MOP of the form (5.1) is transformed into the *weighted L_p-problem*

$$\left.\begin{aligned} \text{minimise} \quad & \left(\sum_{i=1}^{q} w_i |f_i(\boldsymbol{x}) - z_i^\star|^p\right)^{1/p}, \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}, \end{aligned}\right\} \tag{5.3}$$

for $1 \leq p < \infty$. It is usually assumed that $w_i \geq 0$ for all $i = 1, \ldots, q$ and that $\sum_{i=1}^{q} w_i = 1$ [139]. In the case where $p = \infty$, the metric is also known as a *Chebyshev metric*, thus giving rise to the *weighted Chebyshev problem*

$$\left.\begin{aligned} \text{minimise} \quad & \max_{i=1,\ldots,q} \left(w_i |f_i(\boldsymbol{x}) - z_i^\star|\right), \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}. \end{aligned}\right\} \tag{5.4}$$

Denominators may be added to (5.3) and (5.4) so as to normalise/scale the terms in the scalarising function, *e.g.* $|f_i(\boldsymbol{x}) - z_i^\star|/|z_i^\star|$.

In (5.3), the method of weighted metrics exhibits the same weakness as the weighting method for $w_i > 0$ for all $i = 1, \ldots, q$, namely that it cannot uncover all the Pareto optimal solutions if the problem is nonconvex. On the other hand, in (5.4), every Pareto optimal solution can be obtained by some combination of strictly positive weighting coefficients $w_1, \ldots, w_q$, irrespective of whether or not the problem is convex [139]. Although this is a very desirable theoretical result, it still has a weakness — unless a solution is unique, the method may yield weakly Pareto optimal solutions as well. One of the approaches specified by Miettinen [139] for overcoming this weakness is to vary the Chebyshev metric by an augmentation term such that weakly Pareto optimal solutions are avoided. Some Pareto optimal solutions may, however, be impossible to obtain using this approach and therefore only properly Pareto optimal solutions are of interest [139]. Accordingly, the problem to consider, now using the utopian objective vector $\boldsymbol{z}^{\star\star}$ as reference point, is the *augmented weighted Chebyshev problem*

$$\left.\begin{aligned} \text{minimise} \quad & \max_{i=1,\ldots,q} \left(w_i |f_i(\boldsymbol{x}) - z_i^{\star\star}|\right) + \mu \sum_{j=1}^{q} |f_j(\boldsymbol{x}) - z_j^{\star\star}|, \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}, \end{aligned}\right\} \tag{5.5}$$

where μ is a "sufficiently small" positive scalar [139]. It is known that a solution to (5.5) is properly Pareto optimal (if $w_i > 0$ for all $i = 1, \ldots, q$ and $\mu > 0$) and any properly Pareto optimal can be obtained for convex and nonconvex problems [139].

One of the difficulties that may arise in the scalarisation of the MICFMO problem (4.1) to an augmented weighted Chebyshev problem, is the likely absence of the ideal objective vector. For many real-world optimisation problems, including the MICFMO problem, the ideal objective

vector is typically not known. This, in turn, means that the required utopian objective vector in (5.5) cannot be determined.

Consider also that the method of weighted metrics is intended to be used as an *a posteriori* method [139]. A set of properly Pareto optimal solutions can therefore be obtained through repeated solution of (5.5) by using different (strictly positive) weighting coefficients. This leads to another difficulty that may arise in the context of MICFMO. Since it has been assumed in §4.2.6 that only a limited computational budget is allowed for solving an MICFMO problem instance, it would be impractical from a computational point of view to solve (5.5) repeatedly using different weighting coefficients.

Taking cognisance of the augmented weighted Chebyshev problem's desirable theoretical result and of its associated practical difficulties within the context of MICFMO, the following scalarisation approach is proposed. The decision maker (or an analyst) has to specify partial *a priori* preference information in the form of aspiration levels, denoted by $\bar{z}_i$, for every objective $i \in \{1, \ldots, q\}$. The vector $\bar{\boldsymbol{z}} = [\bar{z}_1, \ldots, \bar{z}_q] \in \mathbb{R}^q$ formed by these levels is referred to as an *aspiration vector*. Note that achievement of these levels does not necessarily have to be feasible. This aspiration vector is then used as reference point within an augmented Chebyshev metric-based scalarising objective function $\tilde{F}_q(\boldsymbol{x})$, given as

$$\tilde{F}_q(\boldsymbol{x}) = \underbrace{\max_{i=1,\ldots,q} \left| \frac{f_i(\boldsymbol{x}) - \bar{z}_i}{\bar{z}_i} \right|}_{\text{Chebyshev term}} + \underbrace{\mu \sum_{j=1}^{q} \left| \frac{f_j(\boldsymbol{x}) - \bar{z}_j}{\bar{z}_j} \right|}_{\text{augmentation term}}, \tag{5.6}$$

where all symbols have the same meaning as before. The denominators in (5.6) ensure that all the terms are similarly scaled by order of magnitude.

The idea behind using this scalarising objective function, as is the case in problem (5.5), is to minimise the Chebyshev distance between the objective vector $\boldsymbol{f}(\boldsymbol{x})$ and a reference point, now selected as the aspiration vector $\bar{\boldsymbol{z}}$. Minimisation of (5.6) therefore improves, at any given time, the worst deviation between any objective and its aspiration level. Usage of the Chebyshev metric typically results in solutions that are "well-balanced" with respect to all the objective functions [205]. If the worst deviation cannot be improved upon anymore for some specific decision vector, it may still be possible to improve achievements in the other objectives. These improvements may be realised through the inclusion of the augmentation term in (5.6). The value of μ should, however, be sufficiently small so as to ensure that the augmentation term does not negate the Chebyshev term.

Regarding the aspiration vector, known target or goal values (based, for example, on historical data or expert judgement) may be taken as the aspiration levels. This implies that objectives need only be *improved up to sufficient satisfaction*, and not necessarily to optimality. Another approach would be to take unattainable, but still realistic, values as the aspiration levels. In doing so, objectives are to be *improved as best possible*, and not only up to their target values. It is important to recognise that, for an appropriate choice of aspiration vector, namely $\bar{\boldsymbol{z}} = \boldsymbol{z}^{\star\star}$, (5.6) corresponds to the scalarising objective function in (5.5) with $w_i = 1$ for all $i = 1, \ldots, q$. As such, the function has the property that its solution (*i.e.* when minimised) is properly Pareto optimal irrespective of whether or not the problem is convex. Accordingly, usage of (5.6) addresses the shortcoming of the weighting method employed in the current MICFMO literature and within the inherited OSCAR-4 methodology.

Unlike the method of weighted metrics, it is proposed that (5.6) be employed partially within the context of an *a priori* method because of the limited computational budget assumed for MICFMO. This is also the primary motivation for not having weighting coefficients associated

with any of the objective functions in (5.6) — it would be computationally impractical to optimise the scalarising function repeatedly using different weighting coefficients. Furthermore, the misleading role that weighting coefficients may play in an *a priori* method is circumvented when using (5.6). The specification of aspiration levels is arguably a more natural, or easily understandable, manner to convey preferences. In the proposed methodology, however, the aspiration levels correspond only to partial preferences because it is recognised that the single solution obtained from minimising (5.6) may not necessarily be to the satisfaction of the decision maker. Instead, the intention within the methodology is to identify several solutions, preferably also (properly) Pareto optimal. As such, it is proposed that an archive of nondominated decision vectors be maintained during the optimisation of (5.6) as and when they are identified. Upon completion of the solution process, nondominated decision vectors in the vicinity of the final solution may have been uncovered. It is therefore possible to construct a set of nondominated solutions for an MOP without repeatedly having to solve an SOP (as would be the case for typical *a posteriori* scalarisation). Such a nondominated set may be considered an approximation of the Pareto set, although it will not necessarily exhibit a good spread over the true Pareto set due to the usage of a fixed scalarising objective function and the choice of $\bar{\boldsymbol{z}}$. It will, however, provide the decision maker with good alternative trade-off solutions to consider.

5.3.2 Constraint handling

By using (5.6) to transform MICFMO problem (4.1) into an SOP, the following optimisation problem is obtained

$$\left.\begin{aligned} \text{minimise} \quad & \tilde{F}_q(\boldsymbol{x}) = \max_{k=1,\ldots,q} \left| \frac{f_k(\boldsymbol{x}) - \bar{z}_k}{\bar{z}_k} \right| + \mu \sum_{\ell=1}^{q} \left| \frac{f_\ell(\boldsymbol{x}) - \bar{z}_\ell}{\bar{z}_\ell} \right|, \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq g_i^{\lim}, \qquad i = 1, \ldots, r, \\ & h_j(\boldsymbol{x}) = h_j^{\lim}, \qquad j = 1, \ldots, s, \\ & \boldsymbol{x} \in \mathcal{X}, \end{aligned}\right\} \tag{5.7}$$

where all symbols have the same meaning as before. In order to solve an optimisation problem such as (5.7), a suitable constraint handling technique is required for accommodating the constraints. A popular approach, which is also adopted here, is to consider all decision vectors during optimisation, but to penalise those that violate any constraint. It is therefore proposed that an *additive penalty function* (APF) be employed as constraint handling technique within the methodology.

According to this APF technique, if a decision vector violates any constraint, a corresponding penalty value related to the magnitude of that violation is incurred. The total penalty value for all the constraint violations is then calculated and added to scalarising objective function $\tilde{F}_q(\boldsymbol{x})$ in (5.7) in order to penalise the decision vector for being infeasible. Note that a feasible decision vector has zero penalty.

Let

$$G(\boldsymbol{x}) = \sum_{i=1}^{r} \max \left\{ 0, \frac{g_i(\boldsymbol{x}) - g_i^{\lim}}{|g_i^{\lim}|} \right\} \tag{5.8}$$

be the total scaled constraint violation associated with the inequality constraints in problem (5.7). Similarly, let

$$H(\boldsymbol{x}) = \sum_{j=1}^{s} \left| \frac{h_j(\boldsymbol{x}) - h_j^{\lim}}{h_j^{\lim}} \right| \tag{5.9}$$

be the total scaled constraint violation associated with the equality constraints in problem (5.7). The penalty function in the APF technique is then defined as

$$P_a(\boldsymbol{x}) = \gamma \big(G(\boldsymbol{x}) + H(\boldsymbol{x})\big), \tag{5.10}$$

where γ is a strictly positive severity factor. The manner in which γ may be determined empirically is discussed later in this section.

All the constraint violations in (5.8) and (5.9) are assumed to be equally important, thereby eliminating the need to specify additional importance weights for the different constraints. This is a reasonable assumption if feasible solutions only are of interest. Furthermore, each constraint violation is scaled in order to obtain values that are similarly scaled by order of magnitude. This allows for the various constraint violations to be aggregated together in a responsible manner.

The penalty function (5.10) may now be added to $\tilde{F}_q(\boldsymbol{x})$ in (5.7) so that the scalarising objective function for MICFMO becomes

$$F_q(\boldsymbol{x}) = \max_{k=1,\ldots,q} \left| \frac{f_k(\boldsymbol{x}) - \bar{z}_k}{\bar{z}_k} \right| + \mu \sum_{\ell=1}^{q} \left| \frac{f_\ell(\boldsymbol{x}) - \bar{z}_\ell}{\bar{z}_\ell} \right| + P_a(\boldsymbol{x}). \tag{5.11}$$

The value of the parameter γ in (5.10) may be chosen so that the value of $F_q(\boldsymbol{x})$ for an infeasible decision vector is worse than the value of $F_q(\boldsymbol{x})$ for the majority of feasible decision vectors. By repeatedly minimising $F_q(\boldsymbol{x})$ off-line for different values of γ, and evaluating the quality of the solutions thus obtained, an acceptable value may be settled upon and selected for use during the actual optimisation.

Therefore, in the proposed scalarisation-based methodology for MICFMO, the general constrained MICFMO problem (4.1) may always be transformed into an SOP, namely

$$\left.\begin{array}{ll} \text{minimise} & F_q(\boldsymbol{x}), \\ \text{subject to} & \boldsymbol{x} \in \mathcal{X}, \end{array}\right\} \tag{5.12}$$

where $F_q(\boldsymbol{x})$ is defined in (5.11). Accordingly, single-objectove optimisation methods may be employed to solve (5.12).

It may be observed that the scalarising objective function in (5.11) is not only applicable to the MICFMO problem, but also to the single-objective ICFMO problem. The penalty function $P_a(\boldsymbol{x})$ in (5.10) is already independent of the number of objectives. For $q = 1$, the max-operator reduces to the identity operator. Furthermore, the augmentation term simply has a scaling effect on the objective function value and, as such, has no influence on the optimisation process. Therefore, if an unattainable aspiration level is selected, then solving (5.12) for $q = 1$ will yield an optimal solution.

Another advantage of this scalarisation approach, then, is that SOP and MOP variants of the ICFMO problem, be they constrained or unconstrained, may be modelled in the exact same manner, while the same single-objective optimisation method(s) may be utilised for solving any resulting problem instance. Such flexibility may be especially useful for research reactors that have to pursue different objectives during different operational cycles.

5.3.3 Harmony search as a solution technique

It was mentioned earlier that a metaheuristic, called *harmony search* (HS) [64], is employed as the solution technique in the MICFMO feature within the OSCAR-4 system inherited by

the author [182]. Apart from its usage in [182], the HS algorithm has also been applied in the context of ICFMO by Poursalehi *et al.* [165]. The algorithm was found to yield competitive results when compared to a Hopfield neural network/SA hybrid method and a GA.

Recall from §3.3.3 that HS is a constructive metaheuristic based on an analogy to an improvisation jazz band playing different sounds on their instruments (different values for decision variables) in order to find a pleasing combination of sound (good objective function values). In the analogy, a *harmony* corresponds to a decision vector and new harmonies are said to be *improvised*.

The standard HS algorithm consists of the following four steps:

1. Initialise a memory structure, called the *harmony memory* (HM), with random[3] harmonies.
2. Improvise a new harmony on a variable-by-variable basis according to guidelines that probabilistically consider the HM, local perturbations and pure randomisation.
3. Compare the newly-improvised harmony with the worst harmony contained in the HM, in terms of objective function value. If the new harmony is better than the worst one, replace that worst harmony in the HM with the new one.
4. If some pre-determined termination criteria are met, terminate the algorithm; otherwise, return to the second step.

The size of the HM, denoted by N_{hm}, is generally problem-dependent and its value may be determined empirically. The guidelines mentioned in the second step on how to improvise a new harmony (*i.e.* decision vector) are described next by means of an example. Consider an SOP in which objective function $f(\boldsymbol{x})$ is to be minimised, subject to the constraints $x_i \in \mathcal{X}_i$ for $i = 1, \ldots, 4$, where $\boldsymbol{x} = [x_1, x_2, x_3, x_4]$ is a vector of continuous decision variables, and $\mathcal{X}_i = \{x \,|\, 0 \leq x \leq 1, x \in \mathbb{R}\}$ is the set of allowable values for variable i. Furthermore, suppose $N_{\text{hm}} = 3$ and that the HM at the start of some iteration during the algorithm's execution is given by

$$\text{HM} = \begin{array}{c|cccc|c} & x_1 & x_2 & x_3 & x_4 & f(\boldsymbol{x}) \\ \hline \boldsymbol{x}_a & 0.12 & 0.35 & 0.01 & 0.16 & 8.9 \\ \boldsymbol{x}_b & 0.55 & 0.45 & 0.80 & 0.93 & 3.2 \\ \boldsymbol{x}_c & 0.76 & 0.67 & 0.43 & 0.50 & 6.5 \end{array} .$$

In the second step, a new harmony $\boldsymbol{x}'$ is improvised on a variable-by-variable basis. For each variable x_i, a value is randomly selected, either from the HM, or from its allowable range $\mathcal{X}_i$. The *harmony memory consideration rate* (HMCR), denoted by p_{hm}, is a parameter that guides this selection towards values from the HM. Consider, for example, the decision variable x'_2. With probability p_{hm}, a value for x'_2 is randomly selected from the list $\{0.35, 0.45, 0.67\}$. The values in this list correspond to those appearing in the shaded column x_2 of the HM. The list may contain repeated values and, since each element in the list is equally likely to be selected, any repeated values will therefore have a greater likelihood of selection. Alternately, with probability $1 - p_{\text{hm}}$, a value for x'_2 is randomly selected from $\mathcal{X}_2$. According to Yang [240], typical values for the HMCR range between 0.7 and 0.95.

Whenever a decision variable takes a value from the HM, a process called *pitch adjustment* may be performed. In this process, the value of the decision variable is perturbed in order

[3] In this section, where any reference is made to *random* selection, it is assumed that a uniform distribution is employed during selection.

to generate a slightly different harmony. The probability of performing the pitch adjustment process is determined by a parameter called the *pitch adjustment rate* (PAR), and is denoted by p_{par}. Values for the PAR, according to Yang [240], typically range between 0.1 and 0.5. So, consider a decision variable $x'_i \in \mathbb{R}$ whose value has been taken from the HM. With probability p_{par}, that value may be perturbed according to $x'_{i,\text{new}} = x'_{i,\text{old}} + b_{\text{par}} \cdot \varepsilon$, where b_{par} is the so-called *bandwidth parameter*, and ε is a random number in the range $[-1, 1]$. The bandwith parameter determines the size of the perturbation. If $x'_i \in \mathbb{Z}$, then ε is a random number from the set $\{-1, 0, +1\}$. Alternately, if the value of x'_i may only be perturbed to another value within some discrete set of neighbouring values, then the pitch adjustment process randomly selects one of those neighbouring values. In such a case, the bandwith parameter determines the number of neighbouring values to consider in this selection. The value of b_{par} is therefore problem-dependent.

In Figure 5.2, a flow diagram is presented which illustrates the working of the HS algorithm, as described above.

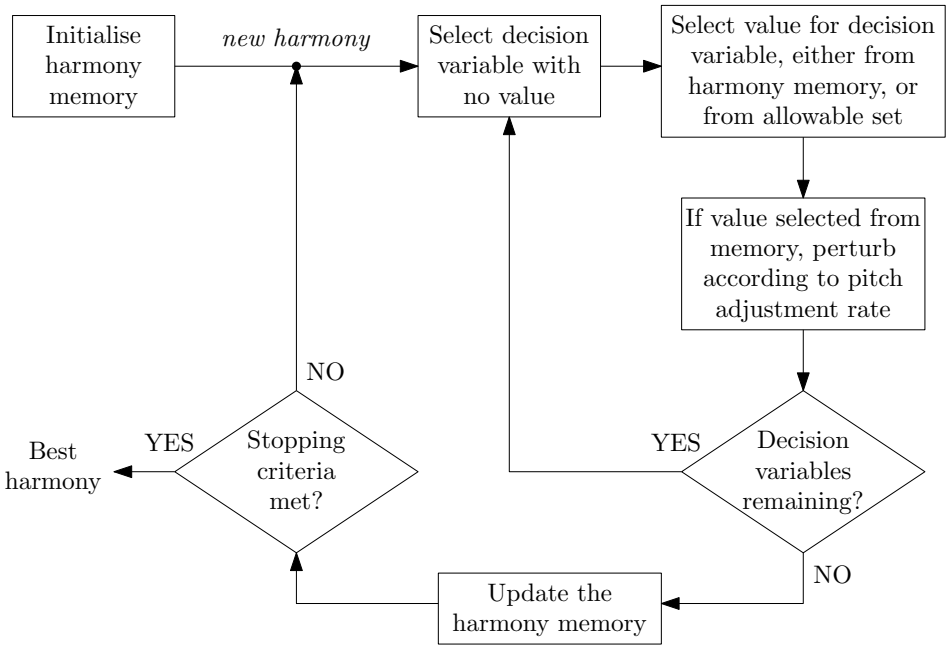


FIGURE 5.2: *Flow diagram of the harmony search algorithm.*

In order to employ HS in the context of MICFMO, the standard algorithm described above has to be adapted. First, since a permutation-based decision vector representation is implemented in the algorithm, the second step of the HS algorithm (in which new harmonies are improvised) should explicitly preserve the structure of permutations, and do so in an unbiased manner. Secondly, feasible nondominated decision vectors have to be archived during the execution of the algorithm so that an approximate Pareto set may be returned upon its termination (as proposed in §5.3.1). The adapted HS algorithm, which has been entirely re-implemented in the OSCAR-4 feature, is presented in pseudocode form as Algorithm 5.1.

As part of ensuring that the algorithm preserves the permutation structure of decision vectors in an unbiased manner, so-called *allowable lists* have been introduced in line 8 of Algorithm 5.1. The allowable memory consideration list $\mathcal{M}$ contains those fuel assembly labels in the HM that correspond to selected position i, excluding those labels that have already been selected in other positions of the partially constructed decision vector $\boldsymbol{x}'$. Similarly, the allowable random list $\mathcal{R}$ contains all the fuel assembly labels, excluding those labels that have already been selected

Algorithm 5.1: HS algorithm in the scalarisation-based methodology for MICFMO

Input : An MICFMO problem of the form (5.12), algorithmic stopping criteria, an HM size N_{hm}, an HMCR p_{hm}, and a PAR p_{par}.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

1 Initialise the HM with N_{hm} randomly generated decision vectors, each corresponding to a random permutation of n unique fuel assembly labels
2 Initialise the feasible nondominated decision vector archive $\mathcal{A}$ with the nondominated decision vectors in the HM
3 **while** *algorithmic stopping criteria not met* **do**
4 Reset the new decision vector $\boldsymbol{x}'$
5 Reset the pitch adjusting index set $\mathcal{V} \leftarrow \emptyset$
6 **while** *new decision vector only partially created* **do**
7 Randomly select a new loading position i
8 Construct the allowable memory consideration list $\mathcal{M}$, and the allowable random list $\mathcal{R}$
9 Generate a random number $r \sim U(0,1)$
10 **if** $r \leq p_{hm}$ **then**
11 Randomly select a fuel assembly label for position x'_i from $\mathcal{M}$ if $\mathcal{M} \neq \emptyset$; otherwise, select it from $\mathcal{R}$
12 $\mathcal{V} \leftarrow \mathcal{V} \cup \{\text{loading position } i\}$
13 **else**
14 Randomly select a fuel assembly label for position x'_i from $\mathcal{R}$
15 **end if**
16 **end while**
17 Perform the pitch adjustment process using Algorithm 5.2
18 Evaluate new decision vector $\boldsymbol{x}'$ with a core simulator and determine $F_q(\boldsymbol{x}')$
19 **if** $F_q(\boldsymbol{x}') <$ *maximum* F_q *in HM* **then**
20 Replace decision vector in HM having the maximum F_q with new decision vector $\boldsymbol{x}'$
21 **end if**
22 $\mathcal{A} \leftarrow$ feasible nondominated decision vectors in $\mathcal{A} \cup \{\boldsymbol{x}'\}$
23 **end while**
24 $\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}$

in other positions of the partially constructed decision vector $\boldsymbol{x}'$. In the inherited OSCAR-4 feature, an approach similar to these lists was taken for preserving the permutation structure of decision vectors. It involved the usage of logical flags to indicate whether fuel assemblies were available or not. The explicit generation of the allowable lists is, however, arguably a simpler and less error-prone approach to follow. An example of how these lists are generated in the context of ICFMO is presented next.

Suppose $n = 4$ and that the HM (of size 4, for example) at the start of some iteration during the algorithm's execution is given by

$$\text{HM} = \begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline \boldsymbol{x}_a & 1 & 3 & 2 & 4 \\ \boldsymbol{x}_b & 1 & 2 & 4 & 3 \\ \boldsymbol{x}_c & 2 & 1 & 3 & 4 \\ \boldsymbol{x}_d & 3 & 2 & 4 & 1 \end{array} .$$

Furthermore, suppose that a new decision vector has only been constructed partially as $\boldsymbol{x}' = [4, x_2, x_3, 3]$, and that the new loading position is randomly selected as $i = 3$ (line 7 of Algorithm 5.1). Then, $\mathcal{M} = \{2,4,3,4\} \setminus \{4,3\} = \{2\}$, which corresponds to the labels in the shaded column x_3 of the HM, excluding the labels already selected in the partial solution $\boldsymbol{x}'$. Similarly, $\mathcal{R} = \{1,2,3,4\} \setminus \{4,3\} = \{1,2\}$.

An adapted pitch adjustment process is presented in pseudocode form as Algorithm 5.2. It differs significantly from the process described in the standard HS algorithm (which was employed in the inherited OSCAR-4 feature). Essentially, pitch adjustment has been redefined so that the process now performs pair-wise exchanges between variables in a fully-constructed decision vector. Therefore, pitch adjustment is no longer performed on a variable-by-variable basis during the improvisation of a new harmony.

Algorithm 5.2: Adapted pitch adjustment process

Input : A decision vector $\boldsymbol{x}'$, a pitch adjusting index set $\mathcal{V}$, and a PAR p_{par}.
Output : Pitch adjusted decision vector $\boldsymbol{x}'$.

```
1  while |V| ≥ 2 do
2  |  Randomly select loading position i ∈ V
3  |  Generate a random number r ~ U(0, 1)
4  |  if r ≤ p_par then
5  |  |  Find k_+ such that x'_{k_+} = x'_i + 1
6  |  |  Find k_- such that x'_{k_-} = x'_i - 1
7  |  |  if k_+ or k_- found then
8  |  |  |  Randomly select loading position k ∈ {k_+, k_-}
9  |  |  |  Exchange the values of x'_i and x'_k
10 |  |  end if
11 |  end if
12 |  if k ∈ V then
13 |  |  V ← V \ {i, k}
14 |  else
15 |  |  V ← V \ {i}
16 |  end if
17 end while
```

The indices of those variables that took values during harmony memory consideration are stored in a pitch adjusting index set $\mathcal{V}$. Once a decision vector has been fully constructed, the pitch adjustment process is performed. The values of the variables identified in $\mathcal{V}$ may then only be exchanged pair-wise with neighbouring values, as seen in lines 5 and 6 of Algorithm 5.2. This exchange between neighbouring values only may be interpreted as the selection of a bandwidth parameter $b_{\text{par}} = 1$. A qualitative pilot study indicated that this adapted pitch adjustment process yields, on average, improved results when compared to the standard process — hence its adoption in the adapted HS algorithm.

Finally, an archive consisting of feasible nondominated decision vectors is updated at the end of each iteration, thus giving effect to the proposal in §5.3.1. During this update, as specified in line 22 of Algorithm 5.1, the newly-improvised decision vector is considered for possible inclusion in the archive. Note that if $q = 1$ in (5.12), the archive simply contains the best feasible decision vector found thus far.

5.4 Application of the proposed methodology to SAFARI-1

The scalarisation-based methodology for MICFMO proposed in this dissertation has been implemented by the author within a completely revised version of the MICFMO decision support feature in the OSCAR-4 system. In order to demonstrate the applicability of the methodology, this feature is utilised to solve (approximately) three different ICFMO problem instances that

are based on the SAFARI-1 reactor. For benchmarking purposes, the results thus obtained are compared to the current reload configuration design approach employed at the reactor.

The first problem instance considered is an SOP variant of the ICFMO problem and is included here to confirm that the methodology may indeed be applied to both SOP and MOP variants of the problem, as claimed in §5.3.2. The second and third problem instances, however, are MOP variants of the ICFMO problem.

In addition, the inherited weighting method has also been utilised to solve the second and third (MOP) problem instances approximately. A number of arbitrary values for the weighting coefficients have been selected based on different prioritisations of the objectives so as to demonstrate their effect on the solutions. Based on the shortcomings of the method described in §5.2 (*e.g.* inability to obtain all Pareto optimal solutions, risk of misleading results in respect of the Pareto set, and unexpected results from changes in coefficients), the aforementioned effect on the solutions should be apparent. The results thus obtained are then placed into context by comparing them to the nondominated fronts obtained using the the new methodology.

5.4.1 Problem instances for the SAFARI-1 reactor

Since the SAFARI-1 reactor, along with the typical objectives and constraints associated with it, has already been described in some detail in §4.5, a full problem description is not repeated here. The three problem instances considered in this section are based on an actual SAFARI-1 operational cycle during the year 2012, which is designated as cycle C1211-1. They consist of hypothetical scenarios in which the reactor operators have to deviate from their current reload configuration design approach for the cycle because new safety and utilisation requirements are supposedly imposed on the reactor.

The actual reload configuration loaded into the core during cycle C1211-1 was designed according to the approach described in §4.5.3. This configuration will therefore be treated as a basis for comparison in the problem instances, and will hereafter be referred to as the *historical SAFARI-1 reload configuration* (HSRC). Accordingly, the twenty-six fuel assemblies to be used in the problem instances correspond to those that have been used in the HSRC.

In each of the three problem instances, the entire constraint set specified in §4.5.2 has to be adhered to. The feasible region in the decision space effected by this constraint set, along with $\boldsymbol{x} \in \mathcal{X}$, is denoted by $\mathcal{S}_S$. It is noted, however, that the first three constraints in that set correspond to newly-imposed requirements for SAFARI-1, resulting in the problem instances being potentially more difficult to solve. As a result, the HSRC no longer meets all of the new requirements, specifically that of the third constraint (*i.e.* peak axial production capability in the IPR facilities) which is now violated by approximately 9%.

In the first hypothetical scenario for SAFARI-1, the reactor is required to operate during cycle C1211-1 for an extended period of time beyond the typical cycle length. This scenario therefore conforms to the pursuit of objective S1 (see Table 4.1), namely the maximisation of the reactor's cycle length, using excess reactivity as proxy. Accordingly, the first problem instance is an SOP in which the goal is to

$$\left.\begin{array}{ll} \text{maximise} & f_{\text{S1}}(\boldsymbol{x}), \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{5.13}$$

In the second hypothetical scenario, the reactor is required to enhance its beam line research and experimental utilisation during the cycle. Of interest, then, are objective S5 and S6 (see Table 4.1) in which the research capability at beam tubes 1 & 2 and beam tube 5, respectively,

are to be maximised. The second problem instance, therefore, is a bi-objective MOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & [f_{\text{S5}}(\boldsymbol{x}), f_{\text{S6}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{5.14}$$

Finally, in the third hypothetical scenario for SAFARI-1, the reactor has to respond to market fluctuations by optimising the commercial services that it renders during cycle C1211-1. Accordingly, the optimisation of objectives S3, S4, S7 and S8 (see Table 4.1) are pursued, namely the production of ^{99}Mo isotopes, the utilisation of the silicon doping facility, and the production of isotopes in the IPR facilities. The third problem instance, therefore, is a tetra-objective MOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & [f_{\text{S3}}(\boldsymbol{x}), f_{\text{S4}}(\boldsymbol{x}), f_{\text{S7}}(\boldsymbol{x}), f_{\text{S8}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{5.15}$$

In §4.5, it was mentioned that a typical shutdown and reload period for the SAFARI-1 reactor lasts five days. It is therefore assumed that an acceptable computational run time for solving an ICFMO problem instance should not exceed three days.

5.4.2 Experimental design

All the calculations in this demonstration were performed on a personal computer with the following specifications: An Intel® Core™ i5-2500 CPU with 4 GB RAM operating at 3.30 GHz within a 32-bit operating system. The evaluation of a single reload configuration in OSCAR-4 requires approximately four minutes on such a personal computer, given the objectives and constraints adopted in the problem instances above. The acceptable computation time of three days therefore correspond to approximately 1 000 evaluations of objective function $F_q(\boldsymbol{x})$ in (5.12).

In accordance with the proposed methodology, aspiration levels have to be specified by a decision maker for all the objectives. For the SOP instance (5.13), an unattainable aspiration level is chosen so that, as explained in §5.3.1, an optimal solution may be sought. In (5.14) and (5.15), however, realistic aspiration levels are chosen for the objectives based on previous SOP pilot study results and expert judgement. As explained in §1.4, these aspiration levels are proprietary knowledge and are therefore not divulged here. Algorithm 5.1, implemented within the OSCAR-4 feature, is then used to solve (approximately) each scalarised problem instance now having the general form (5.12).

The SAFARI-1 instances were solved multiple times, independently on the same computer, resulting in ten optimisation runs per problem instance. Different random number generator seeds were employed in each run since HS is a stochastic algorithm and may therefore yield different results when executed each time. Descriptive statistics could thus be gathered for an analysis of the optimisation results. In practice, however, the best overall result or a combination of several results may be used to render decision support.

Qualitative pilot studies were performed in order to determine reasonable values for the free parameters in Algorithm 5.1, as well as for the parameters in (5.11). The following values were settled upon for Algorithm 5.1: $N_{\text{hm}} = 15$, $p_{\text{hm}} = 0.95$ and $p_{\text{par}} = 0.25$. As motivated above, a maximum of 1 000 iterations was selected as the stopping criterion for the algorithm. The parameter values in the scalarising objective function were selected as $\mu = 0.01$ and $\gamma = 2.5$.

For the application of the inherited weighting method to the MICFMO problem instances, the objectives are scaled according to their aspiration levels. Different weighting coefficients, corresponding to different prioritisations of the objectives, are then employed during each optimisation run. These prioritisations are expressed in terms of importance ratios, *e.g.* if the first objective is twice as important as the second objectives, the ratio is 2:1. This example then has corresponding weights of $w_1 = 0.67$ and $w_1 = 0.33$. Problem instance (5.14) is solved three times, with the prioritisations 1:1, 1:3 and 3:1, respectively. Similarly, problem instance (5.15) is solved four times, with prioritisation 1:1:1:1, 1:1:2:2, 1:3:1:1 and 3:1:1:1, respectively.

5.4.3 Numerical results

The optimisation results obtained by following the experimental design discussed in §5.4.2 are presented in this section. Note that all the values in these results have been scaled according to the percentage improvement in objective value(s) over that of the HSRC.

Problem instance (5.13): Cycle length

The best feasible solution obtained for problem instance (5.13), across the ten independent optimisation runs, yielded an improvement of 29.5% in the value of objective S1 (*i.e.* excess reactivity) over that of the HSRC. In addition, the average value over the ten runs for the best improvement was 28.5%, with a relatively small absolute standard deviation of 0.4%. The convergence graph of the average best-found objective value over the optimisation runs is presented in Figure 5.3, along with the corresponding standard deviation band.

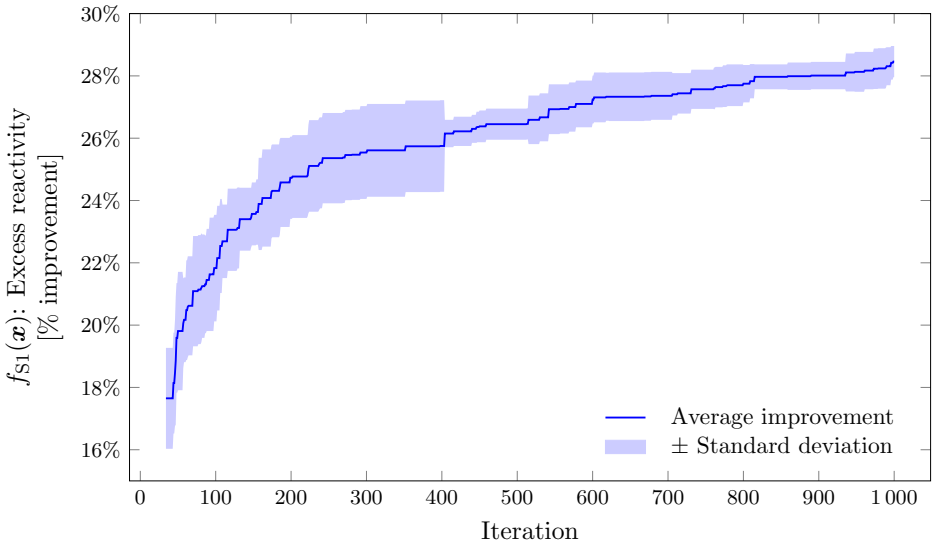


FIGURE 5.3: *Convergence graph of the average feasible best-found objective value (i.e. excess reactivity) for problem instance* (5.13).

As observed in Figure 5.3, the HS algorithm sharply improves the objective value within the first quarter of iterations, followed by a more gradual improvement up to the final iteration. Furthermore, the narrowness of the standard deviation band in the latter half of iterations confirms that the different optimisation runs yield results of similar quality. This demonstrates the robustness of the algorithm in the sense that it is fairly insensitive to different starting conditions.

In Figure 5.4, the reload configurations of the HSRC and the best-found solution for problem instance (5.13) are presented in terms of the ^{235}U mass in each fuel assembly. The best-found configuration seems to mimic a type of *in-out* loading which is consistent with what one might expect from a core in which the cycle length is maximised. In contrast, the HSRC configuration mimics more of an *out-in* loading which is essentially the opposite of the best-found configuration. This explains why such significant improvements were already achieved at commencement of optimisation — the HMCR is poorly designed in terms of cycle length maximisation. The results of this problem instance, therefore, show that the proposed methodology is able to suggest good approximate solutions for a problem in which the solutions depart significantly from the current SAFARI-1 design approach.

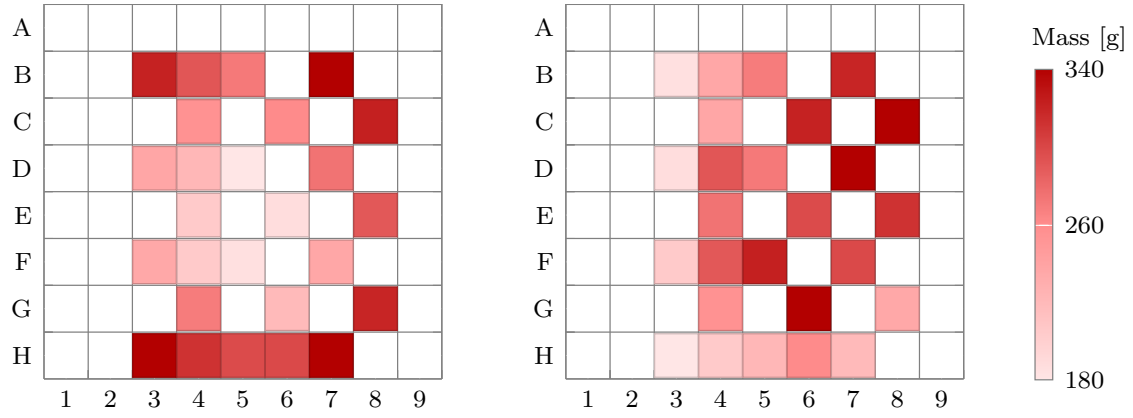


FIGURE 5.4: *The historical SAFARI-1 reload configuration (left) and the configuration of the best-found solution for problem instance* (5.13) *(right).*

Problem instance (5.14): Research utilisation

The final feasible nondominated fronts obtained by each of the ten optimisation runs for problem instance (5.14) have been pooled together and they are presented in Figure 5.5, along with the HSRC and the inherited weighting method results. By further isolating the nondominated solutions from this pool, an approximate Pareto set may be determined, *i.e.* the "overall" nondominated set. The nondominated front corresponding to this approximate Pareto set is hereafter referred to as an *attainment front*, which is also presented in Figure 5.5.

It is observed that the attainment front yields an improvement of between 2.5% and 6% in the value of objective S5 (*i.e.* beam tubes 1 & 2) over that of the HSRC, at the cost of a deterioration between of 3% and 12% in the value of objective S6 (*i.e.* beam tube 5). Although no feasible solutions were obtained that simultaneously improves upon the HSRC in both objectives, the reader is reminded that the HSRC violates the third constraint in these problem instances, *i.e.* it is infeasible in relation to the approximate Pareto set. The proposed methodology is therefore able to yield a good set of trade-off solutions which meets all the newly-imposed constraints. This affords a SAFARI-1 operator with flexibility in his decision making. Since the true Pareto front is not known for this problem instance, no verdict on the shape or closeness to optimality of the attainment front can be delivered. The methodology does, however, succeed in suggesting feasible reload configurations that improve upon the HSRC at some trade-off cost. It is also observed in Figure 5.5 that the inherited weighting method yields high-quality solutions that enhance the attainment front. In this instance, the different combinations of weighting coefficients yielded an even spread of the solutions in objective space, slightly shifted towards the one extreme of the attainment front obtained.

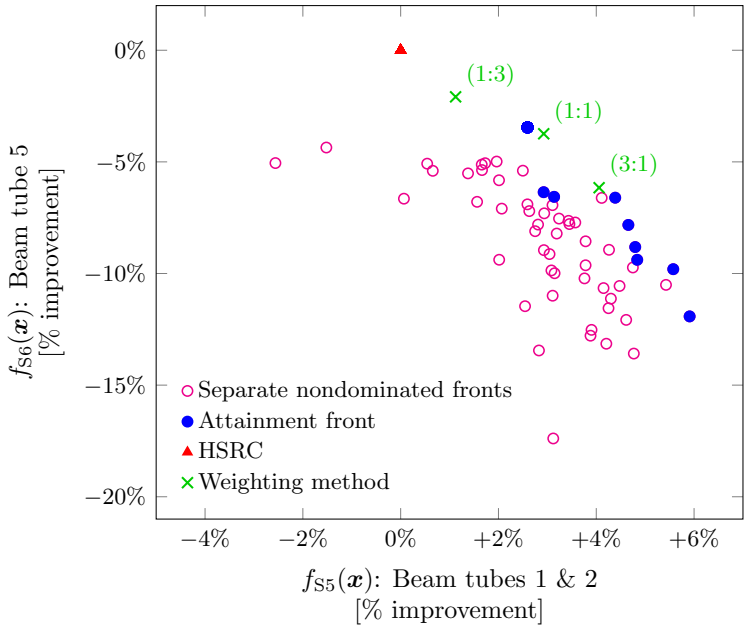


FIGURE 5.5: *Nondominated fronts obtained by ten optimisation runs for problem instance* (5.14).

In order to demonstrate the effect of the third constraint on the quality of solutions, the attainment front above is compared to a modifed front which has been determined as follows. The same results yielded by the ten optimisation runs are used. Decision vectors which also violate the third constraint up to the same level as the HSRC are, however, now also considered. The attainment and modified fronts are presented in Figure 5.6, along with the HSRC.

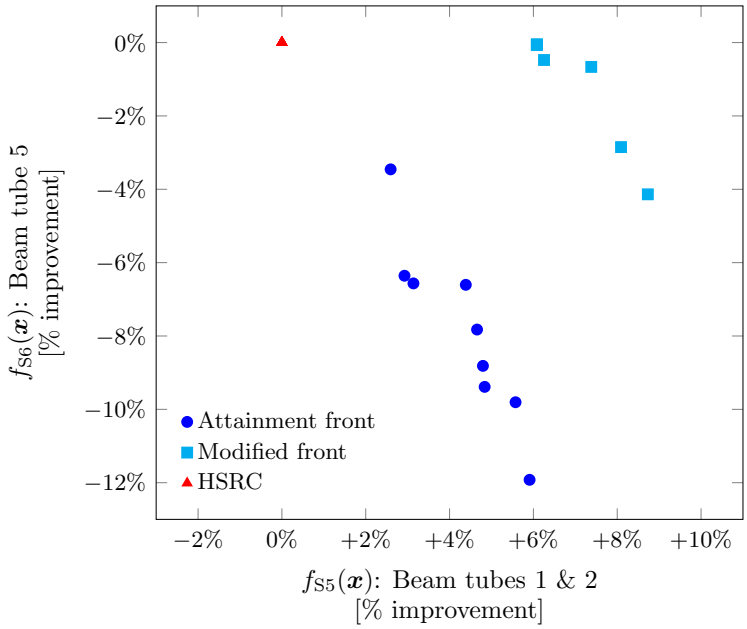


FIGURE 5.6: *Different attainment fronts for problem instance* (5.14) *depending in constraint limits.*

It is observed in Figure 5.6 that much better performance is achievable when the third constraint has a lower limiting value. It is therefore expected that optimisation of problem instance (5.14) using such a lower limiting value would yield results even better than those of the modified front.

Problem instance (5.15): Commercial services

An approximate Pareto set and corresponding attainment front was determined from the results of all ten optimisation runs for problem instance (5.15) in the same manner as before. This front is not represented visually because of the four-dimensional nature of the results. Instead, a so-called *payoff table* [139] is constructed so as to present an approximation of the available ranges found in each objective. A pay-off table is constructed by isolating each solution in the approximate Pareto set which yields the best performance in each objective. The objective vectors corresponding to these solutions then form the rows of the pay-off table. For problem instance (5.15), such a pay-off table is presented in Table 5.1. The columns of the table correspond to the different objectives, whereas the bold-faced values represent the best performance found for each objective.

Objective with best performance	Percentage improvement obtained in: $f_{S3}(\boldsymbol{x})$	$f_{S4}(\boldsymbol{x})$	$f_{S7}(\boldsymbol{x})$	$f_{S8}(\boldsymbol{x})$
$f_{S3}(\boldsymbol{x})$	**−2.7%**	−2.2%	6.1%	10.4%
$f_{S4}(\boldsymbol{x})$	−6.3%	**18.1%**	−0.2%	14.4%
$f_{S7}(\boldsymbol{x})$	−4.5%	−11.0%	**13.0%**	14.0%
$f_{S8}(\boldsymbol{x})$	−6.3%	4.6%	7.6%	**19.9%**

TABLE 5.1: *Payoff table for problem instance* (5.15). *The bold-faced entries correspond to the best performance found for each objective.*

It is observed in Table 5.1 that significant improvements may be achieved in objectives S4, S7 and S8 over that of the HSRC. The performance of objective S4 (*i.e.* silicon doping) exhibits the largest available range, with values between −11% and 18.1%. The available range in objective S7 (*i.e.* first IPR facility) lies between −0.2% and 13%, whereas objective S8 (*i.e.* second IPR facility) has an available range between 10.4% and 19.9%. These two results indicate that moderate improvements in both IPR facility objectives are almost always possible within the approximate Pareto set.

The best solution found in terms of objective S3 (*i.e.* ^{99}Mo) yields a performance deterioration of 2.7% from the HSRC. The range of available values in this objective lies between −6.3% and −2.7%. Any solution within the approximate Pareto set, therefore, sacrifices performance in objective S3 with respect to the HSRC. The current reload configuration design approach at SAFARI-1 is, however, largely geared towards maximisation of objective S3. As a result, newly-imposed constraints and/or the optimisation of other objectives simultaneously would likely always deteriorate this objective. The trade-off in the performance of objective S3 may, however, be acceptable to a decision maker given that its range of available values is much smaller than the ranges of improvements within objectives S4, S7 and S8. The proposed methodology was again able to suggest a good set of feasible trade-off solutions in response to a change in the utilisation requirements of the reactor.

Apart from the pay-off table, the percentage improvement in objective function values corresponding to the best solution found (with respect to F_q) in each of the ten optimisation runs for problem instance (5.15) are presented in Table 5.2. It is observed that the best solution found during each run achieves similar objective values, namely improvements of approximately 8%, 6% and 17% in objectives S4, S7 and S8, respectively, at the cost of a deterioration in objective S3 by approximately 6%. This, again, demonstrates the robustness of the algorithm in the sense of being fairly insensitive to different starting conditions.

Run	$f_{S3}(\boldsymbol{x})$	$f_{S4}(\boldsymbol{x})$	$f_{S7}(\boldsymbol{x})$	$f_{S8}(\boldsymbol{x})$	$F_q(\boldsymbol{x})$
1	−6.3%	7.9%	5.6%	16.7%	0.15852
2	−6.0%	9.0%	6.2%	17.6%	0.15376
3	−6.0%	8.4%	6.6%	16.8%	0.15394
4	−6.0%	8.4%	6.8%	17.5%	0.15361
5	−6.3%	8.7%	5.5%	17.4%	0.15961
6	−6.3%	8.2%	5.8%	16.8%	0.15759
7	−6.3%	10.2%	6.0%	17.0%	0.15527
8	−6.0%	8.7%	7.1%	16.9%	0.15162
9	**−6.3%**	**9.3%**	**7.1%**	**17.8%**	**0.14646**
10	−6.3%	8.3%	5.3%	16.0%	0.16097

TABLE 5.2: *Percentage improvement in objective function values corresponding to the best solution found (with respect to F_q) in each of the ten optimisation runs for problem instance* (5.15). *The row of bold-faced values represents the overall best performance.*

The results obtained using the inherited weighting method for solving problem instance (5.15) are presented in Table 5.3. It is observed that the weighting method yielded a poorly-balanced solution during optimisation run 1, where the objectives had equal importance, as compared to the solutions obtained by means of the newly proposed methodology (see Table 5.2). During optimisation runs 2 and 3, the different weighting coefficients resulted in noticable improvements in each objective having a higher priority. An improvement over the best solution found by the proposed methodology for objective S4 was even obtained during the third run. It is observed, however, that the weighting method yielded a counter-intuitive result during run 4. Although the priority for objective S3 was three times higher than the other objectives, the percentage improvement in objective function values corresponding to the solution are virtually the same as the those of the solution obtained in run 1, where all objectives had equal importance.

Run	Importance ratio	$f_{S3}(\boldsymbol{x})$	$f_{S4}(\boldsymbol{x})$	$f_{S7}(\boldsymbol{x})$	$f_{S8}(\boldsymbol{x})$
1	1:1:1:1	−5.95%	16.48%	2.95%	15.13%
2	1:1:2:2	−5.65%	3.52%	9.58%	17.92%
3	1:3:1:1	−6.25%	19.90%	0.16%	14.33%
4	3:1:1:1	−5.36%	16.98%	1.67%	14.39%

TABLE 5.3: *Percentage improvement in objective function values corresponding to the solutions obtained using the inherited weighting method for solving problem instance* (5.15).

As may be seen in Table 5.2, the overall best performance (with respect to F_q) using the proposed methodology was obtained during optimisation run 9. In Figure 5.7, the reload configurations of the HSRC and this overall best solution for problem instance (5.15) are presented in terms of the ^{235}U mass in each fuel assembly. It is observed that heavier-massed assemblies are assigned to positions near the IPR facilities (D6 and F6) within the best-found configuration, as opposed to the HSRC configuration in which they are assigned to peripheral core positions in row B and column 8. This assignment in the optimised solution, however, still occurs within the confines of the newly-imposed constraints for the problem instance.

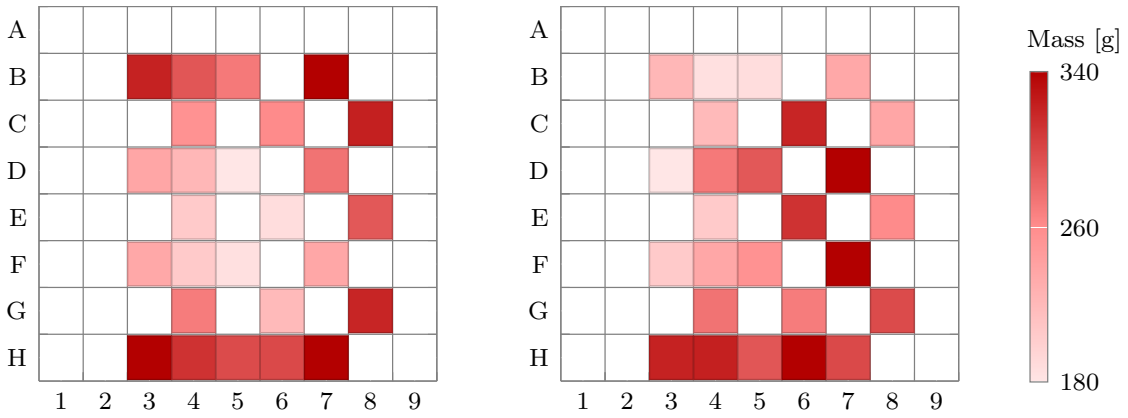


FIGURE 5.7: *The historical SAFARI-1 reload configuration (left) and the configuration of the overall best solution, with respect to F_q, for problem instance* (5.15) *(right).*

5.5 Application of the proposed methodology to HOR

This revised version of the MICFMO feature is also utilised to solve (approximately) four different ICFMO problem instances based on the HOR reactor as a further demonstration of the applicability of the proposed methodology. As before, the results thus obtained are compared to the current reload configuration design approach employed at the reactor for benchmarking purposes. Also, the first problem instance is, again, an SOP variant of the ICFMO problem, with the remaining problem instance being MOP variants.

The inherited weighting method is also utilised to solve (approximately) the three multiobjective problem instances for the HOR reactor in order to demonstrate the type of results obtainable. As before, a number of arbitrary values for the weighting coefficients have been selected based on different prioritisations for the objectives so as to demonstrate their effect on the solutions (taking cognisance of the shortcomings of the method, as described in §5.2). The results thus obtained are then finally also placed into context by comparing them to the nondominated fronts obtained using the the new methodology.

5.5.1 Problem instances for the HOR reactor

Since the HOR reactor, along with typical objectives and constraints associated with it, has already been described in some detail in §4.6, a full problem description is not repeated here. As before, the four different problem instances are based on an actual HOR operational cycle during the year 2015, which is designated as cycle C1501. The problem instances consist of hypothetical scenarios in which typical objectives associated with the HOR reactor are pursued.

The actual reload configuration loaded into the core during cycle C1501 was designed according to the approach described in §4.6.3. This configuration will, again, be treated as a basis for comparison in the problem instances, and will hereafter be referred to as the *historical HOR reload configuration* (HHRC). The sixteen fuel assemblies to be used in the problem instances therefore correspond to those that have been used in the HHRC. In each of the four problem instances, the stop margin constraint specified in §4.5.2 has to be adhered to. The feasible region in the decision space effected by this constraint, along with $\boldsymbol{x} \in \mathcal{X}$, is denoted by $\mathcal{S}_H$.

Since the current reload configuration design approach at HOR is aimed at maximising the cycle length of the reactor, the first scenario mimics this aim. Accordingly, it corresponds to

the pursuit of objective H1 (see Table 4.2) in which excess reactivity is used as a proxy for cycle length. The first problem instance, therefore, is an SOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & f_{\text{H1}}(\boldsymbol{x}), \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_H. \end{array}\right\} \tag{5.16}$$

In the second scenario, the reactor is required to enhance its beam line research in conjunction with cycle length maximisation. The pursuit of objectives H1 and H2 (see Table 4.2) are therefore of interest, noting that H2 corresponds to the maximisation of the research capability at the beam tube. As such, the second problem instance is a bi-objective MOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & [f_{\text{H1}}(\boldsymbol{x}), f_{\text{H2}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_H. \end{array}\right\} \tag{5.17}$$

In the third scenario, the utilisation of the two in-core irradiation rigs at HOR has to be optimised. Of interest, therefore, are objective H3 and H4 (see Table 4.2) which correspond to the Small BeBe and Big BeBe rigs, respectively. Accordingly, the third problem instance is a bi-objective MOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & [f_{\text{H3}}(\boldsymbol{x}), f_{\text{H4}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_H. \end{array}\right\} \tag{5.18}$$

Finally, in the fourth scenario for HOR, the reactor is required to enhance its beam line research as well as its utilisation of the central in-core irradiation rig. This corresponds to the pursuit of objectives H2 and H3 (see Table 4.2), namely the research capability at the beam tube and the utilisation of the Small BeBe rig. Therefore, the fourth and final problem instance is a bi-objective MOP in which the goal is to

$$\left.\begin{array}{rl} \text{maximise} & [f_{\text{H2}}(\boldsymbol{x}), f_{\text{H3}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_H. \end{array}\right\} \tag{5.19}$$

The typical shutdown and reload period for the HOR reactor may vary from one cycle to the next, unlike for SAFARI-1. For the sake of simplicity, however, it is also assumed that only three days of computational run time is available for solving an ICFMO problem instance.

5.5.2 Experimental design

The same personal computer, described in §5.4.2, was used to perform the calculations in this demonstration. Approximately 1 000 evaluations may again be performed during the three days of computational run time. As before, an unattainable aspiration level is chosen for the SOP instance (5.16) so that an optimal solution may be sought. Realistic aspiration levels, based on previous SOP pilot study results and expert judgement, are chosen for the objectives in problem instances (5.17)–(5.19). As explained in §1.4, these aspiration levels are proprietary knowledge and are therefore not divulged here.

Unlike for the SAFARI-1 reactor in §5.4.2, each of the HOR problem instances is solved only once using Algorithm 5.1 within the revised OSCAR-4 feature. This is motivated by the practical situation that may arise in which only one computer is available to perform optimisation calculations within the time allotted. It has been demonstrated in §5.4 that the HS algorithm is fairly

insensitive to different starting conditions; hence, the versatility of the proposed methodology is further demonstrated in this section.

The same values selected in §5.4.2 for the free parameters in Algorithm 5.1 are also selected here, namely $N_{\text{hm}} = 15$, $p_{\text{hm}} = 0.95$ and $p_{\text{par}} = 0.25$. Furthermore, as motivated above, a maximum of 1 000 iterations has been selected as the stopping criterion for the algorithm. Finally, the parameter values in the scalarising objective function are selected as $\mu = 0.05$ and $\gamma = 10\,000$.

For the application of the inherited weighting to the MICFMO problem instances, the same procedure described in §5.4.2 is followed here. Each of the instances (5.17)–(5.19) is, however, solved three times and the different objective prioritisations are 1:1, 1:3 and 3:1, respectively.

5.5.3 Numerical results

The optimisation results obtained by following the experimental design discussed in §5.5.2 are presented in this section. Again, the values in these results have been scaled according to the percentage improvement in objective value(s) over that of the HHRC.

Problem instance (5.16): Cycle length

The best feasible solution obtained for problem instance (5.16) yielded an improvement of 3.20% in the value of objective H1 (*i.e.* excess reactivity) over that of the HHRC. According to Winkelman [231], such an improvement translates to approximately 13 MWd additional duration in cycle length for the HOR reactor. In Figure 5.8, the reload configurations of the HHRC and the best-found solution for problem instance (5.16) are presented visually in terms of the ^{235}U mass in each fuel assembly.

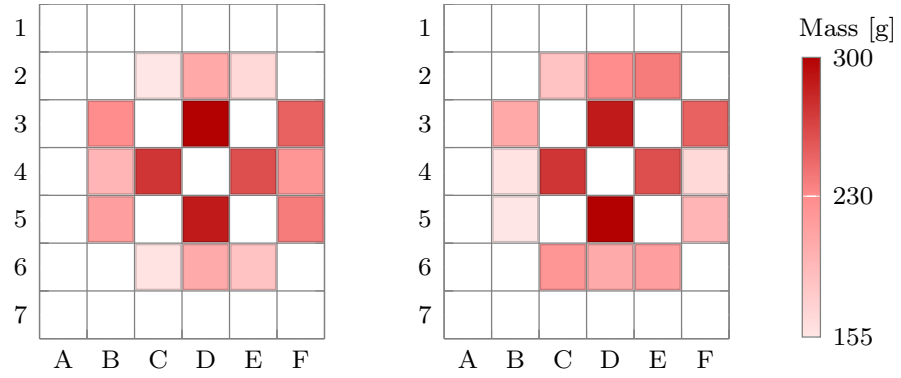


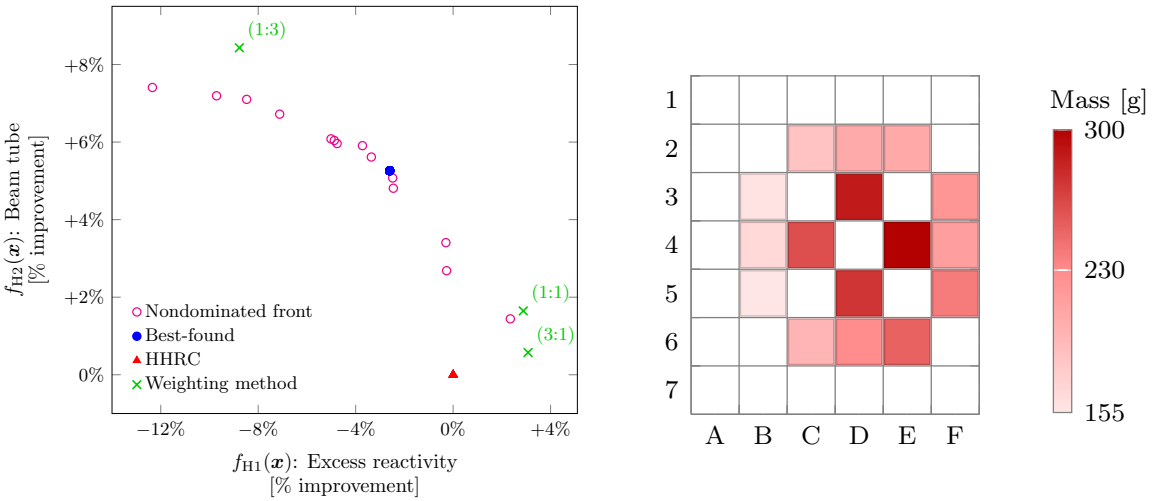
FIGURE 5.8: *The historical HOR reload configuration (left) and the configuration of the best-found solution for problem instance* (5.16) *(right).*

It is observed that the two reload configurations are very similar and mimic an *in-out* loading. This comes as no surprise given that the current reload configuration design approach at HOR is already aimed at maximising the cycle length. The results for this problem instance, therefore, show that by employing the proposed methodology, further improvement is possible for a problem in which the solutions are very similar to the current HOR design approach.

Problem instance (5.17): Cycle length and beam line research

The nondominated front obtained for problem instance (5.17) is presented in Figure 5.9(a), along with the best solution found with respect to F_q, the HHRC and the results obtained using the inherited weighting method. It is observed that a significant spread of trade-off solutions, ranging from an improvement of 7.4% in the value of objective H2 over that of the HHRC (at the cost of a 12.3% deterioration in objective H1), to simultaneous improvements of 2.3% and 1.4% in the values of objectives H1 and H2, respectively, were obtained by the proposed methodology.

The best solution found is well-balanced within the ranges of available objective values in the nondominated front, as may be observed in Figure 5.9(a). It improves the value of objective H2 (*i.e.* beam tube) by 5.3% over that of the HHRC, at the cost of a deterioration in objective H1 (*i.e.* excess reactivity) by 2.6%. Furthermore, it is observed that the inherited weighting method yielded solutions near the two extremes of the nondominated front. There is very little difference between the performances of the solutions obtained by taking a 1:1 and a 3:1 importance ratio. However, using a 1:3 ratio yielded a solution whose performance "jumped" across the objective space. This instance, therefore, is an example in which counter-intuitive behaviour is exhibited when different combinations of weights are specified in the weighting method.



(a) *Nondominated front* (b) *Best-found reload configuration*

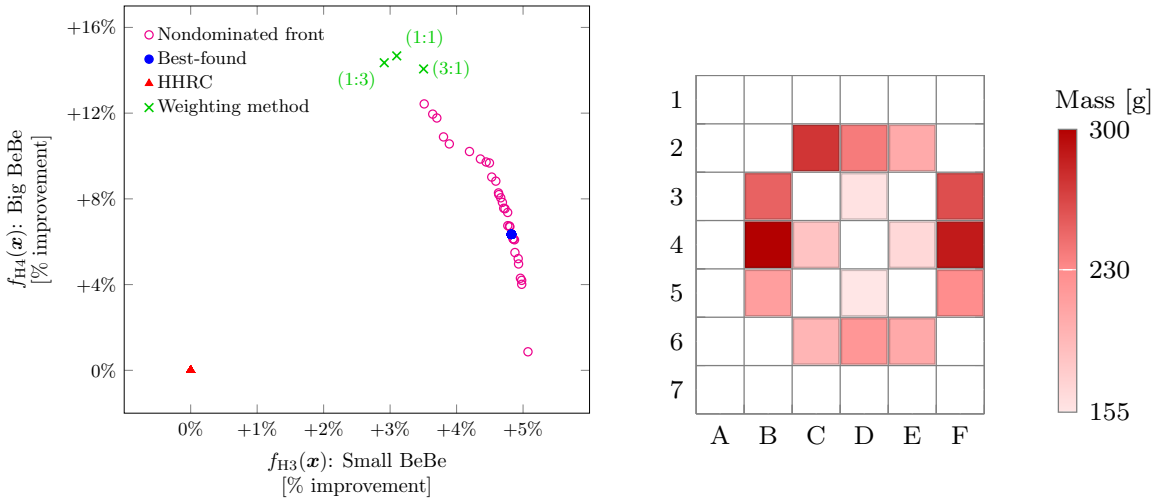
FIGURE 5.9: *Results obtained for problem instance* (5.17).

In Figure 5.9(b), the reload configuration of the best-found solution for problem instance (5.17) is presented visually in terms of the ^{235}U mass in each fuel assembly. It is observed that this configuration is similar to those in Figure 5.8, with the heaviest-massed assemblies assigned to the central core positions (as one would expect from a configuration that maximises cycle length). The remaining heavier-massed assemblies are, however, now all assigned to positions close to the beam tube, resulting in enhanced beam line research capability at the beam facility.

Problem instance (5.18): Irradiation rigs

In Figure 5.10, the nondominated front obtained for problem instance (5.18), the best solution found with respect to F_q and the HHRC are presented, along with the results obtained using the inherited weighting method. It is observed that every solution represented in the nondominated front achieved simultaneous improvements in the performance of objectives H3 and H4

(*i.e.* Small BeBe and Big BeBe rigs, respectively) over that of the HHRC. Furthermore, the best solution found is also fairly well-balanced within the nondominated front, attaining simultaneous improvements of approximately 5% in each objective. The proposed methodology, therefore, succeeds in suggesting feasible reload configurations that improve upon the HHRC in both objectives simultaneously. The inherited weighting method also yielded solutions that simultaneously improved the performance in both objectives. It is observed, however, that those improvements are clustered near one extreme of the nondominated front. The different weighting coefficients therefore caused misleading results with respect to the shape/spread of the nondominated front.



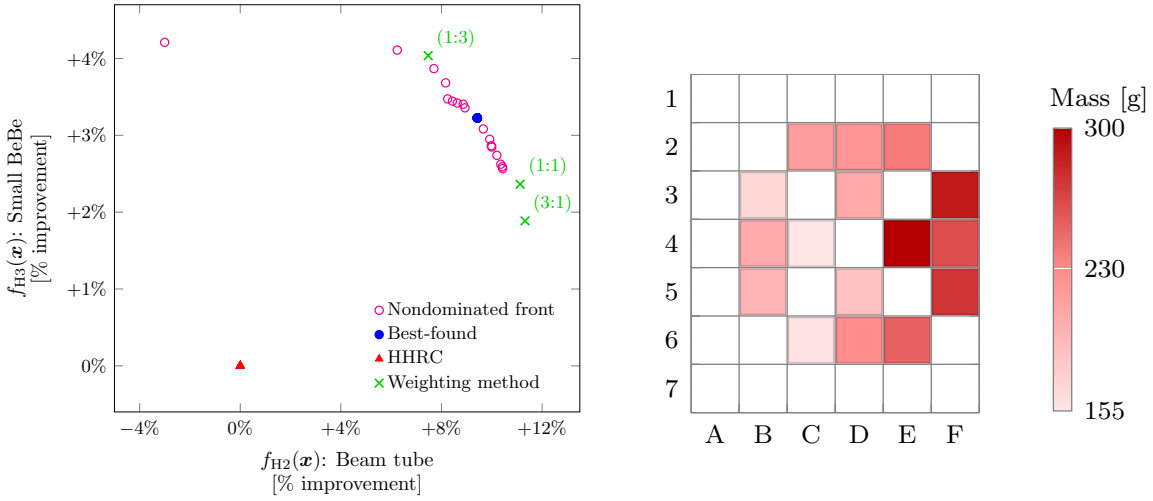
(a) *Nondominated front* (b) *Best-found reload configuration*

FIGURE 5.10: *Results obtained for problem instance* (5.18).

The reload configuration of the best solution found for problem instance (5.18) is presented in Figure 5.10(b) in terms of the ^{235}U mass in each fuel assembly. It is observed that the heavier-massed fuel assemblies are assigned to peripheral core positions, with an emphasis near position 2B where the Big BeBe rig is located. It is interesting to note that lighter-massed assemblies have been assigned to the central positions surrounding the Small BeBe rig, unlike the case for SAFARI-1, shown in Figure 5.7, where the IPR facilities were surrounded with heavier-massed assemblies. An analysis performed by Prinsloo [167] revealed that a neutron flux tilt present in the HHRC due to the beam line facility is flattened to some degree by the suggested reload configuration in Figure 5.10(b). Although this configuration worsens the fast neutron flux in the Small BeBe rig, the adjustment to the flux tilt notably imporoves the thermal neutron flux in that position.

Problem instance (5.19): Beam line research and central irradiation rig

The nondominated front obtained for problem instance (5.19) is presented in Figure 5.11(a), along with the best-found solution (with respect to F_q), the HHRC and the results obtained using the inherited weighting method. It is observed that significant improvements in the performance of objectives H2 and H3 (*i.e.* beam tube and Small BeBe rig, respectively) over that of the HHRC may be achieved simultaneously. The best solution found is also fairly well-balanced within the ranges of available objective function values in the nondominated front, when considering only those solutions in which simultaneous improvements have been realised. The weighting method, however, yielded solutions exhibiting the same behaviour as in problem instance (5.17). As before, a "jump" across the objective space is observed in Figure (5.11a) between the solution obtained using a 1:3 importance ratio, and the solutions obtained using ratios 1:1 and 3:1.



(a) *Nondominated front* (b) *Best-found reload configuration*

FIGURE 5.11: *Results obtained for problem instance* (5.19).

In Figure 5.11(b), the reload configuration of the best-found solution for problem instance (5.17) is presented visually in terms of the ^{235}U mass in each fuel assembly. A more pronounced assigment of heavier-massed fuel assemblies to positions closest to the beam tube is observed in Figure 5.11(b), when compared to the configuration in Figure 5.9(b).

5.6 Chapter summary

In this chapter, a scalarisation-based methodology for MICFMO was proposed in order to address the shortcomings present in the inherited optimisation feature within OSCAR-4. First, in §5.1, the notion of Pareto optimality and other related concepts were discussed in some detail. By doing so, the reader was supplied with the necessary background knowledge to gain a better understanding of the MOO modelling process and solution techniques employed throughout this dissertation.

The weighting method, which is used in the inherited methodology, was described in §5.2. Thereafter, in §5.3, the proposed methodology for MICFMO was described. It entailed descriptions of an alternative scalarisation approach, a constraint handling technique, and an adapted HS algorithm. In order to demonstrate the applicability of the proposed methodology, it was used to solve (approximately) several ICFMO problem instances for the SAFARI-1 and HOR reactors in §5.4 and §5.5, respectively.

Numerical results indicated that the newly proposed methodology is robust and versatile, and it may therefore be used as an effective decision support tool for designing reload configurations. Furthermore, the results obtained using the inherited weighting method revealed that there is no intuitive connection between the selection of weighting coefficients and the solutions obtained (which is consistent with the shortcomings of the method pointed out in §5.2). Although localised improvements in the attainment fronts were obtained using the weighting method, a good spread across the nondominated front is preferred because it is more informative to a decision maker. The new methodology was able to achieve such good spreads. The uncertainty associated with the selection of weighting coefficients may therefore be removed when adopting the new methodology.

CHAPTER 6

ANNs for the prediction of SAFARI-1 core parameters

Contents

In this chapter, several ANNs are constructed for the prediction of SAFARI-1 core parameters corresponding to various ICFMO objectives and constraints. The chapter opens with a motivation of the necessity of these neural networks. Thereafter, general concepts pertaining to ANNs are presented before moving on to a more comprehensive description of multilayer feedforward neural networks. Details on the construction of a suite of neural networks for SAFARI-1 are described next, before the chapter closes with numerical results obtained during the training and application of the networks, as well as a discussion of these results.

6.1 Introduction

In Chapter 5, a scalarisation-based methodology for MICFMO was proposed in order to address the shortcomings present in the optimisation feature within OSCAR-4, as it was inherited by the author. The first priority in this dissertation, as described in §1.3, has therefore been addressed. Attention may now turn towards the second priority, namely to investigate alternative

multiobjective computational methods in terms of their suitability in finding sets of high-quality trade-off solutions to the MICFMO problem.

It is apparent from the results reported in Chapter 5 that solving an MICFMO problem instance, using the OSCAR-4 system for objective and constraint function evaluations, is a computationally expensive task. Any "meaningful" investigation in which a number of multiobjective computational methods are to be applied multiple times (*i.e.* for the purpose of obtaining good solution run statistics) to several MICFMO problem instances would therefore require an excessive amount of computation time when using OSCAR-4. As mentioned in §3.4, the computational cost associated with these function evaluations may be reduced by replacing the core simulator with a computationally cheaper surrogate model. It was also remarked that a popular and effective approach adopted for this purpose in the literature is to employ ANNs for the prediction of reactor core parameters corresponding to objectives and constraints in ICFMO problems.

Given the successful application of ANNs in the ICFMO literature (*i.e.* the achievement of sufficiently accurate core parameter predictions at only a fraction of the computation time when compared to a core simulator [56, 131, 236]) they are also adopted in this dissertation to reduce the computational cost of objective and constraint function evaluations. The availability of off-the-shelf ANN software tools, such as the Neural Network Toolbox [215] within the Matlab software suite [214], is a further motivation for the usage of ANNs in this dissertation. Therefore, in order to aid in an investigation into appropriate methods for solving the MICFMO problem, ANN surrogate models are constructed in this chapter for the prediction of SAFARI-1 core parameters corresponding to objectives and constraints.

6.2 Fundamental notions related to ANNs

ANNs were originally designed in an attempt to find mathematical models for information processing in biological systems, with McCulloch and Pitts [132] generally credited for designing the first neural networks in 1943 [58]. According to the definition proposed by Fausett [58], "*an artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural networks.*" There are numerous fields of application for ANNs, including classification, clustering, function approximation and optimisation [134].

As an information-processing system, an ANN consists of several simple processing units called *artificial neurons*. These neurons are connected to one another by means of directed communication links over which signals may be passed. The specific pattern (or topology) in which the neurons are connected is called the *architecture* of the network. Each communication link is also associated with a *weight* which typically scales any signal being sent. The process of determining the weights of an ANN is referred to as *training*, which is governed by a so-called *training algorithm*. Regarding the processing capability of neurons in ANNs — each neuron is able to receive input signals over a number of communication links, to process an aggregation of these signals by means of an associated *activation function*, and then to send an output signal over the communication links to other neurons. This output signal returned by an activation function for a given neuron is referred to as the *activation* of that neuron. A graphical example of an artificial neuron and its processing capability is presented in Figure 6.1.

An ANN may generally be characterised by three of the aforementioned properties, namely its architecture, the training algorithm employed and the activation function used [58]. Each of these characteristic properties are touched upon next.

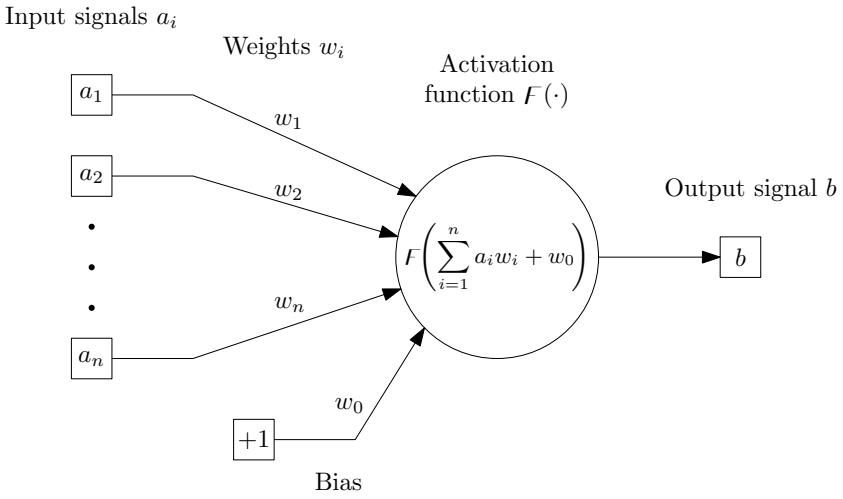


FIGURE 6.1: *Example of an artificial neuron and its processing capability.*

In terms of the architecture, neurons are often partitioned into subsets, called *layers*, such that those in the same layer typically behave in the same manner. This layered "behaviour" generally refers to the use of the same activation function and the same pattern of communication link connections. Accordingly, ANNs are often classified as *single-layer* or *multilayer* networks [58]. Another classification is based on the flow of signals across the network. In *feedforward* networks, signals may only be sent from one node to the next in a forward direction across the network layers, while in *recurrent* networks, there are closed tours from a neuron back to itself [58].

The training algorithm, as mentioned above, is used to determine the values of the weights in an ANN. There are generally two different training paradigms that may be considered, namely *supervised* and *unsupervised* training. In a supervised paradigm, a network is trained by presenting it with a set of known input-output pairs. The weights in the network are then adjusted by a training algorithm so that the predicted and target (*i.e.* known) outputs are close to one another. In an unsupervised paradigm, however, a network is presented with a set of inputs only. A training algorithm should then adjust the weights such that similar inputs are grouped together and assigned to the same output [58]. Each iteration of adjusting the network weights, supervised or unsupervised, using the entire input set is called an *epoch*.

There are a number of different activation functions that may be employed within an ANN, each having its own advantages and disadvantages for various applications. Recall that neurons within the same layer usually employ the same activation function, although this is not a requirement. For input neurons, the identity function is usually employed, *i.e.* an external input signal is simply transmitted onwards as is. Typical activation functions employed for the other neurons include the Heaviside step function, linear functions, sigmoidal functions (*e.g.* the logistic function), and Gaussian functions [58]. In multilayer neural networks, it is required that nonlinear activation functions be employed in at least one of the layers so as to benefit from the architecture (since the results obtained by transmitted signals through two or more layers of neurons employing linear activation functions can also be obtained using a single layer) [58].

6.3 Multilayer feedforward neural networks

An important class of ANNs is *multilayer feedforward neural networks* (MFNNs). As the name suggests, these networks consist of multiple layers of neurons, while signals are only sent in a

forward direction over the network. The first layer of neurons in a multilayer network is called the *input layer* whereas the last layer is called the *output layer*. All the layers in between are referred to as *hidden layers*. Recall from the previous section that the identity function is typically employed as activation function within input neurons. Accordingly, in an MFNN, the input layer simply transmits the external input signal onwards. Neurons within the same layer are not connected to one another. Each neuron within a particular layer is, however, typically connected to every neuron in the following layer. An illustration of the general architecture of an MFNN is presented in Figure 6.2

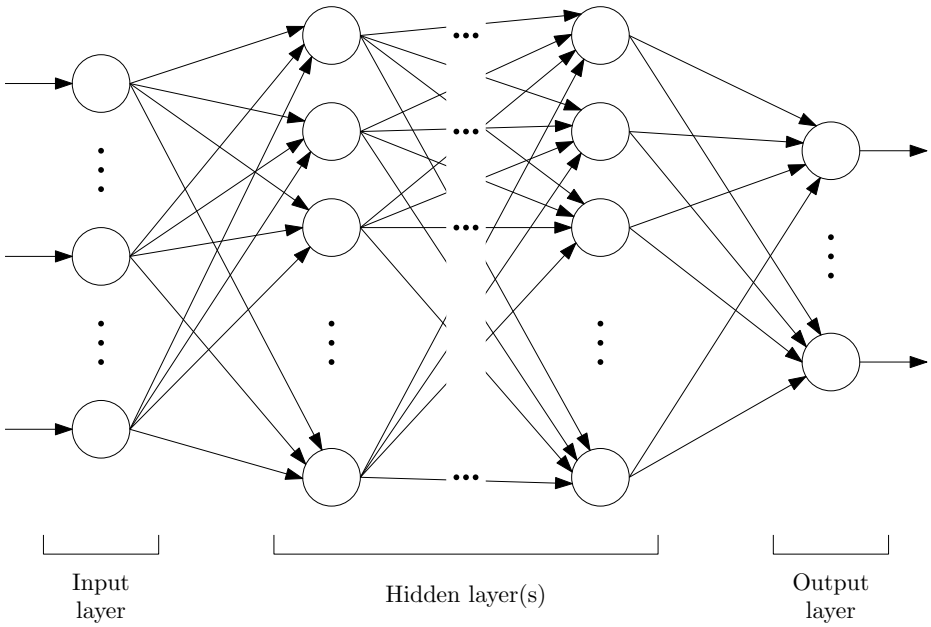


FIGURE 6.2: *The general architecture of a multilayer feedforward neural network.*

MFNNs have very general approximation properties, hence their popularity and wide-spread usage [18, 58]. In particular, these networks are considered to be universal function approximators. Hornik *et al.* [85] showed that MFNNs are capable of approximating virtually any function of interest to an arbitrary degree of accuracy, provided that a sufficient number of hidden neurons are available. Due to the powerful property of being universal approximators, MFNNs are considered as surrogates in this dissertation.

6.3.1 An MFNN with one hidden layer

Although several hidden layers in an MFNN may be beneficial in certain applications, using only one hidden layer should be sufficient in almost all function approximation applications of these networks [58]. For the purposes of this dissertation, therefore, MFNNs consisting of one hidden layer are employed for the prediction of SAFARI-1 core parameters.

Consider an MFNN with one hidden layer which employs the identity function as activation function in the input layer. Let $w_{ji}^{(1)}$ denote a weight in the first layer of the network, associated with the communication link from input neuron i to hidden neuron j, while $w_{j0}^{(1)}$ denotes the so-called *bias* for hidden unit j. Similarly, let $w_{kj}^{(2)}$ denote a weight in the second layer, associated with the communication link from hidden neuron j to output neuron k, while $w_{k0}^{(2)}$ denotes the bias for output neuron k. Suppose there are ℓ inputs to the network $a_1, \ldots, a_\ell$. Since the identity

function is employed in the input layer, it is not specifically denoted. Therefore, the activation of input neuron i simply corresponds to a_i for $i = 1, \ldots, \ell$. Let the activation function employed in the hidden layer be denoted by $F_{(1)}(\cdot)$, with b_j denoting the activation of hidden neuron j for $j = 1, \ldots, m$. Similarly, let $F_{(2)}(\cdot)$ denote the activation function employed in the output layer, with c_k denoting the activation of hidden neuron k for $k = 1, \ldots, n$. An illustration of the MFNN with one hidden layer, using the aforementioned notation, is presented in Figure 6.3. Note that the bias $w_{j0}^{(1)}$ may be represented as a weight from an additional input a_0 which is permanently set to unity (and likewise for $w_{k0}^{(2)}$ with an extra hidden neuron $b_0 = 1$) [17].

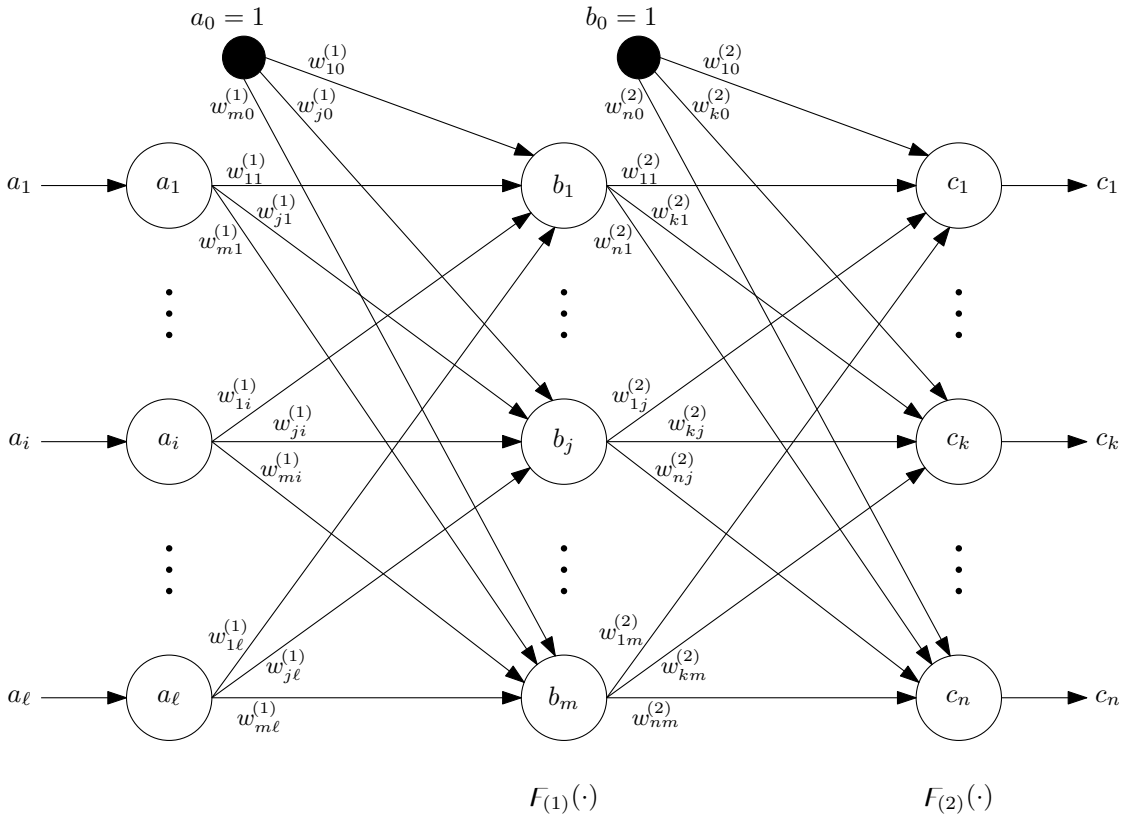


FIGURE 6.3: *A multilayer feedforward neural network with one hidden layer.*

Following the discussion in Bishop [17], an analytical expression for the function represented by the MFNN in Figure 6.3 may be written as follows. First, the *net input* to hidden neuron j, denoted by $\eta_j^{(1)}$, is determined by forming a weighted linear combination of the input activations, and adding a bias, such that

$$\begin{aligned} \eta_j^{(1)} &= \sum_{i=1}^{\ell} w_{ji}^{(1)} a_i + w_{j0}^{(1)}, \\ &= \sum_{i=0}^{\ell} w_{ji}^{(1)} a_i. \end{aligned} \tag{6.1}$$

Next, the activation of hidden neuron j is obtained by processing the net input (6.1) using the activation function $F_{(1)}$ associated with the hidden layer. Therefore

$$b_j = F_{(1)}\left(\eta_j^{(1)}\right). \tag{6.2}$$

Similarly, by forming a weighted linear combination of the hidden activations, and adding a bias, the net input to output neuron k, denoted by $\eta_k^{(2)}$, is given as

$$\begin{aligned}\eta_k^{(2)} &= \sum_{j=1}^{m} w_{kj}^{(2)} b_j + w_{k0}^{(2)}, \\ &= \sum_{j=0}^{m} w_{kj}^{(2)} b_j. \end{aligned} \tag{6.3}$$

Accordingly, the activation of output neuron k is then obtained by applying the activation function $F_{(2)}$ associated with the output layer to process the net input (6.3), such that

$$c_k = F_{(2)}\left(\eta_k^{(2)}\right). \tag{6.4}$$

Now, by combining (6.1)–(6.4), an analytical expression is obtained for the function represented by the MFNN in Figure 6.3, namely

$$c_k = F_{(2)}\left(\sum_{j=0}^{m} w_{kj}^{(2)} F_{(1)}\left(\sum_{i=0}^{\ell} w_{ji}^{(1)} a_i\right)\right). \tag{6.5}$$

An MFNN such as the one in Figure 6.3 is therefore simply a nonlinear function controlled by adjustable weighting parameters [17]. The manner in which values for these weights are determined is touched upon next.

6.3.2 The backpropagation training algorithm

The training paradigm adopted within the context of MFNNs for function approximation is that of supervised training. Accordingly, a training set of known input-output pairs is utilised during the adjustment of the network weights. The popular *backpropagation training algorithm* is often employed for training MFNNs, and it is essentially a gradient-based optimisation technique for minimising some appropriate error function [17, 58]. Within the algorithm, a method known as *backpropagation of errors* is utilised for determining the derivatives of the error function with respect to the network weights. These derivatives are required in the *gradient descent method* employed in the training algorithm to minimise the error function. In Appendix A, a derivation of the backpropagation of errors method is presented within the context of the MFNN shown in Figure 6.3. How the derivatives are employed in the gradient descent method is also revealed in the appendix.

There are three phases involved during backpropagation training of neural networks, namely [58]:

1. The forward propagation of an input training vector;
2. The calculation and backpropagation of the associated error; and
3. The adjustment of the network weights.

During the forward propagation phase, an input training vector is presented to and transmitted over the network. The activations of all the hidden and output neurons are then calculated. Note that the output neuron activations form the response of the network for the given input training vector. During the second phase, the network response is compared to the target value using an appropriate error function. So-called *errors* associated with the output neurons, denoted

by δ_k, are then calculated and propagated backwards over the network in order to calculate corresponding errors associated with the hidden neurons, denoted by δ_j (see Appendix A for details). Then, during the third phase, all the network weights are adjusted simultaneously based on these δ-values and the neuron activations. There are, in general, two choices for when to update the weights: (1) after each input training vector has been presented to the network, which is referred to as *online learning*, or (2) after all the input vectors in the training set have been presented to the network, which is referred to as *batch learning*.

According to Fausett [58], an activation function for a backpropagation MFNN should be continuous, differentiable and monotonically non-decreasing. In addition, adopting a function whose derivative is easy to compute is very desirable from a computational efficiency point of view. Typically, a sigmoidal function (which satisfies these criteria) is chosen as the activation function for the hidden layer.

Several variations on the standard backpropagation training algorithm have been developed. These variations include, for example, heuristic modifications to the gradient descent method, and the adoption of techniques based on conjugate gradients and quasi-Newton methods to minimise the error function [17]. A particular variation, called the *Levenberg-Marquardt backpropagation algorithm*, is regarded as one of the most efficient training algorithms for MFNNs [12, 70, 241]. The algorithm is applicable in cases where a sum-of-squares error function is adopted, and is generally recommended as a first-choice algorithm for training small- and medium-sized networks [12, 241].

6.4 The architecture and training of MFNNs

The ability of an ANN to respond correctly to the input vectors used during training is known as *memorisation*. Its ability to respond reasonably well to new (unseen) input vectors that are similar to, but different from, the training vectors, on the other hand, is known as *generalisation*. Within the context of MFNNs for function approximation, the aim, in general, is to train the network so that a balance is achieved between the two competing abilities of memorisation and generalisation [58]. Training that favours memorisation is also known as *overfitting* and should generally be avoided.

It is convenient to draw an analogy between ANN training and polynomial curve fitting in order to illustrate the notion generalisation. A neural network model that is too simple or too complicated will exhibit poor generalisation, analogous to a polynomial constructed during curve fitting with too small or to large a degree, which will yield poor predictions for new data points [17]. This notion of poor generalisation is illustrated graphically in Figure 6.4.

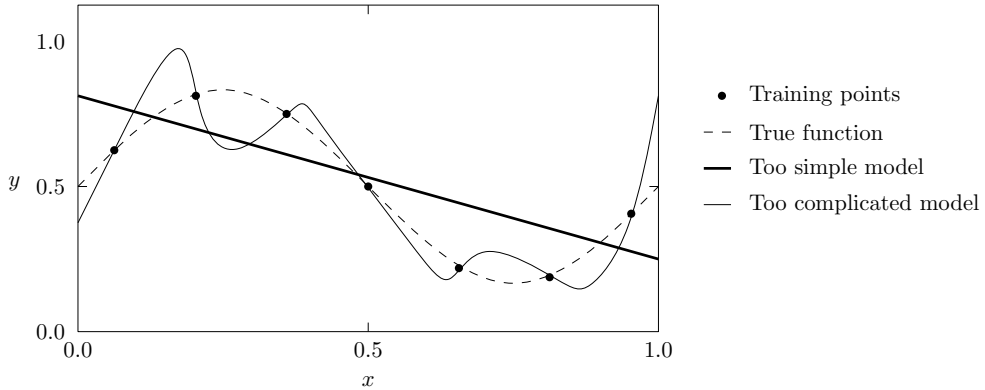


FIGURE 6.4: *Example of poor generalisation.*

Several techniques have been developed that may assist in obtaining good generalisation for a neural network, examples of which include early stopping, regularisation, training with noise and cross-validation [17]. Essentially, these techniques all aim to control the complexity of the network model in one way or another. The technique of *regularisation* is adopted in this dissertation and it attempts to control model complexity by adding a penalty term to the network's error function. This is done to encourage smoother network mappings [17], *i.e.* to avoid the behaviour observed in the case of the "Too complicated model" in Figure 6.4. The penalty term is usually related to the size of the network weights and, when the weights are kept small, the network mapping is typically smooth [59].

The nature of the training data is also an important factor to consider in terms of the generalisation of a network. Input training vectors have to form a representative subset of the region of vectors to which one wishes to generalise [211].

A critical aspect that arises during the construction of an MFNN with one hidden layer is the required number of hidden neurons. According to Svozil *et al.* [211], this number of neurons depends on the size of the training set, the amount of noise present in the training vectors, the complexity of the function being approximated, and the technique employed to obtain good generalisation. It is clear, then, that the required number of hidden neurons is very problem-dependent and essentially has to be determined empirically. In general, however, networks of smaller size are preferred over larger networks because the latter may be able to memorise the training data (*i.e.* overfitting) which leads to poor generalisation [134].

6.5 Constructing MFNNs for SAFARI-1

The Neural Network Toolbox [215] within the Matlab software suite [214] was utilised as an off-the-shelf software tool for constructing the MFNNs in this dissertation. The core parameters corresponding to the typical objectives and constraints for the SAFARI-1 reactor described in §4.5 are considered here for neural network modelling purposes and are listed in Table 6.1. In this section, the steps that were followed during construction of the associated neural networks in the Toolbox are presented.

Symbol	Reactor core parameter
ϕ_{B12}	Average thermal neutron flux over the faces of beam tubes 1 and 2
ϕ_{B5}	Average thermal neutron flux over the face of beam tube 5
ϕ_{Si}	Average thermal neutron flux over the silicon doping facility
ϕ_{I1}	Maximum axial thermal neutron flux in the first IPR facility
ϕ_{I2}	Maximum axial thermal neutron flux in the second IPR facility
$\psi_{\mathrm{Mo}}^{\mathrm{tot}}$	Assembly-averaged power levels in all molybdenum rigs
$\psi_{\mathrm{Mo}}^{\mathrm{min}}$	Assembly-averaged power level in the molybdenum rig with the minimum power
ρ_{cbw}	Control bank worth
ρ_{sdm}	Shutdown margin
ρ_{ex}	Excess reactivity
ψ_{ppf}	Power peaking factor

TABLE 6.1: *SAFARI-1 core parameters considered for ANN modelling.*

6.5.1 The training data

In order to train MFNNs for the prediction of SAFARI-1 core parameters, the same operational cycle for the reactor considered in §5.4.1 (*i.e.* cycle C1211-1) is considered here. The training data comprise a set of 20 000 fuel reload configurations along with their respective core parameter values returned by the OSCAR-4 system. The manner in which these data were selected is described below. This set has been randomly partitioned into a *training set* of 17 000 configurations, and a *test set* of 3 000 configurations for each network, as described below. The training set, therefore, only constitutes approximately $4.2 \times 10^{-21}\%$ of the decision space for an MICFMO problem instance based on the SAFARI-1 reactor.

In previous studies conducted by the author, numerous reload configurations for cycle C1211-1 were evaluated as part of solving ICFMO problem instances. These instances involved different combinations of the core parameters corresponding to the typical SAFARI-1 objectives and constraints. From this collection of evaluated reload configurations, a subset containing approximately 5 000 configurations that achieved the largest and smallest values for each core parameter listed in Table 6.1 has been selected for inclusion in the training set. By including these configurations, an attempt is made to cover the extremes of the core parameter space in which predictions are to be made.

Apart from the above, new reload configurations for cycle C1211-1 were generated randomly, evaluated using the OSCAR-4 system, and added to the training set until it reached a size of 20 000. These configurations correspond to random permutations of the twenty-six fuel assemblies loaded into the core and have been generated according to the Fisher-Yates shuffle[1]. By including these configurations in the training set, an attempt is made to achieve diversity in both the reload configuration space and core parameter space.

6.5.2 The network architecture

Eleven MFNNs were constructed — one for predicting each core parameter listed in Table 6.1. Accordingly, each network contains only one output neuron. Although a single network could have been constructed for predicting all eleven parameters simultaneously, preliminary testing indicated that unacceptably large prediction errors would be incurred. In each network, twenty-six input neurons are employed and they correspond to the fixed fuel loading positions in the SAFARI-1 reactor core. Furthermore, the inputs to the networks were chosen as the ^{235}U mass of a fuel assembly assigned to a specific loading position in a reload configuration. As already motivated in §6.3.1, MFNNs consisting of one hidden layer are employed. The number of hidden neurons within each network was determined empirically. The specific procedure that was followed to determine these numbers, however, is described in a later section.

In each network, the activation functions suggested by the Toolbox were adopted. For neurons in the hidden layer, this corresponds to a hyperbolic tangent sigmoidal function whereas, for the output neuron, it corresponds to a linear function. The values of the network inputs and responses were also normalised to the range $[-1, 1]$ during preprocessing performed by the Toolbox. This normalisation was performed to improve the efficiency with which the networks are trained [12].

[1]The *Fisher-Yates shuffle* is an algorithm which produces an unbiased permutation, *i.e.* every permutation is equally likely [21].

6.5.3 Training of the neural networks

A number of different training algorithms are available for use in the Toolbox. Although the Levenberg-Marquardt backpropagation algorithm is generally recommended as a first-choice algorithm for training MFNNs due to its efficiency (see §6.3.2), more emphasis is placed in this dissertation on obtaining good generalisation in the networks, as opposed to the speed of training them. This is due to the intended purpose of using these networks to predict core parameters corresponding to hundreds of thousands of reload configurations during MICFMO comparative studies. As a result, the *Bayesian regularisation backpropagation algorithm* within the Toolbox was chosen as the training algorithm to employ. In a Bayesian learning framework, the weights in a neural network are considered to be random variables and their probability density functions may be updated according to Bayes' rule [127].

Recall that a penalty term is added to the network error function in regularisation techniques. Therefore, the function asopted to minimise during training has the form $\tilde{E} = \beta E + \alpha E_w$, where E is the original network error function, E_w is the penalty term, and α and β are importance parameters, also referred to as *regularisation coefficients*. Typically, E corresponds to the sum of squared errors, whereas E_w is usually the sum of squares of the network weights. One of the difficulties with regularisation, however, is the selection of appropriate values for α and β. By extending the Bayesian framework to include regularisation, these importance parameters may also be optimised during training by application of Bayes' rule. Foresee and Hagan [59] demonstrated that networks trained using Bayesian regularisation typically exhibit excellent generalisation.

One of the features of the Bayesian regularisation algorithm is that it provides a factor called the *effective number of parameters*, denoted here by ξ, which may be regarded as a measure of how many network weights are effectively used in reducing the error function. This factor ξ may be utilised as an aid to decide whether an MFNN consists of an appropriate number of hidden neurons [59]. If, for example, the value of ξ is very close to the total number of weights in the network upon termination of the training process, it generally indicates that more hidden neurons should be added to the network. According to Foresee and Hagan [59], the algorithm will still yield comparable values for ξ, E and E_w when too many hidden neurons are added to a network. As such, overfitting of a network that is too large may be avoided when the Bayesian regularisation algorithm is employed.

The number of hidden neurons within each network was determined empirically. In the empirical study, the total number of hidden neurons within a network was incrementally increased from 200 up to a satisfactory number (in increments of 50). The starting value of 200 was selected based on preliminary calculations using one network only, and by considering the literature as a guideline (the number varies between 110 and 400 [56, 99, 131, 140]). In each case, the network was trained using the default stopping criteria provided in the Toolbox. Upon termination of the training algorithm, the results were inspected so as to ascertain whether the process had, in fact, converged. The process was deemed to have converged if the network error function value over the training and test sets, and the effective number of parameters ξ, remain relatively constant over several epochs [12]. The *mean squared error* (MSE) function is employed during training. If the process did not converge, the number of epochs was increased and the training continued. Otherwise, when the algorithm did converge, it was verified that the value of ξ was not too close to the total number of weights in the network, and that the absolute relative prediction errors for the training and test sets were acceptable. The number of hidden neurons in the network was increased and the training algorithm restarted if either of the verifications failed.

6.6 Training results

Following the approach described in the previous section, eleven MFNNs were constructed for the prediction of the SAFARI-1 core parameters listed in Table 6.1. All calculations were performed on a personal computer with the following specifications: An Intel® Core™ i7-2720QM CPU with 8 GB of RAM operating at 2.20 GHz within a 64-bit operating system. The computation time required for training each of the final networks varied between six and thirty-seven hours, depending on the number of hidden neurons in the network and the number of epochs required for convergence.

Arguably the most important result to report concerns the computational efficiency of the networks. By using the MFNNs instead of the OSCAR-4 system, the computation time required for the evaluation of a single reload configuration for the SAFARI-1 reactor may be reduced by four orders of magnitude.

A summary of the performance of the trained networks in respect of their prediction errors is presented in Table 6.2. This table contains the average and maximum absolute relative prediction errors of the networks in respect of the training and test sets.

	Training set		Test set	
Network	Average	Maximum	Average	Maximum
ϕ_{B12}	0.07 %	0.35 %	0.13 %	0.75 %
ϕ_{B5}	0.06 %	0.38 %	0.12 %	0.73 %
ϕ_{Si}	0.07 %	0.37 %	0.13 %	0.61 %
ϕ_{I1}	0.09 %	0.52 %	0.14 %	0.92 %
ϕ_{I2}	0.09 %	0.44 %	0.12 %	0.95 %
$\psi_{\text{Mo}}^{\text{tot}}$	0.19 %	0.93 %	0.20 %	0.90 %
$\psi_{\text{Mo}}^{\text{min}}$	0.75 %	2.96 %	0.82 %	3.61 %
ρ_{cbw}	0.03 %	0.20 %	0.07 %	0.33 %
ρ_{sdm}	0.12 %	0.67 %	0.24 %	1.28 %
ρ_{ex}	0.09 %	0.60 %	0.18 %	0.97 %
ψ_{ppf}	0.80 %	5.52 %	1.62 %	8.71 %

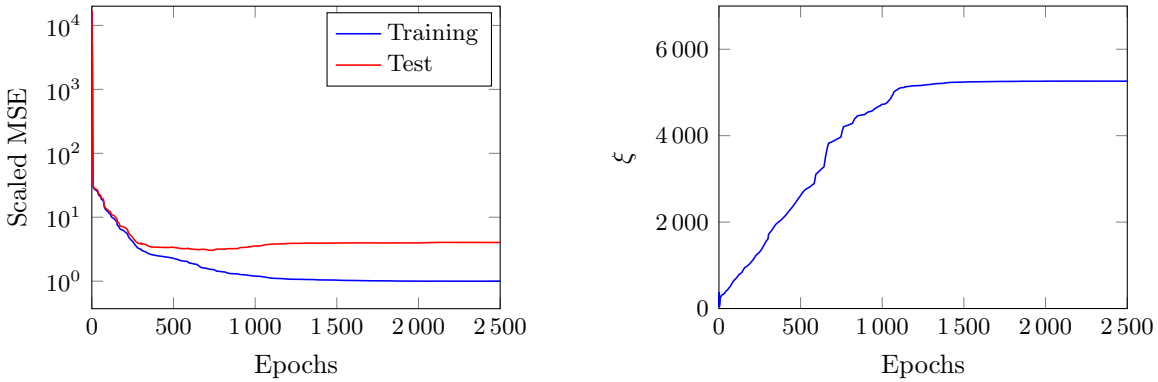
TABLE 6.2: *Average and maximum absolute relative prediction errors on the training and test sets.*

It is observed in Table 6.2 that high-quality predictions are, on average, produced by the networks. An average error of less than 1% on the training set is produced by all eleven networks, while ten of them also produce an average error of less than 1% on the test set. Furthermore, the maximum errors for the five networks that predict neutron flux levels, *i.e.* those denoted by ϕ, are all less than 1% on the training and tests sets. The comparatively large maximum errors obtained by the ψ_{ppf} network (approximately 5% and 8% on the training and test sets, respectively) are still deemed to be of an acceptable accuracy for use in MICFMO calculations when compared to an error of 14% found in the literature [131]. It is possible that the network inputs (*i.e.* the ^{235}U mass in a fuel assembly) are insufficient for the purpose of training this network because axial disparities within an assembly, which affect the three-dimensional power peaking calculation in the OSCAR-4 system, are not taken into consideration. An investigation into this phenomenon is, however, left for future studies.

Several of the graphical results produced during the training of the networks are also presented. For the purpose of improved readability, however, only a subset of the results are presented in this section. The subset includes convergence graphs for the ϕ_{B5} and $\psi_{\text{Mo}}^{\text{min}}$ networks, as well

as prediction quality graphs for the ϕ_{B5}, $\psi_{\text{Mo}}^{\text{min}}$ and ψ_{ppf} networks. These networks all have 200 hidden neurons in their architecture. The remaining results may be found in Appendix B. Note that the MSE values within the graphs have been scaled relative to the minimum value obtained. Similarly, the network prediction and target values have been scaled so as to correspond to the $[-1, 1]$ range employed during training (see §6.5.2).

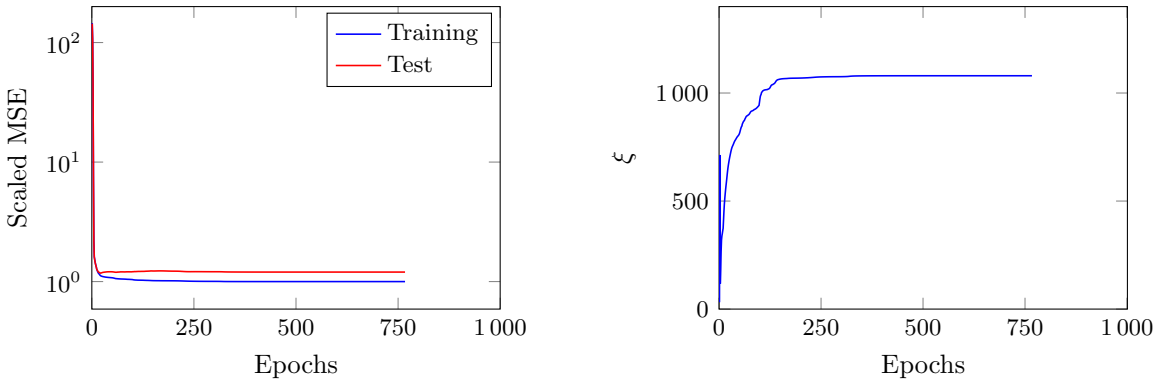
In Figure 6.5, the convergence graphs of the training process for network ϕ_{B5} are presented. It is observed in Figure 6.5(a) that the scaled MSE of the training and tests sets remain fairly constant for several hundred epochs, as is also the case in Figure 6.5(b) for the effective number of parameters ξ. The network was therefore deemed to have converged. Initially, however, this network did not achieve convergence after a default stopping criterion of 1 000 epochs had been reached. Accordingly, the number of epochs was increased before resuming the training.



(a) *Scaled MSE of the training and test sets* (b) *The effective number of parameters*

FIGURE 6.5: *Convergence results for the ϕ_{B5} neural network.*

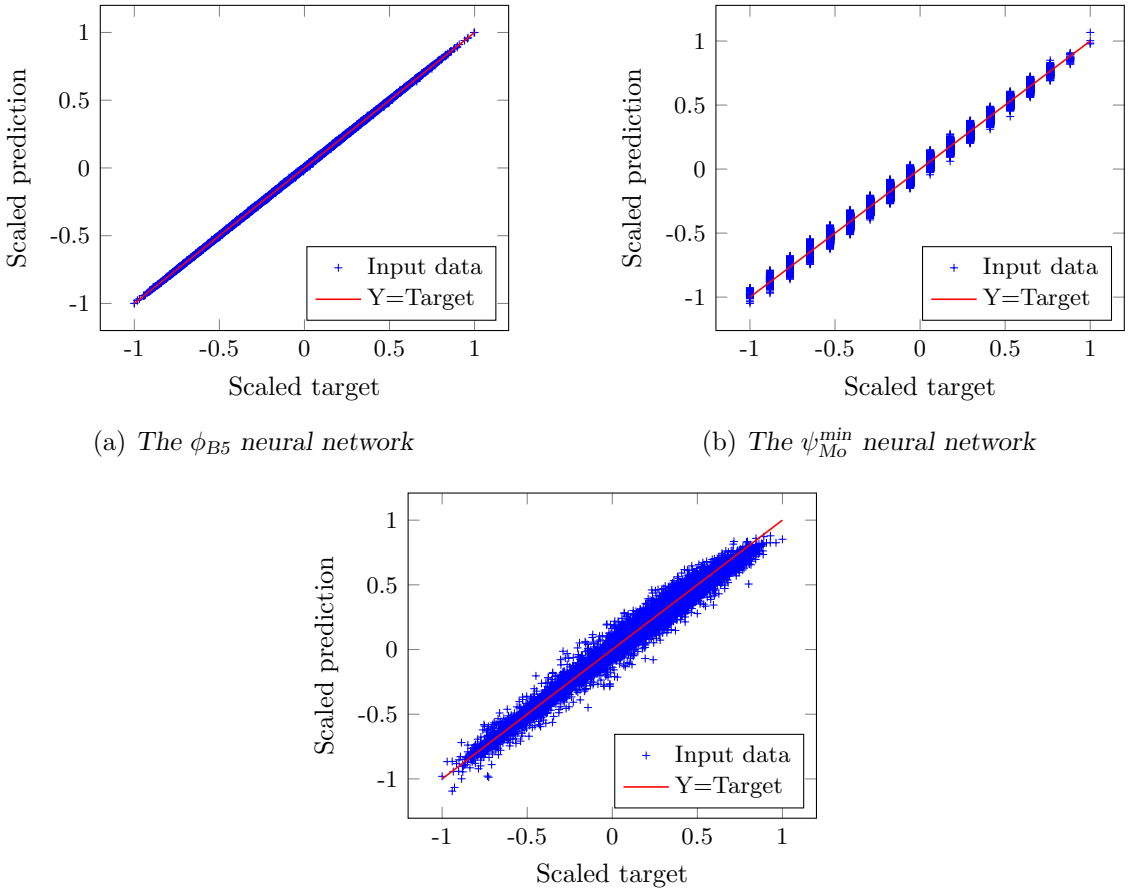
The convergence graphs of the training process for network $\psi_{\text{Mo}}^{\text{min}}$ are presented in Figure 6.6. This network, unlike the previous example, converged after 767 epochs according to the default stopping criteria of the Toolbox. Although $\xi \approx 1\,080$ is much smaller than the total number of weights in the network, namely 5 601, due to the properties of the Bayesian regularisation algorithm (as described in §6.5.3), overfitting would have been avoided.



(a) *Scaled MSE of the training and test sets* (b) *The effective number of parameters*

FIGURE 6.6: *Convergence results for the ψ_{Mo}^{min} neural network.*

Graphical results pertaining to the networks' predictive capabilities are presented in Figure 6.7. A scatter graph of the 20 000 target values in the combined training and test sets versus their predicted values is presented for each network ϕ_{B5}, $\psi_{\text{Mo}}^{\text{min}}$ and ψ_{ppf}. The straight line in each graph, labelled "Y=Target" in the legend, corresponds to the theoretical case in which perfect predictions are achieved.



(a) *The ϕ_{B5} neural network*

(b) *The ψ_{Mo}^{min} neural network*

(c) *The ψ_{ppf} neural network*

FIGURE 6.7: *Prediction quality results for the ϕ_{B5}, ψ_{Mo}^{min} and ψ_{ppf} neural networks.*

In Figure 6.7(a), it is observed for the ϕ_{B5} network that an exceptionally good fit is achieved of the network predictions to their target values. A more peculiar graph is found in Figure 6.7(b) for the $\psi_{\text{Mo}}^{\text{min}}$. This peculiarity may be attributed to round-off effects in the OSCAR-4 results, making the target values discrete. It is observed, however, that the network is still able to achieve good predictions to these discrete targets. Finally, in Figure 6.7(c), the results are presented for the ψ_{ppf} network. The predictions for this network are distributed wider about the target line, as may be seen in the graph, when compared to those of the ϕ_{B5} and $\psi_{\text{Mo}}^{\text{min}}$ networks. Nevertheless, the quality of predictions are still acceptable, and there are no outliers to be concerned about.

6.7 Application of the networks on cycle C1211-1

The results presented in the previous section demonstrate that MFNNs have the ability to predict SAFARI-1 core parameters (with acceptable accuracy) much quicker than when using explicit core simulator calculations, *e.g.* using the OSCAR-4 system.

Given that the test set employed in the previous section contained only 3 000 reload configurations, it would be advantageous to further verify the accuracy and generalisation of the networks on a larger set of new (unseen) reload configurations for cycle C1211-1. Therefore, a *verification set* consisting of 30 000 random (according to a uniform distribution) reload configurations evaluated by the OSCAR-4 system was created for this purpose. This verification set was presented to the neural networks in order to predict the respective SAFARI-1 core parameters.

A summary of the prediction results for the verification set is presented in Table 6.3. This table contains, as before, the average and maximum absolute relative prediction errors of the networks. It is observed that the quality of the predictions for the verification set is of the same order of magnitude to that of the test set in Table 6.2. Ten of the eleven networks produce an average error of less than 1%, while all the flux networks also produce a maximum error of less than 1%. The maximum error produced by the ψ_{ppf} network has, however, worsened when compared to the test set. Fortunately, it remains at an acceptable level given the 14% error found in the literature [131].

	Verification set	
Network	Average	Maximum
ϕ_{B12}	0.13 %	0.88 %
ϕ_{B5}	0.13 %	0.82 %
ϕ_{Si}	0.13 %	0.80 %
ϕ_{I1}	0.14 %	0.88 %
ϕ_{I2}	0.12 %	0.71 %
$\psi_{\text{Mo}}^{\text{tot}}$	0.20 %	0.96 %
$\psi_{\text{Mo}}^{\text{min}}$	0.85 %	3.60 %
ρ_{cbw}	0.07 %	0.41 %
ρ_{sdm}	0.25 %	1.73 %
ρ_{ex}	0.19 %	1.24 %
ψ_{ppf}	1.67 %	10.59 %

TABLE 6.3: *Average and maximum absolute relative prediction errors on the verification set.*

The application of the MFNNs on the verification set therefore reinforced the conclusion that these networks exhibit good generalisation in the context of cycle C1211-1, and have the ability to predict SAFARI-1 core parameters with an acceptable accuracy. As a result, the networks are sufficient for the purpose of reducing the computational cost of function evaluations in an investigation of appropriate methods for solving the MICFMO problem.

6.8 Application of the networks to other operational cycles

One of the reasons for using the ^{235}U mass of a fuel assembly as network input is to enable the networks to predict SAFARI-1 core parameters for a different operational cycle than C1211-1 on which they were trained. Such different predictions are possible because the ^{235}U mass in the fuel assemblies loaded into the SAFARI-1 core during different cycles are typically destributed in a fairly similar manner. This may be observed in Figure 6.8, which consists of a graph illustrating the averages and standard deviations of the ^{235}U masses loaded into the SAFARI-1 reactor over a period of five years.

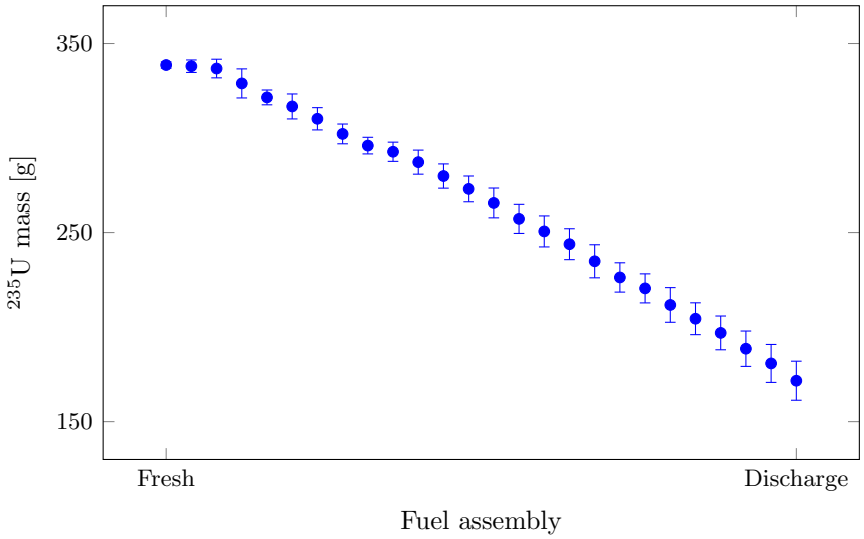


FIGURE 6.8: *The ^{235}U mass distribution in assemblies loaded into the SAFARI-1 reactor over a period of five years.*

In order to test the extended predictive capability of the networks, two other operational cycles were selected from the SAFARI-1 reactor history which exhibit relatively different fuel distributions than the original cycle (*i.e.* they were selected to represent extreme test cases). These two cycles are designated as C1003-1 and C1401-2. A test set of 4 500 random reload configurations was created (according to the Fisher-Yates shuffle) for each of the two cycles, and subsequently evaluated using the OSCAR-4 system. These test sets were then presented to the neural networks in order to predict the respective SAFARI-1 core parameters. The average and maximum absolute relative prediction errors produced by the networks are presented in Table 6.4.

	Cycle C1003-1		Cycle C1401-2	
Network	Average	Maximum	Average	Maximum
ϕ_{B12}	1.84 %	7.11 %	2.58 %	9.01 %
ϕ_{B5}	2.18 %	9.47 %	4.91 %	17.62 %
ϕ_{Si}	9.52 %	15.37 %	5.50 %	11.81 %
ϕ_{I1}	2.61 %	7.24 %	5.78 %	10.89 %
ϕ_{I2}	4.17 %	8.27 %	5.00 %	8.40 %
ψ_{Mo}^{tot}	0.31 %	1.50 %	3.62 %	4.76 %
ψ_{Mo}^{min}	3.55 %	8.05 %	7.47 %	12.91 %
ρ_{cbw}	3.25 %	8.64 %	2.86 %	9.45 %
ρ_{sdm}	2.87 %	16.17 %	12.48 %	24.87 %
ρ_{ex}	2.00 %	6.68 %	16.71 %	22.68 %
ψ_{ppf}	5.09 %	20.04 %	4.47 %	19.61 %

TABLE 6.4: *Average and maximum absolute relative prediction errors on the test sets for operational cycles C1003-1 and C1401-2 of the SAFARI-1 reactor.*

It may be observed in Table 6.4 that the neural networks yield predictions of unacceptable quality for cycles C1003-1 and C1401-2. The maximum and average prediction errors for these two cycles are worse than those of cycle C1211-1 by approximately an entire order of magnitude. As a result, these neural networks should not be employed for predicting SAFARI-1 core parameters for operational cycles other than C1211-1 on which it was trained. If neural networks are sought

in which sufficient generalisation is achieved so as to predict parameters for an arbitrary cycle of the SAFARI-1 reactor, a more extensive training set is required. Such a set would have to contain reload configurations that incorporate the full range of typical ^{235}U mass distributions, as shown in Figure 6.8.

6.9 Chapter summary

In this chapter, several ANNs were constructed for the prediction of SAFARI-1 core parameters corresponding to MICFMO objectives and constraints. The motivation for the construction of these neural networks, namely to aid in the investigation of different computational methods for solving the MICFMO problem, was presented in §6.1. Next, an introduction to some general concepts pertaining to ANNs was presented in §6.2.

A specific class of neural networks, namely MFNNs, was reviewed in §6.3. In particular, a description of an MFNN with one hidden layer was presented in §6.3.1, while an overview of the backpropagation training algorithm, often employed for training such networks, was given in §6.3.2. Several important notions pertaining to the architecture and training of MFNNs were elucidated in §6.4 before details in respect of the construction of the SAFARI-1 neural networks were presented in §6.5.

In §6.6, the results obtained during the training of the neural networks in respect of a specific SAFARI-1 operational cycle were presented. It was found that the MFNNs have the ability to predict SAFARI-1 core parameters much quicker (and with an acceptable accuracy) than when using the OSCAR-4 system (typically by four orders of magnitude). Additional verification of the networks was performed in §6.7, followed by application tests in §6.8 to ascertain their prediction capabilities in the contexts of other operational cycles. The latter tests indicated that the networks should only be used to predict core parameters for the cycle they were trained on.

CHAPTER 7

Multiobjective metaheuristics for solving the MICFMO problem

Contents

As mentioned in §1.3, one of the priorities in this dissertation is to investigate different multiobjective computational methods in terms of their suitability in respect of finding sets of high-quality trade-off solutions to the MICFMO problem. In this chapter, several multiobjective metaheuristics, employed in a comparative study later in this dissertation, are discussed. Two different constraint handling techniques for MOO are also described. The chapter then closes with a discussion on the performance assessment of MOO solution techniques.

7.1 Introduction

In Chapters 3 and 5, it was pointed out that a very limited amount of literature is available on MICFMO. In particular, the appropriateness of several modern, Pareto-based *multiobjective optimisation algorithms* (MOAs) for solving the MICFMO problem is not known, especially in terms of their comparative performance. Here, MOAs refer to *a posteriori* methods designed to find an (approximate) set of Pareto optimal solutions to an MOP instance and therefore include multiobjective metaheuristics.

7.1.1 The comparative studies

Persuant to the second priority in this dissertation, as mentioned in §1.3, a total of eight modern state-of-the-art multiobjective metaheuristics are investigated within a comparative study to ascertain which of these MOAs is the most suitable in the context of constrained MICFMO. The metaheuristics considered in this comparative study consist of two evolutionary algorithms, namely the *nondominated sorting genetic algorithm II* (NSGA-II) [34] and the *strength Pareto evolutionary algorithm 2* (SPEA2) [248]; two swarm intelligence algorithms, namely *optimised multiobjective particle swarm optimisation* (OMOPSO) [174] and *Pareto ant colony optimisation* (P-ACO) [38]; two local search algorithms, namely *archived multiobjective simulated annealing* (AMOSA) [10] and *multiobjective variable neighbourhood search* (MOVNS) [65]; a probabilistic model-based algorithm called the *multiobjective optimisation using cross-entropy method* (MOOCEM) [15]; and finally, a *multiobjective harmony search* (MOHS) algorithm [196]. These MOAs have been sourced from different classes of metaheuristics in an attempt to represent the diversity of algorithms available in the literature. The general working of each metaheuristic is described in this chapter, along with the problem-specific modifications that the author had to make during their implementation for the purpose of solving constrained MICFMO problem instances. The solution representation (also known as the *encoding scheme*) adopted in each of the metaheuristics is that of a permutation-based encoding. This was a natural choice given that a reload configuration may easily be represented as a permutation decision vector, as was illustrated in §4.3.

Another area of research in which very little literature is available involves constraint handling techniques within the context of MOO [116]. A new constraint handling technique for MOO, based on a multiplicative penalty function, is proposed in this dissertation. Its effectiveness is also investigated (in the context of MICFMO) by comparing it to the well-known constrained-domination technique from the literature, originally proposed by Deb *et al.* [34]. Descriptions of each constraint handling technique is therefore presented in this chapter. Both constraint handling techniques are implemented in each of the metaheuristics discussed in this chapter, except for the P-ACO algorithm and the MOOCEM. In the case of the latter two metaheuristics, only the multiplicative penalty function technique is implemented, because the constrained-domination technique is not appropriate (for reasons that will be explained later).

Recall from §3.3.4 that multiobjective metaheuristics are approximate solution techniques. The nondominated set of solutions yielded by each of the aforementioned metaheuristics is therefore referred to as an *approximate Pareto set* and is denoted by $\widetilde{\mathcal{P}}_S$. Very often, the true Pareto set for a real-world optimisation problem instance such as an MICFMO problem instance is not available. No claim is therefore made in respect of closeness to optimality when referring to any approximation in this dissertation.

7.1.2 The Pareto rank of a solution

Multiobjective metaheuristics often employ a quantity known as the nondominated rank (as mentioned in §3.3.4) in order to distinguish between the quality of two or more solutions. This quantity is also known as the *Pareto rank* of a solution, and is denoted here by ρ. Due to the fact that this quantity features prominently in the metaheuristics discussed in this chapter, a more comprehensive description of Pareto ranks is presented in this section than was given in §3.3.4.

Apart from its Pareto front, there may exist several nondominated sub-fronts for an MOP instance into which objective vectors (and their corresponding decision vectors) may be classified. Each lower front is dominated by the one above it (in a maximisation paradigm). A vector may then be assigned a front depth which is related to the number of the front to which it belongs. This front depth is the nondominated rank, or Pareto rank, of the decision and objective vectors. Without loss of generality, the numbering of Pareto rank in this dissertation starts at zero, *i.e.* a vector in the Pareto front (or the first nondominated front) is assigned a Pareto rank $\rho = 0$.

In order to illustrate these concepts graphically, consider an MOP instance of the form (5.1) with two objectives. The objective vectors $\boldsymbol{z} = [z_1, z_2]$ corresponding to the decision vectors in the feasible region may be depicted as points in a Cartesian plane, as illustrated in Figure 7.1. The Pareto front and nondominated sub-fronts are shown in Figure 7.1, along with an example of Pareto rank assignment to the vectors.

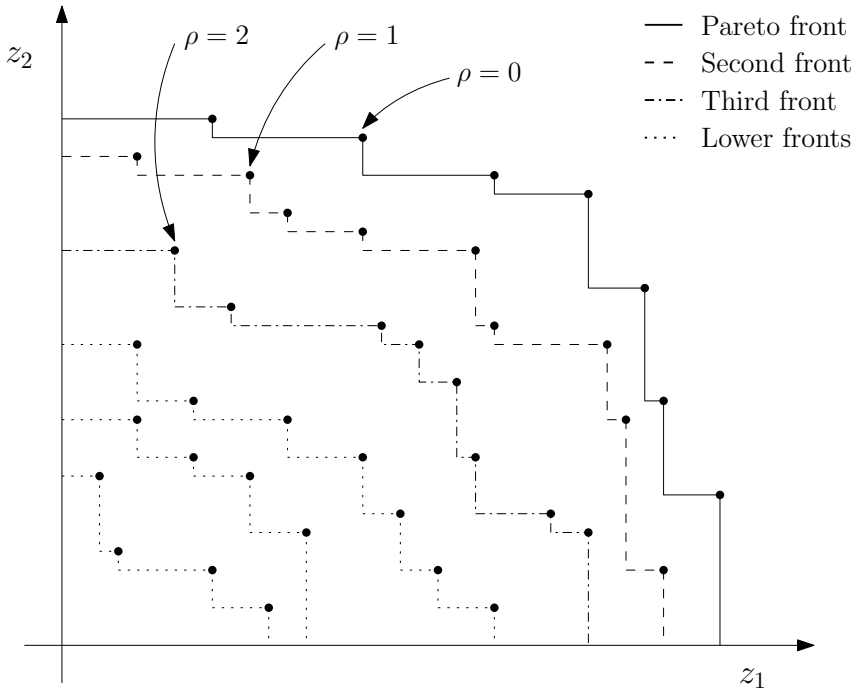


FIGURE 7.1: *Example of nondominated fronts and Pareto rank assignment.*

7.2 Constraint handling

The two constraint handling techniques for MOO mentioned above are described in this section. The first technique may be found in the literature, whereas the second technique is newly proposed in this dissertation.

Consider an MOP having the general form of the constrained MICFMO problem (4.1) with q objective functions. As before, let

$$G(\boldsymbol{x}) = \sum_{i=1}^{r} \max\left\{0, \frac{g_i(\boldsymbol{x}) - g_i^{\lim}}{|g_i^{\lim}|}\right\}$$

be the total scaled constraint violation associated with the inequality constraints in (4.1). Similarly, let

$$H(\boldsymbol{x}) = \sum_{j=1}^{s} \left|\frac{h_j(\boldsymbol{x}) - h_j^{\lim}}{h_j^{\lim}}\right|$$

be the total scaled constraint violation associated with the equality constraints in (4.1).

7.2.1 The constrained-domination principle

Deb *et al.* [34] proposed a constraint handling technique for MOO based on their so-called *constrained-domination principle* (CDP). The definition of traditional domination between two solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ is essentially modified according to this principle. Now, a solution $\boldsymbol{x}$ is said to "*constrained-dominate*" a solution $\boldsymbol{y}$ if any of the following conditions hold:

1. Solution $\boldsymbol{x}$ is feasible and solution $\boldsymbol{y}$ is infeasible,
2. Solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ are both infeasible, but solution $\boldsymbol{x}$ exhibits a smaller overall constraint violation, or
3. Solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ are both feasible, and $\boldsymbol{x}$ dominates $\boldsymbol{y}$.

In the constraint handling technique, this CDP may be employed during a comparison between two solutions instead of traditional domination. This principle may also be employed during the sorting of solutions into different nondominated fronts, *i.e.* nondominated sorting. An infeasible solution with a larger overall constraint violation than another solution is therefore sorted as a member of the next nondominated front [34]. In this dissertation, the overall constraint violation to employ in the CDP technique is calculated as $G(\boldsymbol{x}) + H(\boldsymbol{x})$.

This constraint handling technique is hereafter referred to as the CDP technique. One of its major advantages is the fact that no free parameters are employed. Accordingly, it is a problem-independent technique that does not require any parameter tuning.

7.2.2 The proposed multiplicative penalty function

The CDP technique cannot necessarily be adopted within any multiobjective metaheuristic. It is, for example, not suitable for the P-ACO algorithm and the MOOCEM. This prompted the development of a new constraint handling technique by the author, based on a *multiplicative penalty function* (MPF). In the proposed MPF technique, if a solution violates any constraint, a corresponding penalty value related to the magnitude of that violation is incurred. Then, an exponential function, taking the total constraint violation as argument, is used to calculate a scalar penalty value. The objective vector is finally penalised by multiplication by this scalar value.

The reader is reminded that all the objective functions in (4.1) are assumed, without loss of generality, to be maximised because an objective function may be transformed from a minimisation

paradigm to a maximisation paradigm by taking its negative value. It is further assumed here that, originally, all the objective functions are positive. In order to apply the MPF technique, it is now necessary to make a distinction between objectives that are intended for maximisation or minimisation. Let $f^{(+)}$ denote an objective function originally intended for maximisation, which therefore remains unaffected. Similarly, let $f^{(-)}$ denote an objective function, originally intended for minimisation, *after* it has been multiplied by -1 so as to transform it from a minimisation paradigm to a maximisation paradigm. If there are u unaffected objective functions and v transformed ones, then the objective vector may be written as $\boldsymbol{f}(\boldsymbol{x}) = [f_1^{(+)}, \ldots, f_u^{(+)}, f_{u+1}^{(-)}, \ldots, f_{u+v}^{(-)}]$, with $u + v = q$.

The exponential penalty function $P_m(\boldsymbol{x})$ employed in the MPF technique is also partitioned into a case for maximisation objectives, $P_m^{(+)}$, and a case for minimisation objectives, $P_m^{(-)}$. The function is defined as

$$P_m(\boldsymbol{x}) = \begin{cases} P_m^{(+)} = 2 - \exp\Big(\gamma\big(G(\boldsymbol{x}) + H(\boldsymbol{x})\big)\Big), & \text{for } f^{(+)}, \\ P_m^{(-)} = \exp\Big(\gamma\big(G(\boldsymbol{x}) + H(\boldsymbol{x})\big)\Big), & \text{for } f^{(-)}, \end{cases} \tag{7.1}$$

where γ is a strictly positive severity factor, as before.

A penalised objective vector $\boldsymbol{f}_P$ is then determined by multiplying the components of the objective vector in (4.1) by their corresponding penalty function values, that is $\boldsymbol{f}_P(\boldsymbol{x}) = P_m(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$. This effectively transforms a constrained MOP into an unconstrained optimisation problem whose objective vector is now given by

$$\boldsymbol{f}_P(\boldsymbol{x}) = \left[P_m^{(+)}(\boldsymbol{x})\left[f_1^{(+)}(\boldsymbol{x}), \ldots, f_u^{(+)}(\boldsymbol{x})\right], P_m^{(-)}(\boldsymbol{x})\left[f_{u+1}^{(-)}(\boldsymbol{x}), \ldots, f_{u+v}^{(-)}(\boldsymbol{x})\right]\right]. \tag{7.2}$$

An advantage of the MPF constraint handling technique is that all q objectives functions are penalised using a single scalar value, irrespective of their orders of magnitude. Traditional domination may then be applied within any solution method using the penalised objective vector $\boldsymbol{f}_P(\boldsymbol{x})$. Furthermore, only one free parameter has to be tuned within the technique.

7.3 Multiobjective evolutionary algorithms

Two prominent *multiobjective evolutionary algorithms* (MOEAs), namely the NSGA-II [34] and the SPEA2 [248] are described in this section. Although both algorithms have been in existence for more than a decade now, they are still regarded as state-of-the-art MOAs due to their excellent performance and wide-spread usage in the literature [45, 246]. This motivated their inclusion in the comparative study for solving MICFMO problem instances.

Numerous metaheuristics have been developed within the class of MOEAs and the reader is referred to Coello Coello *et al.* [29] for a good introductory text. MOEAs are mostly population-based metaheuristics with a generic algorithmic structure, functioning as follows:

1. *Initialisation*: Generate an initial population of solutions and evaluate their fitness.
2. *Selection for reproduction*: Create a mating pool of solutions for reproduction by selecting them from the current population. The fitness level of each parent solution represents its desirability of being selected, and is typically based on Pareto dominance concepts.
3. *Reproduction*: Apply variation operators (*e.g.* crossover and mutation) to the mating pool in order to generate a new population of offspring solutions, and evaluate their fitness.

4. *Selection for replacement*: Update the current population by selecting solutions from both the previous population and the offspring population. As before, the fitness level of each solution represents its desirability of being selected. MOEAs may also employ a secondary population, referred to as an archive, of nondominated solutions which may further influence the selection for replacement.

5. *Stopping criteria*: If the stopping criteria have been satisfied, terminate the algorithm and return the current population (or archive); otherwise, return to Step 2.

Both the NSGA-II and the SPEA2 follow this generic structure of MOEAs. In addition, solutions in both algorithms are selected for reproduction according to a technique known as *deterministic binary tournament selection*. According to this technique, two solutions are randomly[1] chosen from the population, and the one with the best fitness is then selected for reproduction. This tournament selection is repeated as many times as required [41].

Finally, since there are several dedicated permutation-based variation operators available in the evolutionary computation literature [213], a number of these are compared within the context of constrained MICFMO in order to indentify a suitable choice for implementation within the NSGA-II and the SPEA2. Three crossover operators are considered for comparison, namely the *partially matched/mapped crossover* (PMX), the *position-based crossover* (POS) and the *cycle crossover* (CX) operators [113], whereas two mutation operators are also considered, namely the *swap* and *scramble* operators [198]. A description of each operator may be found in Appendix C.

7.3.1 The NSGA-II

The NSGA-II was developed in 2002 by Deb *et al.* [34] as an improvement upon its predecessor, the NSGA [201]. Criticism against the NSGA involved the computational complexity of its nondominated sorting algorithm, its lack of elitism, and its need for a fitness sharing parameter, all of which have been addressed in the NSGA-II [34]. The generic structure of an MOEA is followed by the NSGA-II, but its fitness assignment and selection procedures distinguish it from other MOEAs.

Each solution in a population is assigned two attributes that constitute its fitness, namely a Pareto rank and a crowding distance. In order to determine the Pareto rank for each solution $\boldsymbol{a}$ in a population $\mathcal{P}$, a nondominated sorting procedure may be applied to $\mathcal{P}$. Deb *et al.* [34] developed the *fast nondominated sorting algorithm* (FNSA), which is used in NSGA-II, for this purpose. The algorithm has an improved computational complexity of $\mathcal{O}(qN^2)$ compared to its predecessor's computational complexity of $\mathcal{O}(qN^3)$, where q is the number of objectives and N is the population size.

According to the first stage of the FNSA, the domination count $n_{\boldsymbol{a}}$ for each solution $\boldsymbol{a} \in \mathcal{P}$ is determined (*i.e.* the number of solutions that dominate solution $\boldsymbol{a}$), along with the set $\mathcal{S}_{\boldsymbol{a}}$ containing all the solutions that are dominated by solution $\boldsymbol{a}$. By definition, solutions in the first nondominated front have a domination count of zero, and are therefore assigned a Pareto rank of zero. Furthermore, those solutions for which $n_{\boldsymbol{a}} = 0$ are then placed in a separate set $\mathcal{F}_1$. During the second stage of the FNSA, each solution $\boldsymbol{a}$ in $\mathcal{F}_1$ is visited, and for each solution $\boldsymbol{b}$ in the corresponding set $\mathcal{S}_{\boldsymbol{a}}$, the domination count $n_{\boldsymbol{b}}$ is decreased by one. This decrement effectively discounts the contribution of solution $\boldsymbol{a}$ to the domination count of solution $\boldsymbol{b}$. Subsequently, solutions that are part of the second nondominated front will now have a domination count of

[1] Unless specifically stated otherwise, where any reference is made in this chapter to *random* selection, it is assumed that a uniform distribution is employed during selection.

zero. Those solutions are then assigned a Pareto rank of one and placed in a separate set $\mathcal{F}_2$. This procedure is repeated until all the fronts have been identified with the solutions ranked correspondingly. A pseudo-code listing of the FNSA, with minor adjustments in respect of how it appeared in [34], is presented in Algorithm 7.1. In terms of fitness, solutions in the population associated with a lower Pareto rank are preferred.

Algorithm 7.1: Fast nondominated sorting algorithm (FNSA) [34]

Input : A population of solutions $\mathcal{P}$.
Output: A partitioning of the input-population into n successive nondominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_n$, with each solution $\boldsymbol{x}$ assigned a Pareto rank $\rho_{\boldsymbol{x}}$.

$\mathcal{F}_1 \leftarrow \emptyset$
for each $\boldsymbol{a} \in \mathcal{P}$ **do**
  $\mathcal{S}_{\boldsymbol{a}} \leftarrow \emptyset$
  $n_{\boldsymbol{a}} \leftarrow 0$
  for each $\boldsymbol{b} \in \mathcal{P}$ **do**
    if $\boldsymbol{a} \succ \boldsymbol{b}$ **then** // If $\boldsymbol{a}$ dominates $\boldsymbol{b}$
      $\mathcal{S}_{\boldsymbol{a}} \leftarrow \mathcal{S}_{\boldsymbol{a}} \cup \{\boldsymbol{b}\}$ // Add $\boldsymbol{b}$ to set of solutions dominated by $\boldsymbol{a}$
    else if $\boldsymbol{b} \succ \boldsymbol{a}$ **then**
      $n_{\boldsymbol{a}} \leftarrow n_{\boldsymbol{a}} + 1$ // Increment the domination counter of $\boldsymbol{a}$
    end if
  end for
  if $n_{\boldsymbol{a}} = 0$ **then** // $\boldsymbol{a}$ belongs to the first front
    $\rho_{\boldsymbol{a}} \leftarrow 0$
    $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{\boldsymbol{a}\}$
  end if
end for
$i \leftarrow 1$ // Initialise the front counter
while $\mathcal{F}_i \neq \emptyset$ **do**
  $\mathcal{Q} \leftarrow \emptyset$ // Store the solutions of the next front
  for each $\boldsymbol{a} \in \mathcal{F}_i$ **do**
    for each $\boldsymbol{b} \in \mathcal{S}_{\boldsymbol{a}}$ **do**
      $n_{\boldsymbol{b}} \leftarrow n_{\boldsymbol{b}} - 1$
      if $n_{\boldsymbol{b}} = 0$ **then** // $\boldsymbol{b}$ belongs to the next front
        $\rho_{\boldsymbol{b}} \leftarrow i$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\boldsymbol{b}\}$
      end if
    end for
  end for
  $i \leftarrow i + 1$
  $\mathcal{F}_i \leftarrow \mathcal{Q}$
end while

The crowding distance associated with a solution is a measure of the density of solutions surrounding it within the same nondominated front. In order to calculate the crowding distances, a nondominated population $\mathcal{I}$ has to be sorted in ascending order of magnitude for each objective function $k \in \{1, \ldots, q\}$. Let $\mathcal{I}[i]_{\text{dist}}$ denote the crowding distance of the i-th solution in the sorted population. Furthermore, let $\ell = |\mathcal{I}|$, where $|\cdot|$ denotes the cardinality of a set. The solutions at the two endpoints of the population are assigned a crowding distance of infinity, *i.e.* $\mathcal{I}[1]_{\text{dist}} = \mathcal{I}[\ell]_{\text{dist}} = \infty$. The crowding distance of each intermediate solution i, however, is increased by the distance between the function values in the k-th objective of its two neighbouring solutions $i-1$ and $i+1$, normalised by the current range of objective k. Accordingly, crowding distance is accumulated over the q objectives. A pseudo-code listing of the crowding

distance assignment algorithm, similar to how it appeared in [34], is presented in Algorithm 7.2. The function value of objective k, corresponding to the i-th solution in $\mathcal{I}$ is denoted by $\mathcal{I}[i].f_k$ in Algorithm 7.2. In terms of fitness, solutions within the same nondominated front associated with a larger crowding distance are preferred. Such solutions are in a less "crowded" region of the objective space and retaining them, therefore, promotes the preservation of diversity [34].

Algorithm 7.2: Crowding distance assignment algorithm [34]

Input : A population of nondominated solutions $\mathcal{I}$.
Output : The crowding distance associated with each solution, $\mathcal{I}[1]_{\text{dist}}, \ldots, \mathcal{I}[\ell]_{\text{dist}}$.

$\ell \leftarrow |\mathcal{I}|$ // Number of solutions in $\mathcal{I}$
for each $i \in \mathcal{I}$ **do**
  $\mathcal{I}[i]_{\text{dist}} \leftarrow 0$ // Initialise crowding distance
end for
for each *objective* $k \in \{1, \ldots, q\}$ **do**
  $\mathcal{I} \leftarrow \text{sort}(\mathcal{I}, k)$ // Sort population using value of objective k
  $\mathcal{I}[1]_{\text{dist}} \leftarrow \infty$ // So that endpoints are always selected
  $\mathcal{I}[\ell]_{\text{dist}} \leftarrow \infty$
  for $i \leftarrow 2$ **to** $(\ell - 1)$ **do** // For all intermediate solutions
    $\mathcal{I}[i]_{\text{dist}} \leftarrow \mathcal{I}[i]_{\text{dist}} + (\mathcal{I}[i+1].f_k - \mathcal{I}[i-1].f_k)/(f_k^{\max} - f_k^{\min})$
  end for
end for

Given that each solution $\boldsymbol{a}$ is assigned a Pareto rank $\rho_{\boldsymbol{a}}$ and a crowding distance $\boldsymbol{a}_{\text{dist}}$, two solutions may be compared according to the so-called *crowded comparison operator* (denoted by $\succ_{\text{cc}}$). According to this operator, if two solutions $\boldsymbol{a}$ and $\boldsymbol{b}$ have differing Pareto ranks, the solution with the lower rank is preferred (more fit for selection). Otherwise, where two solutions have equal rank, the solution with the greater crowding distance is preferred. Mathematically, the operator is described as

$$\boldsymbol{a} \succ_{\text{cc}} \boldsymbol{b} \quad \text{if} \quad \begin{cases} \rho_{\boldsymbol{a}} < \rho_{\boldsymbol{b}}, \text{ or} \\ \rho_{\boldsymbol{a}} = \rho_{\boldsymbol{b}} \text{ and } \boldsymbol{a}_{\text{dist}} > \boldsymbol{b}_{\text{dist}}. \end{cases}$$

By using the crowded comparison operator within NSGA-II, the selection process of the algorithm favours the exploration of diverse solutions [34].

The full NSGA-II may now be described, and a pseudo-code listing thereof is presented in Algorithm 7.3. The NSGA-II starts by randomly generating an initial parent population $\mathcal{P}_0$ of size N. This population is then ranked and sorted using the FNSA. Initially, each solution is assigned a fitness equal to its Pareto rank only. An offspring population $\mathcal{Q}_0$ of size N is then generated using a deterministic binary tournament selection procedure (assuming minimisation of fitness, since lower Pareto ranks are preferred), along with the relevant crossover and mutation operators. At this point, the generation counter t is set to zero, and the following procedure is iterated until the relevant stopping criterion has been met (*e.g.* a maximum number of generations reached):

1. Create a combined population $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$ from the parent and offspring populations.
2. Rank and sort population $\mathcal{R}_t$ into nondominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_n$ using the FNSA, and calculate the crowding distance of each solution.
3. Create the next population $\mathcal{P}_{t+1}$ by including all solutions from the first front $\mathcal{F}_1$, then all solutions from the second front $\mathcal{F}_2$, and so forth, until the inclusion of all solutions from the next front would result in a population size greater than N. In order to limit the size

of $\mathcal{P}_{t+1}$ to N, the solutions in this next front are sorted in decreasing order of crowding distance. Solutions from this sorted front are then included one-by-one until $|\mathcal{P}_{t+1}| = N$.

4. Create a mating pool of solutions by using a deterministic binary tournament selection procedure based on the crowded comparison operator ($\succ_{\text{cc}}$).

5. Generate the next offspring population $\mathcal{Q}_{t+1}$ using solutions from the mating pool, along with the relevant crossover and mutation operators.

6. Increment the value of the generation counter $t \leftarrow t + 1$.

Algorithm 7.3: Nondominated sorting genetic algorithm II (NSGA-II) [34]

Input : An MOP (possibly constrained), a population size N, a maximum number of generations $t_{\max}$, a crossover probability p_c, and a mutation probability p_m.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Randomly generate an initial population $\mathcal{P}_0$ of size N
Rank and sort $\mathcal{P}_0$ using the FNSA in Algorithm 7.1
Assign a fitness value to each solution in $\mathcal{P}_0$ equal to its Pareto rank
Create a mating pool of solutions from $\mathcal{P}_0$ by means of deterministic binary tournament selection (assuming minimisation of fitness)
Generate an offspring population $\mathcal{Q}_0$ of size N using the mating pool, a crossover operator (with p_c) and a mutation operator (with p_m)
$t \leftarrow 0$
while $t < t_{max}$ **do**
    $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$
    Partition $\mathcal{R}_t$ into nondominated fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ using the FNSA in Algorithm 7.1
    $\mathcal{P}_{t+1} \leftarrow \emptyset$
    $i \leftarrow 1$
    while $|\mathcal{P}_{t+1}| < N$ **do**
        if $|\mathcal{P}_{t+1}| + |\mathcal{F}_i| \leq N$ **then**
            $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i$
            $i \leftarrow i + 1$
        else
            Calculate the crowding distance for each solution in $\mathcal{F}_i$ using Algorithm 7.2
            Sort $\mathcal{F}_i$ in decreasing order of crowding distance
            $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \{\text{the first } N - |\mathcal{P}_{t+1}| \text{ solutions in } \mathcal{F}_i\}$
        end if
    end while
    Calculate the crowding distance for each solution in $\mathcal{P}_{t+1}$ using Algorithm 7.2
    Create a mating pool of solutions from $\mathcal{P}_{t+1}$ by means of deterministic binary tournament crowded comparison selection
    Generate an offspring population $\mathcal{Q}_{t+1}$ of size N using the mating pool, a crossover operator (with p_c) and a mutation operator (with p_m)
    $t \leftarrow t + 1$
end while
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{P}_{t_{\max}}$

7.3.2 The SPEA2

The SPEA2 was developed in 2001 by Zitzler *et al.* [248], also as an improvement upon its predecessor, the SPEA [249, 250]. The proposed improvements consist of a new fitness assignment strategy, a density estimation technique, and a new archive truncation technique, since

the algorithm employs an archive of nondominated solutions. Fitness values are assigned at each generation to solutions in both the current population and the archive before selection occurs. This new fitness assignment is based on domination principles and a density estimation technique in order to promote diversity.

Consider the population $\mathcal{P}_t$, of size N, at a given iteration t during execution of the SPEA2, along with the archive $\mathcal{A}_t$ of size $\overline{N}$. Let $\mathcal{Q}_t \leftarrow \mathcal{P}_t \cup \mathcal{A}_t$ be the combined set of the population and the archive. The first step in calculating the fitness of each solution $\boldsymbol{a} \in \mathcal{Q}_t$ is to determine its associated *strength* value $S(\boldsymbol{a})$. The strength of a solution represents the total number of solutions that it dominates, and is calculated as $S(\boldsymbol{a}) = \left|\{\boldsymbol{b} \,|\, \boldsymbol{b} \in \mathcal{Q}_t, \boldsymbol{a} \succ \boldsymbol{b}\}\right|$. Consider next the set of all solutions $\boldsymbol{b} \in \mathcal{Q}_t$ that dominate solution $\boldsymbol{a}$, and denote it by $\mathcal{D}_{\boldsymbol{a}}$. The so-called *raw fitness* of solution $\boldsymbol{a}$ is then calculated as $R(\boldsymbol{a}) = \sum_{\boldsymbol{b} \in \mathcal{D}_{\boldsymbol{a}}} S(\boldsymbol{b})$. Note that this raw fitness is to be minimised, and any nondominated solution has a raw fitness of zero.

Since different solutions may have identical raw fitness values, density information is incorporated as a distinguishing factor. For each solution $\boldsymbol{a}$, the distance (in objective space) to every other solution $\boldsymbol{b} \in \mathcal{Q}$ is determined, and the list of distances is then sorted in increasing order of magnitude. The distance to the k-th nearest neighbour of solution $\boldsymbol{a}$, denoted by $\sigma_{\boldsymbol{a}}^k$, then corresponds to the k-th value in the list of distances and represents the density information to use. The Euclidean distance has been selected here as distance measure. Typically, the value of k is selected as the square root of the sample size, thus $k = \sqrt{N + \overline{N}}$. The *density* of solution $\boldsymbol{a}$ is then calculated as $D(\boldsymbol{a}) = 1/(\sigma_{\boldsymbol{a}}^k + 2)$ and added to the raw fitness. Note that $0 < D(\boldsymbol{a}) < 1$ [248]. Finally, the fitness of solution $\boldsymbol{a}$ is given by $F(\boldsymbol{a}) = R(\boldsymbol{a}) + D(\boldsymbol{a})$.

According to the new archive truncation technique within the SPEA2, the number of solutions in the archive is kept constant during execution of the algorithm and the technique prevents extremal solutions (*i.e.* boundary solutions) from being removed [248]. The technique works as follows. First, all the nondominated solutions in $\mathcal{Q}_t$, *i.e.* those solutions with $F(\boldsymbol{a}) < 1$, are copied into the new archive $\mathcal{A}_{t+1}$. If $|\mathcal{A}_{t+1}| = \overline{N}$, then the selection for replacement is complete. Otherwise, one of two cases may be experienced. In the first case, if $|\mathcal{A}_{t+1}| < \overline{N}$, the archive is simply filled with the best $\overline{N} - |\mathcal{A}_{t+1}|$ dominated solutions (according to fitness) remaining in $\mathcal{Q}_t$. In the second case, however, if $|\mathcal{A}_{t+1}| > \overline{N}$, then a truncation procedure is applied which iteratively eliminates solutions from the archive until $|\mathcal{A}_{t+1}| = \overline{N}$. During each elimination iteration, the solution with the smallest Euclidean distance to another solution (in objective space) is selected for removal. In the event that some solutions have equal distances, the tie is broken by considering the second smallest distance (and the third, and the fourth, *etc.*). Since any boundary solution will always have some k-th distance that is larger than the k-th distance of another solution closest to it, this technique guarantees that extremal solutions will not be removed.

The full SPEA2 may now be described. The algorithm starts by randomly generating an initial population $\mathcal{P}_0$ of size N, and initialising an empty archive $\mathcal{A}_0 \leftarrow \emptyset$. The generation counter t is set to zero, and the following procedure is then iterated until the relevant stopping criterion has been met (*e.g.* a maximum number of generations reached):

1. Create a combined set of solutions $\mathcal{Q}_t \leftarrow \mathcal{P}_t \cup \mathcal{A}_t$ from the population and the archive.

2. Assign a fitness to each solution in $\mathcal{Q}_t$ by calculating the strength, raw fitness and density values of the solutions.

3. Create the next archive $\mathcal{A}_{t+1}$ by transferring the nondominated solutions in $\mathcal{Q}_t$ to the archive. If $|\mathcal{A}_{t+1}| < \overline{N}$, then fill $\mathcal{A}_{t+1}$ with the best solutions remaining in $\mathcal{Q}_t$. Otherwise, if

$|\mathcal{A}_{t+1}| > \overline{N}$, eliminate solutions from the archive using the truncation procedure described above.

4. Create a mating pool of solutions from $\mathcal{A}_{t+1}$ by using deterministic binary tournament selection according to fitness.

5. Generate the next population $\mathcal{P}_{t+1}$ using solutions from the mating pool, along with the relevant crossover and mutation operators.

6. Increment the value of the generation counter $t \leftarrow t + 1$.

A pseudo-code listing of the SPEA2 is presented in Algorithm 7.4. Note that this pseudo-code listing differs slightly from the original algorithm presented in [248] — the test for algorithmic termination has been moved and now occurs after the creation of the next population (as is the case for the NSGA-II), as opposed to before creation.

Algorithm 7.4: Strength Pareto evolutionary algorithm 2 (SPEA2) [248]

Input : An MOP (possibly constrained), a population size N, an archive size $\overline{N}$, a maximum number of generations $t_{\max}$, a crossover probability p_c, and a mutation probability p_m.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Randomly generate an initial population $\mathcal{P}_0$ of size N
Initialise an empty archive $\mathcal{A}_0 \leftarrow \emptyset$
$t \leftarrow 0$
while $t < t_{max}$ **do**
  $\mathcal{Q}_t \leftarrow \mathcal{P}_t \cup \mathcal{A}_t$
  Calculate the strength value $S(\boldsymbol{a})$, the raw fitness $R(\boldsymbol{a})$ and the density $D(\boldsymbol{a})$ for each solution $\boldsymbol{a} \in \mathcal{Q}_t$
  Determine the fitness of each solution $\boldsymbol{a} \in \mathcal{Q}_t$ according to $F(\boldsymbol{a}) = R(\boldsymbol{a}) + D(\boldsymbol{a})$
  Copy all the nondominated solutions in $\mathcal{Q}_t$ (*i.e.* those for which $F(\boldsymbol{a}) < 1$) to the next archive $\mathcal{A}_{t+1}$
  if $|\mathcal{A}_{t+1}| < \overline{N}$ **then**
    Sort the solutions in $\mathcal{Q}_t$ according fitness in increasing order of magnitude
    Fill $\mathcal{A}_{t+1}$ with the first $\overline{N} - |\mathcal{A}_{t+1}|$ solutions for which $F(\boldsymbol{a}) \geq 1$
  else if $|\mathcal{A}_{t+1}| > \overline{N}$ **then**
    Iteratively eliminate solutions from $\mathcal{A}_{t+1}$ by means of the nearest neighbour truncation procedure until $|\mathcal{A}_{t+1}| = \overline{N}$
  end if
  Create a mating pool of solutions from $\mathcal{A}_{t+1}$ by using deterministic binary tournament selection according to fitness
  Generate the next population $\mathcal{P}_{t+1}$ of size N using the mating pool, a crossover operator (with p_c) and a mutation operator (with p_m)
  $t \leftarrow t + 1$
end while
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}_{t_{\max}}$

7.4 Multiobjective swarm intelligence algorithms

Several different metaheuristics reside within the class of swarm intelligence algorithms, with PSO and ACO arguably the most established approaches. In this dissertation, a multiobjective variant of each approach is employed as part of the comparative study for solving constrained MICFMO problem instances.

7.4.1 The OMOPSO algorithm

In an experimental comparison between six state-of-the-art multiobjective PSO algorithms, it was found that the OMOPSO algorithm, developed by Reyes Sierra and Coello Coello [174], performed the best in the context of several benchmark problems [44]. This experimental outcome motivated the selection of the OMOPSO algorithm for use in this dissertation.

In the algorithm, a regular swarm of particles is employed together with a so-called *leader swarm*, consisting of nondominated solutions only, as well as an archive employing the concept of ϵ-dominance[2]. Mutation operators are also borrowed from the evolutionary computation literature and applied within the OMOPSO algorithm. Once the positions of the particles in the regular swarm have been updated using the flight operators, the swarm is partitioned into three subsets. The particles in the first subset remain as is. Different mutation operators are, however, applied to the particles in the second and third subsets.

A key feature of the OMOPSO algorithm is that different global best positions are selected for the particles during every iteration. The solutions in the leader swarm are each associated with a crowding distance, as defined in the NSGA-II, in order to differentiate between the quality of different solutions. A deterministic binary tournament selection procedure based on crowding distance is then performed during each iteration in order to select a global best position for each particle.

Recall from §3.3.3 the working of a basic single-objective PSO algorithm. Each particle is associated with a position and a velocity. Let $\boldsymbol{x}_i^t$ denote the position of particle i during iteration t and, similarly, let $\boldsymbol{v}_i^t$ denote its velocity. The flight operator updating the position of a particle is given by

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}. \tag{7.3}$$

Assuming a fully-connected network topology, let $\boldsymbol{x}_{\text{pb},i}$ denote the personal best position of particle i and let $\boldsymbol{x}_{\text{gb}}$ denote the global best position of the swarm. Then, using the same notation as in [174], the flight operator updating the velocity is given by

$$\boldsymbol{v}_i^{t+1} = W\boldsymbol{v}_i^t + C_1 r_1 \left(\boldsymbol{x}_{\text{pb},i} - \boldsymbol{x}_i^t\right) + C_2 r_2 \left(\boldsymbol{x}_{\text{gb}} - \boldsymbol{x}_i^t\right), \tag{7.4}$$

where W is the inertia weight, C_1 and C_2 are learning factors, and $r_1, r_2 \in [0, 1]$ are random numbers.

The full working of the OMOPSO algorithm may now be described, and a pseudo-code listing thereof is presented in Algorithm 7.5. The OMOPSO algorithm starts by randomly generating an initial regular swarm $\mathcal{R}$ consisting of N particles. Set the personal best position for each particle i as $\boldsymbol{x}_{\text{pb},i} \leftarrow \boldsymbol{x}_i^0$. The nondominated solutions in $\mathcal{R}$ are then identified and copied into the leader swarm $\mathcal{L}_0$, of maximum size N, and the crowding distance associated with each solution is calculated using Algorithm 7.2. The ϵ-archive $\mathcal{A}_0^\epsilon$ is determined next using the solutions in the leader swarm. At this point, the iteration counter t is set to zero, and the following procedure is iterated until the relevant stopping criterion has been met (*e.g.* a maximum number of iterations reached):

1. For each particle $i \in \mathcal{R}$, select a global best position $\boldsymbol{x}_{\text{gb}}$ from the leader swarm $\mathcal{L}_t$ according to deterministic binary tournament selection based on crowding distance. Update the velocity and position of particle i according to (7.4) and (7.3), respectively.

[2] A decision vector $\boldsymbol{x}_1$ is said to ϵ-dominate another decision vector $\boldsymbol{x}_2$ if $(1+\epsilon) f_i(\boldsymbol{x}_1) \geq f_i(\boldsymbol{x}_2)$ for all $i = 1, \ldots, q$ (in a maximisation problem).

2. Partition $\mathcal{R}$ into three subsets and apply different mutation operators to the particles in the second and third subsets, respectively.

3. For each particle $i \in \mathcal{R}$, update its personal best position according to the following procedure: if $\boldsymbol{x}_i^{t+1}$ dominates $\boldsymbol{x}_{\text{pb},i}$ or if $\boldsymbol{x}_i^{t+1}$ and $\boldsymbol{x}_{\text{pb},i}$ are nondominated with respect to each other, then set $\boldsymbol{x}_{\text{pb},i} \leftarrow \boldsymbol{x}_i^{t+1}$.

4. Determine the new leader swarm $\mathcal{L}_{t+1}$ by identifying the nondominated solutions in $\mathcal{L}_t \cup \mathcal{R}$, and then calculate the corresponding crowding distance for each solution using Algorithm 7.2. If $|\mathcal{L}_{t+1}| > N$, then eliminate the worst solutions according to crowding distance until $|\mathcal{L}_{t+1}| = N$.

5. Update the ϵ-archive $\mathcal{A}_{t+1}^{\epsilon}$.

6. Increment the value of the iteration counter $t \leftarrow t + 1$.

Since the OMOPSO algorithm was originally proposed for solving continuous optimisation problems, its flight operators may have to be modified in order to be applicable to the MICFMO

Algorithm 7.5: Optimised multiobjective particle swarm optimisation (OMOPSO) [174]

Input : An MOP (possibly constrained), a swarm size N, a maximum number of iterations $t_{\max}$, a mutation probability p_m, and an ϵ-value.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Randomly generate an initial regular swarm $\mathcal{R}$ consisting of N particles
Set the personal best position for each particle i as $\boldsymbol{x}_{\text{pb},i} \leftarrow \boldsymbol{x}_i^0$
Identify the nondominated solutions in $\mathcal{R}$ and copy them into the initial leader swarm $\mathcal{L}_0$
Calculate the crowding distance for each solution in $\mathcal{L}_0$ using Algorithm 7.2
Determine the initial ϵ-archive $\mathcal{A}_0^{\epsilon}$ using the solutions in $\mathcal{L}_0$
$t \leftarrow 0$
while $t < t_{max}$ **do**
    for each *particle* $i \in \mathcal{R}$ **do**
        Select $\boldsymbol{x}_{\text{gb}}$ from $\mathcal{L}_t$ according to deterministic binary tournament selection based on crowding distance
        Apply the velocity flight operator (7.4) to obtain $\boldsymbol{v}_i^{t+1}$
        Apply the position flight operator (7.3) to obtain $\boldsymbol{x}_i^{t+1}$
    end for
    Partition $\mathcal{R}$ into three subsets of equal size $\mathcal{Q}_1, \mathcal{Q}_2$ and $\mathcal{Q}_3$
    Apply different mutation operators to $\mathcal{Q}_2$ and $\mathcal{Q}_3$ (both with p_m)
    for each *particle* $i \in \mathcal{R}$ **do**
        if $\boldsymbol{x}_i^{t+1} \succ \boldsymbol{x}_{pb,i}$ **or** $\boldsymbol{x}_i^{t+1}, \boldsymbol{x}_{pb,i}$ *nondominated* **then**
            $\boldsymbol{x}_{\text{pb},i} \leftarrow \boldsymbol{x}_i^{t+1}$
        end if
    end for
    Identify the nondominated solutions in $\mathcal{L}_t \cup \mathcal{R}$ and copy them to $\mathcal{L}_{t+1}$
    Calculate the crowding distance for each solution in $\mathcal{L}_{t+1}$ using Algorithm 7.2
    if $|\mathcal{L}_{t+1}| > N$ **then**
        Sort $\mathcal{L}_{t+1}$ in decreasing order according to crowding distance
        Eliminate the last $|\mathcal{L}_{t+1}| - N$ solutions from $\mathcal{L}_{t+1}$
    end if
    Update $\mathcal{A}_{t+1}^{\epsilon}$ using $\mathcal{L}_{t+1} \cup \mathcal{A}_t^{\epsilon}$
    $t \leftarrow t + 1$
end while
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}_{t_{\max}}^{\epsilon}$

problem. Two permutation-based approaches available in the literature are compared within the context of constrained MICFMO in order to find a suitable choice for implementation within the OMOPSO algorithm. The first approach is a method known as *random keys* [13]. In the second approach, proposed by Hu *et al.* [86], the position flight operator is redefined using a probabilistic interpretation of the velocity. A description of each approach may be found in Appendix C.

In this dissertation, two modifications have made to the OMOPSO algorithm. First, the regular swarm is only partitioned into two subsets (instead of three) and the swap mutation operator is applied to the second subset. As before, the first subset does not undergo mutation. Secondly, as was the case in the experimental comparison in [44], it decided not to employ ϵ-archive. Instead, the algorithm yields the final leader swarm as an approximate Pareto set.

7.4.2 The P-ACO algorithm

The P-ACO algorithm was developed by Doerner *et al.* [38] and it recently featured in a taxonomy of representative multiobjective ACO algorithms available in the literature [4] — hence its usage in this dissertation. In the algorithm, a single colony of ants and an archive of nondominated solutions are employed. A separate pheromone matrix is adopted for each objective, while a single heuristic information matrix is utilised. The transition rule within the P-ACO algorithm is based on that of the *ant colony system* (ACS) [41], along with a random aggregation of the pheromone matrices. Finally, pheromone updates are performed locally whenever an ant has traversed an edge, and globally after all the ants have completed their paths.

Assume that an MOP may be represented as a problem in which an optimal path along a complete graph is sought. In this context, the P-ACO algorithm then works as follows. A seperate pheromone matrix $\boldsymbol{\tau}^k = [\tau_{ij}^k]$ is initialised for each objective $k = 1, \ldots, q$. Furthermore, the archive of nondominated solution $\mathcal{A}_0$ is intialised as an empty set. Then, during each iteration of the algorithm, N ants traverse the graph, each constructing a complete path. Given a single heuristic information matrix $\boldsymbol{\eta} = [\eta_{ij}]$, an ant at a vertex i will move to vertex j according to the transition rule

$$j = \begin{cases} \arg\max_{j \in \mathcal{N}_i} \left\{ \left(\sum_{k=1}^{q} w_k \, \tau_{ij}^k \right)^{\alpha} (\eta_{ij})^{\beta} \right\}, & \text{if } r \leq r_0, \\ \hat{j}, & \text{otherwise,} \end{cases} \tag{7.5}$$

where $\mathcal{N}_i$ is the set of neighbouring vertices of vertex i that have not been visited yet by the ant, w_k is a random weighting coefficient for objective k, and α and β are bias parameters. Furthermore, r is a random number in the range $[0, 1)$ and r_0 is a free parameter representing the probability that the next vertex should be selected as the one with the highest aggregate value of pheromone and heuristic information [38]. The vertex $\hat{j}$ in (7.5) is selected with probability

$$p_{ij} = \begin{cases} \dfrac{\left(\sum_{k=1}^{q} w_k \, \tau_{ij}^k \right)^{\alpha} (\eta_{ij})^{\beta}}{\sum_{\ell \in \mathcal{N}_i} \left(\left(\sum_{k=1}^{q} w_k \, \tau_{i\ell}^k \right)^{\alpha} (\eta_{i\ell})^{\beta} \right)}, & \text{if } j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{7.6}$$

As mentioned earlier, a local pheromone update is performed whenever an ant traverses an edge in the graph. Suppose that an edge (i, j) has been traversed. Then each pheromone matrix is (locally) updated according to

$$\tau_{ij}^k = (1 - \varrho)\tau_{ij}^k + \varrho \tau_0, \tag{7.7}$$

where τ_0 is the initial pheromone level, and ϱ is the evaporation rate.

A global pheromone update is, however, only performed once all the ants have completed their path constructions. Only the best and second-best solutions obtained by the ants (in respect of each objective) are utilised during this update. Let $\boldsymbol{s}_{\text{b}}^k$ and $\boldsymbol{s}_{\text{sb}}^k$ denote the paths corresponding to the best and second-best solutions, respectively. The pheromone matrix for each objective k is then (globally) updated according to

$$\tau_{ij}^k = (1-\varrho)\tau_{ij}^k + \varrho\Delta\tau_{ij}^k, \tag{7.8}$$

where $\Delta\tau_{ij}^k$ is related to $\boldsymbol{s}_{\text{b}}^k$ and $\boldsymbol{s}_{\text{sb}}^k$, and may be calculated as

$$\Delta\tau_{ij}^k = \begin{cases} 10, & \text{if } (i,j) \in \boldsymbol{s}_{\text{b}}^k, \\ 5, & \text{if } (i,j) \in \boldsymbol{s}_{\text{sb}}^k, \\ 0, & \text{otherwise.} \end{cases} \tag{7.9}$$

Finally, at the end of each iteration t, the archive $\mathcal{A}_t$ is updated. The algorithm terminates when the relevant stopping criterion has been met (*e.g.* a maximum number of iterations reached). A pseudo-code listing of the P-ACO algorithm is presented in Algorithm 7.6.

Algorithm 7.6: Pareto ant colony optimisation (P-ACO) [38].

Input : An MOP (possibly constrained), the number of ants N, a maximum number of iterations $t_{\max}$, bias parameters α and β, a selection probability r_0 , the evaporation rate ϱ, an initial pheromone level τ_0, and a heuristic information matrix $\boldsymbol{\eta}$.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

```
Initialise a pheromone matrix τ^k for each objective k = 1, ..., q using τ_0
Set A_0 ← ∅
t ← 0
while t < t_max do
    for each ant i ∈ {1, ..., N} do
        Randomly select a starting vertex
        while tour incomplete do
            Select the next vertex according to the transition rules in (7.5) and (7.6)
            Perform a local pheromone update according to (7.7)
        end while
    end for
    for each objective k ∈ {1, ..., q} do
        Identify the best and second-best solutions in respect of objective k
        Perform a global pheromone update according to (7.8) and (7.9)
    end for
    Identify the nondominated solutions in A_t ∪ {ants} and copy them to A_{t+1}
    t ← t + 1
end while
P̃_S ← A_{t_max}
```

In order to apply the P-ACO algorithm to the MICFMO problem, a corresponding ant model is required, while the heuristic information matrix has to be specified as well. An ant model similar to the one described in §3.3.3 is adopted for the MICFMO problem. In this model, vertices in the graph correspond to the fuel assemblies whereas the edges now depend on the loading positions. An ordered mapping is employed to assign a fuel assembly selected by an ant traversing the graph to a specific loading position, thereby ensuring that a complete path translates into a valid reload configuration. An edge, therefore, represents the possibility of a fuel assembly being selected and assigned to a specific loading position. As such, the transition

rules, pheromone updates and heuristic information are now applicable to assignments of fuel assemblies to loading positions.

Since a single heuristic information matrix cannot cater simultaneously to multiple objectives in the MICFMO problem, safety-related constraint information is rather opted for in this dissertation. Recall that the well-known HMLF fuel management strategy attempts to flatten the power distribution over a reactor core and, in doing so, typically yields a safe reload configuration. This strategy served as inspiration for constructing the heuristic information matrix for a reactor. The construction procedure is as follows. First, a normalised thermal neutron flux profile in the fuel loading positions of the reactor under consideration is calculated using a core simulator. This may be achieved by using a core loaded with fresh fuel assemblies only in order to obtain a typical flux profile. An $n\times1$ column vector $\boldsymbol{u}$ is formed in which u_i corresponds to the normalised flux value for fuel loading position $i \in \{1,\ldots,n\}$, where n is the number of positions. Next, the ^{235}U mass distribution in the m fuel assemblies is normalised and an $m\times1$ column vector $\boldsymbol{v}$ is formed in which v_j corresponds to the normalised mass value for fuel assembly $j \in \{1,\ldots,m\}$. Finally, the outer product of $\boldsymbol{u}$ and $\boldsymbol{v}$ is calculated to form an $n\times m$ matrix and the reciprocal of every matrix element is taken. Once this matrix is normalised again, it corresponds to a heuristic information matrix in which larger values represent preferred assignments of fuel assemblies to loading positions.

It is noted that the P-ACO algorithm does not utilise Pareto dominance *per se* during its execution. Accordingly, the CDP technique is not an appropriate constraint handling technique to employ in this algorithm. Only the MPF technique is therefore considered within the P-ACO algorithm.

7.5 Multiobjective local search algorithms

Although multiobjective local search metaheuristics are less popular in the MOA literature than their population-based counterparts, they are still widely-employed [46, 118, 209]. In this dissertation, multiobjective variants of the SA algorithm and a *variable neighbourhood search* algorithm from this class are employed as part of the comparative study for solving MICFMO problem instances.

7.5.1 The AMOSA algorithm

The AMOSA algorithm was developed by Bandyopadhyay *et al.* [10] and is one of the latest multiobjective simulated annealing methods available in the literature. In the algorithm, an archive of nondominated solutions is employed whose size is controlled by means of a clustering technique. The AMOSA algorithm also incorporates a quantity referred to as the *amount of domination* during the calculation of the acceptance probability of a new solution.

Newly-identified nondominated solutions are included in the archive on a continual basis during the execution of the algorithm. The archive has two size limits, namely a hard limit $\overline{N}_{\text{HL}}$ and a soft limit $\overline{N}_{\text{SL}}$, with $\overline{N}_{\text{SL}} > \overline{N}_{\text{HL}}$. Once the number of nondominated solutions in the archive reaches $\overline{N}_{\text{SL}}$, the well-known *single-link* (or *nearest-neighbour*) clustering technique [89] is applied in order to reduce the number of solutions down to $\overline{N}_{\text{HL}}$ again. During single-link clustering, each element in some set is initially considered as a cluster. Then, the distance between the clusters are calculated and the two clusters closest to each other are merged into a single new cluster. Accordingly, the number of clusters is reduced by one. Within the context of MOO, distances are measured in objective space and the Euclidean distance has been employed

here (as was the case in the SPEA2). This process is then repeated until the required number of clusters have been obtained. In the AMOSA algorithm, upon conclusion of the clustering process, a representative solution is identified within each cluster for retention in the archive, while the remaining solutions are discarded. This representative solution corresponds to the solution within a cluster whose average distance to the other solutions in that cluster is the smallest.

As mentioned earlier, the AMOSA algorithm incorporates a quantity called the *amount of domination* during the calculation of acceptance probabilities. Given two solutions $\boldsymbol{x}$ and $\boldsymbol{y}$, the amount of domination is defined as

$$\Delta D_{\boldsymbol{x},\boldsymbol{y}} = \prod_{\substack{k=1 \\ f_k(\boldsymbol{x}) \neq f_k(\boldsymbol{y})}}^{q} \frac{|f_k(\boldsymbol{x}) - f_k(\boldsymbol{y})|}{R_k}, \tag{7.10}$$

where q is the number of objectives, and R_k is the range of objective function k. This range may be approximated by using the solutions present in the archive, along with the current and new solutions.

The working of the AMOSA algorithm may now be described, and a pseudo-code listing thereof is presented in Algorithm 7.7. A random initialisation procedure is performed in order to fill the archive $\mathcal{A}$ with at least one nondominated solution. A current solution $\boldsymbol{x}_c$ is then randomly selected from the archive, and the temperature T is initialised. The current solution is perturbed according to some predefined neighbourhood move operator and a modified solution $\boldsymbol{x}_m$ is obtained. The "domination status" of $\boldsymbol{x}_m$ is determined with respect to $\boldsymbol{x}_c$ and all the solutions in $\mathcal{A}$. Depending on this status, different cases may arise:

Case 1 : If $\boldsymbol{x}_c \succ \boldsymbol{x}_m$, then the subset $\mathcal{Q}$, consisting of the solutions in $\mathcal{A}$ that dominate $\boldsymbol{x}_m$, has to be determined. An average amount of domination is then calculated as

$$\Delta D^1_{\text{avg}} = \frac{1}{|\mathcal{Q}| + 1} \left(\sum_{j=1}^{|\mathcal{Q}|} \Delta D_{\boldsymbol{x}_j,\boldsymbol{x}_m} + \Delta D_{\boldsymbol{x}_c,\boldsymbol{x}_m} \right), \tag{7.11}$$

which is used to determine the probability of $\boldsymbol{x}_m$ being accepted as the new current solution. This probability is calculated as

$$p_{\text{C1}} = \frac{1}{1 + \exp(\Delta D^1_{\text{avg}} T)}. \tag{7.12}$$

Case 2 : If $\boldsymbol{x}_c$ and $\boldsymbol{x}_m$ are nondominated with respect to each other, then $\boldsymbol{x}_m$ is accepted as the new current solution with certainty when it is also nondominated with respect to every solution in $\mathcal{A}$, or when it dominates at least one solution in $\mathcal{A}$. Otherwise, the subset $\mathcal{Q}$ is identified again and a different average amount of domination is calculated as

$$\Delta D^2_{\text{avg}} = \frac{1}{|\mathcal{Q}|} \left(\sum_{j=1}^{|\mathcal{Q}|} \Delta D_{\boldsymbol{x}_j,\boldsymbol{x}_m} \right). \tag{7.13}$$

The probability of $\boldsymbol{x}_m$ now being accepted as the new current solution is

$$p_{\text{C2}} = \frac{1}{1 + \exp(\Delta D^2_{\text{avg}} T)}. \tag{7.14}$$

Case 3 : If $\boldsymbol{x}_m \succ \boldsymbol{x}_c$, then $\boldsymbol{x}_m$ is accepted as the new current solution with certainty when it is also nondominated with respect to every solution in $\mathcal{A}$, or when it dominates at least one solution in $\mathcal{A}$. Otherwise, the subset $\mathcal{Q}$ is identified again and a so-called minimum amount of domination is calculated as

$$\Delta D_{\min} = \min_{j=1,\ldots,|\mathcal{Q}|} \left\{ \Delta D_{\boldsymbol{x}_m, \boldsymbol{x}_j} \right\}. \tag{7.15}$$

Now, the solution in the archive corresponding to this minimum amount of domination is accepted as the new current solution with probability

$$p_{\text{C3}} = \frac{1}{1 + \exp(-\Delta D_{\min})}, \tag{7.16}$$

whereas the probability of accepting $\boldsymbol{x}_m$ as the new current solution is $1 - p_{\text{C3}}$.

The well-known *geometric cooling schedule* is adopted in the AMOSA algorithm for managing the decrease in temperature T. According to this schedule, a temperature reduction is calculated as $T_{\text{new}} = \varphi T_{\text{old}}$, where φ is the cooling rate.

In order to apply the AMOSA algorithm to the MICFMO problem, a perturbation method (*i.e.* a neighourhood move operator) has to be specified. The swap and scramble mutation operators may be borrowed from the evolutionary computation literature and adopted as neighbourhood move operators. These two operators are compared in the context of constrained MICFMO in order to determine which is the most suitable choice. Finally, in this dissertation, the stopping criterion of the AMOSA algorithm has been modified — instead of using a fixed minimum temperature, a fixed number of iterations is adopted so as to control explicitly the duration of the algorithm's execution.

7.5.2 The MOVNS algorithm

The MOVNS algorithm was developed by Geiger [65] and it is a fairly simple multiobjective metaheuristic. Recall that the *neighbourhood* of a solution consists of several solutions that are "close" to it and may be generated by applying a neighbourhood move operator to the solution in question. In the MOVNS algorithm, a predefined set of neighbourhood move operators $\mathcal{M}$ is employed, along with an archive of nondominated solutions. During each iteration of the algorithm, the neighbourhood of a solution in the archive is explored whose neighbourhood has not yet been generated by one of the operators. The selection of the solution and its corresponding neighbourhood is determined randomly until all the solutions in the archive have been explored in full. A pseudo-code listing of the MOVNS algorithm is presented in Algorithm 7.8.

The algorithm starts by generating a random initial solution and assigning it to the archive $\mathcal{A}_0$. Each neighbourhood of this solution, $\mathcal{N}_i(\boldsymbol{x})$, that may be generated by the application of a neighbourhood move operator $i \in \mathcal{M}$, is marked as "unexplored." The iteration counter t is then set to zero, and the following procedure is iterated until the relevant stopping criterion has been met:

1. Randomly select a solution $\boldsymbol{x} \in \mathcal{A}_t$ whose neighbourhood $\mathcal{N}_i(\boldsymbol{x})$, for at least one $i \in \mathcal{M}$, has not yet been explored.
2. Identify the neighbourhood move operators in $\mathcal{M}$ corresponding to all unexplored neighbourhoods of $\boldsymbol{x}$, and randomly select one operator, say j.
3. Generate the unexplored neighbourhood $\mathcal{N}_j(\boldsymbol{x})$.

Algorithm 7.7: Archived multiobjective simulated annealing (AMOSA) [10].

Input : An MOP (possibly constrained), a maximum and a minimum temperature $T_{\max}$ and $T_{\min}$, limits for the archive size $\overline{N}_{\text{HL}}$ and $\overline{N}_{\text{SL}}$, a cooling rate φ, and the number of iterations to perform during each temperature step $i_{\max}$.

Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Populate the archive $\mathcal{A}$ according to a random initialisation procedure
Set the temperature $T \leftarrow T_{\max}$
Randomly select the current solution $\boldsymbol{x}_c$ from the archive $\mathcal{A}$
while $T > T_{min}$ **do**
  for $i \leftarrow 1$ **to** i_{max} **do**
    Perturb $\boldsymbol{x}_c$ to obtain $\boldsymbol{x}_m$
    if $\boldsymbol{x}_c \succ \boldsymbol{x}_m$ **then**
      Determine the subset $\mathcal{Q}$ consisting of the solutions in $\mathcal{A}$ that dominate $\boldsymbol{x}_m$
      Calculate ΔD^1_{avg} using (7.11)
      Accept $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$ with probability (7.12)
    else if $\boldsymbol{x}_c, \boldsymbol{x}_m$ *nondominated* **then**
      if *any* $\boldsymbol{x}_j \in \mathcal{A} \succ \boldsymbol{x}_m$ **then**
        Determine the subset $\mathcal{Q}$ consisting of the solutions in $\mathcal{A}$ that dominate $\boldsymbol{x}_m$
        Calculate ΔD^2_{avg} using (7.13)
        Accept $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$ with probability (7.14)
      else if $\boldsymbol{x}_m, \mathcal{A}$ *nondominated* **then**
        $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$, and $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}_m\}$
        if $|\mathcal{A}| > \overline{N}_{SL}$ **then**
          Perform single-link clustering on $\mathcal{A}$ to reduce its size to $\overline{N}_{\text{HL}}$
        end if
      else if $\boldsymbol{x}_m \succ$ *any* $\boldsymbol{x}_j \in \mathcal{A}$ **then**
        $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$, and $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}_m\}$, and remove any dominated solutions from $\mathcal{A}$
      end if
    else if $\boldsymbol{x}_m \succ \boldsymbol{x}_c$ **then**
      if *any* $\boldsymbol{x}_j \in \mathcal{A} \succ \boldsymbol{x}_m$ **then**
        Determine the subset $\mathcal{Q}$ consisting of the solutions in $\mathcal{A}$ that dominate $\boldsymbol{x}_m$
        Calculate $\Delta D_{\min}$ using (7.15) and identify the corresponding solution $\boldsymbol{x}_k$
        Accept $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_k$ with probability (7.16); otherwise $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$
      else if $\boldsymbol{x}_m, \mathcal{A}$ *nondominated* **then**
        $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$, and $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}_m\}$
        if $\boldsymbol{x}_c \in \mathcal{A}$ **then**
          $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\boldsymbol{x}_c\}$
        else if $|\mathcal{A}| > \overline{N}_{SL}$ **then**
          Perform single-link clustering on $\mathcal{A}$ to reduce its size to $\overline{N}_{\text{HL}}$
        end if
      else if $\boldsymbol{x}_m \succ$ *any* $\boldsymbol{x}_j \in \mathcal{A}$ **then**
        $\boldsymbol{x}_c \leftarrow \boldsymbol{x}_m$, and $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}_m\}$, and remove any dominated solutions from $\mathcal{A}$
      end if
    end if
  end for
  $T = \varphi T$
end while
if $|\mathcal{A}| > \overline{N}_{SL}$ **then**
  Perform single-link clustering on $\mathcal{A}$ to reduce its size to $\overline{N}_{\text{HL}}$
end if
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}$

4. Determine the next archive $\mathcal{A}_{t+1}$ by identifying the nondominated solutions in $\mathcal{A}_t \cup \mathcal{N}_j(\boldsymbol{x})$.
5. If $\boldsymbol{x}$ is present in the next archive, mark its neighbourhood $\mathcal{N}_j(\boldsymbol{x})$ as "explored."
6. Increment the value of the iteration counter $t \leftarrow t + 1$.

Liang and Chuang [117] proposed three variants of the MOVNS algorithm, which they refer to as *basic*, *perturbation* and *perturbation + base solution*. These variants involve different approaches toward selecting a solution from the archive for exploration, as well as the marking of neighbourhoods.

Basic variant: The only difference between the original algorithm and this variant is that the variant does not terminate when all neighbourhoods have been explored. Instead, all the neighourhoods are reset to "unexplored" so that every solution and neighbourhood are eligible for selection again.

Perturbation variant: In this variant, when all neighbourhoods have been explored, the solution selected during the previous iteration is perturbed so as to create a new solution whose neighbourhoods may be generated.

Perturbation + base solution variant: During the execution of this variant, a new solution is randomly selected from a subset of solutions in the archive. The subset consists of those solutions corresponding to extremal points in the nondominated front, as well as a single other solution selected from the archive at random. In addition, when all neighbourhoods have been explored in this variant, the solution selected during the previous iteration is perturbed, as before, so as to create a new solution.

These three variants of the MOVNS algorithm are compared within the context of constrained MICFMO in order to determine which is the most suitable choice. Finally, in order to apply the MOVNS algorithm (and its variants) to the MICFMO problem, a set of neighbourhood move operators have to be defined. In this dissertation, the swap and scramble mutation operators are

Algorithm 7.8: Multiobjective variable neighbourhood search (MOVNS) [65].

Input : An MOP (possibly constrained), and a set of neighbourhood move operators $\mathcal{M}$.
Output : An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Generate a random initial solution $\boldsymbol{x}$
$\mathcal{A}_0 \leftarrow \{\boldsymbol{x}\}$
Mark all neighbourhoods $\mathcal{N}_i(\boldsymbol{x})$ for $i \in \mathcal{M}$ as "unexplored"
$t \leftarrow 0$
while *any* $\mathcal{N}_i(\boldsymbol{x})$ *unexplored for* $i \in \mathcal{M}$ **do**
    Randomly select $\boldsymbol{x} \in \mathcal{A}_t$ whose neighbourhood $\mathcal{N}_i(\boldsymbol{x})$, for at least one $i \in \mathcal{M}$, has not yet been explored
    Identify the neighbourhood move operators in $\mathcal{M}$ corresponding to all unexplored neighbourhoods of $\boldsymbol{x}$, and randomly select one operator, say j
    Generate the neighbourhood $\mathcal{N}_j(\boldsymbol{x})$
    $\mathcal{A}_{t+1} \leftarrow$ nondominated solutions in $\mathcal{A}_t \cup \mathcal{N}_j(\boldsymbol{x})$
    if $\boldsymbol{x} \in \mathcal{A}_{t+1}$ **then**
        Mark neighbourhood $\mathcal{N}_j(\boldsymbol{x})$ as "explored"
    end if
    $t \leftarrow t + 1$
end while
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}$

borrowed from the evolutionary computation literature so as to constitute the set of neighbourhood move operators for the MOVNS algorithm. Furthermore, only a fixed number of solutions from the entire neighbourhood of a solution are generated (randomly) for exploration during each iteration in order to control explicitly the duration of the algorithm's execution.

7.6 A multiobjective probabilistic model-based algorithm: MOOCEM

The *cross-entropy method* (CEM) for optimisation, originally developed by Rubinstein [177], is based on an adaptive algorithm for estimating rare event probabilities and involves variance reduction by means of importance sampling [176]. A multiobjective extension of the CEM, called the MOOCEM, was recently proposed by Bekker and Aldrich [15]. The MOOCEM was specifically designed for reducing the number of function evaluations required in computationally expensive simulation-based MOPs and is therefore a natural choice for application to the MICFMO problem.

In the MOOCEM, solutions are generated by sampling a parameterised probability distribution (*i.e.* a probability model). An archive containing the best solutions (according to Pareto rank) is also employed in the algorithm. Based on the quality of solutions obtained, the current probability distribution is then re-estimated according to a problem-specific updating rule. During re-estimation, the *Kullback-Leibler divergence*[3] is employed as a measure of "distance" between two distributions, to be minimised. A sequence of probability distributions is therefore created during execution of the method until it converges to a distribution whose probability mass is concentrated in the vicinity of Pareto optimal solutions. In order to apply the MOOCEM to an MOP, an appropriate family of probability distributions has to be specified, as well as a procedure for generating the samples. Furthermore, an appropriate updating rule has to be derived in order to estimate the next probability distribution in the sequence, based on minimisation of the cross-entropy between two distributions.

The MOOCEM, as presented in [15], was primarily designed for solving continuous and integer optimisation problems. Combinatorial optimisation problems are, however, not well suited to the aforementioned MOOCEM formulation as it stands. This is reflected in the work by Bekker [14] in which markedly different algorithmic formulations are provided for the *vehicle routing problem with soft time windows* (VRPSTW) and the *buffer allocation problem* (BAP). Since the MICFMO is also a combinatorial optimisation problem, a generic version of the MOOCEM for combinatorial optimisation is presented here which is based on the algorithmic formulations for the VRPSTW and BAP presented in [14].

At the start of the MOOCEM, an empty archive $\mathcal{A}$ (also known as the *elite set*) is initialised. A probability distribution $p(\cdot;\boldsymbol{v})$, parameterised by a vector $\boldsymbol{v}$, is also specified, along with an initial parameter vector $\boldsymbol{v}_0$. The iteration counter t is the set to 1, and the following procedure is repeated until the relevant stopping criterion has been met:

1. Populate the so-called *working matrix*, $\mathcal{W}_t$, with N solutions sampled according to the probability distribution $p(\cdot;\boldsymbol{v}_{t-1})$.

2. Create a combined set of solutions $\mathcal{R}_t \leftarrow \mathcal{W}_t \cup \mathcal{A}$ from the working matrix and elite set. Then, calculate the Pareto rank ρ of each solution in $\mathcal{R}_t$.

[3]The Kullback-Leibler divergence between two probability distributions a and b is defined as the expectation of the logarithmic difference between the distributions, taken using the probabilities from a. It may be written mathematically as $D_{\mathrm{KL}}(a,b) = \mathbb{E}_a\left(\ln \frac{a(\boldsymbol{X})}{b(\boldsymbol{X})}\right) = \int a(\boldsymbol{x}) \ln a(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \int a(\boldsymbol{x}) \ln b(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ [15].

3. Clear the elite set and copy all solutions in $\mathcal{R}_t$ whose Pareto rank does not exceed a pre-specified threshold value ρ_E, into the new elite set $\mathcal{A}$.

4. Using the solutions in $\mathcal{R}_t$, solve for $\boldsymbol{v}$ in the so-called *stochastic program*

$$\underset{\boldsymbol{v}}{\text{maximise}} \quad \frac{1}{|\mathcal{R}_t|} \sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} \ln p(\boldsymbol{x}_k; \boldsymbol{v}) \tag{7.17}$$

   and denote its solution by $\hat{\boldsymbol{v}}_t$. The indicator function $I_{\{\rho_k \leq \rho_E\}}$ in (7.17) is defined as

$$I_{\{\rho_k \leq \rho_E\}} = \begin{cases} 1 & \text{if } \rho_k \leq \rho_E, \\ 0 & \text{otherwise.} \end{cases} \tag{7.18}$$

   In many cases, an analytic solution to (7.17) may be obtained in closed form, allowing $\hat{\boldsymbol{v}}_t$ to be calculated using a fixed mathematical expression (updating rule).

5. Calculate the new parameter vector $\boldsymbol{v}_t$ by smoothing $\hat{\boldsymbol{v}}_t$ according to

$$\boldsymbol{v}_t = \omega \hat{\boldsymbol{v}}_t + (1 - \omega) \boldsymbol{v}_{t-1}, \tag{7.19}$$

   where ω is a smoothing parameter.

6. Increment the value of the iteration counter $t \leftarrow t + 1$.

A pseudo-code listing of the MOOCEM is presented in Algorithm 7.9. Note that an extension to the basic procedure described above is presented in the pseudo-code. The extension corresponds to the VRPSTW algorithmic formulation in [14] and it contains an inner and an outer iteration loop, in which different values for the Pareto rank threshold are adopted in each loop.

In order to apply the MOOCEM to the MICFMO problem, a parameterised probability distribution has to be specified and the stochastic program (7.17) has to be solved in order to obtain/derive an updating rule for the parameter vector. Furthermore, a procedure for sampling a solution from the probability distribution has to be specified.

So, for the MICFMO problem, the chosen probability distribution is fully parameterised by an $n \times n$ probability matrix $\boldsymbol{P} = [p_{ij}]$, where n is the number of fuel assemblies and the number of loading positions in a reactor core. The entry p_{ij} corresponds to the probability of assigning fuel assembly j to loading position i in solution $\boldsymbol{x} = [x_1, \ldots, x_n]$. The corresponding updating rule is given by

$$p_{ij} = \frac{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} I_{\{\boldsymbol{x} \in \mathcal{X}_{ij}\}}}{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}}}, \tag{7.20}$$

where $\mathcal{X}_{ij}$ is the set of all reload configurations $\boldsymbol{x} \in \mathcal{X}$ for which $x_i = j$, and where $I_{\{\boldsymbol{x} \in \mathcal{X}_{ij}\}}$ is the indicator function defined as

$$I_{\{\boldsymbol{x} \in \mathcal{X}_{ij}\}} = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \mathcal{X}_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The detailed derivation of updating rule (7.20) for the MICFMO problem may be found in Appendix C.

Finally, a pseudo-code listing of the procedure for sampling a permutation solution to the MICFMO problem from the probability matrix $\boldsymbol{P}$ is presented in Algorithm 7.10.

Algorithm 7.9: Multiobjective optimisation using cross-entropy method (MOOCEM) for combinatorial optimisation problems [15].

Input : An MOP (possibly constrained), the sample size N, a parameterised probability distribution $p(\cdot;\boldsymbol{v})$, a smoothing parameter ω, and a maximum number of outer and inner loop iterations $\ell_{\max}$ and $t_{\max}$.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Set $\mathcal{A} \leftarrow \emptyset$
$\ell \leftarrow 0$
while $\ell < \ell_{max}$ **do**
  Initialise $\boldsymbol{v}_0$
  $t \leftarrow 1$
  Set $\rho_E = 2$
  while $t \leq t_{max}$ **do**
    Populate $\mathcal{W}_t$ with N solutions that are sampled according to $p(\cdot;\boldsymbol{v}_{t-1})$
    $\mathcal{R}_t \leftarrow \mathcal{W}_t \cup \mathcal{A}$
    Calculate the Pareto rank ρ_k for each solution $\boldsymbol{x}_k \in \mathcal{R}_t$
    Set $\mathcal{A} \leftarrow \emptyset$ and copy to $\mathcal{A}$ the solutions in $\mathcal{R}_t$ for which $\rho_k \leq \rho_E$
    Determine $\hat{\boldsymbol{v}}_t$ by solving the stochastic program (7.17)
    Calculate $\boldsymbol{v}_t$ by smoothing $\hat{\boldsymbol{v}}_t$ according to (7.19)
    $t \leftarrow t + 1$
  end while
  Set $\rho_E = 1$
  Trim the elite set $\mathcal{A}$ according to the new ρ_E
  $\ell \leftarrow \ell + 1$
end while
Set $\rho_E = 0$
Trim the elite set $\mathcal{A}$ according to the new ρ_E
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{A}$

Algorithm 7.10: Generation of random permutation solutions in the MOOCEM.

Input : The number of fuel assemblies and loading positions n in an MICFMO problem instance, and a probability matrix $\boldsymbol{P}$.
Output: A random permutation solution $\boldsymbol{x} = [x_1, \ldots, x_n]$ sampled using $\boldsymbol{P}$.

Generate a uniformly random permutation $[\pi_1, \ldots, \pi_n]$ of the set of positions $\{1, \ldots, n\}$
Set $\boldsymbol{P}^{(1)} \leftarrow \boldsymbol{P}$
$a \leftarrow 1$
while $a \leq n$ **do**
  Sample x_{π_a} according to the distribution formed by the π_a-th row of $\boldsymbol{P}^{(a)}$, namely $[p_{\pi_a 1}, \ldots, p_{\pi_a n}]$ `/* fuel assembly` x_{π_a} `is assigned to position` π_a `*/`
  $\boldsymbol{P}^{(a+1)} \leftarrow \boldsymbol{P}^{(a)}$
  Set the x_{π_a}-th column of $\boldsymbol{P}^{(a+1)}$ to $\boldsymbol{0}$ and renormalise each row so that it sums up to 1
  $a \leftarrow a + 1$
end while

Since the MOOCEM employs an explicit Pareto rank threshold during its execution, the CDP constraint handling technique is not entirely suitable because it assigns different ranks to different infeasible solutions. If $\rho_E = 2$, for example, then using the CDP technique may often lead to the situation where only three solutions constitute the elite set. This may be detrimental to the updating of probability matrix $\boldsymbol{P}$. Accordingly, only the MPF technique is considered within the MOOCEM.

7.7 A multiobjective harmony search algorithm: MOHS

The MOHS algorithm, as a multiobjective extension of the HS algorithm, was proposed by Sivasubramani and Swarup [196] and is largely based on the NSGA-II. During each iteration of the MOHS algorithm, a set of solutions is generated according to the same improvisation procedure employed in the single-objective algorithm (see §5.3.3). The new set of solutions is then combined with the current harmony memory before the FNSA in Algorithm 7.1 is applied to partition the set of solutions into different nondominated fronts. Solutions are then assigned their corresponding Pareto rank. Furthermore, each solution is also assigned a crowding distance using Algorithm 7.2. Then, the same truncating procedure used in the NSGA-II is employed to retain the best solutions into the next harmony memory.

The working of the MOHS algorithm may now described, and a pseudo-code listing thereof is presented in Algorithm 7.11. The algorithm starts by randomly generating an initial harmony memory $\mathcal{P}_0$ of size N. The iteration counter t is set to zero, and the following procedure is iterated until the relevant stopping criterion has been met (*e.g.* a maximum number of iterations reached):

1. Improvise a set of new solutions $\mathcal{Q}_t$ of size N using harmony memory consideration, pitch adjustment and randomisation procedures.
2. Create a combined set of solutions $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$ from the harmony memory and the set of new solutions.
3. Rank and sort $\mathcal{R}_t$ into nondominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_n$ using the FNSA, and calculate the crowding distance for each solution.
4. Create the next harmony memory $\mathcal{P}_{t+1}$ by including all solutions from the first front $\mathcal{F}_1$, then all solutions from the second front $\mathcal{F}_2$, and so forth, until the inclusion of all solutions from the next front would result in a harmony memory whose size is greater than N. In order to limit the size of $\mathcal{P}_{t+1}$ to N, sort the solutions in this next front in decreasing order of crowding distance. Include solutions from this sorted front one-by-one in $\mathcal{P}_{t+1}$ until $|\mathcal{P}_{t+1}| = N$.
5. Increment the value of the iteration counter $t \leftarrow t + 1$.

In order to apply the MOHS algorithm to the MICFMO problem, improvisation procedures within the context of the permutation-based encoding scheme have to be specified. Since the single-objective HS algorithm in §5.3.3 has already been adapted for application to permutation-based ICFMO problems, the same improvisation procedures implemented in Algorithm 5.1 are adopted in the MOHS algorithm.

7.8 Performance assessment of MOAs

Recall that the aim of MOO is to find an (approximate) set of Pareto optimal solutions for a given MOP instance. The quality performance of an MOA is, however, usually assessed in objective space, *i.e.* in terms of the (approximate) Pareto front obtained by the algorithm. Each of the multiobjective metaheuristics described in this chapter yields an approximate Pareto set as output along with its corresponding approximate Pareto front. In the absence of a true Pareto front, the performance assessment for these MOAs necessarily has to involve a comparative

Algorithm 7.11: Multiobjective harmony search (MOHS) [196].

Input : An MOP (possibly constrained), a harmony memory size N, an HMCR p_{hm}, a PAR p_{par}, and a maximum number of iterations $t_{\max}$.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

```
Randomly generate an initial harmony memory P_0 of size N
t ← 0
while t < t_max do
    Generate a set of new solutions Q_t of size N using improvisation procedures
    R_t ← P_t ∪ Q_t
    Partition R_t into nondominated fronts F_1, F_2, ... using the FNSA in Algorithm 7.1
    P_{t+1} ← ∅
    i ← 1
    while |P_{t+1}| < N do
        if |P_{t+1}| + |F_i| ≤ N then
            P_{t+1} ← P_{t+1} ∪ F_i
            i ← i + 1
        else
            Calculate the crowding distance for each solution in F_i using Algorithm 7.2
            Sort F_i in decreasing order of crowding distance
            P_{t+1} ← P_{t+1} ∪ {the first N − |P_{t+1}| solutions in F_i}
        end if
    end while
    t ← t + 1
end while
P̃_S ← P_{t_max}
```

analysis between the approximate Pareto fronts returned by each algorithm, relative to one another. For the sake of brevity, an approximate Pareto front is hereafter simply referred to as an *approximation front*.

A number of performance indicators that assign a scalar value to an approximation front as a measure of its quality have been proposed in the MOO literature [22, 29, 105, 106, 251]. It is generally recommended that the indicators adopted should have the property of *monotonicity* (or *Pareto compliance*) [22, 29]. This property means that, whenever an approximation front $\mathcal{A}$ is preferred over another front $\mathcal{B}$ with respect to weak Pareto dominance, the indicator value for $\mathcal{A}$ should not be worse than the value for $\mathcal{B}$ [106]. Two recommended indicators that fulfil this desirable property of monotonicity is the unary hypervolume indicator and the unary $R2$ indicator which are described in this section.

7.8.1 The hypervolume indicator

The *hypervolume* indicator (denoted by I_{HV}) was proposed by Zitzler and Thiele [250], and it essentially measures the portion of objective space dominated by an approximation front, relative to a specified reference point. This reference point, however, has to be dominated by the entire approximation front. Larger values of hypervolume correspond to preferred approximation fronts. An example of the concept of hypervolume for a bi-objective approximation front is presented graphically in Figure 7.2. In the figure, the surface area shaded in gray corresponds to the value of the hypervolume indicator for the associated approximation front, relative to the reference point.

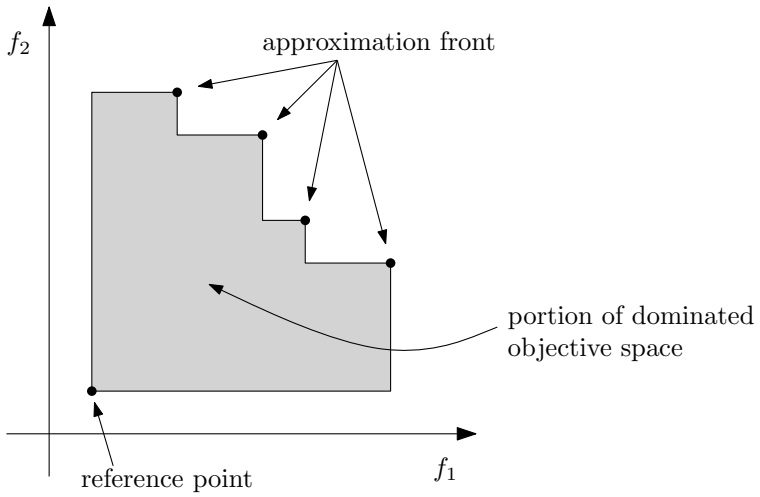


FIGURE 7.2: *Example of the concept of hypervolume for a bi-objective approximation front. The surface area shaded in gray corresponds to the value of the hypervolume indicator for this approximation front, relative to the reference point.*

In this dissertation, a variant of the hypervolume indicator is adopted as one of the performance measures. This variation is called the unary *hypervolume difference to reference* and is denoted by I_{HVD}. It is calculated simply as the difference between the hypervolume of some fixed reference approximation front $\mathcal{R}$ and the approximation front $\mathcal{A}$ under consideration. Therefore, $I_{\text{HVD}}(\mathcal{R}, \mathcal{A}) = I_{\text{HV}}(\mathcal{R}) - I_{\text{HV}}(\mathcal{A})$. Smaller values of the indicator now correspond to preferred approximation fronts.

Several different algorithms have been proposed in the literature for calculating hypervolume [229]. The *hypervolume by slicing objectives* algorithm, proposed by While *et al.* [230], is adopted here to calculate hypervolumes exactly.

7.8.2 The $R2$ indicator

The $R2$ indicator, denoted by I_{R2}, forms part of the R indicator family proposed by Hansen and Jaszkiewicz [72]. These indicators essentially compare approximation fronts on the basis of a set of utility functions. Suppose that a decision maker's preferences are specified in terms of a utility function $U_{\boldsymbol{w}}$, parameterised by the vector $\boldsymbol{w}$ over some set of parameters $\mathcal{W}$. The unary $R2$ indicator is then defined as

$$I_{R2}(\mathcal{R}, \mathcal{A}) = \frac{\sum_{\boldsymbol{w} \in \mathcal{W}} U^*(\boldsymbol{w}, \mathcal{R}) - U^*(\boldsymbol{w}, \mathcal{A})}{|\mathcal{W}|},$$

where $\mathcal{R}$ is some fixed reference approximation front and $\mathcal{A}$ is the approximation front under consideration. Furthermore, U^* is the maximum value obtained by the utility function $U_{\boldsymbol{w}}$ with parameter vector $\boldsymbol{w}$ over an approximation set, say $\mathcal{B}$. Therefore, $U^*(\boldsymbol{w}, \mathcal{B}) = \max_{\boldsymbol{z} \in \mathcal{B}} U_{\boldsymbol{w}}(\boldsymbol{z})$.

In [22, 72], it is recommended that the choice of utility function should be based on the weighted L_p-metrics and, in particular, the augmented weighted Chebyshev metric is suggested. Therefore, the $R2$ indicator, in which the chosen utility function is given by

$$U_{\boldsymbol{w}}(\boldsymbol{z}) = -\left(\max_{i=1,\ldots,q} (w_i |z_i - z_i^{\star\star}|) + \mu \sum_{j=1}^{q} |z_j - z_j^{\star\star}| \right),$$

is adopted as the other performance measure in this dissertation. The parameter vector $\boldsymbol{w}$ then consists of the weighting coefficients associated with each of the q objectives. A sufficiently large number of "*uniformly dispersed*" normalised weight vectors should be contained in the set $\mathcal{W}$ [22, 72]. The approach proposed in [72] is adopted here to generate the set of weight vectors in which each weighting coefficient takes on one of the values in $\{\frac{\ell}{k}, \ell = 0, \ldots, k\}$, where k is a parameter defining the number of weight levels. If, for example, there are three objectives and $k = 3$, then $\mathcal{W} = \{[0, 0, 1], [0, 1/3, 2/3], [0, 2/3, 1/3], [0, 1, 0], [1/3, 0, 2/3], [1/3, 1/3, 1/3], [1/3, 2/3, 0], [2/3, 0, 1/3], [2/3, 1/3, 0], [1, 0, 0]\}$, as shown in [72].

7.9 Chapter summary

Several modern state-of-the-art multiobjective metaheuristics were discussed in §7.3–§7.7. These metaheuristics are two evolutionary algorithms (namely NSGA-II and SPEA2), two swarm intelligence algorithms (namely OMOPSO and P-ACO), two local search algorithms (namely AMOSA and MOVNS), a probabilistic model-based algorithm (called MOOCEM), and an alternative algorithm (MOHS). These MOAs were deliberately sourced from different classes of metaheuristics in an attempt to encompass the diversity of algorithms available in the literature. Two constraint handling techniques for MOO were also described in §7.2, namely the CDP technique and the newly-proposed MPF technique.

Finally, in §7.8, the topic of performance assessment for MOAs was discussed. In particular, two recommended performance indicators from the literature (namely the hypervolume and $R2$ indicators) were described in some detail.

CHAPTER 8

MICFMO experimental results

Contents

In this chapter, the constraint handling techniques and multiobjective metaheuristics of Chapter 7 are applied to a test suite of constrained MICFMO problem instances, and the results thus obtained are compared in terms of solution quality (given a fixed algorithmic computation budget). In accordance with second priority in this dissertation (as mentioned in §1.3), these comparisons serve the purpose of identifying which computational methods are most suitable in the context of constrained MICFMO. It has been advocated in the literature that structured and statistically sound procedures should be employed when comparing the efficacies of metaheuristics [11, 22, 36, 172]. Accordingly, an extensive nonparametric statistical analysis is conducted in this chapter in the context of the results obtained.

8.1 A test suite for constrained MICFMO

Due to the absence of standard benchmark problem instances for MICFMO in the literature, a test suite of sixteen constrained MICFMO problem instances, based on the SAFARI-1 reactor, was created for the comparative study in this dissertation. Since the SAFARI-1 reactor is utilised for multiple purposes, as discussed in §4.5, several objectives may be pursued simultaneously in a realistic setting during MICFMO for the reactor. This makes for a diversity of objective space landscapes in the test suite generated. Little attention has been afforded in the literature to the study of different objective space landscapes, as evidenced by the discussions in §3.2.1. Similarly, a comprehensive and realistic constraint set may also be imposed for each instance in the test suite.

A large decision space for each MICFMO problem instance is also obtained by basing it on the SAFARI-1 reactor, because fuel assemblies are considered to be distinct from one another. As explained in §4.2.2, this is due to the fact that each fuel assembly accrues its own burnup history and associated isotopic composition (which also has an axial distribution), as it moves through an asymmetric reactor core along a unique path over various reactor operational cycles during its lifetime. Therefore, the decision space for an MICFMO problem instance based on the SAFARI-1 reactor cannot be reduced by considering fuel regions or batches (which is often done in the context of power reactors).

The typical objectives associated with the SAFARI-1 reactor, as listed in Table 4.1, were adopted in various combinations to form the test suite. For the sake of simplicity, a new objective (denoted by S9) is defined here as the sum of objectives S7 and S8. This objective is therefore the maximisation of the combined production of other isotopes in the two IPR facilities in SAFARI-1. Objectives S1–S6, together with compound objective S9, constitute the objectives considered during the construction of the test suite. In respect of constraints, the safety and utilisation requirements specified in §4.5.2 for the SAFARI-1 reactor constitute the constraint set imposed on all the problem instances in the test suite, as was the case in §5.4.1. The specific limiting values of the constraints have, however, been relaxed so that a reload configuration designed according to the current SAFARI-1 approach (described in §4.5.3) may be included in the feasible region of decision space. As mentioned before, these limiting values are proprietary knowledge and are therefore not divulged here (see §1.4).

By considering the aforementioned objectives in various combinations (along with the constraint set), a test suite of sixteen constrained MICFO problem instances, partitioned into three classes, was constructed for the comparative study in this dissertation. Class 1 contains six bi-objective problem instances, class 2 contains six tri-objective problem instances and class 3 contains four tetra-objective problem instances. These problem instances are considered for the specific SAFARI-1 operational cycle, C1211-1, for which the ANNs in Chapter 6 were constructed. Accordingly, the twenty-six fuel assemblies to be used in the problem instances correspond to those that were actually loaded into the SAFARI-1 core during that cycle. The objective function combinations of the problem instances are presented in Table 8.1.

8.2 Experimental design

The eight MOAs described in Chapter 7 are investigated here within a comparative study aimed at determining their ability to conduct constrained MICFMO on the aforementioned test suite of MICFMO problem instances. The MOAs were implemented within the Matlab software suite [214] so that objective and constraint function evaluations may be performed using the ANNs constructed in Chapter 6. The same personal computer, described in §6.6, was used to perform all the calculations in this comparative study.

8.2.1 General considerations

Recall from §5.4.1 that a typical shutdown and reload period for the SAFARI-1 reactor lasts five days, and that three days of computation time during those periods are available for optimisation. As mentioned, this corresponds to approximately 1 000 reload configurations that may be evaluated by the OSCAR-4 system. Although many more reload configurations may be evaluated in that time using the ANNs, a maximum limit (*i.e.* stopping criterion) of 1 050 evaluations is imposed for each metaheuristic in the comparative study. Accordingly, the practical limitation of MICFMO for the SAFARI-1 reactor is adhered to.

	Problem	Objectives						
		S1	S2	S3	S4	S5	S6	S9
Class 1	P1.1	✓	✓					
	P1.2	✓		✓				
	P1.3			✓	✓			
	P1.4		✓			✓		
	P1.5					✓	✓	
	P1.6				✓		✓	
Class 2	P2.1	✓	✓	✓				
	P2.2			✓	✓			✓
	P2.3	✓		✓	✓			
	P2.4				✓	✓	✓	
	P2.5		✓			✓	✓	
	P2.6	✓				✓	✓	
Class 3	P3.1	✓	✓	✓	✓			
	P3.2	✓		✓	✓			✓
	P3.3	✓	✓			✓	✓	
	P3.4			✓	✓	✓	✓	

TABLE 8.1: *The MICFMO test suite based on the SAFARI-1 reactor. Each test problem instance is denoted by "P#.#" in which the first number represents the class to which it belongs, and the second number is an enumeration over the problem instances in that class. The objective function labels correspond to those labels listed in Table 4.1.*

Each of the MOAs considered in the comparative study is a stochastic algorithm, meaning that different approximation fronts are returned by an algorithm if it is applied multiple times to the same problem instance. In order to obtain a representative indication of the average performance and variability of these MOAs, fifty optimisation runs of each metaheuristic (and its two constraint handling technique variants, where applicable) were executed in respect of each problem instance in the test suite described in §8.1. The outcome of each run is an approximation front corresponding to *feasible* solutions only. An attempt was also made to reduce variability due to initial conditions and random processes employed in the MOAs. Therefore, a fixed set of fifty different random number generator seeds was employed for each problem instance, along with a fixed set of random initial solutions sampled according to a uniform distribution. As a result, so-called *matched samples* (*i.e.* matched results) were obtained within each problem instance across the different MOAs.

In order to facilitate comparisons, the objective function values obtained for objective S2 — the only minimisation objective — were linearly transformed so that they correspond to values in a maximisation paradigm. Thereafter, all the objective function values in the approximation fronts were scaled to the range $(0, 1)$, using the maximum and minimum values attained for each objective, so that the results are of approximately the same magnitude.

Finally, a qualitative pilot study was performed in order to determine a suitable value for the severity factor in the MPF constraint handling technique. It was found that a value of $\gamma = 3$ is suitable for the problem instances in the test suite, irrespective of the metaheuristic in which the technique is employed.

8.2.2 Performance indicator considerations

According to the definitions of the I_{HVD} and I_{R2} indicators described in §7.8, a reference approximation front is required for a given problem instance in order to calculate the indicator values corresponding to an obtained approximation front. For each problem instance in the test suite, the approximation fronts yielded by all the optimisation runs, for all the MOAs, were pooled together. A reference approximation front associated with that problem instance was then determined by identifying the combined nondominated front from this pool.

The reference points required in the calculation of I_{HVD} were selected as $[0, 0]$, $[0, 0, 0]$ and $[0, 0, 0, 0]$ for the problem instances in classes 1, 2 and 3, respectively. Similarly, the utopian objective vectors required in the calculation of I_{R2} were selected as $[1.01, 1.01]$, $[1.01, 1.01, 1.01]$ and $[1.01, 1.01, 1.01, 1.01]$ for the problem instances in classes 1, 2 and 3, respectively.

As mentioned in §7.8.2, sets of "*uniformly dispersed*" weight vectors required in the calculation of I_{R2} were generated according to the approach proposed in [72]. The number of weight levels for the problem instances in class 1 was selected as $k = 499$, and a normalised weight vector set of size 500 was thus created. Similarly, for the problem instances in classes 2 and 3, normalised weight vector sets of sizes 741 and 969 were created, taking $k = 37$ and $k = 16$, respectively.

8.2.3 Individual metaheuristic considerations

Several qualitative pilot studies (*i.e.* parameter sensitivity analyses) were performed in order to determine reasonable values for the various tuning parameters present in each metaheuristic. A parameter was varied within its recommended range (typically specified in the source publication of the metaheuristic) at coarse intervals. Appropriate values were then determined such that reasonably good-quality solutions were obtained by the MOAs across the different problem instances within the test suite. In order to facilitate fair comparisons between the MOAs, an emphasis was also placed on adopting comparable population sizes. The outcomes of these pilot studies are summarised below and, although they do not necessarily represent optimal selections for each metaheuristic-problem instance pair, the parameter values were found to be relatively robust across the different problem instances.

For both MOEAs, namely the NSGA-II and the SPEA2, the population size was selected as $N = 30$, the crossover probability as $p_c = 0.9$, and the mutation probability as $p_m = 1/n$, where n denotes the length of the solution vector. Furthermore, the archive size in the SPEA2 was selected as $\overline{N} = 30$. In a separate comparative pilot study, it was found that the PMX crossover operator generally outperformed the POS and CX operators, while the scramble mutation operator yielded more promising results than the swap operator. Accordingly, the PMX and scramble operators were implemented within the NSGA-II and the SPEA2.

In the OMOPSO algorithm, the swarm size (which is also the maximum number of leaders) was selected as $N = 30$ and the mutation probability as $p_m = 1/n$. During a separate comparative pilot study, it was found that more promising results were obtained using the permutation-based approach proposed by Hu *et al.* [86] than by the method of random keys. As such, the former approach was implemented within the OMOPSO algorithm.

The number of ants employed in the P-ACO algorithm was selected as $N = 30$, while the bias parameters were selected as $\alpha = 1.75$ and $\beta = 0.75$, the selection probability as $r_0 = 0.8$, the evaporation rate as $\varrho = 0.2$, and the initial pheromone level as $\tau_0 = 1$.

For the AMOSA algorithm, hard and soft limits for the archive size were selected as $\overline{N}_{\text{HL}} = 30$ and $\overline{N}_{\text{SL}} = 45$, respectively. Furthermore, the cooling rate was selected as $\varphi = 0.85$, the initial

temperature as $T_{\max} = 100$, and the number of iterations performed during each temperature step as $i_{\max} = 15$. It was also found that the scramble neighbourhood move operator generally outperformed the swap operator during a separate comparative pilot study. Accordingly, the scramble operator was implemented within the AMOSA algorithm.

In the MOVNS algorithm, the number of neighbouring solutions generated and explored during each iteration was selected as 15. Also, the perturbation variant of the algorithm was implemented after it was found to be the most promising of the three variants proposed by Liang and Chuang [117] during a separate comparative pilot study.

For the MOOCEM, the sample size was selected as $N = 30$ and the smoothing parameter as $\omega = 0.8$. Furthermore, the ranking threshold was fixed at $\rho_E = 0$ throughout the method so that nondominated solutions only should be considered in the elite set. This, in turn, means that the outer loop in Algorithm 7.9 is no longer required (as was the case for the BAP in [14]).

The harmony memory size in the MOHS algorithm was selected as $N = 30$, the HMCR as $p_{\text{hm}} = 0.9$, and the PAR as $p_{\text{par}} = 0.25$.

Finally, wherever the scramble operator was implemented in a metaheuristic, the size of the subset of vector components permuted was selected as 4, and the set does not necessarily have to contain contiguous components.

8.3 Statistical analysis

Two types of analyses are typically performed in comparative studies between metaheuristics, namely single-problem and multi-problem analyses [36]. In a *single-problem analysis*, results obtained over several execution runs of metaheuristics on a particular optimisation problem instance are considered, whereas a result per metaheuristic/problem instance pair is considered in a *multi-problem analysis* [36]. In this dissertation, the outcomes of both these types of analyses are presented for the constrained MICFMO comparative study in order to demonstrate the inferences that may be drawn from each.

As suggested in [22, 36], a hypothesis testing approach from the field of inferential statistics is adopted in this dissertation in order to analyse the results. The *null hypothesis*, denoted by H_0, is typically a statement of no effect (or no difference in solution quality) and is assumed to be true. The *alternative hypothesis*, denoted by H_1, on the other hand, corresponds to the presence of an effect (or difference) [36]. A statistical test is then applied using samples of data, also referred to as *observations*, in order to determine whether the assumed H_0 should be rejected in favour of H_1, or not. Rejection of H_0 is determined by a parameter called the *significance level*, denoted here by $\tilde{\alpha}$. The statistical test may yield a so-called p-value representing the probability of obtaining an effect at least as extreme as that obtained in the data samples, assuming that H_0 is true [36]. Accordingly, if the p-value is smaller than $\tilde{\alpha}$, then H_0 is rejected in favour of H_1 at a significance level of $\tilde{\alpha}$.

Numerous statistical procedures are available in the literature and selecting an appropriate test depends on the experiment performed and the properties of the data. In the context of metaheuristic comparisons, it is generally recommended that *nonparametric tests* (also known as *distribution-free tests*) be employed because these procedures do not make any assumptions about the underlying distribution of the data [36, 82, 106]. In this chapter, the nonparametric *Wilcoxon signed rank test* [82] is employed for the constraint handling technique comparison, while the nonparametric *Friedman test* with the *Nemenyi post hoc procedure* is employed for the metaheuristic solution comparison [80, 82].

The Wilcoxon signed rank test may be used to compare two matched samples (*i.e.* pairs of data points) to assess whether there is a significant difference between the sample medians. In the procedure, two samples are converted into a single sample by taking the difference between each data point pair. The null hypothesis, then, is that this sample has a median of zero, while the two-tailed alternative hypothesis is that the median is not zero [82].

The Friedman test may be used to compare a set of two or more matched samples and is an example of an omnibus test. The null hypothesis H_0 is that all the medians of the samples are equal, while the alternative hypothesis H_1 is that the medians are not all equal. Rejection of H_0 in favour of H_1 therefore implies that at least two sample medians are significantly different [36]. The procedure is based on a transformation of the data points into so-called *Friedman ranks*. These ranks are determined by ordering the data points separately, within each matching, from least to greatest and then assigning corresponding rank values [82].

The Nemenyi *post hoc* procedure may follow a Friedman test if its null hypothesis was rejected. The rejection of the omnibus test only reveals that a significant difference exists between at least two of the samples — an appropriate *post hoc* multiple-comparisons procedure has to be performed to isolate the individual differences between pairs of samples. The Nemenyi procedure does so by performing two-tailed pairwise significance tests between all pairs of samples (using the Friedman ranks), correcting for the multiple inferences it makes [80, 82]. These corrections are very important, because they ensure that the experiment-wide significance level of $\tilde{\alpha}$ is adhered to. The Nemenyi procedure is regarded as a conservative *post hoc* procedure due to the manner in which it controls the *Type I error*[1] at level $\tilde{\alpha}$ under the overall null hypothesis [80].

The nonparametric tests mentioned above were performed utilising the Statistics and Machine Learning Toolbox [216] within the Matlab software suite [214], and a significance level of $\tilde{\alpha} = 0.05$ was adopted for all the cases presented in this chapter.

8.4 Numerical results

The comparative study results obtained by following the experimental design discussed in §8.2 are analysed in two stages. During the first stage, each of the six MOAs in which both constraint handling techniques have been implemented is considered separately. The aim of this stage is to ascertain whether the newly-proposed MPF technique performs better or worse (or the same) than the existing CDP technique. The best-performing variants of these six MOAs are then carried forward into the second stage of analysis for use together with the two remaining MOAs. During the second stage, all eight MOAs are then considered in order to determine their ability to conduct constrained MICFMO and gauge their comparative performances.

8.4.1 Constraint handling technique comparison

Consider the two constraint handling technique variants of the NSGA-II. A single-problem analysis is performed first, followed by a multi-problem analysis within each problem class.

[1] A *Type I error* is the incorrect rejection of the true null hypothesis, and it is colloquially referred to as a *false positive* [80].

Single-problem analysis

In the single-problem analysis, there are two samples of data for each problem instance in the test suite — matched pairs of indicator values corresponding to the fifty optimisation runs, obtained by the MPF and CDP variants. As mentioned in §8.3, the two samples are converted into a single sample for use within the Wilcoxon signed rank test by taking the difference between each data point pair. Let ΔI_{HVD} denote the converted sample of I_{HVD} values and, similarly, let ΔI_{R2} denote the converted sample of I_{R2} values. Negative values in these converted samples correspond to superior performance by the MPF technique, whereas positive values correspond to superior performance by the CDP technique.

Box plots, also known as box-whisker plots, are often employed during exploratory data analyses and provide a comprehensive view of the central tendency and spread of data samples. In general, the usage of box plots are more informative than the usage of averages and standard deviations of samples [110]. The converted ΔI_{HVD} and ΔI_{R2} samples obtained for each test problem using the NSGA-II are presented in the form of box plots in Figure 8.1. The average value of each sample is also included in the graphs as a black diamond point.

Note that the box plots in Figure 8.1 should not be compared explicitly to one another — the aim is rather to investigate whether each converted sample is symmetrically distributed about a median of zero, or not (so as to determine whether one constraint handling technique outperforms the other for a given problem instance). It may be observed in Figure 8.1 that the samples are, in fact, generally well-distributed about zero. As such, it indicates that the MPF and CDP techniques yield results of similar quality in respect of the I_{HVD} and I_{R2} performance indicators. There are, however, hints of the MPF technique outperforming the CDP technique (with respect to both indicators) for some of the problem instances in class 1 (*e.g.* P1.3 and P1.4) and in class 2 (*e.g.* P2.1 and P2.6).

The next step is to determine whether there are statistically significant differences between the MPF and CDP techniques. The two-tailed Wilcoxon signed rank test is therefore applied to the samples obtained for each test problem instance. The resulting p-values are presented in Table 8.2. Bold-faced entries in the table represent a statistically significant difference (for $\tilde{\alpha} = 0.05$). If a significant difference is detected, the box plots in Figure 8.1 may be referred to in order to pronounce which constraint handling technique outperformed the other.

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	0.0849	0.7102	0.0733	0.0733	0.2994	0.0604
ΔI_{R2}	0.0688	0.6675	**0.0254**	0.1080	0.2994	**0.0452**
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.0578	0.0659	0.4780	0.3667	0.8281	**0.0020**
ΔI_{R2}	**0.0126**	0.6887	0.6958	0.7684	0.5921	**0.0020**
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.3719	0.3566	0.7102	0.9730		
ΔI_{R2}	0.4258	0.9193	0.3876	0.9040		

TABLE 8.2: *Single-problem analysis results for comparing constraint handling techniques within the NSGA-II. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

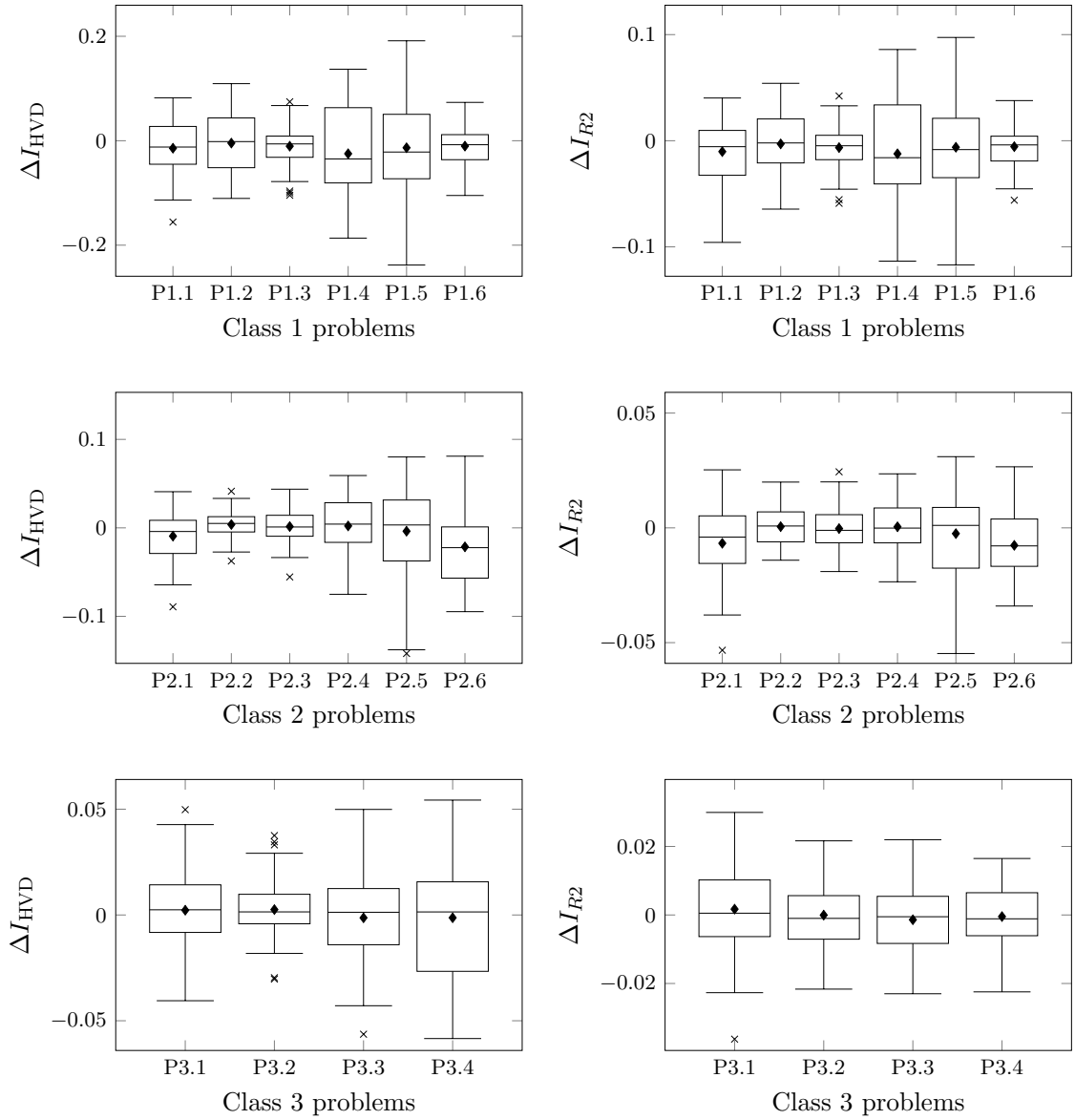


FIGURE 8.1: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the NSGA-II.*

For the majority of problem instances, as may be seen in Table 8.2, there is no statistically significant difference between the MPF and CDP constraint handling techniques (with respect to both I_{HVD} and I_{R2}). In the five instances where significant differences do, however, occur, it is found that the MPF technique outperforms the CDP technique. It may therefore be inferred that, based on a single-problem comparative analysis, the newly-proposed MPF constraint handling technique is a competitive alternative to the existing CDP technique, within the context of the NSGA-II applied to constrained MICFMO problem instances.

Multi-problem analysis

Next, a multi-problem analysis is performed, *within each class* of problem instances, in respect of the results obtained by the two variants of the NSGA-II. For this analysis, an average indicator value, calculated over the fifty optimisation runs, is determined for each problem instance

(and each constraint handling technique variant). Accordingly, average indicator values per variant/problem instance pair constitute the samples. As before, the two samples are converted into a single sample by taking their difference. Let $\overline{\Delta I}_{\text{HVD}}$ denote the converted sample of average I_{HVD} values and, similarly, let $\overline{\Delta I}_{R2}$ denote the converted sample of average I_{R2} values. Due to the small $\overline{\Delta I}_{\text{HVD}}$ and $\overline{\Delta I}_{R2}$ sample sizes, it is not appropriate to present the samples in the form of box plots. Instead, these converted samples are presented in tabular form in Table 8.3. Note that their values correspond to the diamond points in Figure 8.1.

Problem		Samples	
		$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$
Class 1	P1.1	−0.0144	−0.0103
	P1.2	−0.0045	−0.0030
	P1.3	−0.0104	−0.0065
	P1.4	−0.0249	−0.0123
	P1.5	−0.0134	−0.0061
	P1.6	−0.0103	−0.0056
Class 2	P2.1	−0.0094	−0.0067
	P2.2	0.0038	0.0005
	P2.3	0.0015	−0.0003
	P2.4	0.0022	0.0005
	P2.5	−0.0038	−0.0025
	P2.6	−0.0214	−0.0076
Class 3	P3.1	0.0022	0.0017
	P3.2	0.0026	0.0000
	P3.3	−0.0013	−0.0014
	P3.4	−0.0013	−0.0005

TABLE 8.3: *The converted samples of average indicator values, $\overline{\Delta I}_{HVD}$ and $\overline{\Delta I}_{R2}$, obtained for each problem instance class using the NSGA-II.*

It may be observed in Table 8.3 that all the average values in the samples for problem instance class 1 are negative, which suggests that the MPF technique outperforms the CDP technique with respect to both I_{HVD} and I_{R2}. The mix of positive and negative average values in the samples for classes 2 and 3, however, suggest that there is little difference between the two constraint handling techniques there.

The next step is to determine whether there is a statistically significant difference between the MPF and CDP techniques within a multi-problem setting. The two-tailed Wilcoxon signed rank test is therefore applied to the average samples obtained for each class of problem instances. The resulting p-values are presented in Table 8.4 and, as before, bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).

Although the sample size is small (only six observations), the multi-problem analysis shows that there is a statistically significant difference between the MPF and CDP constraint handling techniques within problem instance class 1. Referring back to the sample values in Table 8.3, it is found that the MPF technique outperforms the CDP technique in that class (with respect to both I_{HVD} and I_{R2}). In problem classes 2 and 3, however, no significant differences are detected. Therefore, as was the case in the single-problem analysis, it may also be inferred, based on a multi-problem analysis, that the newly-proposed MPF constraint handling technique is a competitive alternative to the existing CDP technique, within the context of the NSGA-II applied to constrained MICFMO problem instances.

Sample	Wilcoxon signed rank test p-values		
	Class 1	Class 2	Class 3
$\overline{\Delta I}_{\text{HVD}}$	**0.03125**	0.4375	0.625
$\overline{\Delta I}_{R2}$	**0.03125**	0.3125	0.875

TABLE 8.4: *Multi-problem analysis results for comparing constraint handling techniques within the NSGA-II. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the* $\overline{\Delta I}_{HVD}$ *and* $\overline{\Delta I}_{R2}$ *samples for each problem class. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

The remaining metaheuristics

The aforementioned single-problem and multi-problem analyses for comparing constraint handling techniques were performed not only for the NSGA-II, but also for the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms. For the purpose of improved readability, however, only a summary of the findings of those analyses are presented in this section. The full results may be found in Appendix D.

SPEA2: Very similar results to those for the NSGA-II are obtained in the single-problem analysis for the SPEA2. Statistically significant differences are detected in four instances. Referring to the associated box plots for the SPEA2, it is found that the MPF technique outperforms the CDP technique in three of those instances. In the multi-problem analysis, a significant difference between the constraint handling techniques is detected only for problem instance class 1. The sample values in the associated table indicate that the MPF technique outperforms the CDP technique in that class.

OMOPSO: A statistically significant difference is detected for one instance in the single-problem analysis for the OMOPSO algorithm, and it is in favour of the MPF technique. A multi-problem analysis, however, does not reveal any significant differences between the constraint handling techniques in any of the three problem classes.

AMOSA: Very interesting results are obtained during the single-problem analysis for the AMOSA algorithm. Highly significant differences between the constraint handling techniques are detected in all the instances within class 1, where most of the p-values are less than 0.0005. According to the associated box plots, the MPF technique outperforms the CDP technique in all of those instances. The AMOSA algorithm, therefore, strongly benefits from the MPF technique with respect to both I_{HVD} and I_{R2} when solving bi-objective problem instances within the context of constrained MICFMO. Statistically significant differences are also detected in five additional instances (*i.e.* in classes 2 and 3) of which four are in favour of the MPF technique. Unsurprisingly, a multi-problem analysis for the AMOSA algorithm reveals a significant difference between the two constraint handling techniques in problem instance class 1, with the sample values in the associated table indicating that it is the MPF technique which outperforms the CDP technique. No significant differences are detected in the remaining two problem classes.

MOVNS: In five instances, significant differences are detected during a single-problem analysis for the MOVNS algorithm. Based on the associated box plots, the MPF technique outperforms the CDP technique in four of those instances. The multi-problem analysis does not detect any statistically significant differences in any of the three problem instance classes.

MOHS: Finally, in a single-problem analysis for the MOHS algorithm, significant differences are detected in seven instances. Six of those are in favour of the MPF technique, according to the associated box plots. A multi-problem analysis, however, does not detect any statistically significant differences in any of the three problem instance classes.

Conclusion

The newly-proposed MPF constraint handling technique is therefore not only a competitive alternative to the existing CDP technique within the NSGA-II, but also within the SPEA2 as well as the OMOPSO, AMOSA, MOVNS and MOHS algorithms. This inference is supported by both single-problem and multi-problem analyses within the context of constrained MICFMO. The analyses further indicate that the MPF technique performs particularly well in bi-objective problem instances, especially within the AMOSA algorithm where the MPF technique significantly outperforms the CDP technique.

Based on the outcome of this first stage of the comparative study, the MPF technique is selected within all six metaheuristics for solving problem instances in class 1 and class 2 of the test suite. The CDP technique is, however, selected within all six metaheuristics for solving problem instances in class 3. Accordingly, these selected variants are carried forward into the second stage of the comparative analysis.

8.4.2 Multiobjective metaheuristic solution comparison

The selected variants of the NSGA-II, the SPEA2, and the OMOPSO, AMOSA, MOVNS and MOHS algorithms, carried over from the first stage of the comparative study, are now considered in conjunction with the P-ACO algorithm and the MOOCEM within the second stage of the comparative analysis. These eight metaheuristics are compared in this section, first according to a single-problem analysis and then based on a multi-problem analysis.

Single-problem analysis

In the single-problem analysis, there are eight samples of data for each problem instance in the test suite — matched indicator values corresponding to the fifty optimisation runs for each metaheuristic. These samples, obtained for the problem instances in classes 1, 2 and 3, are presented in the form of box plots in Figures 8.2, 8.3 and 8.4, respectively. The average value of each sample has been included, as before, as black diamond points in the graphs. This time, however, the aim is to compare the box plots with one another within each graph. A visual exploratory analysis may thus be performed in respect of each problem instance, with respect to both the I_{HVD} and I_{R2} indicators.

It is observed in Figures 8.2–8.4 that the OMOPSO algorithm performs poorly (with respect to both indicators) across all three classes of problem instances. The algorithm's performance is, however, particularly poor in the context of the problem instances within class 1, whereas the other seven metaheuristics perform fairly similarly in respect of those instances, as may be seen in Figure 8.2. Although the two local search metaheuristics, namely the AMOSA and the MOVNS algorithm, perform well for the bi-objective problem instances in class 1, they do not appear to scale well to the problem instances in classes 2 and 3, in which more than two objectives are present, as may be seen in Figures 8.3 and 8.4. This behaviour is apparent for both the I_{HVD} and I_{R2} indicators.

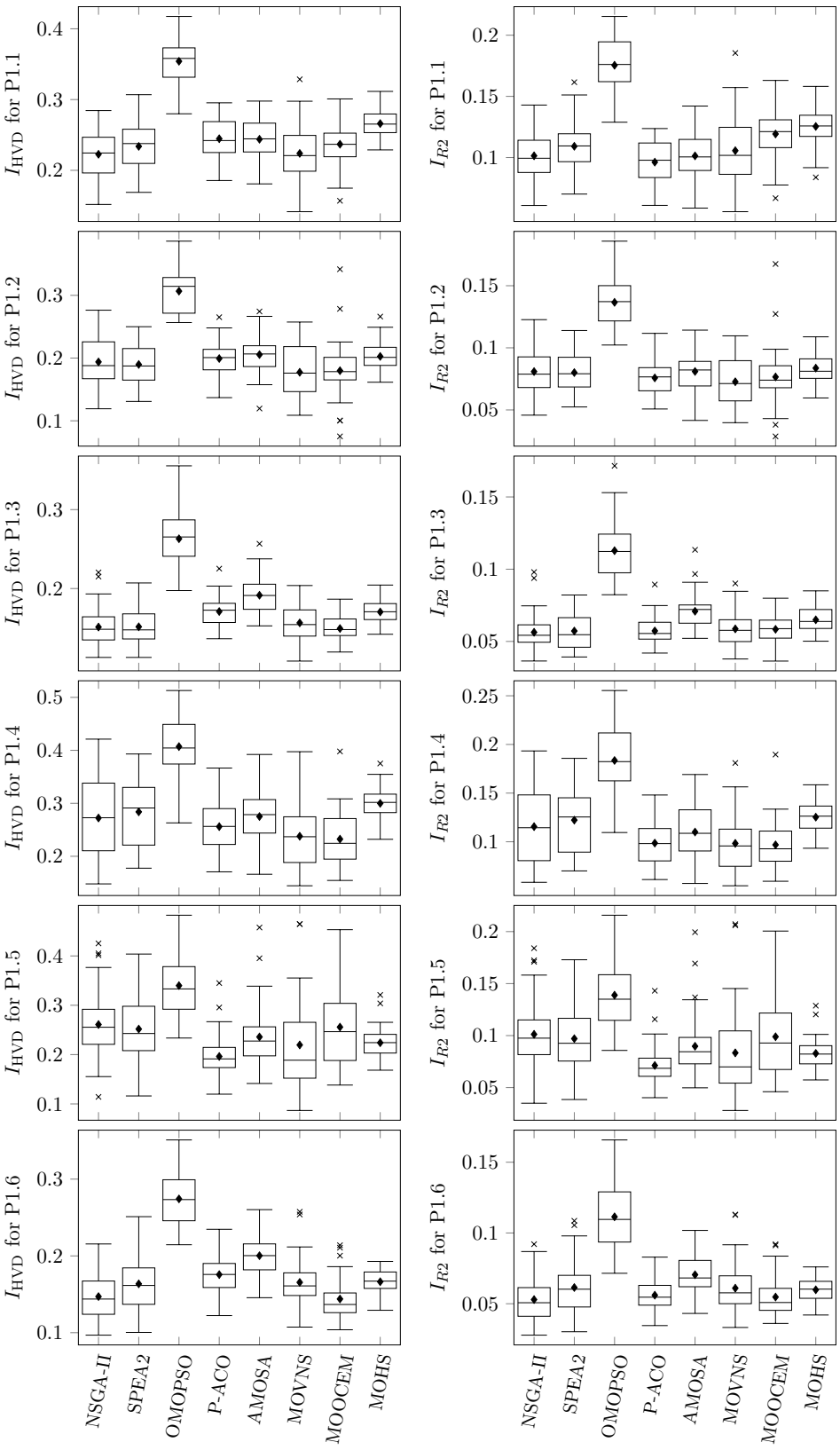


FIGURE 8.2: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all eight metaheuristics for each problem instance in class 1.*

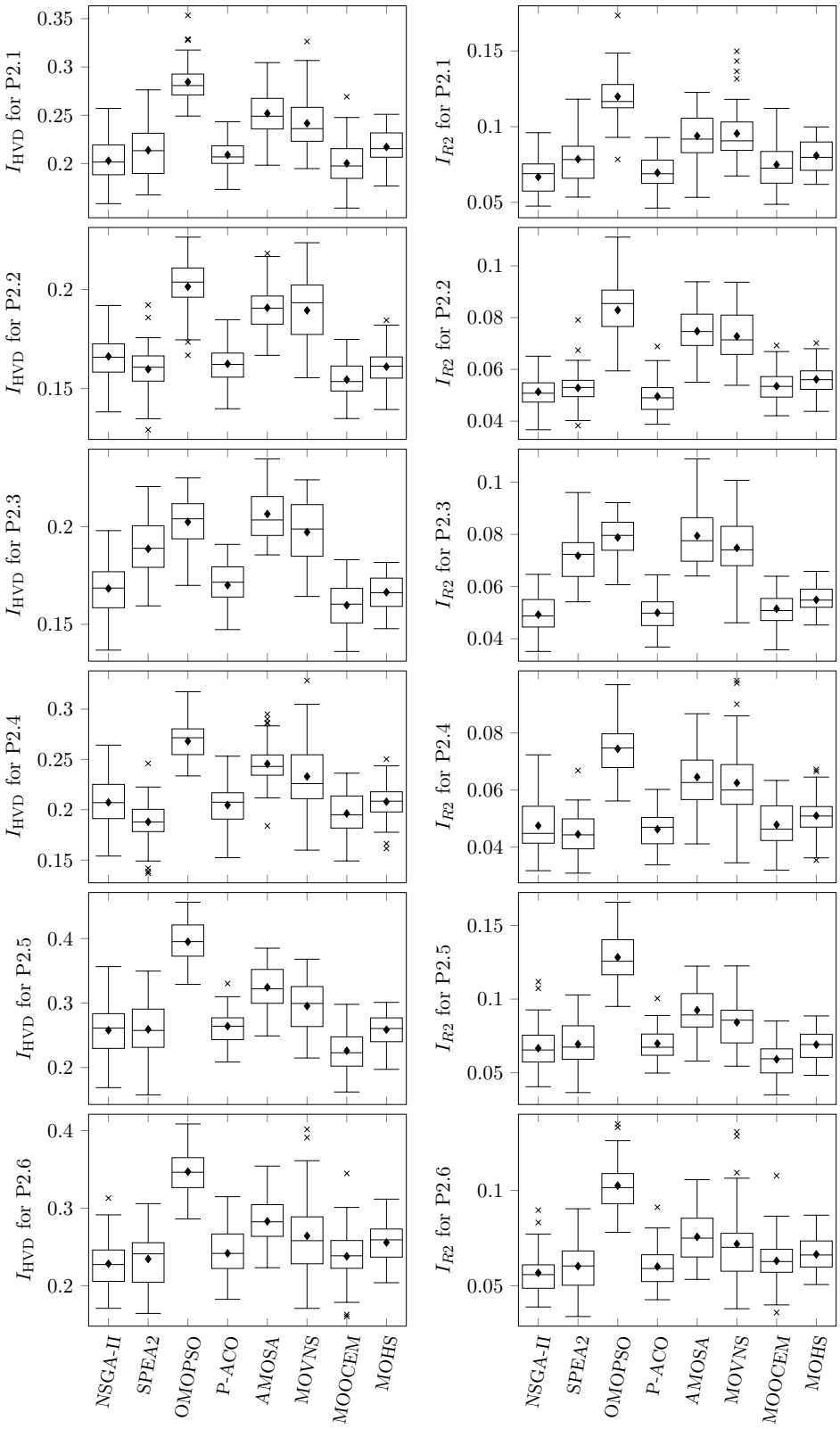


FIGURE 8.3: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all eight metaheuristics for each problem instance in class 2.*

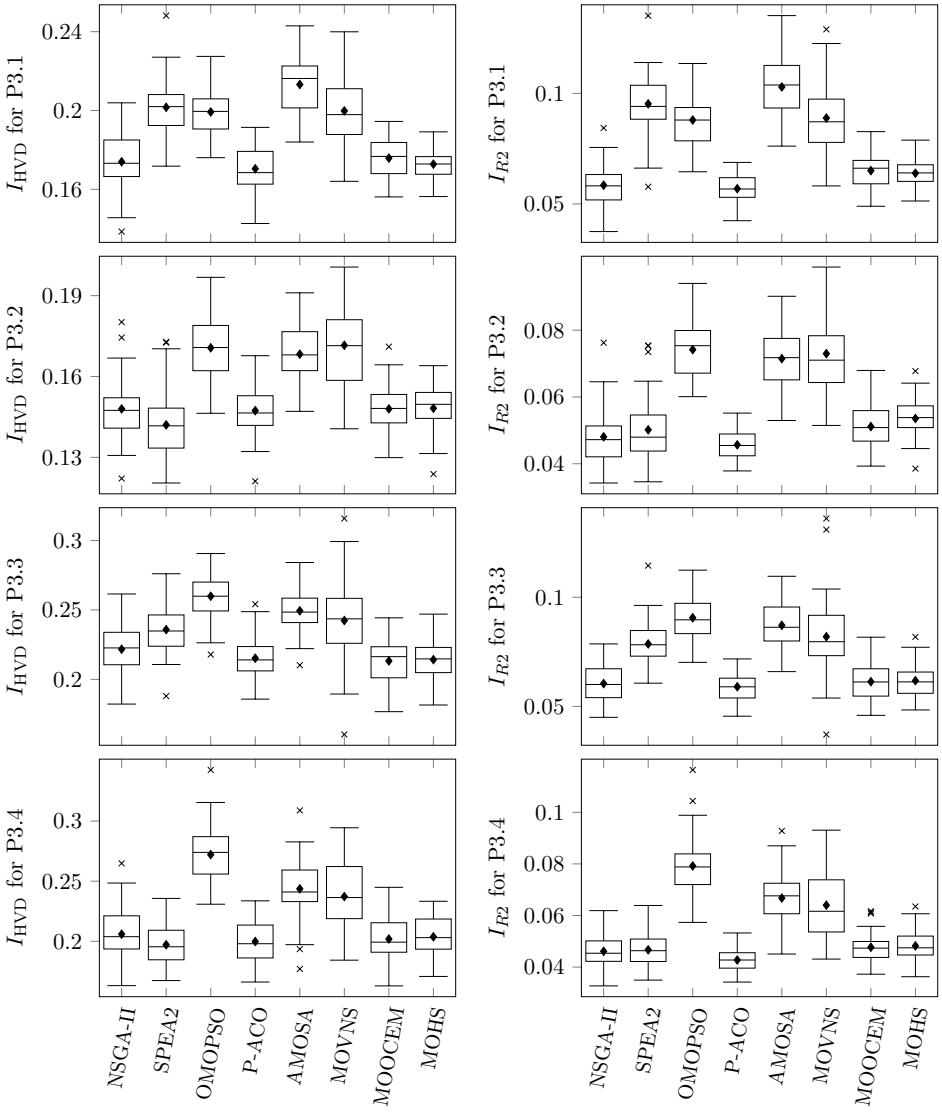


FIGURE 8.4: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all eight metaheuristics for each problem instance in class 3.*

The MOHS algorithm appears, in general, to be the most robust of the metaheuristics in terms of sample variability (*i.e.* it is fairly insensitive to different starting conditions) although its performance is only average in the majority of instances. The NSGA-II, the P-ACO algorithm and the MOOCEM, on the other hand, appear to perform consistently well across the problem instances in all three classes with respect to both indicators. The SPEA2 also performs well in most cases, although it seems to suffer sporadically from poor performance — in problem instances P2.3, P3.1 and P3.3, for example, as may be seen in Figures 8.3 and 8.4.

The next step is to determine whether there is a statistically significant difference between the metaheuristics in this single-problem analysis. The Friedman test is therefore applied to the samples obtained for each test problem instance. Recall that the test only detects whether a significant difference exists between at least two samples. Accordingly, if such a difference has been detected, the Nemenyi *post hoc* procedure is applied in order to identify the individual differences between pairs of samples. Since there are eight metaheuristics in this study, the Nemenyi procedure performs $\binom{8}{2} = 28$ pairwise significance tests.

Due to the multitude of tests that have to be carried out, the details of the single-problem analysis are not presented here. Instead, only a summary of the findings is presented in order to improve the readability of the main text. The reader is referred to Appendix D for the unabridged statistical results.

The Friedman test detects a statistically significant difference (for $\tilde{\alpha} = 0.05$) for every problem instance in the test suite with respect to both indicators. The p-values are, in fact, numerically zero in each test. This outcome does not come as a surprise, given the poor performance observed for the OMOPSO algorithm during the visual exploratory analysis. Following application of the Nemenyi *post hoc* procedure, it is found that the OMOPSO algorithm is significantly different from (*i.e.* worse than) every other metaheuristic across all the problem instances in class 1, with respect to both indicators. Furthermore, the OMOPSO, AMOSA and MOVNS algorithms are also significantly different from (*i.e.* worse than) every other metaheuristic in the majority of problem instances within class 2 and class 3, again with respect to both the I_{HVD} and I_{R2} indicators. A number of other statistically significant differences are also detected during the *post hoc* analyses, but these are scattered throughout the results which make any meaningful inferences difficult. As such, these detections are not mentioned individually in this section.

Multi-problem analysis

A multi-problem analysis, *within each class* of problem instances, is performed next. As before, an average indicator value, calculated over the fifty optimisation runs, is determined for each metaheuristic/problem instance pair. The average indicator values for I_{HVD} and I_{R2} correspond to the black diamond points in Figures 8.2–8.4.

Suppose the metaheuristics are ranked according to the average indicator values, for each problem instance. The average values may then be replaced by integers (*i.e.* Friedman ranks) from 1 through to 8, where 1 corresponds to the best average value and 8 to the worst. Thereafter, an average rank R_{avg} may be calculated for every metaheuristic over each class of problem instances. These average ranks may then be utilised to compare how well the different metaheuristics performs in a multi-problem context. This intuitive approach of comparison actually corresponds, in part, to the Friedman test — the average ranks described above are employed in the Friedman test and the Nemenyi *post hoc* procedure. These ranks are returned to again later in this section.

In order to determine whether there is a statistically significant difference between the metaheuristics in a multi-problem analysis, the Friedman test is applied to the average indicator value samples for each class of problem instances. The resulting p-values are presented in Table 8.5 and, as before, bold-faced entries correspond to the detection of a significant difference (for $\tilde{\alpha} = 0.05$).

Sample	Friedman test p-values		
	Class 1	Class 2	Class 3
I_{HVD}	$\mathbf{8.122 \times 10^{-4}}$	$\mathbf{9.451 \times 10^{-6}}$	$\mathbf{9.008 \times 10^{-3}}$
I_{R2}	$\mathbf{2.836 \times 10^{-3}}$	$\mathbf{7.249 \times 10^{-6}}$	$\mathbf{1.139 \times 10^{-3}}$

TABLE 8.5: *Multi-problem analysis results for comparing the metaheuristics. The table contains the p-values obtained by Friedman tests applied to the average indicator value samples for each problem instance class. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Although the sample sizes are small, the Friedman test is able to detect a statistically significant difference between the metaheuristics in all three problem instance classes, for both indicators. As was the case in the single-problem analysis, this outcome does not come as a surprise, given the poor performance of the OMOPSO algorithm observed during the visual exploratory analysis.

Since statistically significant differences are detected in the Friedman tests, the Nemenyi *post hoc* procedure is applied to the samples. The full set of statistical results (*i.e.* the p-values obtained from the pairwise significance tests performed during the Nemenyi procedure) may be found in Appendix D. In order to assist in the interpretation of those results, the outcomes of the *post hoc* analyses are presented in tabular form along with the average ranks R_{avg}, as described above. As such, the metaheuristics are ranked according to their average rank values, within each problem instance class (for both I_{HVD} and I_{R2}). These rank results for class 1, class 2 and class 3 are presented in Tables 8.6, 8.7 and 8.8, respectively. If the Nemenyi procedure detects a significant difference between two metaheuristics, the detection is indicated in the table with matching alphabetic letters next to the corresponding metaheuristics in the column labelled "S.Diff".

I_{HVD} *post hoc* results			I_{R2} *post hoc* results		
R_{avg}	Metaheuristic	S.Diff	R_{avg}	Metaheuristic	S.Diff
2.5	MOVNS	a	2.167	P-ACO	a
2.5	MOOCEM	b	3.333	MOVNS	b
3.333	NSGA-II	c	3.667	NSGA-II	c
3.833	SPEA2	—	3.667	MOOCEM	d
4.5	P-ACO	—	4.667	SPEA2	—
5.5	MOHS	—	5	AMOSA	—
5.833	AMOSA	—	5.5	MOHS	—
8	OMOPSO	abc	8	OMOPSO	abcd

TABLE 8.6: *Multi-problem analysis results for the* post hoc *procedure applied to problem instance class 1. Metaheuristics are ranked according to their average rank values. Significant differences between any two metaheuristics are denoted by matching alphabetic letters in the column labelled "S.Diff".*

In Table 8.6, it may be observed that the OMOPSO algorithm is significantly different from the MOVNS algorithm, the MOOCEM, and the NSGA-II with respect to both performance indicators for problem instance class 1, and also from the P-ACO algorithm with respect to I_{R2}. By referring to the average ranks, it is clear that the OMOPSO algorithm is the worst-performing metaheuristic in class 1 (with respect to both indicators) and this is consistent with the finding of the single-problem analysis. Furthermore, the MOVNS algorithm and the MOOCEM are jointly the best-performing metaheuristics with respect to I_{HVD}, while the P-ACO algorithm performs the best with respect to I_{R2}. Interestingly, it is noted that, for the P-ACO algorithm, the rankings are quite different between the two indicators. Accordingly, this demonstrates that the selection of performance indicator may influence the choice of multiobjective metaheuristic to employ. The MOVNS algorithm and the NSGA-II, however, seem to yield consistently good results across both indicators in problem instance class 1.

According to the rankings in Table 8.7, the worst-performing metaheuristics in problem instance class 2 are the MOVNS, AMOSA and OMOPSO algorithms, which is consistent with the single-problem analysis finding. The *post hoc* analysis detects that the OMOPSO algorithm is significantly different from the MOOCEM, the NSGA-II and the SPEA2 with respect to both performance indicators, and also from the P-ACO algorithm with respect to the I_{R2} indica-

I_{HVD} *post hoc* results			I_{R2} *post hoc* results		
R_{avg}	Metaheuristic	S.Diff	R_{avg}	Metaheuristic	S.Diff
1.5	MOOCEM	ade	1.667	NSGA-II	aef
2.833	NSGA-II	bf	2.333	P-ACO	bg
3	SPEA2	c	3.167	MOOCEM	c
3.833	P-ACO	—	3.333	SPEA2	d
3.833	MOHS	—	4.5	MOHS	—
6	MOVNS	d	6.167	MOVNS	e
7.167	AMOSA	ef	7	AMOSA	fg
7.833	OMOPSO	abc	7.833	OMOPSO	abcd

TABLE 8.7: *Multi-problem analysis results for the* post hoc *procedure applied to problem instance class 2. Metaheuristics are ranked according to their average rank values. Significant differences between any two metaheuristics are denoted by matching alphabetic letters in the column labelled "S.Diff".*

tor. It may also be observed that the P-ACO algorithm again performs well with respect to I_{R2}, but less so with respect to I_{HVD}. Finally, the MOOCEM and the NSGA-II are the two best-performing metaheuristics for I_{HVD} and I_{R2}, respectively, over problem instance class 2.

In Table 8.8, it may be observed that no statistically significant differences are detected between any pair of metaheuristics with respect to I_{HVD}, although the Friedman test yields a significant p-value. This outcome may be attributed to the conservatism present in the Nemenyi procedure and the fact that so many metaheuristics are compared in the *post hoc* analysis. With respect to I_{R2}, however, it is found that the P-ACO algorithm is significantly different from the MOVNS, AMOSA and OMOPSO algorithms. Furthermore, the NSGA-II is also significantly different from the OMOPSO algorithm. According to the rankings in Table 8.8, the worst-performing metaheuristics in problem instance class 3 are the MOVNS, AMOSA and OMOPSO algorithms, which is also consistent with the finding of the single-problem analysis. The P-ACO algorithm is observed to be the best-performing metaheuristic in problem instance class 3 with respect to both performance indicators. Unlike before, however, the rankings of the NSGA-II are quite different for the two indicators. It is also noted that the MOHS algorithm seems to perform better on class 3 problem instances (in which four objectives are present) than on class 1 and class 2 problem instances (having fewer than four objectives).

I_{HVD} *post hoc* results			I_{R2} *post hoc* results		
R_{avg}	Metaheuristic	S.Diff	R_{avg}	Metaheuristic	S.Diff
2	P-ACO	—	1	P-ACO	acd
3	MOOCEM	—	2	NSGA-II	b
3.25	MOHS	—	3.75	MOOCEM	—
3.5	SPEA2	—	4.25	MOHS	—
3.75	NSGA-II	—	4.5	SPEA2	—
6.5	MOVNS	—	6.25	MOVNS	c
7	OMOPSO	—	7	AMOSA	d
7	AMOSA	—	7.25	OMOPSO	ab

TABLE 8.8: *Multi-problem analysis results for the* post hoc *procedure applied to problem instance class 3. Metaheuristics are ranked according to their average rank values. Significant differences between any two metaheuristics are denoted by matching alphabetic letters in the column labelled "S.Diff".*

Conclusion

The single-problem and multi-problem statistical analyses conducted in this section suggest that the OMOPSO algorithm may be discarded from future research in the context of constrained MICFMO. Similarly, the two local search MOAs, namely the AMOSA and MOVNS algorithms, may also be discarded for problem instances in which three or more objectives are present. The MOVNS algorithm, however, performs very well when solving bi-objective problem instances. Due to the average performance of the SPEA2 and the MOHS algorithm in the majority of instances, these two MOAs may likely also be eliminated from consideration, although the MOHS algorithm seems to be the most robust metaheuristic in terms of sample variability. Finally, the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing MOAs with respect to both the I_{HVD} and I_{R2} indicators, across the problem instances in all three classes. As such, these MOAs may form an integral part of any future research in the context of constrained MICFMO, along with the bi-objective MOVNS algorithm.

8.5 Chapter summary

In this chapter, the results obtained from a comparative study in which two constraint handling techniques and eight multiobjective metaheuristics were compared relative to one another, were presented. A test suite of constrained MICFMO problem instances, based on the SAFARI-1 reactor, was constructed in §8.1 for this study due to the absence of standard benchmark problem instances for MICFMO in the literature. These problem instances were considered for the specific SAFARI-1 operational cycle, C1211-1, for which the ANNs in Chapter 6 were constructed.

In §8.2, the experimental design that was followed during the comparative study was presented. A description of the nonparametric statistical analysis conducted on the numerical results was presented thereafter in §8.3. The difference between single-problem and multi-problem analyses was also explained as both approaches were followed in this chapter.

Finally, the numerical results of the comparative study were presented in §8.4. The results from the constraint handling technique comparison in §8.4.1 revealed that the newly-proposed MPF technique is a competitive alternative to the existing CDP technique within the context of constrained MICFMO. The metaheuristic comparison in §8.4.2, on the other hand, indicated that the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing MOAs with respect to both the I_{HVD} and I_{R2} performance indicators across the problem instances in the test suite. In addition, it was found that the MOVNS algorithm performs particularly well in the context of the bi-objective problem instances.

CHAPTER 9

A multiobjective hyperheuristic for MICFMO

Contents

In this chapter, a multiobjective hyperheuristic, called the AMALGAM method, is investigated in terms of its ability to conduct constrained MICFMO by incorporating the findings of Chapter 8 into the method. The aim of the investigation is to raise the level of generality at which MICFMO may be performed, and potentially improve the quality of optimisation results. The working of the AMALGAM method is described first, before several variants thereof are applied to the test suite of constrained MICFMO problem instances constructed in §8.1. The results thus obtained, along with a subset of results from Chapter 8, are compared in terms of solution quality (given a fixed algorithmic computation budget). As before, an extensive nonparametric statistical analysis is conducted in the context of the results obtained.

9.1 Introduction

The results obtained during the metaheuristic comparative study described in Chapter 8 revealed that no single MOA was able to consistently outperform the other algorithms with respect to all, or most of, the problem instances in the MICFMO test suite of §8.1 (and with respect to both performance indicators). This outcome is in line with the so-called *No Free Lunch* (NFL) theorem for optimisation, which essentially states that an optimisation algorithm that performs particularly well on one set of objective functions, will also perform correspondingly poorly on all other objective functions [233]. The NFL theorem therefore implies that a metaheuristic which performs well on some set of benchmark problems is not guaranteed to perform well in the context of a new (*i.e.* different) optimisation problem.

Because a diversity of objective functions (and combinations thereof) were adopted in the constrained MICFMO test suite of §8.1, the inferences drawn during the comparative study between different MOAs in Chapter 8 are relatively general in the context of MICFMO applied to the SAFARI-1 reactor. It is therefore expected, although not guaranteed (because of the NFL theorem), that the NSGA-II, the P-ACO algorithm and the MOOCEM should be able to solve various MICFMO problem instances with good effect and, similarly, that the MOVNS algorithm should perform well in respect of bi-objective instances. To the best knowledge of the author, no other study has been published in which the general applicability of an MICFMO solution technique has been tested to such an extent as was done in Chapter 8.

In an attempt to further improve upon the level of generality at which MICFMO may be performed, the focus in this dissertation now turns towards a relatively new and promising field of research, namely *hyperheuristic* solution techniques [23]. Although there is also no universally accepted definition of what a hyperheuristic is, it may generally be thought of as a "*heuristic which chooses heuristics*" [26]. A key feature of a hyperheuristic is therefore that it operates on a space of heuristics by managing the selection of which low-level heuristic (from a given set) should be applied at any given time. Apart from raising the level of general applicability, hyperheuristics have also been shown to achieve improved performance in optimisation studies [23].

Although the overwhelming majority of hyperheuristics available in the literature have been designed in the context of single-objective optimisation, a few multiobjective hyperheuristics have also been proposed to date [23, 125]. In this dissertation, the AMALGAM method[1], recently proposed by Vrugt and Robinson [226], is investigated for application to constrained MICFMO.

9.2 A multiobjective hyperheuristic: AMALGAM

The AMALGAM method is an evolutionary-based multiobjective hyperheuristic which combines two concepts, namely *simultaneous multi-algorithm search* and *self-adaptive offspring creation*, for the solution of MOPs [226]. In the method, k sub-algorithms are employed simultaneously, each creating a number of offspring solutions proportional to the success of the sub-algorithm during the previous generation. Another important aspect of the method is that of *global information sharing* — each sub-algorithm has access to the entire population in order to create its share of the offspring.

The concept of self-adaptive offspring creation in the AMALGAM method is designed to favour sub-algorithms exhibiting the highest reproductive success [226]. Suppose S^i_{t+1} is the number of solutions created by sub-algorithm i as a contribution to parent population $\mathcal{P}_{t+1}$ during generation $t+1$. Furthermore, let N^i_{t+1} be the number of offspring solutions that sub-algorithm $i \in \{1, \ldots, k\}$ should create during generation $t+1$. The reproductive success of sub-algorithm i is then measured as the ratio of the number of successful offspring solutions to the total number of offspring solutions it created, that is S^i_{t+1}/N^i_t. In order to reward the "best" sub-algorithms based on their reproductive success, the number of offspring solutions that sub-algorithm i should generate during the next generation is calculated as approximately

$$N^i_{t+1} = \frac{N\left(\frac{S^i_{t+1}}{N^i_t}\right)}{\sum_{i=1}^{k}\left(\frac{S^i_{t+1}}{N^i_t}\right)}, \tag{9.1}$$

where N is the total number of solutions in the population. The reproductive success of a sub-algorithm is therefore scaled according to the combined success of all the sub-algorithms. It is

[1] AMALGAM is an acronym for *a multi-algorithm, genetically adaptive multiobjective*.

recommended that a minimum value for N_t^i be enforced so as to avoid deactivating any of the sub-algorithms [226].

Finally, the AMALGAM method borrows largely from the NSGA-II and utilises its fitness assignment procedure (*i.e.* Pareto rank and crowding distance assignment), the FNSA for partitioning a set of solutions into different nondominated fronts, and its elitest selection procedure for determining the next population.

The full AMALGAM method may now be described, and a pseudo-code listing therof is presented in Algorithm 9.1. The method starts by randomly generating an initial parent population $\mathcal{P}_0$ of size N. This population is then sorted and ranked according the FNSA in Algorithm 7.1. An offspring population $\mathcal{Q}_0$ of size N is generated next using the simultaneous multi-algorithm search concept. Accordingly, each sub-algorithm i, having access to the entire parent population, creates $N_0^i = N/k$ offspring solutions. At this point, the generation counter t is set to zero, and the following procedure is iterated until the relevant stopping criterion has been met (*e.g.* a maximum number of generations reached):

1. Create a combined population $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$ from the parent and offspring populations.
2. Rank and sort population $\mathcal{R}_t$ into nondominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_n$ using the FNSA, and calculate the crowding distance of each solution using Algorithm 7.2.
3. Create the next population $\mathcal{P}_{t+1}$ by including all solutions from the first front $\mathcal{F}_1$, then all solutions from the second front $\mathcal{F}_2$, and so forth, until the inclusion of all solutions from the next front would result in a population size greater than N. In order to limit the size of $\mathcal{P}_{t+1}$ to N, the solutions in this next front are sorted in decreasing order of crowding distance. Solutions from this sorted front are then included one-by-one in this order until $|\mathcal{P}_{t+1}| = N$.
4. Determine the number of successful solutions created by each sub-algorithm, S_{t+1}^i, and calculate the new number of offspring solutions that each sub-algorithm should generate, N_{t+1}^i, according to equation (9.1).
5. Use each sub-algorithm i to generate N_{t+1}^i new offspring solutions, thus creating the next offspring population $\mathcal{Q}_{t+1}$ of size N.
6. Increment the value of the generation counter $t \leftarrow t + 1$.

In the implementation of the AMALGAM method by Vrugt and Robinson [226], the initial parent population was generated using Latin hypercube sampling [133] and the minimum value of N_t^i was set to 5. Furthermore, the authors employed four sub-algorithms, namely the NSGA-II, a PSO algorithm, an adaptive Metropolis search algorithm, and differential evolution. Their choice of sub-algorithms was motivated by the outcome of numerical experiments [226].

9.3 The AMALGAM method for constrained MICFMO

Based on the outcome of the comparative studies performed in Chapter 8, it was decided that the NSGA-II, the P-ACO algorithm, the MOOCEM and the MOVNS algorithm, all employing the MPF constraint handling technique, should be considered for inclusion as sub-algorithms in the AMALGAM method for solving the constrained MICFMO problem.

A number of modifications to the original AMALGAM formulation is necessary to incorporate the aforementioned sub-algorithms into the method. Apart from the NSGA-II, which may be

Algorithm 9.1: The AMALGAM method [226]

Input : An MOP (possibly constrained), a population size N, a maximum number of generations $t_{\max}$, and a set of k sub-algorithms.
Output: An approximate Pareto set, $\widetilde{\mathcal{P}}_S$.

Randomly generate an initial population $\mathcal{P}_0$ of size N
Rank and sort $\mathcal{P}_0$ using the FNSA in Algorithm 7.1
Set $N_0^i \leftarrow N/k$
Use each sub-algorithm $i \in \{1, \ldots, k\}$ to generate N_0^i new offspring solutions, and create offspring population $\mathcal{Q}_0$ of size N
$t \leftarrow 0$
while $t < t_{max}$ **do**
    $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$
    Partition $\mathcal{R}_t$ into nondominated fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ using the FNSA in Algorithm 7.1
    $\mathcal{P}_{t+1} \leftarrow \emptyset$
    $i \leftarrow 1$
    while $|\mathcal{P}_{t+1}| < N$ **do**
        if $|\mathcal{P}_{t+1}| + |\mathcal{F}_i| \leq N$ **then**
            $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i$
            $i \leftarrow i + 1$
        else
            Calculate the crowding distance for each solution in $\mathcal{F}_i$ using Algorithm 7.2
            Sort $\mathcal{F}_i$ in decreasing order of crowding distance
            $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \{\text{the first } N - |\mathcal{P}_{t+1}| \text{ solutions in } \mathcal{F}_i\}$
        end if
    end while
    Calculate the crowding distance for each solution in $\mathcal{P}_{t+1}$ using Algorithm 7.2
    Determine, for each $i \in \{1, \ldots, k\}$, the number of successful solutions, S_{t+1}^i, contributed by sub-algorithm i to $\mathcal{P}_{t+1}$
    Calculate N_{t+1}^i according to (9.1) for each $i \in \{1, \ldots, k\}$
    Use each sub-algorithm $i \in \{1, \ldots, k\}$ to generate N_{t+1}^i new offspring solutions, and create offspring population $\mathcal{Q}_{t+1}$ of size N
    $t \leftarrow t + 1$
end while
$\widetilde{\mathcal{P}}_S \leftarrow \mathcal{P}_{t_{\max}}$

employed within the AMALGAM method as is, the remaining sub-algorithms require supplementary information other than the current population in which a solution's fitness is determined by its Pareto rank and crowding distance.

In order to employ the P-ACO algorithm, for instance, its pheromone matrix is introduced into the AMALGAM method. Whenever an offspring solution is created using the P-ACO sub-algorithm, a local pheromone update is performed. The global pheromone update is, however, performed using the entire offspring population so as to facilitate global information sharing. Similarly, an elite set and corresponding probability matrix is introduced into the AMALGAM method so as to employ the MOOCEM as a sub-algorithm. The entire offspring population is considered for possible inclusion in the elite set during each generation, before updating the probability matrix. Finally, in order to employ the MOVNS algorithm, a nondominated archive is introduced into the AMALGAM method. This archive corresponds to the aforementioned elite set if the ranking threshold of the latter is set to zero. As before, the entire offspring population is considered for possible inclusion in the nondominated archive, before (possibly) marking the neighbourhood of a solution as "explored."

Introduction of the elite set and nondominated archive increases the complexity of the AMALGAM method, while also necessitating additional computer memory during the algorithmic execution. It is, however, envisaged that the improved generality and/or performance of the method is an acceptable trade-off for these complicating factors.

The AMALGAM method for constrained MICFMO, as described above, was also implemented within the Matlab software suite [214] so that objective and constraint function evaluations may be performed using the ANNs constructed in Chapter 6. In this implementation, the minimum value of N_t^i was set to 4 and the population size was selected as $N = 30$. As already mentioned, the MPF technique was employed across the board as constraint handling technique and the same severity factor adopted before was employed here. Finally, the same tuning parameter values adopted in §8.2.3 for each individual metaheuristic were also selected for use in the sub-algorithms within the AMALGAM method.

The following subscript notation is adopted in this chapter to distinguish between AMALGAM variants in which different combinations of the sub-algorithms are implemented. The subscript 'n' denotes the NSGA-II, 'p' denotes the P-ACO algorithm, 'm' denotes the MOOCEM, and 'v' denotes the MOVNS algorithm. Accordingly, $\text{AMALGAM}_{\text{npmv}}$ denotes the variant of the AMALGAM method in which all four aforementioned sub-algorithms are implemented.

A pilot study involving the $\text{AMALGAM}_{\text{npmv}}$ method was performed in order to test the basic behaviour of the sub-algorithms. In this pilot study, the $\text{AMALGAM}_{\text{npmv}}$ method was applied to solve each problem instance in the MICFMO test suite of §8.1. As before, a stopping criterion of 1 050 evaluations was imposed and fifty optimisation runs were executed in respect of each problem instance. Graphs containing the average number $\overline{S}_t^i$ (over the fifty optimisation runs) of successful offspring solutions contributed by sub-algorithm i per generation t are presented in Figure 9.1 for each problem instance within class 1. The corresponding graphs for each problem instance in class 2 and class 3 are presented in Figures 9.2 and 9.3, respectively.

It may be observed in Figures 9.1–9.3 that, out of the four sub-algorithms, the NSGA-II and the P-ACO algorithm clearly contribute the highest number of offspring solutions (on average over the fifty optimisation runs) to the new parent population per generation. Based on this observation, the following four variants of the AMALGAM method are selected for a more comprehensive investigation: $\text{AMALGAM}_{\text{npmv}}$, $\text{AMALGAM}_{\text{npm}}$, $\text{AMALGAM}_{\text{np}}$ and $\text{AMALGAM}_{\text{npv}}$. These variants correspond to the case in which the NSGA-II and the P-ACO algorithm are always employed as sub-algorithms in the AMALGAM method, together with all possible combinations of using the MOOCEM and the MOVNS algorithm.

9.4 Experimental design

The aforementioned promising variants of the AMALGAM method are investigated in a comparative study aimed at determining their ability to conduct constrained MICFMO on the test suite of MICFMO problem instances created in §8.1.

9.4.1 The two stages of comparison

The comparative study consists of two stages. During the first stage, the four variants of the AMALGAM method were applied to solve each problem instance in the MICFMO test suite. A stopping criterion of 1 050 evaluations was, again, imposed (so that the practical limitation of MICFMO for the SAFARI-1 reactor is adhered to) and fifty optimisation runs were executed in respect of each problem instance. As before, the outcome of each run is an approximation front

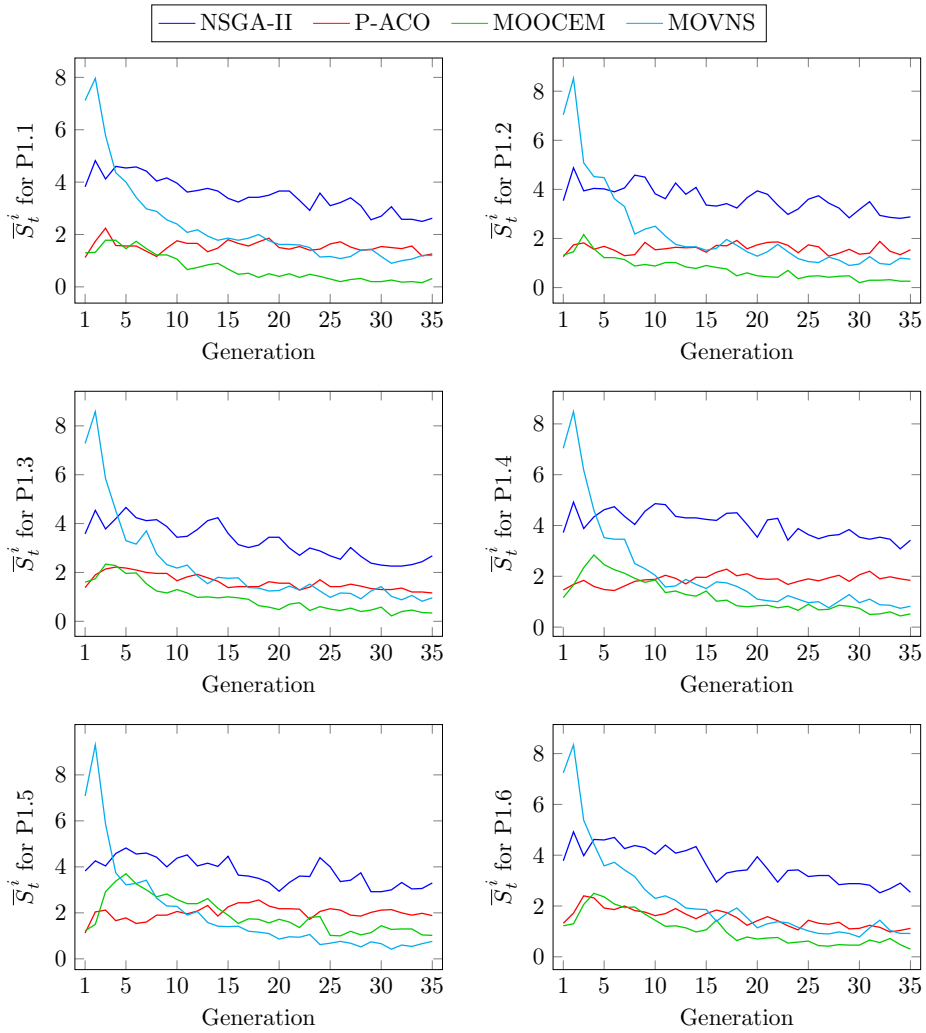


FIGURE 9.1: *The average number $\overline{S}_t^i$ of successful offspring solutions contributed by sub-algorithm i in the AMALGAM$_{npmv}$ method per generation t when solving the problem instances in class 1 of §8.1.*

corresponding to *feasible* solutions only. The same fixed set of fifty different random number generator seeds utilised in §8.2 was employed here for each problem instance, along with the same fixed set of random initial solutions. Accordingly, matched samples were obtained. The different variants of the AMALGAM method were then compared to one another with an aim to identify which is the most suitable in the context of constrained MICFMO.

During the second stage of the study, the results obtained by this preferred variant of the AMALGAM method were compared to the results obtained in Chapter 8 by its corresponding sub-algorithms, *i.e.* when they were employed individually to solve the problem instances in the MICFMO test suite. Since the same random number generator seeds and initial solutions were adopted in all cases, matched samples were obtained within each problem instance. The aim in this stage is to determine whether the AMALGAM method is able to outperform its constituant sub-algorithms in order to pronounce on whether there is a benefit in using the AMALGAM method instead of the individual MOAs.

All the objective function values in the approximation fronts were scaled, as before, to the range $(0, 1)$ using the maximum and minimum values attained for each objective.

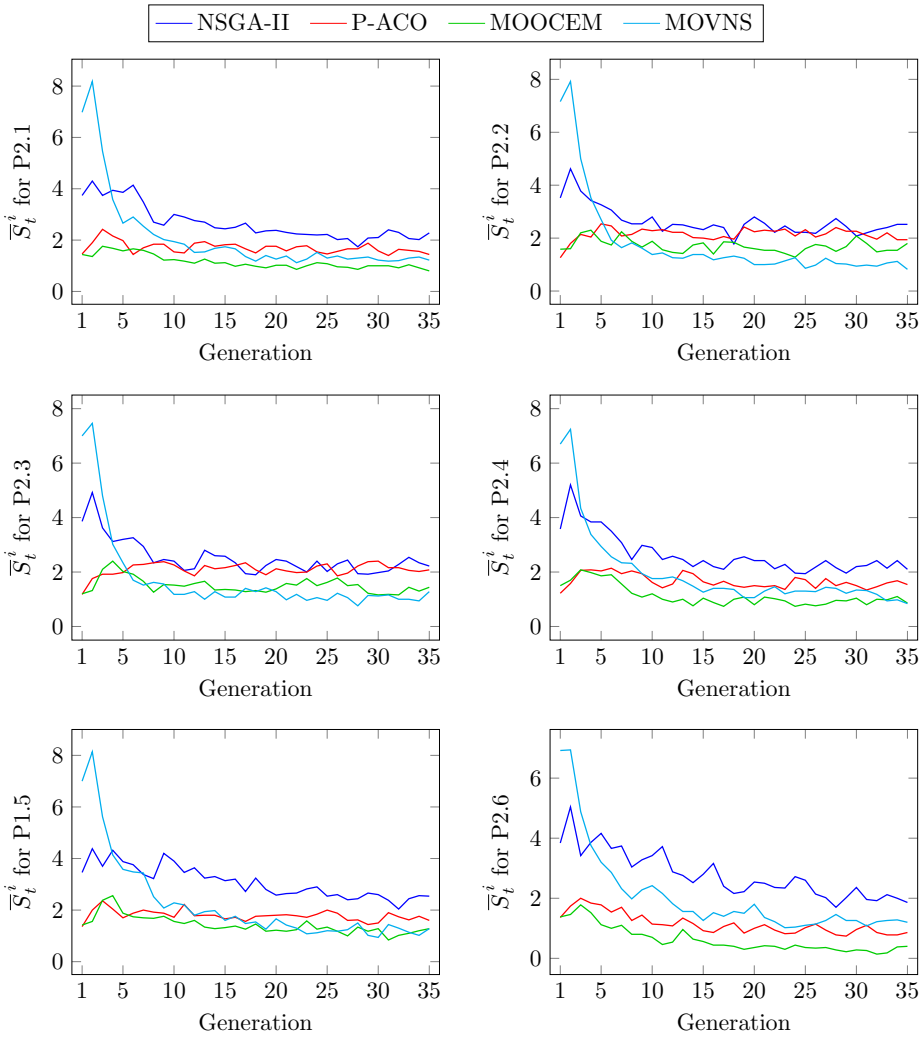


FIGURE 9.2: *The average number $\overline{S}_t^i$ of successful offspring solutions contributed by sub-algorithm i in the AMALGAM$_{npmv}$ method per generation t when solving the problem instances in class 2 of §8.1.*

9.4.2 Performance indicator considerations

Reference approximation fronts required during the calculation of the I_{HVD} and I_{R2} indicators (described in §7.8) were determined here in the same manner as in §8.2. For each problem instance, the approximation fronts yielded by all the optimisation runs, for all four variants of the AMALGAM method, together with those yielded separately by the NSGA-II, the P-ACO algorithm, the MOOCEM and the MOVNS algorithm from Chapter 8, were pooled together. A reference approximation front associated with that problem instance was then determined by identifying the combined nondominated front from this pool.

The reference points and utopian objective vectors selected in §8.2.2 were also adopted in the comparative study, along with the same sets of uniformly dispersed weight vectors required in the calculation of I_{R2}.

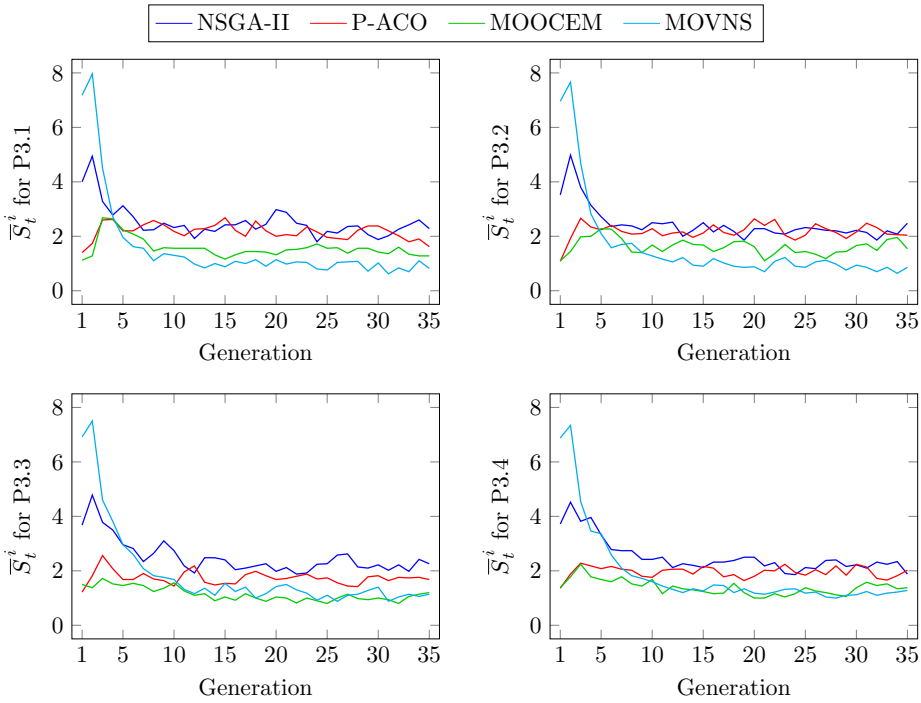


FIGURE 9.3: *The average number $\overline{S}_t^i$ of successful offspring solutions contributed by sub-algorithm i in the AMALGAM$_{\text{npmv}}$ method per generation t when solving the problem instances in class 3 of §8.1.*

9.4.3 Statistical analysis considerations

A nonparametric statistical analysis is conducted in this chapter in the context of the results obtained. Both a single-problem analysis and multi-problem analysis are, again, considered here. The Friedman test with the Nemenyi *post hoc* procedure is employed for the comparison between different hyperheuristic variants during the first stage.

Along with Friedman test, a different *post hoc* multiple-comparisons procedure is, however, employed for the comparison between the hyperheuristic and its corresponding sub-algorithms during the second stage. Since the aim is to compare each sub-algorithm only to the hyperheuristic (and not to one another), an appropriate *post hoc* procedure to employ is the *Nemenyi, Wilcoxon-Wilcox, Miller* (NWWM) procedure [82]. If there are k sub-algorithms within the selected AMALGAM method variant, then k one-tailed pairwise significance tests are performed in the NWWM procedure — one for each sub-algorithm/AMALGAM pair of samples. This procedure also utilises the Friedman ranks and corrects for the multiple inferences it makes.

The Statistics and Machine Learning Toolbox [216] within the Matlab software suite [214] is utilised to apply the Friedman test and Nemenyi procedure. The NWWM procedure, however, is applied utilising NSM3 package [190] within the programming language R [169]. Note that a significance level of $\tilde{\alpha} = 0.05$ was adopted for all the cases presented in this chapter.

9.5 Numerical results

The comparative study results obtained by following the experimental design discussed in §9.4 are presented in this section. As before, all the calculations were performed using the same personal computer described in §6.6.

9.5.1 First stage results: AMALGAM variants

During this first stage of comparison, the AMALGAM$_{\text{npmv}}$, AMALGAM$_{\text{npm}}$, AMALGAM$_{\text{np}}$ and AMALGAM$_{\text{npv}}$ variants are compared relative to one another. A single-problem analysis is performed first, followed by a multi-problem analysis within each problem instance class.

Single-problem analysis

In the single-problem analysis, there are four samples of data for each problem instance in the test suite — matched indicator values corresponding to the fifty optimisation runs for each variant of the AMALGAM method. These samples, obtained for the problem instances in classes 1, 2 and 3, are presented in the form of box plots in Figures 9.4, 9.5 and 9.6, respectively. The average value of each sample is included, as before, as black diamond points in the graphs. The aim is to compare the box plots with one another within each graph. A visual exploratory analysis may thus be performed in respect of each problem instance, with respect to both the I_{HVD} and I_{R2} performance indicators.

It is observed in Figures 9.4–9.6 that all four variants of the AMALGAM method perform fairly similarly (with respect to both indicators) across the problem instances in all three classes of the test suite. This behaviour also appears to be true in respect of the robustness (in terms of sample variability) of the different AMALGAM method variants. Due the similarities present in these results, limited additional insight may be gained by conducting a visual exploratory analysis.

The next step is to determine whether there is a statistically significant difference between the variants of the AMALGAM method in this single-problem analysis. The Friedman test is therefore applied to the samples obtained for each test problem instance. If a significant difference has been detected, the Nemenyi *post hoc* procedure is applied in order to identify the individual differences between pairs of samples. Since there are four variants in this study, the Nemenyi procedure performs $\binom{4}{2} = 6$ pairwise significance tests.

The Friedman test detects a statistically significant difference (for $\tilde{\alpha} = 0.05$) for six problem instances in the test suite with respect to both indicators. A significant difference is also detected for two additional problem instances: P2.3 with respect to I_{HVD}, and P3.4 with respect to I_{R2}. The corresponding p-values obtained by the Friedman test are presented in Table 9.1.

Sample	Friedman test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
I_{HVD}	0.2359	0.1638	0.0788	0.5351	**0.0118**	0.0814
I_{R2}	0.6468	0.1968	0.3843	0.6254	**0.0087**	0.0709
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
I_{HVD}	0.3110	**0.0001**	**0.0352**	0.5543	**0.0124**	0.2091
I_{R2}	0.2244	**0.0010**	0.1777	0.5943	**0.0016**	0.0672
	P3.1	P3.2	P3.3	P3.4		
I_{HVD}	**0.0415**	**0.0173**	**0.0025**	0.2407		
I_{R2}	**0.0007**	**0.0001**	**0.0005**	**0.0034**		

TABLE 9.1: *Single-problem analysis results for comparing the four variants of the AMALGAM method. The table contains the p-values obtained by Friedman tests applied to the I_{HVD} and I_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

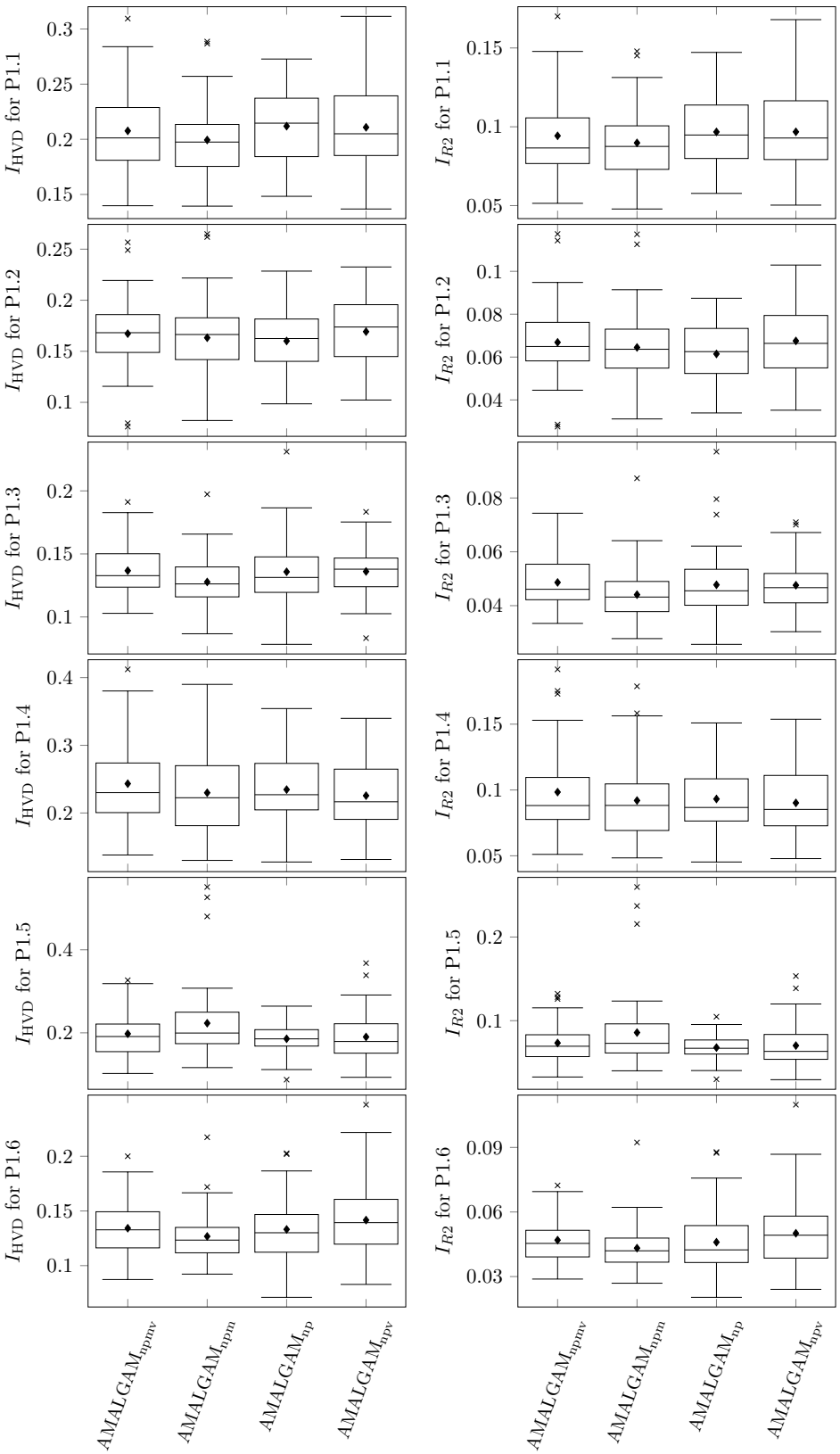


FIGURE 9.4: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all four hyperheuristic variants for each problem instance in class 1 of §8.1.*

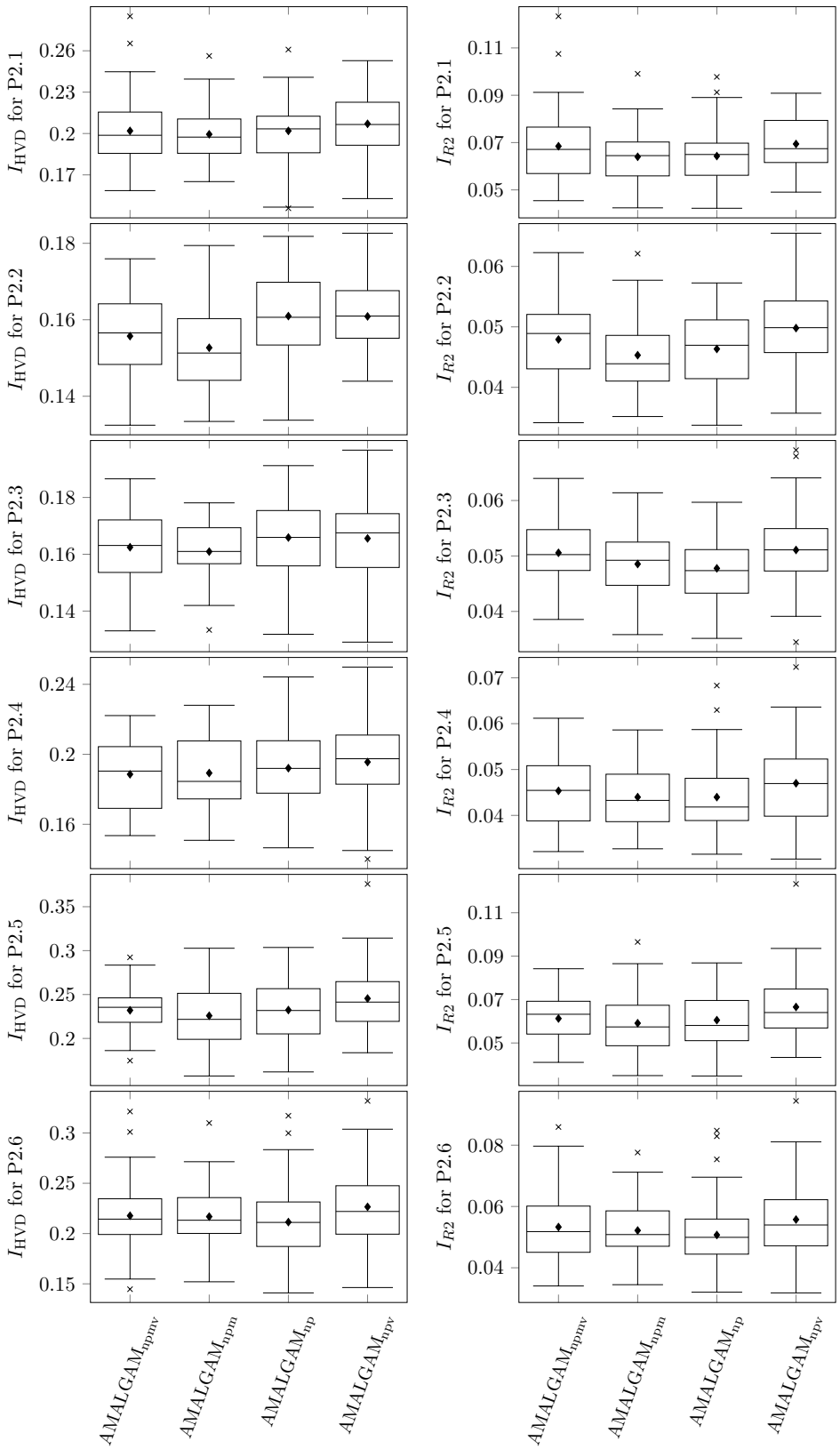


FIGURE 9.5: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all four hyperheuristic variants for each problem instance in class 2 of §8.1.*

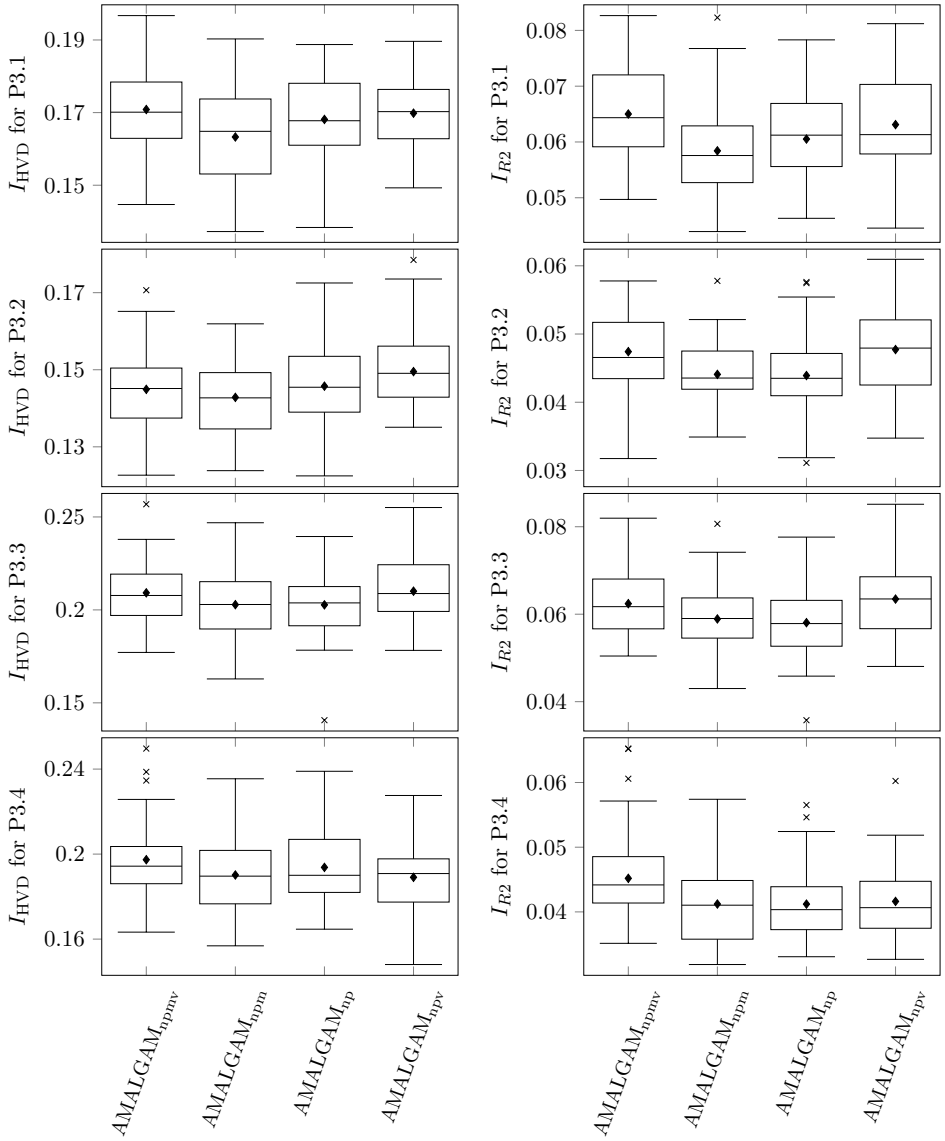


FIGURE 9.6: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by all four hyperheuristic variants for each problem instance in class 3 of §8.1.*

The detailed results of the Nemenyi *post hoc* procedure, applied to the relevant problem instances identified in Table 9.1, may be found in Appendix E. In the majority of instances in which a statistically significant difference is detected during the *post hoc* procedure, the AMALGAM$_{\text{npm}}$ method is involved. These differences are, however, scattered throughout the results (as was the case in §8.4.2), which make any meaningful inferences difficult. As such, these detections are not mentioned individually in this section.

Multi-problem analysis

A multi-problem analysis, *within each class* of problem instances, is performed next. As before, an average indicator value, calculated over the fifty optimisation runs, is determined for each AMALGAM method variant/problem instance pair. The average indicator values for I_{HVD} and I_{R2} correspond to the black diamond points in Figures 9.4–9.6.

In order to determine whether there is a statistically significant difference between the four variants of the AMALGAM method in the multi-problem analysis, the Friedman test is applied to the average indicator value samples for each class of problem instances. The resulting p-values are presented in Table 9.2 and, as before, bold-faced entries correspond to the detection of a significant difference (for $\tilde{\alpha} = 0.05$).

Sample	Friedman test p-values		
	Class 1	Class 2	Class 3
I_{HVD}	0.2615	**0.0141**	0.2123
I_{R2}	0.2165	**0.0010**	**0.0194**

TABLE 9.2: *Multi-problem analysis results for comparing the four variants of the AMALGAM method. The table contains the p-values obtained by Friedman tests applied to the average indicator value samples for each problem instance class. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

The Friedman test is able to detect a statistically significant difference between the AMALGAM method variants in problem instance class 2 for both indicators, and in class 3 for the I_{R2} indicator. The detailed results obtained from the pairwise significance tests performed during the subsequent application of the Nemenyi *post hoc* procedure may be found in Appendix E.

The same strategy adopted in §8.4.2, which involves the calculation of average Friedman ranks R_{avg}, is followed in this section in order to assist in the interpretation of the multi-problem analysis results. The variants of the AMALGAM method are therefore ranked according to their average rank values, within each problem instance class (for both I_{HVD} and I_{R2}). These rank results for class 1, class 2 and class 3 are presented in Tables 9.3, 9.4 and 9.5, respectively. Where applicable, if the Nemenyi procedure detects a significant difference between two variants, the detection is indicated in the table with matching alphabetic letters next to the corresponding variants in the column labelled "S.Diff".

According to the rankings in Table 9.3, the $\text{AMALGAM}_{\text{npm}}$ method is the best-performing hyperheuristic variant in problem instance class 1 with respect to both performance indicators, while the $\text{AMALGAM}_{\text{npmv}}$ method performs the worst. It may also be observed that the different variants have identical rankings with respect to both I_{HVD} and I_{R2}. The performances of the AMALGAM variants are therefore very consistent across the indicators for this problem instance class. Since the Friedman test does not detect a statistically significant difference between the variants of the AMALGAM method for problem instance class 1, the Nemenyi *post hoc* procedure is not applied — hence there are no entries in the column labelled "S.Diff" within Table 9.3.

I_{HVD} indicator results			I_{R2} indicator results		
R_{avg}	Hyperheuristic	S.Diff	R_{avg}	Hyperheuristic	S.Diff
1.833	$\text{AMALGAM}_{\text{npm}}$	—	1.833	$\text{AMALGAM}_{\text{npm}}$	—
2.167	$\text{AMALGAM}_{\text{np}}$	—	2.167	$\text{AMALGAM}_{\text{np}}$	—
2.833	$\text{AMALGAM}_{\text{npv}}$	—	2.833	$\text{AMALGAM}_{\text{npv}}$	—
3.167	$\text{AMALGAM}_{\text{npmv}}$	—	3.167	$\text{AMALGAM}_{\text{npmv}}$	—

TABLE 9.3: *Multi-problem analysis results involving average Friedman ranks for problem instance class 1. The variants of the AMALGAM method are ranked according to their average rank values. Significant differences between any two variants are denoted by matching alphabetic letters in the column labelled "S.Diff".*

I_{HVD} indicator results			I_{R2} indicator results		
R_{avg}	Hyperheuristic	S.Diff	R_{avg}	Hyperheuristic	S.Diff
1.333	AMALGAM$_{\text{npm}}$	a	1.5	AMALGAM$_{\text{npm}}$	a
2.167	AMALGAM$_{\text{npmv}}$	—	1.5	AMALGAM$_{\text{np}}$	b
2.833	AMALGAM$_{\text{np}}$	—	3	AMALGAM$_{\text{npmv}}$	—
3.667	AMALGAM$_{\text{npv}}$	a	4	AMALGAM$_{\text{npv}}$	ab

TABLE 9.4: *Multi-problem analysis results involving average Friedman ranks for problem instance class 2. The variants of the AMALGAM method are ranked according to their average rank values. Significant differences between any two variants are denoted by matching alphabetic letters in the column labelled "S.Diff".*

In Table 9.4, it may be observed that the AMALGAM$_{\text{npm}}$ and AMALGAM$_{\text{np}}$ methods are jointly the best-performing hyperheuristic variants in class 2 with respect to I_{R2}, and they are significantly different from the AMALGAM$_{\text{npv}}$ method (which is the worst-performing variant with respect to both performance indicators). Similarly, the AMALGAM$_{\text{npm}}$ method is the best-performing variant with respect to I_{HVD}, and also significantly different from the AMALGAM$_{\text{npv}}$.

I_{HVD} indicator results			I_{R2} indicator results		
R_{avg}	Hyperheuristic	S.Diff	R_{avg}	Hyperheuristic	S.Diff
1.5	AMALGAM$_{\text{npm}}$	—	1.25	AMALGAM$_{\text{np}}$	—
2.25	AMALGAM$_{\text{np}}$	—	1.75	AMALGAM$_{\text{npm}}$	—
3	AMALGAM$_{\text{npv}}$	—	3.5	AMALGAM$_{\text{npmv}}$	—
3.25	AMALGAM$_{\text{npmv}}$	—	3.5	AMALGAM$_{\text{npv}}$	—

TABLE 9.5: *Multi-problem analysis results involving average Friedman ranks for problem instance class 3. The variants of the AMALGAM method are ranked according to their average rank values. Significant differences between any two variants are denoted by matching alphabetic letters in the column labelled "S.Diff".*

In Table 9.5, it may be observed that no statistically significant differences are detected between any pair of hyperheuristic variants with respect to I_{R2}, although the Friedman test yields a significant p-value. This outcome is similar to what was observed in Table 8.8, and may be attributed to the conservatism present in the Nemenyi procedure. According to the rankings in Table 9.5, the AMALGAM$_{\text{npm}}$ method is the best-performing variant with respect to I_{HVD}, while the AMALGAM$_{\text{np}}$ method performs the best with respect to I_{R2}.

Conclusion

Although the single-problem analysis conducted in this section indicates that there is little to choose between the four variants of the AMALGAM method, the corresponding multi-problem analysis reveals that the AMALGAM$_{\text{npm}}$ method is consistently the best-performing hyperheuristic variant with respect to both I_{HVD} and I_{R2}, across all three problem instance classes. Only for class 3, and with respect to I_{R2}, does this variant not rank at the top — it is, however, ranked second. As such, the AMALGAM$_{\text{npm}}$ method is selected as the preferred hyperheuristic variant in the context of constrained MICFMO, and is carried forward into the second stage of the comparative study.

9.5.2 Second stage results: AMALGAM sub-algorithms

The preferred AMALGAM$_{\text{npm}}$ method, carried over from the first stage of the comparative study, is now compared against the NSGA-II, the P-ACO algorithm and the MOOCEM within the second stage of the comparative study. These three MOAs constitute the sub-algorithms of the AMALGAM$_{\text{npm}}$ method. A single-problem analysis is performed first, followed by a multi-problem analysis within each problem instance class.

Single-problem analysis

In the single-problem analysis, there are four samples of data for each problem instance in the test suite — matched indicator values corresponding to the fifty optimisation runs for the AMALGAM$_{\text{npm}}$ method, and its three constituent sub-algorithms (whose values were taken from Chapter 8). These samples, obtained for the problem instances in class 1, class 2 and class 3, are presented in the form of box plots in Figures 9.7, 9.8 and 9.9, respectively. The average value of each sample is included, as before, as black diamond points in the graphs. A visual exploratory analysis may be performed in respect of each problem instance, with respect to both the I_{HVD} and I_{R2} indicators, by comparing the box plots of the sub-algorithms against that of the AMALGAM$_{\text{npm}}$ method within each graph.

It is observed in Figures 9.7–9.9 that the AMALGAM$_{\text{npm}}$ method performs well compared to its sub-algorithms (with respect to both performance indicators) across the problem instances in all three classes of the test suite. The method therefore seems to scale well as the number of objectives in a problem instance increases. Furthermore, the robustness of the AMALGAM$_{\text{npm}}$ method (in terms of sample variability) appears to be similar to that of its sub-algorithms.

Problem instance P1.5 appears to be the only instance in which the AMALGAM$_{\text{npm}}$ method is outperformed (with respect to both indicators) by one of its sub-algorithms — the P-ACO algorithm — as may be seen in Figure 9.7. Furthermore, in Figure 9.8, it is observed that the performance of the MOOCEM is of similar quality to that of the AMALGAM$_{\text{npm}}$ method in the context of class 2 problem instances, especially with respect to I_{HVD}.

The next step is to determine whether there is a statistically significant difference between the AMALGAM$_{\text{npm}}$ method and the corresponding individual sub-algorithms in this single-problem analysis. The Friedman test is therefore applied to the samples obtained for each test problem instance. As explained in §9.4.3, the aim here is to compare each separate sub-algorithm to the AMALGAM$_{\text{npm}}$ method (and not to one another). Therefore, if a statistically significant difference has been detected by the Friedman test, the NWWM *post hoc* procedure is applied in order to identify which sub-algorithm differs from the AMALGAM$_{\text{npm}}$ method. Since there are three sub-algorithms within this preferred variant of the AMALGAM method, the NWWM procedure performs three *one-tailed* pairwise significance tests. Due to the one-tailed nature of these tests, if a statistically significant difference is detected by the *post hoc* procedure, it may be inferred that the AMALGAM$_{\text{npm}}$ method performs significantly *better* than the associated sub-algorithm.

The Friedman test detects a statistically significant difference (for $\tilde{\alpha} = 0.05$) for every problem instance in the test suite with respect to both indicators. The corresponding p-values are presented in Table 9.6. This outcome does not come as a surprise, given the good performance observed for the AMALGAM$_{\text{npm}}$ method during the visual exploratory analysis.

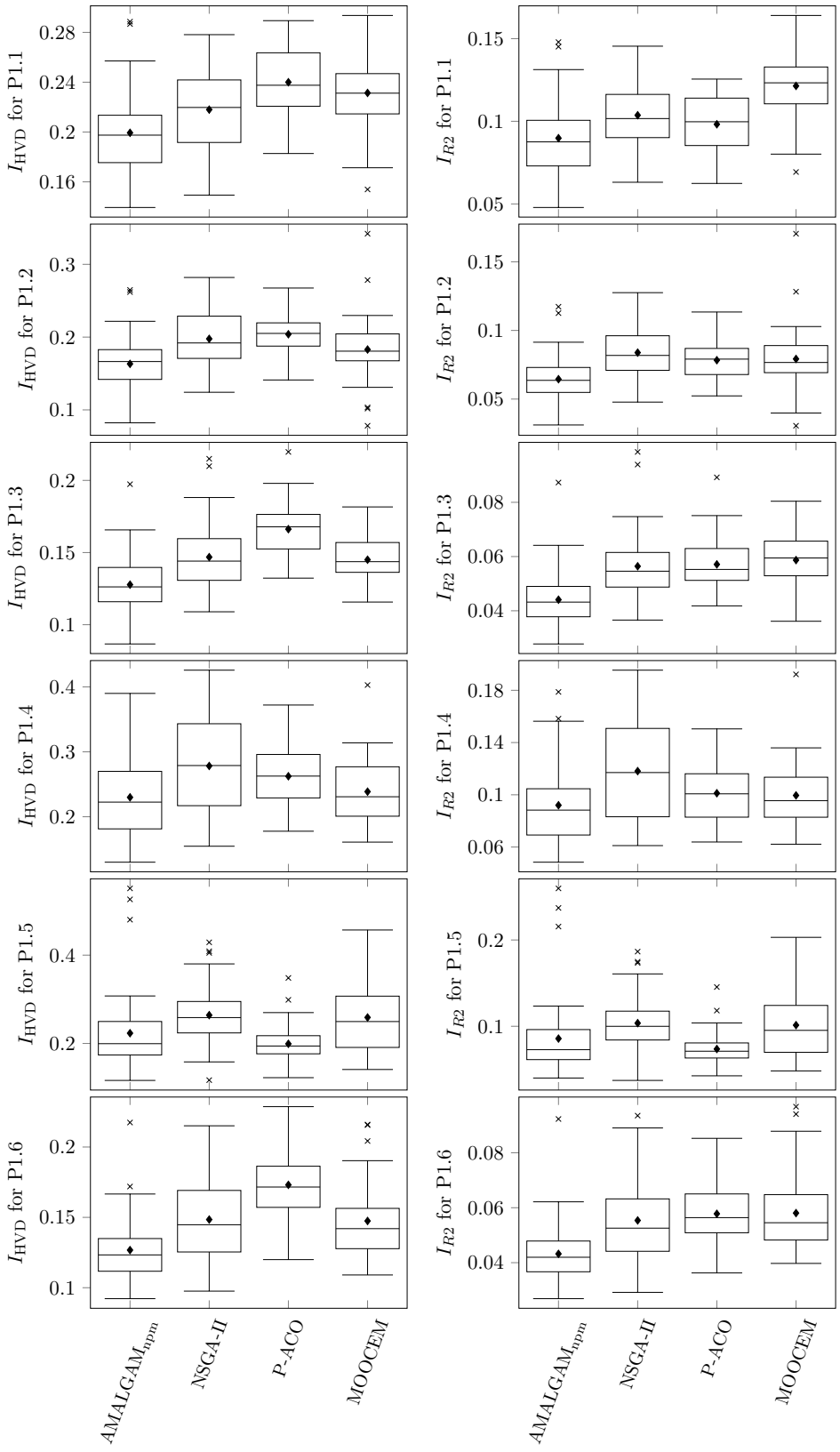


FIGURE 9.7: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by the hyperheuristic and corresponding sub-algorithm metaheuristics for each problem instance in class 1 of §8.1.*

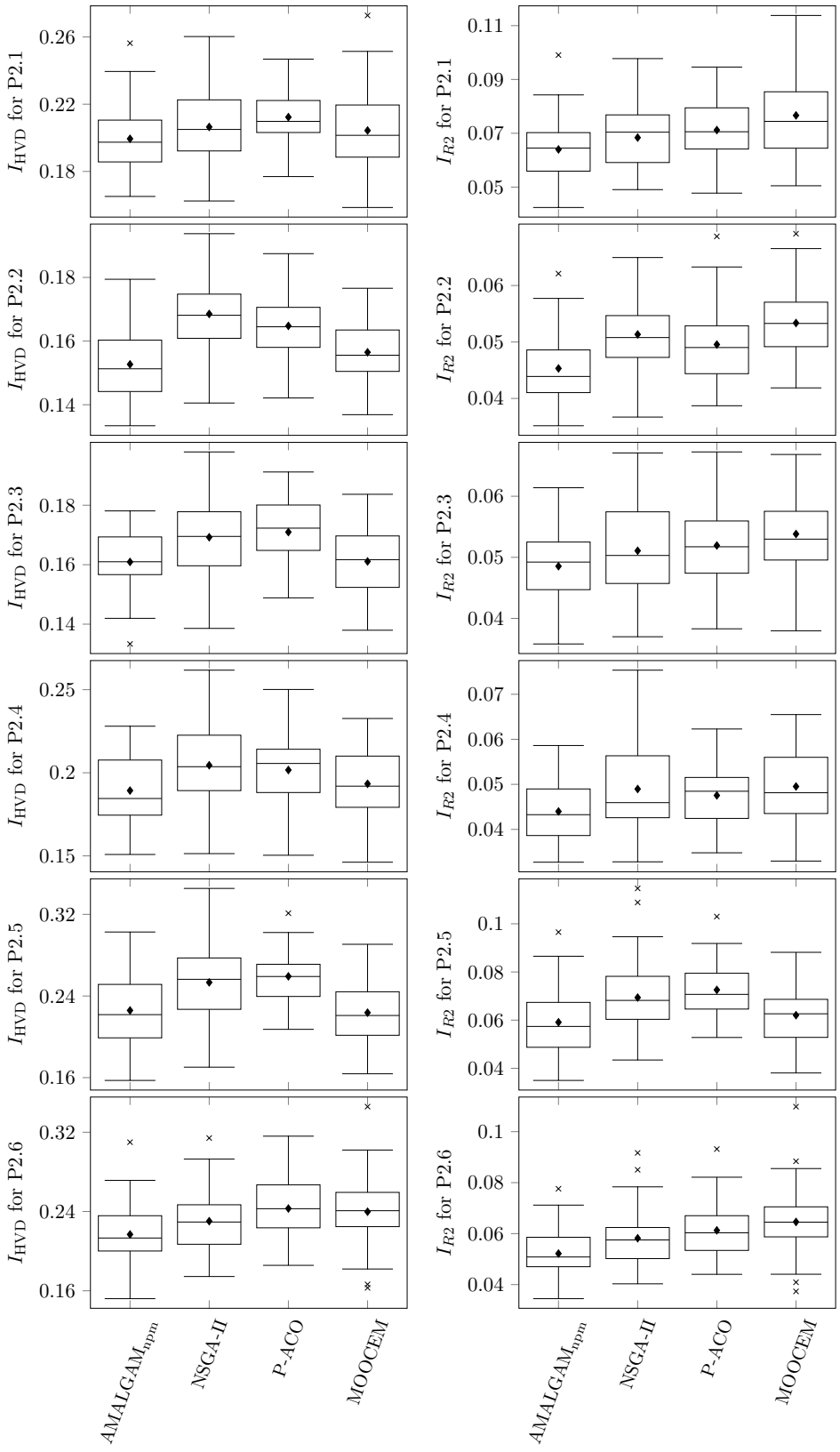


FIGURE 9.8: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by the hyperheuristic and corresponding sub-algorithm metaheuristics for each problem instance in class 2 of §8.1.*

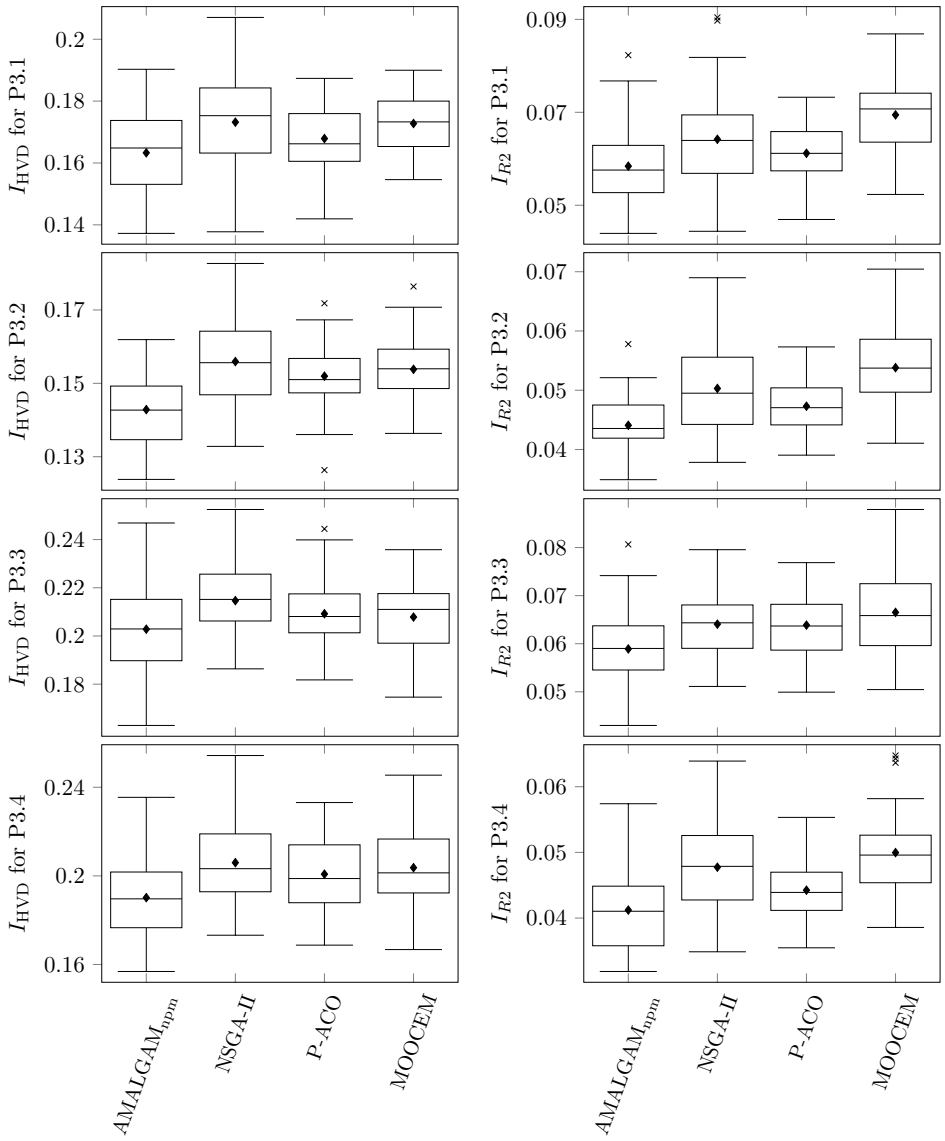


FIGURE 9.9: *Box plots of the I_{HVD} samples (on the left-hand side) and I_{R2} samples (on the right-hand side) obtained by the hyperheuristic and corresponding sub-algorithm metaheuristics for each problem instance in class 3 of §8.1.*

The NWWM *post hoc* procedure is therefore applied to all the problem instances. The detailed statistical results obtained from the corresponding pairwise significance tests may be found in Appendix E. Only a summary of the findings is presented here in order to improve the readability of the main text.

Recall that there are sixteen problem instances in the MICFMO test suite. It is found that the $\text{AMALGAM}_{\text{npm}}$ method performs significantly better than the NSGA-II in fourteen problem instances with respect to I_{HVD}, and in another fourteen instances with respect to I_{R2}. Similarly, the method performs significantly better than the P-ACO algorithm in thirteen problem instances with respect to I_{HVD}, and in another ten instances with respect to I_{R2}. Finally, it is also found that the $\text{AMALGAM}_{\text{npm}}$ method performs significantly better than the MOOCEM in eight problem instances with respect to I_{HVD}, and in another fourteen instances with respect to I_{R2}.

Sample	Friedman test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
I_{HVD}	$\mathbf{2.869\times10^{-6}}$	$\mathbf{5.064\times10^{-6}}$	$\mathbf{2.448\times10^{-14}}$	$\mathbf{9.798\times10^{-6}}$	$\mathbf{4.260\times10^{-7}}$	$\mathbf{1.219\times10^{-11}}$
I_{R2}	$\mathbf{1.280\times10^{-8}}$	$\mathbf{2.586\times10^{-5}}$	$\mathbf{1.961\times10^{-10}}$	$\mathbf{3.110\times10^{-5}}$	$\mathbf{3.297\times10^{-7}}$	$\mathbf{2.600\times10^{-9}}$
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
I_{HVD}	$\mathbf{1.065\times10^{-2}}$	$\mathbf{3.190\times10^{-8}}$	$\mathbf{9.749\times10^{-4}}$	$\mathbf{5.463\times10^{-3}}$	$\mathbf{3.846\times10^{-8}}$	$\mathbf{2.268\times10^{-4}}$
I_{R2}	$\mathbf{8.528\times10^{-6}}$	$\mathbf{3.116\times10^{-8}}$	$\mathbf{1.891\times10^{-2}}$	$\mathbf{1.482\times10^{-3}}$	$\mathbf{1.032\times10^{-6}}$	$\mathbf{2.274\times10^{-6}}$
	P3.1	P3.2	P3.3	P3.4		
I_{HVD}	$\mathbf{7.014\times10^{-4}}$	$\mathbf{7.897\times10^{-7}}$	$\mathbf{8.724\times10^{-3}}$	$\mathbf{8.412\times10^{-4}}$		
I_{R2}	$\mathbf{3.136\times10^{-9}}$	$\mathbf{2.530\times10^{-11}}$	$\mathbf{5.532\times10^{-5}}$	$\mathbf{1.146\times10^{-6}}$		

TABLE 9.6: *Single-problem analysis results for comparing the AMALGAM$_{npm}$ method against its constituent sub-algorithms. The table contains the p-values obtained by Friedman tests applied to the I_{HVD} and I_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Multi-problem analysis

A multi-problem analysis, *within each class* of problem instances, is performed next. As before, an average indicator value, calculated over the fifty optimisation runs, is determined for each MOA/problem instance pair. The different MOAs are the AMALGAM$_{\text{npm}}$ method and its three constituent sub-algorithms (whose values were taken from Chapter 8). The average indicator values for I_{HVD} and I_{R2} correspond to the black diamond points in Figures 9.7–9.9.

In order to determine whether there is a significant difference between the AMALGAM$_{\text{npm}}$ method and the corresponding sub-algorithms in this multi-problem analysis, the Friedman test is applied to the average indicator value samples for each class of problem instances. The resulting p-values are presented in Table 9.7 and, as before, bold-faced entries correspond to the detection of a statistically significant difference (for $\tilde{\alpha} = 0.05$).

Sample	Friedman test p-values		
	Class 1	Class 2	Class 3
I_{HVD}	**0.0141**	**0.0032**	**0.0112**
I_{R2}	**0.0141**	**0.0038**	**0.0074**

TABLE 9.7: *Multi-problem analysis results for comparing the AMALGAM$_{npm}$ method against its constituent sub-algorithms. The table contains the p-values obtained by Friedman tests applied to the average indicator value samples for each problem instance class. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Although the sample sizes are small, the Friedman test is able to detect a statistically significant difference for every problem instance class with respect to both indicators. Accordingly, the NWWM *post hoc* procedure is applied to all three problem instance classes and the corresponding p-values obtained from the pairwise significance tests performed during the procedure are presented in Table 9.8.

It may be observed that the AMALGAM$_{\text{npm}}$ method performs significantly better than the NSGA-II in all three problem instance classes with respect to I_{HVD}, and also in classes 1 and 3 with respect to I_{R2}. The method is also found to perform significantly better than the P-ACO

algorithm in problem instance class 2 with respect to both performance indicators, as well as in class 1 with respect to I_{HVD}. Finally, it is observed in Table 9.8 that the AMALGAM$_{\text{npm}}$ method performs significantly better than the MOOCEM in all three problem instance classes with respect to I_{R2}.

Problem		NWWM procedure p-values		
		NSGA-II	P-ACO	MOOCEM
Class 1	I_{HVD}	**0.0085**	**0.0034**	0.1650
	I_{R2}	**0.0114**	0.1741	**0.0041**
Class 2	I_{HVD}	**0.0111**	**0.0005**	0.3346
	I_{R2}	0.0726	**0.0244**	**0.0002**
Class 3	I_{HVD}	**0.0003**	0.2523	0.0924
	I_{R2}	**0.0484**	0.3481	**0.0001**

TABLE 9.8: *Multi-problem analysis results for comparing the AMALGAM$_{npm}$ method against its constituent sub-algorithms. The table contains the p-values obtained by the NWWM* post hoc *procedure in which three pairwise significance tests are performed for each problem instance class. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Conclusion

Recall from the conclusion in §8.4.2 that the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing MOAs in the metaheuristic comparative study with respect to both the I_{HVD} and I_{R2} indicators, across the problem instances in all three classes. In this second stage of the comparative study, it has been shown that the AMALGAM$_{\text{npm}}$ method significantly outperforms these three MOAs in the majority of instances within the MICFMO test suite — an inference supported by both single-problem and multi-problem statistical analyses. The hyperheuristic achieves the dual goal of raising the level of generality at which MICFMO may be performed and of yielding improved optimisation performance. The AMALGAM$_{\text{npm}}$ method may therefore be recommended as a state-of-the-art MOA for solving (approximately) the constrained MICFMO problem.

9.6 Chapter summary

In this chapter, the AMALGAM method was investigated in terms of its ability to conduct constrained MICFMO in the context of the test suite of problem instances defined in §8.1. The general working of the method was discussed first in §9.2, after which modifications to the original formulation of the method made in this dissertation were described in §9.3. Four variants of the AMALGAM method, depending on the configuration of its sub-algorithms, were also identified for comprehensive investigation.

The experimental design followed during the two-stage comparative study conducted in this chapter was presented in §9.4. During the first stage, the promising variants of the AMALGAM method were compared to one another in order to select a preferred variant. Then, during the second stage, that preferred variant was compared against its constituent sub-algorithms (whose results were taken from Chapter 8), in order to determine whether it is beneficial to use the AMALGAM method instead of the individual MOAs. The numerical results of this comparative study were presented in §9.5.

The results from the first stage of the study in §9.5.1 revealed that the AMALGAM$_{\text{npm}}$ method, whose sub-algorithms are the NSGA-II, the P-ACO algorithm and the MOOCEM, may be selected as the preferred hyperheuristic variant. In the second stage of the comparative study in §9.5.2, it was inferred that the AMALGAM$_{\text{npm}}$ method significantly outperforms the three individual sub-algorithms in the majority of instances within the MICFMO test suite. The hyperheuristic therefore achieved the dual goal of raising the level of generality at which MICFMO may be performed and of yielding improved optimisation performance.

CHAPTER 10

Case studies

Contents

Based on the recommendation in Chapter 9, the preferred AMALGAM$_{\text{npm}}$ method was implemented by the author within a new MICFMO decision support feature in the OSCAR-4 system. In this chapter, the practical applicability of the hyperheuristic is demonstrated by utilising this feature to solve (approximately) a number of realistic case study problem instances, in the contexts of both the SAFARI-1 and HOR reactors.

10.1 Introduction

In the comparative studies performed in Chapters 8 and 9, the performance of each MOA was determined relative to that of the other MOAs (as mentioned in §7.8), because of the absence of the true Pareto front for a given problem instance. Accordingly, the studies did not necessarily reveal whether the solutions obtained by the MOAs are of an acceptable quality for practical usage. Furthermore, the testing was performed using surrogate models for the evaluation of objective and constraint functions instead of an accurate reactor core simulator.

As mentioned above, the AMALGAM$_{\text{npm}}$ method was implemented in the OSCAR-4 system by the author. Unlike in §9.3, however, all nondominated solutions within the elite set are returned by the method in this implementation, instead of only the final population. In order to demonstrate the practical relevance of this new MICFMO capability, it was used to solve three realistic case study problem instances based on the SAFARI-1 reactor, as well as three instances based on the HOR reactor. These reactors were described in Chapter 4.

All the calculations in this chapter were performed on a personal computer with the following specifications: An Intel® Xeon™ E5-2630 CPU with 64 GB RAM operating at 2.60 GHz within a 64-bit operating system. As before, the stopping criterion for the hyperheuristic was set to a maximum of 1 050 evaluations. Furthermore, all tuning parameters in the AMALGAM$_{\text{npm}}$ method were set to the same values adopted in §9.3. Each problem instance was solved five times (based on the availability of computational resources), using a different random number generator seed during each optimisation run. An attainment front for each problem instance was then isolated from the pool of nondominated fronts obtained by the five runs.

These attainment fronts were compared to the actual reload configurations loaded into the cores of the SAFARI-1 and HOR reactors during the operational cycles considered in each problem instance. As before, these configurations are referred to as the *historical SAFARI-1 reload configuration* (HSRC) and the *historical HOR reload configuration* (HHRC), respectively. Note that the HSRC and HHRC were designed according to the current approach followed at each reactor, as described in §4.5.3 and §4.6.3, respectively.

10.2 The SAFARI-1 reactor case study

The three case study problem instances considered in this section are based on the first operational cycle for the SAFARI-1 reactor during the year 2016, which is designated as cycle C1601-1. Each instance corresponds to a realistic scenario that may be pursued at the reactor. The entire constraint set specified in §4.5.2 has to be adhered to in each problem instance. Furthermore, as was the case in §8.1, the specific limiting values of the third constraint (*i.e.* the peak axial production capability in the IPR facilities) have been relaxed so that the HSRC may be included in the feasible region $\mathcal{S}_S$. Finally, the objective function labels adopted in the specification of the problem instances below correspond to those labels listed in Table 4.1.

10.2.1 The problem instances under consideration

In the first case study problem instance, maximisation of the cycle length of the SAFARI-1 reactor and minimisation of its core power peaking factor, are of interest. It therefore conforms to the pursuit of objectives S1 and S2, respectively. Recall that an objective function may be transformed from a minimisation paradigm to a maximisation paradigm by taking its negative value. Accordingly, the first problem instance is a bi-objective MOP in which the goal is to

$$\left.\begin{array}{c} \text{maximise} \quad [f_{\text{S1}}(\boldsymbol{x}), -f_{\text{S2}}(\boldsymbol{x})], \\ \text{subject to} \quad \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{10.1}$$

The second case study problem instance is similar to (5.14) in which the research utilisation of the SAFARI-1 reactor is to be enhanced. Now, however, the cycle length of the reactor is to be maximised in conjunction with the maximisation of the research capability at beam tubes 1 & 2 and at beam tube 5. Accordingly, the optimisation of objectives S1, S5 and S6 is pursued. The second problem instance, therefore, is a tri-objective MOP in which the goal is to

$$\left.\begin{array}{c} \text{maximise} \quad [f_{\text{S1}}(\boldsymbol{x}), f_{\text{S5}}(\boldsymbol{x}), f_{\text{S6}}(\boldsymbol{x})], \\ \text{subject to} \quad \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{10.2}$$

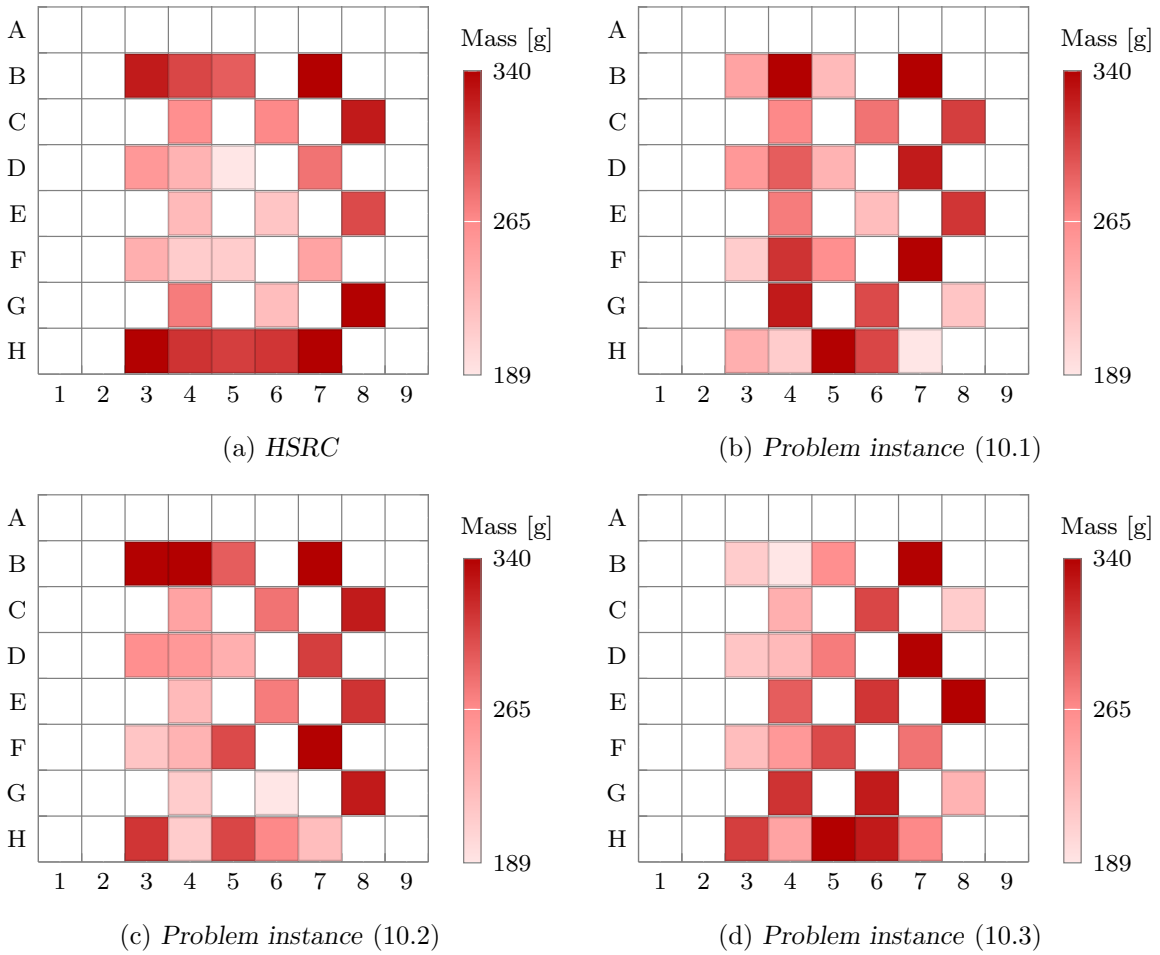
Finally, the third case study problem instance corresponds exactly to (5.15) in which the commercial services rendered by the SAFARI-1 reactor are to be optimised, except that cycle C1601-1

is considered now. Accordingly, the optimisation of objectives S3, S4, S7 and S8 are pursued, namely maximisation of the production of ^{99}Mo isotopes, the utilisation of the silicon doping facility, and the production of isotopes in the two IPR facilities. The third problem instance, therefore, is a tetra-objective MOP in which the goal is to

$$\left.\begin{array}{ll} \text{maximise} & [f_{\text{S3}}(\boldsymbol{x}), f_{\text{S4}}(\boldsymbol{x}), f_{\text{S7}}(\boldsymbol{x}), f_{\text{S8}}(\boldsymbol{x})], \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_S. \end{array}\right\} \tag{10.3}$$

10.2.2 Numerical results achieved

The attainment fronts obtained by following the experimental design discussed in §10.1 are presented graphically in Figures 10.2–10.4 for each of the three problem instances described above, along with the HSRC. Note that all the values in these results have been scaled according to the percentage improvement in objective function value over that of the HSRC. Furthermore, the reload configuration of the HSRC is presented in Figure 10.1(a) in terms of the ^{235}U mass in each fuel assembly. For each problem instance, an example of a reload configuration that features in the respective attainment fronts is also presented in Figure 10.1. These configurations are discussed later in this section.



(a) *HSRC* (b) *Problem instance* (10.1)

(c) *Problem instance* (10.2) (d) *Problem instance* (10.3)

FIGURE 10.1: *Examples of reload configurations for the SAFARI-1 case study.*

Problem instance (10.1)

In Figure 10.2, the attainment front obtained for problem instance (10.1) is partitioned into a set of objective vectors in which a simultaneous improvement in both objectives is achievable (depicted in green), and a corresponding set in which an improvement in one objective only is achievable at the cost of a deterioration in the other objective (depicted in blue). It may be observed that a number of solutions yielding a simultaneous improvement in the values of objectives S1 and S2 (*i.e.* excess reactivity and power peaking factor, respectively) over that of the HSRC were obtained by the hyperheuristic. It is also observed that the attainment front yields an improvement of up to 27.4% in the value of objective S1, at the cost of a 5.0% deterioration in the value of objective S2.

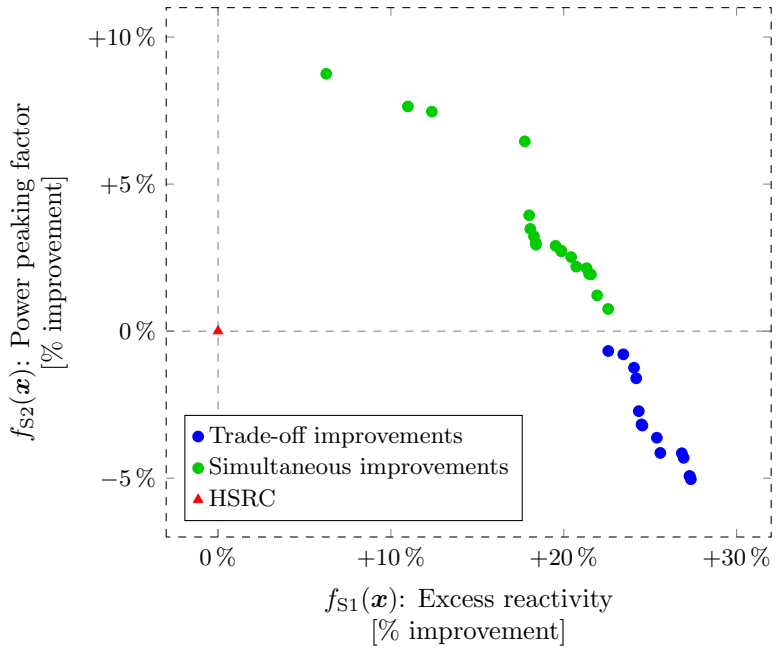


FIGURE 10.2: *Attainment front obtained by five optimisation runs for problem instance* (10.1).

Consider, for example, the reload configuration of the solution which yields a simultaneous improvement of 17.7% and 6.5% in the values of objectives S1 and S2, respectively. This configuration is presented in Figure 10.1(b) in terms of the ^{235}U mass in each fuel assembly. It is observed that the configuration differs largely from that of the HSRC, shown in Figure 10.1(a). The heaviest-massed assemblies are no longer assigned to the core periphery, thus affecting the significant improvement in excess reactivity. Furthermore, the improvement in power peaking factor is affected by the checkerboard-type assignment of heavier-massed and lighter-massed assemblies throughout the core now, especially in the positions of row B in which the peak value was attained for the HSRC.

Problem instance (10.2)

The three-dimensional attainment front obtained for problem instance (10.2) is presented in Figure 10.3. In addition, two-dimensional projections of the front onto the respective planes corresponding to each pair of objectives, are also provided in the figure. As before, the attainment front is partitioned into a set of objective vectors in which a simultaneous improvement

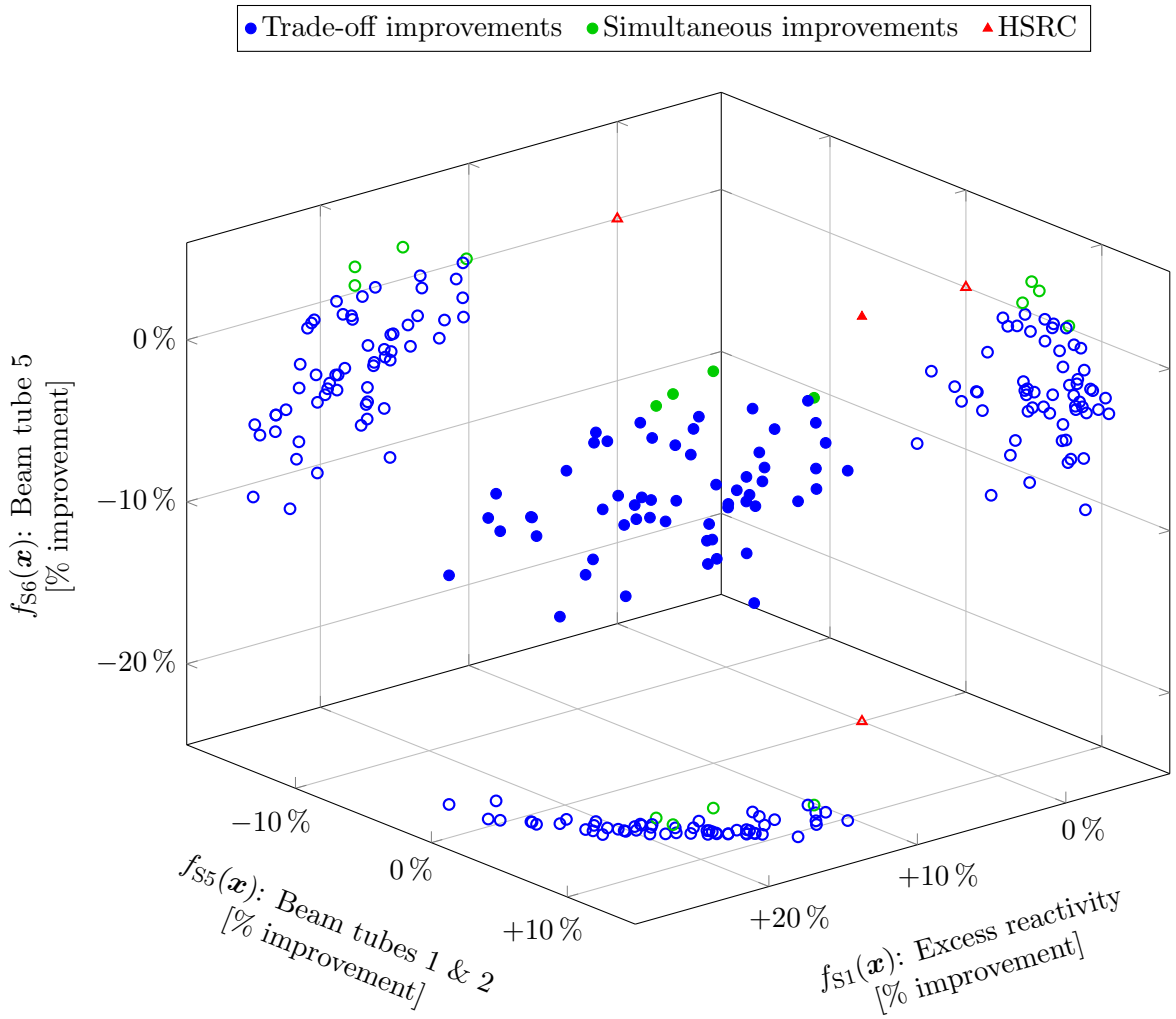


FIGURE 10.3: *Attainment front obtained by five optimisation runs for problem instance* (10.2).

in all three objectives is achievable (depicted in green), and a set in which an improvement in at most two objectives is only achievable at the cost of a deterioration in some other objective (depicted in blue).

In Figure 10.3, it may be observed that the majority of solutions obtained by the hyperheuristic achieve a significant simultaneous improvement in the values of objectives S1 and S5 (*i.e.* excess reactivity and beam tubes 1 & 2, respectively) over that of the HSRC, at the cost of a moderate deterioration in the value of objective S6 (*i.e.* beam tube 5). Four solutions do, however, achieve a simultaneous improvement in all three objectives.

Consider, for example, the solution in which a simultaneous improvement of 14.4%, 4.8% and 2.0% in the values of objectives S1, S5 and S6, respectively, over that of the HSRC is achieved. Its corresponding reload configuration is presented in Figure 10.1(c). It is observed that the heaviest-massed assemblies are assigned to row B in the core, whose positions are close to the beam tubes. This is very similar to the HSRC, shown in Figure 10.1(a); hence only moderate improvements in the values of objectives S5 and S6 are achieved. Very few heavier-massed assemblies are, however, assigned to the unreflected positions in row H, thus leading to lower neutron leakage from the core and the significant improvement in the value of objective S1.

Problem instance (10.3)

In Figure 10.4, the four-dimensional attainment front obtained for problem instance (10.3) is presented in the form of six two-dimensional projections onto the respective planes corresponding to each pair of objectives. Usage of these projections is an alternative to the use of a payoff table (see Chapter 5) for considering the results of a tetra-objective optimisation problem. Unlike in the previous two problem instances, no solutions were obtained for (10.3) in which a simultaneous improvement in all four objectives could be achieved.

It may be observed in Figure 10.4(a) that the HSRC actually forms part of the attainment front, since it is nondominated in terms of objectives S3 and S4 (*i.e.* total ^{99}Mo production and silicon doping, respectively). A slight improvement of 0.3% in the value of objective S3 may be achieved over that of the HSRC, at the significant cost of a 13.8% deterioration in the value of objective S4. Conversely, an improvement of up to 9.2% may be achieved in objective S4, at the cost of a 3.6% deterioration in objective S3. It is observed in Figures 10.4(a)–10.4(c) that objective vectors in which objective S3 is present have spacings that correspond to vertical lines. This peculiar behaviour may be attributed to round-off effects in the OSCAR-4 results, causing those objective function values to be discrete.

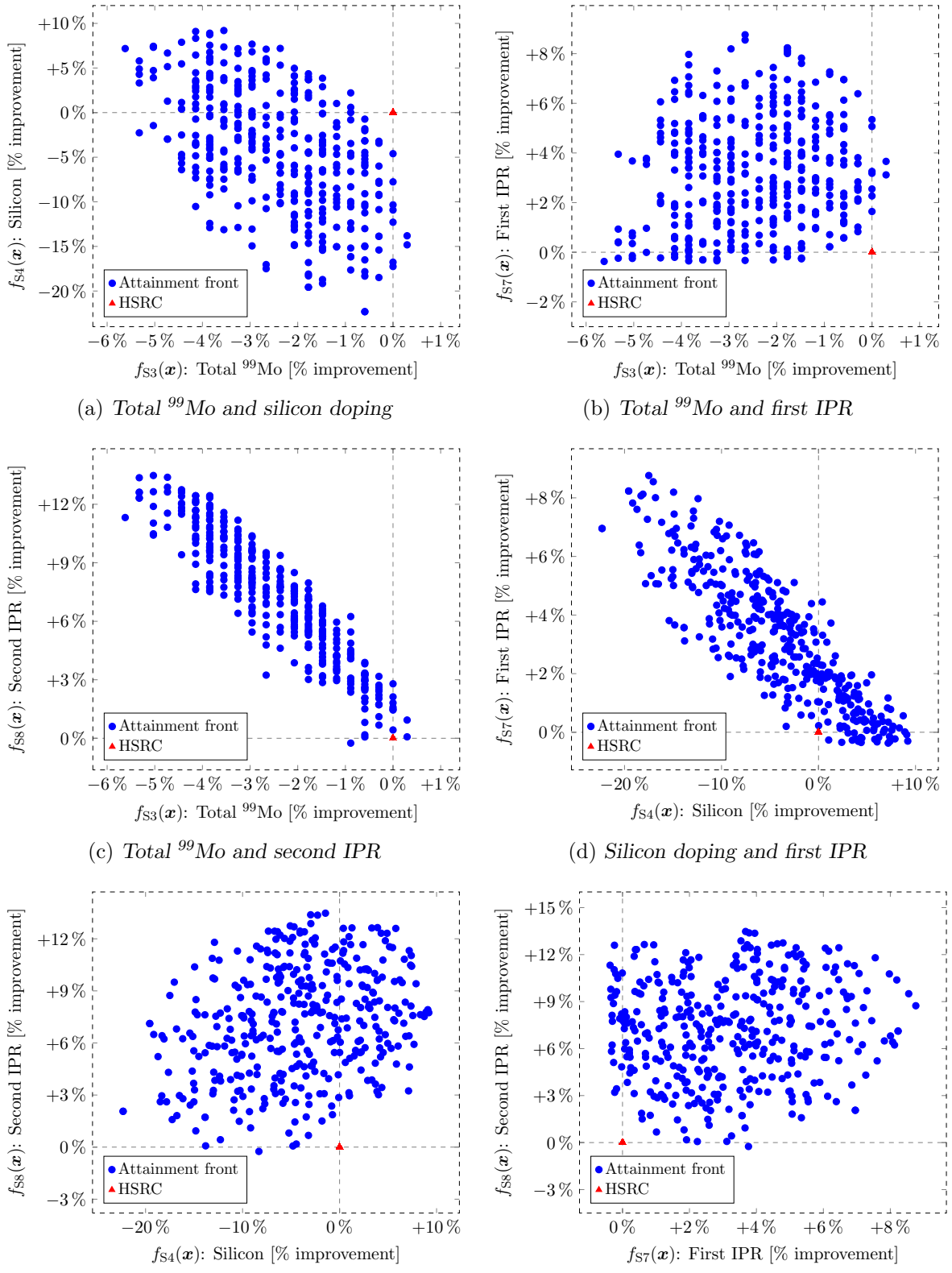
In Figures 10.4(b) and 10.4(c), it may be observed that significant improvements in the performance of objectives S7 and S8 (*i.e.* the first and second IPR facilities, respectively) are readily achievable over that of the HSRC, at the cost of a moderate deterioration in the performance of objective S3. As mentioned in §4.5.3 and §5.4.3, the current SAFARI-1 reload configuration design approach is largely geared towards maximisation of objective S3, while keeping the configuration within its safety limits. As such, the simultaneous optimisation of objective S3 with additional objectives would likely always deteriorate its performance.

Finally, in Figures 10.4(d)–10.4(f), it is observed that the attainment front yields numerous simultaneous improvements in the performance of objectives S4, S7 and S8. In particular, these results indicate that moderate improvements in both IPR facility objectives are almost always possible within the attainment front.

Consider, for example, the solution which yields a 4.4% deterioration in the value of objective S3 and simultaneous improvements of 5.5%, 2.0% and 11.5% in the values of objectives S4, S7 and S8, respectively. Its reload configuration is presented in Figure 10.1(d) in terms of the ^{235}U mass in each fuel assembly. As was the case in Figure 5.7, it is observed that heavier-massed assemblies are assigned to positions near the IPR facilities (D6 and F6) within this configuration, resulting in the improved performance of objectives S7 and S8. In addition, heavier-massed assemblies are also assigned to positions G4 and G6, unlike in the HSRC, shown in Figure 10.1(a). These assignments are likely the cause for the improvement in the value of objective S4.

10.3 The HOR reactor case study

The three case study problem instances considered in this section are based on the first operational cycle for the HOR reactor during the year 2015, which is designated as cycle C1501. Each instance corresponds to a realistic scenario that may be pursued at the HOR reactor. The stop margin constraint specified in §4.5.2 has to be adhered to in each problem instance. Also, the objective function labels in the specification of the instances below correspond to those labels listed in Table 4.2, while the feasible region in the decision space is again denoted by $\mathcal{S}_H$.



(a) *Total ^{99}Mo and silicon doping*

(b) *Total ^{99}Mo and first IPR*

(c) *Total ^{99}Mo and second IPR*

(d) *Silicon doping and first IPR*

(e) *Silicon doping and second IPR*

(f) *First IPR and second IPR*

FIGURE 10.4: *Attainment front projections extracted from five optimisation runs for problem instance* (10.3). *Each subgraph depicts a two-dimensional projection of the front onto the plane corresponding to a pair of objectives.*

10.3.1 The problem instances under consideration

In the first case study problem instance for the HOR reactor, its operational cycle length is to be maximised and the criticality of the reactor during the stop margin requirement (*i.e.* the parameter $k_{\text{eff}}^{\text{sm}}$) is to be minimised. It therefore conforms to the pursuit of objectives H1 and H5, respectively. As before, an objective function may be transformed from a minimisation paradigm to a maximisation paradigm by taking its negative value. The first problem instance is a bi-objective MOP in which the goal is to

$$\left.\begin{aligned} \text{maximise} \quad & [f_{\text{H1}}(\boldsymbol{x}), -f_{\text{H5}}(\boldsymbol{x})], \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}_H. \end{aligned}\right\} \tag{10.4}$$

In the second case study problem instance, the HOR reactor is required to enhance its beam line research, as well as the utilisation of the two in-core irradiation rigs (*i.e.* the Small BeBe and Big BeBe rigs). Of interest, then, is the pursuit of objectives H2, H3 and H4 which, incidentally, combines the objectives considered in (5.18) and (5.19) into a single problem instance. Accordingly, the second problem instance is a tri-objective MOP in which the goal is to

$$\left.\begin{aligned} \text{maximise} \quad & [f_{\text{H2}}(\boldsymbol{x}), f_{\text{H3}}(\boldsymbol{x}), f_{\text{H4}}(\boldsymbol{x})], \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}_H. \end{aligned}\right\} \tag{10.5}$$

Finally, in the third case study problem instance, the cycle length of the HOR reactor is to be maximised in conjunction with the three flux-related objectives pursued in the previous problem instance. Incidentally, this corresponds to the combination of (5.17) and (5.18) into a single problem instance. As such, the third problem instance is a tetra-objective MOP in which the goal is to

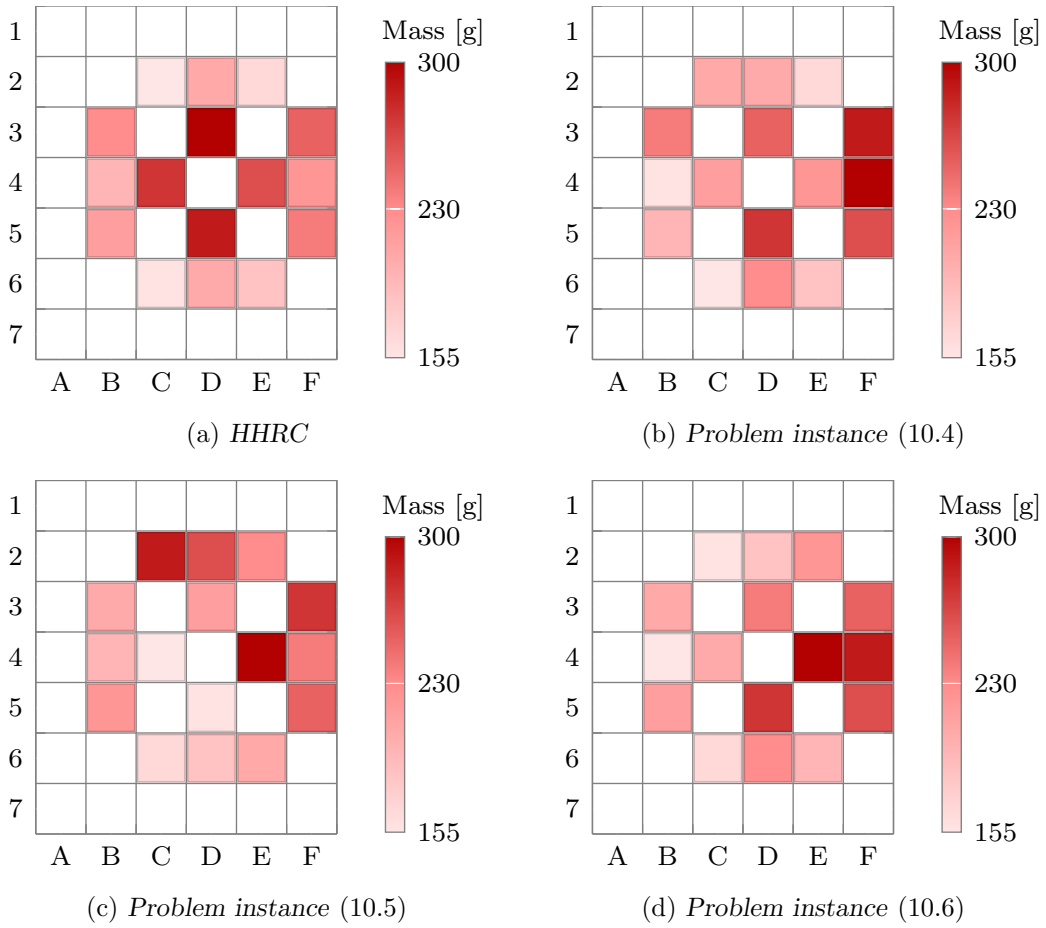
$$\left.\begin{aligned} \text{maximise} \quad & [f_{\text{H1}}(\boldsymbol{x}), f_{\text{H2}}(\boldsymbol{x}), f_{\text{H3}}(\boldsymbol{x}), f_{\text{H4}}(\boldsymbol{x})], \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}_H. \end{aligned}\right\} \tag{10.6}$$

10.3.2 Numerical results achieved

The attainment fronts obtained by following the experimental design discussed in §10.1 are presented in Figures 10.6–10.8 for each of the three problem instances described above, along with the HHRC. Note that all the values in these results have been scaled according to the percentage improvement in objective function value over that of the HHRC. Furthermore, the reload configuration of the HHRC is presented in Figure 10.5(a) in terms of the ^{235}U mass in each fuel assembly. As before, an example of a reload configuration that features in the attainment front of each problem instance is also presented in Figure 10.5. These configurations are discussed later in this section.

Problem instance (10.4)

In Figure 10.6, the attainment front obtained for problem instance (10.4) is again partitioned into a set of objective vectors in which a simultaneous improvement in both objectives are achievable (depicted in green), and a corresponding set in which an improvement in one objective is only achievable at the cost of a deterioration in the other objective (depicted in blue). It may be observed that only a few solutions yield a simultaneous improvement in the values of objectives H1 and H5 (*i.e.* excess reactivity and $k_{\text{eff}}^{\text{sm}}$, respectively) over that of the HHRC, although the percentage value of these improvements are relatively small.



(a) *HHRC* (b) *Problem instance* (10.4)

(c) *Problem instance* (10.5) (d) *Problem instance* (10.6)

FIGURE 10.5: *Examples of reload configurations for the HOR case study.*

As discussed in §4.6.3, the current HOR reload configuration design approach aims to maximise the operational cycle length (*i.e.* objective H1), while satisfying the stop margin constraint. It is therefore not surprising to observe in Figure 10.6 that the value of objective H1 deteriorates significantly (up to 43.7%) as the value of objective H5 improves (up to 2%). The simultaneous optimisation of objective H1 with an additional objective would likely always deteriorate its performance. For this problem instance, it is important to mention that the 2% improvement in the value of objective H5 may be a misleading quantity — this improvement corresponds to approximately 2 000 pcm[1] of reactivity, which is quite significant.

The solution which yields a 1.7% improvement in the value of objective H5 over that of the HHRC, at the cost of a 19.9% deterioration in the value of objective H1 is considered as an example. Its corresponding reload configuration is presented in Figure 10.5(b) in terms of the ^{235}U mass in each fuel assembly. It is observed that heaviest-massed assemblies are assigned to the unreflected positions in column F of the core, unlike the HHRC, which is shown in Figure 10.5(a), whose heaviest-massed assemblies are assigned to the central core positions.

[1]Derived from the Italian "*per cento mille*", meaning per hundred thousands, *pcm* is a unit of reactivity defined as $\%\Delta k/k$, where k is the neutron multiplication factor [152].

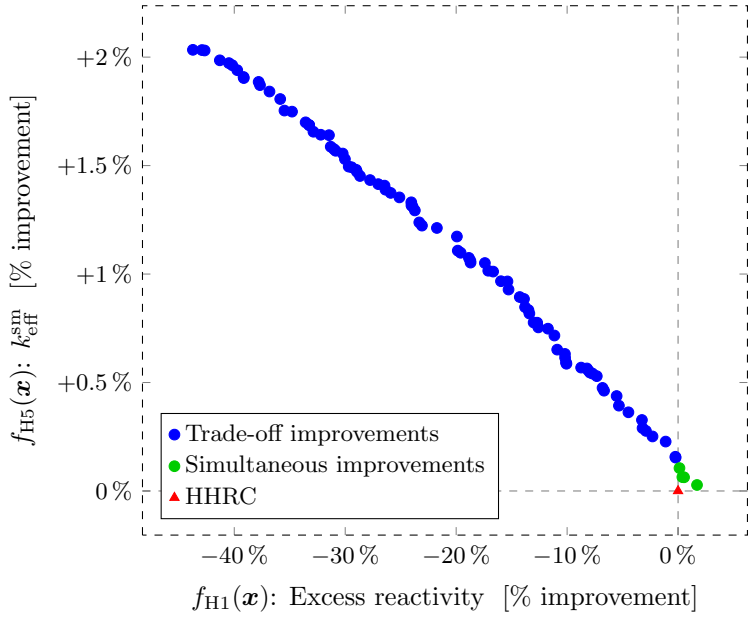


FIGURE 10.6: *Attainment front obtained by five optimisation runs for problem instance* (10.4).

Problem instance (10.5)

The three-dimensional attainment front obtained for problem instance (10.5) is presented in Figure 10.7. In addition, two-dimensional projections of the front onto the respective planes corresponding to each pair of objectives, are also illustrated. As before, the attainment front is partitioned into a set of objective vectors in which a simultaneous improvement in all three objectives is achievable (depicted in green), and a set in which an improvement in at most two objectives is only achievable at the cost of a deterioration in some other objective (depicted in blue).

In Figure 10.7, it may be observed that numerous solutions obtained by the hyperheuristic are able to achieve a simultaneous improvement in the values of all three objectives over that of the HHRC. Furthermore, an improvement in the value of objective H3 (*i.e.* Small BeBe rig) is always achievable within the attainment front. It is also observed that a significant spread of trade-off solutions were obtained by the hyperheuristic in respect of objectives H2 and H4 (*i.e.* beam tube and Big BeBe rig, respectively). The range of objective H2 lies between −10.8% and 10.7%, whereas objective H4 ranges between −8.2% and 13.5%.

Consider, for example, the reload configuration of the solution in which a simultaneous improvement of 4.3%, 3.2% and 3.7% is achieved in the values of objectives H2, H3 and H4, respectively, over that of the HHRC. This configuration is presented in Figure 10.5(c) in terms of the ^{235}U mass in each fuel assembly. It is observed that the heavier-massed assemblies are all assigned to positions near the beam tube and two irradiation rigs (located in positions 2B and 4D), hence the improvements in each of the three corresponding objectives.

Problem instance (10.6)

In Figure 10.8, the four-dimensional attainment front obtained for problem instance (10.6) is presented in the form of six two-dimensional projections onto the respective planes corresponding

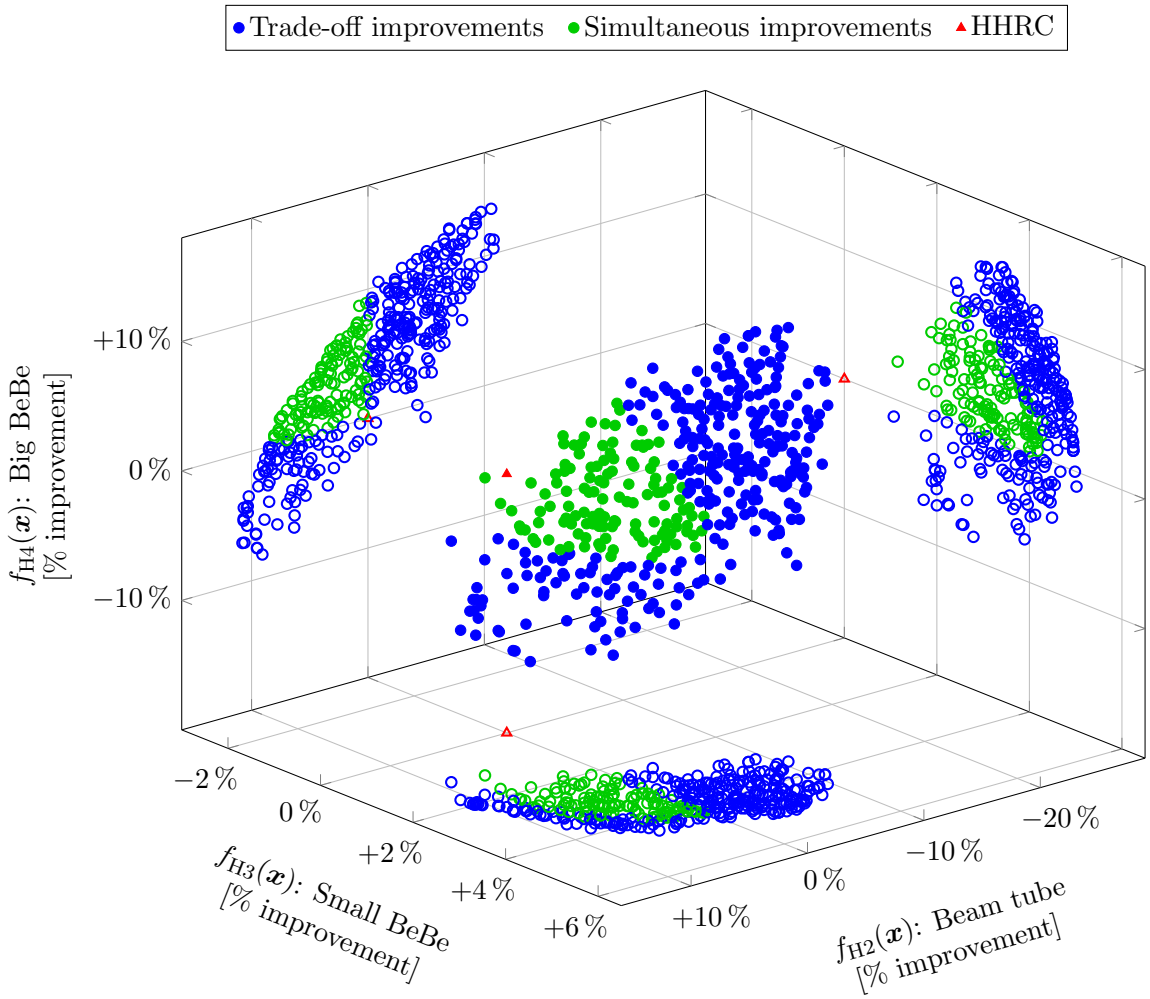
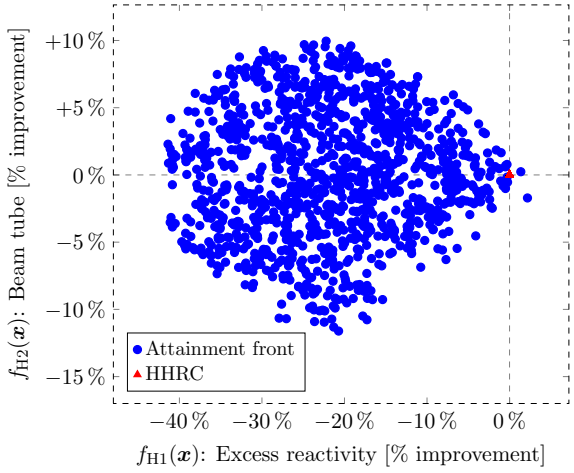


FIGURE 10.7: *Attainment front obtained by five optimisation runs for problem instance* (10.5).

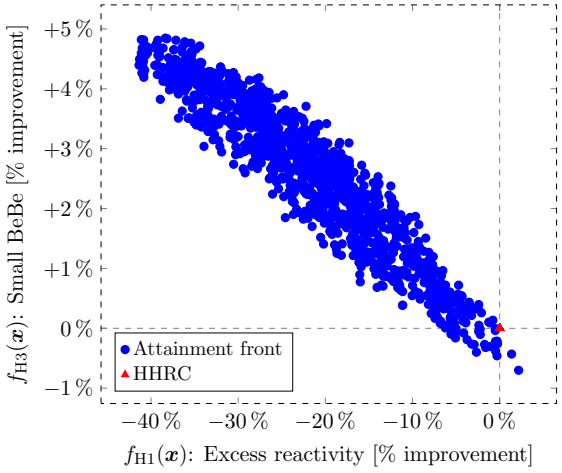
to each pair of objectives. Unlike in the previous two problem instances, no solutions were obtained for (10.6) in which a simultaneous improvement in all four objectives could be achieved.

As was the case in problem instance (10.4), it may be observed in Figures 10.8(a)–10.8(c) that improvements in the values of objectives H2, H3 and H4 (*i.e.* beam tube, Small BeBe and Big BeBe rigs, respectively) over that of the HHRC are achievable at the cost of a significant deterioration in the value of objective H1 (*i.e.* excess reactivity). It is also observed in Figures 10.8(d)–10.8(f) that the performance achievable in objectives H2, H3 and H4 corresponds closely to what was found in Figure 10.7. The reason for this behaviour is simple — these three objectives were also pursued in problem instance (10.5).

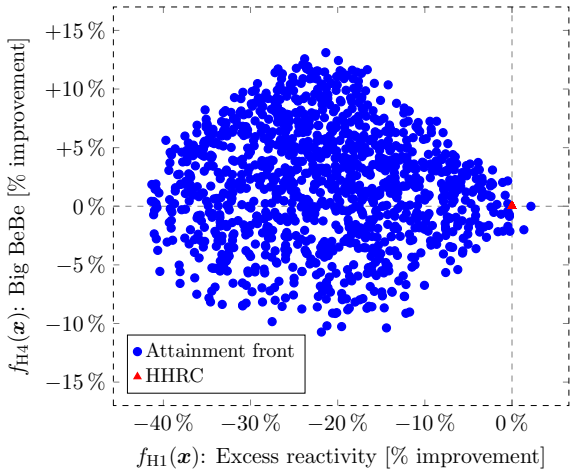
The solution in which improvements of 8.3% and 1.2% are achieved in the values of objectives H2 and H3, respectively, over that of the HHRC, at the cost of a 14.9% and 7.6% deterioration in the values of objectives H1 and H4, respectively, is considered as an example. Its corresponding reload configuration is presented in Figure 10.5(d). It is observed that heavier-massed assemblies are assigned to positions near the beam tube and the Small BeBe rig (position 4D), in pursuit of objectives H2 and H3, respectively. Unlike the case in Figure 10.5(c), only lighter-massed assemblies surround the Big BeBe rig (position 2B), hence the deterioration in objective H4. Off course, all these assignments occur to the detriment of objective H1.



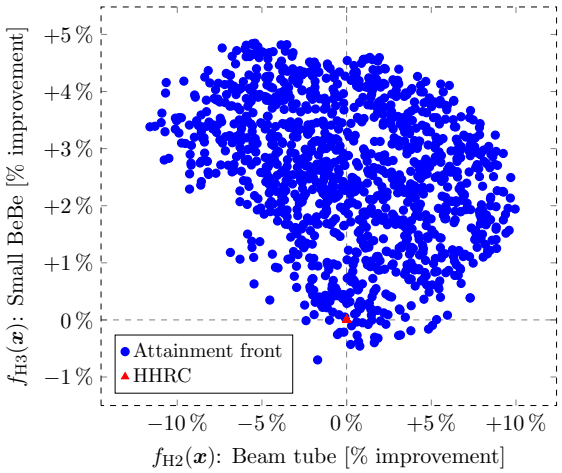
(a) *Excess reactivity and beam tube*



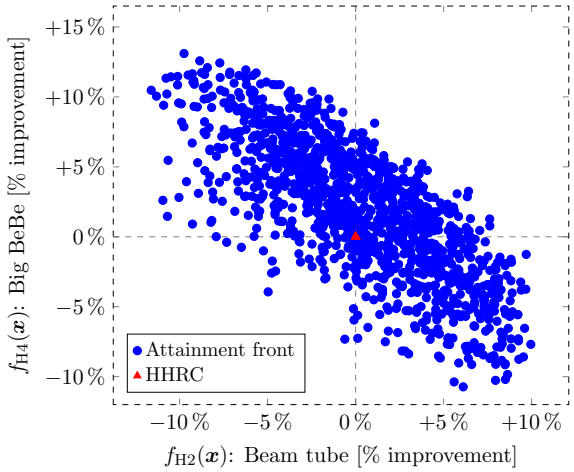
(b) *Excess reactivity and Small BeBe rig*



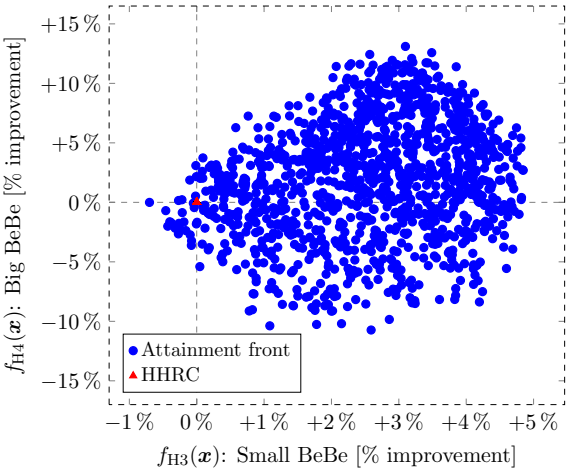
(c) *Excess reactivity and Big BeBe rig*



(d) *Beam tube and Small BeBe rig*



(e) *Beam tube and Big BeBe rig*



(f) *Small BeBe and Big BeBe rig*

FIGURE 10.8: *Attainment front projections extracted from five optimisation runs for problem instance* (10.6). *Each subgraph depicts a two-dimensional projection of the front onto the plane corresponding to a pair of objectives.*

10.4 Conclusion

Based on the results presented above, it may be concluded that the AMALGAM$_{\text{npm}}$ method, as implemented within a new MICFMO decision support feature in the OSCAR-4 system, does, in fact, find solutions to the constrained MICFMO problem that are of adequate quality for practical usage. Numerous simultaneous improvements in the values of several objectives over that of the HSRC and HHRC were obtained. Furthermore, the good spread of solutions obtained in the attainment fronts affords decision makers at the SAFARI-1 and HOR reactors with improved flexibility in their choice of reload configuration design. Accordingly, the method may be used as an effective decision support tool for designing reload configurations.

10.5 Chapter summary

In this chapter, a new MICFMO decision support feature in the OSCAR-4 system was used to solve (approximately) several realistic case study problem instances for the SAFARI-1 and HOR research reactors in §10.2 and §10.3, respectively. The numerical results obtained indicated that the AMALGAM$_{\text{npm}}$ method, on which this OSCAR-4 feature is based, is able to find high-quality solutions to instances of the constrained MICFMO problem that are suitable for practical usage.

CHAPTER 11

A decision support system framework for MICFM

Contents

A conceptual framework for an optimisation-based personal decision support system, dedicated to multiobjective in-core fuel management, is proposed in this chapter. Each constituent component of the system is discussed in some detail, after which a listing of suggestions for populating some of these components is presented.

11.1 Introduction

As already reported in this dissertation, the author implemented the scalarisation-based methodology proposed in Chapter 5 and the multiobjective hyperheuristic investigated in Chapter 9 in the OSCAR-4 code system, thereby enabling the system to render decision support for *multiobjective in-core fuel management* (MICFM). A reactor simulation code has therefore been extended with an MICFM decision support feature. It may, however, be argued that a fully-fledged computerised *decision support system* (DSS) for MICFM, in which a reactor simulation code forms but one part of the greater system, should be aimed for instead.

Recall from Chapter 3 that ICFM has been studied for several decades, and the majority of that research involved power reactors within the context of single-objective optimisation. During that time, a number of computerised tools have been developed, both in academia and industry, that may be regarded as DSSs [1, 147, 223]. An example of a prominent knowledge-based DSS (also

known as an expert system) for ICFM is the system emanating from the FUELCON project (which was discussed in §3.3.1). Metaheuristics are also widely employed in optimisation-based DSSs for solving the ICFMO problem. Examples of systems in which the SA algorithm is employed are the ROSA software package [223], the FORMOSA suite of codes [129, 142], and the XIMAGE/SIMAN graphical fuel management and loading pattern optimisation suite [235]. Similarly, the CIGARO system [35], the GARCO package [1] and the INSIGHT software tool [239] are examples of DSSs in which a GA is adopted as metaheuristic solution technique.

Apart from the FORMOSA suite, the aforementioned tools provide decision support only in the context of single-objective ICFM. In the case of the FORMOSA suite, the MICFMO problem is solved in terms of identifying a nondominated set of reload configurations, which is also the case in this dissertation. A decision maker should then ultimately select only one of these reload configurations according to his subjective preferences. It is, however, not necessarily obvious how to select this preferred reload configuration. Additional support may therefore be required to aid a decision maker in his choice. In *multiple criteria decision analysis* (MCDA), a finite set of predetermined alternatives is available and the aim is to identify a preferred alternative by incorporating the specific preferences of a decision maker during the identification process. It is therefore crucial that some level of MCDA support be rendered during MICFM so that a final reload configuration may be settled upon by a decision maker. To the best knowledge of the author, however, no research is available in the literature involving the development or application of MCDA techniques to MICFM.

Finally, it is noted that the aforementioned tools have all been designed for application to power reactors. MICFM decision support should, however, also include capabilities for research reactors since decision makers at these reactors encounter many of the MICFM challenges that those at power reactors are also faced with (and sometimes even more).

A generic optimisation-based DSS framework for MICFM is therefore proposed in this chapter in an attempt to address the shortcomings within existing DSS tools for ICFM, as identified above. This conceptual framework is intended to serve as a high-level formalisation of a computerised DSS tool which may assist decision makers at nuclear reactors in single-cycle MICFM reload configuration design. In principle, the framework is applicable in the context of any conventional light water power or research reactor.

11.2 Background

According to Shim *et al.* [193], a DSS is a "*computer technology solution that can be used to support complex decision making and problem solving.*" It should be capable of identifying candidate decision alternatives, as well as determining their consequences, before providing recommendations once those consequences have been evaluated [162]. Any DSS exhibiting these capabilities supports the decision making model popularised by Herbert Simon in which three phases, namely *intelligence*, *design* and *choice*, characterise the model [162, 193, 200].

In the classical design for a DSS proposed by Sprague [200], three major components constitute the system, namely a *data subsystem*, a *model subsystem*, and a *user interface*. A diagrammatic representation of the components of this DSS design is given in Figure 11.1. In the data subsystem, sophisticated database management techniques are utilised for accessing external and internal data. Different modelling functions may be found in the model subsystem that are employed for constructing an appropriate decision making model. Finally, the user interface enables a user to interact with the DSS, allowing him/her to analyse and choose between different decision alternatives.

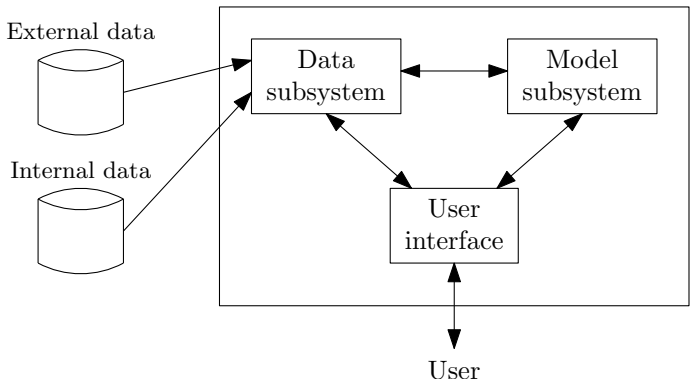


FIGURE 11.1: *The classical design of a DSS, as proposed by Sprague [200].*

An appreciable amount of research has been conducted over the years in respect of each component of this classical DSS design [193]. Today, there are numerous different types of DSSs available in the literature that have evolved from the early designs. A history of these developments, as well as a listing of several different DSS types, may be found in [7, 193].

According to Arnott and Pervan [7], a *personal DSS* is a small-scale system in which the aim is to support one decision task, and it is typically developed for a single user (or a small number of independent users). Such a DSS type is therefore appropriate in the context of MICFM, because the single decision task is that of choosing which reload configuration to load into a reactor core, while the user of the system is typically a nuclear reactor operator or engineer.

As evidenced by the discussion in Chapter 3, the intricacies of MICFM may be captured, to a large extent, within an optimisation model. Accordingly, by Alter's taxonomy of DSSs in [7], it then follows that a *model-orientated* personal DSS for MICFM, based on an optimisation model, would be a suitable type of DSS to pursue. The system may also be referred to as an *intelligent DSS* if so-called artificial intelligence techniques (which includes metaheuristics) are adopted to solve the associated optimisation model [7].

11.3 The proposed optimisation-based DSS framework

In this section, a conceptual framework is presented for a generic optimisation-based personal DSS for MICFM. The framework is novel in the sense that, to the best knowledge of the author, no other framework proposals of similar scope of application exist in the ICFM literature. The DSS in this framework comprises six major components, namely a *database management system* (DBMS), a *problem generator* (PG), an *optimisation engine* (OE), a *function evaluator* (FE), an optional *auxiliary optimisation system* (AOS), and a *human machine interface* (HMI). A diagrammatic representation of the proposed DSS is presented in Figure 11.2. Dashed lines in the diagram indicate that the component or connection is optional.

The HMI component facilitates all interactions between a user and the DSS, which makes it one of the most critical components in the system. An internal collection of general and/or problem-specific information is maintained in the DBMS component, while it also retrieves any required input data from external sources. In the PG component, a specific MICFMO problem instance is generated, along with its associated optimisation model, by utilising input and modelling information sourced from the DBMS and/or through the HMI. Next, the OE component employs an MOO solver to identify a nondominated set of reload configurations (possibly Pareto optimal) by solving the relevant optimisation model. Thereafter, MCDA support (also

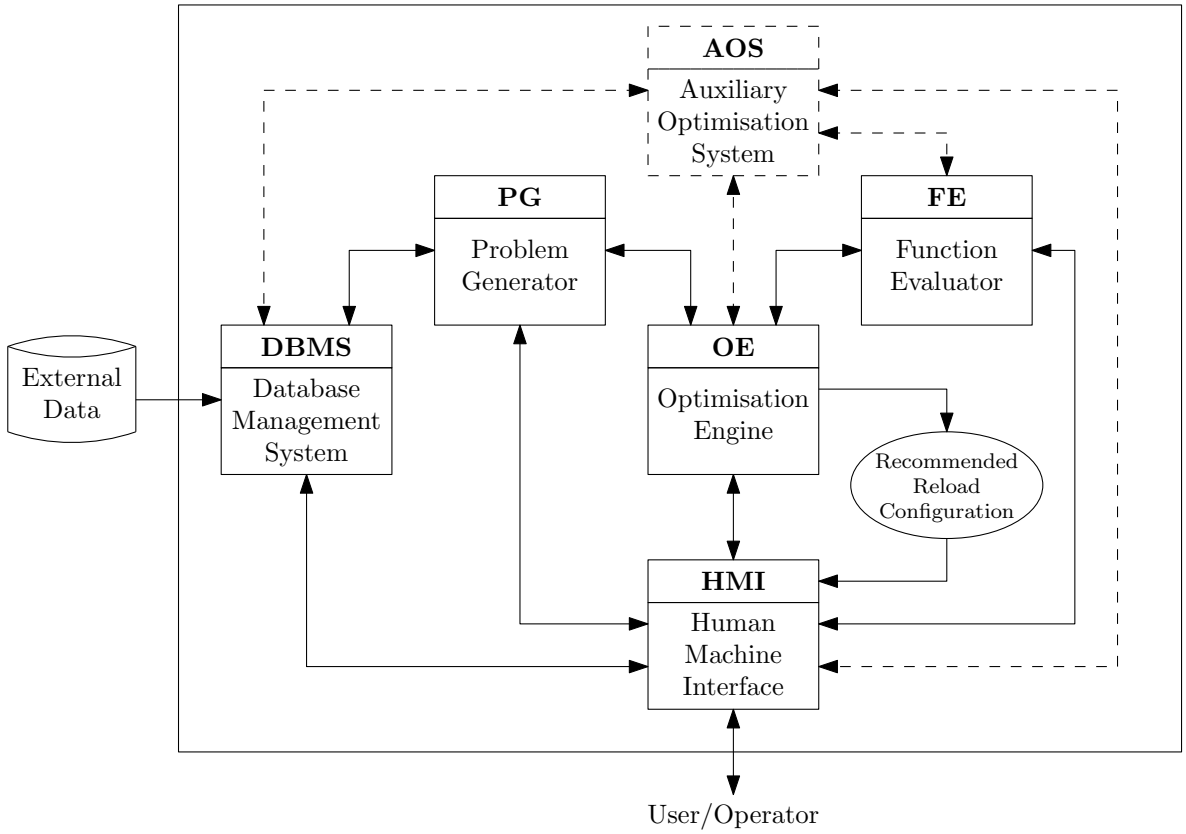


FIGURE 11.2: *A high-level diagrammatic representation of the proposed optimisation-based personal DSS for MICFM.*

within the OE component) is rendered to the user in order to facilitate the selection of one fuel reload configuration only. Any candidate reload configuration is evaluated by employing the FE component. Finally, the optional AOS component contains any supporting features that may enhance the efficiency or effectiveness of optimisation in the DSS.

It is intended that the components in the proposed DSS framework be modular in their design. The robustness and general applicability of the DSS should be improved by this modularity since it allows for components to be extended, replaced or even removed entirely with relative ease. This may also be true for sub-components in the system.

In the sections that follow, an overview of each DSS component is presented diagrammatically, and this is followed by a detailed discussion on its function in each case. As before, optional sub-components are indicated by dashed lines in all the diagrams.

11.3.1 The database management system (DBMS)

A diagrammatic representation of the DBMS component is presented in Figure 11.3. The component contains an internal collection of general and/or problem-specific information and knowledge, as well as data management routines for access, modification and retrieval purposes. Any external data required to conduct MICFM, such as the number of fresh fuel assemblies to consider, what the planned operational cycle length should be, and so forth, are also retrieved by the DBMS. The primary source of external data is the OCFM decision process, which was briefly described in Chapter 4.

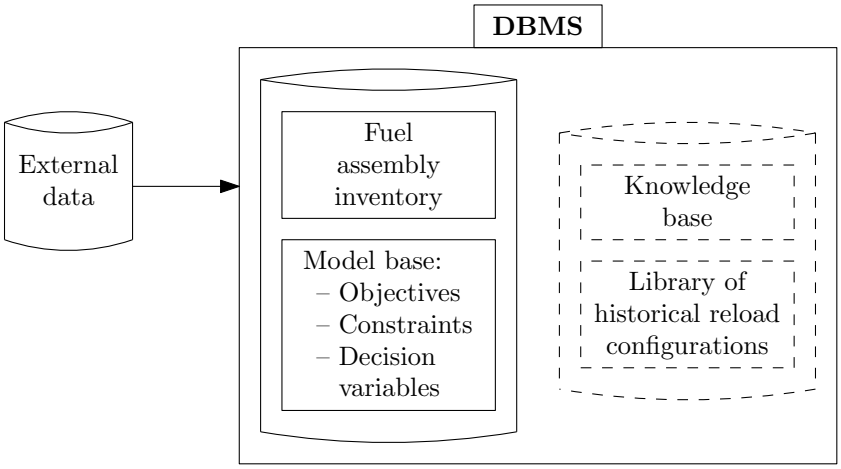


FIGURE 11.3: *The database management system (DBMS) component of the DSS in Figure 11.2.*

The fuel assembly inventory is generally the most important part of the DBMS. In this inventory, the physical properties of every fuel assembly that has previously formed part, or will form part of future, ICFM decisions for a reactor, are tracked (assuming the assembly is not depleted). As explained in Chapter 2, the isotopic composition of any fuel assembly loaded into a reactor core has to be known because of the particular contribution it makes to the neutron flux.

An optimisation model base is also contained in the DBMS and, as mentioned in §11.2, it should consist of various modelling functions that may be employed to generate an MICFMO problem instance (and its associated optimisation model). In accordance with the existing DSSs discussed in §11.1, known objectives, constraints and decision variables within the context of MICFMO should be available for the DSS user to choose from. If, however, some model elements do not form part of the current model base, they may be added by a user through the HMI.

Since a vast amount of experience has been accumulated by field experts charged with designing reload configurations over the years, their knowledge may aid in solving an MICFMO problem instance (*e.g.* as is the case with an expert system). In order to include such invaluable experience in the DSS, it may be captured within a knowledge base as part of the DBMS and subsequently revised when necessary by using the HMI. Typically, a knowledge base takes the form of a set of heuristic rules [147]. Historical reload configurations, known to have performed well for a given problem instance, may similarly be included in a library of configurations kept within the DBMS. Such configurations may be employed, for example, as starting points during the search for new configurations in the OE component.

11.3.2 The problem generator (PG)

The PG component is presented in Figure 11.4 and its function is to generate a specific MICFMO problem instance for consideration. As such, a problem instance sub-component comprises the objectives, constraints and decision variables selected by the user, obtained either from the model base in the DBMS component, or directly from new user input through the HMI. Data required for the specific problem instance in respect of the fuel assemblies are also to be found in the PG component. Only a subset of the fuel assemblies tracked in the DBMS is considered for data extraction. It should correspond to those assemblies selected for potential reloading in the reactor core (*e.g.* determined as part of the OCFM decision process), for the operational cycle under consideration. The relevant data necessary to perform optimisation may, for example, consist of the axial burnup distribution of each fuel assembly, or its total ^{235}U mass, *etc.*

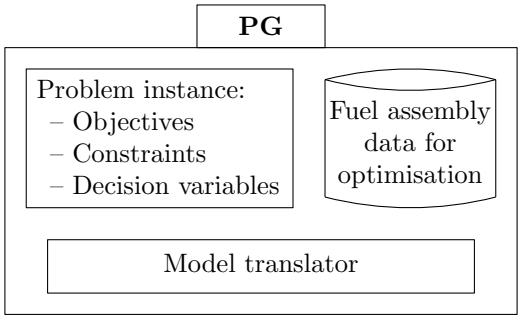


FIGURE 11.4: *The problem generator (PG) component of the DSS in Figure 11.2.*

Finally, in order to construct an optimisation model associated with the particular MICFMO problem instance, a model translator sub-component is employed. This sub-component is responsible for representing the optimisation model in a suitable format for use in the OE component of the DSS, *i.e.* to enable the application of an MOO solver. Examples thereof include the solution encoding of a reload configuration (binary, integer, permutation, *etc.*), closed form expressions for objective and constraint functions in terms of the decision variables, or place-holders for function calls to a reactor core simulator, and so forth.

11.3.3 The optimisation engine (OE)

At the heart of the proposed optimisation-based personal DSS lies its OE component, which is presented diagrammatically in Figure 11.5. An MOO solver sub-component renders the primary level of MICFM decision support and it is utilised for solving the optimisation model associated with the specific problem instance received from the PG component. The MOO solver may employ an exact or approximate solution technique, depending on the available computational budget and the optimisation model adopted. A suitable constraint handling technique may also be required. Given the implications of the NFL theorem for optimisation, as discussed in §9.1, multiple approximate solution techniques should be available for the user to choose from.

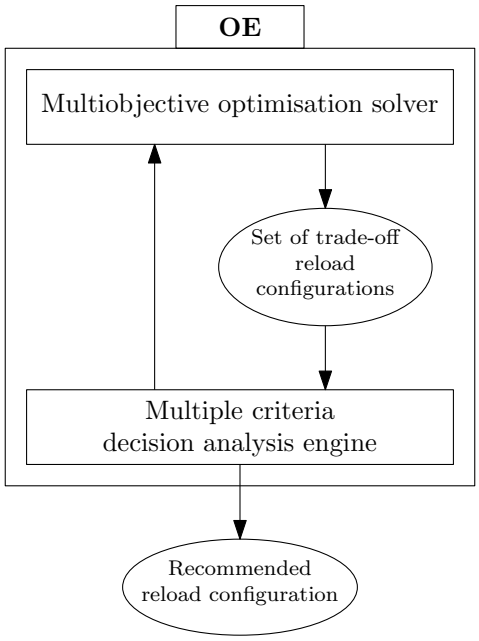


FIGURE 11.5: *The optimisation engine (OE) component of the DSS in Figure 11.2.*

Once the MOO solver concludes its optimisation run(s), a nondominated set of reload configurations (possibly Pareto optimal) is obtained and forwarded to an MCDA engine sub-component. At this point, the MCDA engine renders the secondary level of MICFM decision support. It is utilised to refine the nondominated set by incorporating subjective decision maker preferences, possibly in an interactive manner, until a final reload configuration may be recommended. It is this recommendation which is ultimately the result returned by the DSS. Given the subjective nature of human preference, a variety of MCDA techniques should ideally be available for the user to choose from. Some form of data fusion process may even be adopted to aggregate the outcomes of different techniques.

11.3.4 The function evaluator (FE)

An overview of the FE component is presented in Figure 11.6. This component is employed to evaluate the objective and constraint functions of any reload configuration obtained during the execution of processes in the OE component. A high-fidelity reactor core simulator sub-component is central to the FE, because it is required, at the very least, to validate the function values associated with a candidate reload configuration. The purpose of this validation is two-fold. First, it ensures that a candidate configuration meets all the safety-related regulatory requirements necessary to be considered for actual loading into a reactor core. Secondly, it yields highly accurate objective function values, allowing the user to consider the performance of different reload configurations in the most informed manner. The reactor core simulator should therefore be applied to the nondominated set of reload configurations forwarded from the MOO solver to the MCDA engine. By doing so, any invalid reload configuration may be eliminated from further contention before application of the MCDA engine.

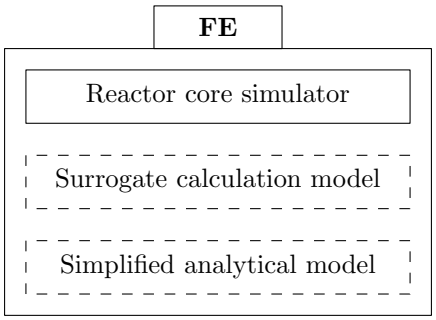


FIGURE 11.6: *The function evaluator (FE) component of the DSS in Figure 11.2.*

During the solution of the MICFMO model in the OE component, objective and constraint function evaluations may be performed using the same reactor core simulator described above, although such usage may be computationally too expensive. In order to improve the efficiency of the DSS, a computationally cheaper surrogate calculation model sub-component may be employed within the FE for these function evaluations (*e.g.* ANNs). Depending on which optimisation model is adopted, function evaluations may also be performed according to a simplified analytical model sub-component.

11.3.5 The auxiliary optimisation system (AOS)

The optional AOS component of the DSS is presented in Figure 11.7. It contains a number of sub-components that may enhance the efficiency or effectiveness of optimisation in the DSS. In

the solver enhancement features sub-component, a surrogate calculation model screening feature may be adopted within a hybrid function evaluation approach. According to such an approach, the surrogate calculation model from the FE component is employed as a screening tool to pre-evaluate reload configurations quickly. Thereafter, only those configurations that passed the screening process are evaluated by the reactor core simulator, thus saving valuable computation time. The work of Yamamoto [236] motivated the inclusion of this screening feature in the AOS. Another enhancement feature, popularly employed in the ICFMO literature [1, 131, 136, 204], may be to utilise the knowledge base within the DBMS during application of the MOO solver. By doing so, certain regions in the decision space of the MICFMO problem are precluded during optimisation so that the search may be focussed in regions known to be promising. Furthermore, as already alluded to in §11.3.1, historical reload configurations kept in the DBMS may be utilised for seeding an MOA with good solutions so as to enhance its search.

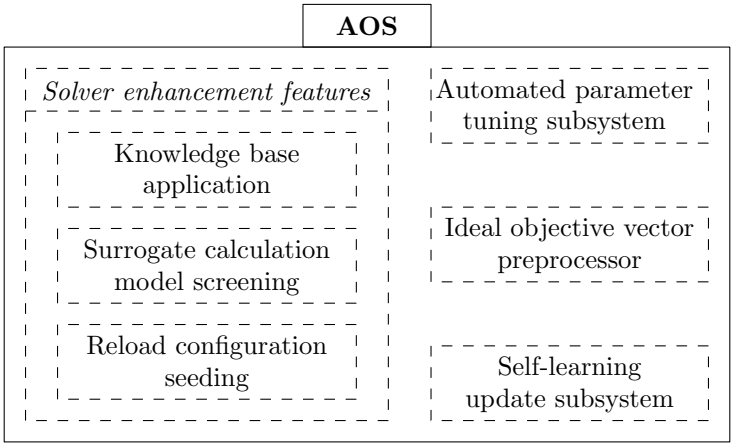


FIGURE 11.7: *The auxiliary optimisation system (AOS) component of the DSS in Figure 11.2.*

Given that many candidate reload configurations are evaluated during optimisation within the OE component, this feedback information may be exploited in a self-learning update subsystem. The idea behind such a subsystem is that it should be capable of revising the surrogate calculation model and/or the knowledge base kept within the DBMS in an automated fashion. Future calculations may then benefit from previous calculations using the enhanced features. Inclusion of this subsystem in the DSS was inspired by the work in [148].

In the majority of MOAs available in the literature, specific tuning parameter values have to be selected by the user prior to optimisation (*e.g.* as experienced with the metaheuristics described in Chapter 7). Since the quality of the optimisation results may depend on these values, an automated parameter tuning subsystem may be included in the AOS component so as to remove the burden of manual parameter tuning from the user. Although tuning methods have been proposed in the literature within the context of single-objective optimisation [47, 141], it is more difficult to find similar approaches involving MOO [228].

Finally, in many MCDA techniques, as well as certain scalarisation approaches for MOO, knowledge of the ideal or utopian objective vector (or an approximation thereof) is required [139]. Usually, these vectors are not known for practical MOP instances, as is the case for MICFMO. Accordingly, inclusion of an ideal objective vector preprocessor sub-component in the AOS may prove useful to the MCDA engine. A single-objective optimisation solver within the preprocessor may then be employed to calculate the ideal objective vector (or an approximation therof) for a given MICFMO problem instance.

11.3.6 The human machine interface (HMI)

Recall that all interactions between a user and the DSS are facilitated by the HMI component. The design and usage of the HMI should be easy to understand by both technical and non-technical users (*e.g.* analysts and managers) and ideally take the form of a fully interactive graphical computer interface. Any user of the DSS should be convinced that his preferences have been suitably captured by the recommended reload configuration. It is therefore crucial that the results obtained by the MOO solver (so as to gain an understanding of what performances are possible) and the MCDA engine be presented to the user in an appropriate and user-friendly manner through the HMI. Furthermore, once the DSS has recommended a final reload configuration, all the pertinent information in respect of that configuration should be displayed in the HMI for detailed scrutiny by a user. Finally, two contrasting design objectives have to be balanced in the HMI. Since humans have limited processing capacities [162], an information overload should be avoided. On the other hand, excessive automation is also detrimental because it may weaken the understanding and interpretation of results by the user.

11.4 Suggestions for populating components of the DSS

Recall from §1.4 that populating the components of the proposed DSS, as well as its subsequent implementation on a personal computer, does not fall within the scope of this dissertation. A number of techniques and approaches may, however, be suggested for populating some of these components. These suggestions may be considered as a potential starting point in future work to demonstrate the functionality of the system.

Given the outcomes of the extensive investigation performed in this dissertation into the ability of several different MOAs to conduct constrained MICFMO, it is suggested that the AMALGAM method be adopted in the MOO solver sub-component of the OE. Similarly, the augmented Chebyshev scalarising function, defined in (5.11), may be considered for application in the MCDA engine for identifying a preferred reload configuration. It has also been demonstrated that MFNNs are well-suited to take on the role of a surrogate calculation model in the FE component of the DSS. The OSCAR-4 system may, of course, be adopted as the reactor core simulator, while its related service codes may be employed for the purpose of tracking the fuel assembly inventory in the DBMS.

Suggestions not based on the research conducted in this disseration are as follows. The screening approach proposed by Yamamoto [236] may be adopted as part of the solver enhancement features in the AOS component. Similarly, the extensive knowledge base developed within the FUELCON expert system may be utilised in the DBMS and applied within the AOS as an enhancement feature. Finally, the model developed by Quist *et al.* [168] may be adopted as a simplified analytical model in the FE.

11.5 Chapter summary

In this chapter, a conceptual framework was proposed for an optimisation-based personal DSS in the context of MICFM. A motivation for the proposal was provided in §11.1. Thereafter, some basic background information concerning DSSs was provided in §11.2 in order to familiarise the reader with the relevant concepts.

In §11.3, the proposed DSS framework was presented. This framework consists of six major components, namely a database management system, a problem generator, an optimisation engine,

a function evaluator, an optional auxiliary optimisation system, and a human machine interface. Each of these components were discussed in some detail in §11.3.1–§11.3.6, respectively. This framework may serve as the basis for developing a computerised tool dedicated to aid nuclear reactor operators or engineers in designing suitable reload configurations.

Finally, several suggestions for populating some of the components of the DSS were presented in §11.4. These suggestions may be implemented in future work to demonstrate the functioning of the system.

CHAPTER 12

Conclusion

Contents

The dissertation closes with a summary of the work contained therein, an appraisal of the contributions of the dissertation, and suggestions for related future work.

12.1 Dissertation summary

The dissertation opened in Chapter 1 with a brief history of nuclear fission and the development of nuclear reactors. An informal description of the ICFMO problem was presented in the second section of the chapter. It also included a brief discussion of the problem within the context of the SAFARI-1 research reactor in South Africa. This led to the introduction of the MICFMO problem, which was the topic considered in this dissertation. In the next section, two priorities for the dissertation were identified, namely: (1) that shortcomings present in the existing optimisation methodology within the reactor core calculation code system, OSCAR-4, should be addressed; and, (2) that the suitability of alternative multiobjective computational methods should be investigated in the context of the MICFMO problem. The chapter closed with an outline of the scope and objectives to be pursued in the dissertation, as well as a brief description of the organisation of the material contained therein.

Fundamental concepts and terminology found in the literature on nuclear reactor analysis and theory were introduced in Chapter 2 (in fulfilment of Dissertation Objective I of §1.4). This included a description of the most important nuclear reactions, and a more detailed discussion on nuclear fission and its chain reaction. A brief overview of the basic components of a nuclear reactor core was also presented. Furthermore, the process of neutron transport (along with its diffusion approximation) was discussed. The chapter closed with a description of the primary neutronic aspects of interest within nuclear reactor analysis, as well as a discussion on the necessity of reactor core calculation code systems.

A comprehensive literature survey on the ICFMO problem was presented in Chapter 3. It contained an overview of the most popular objective functions and constraints adopted in model formulations for the ICFMO problem (in partial fulfilment of Dissertation Objective II of §1.4), as well as a discussion on typical solution techniques that have previously been employed in the literature to solve the problem (also in partial fulfilment of Dissertation Objective II). The

chapter closed with a brief discussion on different approaches considered in the literature for reducing the computational burden associated with ICFMO (in final fulfilment of Dissertation Objective II of §1.4).

In Chapter 4, the topic of ICFMO was placed in its proper context within the broader scope of nuclear fuel management. Several necessary problem assumptions were presented before the optimisation model adopted in this dissertation for the MICFMO problem was formulated (in fulfilment of Dissertation Objective III of §1.4). Thereafter, the reactor core calculation code system utilised in this dissertation, namely the OSCAR-4 system, was briefly described. The chapter then closed with descriptions of two nuclear research reactors considered as case studies in this dissertation, namely the SAFARI-1 and HOR reactors. This included the specification of typical MICFMO objectives and constraints associated with each reactor, as well as descriptions of the current reload configuration design approaches employed at the reactors.

Chapter 5 opened with a detailed discussion on the notion of Pareto optimality and other related concepts. In the next section, the widely-employed weighting method (utilised in the existing OSCAR-4 optimisation feature) was described and its shortcomings were pointed out. Thereafter, in accordance with the first priority in this dissertation, a scalarisation-based methodology for MICFMO was proposed in order to address the shortcomings present in the existing methodology (in partial fulfilment of Dissertation Objective IV of §1.4). The components of the new methodology, namely an augmented Chebyshev metric-based scalarising objective function, an additive penalty function constraint handling technique and an adapted HS algorithm, were described in some detail. The proposed methodology was furthermore implemented by the author within a completely revised version of the MICFMO decision support feature in the OSCAR-4 system (in partial fulfilment of Dissertation Objective IV of §1.4). This feature was used to solve several ICFMO problem instances for the SAFARI-1 and HOR reactors. The results thus obtained were compared to reload configurations that were designed according to the current reload approach followed at each reactor (in final fulfilment of Dissertation Objective IV of §1.4). Numerical results indicated that the newly proposed methodology is robust and versatile, and that it may be used as an effective decision support tool for designing reload configurations.

Several ANNs were constructed in Chapter 6 as surrogate models for the prediction of SAFARI-1 core parameters corresponding to MICFMO objectives and constraints (in fulfilment of Dissertation Objective V of §1.4). The chapter opened with a motivation of the necessity of these neural networks, namely to aid in the investigation of different computational methods for solving the MICFMO problem. General concepts pertaining to ANNs were presented in the next section, and this was followed by a more detailed description of MFNNs, as well as important notions pertaining to the architecture and training thereof. Details on the construction of a suite of neural networks for the SAFARI-1 reactor were described next, before the chapter closed with a presentation of the results obtained during the training and application of the networks. It was found that the MFNNs have the ability to predict SAFARI-1 core parameters much quicker (and with an acceptable accuracy) than when using the OSCAR-4 system (typically by four orders of magnitude). This finding was made within the context of a single operational cycle of the SAFARI-1 reactor, using a fixed set of fuel assemblies as network input.

In Chapter 7, several multiobjective metaheuristics considered for application in the context of MICFMO were discussed (in support of the fulfilment of Dissertation Objective VII of §1.4). These metaheuristics include two evolutionary algorithms (NSGA-II and SPEA2), two swarm intelligence algorithms (OMOPSO and P-ACO), two local search algorithms (AMOSA and MOVNS), a probabilistic model-based algorithm (MOOCEM), and a multiobjective version of the HS algorithm (MOHS). Each metaheuristic was described in some detail, and a pseudo-code listing thereof was provided. These MOAs were deliberately sourced from different classes of

metaheuristics in an attempt to encompass the diversity of algorithms available in the literature. A new constraint handling technique for MOO, referred to as the MPF technique, was proposed in this chapter (in partial fulfilment of Dissertation Objective VIII of §1.4), while a description of the existing CDP technique was also presented. The observation that the CDP technique cannot necessarily be adopted within any MOA served as motivation for the proposal of the MPF technique. The chapter closed with a discussion on the topic of performance assessment for MOAs.

Some of the main results of this dissertation were presented in Chapter 8, where the constraint handling techniques and multiobjective metaheuristics of Chapter 7 were applied to a test suite of constrained MICFMO problem instances. The results thus obtained were compared in terms of solution quality (given a fixed computation budget). These comparisons served the purpose of identifying which computational methods are most suitable in the context of constrained MICFMO, in accordance with second priority in this dissertation. The chapter opened with the creation of the aforementioned test suite (in fulfilment of Dissertation Objective VI of §1.4) which consisted of sixteen constrained MICFMO problem instances based on the SAFARI-1 reactor. The construction of this test suite was motivated by the absence of standard benchmark problem instances for MICFMO in the literature. The experimental design followed during the comparative study was presented in the next section, after which a description of the nonparametric statistical analysis conducted on the numerical results was presented. In fulfilment of Dissertation Objective VII, and in final fulfilment of Dissertation Objective VIII of §1.4, the chapter closed with a presentation of the numerical results of the comparative study. It was found that the newly-proposed MPF technique is a competitive alternative to the existing CDP technique within the context of constrained MICFMO. The metaheuristic comparison, on the other hand, indicated that the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing MOAs across the problem instances in the context of the test suite. In addition, it was found that the MOVNS algorithm also performs well in the context of the bi-objective problem instances.

In Chapter 9, a multiobjective hyperheuristic, called the AMALGAM method, was investigated in terms of its ability to conduct effective constrained MICFMO in the context of the test suite of problem instances established in Chapter 8 (in fulfilment of Dissertation Objective IX of §1.4). The aim of the investigation was to improve upon the level of generality at which MICFMO may be performed, in further support of the second priority in this dissertation. The chapter opened with a discussion on the general working of the AMALGAM method, after which the findings of Chapter 8 were incorporated into the method. The necessary modifications made in this dissertation to the original formulation of the method were also described. Four variants of the AMALGAM method (depending on the combination of sub-algorithms implemented) were then identified for comprehensive investigation in a two-stage comparative study. During the first stage, the promising variants of the AMALGAM method were compared to one another in order to select a preferred variant. Then, during the second stage, that preferred variant was compared against its constituent sub-algorithms (whose results were taken from Chapter 8), in order to determine whether it is beneficial to use the AMALGAM method instead of the individual MOAs. As before, an extensive nonparametric statistical analysis was conducted in the context of the results obtained. It was found that the $\text{AMALGAM}_{\text{npm}}$ method, whose sub-algorithms are the NSGA-II, the P-ACO algorithm and the MOOCEM, is the preferred variant of the hyperheuristic. Furthermore, it was inferred that this variant of the method significantly outperforms the three individual sub-algorithms (with a significance level of $\tilde{\alpha} = 0.05$) in the majority of instances within the MICFMO test suite. The hyperheuristic therefore achieved the dual goal of raising the level of generality at which MICFMO may be performed and of yielding improved optimisation performance.

Based on the recommendation in Chapter 9, the preferred AMALGAM$_{\text{npm}}$ method was implemented by the author within a new MICFMO decision support feature in the OSCAR-4 system (in partial fulfilment of Dissertation Objective X of §1.4). In Chapter 10, this new feature was applied to carry out several MICFMO case studies in the context of the SAFARI-1 and HOR reactors (in final fulfilment of Dissertation Objective X of §1.4). Six problem instances were solved using this feature in order to demonstrate the practical applicability of the AMALGAM$_{\text{npm}}$ method. Numerical results indicated that the method is able to find good-quality solutions to constrained MICFMO problem instances that are suitable for practical usage. The new feature may therefore be used as an effective decision support tool for designing reload configurations.

Finally, in Chapter 11, a conceptual framework was proposed for an optimisation-based personal DSS, dedicated to MICFM (in fulfilment of Dissertation Objective XI of §1.4). The chapter opened with a motivation for the proposal, along with some basic background information concerning DSSs. Thereafter, the DSS framework was presented and each of its constituent components was discussed in some detail. Several suggestions were also put forward for populating the various generic components of the system.

12.2 Appraisal of dissertation contributions

The main contributions of this dissertation are sixfold. The first contribution centres around the proposal of the scalarisation-based methodology for constrained MICFMO in §5.3, its implementation in the OSCAR-4 system and its application to several problem instances based on the SAFARI-1 and HOR research reactors in §5.4 and §5.5, respectively. The theory which underlies the augmented Chebyshev-based scalarising objective function in the methodology ensures that the shortcomings of the widely-employed weighting method (as identified in §5.2) are addressed. The arbitrariness of weighting coefficient selection is also avoided by adopting a more natural approach in which aspiration levels have to be specified. The APF constraint handling technique ensures that the methodology may be applied to both constrained and unconstrained problem instances, while the adapted HS algorithm is an adequate solution technique. It was also shown that both single- and multiobjective problem instances may be modelled and solved using the methodology, thus enhancing its flexibility. Feedback received from users utilising this MICFMO feature in the OSCAR-4 system has been positive thus far, although the usage of the feature has not progressed beyond scoping analyses. This work has been published in [183, 187, 189].

The second contribution of this dissertation is the MFNNs constructed as surrogate models for the prediction of various SAFARI-1 core parameters in §6.5. It constitutes the first application of ANN modelling to the SAFARI-1 reactor and the results obtained were very promising. It was found that the MFNNs have the ability to predict SAFARI-1 core parameters much quicker (and with acceptable accuracy) than when using the OSCAR-4 system (typically by four orders of magnitude). To the best knowledge of the author, the content of Chapter 6 also constitutes the most comprehensive demonstration to date of the capability of ANNs to predict nuclear reactor core parameters, in the sense that all of the following parameters have been modelled successfully: thermal neutron flux in various locations across the core (in the context of both average and maximum values over a region), power levels in certain locations across the core (in the context of both total and minimum values over a region), the power peaking factor, the shutdown margin, the excess reactivity and the control bank worth. This work has been published in [186].

The third contribution consists of two parts, namely the development of a new multiplicative penalty function constraint handling technique for MOO in §7.2.2, and the first application of the

MOOCEM to the MICFMO problem. These two sub-contributions are closely linked since the MPF technique was developed initially for application within the MOOCEM. The mechanism of the MPF technique is simple — instead of adding different penalty values to incommensurate objectives (as is typically done in the literature), a single value is used to penalise all the objectives simultaneously (by multiplication). Traditional domination may then be applied within any MOA using the penalised objective vector. Furthermore, only one free parameter has to be tuned within the technique. As mentioned, the MPF technique was applied within the MOOCEM, which led to the first application of the method to the constrained MICFMO problem. A rigorous derivation of the problem-specific updating rule in the MOOCEM, so as to apply the method in the context of MICFMO, was also performed. This work has been published in [188].

The fourth contribution of this dissertation revolves around the work contained in Chapter 8, namely the comparative study between the newly-proposed MPF techique and the existing CDP technique within the context of constrained MICFMO (in §8.4.1), and the extensive multiobjective metaheuristic comparative study involving eight different MOAs for solving constrained MICFMO problem instances (in §8.4.2). A test suite of various MICFMO problem instances based on the SAFARI-1 reactor was established specifically for these comparisons. In addition, these studies constitute the first application of the SPEA2, and the OMOPSO, P-ACO, AMOSA, MOVNS and MOHS algoritms to the MICFMO problem. The comparisons were performed in a structured and statistically sound manner which involved the application of nonparametric statistical procedures, within the context of both a single-problem and a multi-problem analysis. As such, it supports the ongoing drive in the literature towards more structured experimental studies. To the best knowledge of the author, the content of §8.4.2 constitutes the largest single study of its kind, comparing (for the first time) eight modern state-of-the-art multiobjective metaheuristics in the context of constrained MICFMO on the largest number of problem instances, utilising two MOO performance indicators. In addition, the number of objectives considered in each of these MICFMO instances was also varied between two, three, and four objectives, in order to gauge the scalability of the metaheuristics. It was found that the NSGA-II, the P-ACO algorithm and the MOOCEM are generally the best-performing MOAs with respect to both the I_{HVD} and I_{R2} performance indicators across the problem instances in the test suite. Furthermore, it was found that the MOVNS algorithm also performs well in the context of the bi-objective problem instances. Finally, it was also found that the MPF technique is a competitive alternative to the CDP technique, with neither approach being able to consistently outperform the other. This work has been published in [184].

The fifth contribution centres around the investigation, implementation and application of the AMALGAM method to constrained MICFMO in Chapters 9 and 10. This work constitutes the first application of a multiobjective hyperheuristic (*i.e.* the AMALGAM method) to the MICFMO problem. The investigation into the suitability of the method, also following a structured and statistically sound procedure, may be viewed as an extension of the metaheuristic comparative study. Whereas no single multiobjective metaheuristic was able to consistently outperform the other algorithms in the comparative study of §8.4.2, it was found that the $\text{AMALGAM}_{\text{npm}}$ method, whose sub-algorithms are the NSGA-II, the P-ACO algorithm and the MOOCEM, significantly outperforms its three individual sub-algorithms in the majority of instances within the MICFMO test suite. The hyperheuristic therefore raises the level of generality at which MICFMO may be performed and it is capable of yielding improved optimisation performance. This variant of the AMALGAM method was also implemented in the OSCAR-4 system, although it has not been released yet to users of the system. In respect of several MICFMO case studies that were conducted based on the SAFARI-1 and HOR reactors, using this capability of the OSCAR-4 system, it was found that good-quality solutions to the con-

strained MICFMO problem are obtained, which are suitable for practical usage. The intention is to submit this work for possible future publication.

Finally, the sixth contribution of the dissertation was the design of a conceptual framework for an optimisation-based personal DSS, dedicated to MICFM in Chapter 11. This framework was informed by existing approaches within the literature and by requirements further identified by the author. The framework is novel in the sense that, to the best knowledge of the author, no other framework proposals of a similar scope of application exist in the ICFM literature. Several suggestions were also put forward for populating the various generic components of the system. This work has been accepted for publication [185].

12.3 Suggestions for future work

In fulfilment of Dissertation Objective XII of §1.4, a number of suggestions are made in this section with respect to possible future work related to the contents of this dissertation.

Suggestion 12.1: *Apply the scalarisation-based methodology and the AMALGAM$_{npm}$ method to MICFMO problem instances involving power reactors.*

The scalarisation-based methodology and the AMALGAM$_{\text{npm}}$ method, as implemented within the OSCAR-4 system, are capable of solving (approximately) any MICFMO problem instance conforming to the model in (4.1) presented in §4.3. Following the work conducted in this dissertation in respect of research reactor application, a natural next step would be to apply the aforementioned two approaches for solving MICFMO problem instances involving power reactors. The model in (4.1) may also be extended so that fuel assembly rotations and burnable absorber placement are incorporated in the optimisation problem, thus enhancing the realism for power reactor applications. Such model changes would necessitate algorithmic modifications to these MOO approaches, and may therefore also be pursued in future work.

Suggestion 12.2: *Consider the investigation of alternative metaheuristics for application to constrained MICFMO.*

Although a diverse mix of multiobjective metaheuristics were considered in this dissertation, the realm of MOAs is vast. Several alternative MOAs may therefore be investigated for use in MICFMO. The following examples constitute a small subset of additional metaheuristics, based on different design paradigms, that may be pursued: the *Pareto efficient global optimization* (ParEGO) algorithm [104], the *multiobjective evolutionary algorithm based on decomposition* (MOEA/D) [244], the *archive-based hybrid scatter search* (AbYSS) algorithm [144], the *multiobjective Tchebycheff-based genetic algorithm* (MOTGA) [3], the *R2 indicator-based evolutionary algorithm* (R2-IBEA) [160], and the *probabilistic model-based multiobjective evolutionary algorithm* (MMEA) [247]. Similarly, there surely exist other single-objective metaheuristics which may obtain better solutions than the adapted HS algorithm currently utilised in the scalarisation-based methodology for MICFMO. Alternative metaheuristics such as SA, tabu search, GAs, differential evolution, ACO, PSO, estimation of distribution algorithms, and many more [41, 213] may therefore also be investigated.

Suggestion 12.3: *Consider further investigation of the AMALGAM method.*

As pointed out by Raad [170], the generic version of the AMALGAM method is not without its shortcomings. First, the selection for replacement of solutions, from one generation to the next, is currently performed according to the strategy of the NSGA-II. Adopting a different strategy may, however, lead to improved results. In [171], for example, a variant of the AMALGAM method in which the SPEA2 selection strategy is adopted yielded superior results in the context

of water distribution systems design. Secondly, the offspring partitioning/reward scheme of (9.1) does not explicitly take the quality of fitness improvement into account. A sub-algorithm yielding few successful offspring solutions of superior quality may therefore be rewarded insufficiently within the existing AMALGAM scheme, compared to a sub-algorithm yielding many successful offspring solutions of inferior quality. It is proposed that indicator-based selection strategies and reward schemes may be investigated to alleviate this problem. The contribution of a successful offspring solution to the hypervolume of the population may, for example, be incorporated into the reward scheme, while a similar approach may be used in the selection strategy (as was done in [48]). Another example may be to incorporate the R2 indicator into the selection strategy/reward scheme, taking inspiration from the work in [160].

Suggestion 12.4: *Extend the investigation of the multiobjective metaheuristics to a scenario in which a greater computational budget is allowed, or a measure of convergence is adopted as algorithmic stopping criterion.*

In the comparative studies conducted in this dissertation, a fixed computational budget of approximately 1 000 evaluations was adopted as the stopping criterion for each MOA. This scope delimitation was motivated in §4.2.6 and is in line with the current practical limitation of MICFMO for the SAFARI-1 reactor. The inferences drawn from these studies are, however, not necessarily valid for instances in which a larger computational budget is allowed. Some MOAs may yield poor results during their initial stages but ultimately outperform other algorithms given sufficient time. A natural next step would therefore be to investigate the performance of each MOA when the computational budget is relaxed, for example, to 10 000 evaluations. Another option may be to adopt some measure of convergence as the algorithmic stopping criterion (*e.g.* no change in the population for ten consecutive generations) instead of a fixed computational budget. Examples of scenarios in which such a relaxed computational budget is allowed may be when offline optimisation calculations are sought, instead of cycle-to-cycle online calculations, or when a reactor experiences a prolonged shutdown period before the next operational cycle.

Suggestion 12.5: *Resolve the permutation sampling discrepancy in the MOOCEM.*

As pointed out in §C.3 during the derivation of the MICFMO updating rule of the MOOCEM, the algorithm for generating permutation solutions is simply employed as a heuristic to speed up the solution generation process during optimisation. Accordingly, solutions generated by Algorithm 7.10 do not follow the exact same probability distribution as those generated according to the sampling procedure used earlier in the derivation (in which the decision variables were assumed to be independent). This discrepancy is not limited to the MICFMO problem, and has also been encountered in, for example, the travelling salesman problem [30] and the BAP [2]. A direction for future work may therefore be to design a permutation sampling algorithm in which the probability distribution is followed more closely. Another direction may be to derive an updating rule according to a probability distribution in which the decision variables are assumed to be dependent.

Suggestion 12.6: *Establish a database of MICFMO benchmark problem instances and associated MOO results.*

A significant difficulty associated with an algorithmic comparative study in the context of MICFMO, such as the one conducted in this dissertation, is the lack of readily available data for the validation of optimisation results. As mentioned in §8.1, no standard benchmark problem instances for MICFMO exist in the literature. It is therefore suggested that the ICFM research community establish an online repository of MICFMO benchmarks that may be used by future researchers to enhance the quality of experimental studies conducted. This would also necessi-

tate the usage of a single (freely-available) deterministic reactor core calculation code system, or a corresponding deterministic surrogate model for each nuclear reactor considered, in order to ensure that function evaluations are performed consistently. Such a benchmark repository may also double as a database of MOO results obtained for each problem instance.

Suggestion 12.7: *Consider incorporating heuristic information into the MOAs.*

Apart from adopting a permutation encoding within each MOA, and the heuristic information matrix forming part of the P-ACO algorithm, no *a priori* knowledge in respect of the MICFMO problem was incorporated into any of the metaheuristics considered in this dissertation. It is, however, well established in the literature that inclusion of problem-specific information into a solution technique may accelerate and/or improve the quality of solutions obtained for a given instance. The generality of the solution technique may, however, be adversely affected by such an inclusion of knowledge (as per the NFL theorem). Inclusion of heuristic information into the MOAs may therefore be investigated, although special care would have to be taken in order to avoid a significant loss of generality. It is likely, however, that the MOVNS algorithm will be an exception to this phenomenon. Several neighbourhood move operators may be designed based on different heuristics. These operators may then be employed in conjunction with swap and scramble operators, thus potentially improving the generality of the algorithm.

Suggestion 12.8: *Consider enhancing the generalisation of the MFNNs in order to utilise them in the context of multicycle ICFMO.*

It was demonstrated in Chapter 6 that the predictive cabilities of the MFNNs constructed in this dissertation are restricted to the operational cycle on which the networks were trained. In other words, the networks exhibit insufficient generalisation for applicability to other operational cycles. A next step would be to reconstruct the networks (by employing a larger and more diverse training set which may consist of different or additional input parameters, and possibly different architectures) and attempt to achieve sufficient generalisation so that predictions of acceptable accuracy may be realised for an arbitrary operational cycle. Such neural networks may then be employed to aid investigations within the context of *multicycle* MICFMO which, ordinarily, is computationally much more expensive than single-cycle optimisation.

Suggestion 12.9: *Consider the implementation of the proposed optimisation-based personal DSS in a computerised tool.*

The proposed optimisation-based personal DSS, dedicated to MICFMO, may be implemented as a computerised tool in order to demonstrate its functionality. A number of techniques and approaches were suggested in §11.4 for populating some of the generic DSS components. Those suggestions may be used as a starting point in the development of such a concept demonstrator.

Suggestion 12.10: *Consider the investigation of MCDA techniques in the context of MICFMO.*

As highlighted in §11.1, a decision maker should ultimately choose only one reload configuration from a nondominated set of configurations returned by an MOA, according to his subjective preferences. Since it is not necessarily obvious how to choose this preferred reload configuration, some level of MCDA support should form a crucial component in MICFM. To the best knowledge of the author, no research on the development or application of MCDA techniques to MICFM is available in the literature. As such, future work on the subject is highly encouraged.

References

[1] Alim F, Ivanov K & Levine SH, 2008, *New genetic algorithms (GA) to optimize PWR reactors Part I: Loading pattern and burnable poison placement optimization techniques for PWRs*, Annals of Nuclear Energy, **35**, pp. 93–112.

[2] Alon G, Kroese DP, Raviv T & Rubinstein RY, 2005, *Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment*, Annals of Operations Research, **134**, pp. 137–151.

[3] Alves MJ & Almeida M, 2007, *MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem*, Computers & Operations Research, **34**, pp. 3458–3470.

[4] Angelo JS & Barbosa HJC, 2011, *On ant colony optimization algorithms for multi-objective problems*, pp. 53–74 in Ostfeld A (Ed), *Ant colony optimization — Methods and applications*, InTech, Rijeka.

[5] Arctic Expo Centre — Icebreaker Lenin, 2012, *Arctic Centre leads transformation of the first nuclear icebreaker into a Science Centre*, [Online], [Cited November 2nd, 2015], Available from `https://leninicebreaker.wordpress.com/2012/06/05/press-release/`.

[6] Argonne National Laboratory, 2014, *CP-1 (Chicago Pile 1 Reactor)*, [Online], [Cited November 1st, 2015], Available from `http://www.ne.anl.gov/About/reactors/early-reactors.shtml`.

[7] Arnott D & Pervan G, 2005, *A critical analysis of decision support systems research*, Journal of Information Technology, **20**, pp. 67–87.

[8] Arshi SS, Zolfaghari A & Mirvakili SM, 2014, *A multi-objective shuffled frog leaping algorithm for in-core fuel management optimization*, Computer Physics Communications, **185**, pp. 2622–2628.

[9] Babazadeh D, Boroushaki M & Lucas C, 2009, *Optimization of fuel core loading pattern design in a VVER nuclear power reactors using particle swarm optimization (PSO)*, Annals of Nuclear Energy, **36**, pp. 923–930.

[10] Bandyopadhyay S, Saha S, Maulik U & Deb K, 2008, *A simulated annealing-based multiobjective optimization algorithm: AMOSA*, IEEE Transactions on Evolutionary Computation, **12(3)**, pp. 269–283.

[11] Barr RS, Golden BL, Kelly JP, Resende MGC & Stewart WR, 1995, *Designing and reporting on computational experiments with heuristic methods*, Journal of Heuristics, **1**, pp. 9–32.

[12] Beale MH, Hagan MT & Demuth HB, 2014, *Neural Network Toolbox User's Guide*, The Mathworks Inc., Natick (MA).

[13] Bean JC, 1994, *Genetic algorithms and random keys for sequencing and optimization*, ORSA Journal on Computing, **6(2)**, pp. 154–160.

[14] Bekker J, 2012, *Applying the cross-entropy method in multi-objective optimisation of dynamic stochastic systems*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[15] Bekker J & Aldrich C, 2011, *The cross-entropy method in multi-objective optimisation: An assessment*, European Journal of Operational Research, **211**, pp. 112–121.

[16] Ben Hmaida IA, Carter JN, de Oliveira CRE, Goddard AJH & Parks GT, 1999, *Nuclear in-core fuel management optimisation using the tabu search method*, Proceedings of the International Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, Madrid, pp. 1658–1666.

[17] Bishop CM, 1995, *Neural networks for pattern recognition*, Clarendon Press, Oxford.

[18] Bishop CM, 2006, *Pattern recognition and machine learning*, Springer, New York (NY).

[19] Blaauw M, 2016, *Pictures reactor core*, [Online], [Cited June 16th, 2016], Available from `http://www.tnw.tudelft.nl/en/cooperation/facilities/reactor-instituut-delft/research/facilities/hoger-onderwijs-reactor-hor/pictures-reactor-core/`.

[20] Blaauw M, 2016, *Reactor Institute Delft*, [Brochure], Delft University of Technology, Delft.

[21] Black PE, 2015, *Fisher-Yates shuffle*, [Online], [Cited August 31st, 2016], Available from `http://www.nist.gov/dads/HTML/fisherYatesShuffle.html`.

[22] Branke J, Deb K, Miettinen K & Slowinski R (Eds), 2008, *Multiobjective optimization: Interactive and evolutionary approaches*, Springer, Berlin.

[23] Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E & Qu R, 2013, *Hyper-heuristics: A survey of the state of the art*, Journal of the Operational Research Society, **64**, pp. 1695–1724.

[24] Burke EK & Kendall G (Eds), 2005, *Search methodologies: Introductory tutorials in optimization and decision support techniques*, Springer, New York (NY).

[25] Caldas GHF & Schirru R, 2008, *Parameterless evolutionary algorithm applied to the nuclear reload problem*, Annals of Nuclear Energy, **35**, pp. 583–590.

[26] Chakhlevitch K & Cowling P, 2008, *Hyperheuristics: Recent developments*, pp. 3–29 in Cotta C, Sevaux M & Sörenson K (Eds), *Adaptive and multilevel metaheuristics*, Springer-Verlag, Berlin Heidelberg.

[27] Chapot JLC, da Silva FC & Schirru R, 1999, *A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem*, Annals of Nuclear Energy, **26**, pp. 641–655.

[28] Chen Y, Mingle JO & Eckhoff ND, 1977, *Optimal power profile fuel management*, Annals of Nuclear Energy, **4**, pp. 407–415.

[29] Coello Coello CA, Lamont GB & Van Veldhuizen DA, 2007, *Evolutionary algorithms for solving multi-objective problems*, 2nd Edition, Springer, New York (NY).

[30] de Boer PT, Kroese DP, Mannor S & Rubinstein RY, 2005, *A tutorial on the cross-entropy method*, Annals of Operations Research, **134(1)**, pp. 19–67.

[31] de Klerk E, Roos C, Terlaky T, Illés T, de Jong AJ, Valkó J & Hoogenboom JE, 1997, *Optimization of nuclear reactor reloading patterns*, Annals of Operations Research, **69**, pp. 65–84.

[32] de Lima AMM, Schirru R, da Silva FC & Medeiros JACC, 2008, *A nuclear reactor core fuel reload optimization using artificial ant colony connective networks*, Annals of Nuclear Energy, **35**, pp. 1606–1612.

[33] de Oliveira IMS & Schirru R, 2011, *Swarm intelligence of artificial bees applied to in-core fuel management optimization*, Annals of Nuclear Energy, **38**, pp. 1039–1045.

[34] 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.

[35] DeChaine MD & Feltus MA, 1995, *Nuclear fuel management optimization using genetic algorithms*, Nuclear Technology, **111**, pp. 109–114.

[36] Derrac J, García S, Molina D & Herrera F, 2011, *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm and Evolutionary Computation, **1**, pp. 3–18.

[37] Do BQ & Nguyen LP, 2007, *Application of a genetic algorithm to the fuel reload optimization for a research reactor*, Applied Mathematics and Computation, **187**, pp. 977–988.

[38] Doerner K, Gutjahr W, Hartl RF, Strauss C & Stummer C, 2004, *Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection*, Annals of Operations Research, **131**, pp. 79–99.

[39] Dorigo M, Maniezzo V & Colorni A, 1991, *Positive feedback as a search strategy*, (Unpublished) Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milano.

[40] Downar TJ & Sesonske A, 1988, *Light water reactor fuel cycle optimization: Theory versus practice*, pp. 71–126 in Lewins J & Becker M (Eds), *Advances in nuclear science and technology, Volume 20*, Springer, New York (NY).

[41] Dréo J, Pétrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimization — Methods and case studies*, Springer-Verlag, Berlin.

[42] Duderstadt JJ & Hamilton LJ, 1976, *Nuclear reactor analysis*, John Wiley & Sons, Inc., New York (NY).

[43] Dumas M & Robeau D, 1981, *Fuel management optimization for a PWR*, Proceedings of the International Topical Meeting on Advances in Mathematical Methods for Nuclear Engineering Problems, Munich, pp. 357–367.

[44] Durillo JJ, García-Nieto J, Nebro AJ, Coello Coello CA, Luna F & Alba E, 2009, *Multi-objective particle swarm optimizers: An experimental comparison*, Proceedings of the International Conference on Evolutionary Multi-criterion Optimization (EMO 2009), Nantes, pp. 495–509.

[45] Durillo JJ, Nebro AJ, Luna F, Coello Coello CA & Alba E, 2010, *Convergence speed in multi-objective metaheuristics: Efficiency criteria and empirical study*, International Journal for Numerical Methods in Engineering, **84**, pp. 1344–1375.

[46] Ehrgott M & Gandibleux X, 2008, *Hybrid metaheuristics for multi-objective combinatorial optimization*, pp. 221–259 in Blum C, Aguilera MJB, Roli A & Sampels M (Eds), *Hybrid metaheuristics: An emerging approach to optimization*, Springer, Berlin.

[47] Eiben AE & Smit SK, 2011, *Parameter tuning for configuring and analyzing evolutionary algorithms*, Swarm and Evolutionary Computation, **1**, pp. 19–31.

[48] Emmerich M, Beume N & Naujoks B, 2005, *An EMO algorithm using the hypervolume measure as selection criterion*, Proceedings of the Third International Conference on Evolutionary Multi-criterion Optimization (EMO 2005), Guanajuato, pp. 62–76.

[49] Encyclopaedia Britannica, 2015, *Manhattan Project*, [Online], [Cited November 1st, 2015], Available from `http://www.britannica.com/event/Manhattan-Project`.

[50] Encyclopaedia Britannica, 2015, *Nuclear fission*, [Online], [Cited October 28th, 2015], Available from `http://www.britannica.com/science/nuclear-fission`.

[51] Encyclopaedia Britannica, 2015, *Nuclear reactor*, [Online], [Cited November 1st, 2015], Available from `http://www.britannica.com/technology/nuclear-reactor`.

[52] Encyclopaedia Britannica, 2015, *Plutonium (Pu)*, [Online], [Cited November 1st, 2015], Available from `http://www.britannica.com/science/plutonium`.

[53] Encyclopaedia Britannica, 2016, *Positron*, [Online], [Cited June 16th, 2016], Available from `http://www.britannica.com/science/positron`.

[54] *Energy, electricity and nuclear power estimates for the period up to 2050*, 2015, International Atomic Energy Agency, Vienna.

[55] Engrand P, 1998, *A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management*, (Unpublished) Technical Report 98NB00037, Electricité de France (EDF), Clamart, France.

[56] Erdoğan A & Geçkinli M, 2003, *A PWR reload optimisation code (XCore) using artificial neural networks and genetic algorithms*, Annals of Nuclear Energy, **30**, pp. 35–53.

[57] Faria EF & Pereira C, 2003, *Nuclear fuel loading pattern optimisation using a neural network*, Annals of Nuclear Energy, **30**, pp. 603–613.

[58] Fausett LV, 1994, *Fundamentals of neural networks: Architectures, algorithms, and applications*, Prentice-Hall, Englewood Cliffs (NJ).

[59] Foresee FD & Hagan MT, 1997, *Gauss-Newton approximation to Bayesian learning*, Proceedings of the International Conference on Neural Networks, Houston (TX), pp. 1930–1935.

[60] Galperin A, 1995, *Exploration of the search space of the in-core fuel management problem by knowledge-based techniques*, Nuclear Science and Engineering, **119**, pp. 144–152.

[61] Galperin A, Kimhi S & Segev M, 1989, *A knowledge-based system for optimization of fuel reload configurations*, Nuclear Science and Engineering, **102**, pp. 43–53.

[62] Galperin A & Kimhy Y, 1991, *Application of knowledge-based methods to in-core fuel management*, Nuclear Science and Engineering, **109**, pp. 103–110.

[63] Galperin A & Nissan E, 1988, *Application of a heuristic search method for generation of fuel reload configurations*, Nuclear Science and Engineering, **99**, pp. 343–352.

[64] Geem ZW, Kim JH & Loganathan GV, 2001, *A new heuristic optimization algorithm: Harmony search*, Simulation, **76(2)**, pp. 60–68.

[65] Geiger MJ, 2004, *Randomised variable neighbourhood search for multi objective optimisation*, Proceedings of the EU/ME workshop: Design and evaluation of advanced hybrid meta-heuristics, Nottingham, pp. 34–42.

[66] General Atomics, 2015, *TRIGA nuclear reactors*, [Online], [Cited November 5th, 2015], Available from `http://www.ga.com/triga`.

[67] Geoffrion AM, 1968, *Proper efficiency and the theory of vector maximization*, Journal of Mathematical Analysis and Applications, **22(3)**, pp. 618–630.

[68] Glover F & Sörensen K, 2015, *Metaheuristics*, Scholarpedia, **10(4)**, pp. 6532.

[69] Goldberg DE, 1989, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Boston (MA).

[70] Hagan MT & Menhaj MB, 1994, *Training feedforward networks with the Marquardt algorithm*, IEEE Transactions on Neural Networks, **5(6)**, pp. 898–993.

[71] Hamasaki M & Takeda T, 1986, *Application of depletion perturbation theory to fuel loading optimization*, Journal of Nuclear Science and Technology, **23(1)**, pp. 1–10.

[72] Hansen MP & Jaszkiewicz A, 1998, *Evaluating the quality of approximations to the non-dominated set*, (Unpublished) Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark.

[73] Hébert A, 2010, *Multigroup neutron transport and diffusion computations*, pp. 751–911 in Cacuci DG (Ed), *Handbook of nuclear engineering*, Springer, New York (NY).

[74] Hedayat A, 2014, *Developing a practical optimization of the refueling program for ordinary research reactors using a modified simulated annealing method*, Progress in Nuclear Energy, **76**, pp. 191–205.

[75] Hedayat A, Davilu H, Barfrosh AA & Sepanloo K, 2009, *Optimization of the core configuration design using a hybrid artificial intelligence algorithm for research reactors*, Nuclear Engineering and Design, **239**, pp. 2786–2799.

[76] Hill NJ & Parks GT, 2015, *Pressurized water reactor in-core fuel management by tabu search*, Annals of Nuclear Energy, **75**, pp. 64–71.

[77] Ho ALB & Sesonske A, 1982, *Extended burnup fuel cycle optimization for pressurized water reactors*, Nuclear Technology, **58**, pp. 422–436.

[78] Ho L & Rohach AF, 1982, *Perturbation theory in nuclear fuel management optimization*, Nuclear Science and Engineering, **82**, pp. 151–161.

[79] Hoareau F, 2008, *Loading pattern optimization using ant colony algorithm*, Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2008), Interlaken, Available on CD-ROM.

[80] Hochberg Y & Tamhane AC, 1987, *Multiple comparison procedures*, John Wiley & Sons, New York (NY).

[81] Holland JH, 1975, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (MI).

[82] Hollander M, Wolfe DA & Chicken E, 2014, *Nonparametric statistical methods*, 3rd Edition, John Wiley & Sons, Hoboken (NJ).

[83] Hongchun W, 2001, *Pressurized water reactor reloading optimization using genetic algorithms*, Annals of Nuclear Energy, **28**, pp. 1329–1341.

[84] Hooke R & Jeeves TA, 1961, *"Direct search" solution of numerical and statistical problems*, Journal of the ACM, **8(2)**, pp. 212–229.

[85] Hornik K, Stinchcombe M & White H, 1989, *Multilayer feedforward networks are universal approximators*, Neural Networks, **2**, pp. 359–366.

[86] Hu X, Eberhart RC & Shi Y, 2003, *Swarm intelligence for permutation optimization: A case study of n-queens problem*, Proceedings of the IEEE Swarm Intelligence Symposium (SIS'03), Indianapolis (IN), pp. 243–246.

[87] Hwang CL & Masud ASM, 1979, *Multiple objective decision making — Methods and applications: A state-of-the-art survey*, Springer-Verlag, Berlin.

[88] International Atomic Energy Agency Research Reactor Database, 2015, *Operational status of research reactors*, [Online], [Cited November 5th, 2015], Available from `https://nucleus.iaea.org/RRDB/Reports/Container.aspx?Id=A1`.

[89] Jain AK & Dubes RC, 1988, *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs (NJ).

[90] Jamalipour M, Gharib M, Sayareh R & Khoshahval F, 2013, *PWR power distribution flattening using quantum particle swarm intelligence*, Annals of Nuclear Energy, **56**, pp. 143–150.

[91] Jamalipour M, Sayareh R, Gharib M, Khoshahval F & Karimi MR, 2013, *Quantum behaved particle swarm optimization with differential mutation operator applied to WWER-1000 in-core fuel management optimization*, Annals of Nuclear Energy, **54**, pp. 134–140.

[92] Jayalal ML, Satya Murty SAV & Sai Baba M, 2014, *A survey of genetic algorithm applications in nuclear fuel management*, Journal of Nuclear Engineering & Technology, **4(1)**, pp. 45–62.

[93] Jiang S, Ziver AK, Carter JN, Pain CC, Goddard AJH, Franklin S & Philips HJ, 2006, *Estimation of distribution algorithms for nuclear reactor fuel management optimisation*, Annals of Nuclear Energy, **3**, pp. 1039–1057.

[94] Karahroudi MR, Shirazi SAM & Sepanloo K, 2013, *Optimization of designing the core fuel loading pattern in a VVER-1000 nuclear power reactor using the genetic algorithm*, Annals of Nuclear Energy, **57**, pp. 142–150.

[95] Keller PM, 2001, *FORMOSA-P constrained multiobjective simulated methodology*, Proceedings of the International Topical Meeting on Mathematical Methods for Nuclear Applications (M&C 2001), Salt Lake City (UT), Available on CD-ROM.

[96] Kennedy J & Eberhart R, 1995, *Particle swarm optimization*, Proceedings of the IEEE International Conference on Neural Networks, Perth, pp. 1942–1948.

[97] Khoshahval F, Zolfaghari A, Minuchehr H & Abbasi MR, 2014, *A new hybrid method for multi-objective fuel management optimization using parallel PSO-SA*, Progress in Nuclear Energy, **76**, pp. 112–121.

[98] Khoshahval F, Zolfaghari A, Minuchehr H, Sadighi M & Norouzi A, 2010, *PWR fuel management optimization using continuous particle swarm intelligence*, Annals of Nuclear Energy, **37**, pp. 1263–1271.

[99] Kim HG, Chang SH & Lee BH, 1994, *A study on the optimal fuel loading pattern design in pressurized water reactor using the artificial neural network and the fuzzy rule based system*, Proceedings of the 4th International Topical Meeting on Nuclear Thermal Hydraulics, Operations and Safety, Taipei, 4B1–4B6.

[100] Kim HG, Chang SH & Lee BH, 1993, *Optimal fuel loading pattern design using an artificial neural network and a fuzzy rule-based system*, Nuclear Science and Engineering, **115**, pp. 152–163.

[101] Kim TK & Kim CH, 1997, *Mixed integer programming for pressurized water reactor fuel-loading-pattern optimization*, Nuclear Science and Engineering, **127**, pp. 346–357.

[102] Kim YJ, Downar TJ & Sesonske A, 1987, *Optimization of core reload design for low-leakage fuel management in pressurized water reactors*, Nuclear Science and Engineering, **96**, pp. 85–101.

[103] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimization by simulated annealing*, Science, **20(4598)**, pp. 671–680.

[104] Knowles J, 2006, *ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimisation problems*, IEEE Transactions on Evolutionary Computation, **10(1)**, pp. 50–66.

[105] Knowles JD & Corne D, 2002, *On metrics for comparing nondominated sets*, Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Honolulu (HI), pp. 711–716.

[106] Knowles JD, Thiele L & Zitzler E, 2006, *A tutorial on the performance assessment of stochastic multiobjective optimizers*, (Unpublished) Technical Report 214, Computer Engineering and Communication Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

[107] Koppes S, 2012, *How the first chain reaction changed science*, [Online], [Cited November 1st, 2015], Available from `http://www.uchicago.edu/features/how_the_first_chain_reaction_changed_science/`.

[108] Kroese DP, 2014, Professor at *The University of Queensland*, [Personal Communication], Contactable at `kroese@maths.uq.edu.au`.

[109] Kropaczek DJ & Turinsky PJ, 1991, *In-core fuel management optimization for pressurized water reactors utilizing simulated annealing*, Nuclear Technology, **95**, pp. 9–32.

[110] Krzywinski M & Altman N, 2014, *Visualising samples with box plots*, Nature Methods, **11(2)**, pp. 119–120.

[111] Lamarsh JR & Baratta AJ, 2001, *Introduction to nuclear engineering*, 3rd Edition, Prentice Hall, Upper Saddle River (NJ).

[112] Land AH & Doig AG, 1960, *An automatic method of solving discrete programming problems*, Econometrica, **28(3)**, pp. 497–520.

[113] Larrañaga P, Kuijpers CMH, Murga RH, Inza I & Dizdarevic S, 1999, *Genetic algorithms for the travelling salesman problem: A review of representations and operators*, Artificial Intelligence Review, **13**, pp. 129–170.

[114] Lee HC, Shim HJ & Kim CH, 2001, *Parallel computing adaptive simulated annealing scheme for fuel assembly loading pattern optimization in PWRs*, Nuclear Technology, **135**, pp. 39–49.

[115] Lewis EE, 2008, *Fundamentals of Nuclear Reactor Physics*, 1st Edition, Academic Press, Cambridge (MA).

[116] Li X & Du G, 2013, *BSTBGA: A hybrid genetic algorithm for constrained multi-objective optimization problems*, Computers & Operations Research, **40**, pp. 282–302.

[117] Liang Y & Chuang C, 2013, *Variable neighbourhood search for multi-objective resource allocation problems*, Robotics and Computer-Integrated Manufacturing, **29**, pp. 73–78.

[118] Liefooghe A, Humeau J, Mesmoudi S, Jourdan L & Talbi E, 2012, *On dominance-based multiobjective local search: Design, implementation and experimental analysis on scheduling and traveling salesman problems*, Journal of Heuristics, **18**, pp. 317–352.

[119] Lin C & Lin B, 2012, *Automatic pressurized water reactor loading pattern design using ant colony algorithms*, Annals of Nuclear Energy, **43**, pp. 91–98.

[120] Lin C, Yang J, Lin K & Wang Z, 1998, *Pressurized water reactor loading pattern using the simple tabu search*, Nuclear Science and Engineering, **129**, pp. 61–71.

[121] Lin K & Lin C, 1998, *Pressurized water reactor reload design by an expert system*, Nuclear Science and Engineering, **130**, pp. 128–140.

[122] Lindahl S & Weiss Z, 1981, *The response matrix method*, pp. 73–154 in Lewins J & Becker M (Eds), *Advances in nuclear science and technology, Volume 5*, Springer, New York (NY).

[123] Liu S & Cai J, 2012, *Studies of fuel loading pattern optimization for a typical pressurized water reactor (PWR) using improved pivot particle swarm method*, Annals of Nuclear Energy, **50**, pp. 117–125.

[124] Luenberger DG & Ye Y, 2008, *Linear and nonlinear programming*, 3rd Edition, Springer, New York (NY).

[125] Maashi MS, 2014, *An investigation of multi-objective hyper-heuristics for multi-objective optimisation*, PhD Thesis, University of Nottingham, Nottingham.

[126] Machado L & Schirru R, 2002, *The Ant-Q algorithm applied to the nuclear reload problem*, Annals of Nuclear Energy, **29**, pp. 1455–1470.

[127] MacKay DJC, 1992, *Bayesian interpolation*, Neural Computation, **4**, pp. 415–447.

[128] Mahlers YP, 1997, *Core loading pattern optimization for research reactors*, Annals of Nuclear Energy, **24(7)**, pp. 509–514.

[129] Maldonado GI & Turinsky PJ, 1995, *Application of nonlinear nodal diffusion generalized perturbation theory to nuclear fuel reload optimization*, Nuclear Technology, **110**, pp. 198–218.

[130] Matai R, Singh SP & Mittal ML, 2010, *Traveling salesman problem: An overview of applications, formulations and solution approaches*, pp. 1–24 in Davendra D (Ed), *Traveling salesman problem — Theory and applications*, InTech, Rijeka.

[131] Mazrou H & Hamadouche M, 2006, *Development of a supporting tool for optimal fuel management in research reactors using artificial neural networks*, Nuclear Engineering and Design, **236**, pp. 255–266.

[132] McCulloch WS & Pitts W, 1943, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics, **5**, pp. 115–133.

[133] McKay MD, Beckman RJ & Conover WJ, 1979, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, **21(2)**, pp. 239–245.

[134] Mehrotra K, Mohan CK & Ranka S, 1997, *Elements of artificial neural networks*, MIT Press, Cambridge (MA).

[135] Meitner L & Frisch OR, 1939, *Disintegration of uranium by neutrons: A new type of nuclear reaction*, Nature, **143**, pp. 239–240.

[136] Meneses AAM, Gambardella LM & Schirru R, 2010, *A new approach for heuristics-guided search in the in-core fuel management optimization*, Progress in Nuclear Energy, **52**, pp. 339–351.

[137] Meneses AAM, Machado MD & Schirru R, 2009, *Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor*, Progress in Nuclear Energy, **51**, pp. 319–326.

[138] Meneses AAM, Rancoita P, Schirru R & Gambardella LM, 2010, *A class-based search for the in-core fuel management optimization of a pressurized water reactor*, Annals of Nuclear Energy, **37**, pp. 1554–1560.

[139] Miettinen K, 1999, *Nonlinear multiobjective optimization*, Kluwer Academic Publishers, Dordrecht.

[140] Mirvakili SM, Faghihi F & Khalafi H, 2012, *Developing a computational tool for predicting physical parameters of a typical VVER-1000 core based on artificial neural network*, Annals of Nuclear Energy, **50**, pp. 82–93.

[141] Montero E, Riff M & Neveu B, 2014, *A beginner's guide to tuning methods*, Applied Soft Computing, **17**, pp. 39–51.

[142] Moore BR, Turinsky PJ & Karve AA, 1999, *FORMOSA-B: A boiling water reactor in-core fuel management optimization package*, Nuclear Technology, **126**, pp. 153–168.

[143] Naft BN & Sesonske A, 1972, *Pressurized water reactor optimal fuel management*, Nuclear Technology, **14**, pp. 123–132.

[144] Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ & Beham A, 2008, *AbYSS: Adapting scatter search to multiobjective optimization*, IEEE Transactions on Evolutionary Computation, **12(4)**, pp. 439–457.

[145] Necsa, 2015, *SAFARI-1*, [Online], [Cited November 12th, 2015], Available from `http://www.necsa.co.za/Necsa/SAFARI-1`.

[146] *Necsa photographic archive*, 2016, Necsa.

[147] Nissan E & Galperin A, 1998, *Refueling in nuclear engineering: The FUELCON project*, Computers in Industry, **37**, pp. 43–54.

[148] Nissan E, Siegelmann H, Galperin A & Kimhi S, 1997, *Upgrading automation for nuclear fuel in-core management: From the symbolic generation of configurations, to the neural adaptation of heuristics*, Engineering with Computers, **13**, pp. 1–19.

[149] Nuclear Energy Agency, 2010, *Comparing nuclear accident risks with those from other energy sources*, Organisation for Economic Co-operation and Development Publications, Paris.

[150] Nuclear Power, 2015, *Reactor core*, [Online], [Cited December 16th, 2015], Available from `http://www.nuclear-power.net/nuclear-power-plant/nuclear-reactor-core/`.

[151] *Nuclear power reactors in the world*, 2015, International Atomic Energy Agency, Vienna.

[152] Obložinský P, Herman M & Mughabghab SF, 2010, *Evaluated nuclear data*, pp. 83–187 in Cacuci DG (Ed), *Handbook of nuclear engineering*, Springer, New York (NY).

[153] *OSCAR-4 system overview: Background and main features*, 2012, [Brochure], Necsa.

[154] *Oxford Dictionary of Computing*, 1996, Oxford University Press.

[155] Palacios-Gomez F, Lasdon L & Enquist M, 1982, *Nonlinear optimization by successive linear programming*, Management Science, **28(10)**, pp. 1106–1120.

[156] Park TK, Joo HG, Kim CH & Lee HC, 2009, *Multiobjective loading pattern optimization by simulated annealing employing discontinuous penalty function and screening technique*, Nuclear Science and Engineering, **162**, pp. 134–147.

[157] Parks GT, 1996, *Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search*, Nuclear Science and Engineering, **124**, pp. 178–187.

[158] Parks GT & Suppapitnarm A, 1999, *Multiobjective optimization of PWR reload core designs using simulated annealing*, Proceedings of the International Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, Madrid, pp. 1435–1444.

[159] Parsopoulos KE & Vrahatis MN, 2002, *Particle swarm optimization method in multiobjective problems*, Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, pp. 603–607.

[160] Phan DH & Suzuki J, 2013, *R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization*, Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, pp. 1836–1845.

[161] Pittsburgh Post-Gazette, 2014, *Koeberg Nuclear Power Station*, [Online], [Cited May 20th, 2016], Available from `http://www.post-gazette.com/image/2014/11/13/ca0,93,2932,2048/Koeberg-Nuclear-Reactor.jpg`.

[162] Pomerol J, 2004, *Practical decision making — From the legacy of Herbert Simon to decision support systems*, Proceedings of the 2004 IFIP WG8.3 International Conference on Decision Support Systems (DSS2004), Tuscany, pp. 647–657.

[163] Poon PW & Parks GT, 1992, *Optimizing PWR reload core designs*, Proceedings of the Second Conference on Parallel Problem Solving from Nature, Brussels, pp. 371–380.

[164] Poursalehi N, Zolfaghari A & Minuchehr H, 2013, *Multi-objective loading pattern enhancement of PWR based on the discrete firefly algorithm*, Annals of Nuclear Energy, **57**, pp. 151–163.

[165] Poursalehi N, Zolfaghari A & Minuchehr H, 2013, *PWR loading pattern optimization using harmony search algorithm*, Annals of Nuclear Energy, **53**, pp. 288–298.

[166] Poursalehi N, Zolfaghari A & Moghaddam AMHK, 2013, *Continuous firefly algorithm applied to PWR core pattern enhancement*, Nuclear Engineering and Design, **258**, pp. 107–115.

[167] Prinsloo RH, 2016, Chief Scientist at *Necsa*, [Personal Communication], Contactable at `rian.prinsloo@necsa.co.za`.

[168] Quist AJ, van Geemert R, Hoogenboom JE, Illés T, de Klerk E, Roos C & Terlaky T, 1999, *Finding optimal nuclear reactor core reload patterns using nonlinear optimization and search heuristics*, Engineering Optimization, **32(2)**, pp. 143–176.

[169] R Core Team, 2015, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, URL: `http://www.R-project.org`.

[170] Raad DN, 2011, *Multi-objective optimisation of water distribution systems design using metaheuristics*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[171] Raad DN, Sinske A & van Vuuren JH, 2011, *Water distribution systems design optimisation using metaheuristics and hyperheuristics*, ORiON, **27(1)**, pp. 17–43.

[172] Rardin RL & Uzsoy R, 2001, *Experimental evaluation of heuristic optimization algorithms: A tutorial*, Journal of Heuristics, **7**, pp. 261–304.

[173] *Research reactors: Purpose and future*, 2010, International Atomic Energy Agency, Vienna.

[174] Reyes Sierra M & Coello Coello CA, 2005, *Improved PSO-based multi-objective optimization using crowding distance, mutation and ϵ-dominance*, Proceedings of the International Conference on Evolutionary Multi-criterion Optimization (EMO 2005), Guanajuato, pp. 505–519.

[175] Ridikas D, Shokr A & Adelfang P, 2015, *Research reactor benchmarking database: Facility specification and experimental data*, (Unpublished) Technical Report Series No. 480, International Atomic Energy Agency, Vienna.

[176] Rubinstein RY, 1997, *Optimization of computer simulation models with rare events*, European Journal of Operational Research, **99**, pp. 89–112.

[177] Rubinstein RY, 1999, *The cross-entropy method for combinatorial and continuous optimization*, Methodology and Computing in Applied Probability, **1**, pp. 127–190.

[178] Rutherford E, 1920, *Bakerian Lecture: Nuclear constitution of atoms*, Proceedings of the Royal Society of London, Series A, Containing Papers of a Mathematical and Physical Character (1905–1934), **97(686)**, pp. 374–400.

[179] Sanchez R & McCormick NJ, 1982, *A review of neutron transport approximations*, Nuclear Science and Engineering, **80**, pp. 481–535.

[180] Savva P, Chatzidakis S, Varvayanni M, Ikonomopoulos A, Chrysanthopoulou N, Catsaros N & Antonopoulos-Domis M, 2014, *Optimized flux trap dimensions in a research reactor core*, Nuclear Technology, **188(3)**, pp. 322–335.

[181] Schlünz EB, 2016, *Proprietary knowledge associated with the doctoral dissertation of EB Schlünz*, (Unpublished) Technical Report RRT-MTD-REP-16001, The South African Nuclear Energy Corporation SOC Limited (Necsa), Pelindaba, South Africa.

[182] Schlünz EB, Bokov PM & Prinsloo RH, 2012, *Application of a harmony search algorithm to the core fuel reload optimisation problem for the SAFARI-1 nuclear research reactor*, Proceedings of the 41st Annual Conference of the Operations Research Society of South Africa (ORSSA 2012), Muldersdrift, pp. 1–9.

[183] Schlünz EB, Bokov PM, Prinsloo RH & van Vuuren JH, 2016, *A unified methodology for single- and multiobjective in-core fuel management optimisation based on augmented Chebyshev scalarisation and a harmony search algorithm*, Annals of Nuclear Energy, **87**, pp. 659–670.

[184] Schlünz EB, Bokov PM & van Vuuren JH, 2016, *A comparative study on multiobjective metaheuristics for solving constrained in-core fuel management optimisation problems*, Computers & Operations Research, **75**, pp. 174–190.

[185] Schlünz EB, Bokov PM & van Vuuren JH, *An optimisation-based decision support system framework for multiobjective in-core fuel management of nuclear reactor cores*, The South African Journal of Industrial Engineering, In press.

[186] Schlünz EB, Bokov PM & van Vuuren JH, 2015, *Application of artificial neural networks for predicting core parameters for the SAFARI-1 nuclear research reactor*, Proceedings of the 44th Annual Conference of the Operations Research Society of South Africa (ORSSA 2015), Hartbeespoort, pp. 12–22.

[187] Schlünz EB, Bokov PM & van Vuuren JH, 2015, *On multiobjective optimisation approaches for in-core fuel management optimisation*, Proceedings of the Topical Meeting on the Advances in Nuclear Fuel Management V (ANFM 2015), Hilton Head Island (SC), Available on CD-ROM.

[188] Schlünz EB, Bokov PM & van Vuuren JH, 2014, *Research reactor in-core fuel management optimisation using the multiobjective cross-entropy method*, Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2014), Kyoto, Available on CD-ROM.

[189] Schlünz EB, Winkelman AJM, Prinsloo RH, Bokov PM & van Vuuren JH, 2016, *In-core fuel management optimisation of the HOR reactor using the OSCAR-4 code system*, Proceedings of the European Research Reactor Conference 2016 jointly organised with the International Group Operating Research Reactors (RRFM/IGORR-2016), Berlin, pp. 462–472.

[190] Schneider G, Chicken E & Becvarik R, 2015, *NSM3: Functions and datasets to accompany Hollander, Wolfe, and Chicken — Nonparametric statistical methods, 3rd edition*, R package version 1.3, URL: `https://cran.r-project.org/package=NSM3`.

[191] Shatilla YA, Little DC, Penkrot JA & Holland RA, 2000, *In-core fuel management with biased multiobjective function optimization*, Nuclear Technology, **130**, pp. 282–295.

[192] Shaukat N, Ahmad S, Majeed A, Ahmad N & Mohsin B, 2010, *Optimization of core reload pattern for PARR-1 using evolutionary techniques*, Nuclear Engineering and Design, **240**, pp. 2831–2835.

[193] Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R & Carlsson C, 2002, *Past, present and future of decision support technology*, Decision Support Systems, **33**, pp. 111–126.

[194] Si S, 2002, *Integer permutation programming and the loading pattern optimization code SUPERLPOS used at SNERDI*, Proceedings of the PHYSOR 2002 International Conference, Seoul, Available on CD-ROM.

[195] Silvennoinen P, 1976, *Reactor core fuel management*, Pergamon Press, Oxford.

[196] Sivasubramani S & Swarup KS, 2011, *Multi-objective harmony search algorithm for optimal power flow problem*, Electrical Power and Energy Systems, **33**, pp. 745–752.

[197] Šmuc T, Pevec D & Petrović B, 1994, *Annealing strategies for loading pattern optimization*, Annals of Nuclear Energy, **21(6)**, pp. 325–336.

[198] Soni N & Kumar T, 2014, *Study of various mutation operators in genetic algorithms*, International Journal of Computer Science and Information Technologies, **5(3)**, pp. 4519–4521.

[199] Sörensen K & Glover F, 2013, *Metaheuristics*, pp. 960–970 in Gass SI & Fu M (Eds), *Encyclopedia of operations research and management science*, Springer, New York (NY).

[200] Sprague RH, 1980, *A framework for the development of decision support systems*, MIS Quarterly, **4(4)**, pp. 1–26.

[201] Srinivas N & Deb K, 1994, *Multiobjective optimization using nondominated sorting in genetic algorithms*, Evolutionary Computation, **2(3)**, pp. 221–248.

[202] Stacey WM, 2001, *Nuclear reactor physics*, John Wiley & Sons, Inc., New York (NY).

[203] Stander G, Prinsloo RH, Müller E & Tomašević DI, 2008, *OSCAR-4 code system application to the SAFARI-1 reactor*, Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2008), Interlaken, Available on CD-ROM.

[204] Stevens JG, Smith KS, Rempe KR & Downar TJ, 1995, *Optimization of pressurised water reactor shuffling by simulated annealing with heuristics*, Nuclear Science and Engineering, **121**, pp. 67–88.

[205] Stewart TJ, 2007, *The essential multiobjectivity of linear programming*, ORiON, **23**, pp. 1–15.

[206] Stillman JA, Chao YA & Downar TJ, 1989, *The optimum fuel and power distribution for a pressurized water reactor burnup cycle*, Nuclear Science and Engineering, **103**, pp. 321–333.

[207] SubGuru, 2007, *USS Nautilus (SSN-571)*, [Online], [Cited November 2nd, 2015], Available from `http://www.subguru.com/nautilus571.htm`.

[208] Suh JS & Levine SH, 1990, *Optimized automatic reload program for pressurized water reactors using simple direct optimization techniques*, Nuclear Science and Engineering, **105**, pp. 371–382.

[209] 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**, pp. 1143–1160.

[210] Suzuki A & Kiyose R, 1971, *Application of linear programming to refueling optimization for light water moderated power reactors*, Nuclear Science and Engineering, **46**, pp. 112–130.

[211] Svozil D, Kvasnička V & Pospíchal J, 1997, *Introduction to multi-layer feed-forward neural networks*, Chemometrics and Intelligent Laboratory Systems, **39**, pp. 43–62.

[212] Tahara Y, Hamamoto K, Takase M & Suzuki K, 1991, *Computer aided system for generating fuel shuffling configurations based on knowledge engineering*, Journal of Nuclear Science and Technology, **28(5)**, pp. 399–408.

[213] Talbi E, 2009, *Metaheuristics: From design to implementation*, John Wiley & Sons, Inc., Hoboken (NJ).

[214] The MathWorks Inc., 2014, *MATLAB Release 2014a, 2015a and 2016a*, Natick (MA), url: `http://www.mathworks.com/products/matlab/`.

[215] The MathWorks Inc., 2014, *Neural Network Toolbox Release 2014a*, Natick (MA), url: `http://www.mathworks.com/products/neural-network/`.

[216] The MathWorks Inc., 2015, *Statistics and Machine Learning Toolbox Release 2015a*, Natick (MA), url: `http://www.mathworks.com/products/statistics/`.

[217] Turinsky PJ, 2010, *Core isotopic depletion and fuel management*, pp. 1241–1312 in Cacuci DG (Ed), *Handbook of nuclear engineering*, Springer, New York (NY).

[218] Turinsky PJ, 2005, *Nuclear fuel management optimization: A work in progress*, Nuclear Technology, **151**, pp. 3–8.

[219] Turinsky PJ & Parks GT, 1999, *Advances in nuclear fuel management for light water reactors*, pp. 137–165 in Lewins J & Becker M (Eds), *Advances in nuclear science and technology, Volume 26*, Springer, New York (NY).

[220] United States Nuclear Regulatory Commission, 2016, *Scram*, [Online], [Cited June 12th, 2016], Available from `http://www.nrc.gov/reading-rm/basic-ref/glossary/scram.html`.

[221] Urli NB, 1978, *Optimization theory and fuel and absorber management problems of light-water reactors*, (Unpublished) Technical Report IAEA-R-1706-F, International Atomic Energy Agency, Vienna.

[222] van Geemert R, Quist AJ, Hoogenboom JE & Gibcus HPM, 1998, *Research reactor in-core fuel management optimization by application of multiple cyclic interchange algorithms*, Nuclear Engineering and Design, **186**, pp. 369–377.

[223] Verhagen FCM, van der Schaar M, de Kruijf WJM, van de Wetering TFH & Jones RD, 1997, *ROSA, a utility tool for loading pattern optimization*, Proceedings of the Topical Meeting on the Advances in Nuclear Fuel Management II (ANFM II), Myrtle Beach (SC).

[224] Villarino EA, 2002, *Usage of burnable poison on research reactors*, Proceedings of the 2002 International Nuclear Atlantic Conference, Rio de Janeiro, Available on CD-ROM.

[225] Vogel DL & Weiss ZJ, 1992, *A general, multigroup formulation of the analytic nodal method*, Proceedings of the 1992 International Topical Meeting on Advances in Reactor Physics, Charleston (SC), pp. 497–508.

[226] Vrugt JA & Robinson BA, 2007, *Improved evolutionary optimisation from genetically adaptive multimethod search*, Proceedings of the National Academy of Sciences, **3**, pp. 708–711.

[227] Wall I & Fenech H, 1965, *The application of dynamic programming to fuel management optimization*, Nuclear Science and Engineering, **22**, pp. 285–297.

[228] Wessing S & Naujoks B, 2010, *Sequential parameter optimization for multi-objective problems*, Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, pp. 1–8.

[229] While L, Bradstreet L & Barone L, 2012, *A fast way of calculating exact hypervolumes*, IEEE Transactions on Evolutionary Computation, **16(1)**, pp. 86–95.

[230] 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.

[231] Winkelman AJM, 2015, Employee at *Reactor Institute Delft*, [Personal Communication], Contactable at `A.J.M.Winkelman@tudelft.nl`.

[232] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th Edition, Brooks/Cole, Canada.

[233] Wolpert DH & Macready WG, 1997, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation, **1(1)**, pp. 67–82.

[234] World Nuclear Association, 2015, *Safety of plants*, [Online], [Cited November 3rd, 2015], Available from `http://www.world-nuclear.org/info/Safety-and-Security/Safety-of-Plants/`.

[235] *XIMAGE/SIMAN-PWR — Nuclear Fuel Analysis Software*, 2016, Studsvik Scandpower, Inc., Waltham (MA).

[236] Yamamoto A, 2003, *Application of neural network for loading pattern screening of in-core optimization calculations*, Nuclear Technology, **144**, pp. 63–75.

[237] Yamamoto A & Hashimoto H, 2000, *Application of temperature parallel simulated annealing to loading pattern optimizations of pressurized water reactors*, Nuclear Science and Engineering, **136**, pp. 247–257.

[238] Yamamoto A & Hashimoto H, 2002, *Application of the distributed genetic algorithm for in-core fuel optimization problems under parallel computational environment*, Journal of Nuclear Science and Technology, **39(12)**, pp. 1281–1288.

[239] Yamamoto A, Noda H, Ito N & Maruyama T, 1997, *INSIGHT: An integrated scoping analysis tool for in-core fuel management of PWR*, Journal of Nuclear Science and Technology, **34(8)**, pp. 847–855.

[240] Yang XS, 2009, *Harmony search as a metaheuristic algorithm*, pp. 1–14 in Geem ZW (Ed), *Music-inspired harmony search algorithm: Theory and applications*, Springer, Berlin.

[241] Yu H & Wilamowski BM, 2011, *Levenberg-Marquardt training*, pp. 12-1–12-15 in Wilamowski BM & Irwin JD (Eds), *The industrial electronics handbook*, CRC Press, Boca Raton (FL).

[242] Zameer A, Mirza SM & Mirza NM, 2014, *Core loading pattern optimization of a typical two-loop 300 MWe PWR using simulated annealing (PWR), novel crossover genetic algorithms (GA and hybrid GA(SA) schemes)*, Annals of Nuclear Energy, **65**, pp. 122–131.

[243] Zavaljevski N, 1990, *A model for fuel shuffling and burnable absorbers optimization in low leakage PWRs*, Annals of Nuclear Energy, **17(4)**, pp. 217–220.

[244] Zhang Q & Li H, 2007, *MOEA/D: A multiobjective evolutionary algorithm based on decomposition*, IEEE Transactions on Evolutionary Computation, **11(6)**, pp. 712–731.

[245] Zhao J, Knight B, Nissan E & Soper A, 1998, *FuelGen: A genetic algorithm-based system for fuel loading pattern design in nuclear power reactors*, Expert Systems With Applications, **14**, pp. 461–470.

[246] Zhou A, Qu B, Li H, Zhao S, Suganthan PN & Zhang Q, 2011, *Multiobjective evolutionary algorithms: A survey of the state of the art*, Swarm and Evolutionary Computation, **1**, pp. 32–49.

[247] Zhou A, Zhang Q & Jin Y, 2009, *Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm*, IEEE Transactions on Evolutionary Computation, **13(5)**, pp. 1167–1189.

[248] Zitzler E, Laumanns M & Thiele L, 2001, *SPEA2: Improving the strength Pareto evolutionary algorithm*, (Unpublished) Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

[249] Zitzler E & Thiele L, 1998, *An evolutionary algorithm for multiobjective optimization: The strength Pareto approach*, (Unpublished) Technical Report 43, Computer Engineering and Communication Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

[250] Zitzler E & Thiele L, 1999, *Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach*, IEEE Transactions on Evolutionary Computation, **3(4)**, pp. 257–271.

[251] Zitzler E, Thiele L, Laumanns M, Fonseca CM & de Fonseca VG, 2003, *Performance assessment of multiobjective optimizers: An analysis and review*, IEEE Transactions on Evolutionary Computation, **7(2)**, pp. 117–132.

APPENDIX A

The backpropagation training algorithm

This appendix contains details concerning the backpropagation training algorithm for neural networks. In particular, a derivation of the backpropagation of errors method is presented within the context of a MFNN with one hidden layer, as illustrated in Figure 6.3. This derivation is a modified reproduction of the general derivation presented in [17]. Furthermore, the gradient-based optimisation technique employed in the training algorithm is also elaborated upon. This appendix should be read in conjunction with Chapter 6, with a particular emphasis on §6.3.1, as the terminologies and notations employed in this appendix were introduced there.

Suitable values for network weights are determined in the backpropagation training algorithm by minimising an appropriate error function, denoted by E. Suppose that individual errors E^o may be defined separately for each input vector $o \in \{1, \ldots, N\}$ in the training set, and that E may be expressed as a summation over all these individual errors, where N is the number of vectors in the set. Assume also that E^o is expressible as a differentiable function of the network outputs, that is $E^o = E^o(c_1, \ldots, c_n)$. The goal is to obtain a procedure (*i.e.* the backpropagation of errors method) according to which the derivatives of the error function E may be evaluated with respect to the network weights so as to use them in a gradient-based optimisation technique. Since $E = \sum_{o=1}^{N} E^o$, these derivatives may be expressed as summations over the set of input training vectors of the derivatives for each vector separately. Accordingly, only one input training vector has to be considered at a time.

A.1 The backpropagation of errors method

Suppose that input training vector o has been presented to the MFNN and transmitted over the network. The activations of all the hidden and output neurons have therefore been calculated using (6.1)–(6.4). This process is typically referred to as *forward propagation* because it may be regarded as information flow in a forward direction over the network.

Consider the derivative of E^o with respect to some weight $w_{kj}^{(2)}$ in the second layer. Although the activations of the neurons depend on vector o, the superscript o is omitted from the respective notations here so as to avoid clutter. It is noted that E^o depends on $w_{kj}^{(2)}$ only through the net input $\eta_k^{(2)}$ to output neuron k. Applying the chain rule for differentiation then yields

$$\frac{\partial E^o}{\partial w_{kj}^{(2)}} = \frac{\partial E^o}{\partial \eta_k^{(2)}} \frac{\partial \eta_k^{(2)}}{\partial w_{kj}^{(2)}}. \tag{A.1}$$

A useful notation introduced at this point is

$$\delta_k^{(2)} = \frac{\partial E^o}{\partial \eta_k^{(2)}}, \tag{A.2}$$

with the δ's typically being referred to as *errors*. Next, by using (6.3), the derivative of $\eta_k^{(2)}$ with respect to the network weight evaluates to

$$\frac{\partial \eta_k^{(2)}}{\partial w_{kj}^{(2)}} = b_j. \tag{A.3}$$

Substituting (A.2) and (A.3) into (A.1) yields

$$\frac{\partial E^o}{\partial w_{kj}^{(2)}} = \delta_k^{(2)} b_j. \tag{A.4}$$

Therefore, in order to evaluate the derivative of E^o with respect to a weight in the second layer of the MFNN, only the value of $\delta_k^{(2)}$ has to be calculated for each output neuron before applying (A.4). When the chain rule is applied once more to (A.2) in conjunction with the definition of c_k in (6.4), it is found that

$$\begin{aligned}
\delta_k^{(2)} &= \frac{\partial E^o}{\partial c_k} \frac{\partial c_k}{\partial \eta_k^{(2)}} \\
&= \frac{\partial E^o}{\partial c_k} \frac{\partial}{\partial \eta_k^{(2)}} \left(F_{(2)}\left(\eta_k^{(2)}\right)\right) \\
&= F'_{(2)}\left(\eta_k^{(2)}\right) \frac{\partial E^o}{\partial c_k}.
\end{aligned} \tag{A.5}$$

Consider next the derivative of E^o with respect to some weight $w_{ji}^{(1)}$ in the first layer of the MFNN. Application of the chain rule, using (6.1) and adopting the δ-notation introduced above, yields

$$\begin{aligned}
\frac{\partial E^o}{\partial w_{ji}^{(1)}} &= \frac{\partial E^o}{\partial \eta_j^{(1)}} \frac{\partial \eta_j^{(1)}}{\partial w_{ji}^{(1)}} \\
&= \delta_j^{(1)} a_i.
\end{aligned} \tag{A.6}$$

It is observed that (A.6) has the same general form as (A.4). Similarly, then, in order to evaluate the derivative of E^o with respect to a weight in the first layer of the MFNN, only the value of $\delta_j^{(1)}$ has to be calculated for each hidden neuron before applying (A.6).

The error function E^o may be considered as a function of all the activations $\eta_k^{(2)}$ which received input from hidden neuron j. In this case, $k = 1, \ldots, n$ such that $E^o = E^o\left(\eta_1^{(2)}, \ldots, \eta_n^{(2)}\right)$. Accordingly,

$$\delta_j^{(1)} = \frac{\partial E^o(\eta_1^{(2)}, \ldots, \eta_n^{(2)})}{\partial \eta_j^{(1)}}. \tag{A.7}$$

Applying the chain rule to (A.7) twice and substituting (A.2) into the expression yields

$$\begin{aligned}\delta_j^{(1)} &= \frac{\partial E^o}{\partial \eta_1^{(2)}}\frac{\partial \eta_1^{(2)}}{\partial \eta_j^{(1)}} + \cdots + \frac{\partial E^o}{\partial \eta_n^{(2)}}\frac{\partial \eta_n^{(2)}}{\partial \eta_j^{(1)}} \\ &= \sum_{k=1}^{n} \frac{\partial E^o}{\partial \eta_k^{(2)}}\frac{\partial \eta_k^{(2)}}{\partial \eta_j^{(1)}} \\ &= \sum_{k=1}^{n} \delta_k^{(2)}\frac{\partial \eta_k^{(2)}}{\partial b_j}\frac{\partial b_j}{\partial \eta_j^{(1)}}. \end{aligned} \tag{A.8}$$

The final expression for $\delta_j^{(1)}$ may now be obtained using (6.3) and the definition of b_j in (6.2) to evaluate (A.8), thus yielding

$$\delta_j^{(1)} = F'_{(1)}\left(\eta_j^{(1)}\right)\sum_{k=1}^{n} w_{kj}^{(2)}\delta_k^{(2)}. \tag{A.9}$$

This so-called *backpropagation formula* generalises to MFNNs with any number of hidden layers — the value of δ for some specific hidden neuron may be obtained by propagating the δ-values backwards over the network from neurons in succeeding layers [17].

In summary, the backpropagation of errors method for evaluating the derivatives of E^o with respect to the network weights in an MFNN with one hidden layer is as follows:

1. Present input training vector $o \in \{1, \ldots, N\}$ to the network and forward propagate over the network using (6.1)–(6.4) to calculate all the activations in the neurons.
2. Evaluate $\delta_k^{(2)}$ for each output neuron using (A.5).
3. Backpropagate the $\delta_k^{(2)}$-values over the network and evaluate $\delta_j^{(1)}$ for each hidden neuron using (A.9).
4. Calculate the derivatives of the error E^o using (A.4) and (A.6).

Since it has been assumed that $E = \sum_{o=1}^{N} E^o$, the derivative of the total error E with respect to the network weights may be obtained by repeating the aforementioned four steps for each input vector in the training set, after which

$$\frac{\partial E}{\partial w_{rs}} = \sum_{o=1}^{N}\frac{\partial E^o}{\partial w_{rs}} \tag{A.10}$$

may be calculated.

A.2 The gradient-based optimisation technique

In order to complete the description of the backpropagation training algorithm, a method for updating the weights based on these error function derivatives has to be specified. Although different gradient-based optimisation techniques may be considered [17], the simplest is to employ the *gradient descent method* for minimisation. According to this method, a decision vector $\boldsymbol{x}$ is iteratively updated according to

$$\boldsymbol{x}_{\text{new}} = \boldsymbol{x}_{\text{old}} - \kappa\nabla\Psi(\boldsymbol{x}_{\text{old}}), \tag{A.11}$$

where $\Psi(\boldsymbol{x})$ is the function to be minimised, and κ is a step size. The method is based on the observation that a differentiable function $\Psi(\boldsymbol{x})$ decreases the "fastest" about a point $\boldsymbol{u}$ if one moves in the direction of the negative gradient of Ψ at point $\boldsymbol{u}$, thus in the direction $-\nabla\Psi(\boldsymbol{u})$. Applying (A.11) in the context of MFNN training yields a rule for adjusting the network weights such that the error function is minimised.

The rule, then, for adjusting the MFNN weights in the second layer within an online learning paradigm is given by

$$w_{kj}^{(2)} = w_{kj}^{(2)} - \varepsilon \frac{\partial E^o}{\partial w_{kj}^{(2)}} \tag{A.12}$$

$$= w_{kj}^{(2)} - \varepsilon \delta_k^{(2)} b_j. \tag{A.13}$$

Similarly, the rule for adjusting the weights in the first layer within an online learning paradigm is given by

$$w_{ji}^{(1)} = w_{ji}^{(1)} - \varepsilon \frac{\partial E^o}{\partial w_{ji}^{(1)}} \tag{A.14}$$

$$= w_{ji}^{(1)} - \varepsilon \delta_j^{(1)} a_i. \tag{A.15}$$

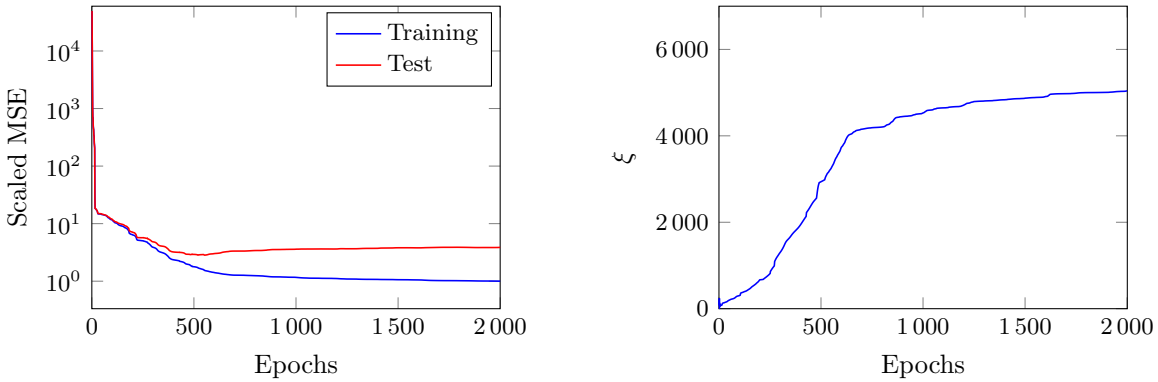
During batch learning, the derivative of error E should be used in (A.12) and (A.14) instead of the derivative of E^o only. This, in turn, will mean that a summation over the input vectors in the training set has to be employed in (A.13) and (A.15), as shown in [17].

APPENDIX B

Graphical results obtained during neural network training

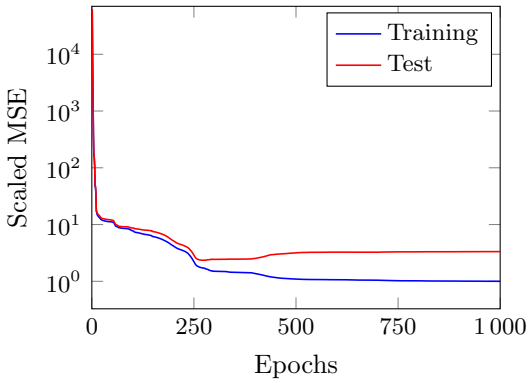
This appendix contains the graphical results obtained during the training of the MFNNs described in Chapter 6, but which were not presented in §6.6 so as to enhance the flow of exposition in the main text. The notation employed in this appendix conforms to the definitions presented in Table 6.1 for each neural network.

In Figures B.1–B.9, the convergence graphs of the training process for networks ϕ_{B12}, ϕ_{Si}, ϕ_{I1}, ϕ_{I2}, $\psi_{\text{Mo}}^{\text{tot}}$, ρ_{cbw}, ρ_{sdm}, ρ_{ex} and ρ_{ppf}, respectively, are presented. Similarly, the graphical results pertaining to the predictive capabilities of networks ϕ_{B12}, ϕ_{Si}, ϕ_{I1}, ϕ_{I2}, $\psi_{\text{Mo}}^{\text{tot}}$, ρ_{cbw}, ρ_{sdm} and ρ_{ex} are presented in Figure B.10. For each network, a scatter graph of the 20 000 target values in the combined training and test sets versus their predicted values is presented in the figure. The straight line in each graph, labelled "Y=Target" in the legend, corresponds to the theoretical case in which perfect predictions are achieved.

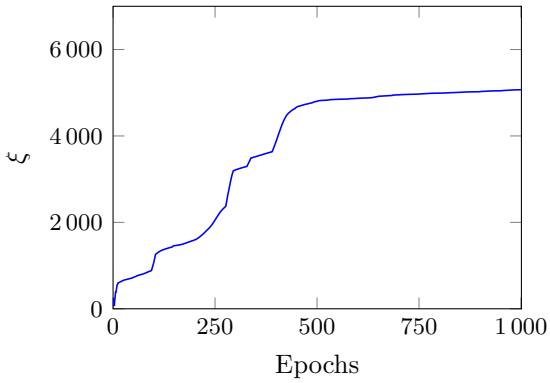


(a) *Scaled MSE of the training and test sets* (b) *The effective number of parameters*

FIGURE B.1: *Convergence results for the ϕ_{B12} neural network.*

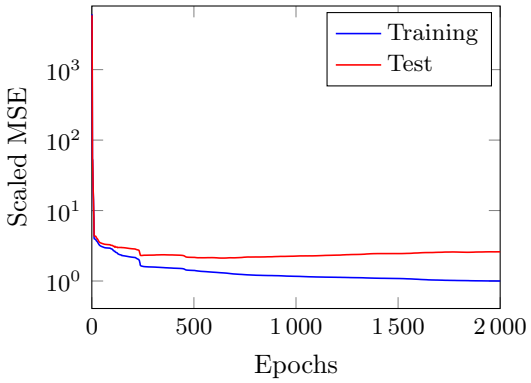


(a) *Scaled MSE of the training and test sets*

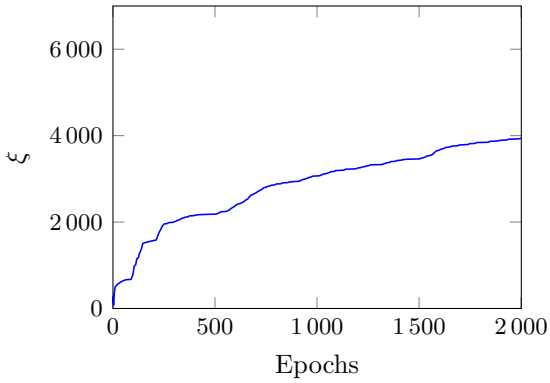


(b) *The effective number of parameters*

FIGURE B.2: *Convergence results for the ϕ_{Si} neural network.*

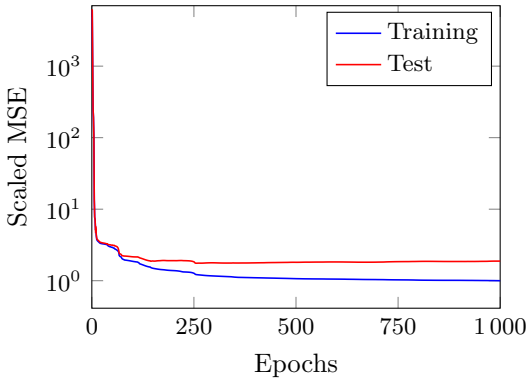


(a) *Scaled MSE of the training and test sets*

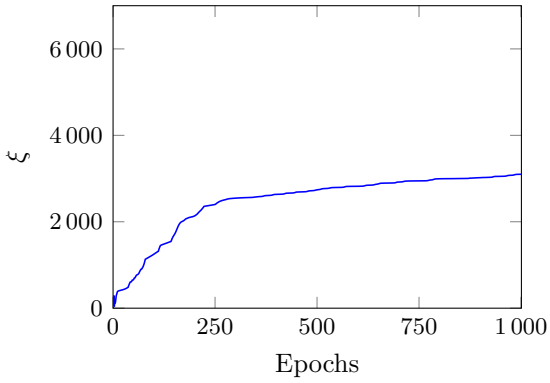


(b) *The effective number of parameters*

FIGURE B.3: *Convergence results for the ϕ_{I1} neural network.*



(a) *Scaled MSE of the training and test sets*

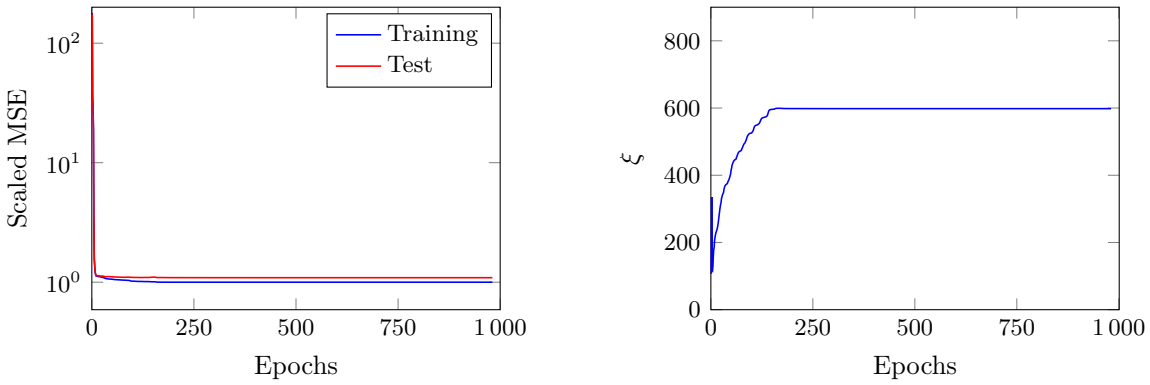


(b) *The effective number of parameters*

FIGURE B.4: *Convergence results for the ϕ_{I2} neural network.*

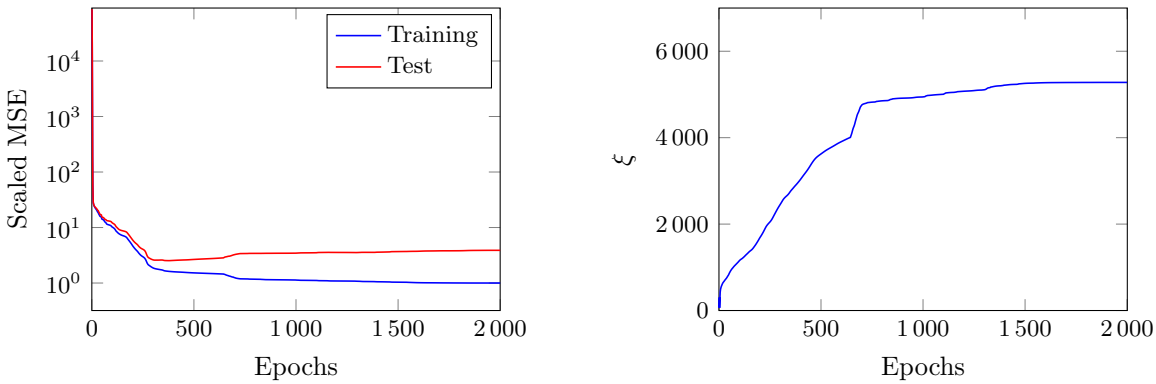
Upon inspection of Figures B.3(b) and B.4(b), it appears that the ϕ_{I1} and ϕ_{I2} networks, respectively, have not converged. Their corresponding scaled MSE values in Figures B.3(a) and B.4(a), however, remain fairly constant for several epochs. Since both networks produced very good predictions, with maximum and average absolute relative errors of less than 1% for the training and test sets (see Table 6.2), additional training was deemed unnecessary.

In Figure B.5, it is observed that network $\psi_{\text{Mo}}^{\text{tot}}$ converges after 981 epochs according to the default stopping criteria of the Toolbox. This is similar to what occured during the training of the $\psi_{\text{Mo}}^{\text{min}}$ network, as shown in Figure 6.6.



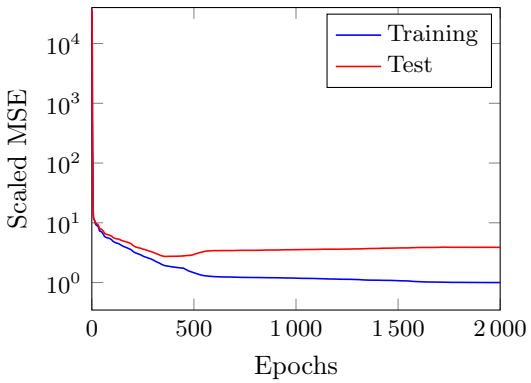
(a) *Scaled MSE of the training and test sets* (b) *The effective number of parameters*

FIGURE B.5: *Convergence results for the ψ_{Mo}^{tot} neural network.*

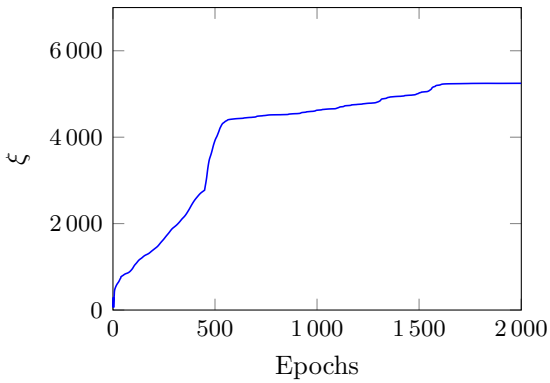


(a) *Scaled MSE of the training and test sets* (b) *The effective number of parameters*

FIGURE B.6: *Convergence results for the ρ_{cbw} neural network.*

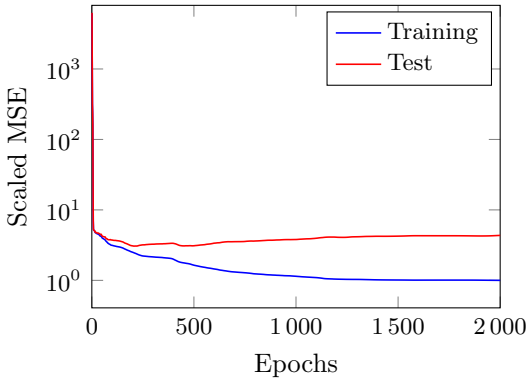


(a) *Scaled MSE of the training and test sets*

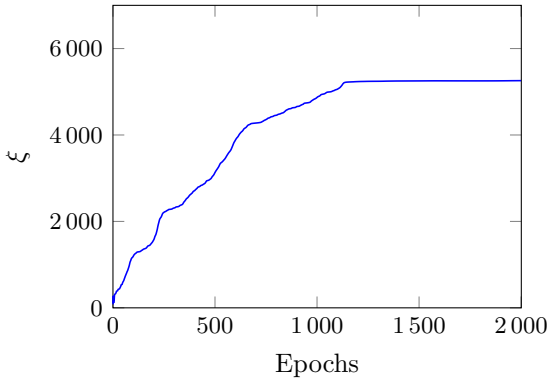


(b) *The effective number of parameters*

FIGURE B.7: *Convergence results for the ρ_{sdm} neural network.*

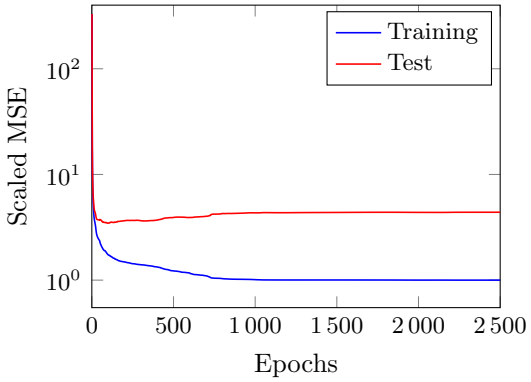


(a) *Scaled MSE of the training and test sets*

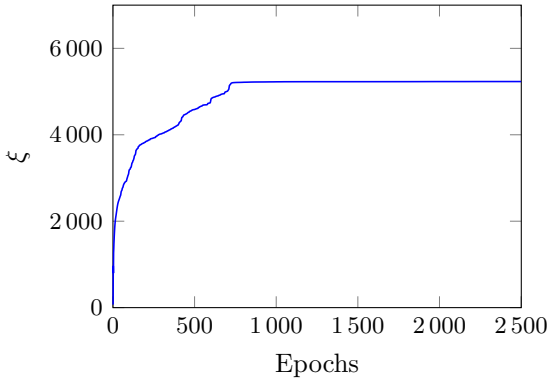


(b) *The effective number of parameters*

FIGURE B.8: *Convergence results for the ρ_{ex} neural network.*

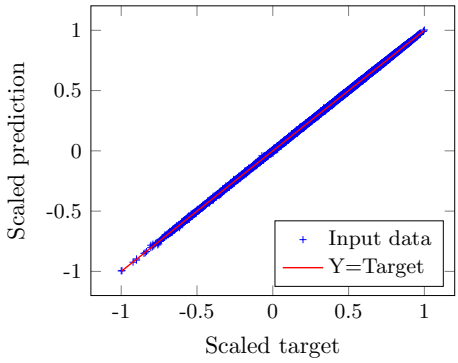


(a) *Scaled MSE of the training and test sets*

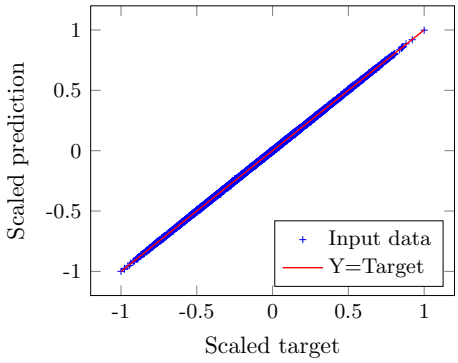


(b) *The effective number of parameters*

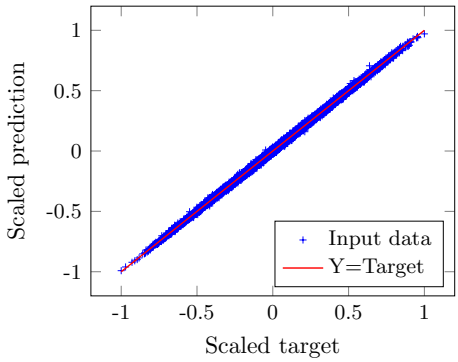
FIGURE B.9: *Convergence results for the ψ_{ppf} neural network.*



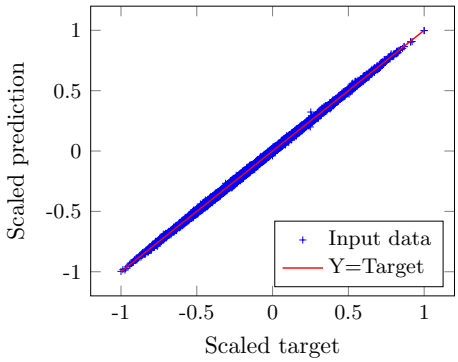
(a) *The ϕ_{B12} neural network*



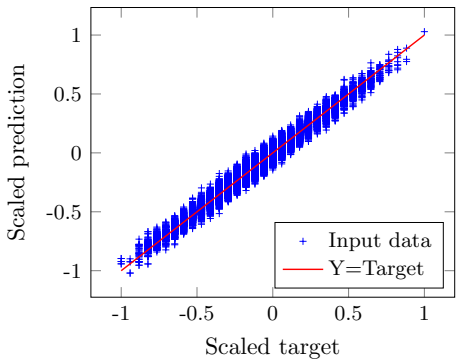
(b) *The ϕ_{Si} neural network*



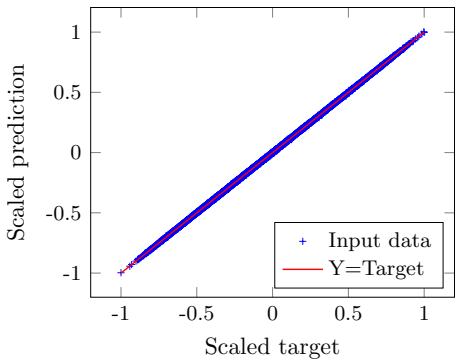
(c) *The ϕ_{I1} neural network*



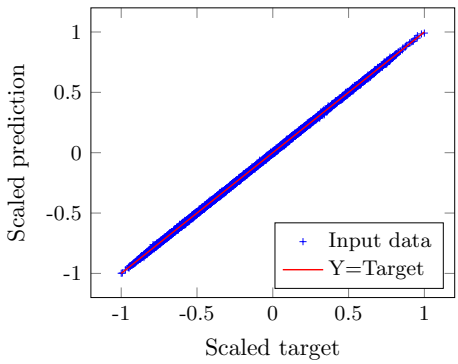
(d) *The ϕ_{I2} neural network*



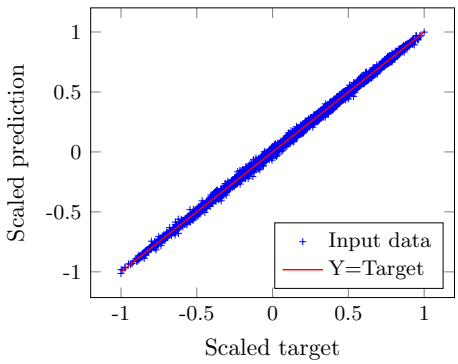
(e) *The ψ_{Mo}^{tot} neural network*



(f) *The ρ_{cbw} neural network*



(g) *The ρ_{sdm} neural network*



(h) *The ρ_{ex} neural network*

FIGURE B.10: *Prediction quality results for the ϕ_{B12}, ϕ_{Si}, ϕ_{I1}, ϕ_{I2}, ψ_{Mo}^{tot}, ρ_{cbw}, ρ_{sdm} and ρ_{ex} neural networks.*

APPENDIX C

Additional information on multiobjective metaheuristics

This appendix contains additional information on the multiobjective metaheuristics employed for solving the MICFMO problem described in Chapter 7. The content of the appendix was not presented in that chapter so as to enhance the exposition of the main text. The notation employed in this appendix conforms to the definitions presented in Chapter 7. Finally, unless specifically stated otherwise, where any reference is made to *random* selection, it is assumed that a uniform distribution is employed during selection.

C.1 Permutation-based MOEA variation operators

In this section, each of the permutation-based variation operators introduced in §7.3, namely the PMX, POS and CX crossover operators, as well as the swap and scramble mutation operators, is described and illustrated by means of an example. Note that the crossover operator examples have been reproduced from Larrañaga *et al.* [113].

The PMX operator

According to the PMX operator, two cut points are selected at random along the parent solutions. Consider, for example, the following two parent solutions, with cut points that have been selected between the third and fourth components, as well as between the sixth and seventh components:

Parent 1: [1 2 3 | 4 5 6 | 7 8],
Parent 2: [3 7 5 | 1 6 8 | 2 4].

In order to create offspring solutions, the substring of components between the two cutpoints in the first parent is copied into the second offspring, and likewise for the second parent and first offspring:

Offspring 1: [∗ ∗ ∗ | 1 6 8 | ∗ ∗],
Offspring 2: [∗ ∗ ∗ | 4 5 6 | ∗ ∗].

These substrings are called *mapping sections* because, in this example, they define the mappings $4 \leftrightarrow 1$, $5 \leftrightarrow 6$ and $6 \leftrightarrow 8$. The remaining components in each offspring $i \in \{1, 2\}$ are now filled by copying the components of the i-th parent into it. Whenever a value is already present in the offspring, however, the mappings are employed to replace them.

In the example, then, the first component in offspring 1 would have taken a value of 1 because it is copied from the first parent. Since the value 1 is already present in the fourth component of offspring 1, however, the mapping $4 \leftrightarrow 1$ is employed to replace it. Therefore, the first component in offspring 1 now takes a value of 4. The second, third and seventh components of offspring 1 may take the values directly from the first parent. The last component, however, would have taken a value of 8, but it already appears in the sixth component of the offspring. Using the mappings $5 \leftrightarrow 6$ and $6 \leftrightarrow 8$, the component now takes a value of 5. Following the same procedure, offspring 2 may also be filled. The new offspring solutions are then

$$\begin{array}{ll} \text{Offspring 1:} & [4\ 2\ 3 \mid 1\ 6\ 8 \mid 7\ 5], \\ \text{Offspring 2:} & [3\ 7\ 8 \mid 4\ 5\ 6 \mid 2\ 1]. \end{array}$$

The POS operator

According to the POS operator, a subset of components in the parent solutions is selected at random. Consider, for example, the following two parents in which the second, third and sixth components have been selected randomly:

$$\begin{array}{ll} & \quad\ \downarrow\downarrow\ \ \downarrow \\ \text{Parent 1:} & [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8], \\ \text{Parent 2:} & [2\ 4\ 6\ 8\ 7\ 5\ 3\ 1]. \end{array}$$

In order to create offspring solutions, the subset of components in the first parent is copied into the second offspring, while the subset in the second parent is copied into the first offspring:

$$\begin{array}{ll} \text{Offspring 1:} & [*\ 4\ 6\ *\ *\ 5\ *\ *], \\ \text{Offspring 2:} & [*\ 2\ 3\ *\ *\ 6\ *\ *]. \end{array}$$

The remaining components in each offspring $i \in \{1, 2\}$ are now filled from left to right by adding the missing values of the i-th parent in the same order that they appear in the parent. Therefore, the new offspring solutions are

$$\begin{array}{ll} \text{Offspring 1:} & [1\ 4\ 6\ 2\ 3\ 5\ 7\ 8], \\ \text{Offspring 2:} & [4\ 2\ 3\ 8\ 7\ 6\ 5\ 1]. \end{array}$$

The CX operator

In the CX operator, so-called *cycles* are identified from the parents and their values are copied to the offspring. The procedure to create a cycle from parent 1, for example, is as follows:

1. Start the cycle with the first component of parent 1;
2. Observe which value is in the corresponding component in parent 2;

3. Go to the component in parent 1 having this observed value and append it to the cycle;

4. Repeat steps 2 and 3 until the first component in the cycle is reached again.

This cycle from the first parent is then copied into offspring 1. The next cycle is identified from the second parent, now excluding the components used in the previous cycle, and copied into offspring 1 as before. This alternating cycle identification is continued until the offspring is fully constructed. Similarly, offspring 2 may be constructed by repeating the above-mentioned procedure, but starting from parent 2. A graphical example of the CX operator is presented in Figure C.1.

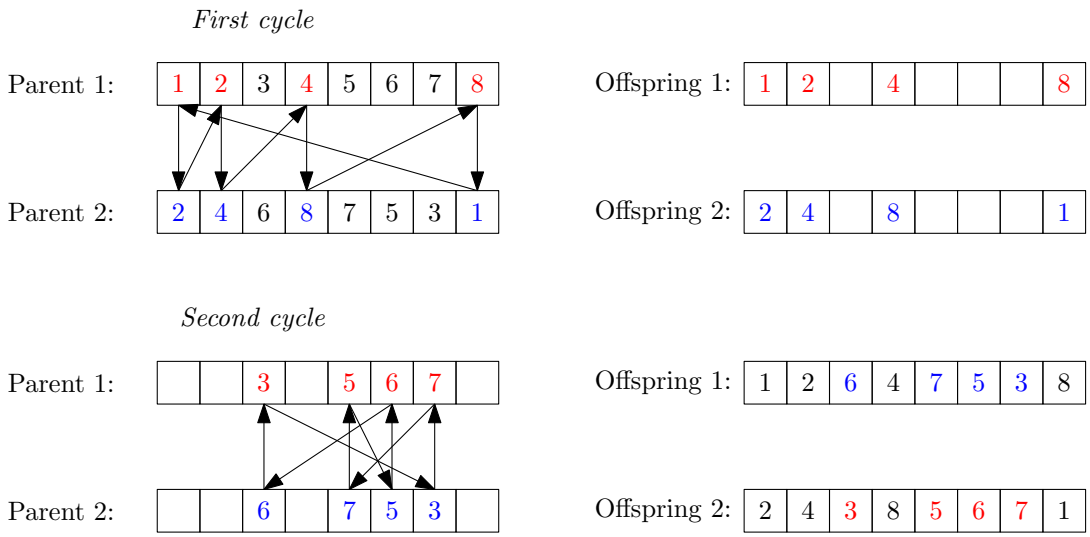


FIGURE C.1: *Example of the cycle crossover operator.*

The swap mutation operator

According to the swap operator, two components in a solution are selected at random, and their corresponding values are simply exchanged to form a new solution. In the example, the third and seventh components have been selected for this binary exchange:

$$[1\ 2\ \underline{3}\ 4\ 5\ 6\ \underline{7}\ 8] \quad \Longrightarrow \quad [1\ 2\ \underline{7}\ 4\ 5\ 6\ \underline{3}\ 8].$$

The scramble mutation operator

In the scramble operator, a subset of components in the solution is selected at random. Then, the values in those components are randomly permuted (*i.e.* rearranged) to form a new solution. The subset does not have to be contiguous. In the following example, four components have been selected and rearranged:

$$[1\ \underline{2\ 3\ 4}\ 5\ 6\ 7\ \underline{8}] \quad \Longrightarrow \quad [1\ \underline{3\ 8\ 2}\ 5\ 6\ 7\ \underline{4}].$$

C.2 Permutation-based approaches within PSO algorithms

In this section, the two permutation-based approaches introduced in §7.4.1 for modifying the OMOPSO algorithm are described. These approaches allow the algorithm to be applicable to the MICFMO problem.

The method of random keys

According to the method of random keys [13], a solution (or in this case, the position of a particle) is encoded by random numbers, and these values are then used as sort keys to decode the real-valued vector into a permutation.

As an example, consider a particle whose position has been encoded as $\boldsymbol{x}_i^t = [0.43, 0.86, 0.23, 0.79, 0.61]$. Sorting the values in increasing order of magnitude, and noting their original indices, yields the decoded permutation vector $\boldsymbol{x} = [3, 1, 5, 4, 2]$, as illustrated in Figure C.2.

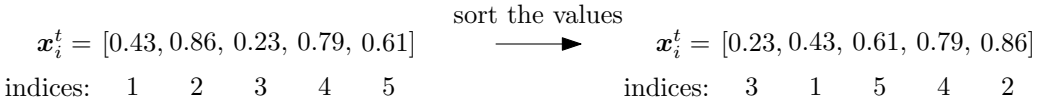


FIGURE C.2: *Example of the random keys method.*

By employing the method of random keys, the flight operators in OMOPSO may be applied, unaltered, to the encoded particle positions. Accordingly, the position of a particle may then be decoded into a solution for use in the remaining parts of the algorithm.

The approach proposed by Hu *et al.* [86]

In the approach proposed by Hu *et al.* [86], the position associated with a particle is explicitly represented as a permutation, while the position flight operator is redefined using a probabilistic interpretation of the velocity.

According to the approach, the velocity flight operator remains the same, although the velocity vector components are now limited to absolute values. In the redefined position flight operator, the value in each velocity component is normalised to the range [0,1] by dividing it by the maximum value attained in the vector. Each of these values then represents the probability with which a binary exchange should occur in the corresponding component of the particle's position. If an exchange is required, the affected component takes the value of the corresponding component in the global best position. This process is illustrated by means of an example in Figure C.3.

A shortcoming of this approach is that a particle having the same position as the global best position would never change. Accordingly, the swap mutation operator, borrowed from the evolutionary computation literature, is applied to a particle whenever this is the case [86].

C.3 Derivation of the MICFMO updating rule in the MOOCEM

In order to apply the MOOCEM to the MICFMO problem, a parameterised probability distribution has to be specified and the stochastic program (7.17) has to be solved in order to

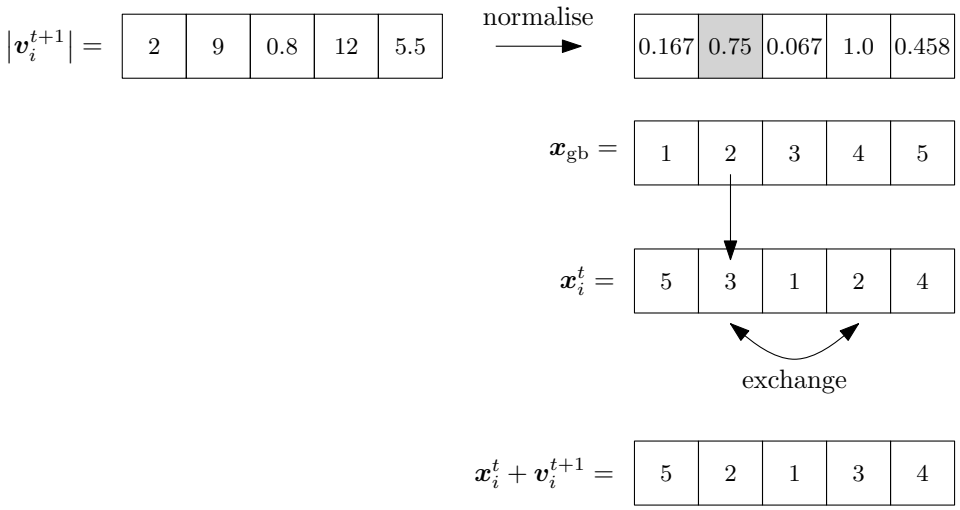


FIGURE C.3: *Example of the permutation-based position flight operator proposed by Hu et al. [86].*

obtain/derive an updating rule for the parameter vector. In this section, a derivation of this updating rule for MICFMO (shown in (7.20)) is presented. This derivation is very similar to those for the TSP in [30] and the BAP in [2].

Consider the unconstrained version of the MICFMO problem (4.1), given by

$$\left.\begin{aligned} \text{maximise} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x})], \\ & \boldsymbol{x} \in \mathcal{X}, \end{aligned}\right\} \tag{C.1}$$

where $\mathcal{X}$, as before, denotes the set of all valid reload configurations (*i.e.* permutations decision vectors of length n). Accordingly, the cardinality of the decision space is $|\mathcal{X}| = n!$.

The first step in the derivation is to relate (C.1) to an equivalent optimisation problem. Let $\widetilde{\mathcal{X}}$ be the set of decision vectors corresponding to reload configurations in which the permutation requirement of $\mathcal{X}$ has been relaxed (*i.e.* the same fuel assembly may be assigned to multiple loading positions). The cardinality of the set is therefore $|\widetilde{\mathcal{X}}| = n^n$. It is important to note that $\mathcal{X} \subset \widetilde{\mathcal{X}}$. Define, also, the objective function $\widetilde{f}_k(\boldsymbol{x})$ on $\widetilde{\mathcal{X}}$ for $k = 1, \ldots, q$ as

$$\widetilde{f}_k(\boldsymbol{x}) = \begin{cases} f_k(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \mathcal{X}, \\ -\infty & \text{otherwise,} \end{cases}$$

and let $\widetilde{\boldsymbol{f}}(\boldsymbol{x})$ denote the corresponding objective vector $[\widetilde{f}_1(\boldsymbol{x}), \ldots, \widetilde{f}_q(\boldsymbol{x})]$. The MICFMO problem (C.1) is therefore equivalent to the optimisation problem

$$\left.\begin{aligned} \text{maximise} \quad & \widetilde{\boldsymbol{f}}(\boldsymbol{x}), \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{X}. \end{aligned}\right\} \tag{C.2}$$

A simple procedure for generating a random decision vector $\boldsymbol{x} = [x_1, \ldots, x_n] \in \widetilde{\mathcal{X}}$ is to sample each x_i independently according to a fixed distribution $\boldsymbol{p}_i = [p_{i1}, \ldots, p_{in}]$ for $i = 1, \ldots, n$, where the component p_{ij} corresponds to the probability of assigning fuel assembly $j \in \{1, \ldots, n\}$ into loading position i. These distributions $\boldsymbol{p}_i$ may be combined into an $n \times n$ probability matrix $\boldsymbol{P} = [p_{ij}]$, with $p_{ij} > 0$ and

$$\sum_{j=1}^{n} p_{ij} = 1, \qquad \text{for } i = 1, \ldots, n. \tag{C.3}$$

Let $\widetilde{\mathcal{X}}_{ij}$ be the set of all reload configurations $\boldsymbol{x} \in \widetilde{\mathcal{X}}$ for which $x_i = j$, and define the indicator function

$$I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}} = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \widetilde{\mathcal{X}}_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The probability mass function $p(\cdot;\boldsymbol{P})$ of $\boldsymbol{x} \in \widetilde{\mathcal{X}}$, parameterised by the matrix $\boldsymbol{P}$, is then given by

$$p(\boldsymbol{x};\boldsymbol{P}) = \prod_{i=1}^{n}\prod_{j=1}^{n}(p_{ij})^{I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}}. \tag{C.4}$$

In order to update the parameter matrix $\boldsymbol{P}$ for the equivalent optimisation problem (C.2), the stochastic program (7.17) has to be solved, where the Pareto ranks in the indicator function (7.18) are now determined according to the objective vector $\widetilde{\boldsymbol{f}}(\boldsymbol{x})$ instead of $\boldsymbol{f}(\boldsymbol{x})$. Furthermore, (7.18) has to be solved under the additional constraint set (C.3). Since the stochastic program is a maximisation problem with equality constraints, the *method of Lagrange multipliers* may be employed to solve it. Let $\lambda_1, \ldots, \lambda_n$ denote the Lagrange multipliers for the n equality constraints in (C.3). The Lagrangian function is given by

$$\mathcal{L}(\boldsymbol{P}, \lambda_1, \ldots, \lambda_n) = \frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \le \rho_E\}} \ln p(\boldsymbol{x}_k;\boldsymbol{P}) + \sum_{i=1}^{n}\lambda_i\left(\sum_{j=1}^{n} p_{ij} - 1\right). \tag{C.5}$$

The natural logarithm of $p(\boldsymbol{x}_k;\boldsymbol{P})$ required in (C.5) is given by

$$\begin{aligned} \ln p(\boldsymbol{x}_k;\boldsymbol{P}) &= \ln\left(\prod_{i=1}^{n}\prod_{j=1}^{n}(p_{ij})^{I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}}\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\ln(p_{ij})^{I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}\ln p_{ij}. \end{aligned} \tag{C.6}$$

Substituting (C.6) into (C.5) yields

$$\mathcal{L}(\boldsymbol{P}, \lambda_1, \ldots, \lambda_n) = \frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \le \rho_E\}}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}\ln p_{ij}\right) + \sum_{i=1}^{n}\lambda_i\left(\sum_{j=1}^{n} p_{ij} - 1\right). \tag{C.7}$$

Next, by differentiating $\mathcal{L}$ in (C.7) with respect to an arbitrary p_{ij}, and setting the result equal to zero, it is found that

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial p_{ij}} &= \frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \le \rho_E\}}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}\frac{\partial}{\partial p_{ij}}\ln p_{ij}\right) + \sum_{i=1}^{n}\lambda_i\left(\sum_{j=1}^{n}\frac{\partial}{\partial p_{ij}}p_{ij} - 0\right) \\ &= \frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \le \rho_E\}}\left(I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}\frac{1}{p_{ij}}\right) + \lambda_i \\ &= 0, \end{aligned}$$

and therefore

$$\frac{1}{|\boldsymbol{R}_t|}\sum_{k=1}^{|\boldsymbol{R}_t|} I_{\{\rho_k \leq \rho_E\}} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}} = -\lambda_i p_{ij}. \tag{C.8}$$

Summing over all possible $j = 1, \ldots, n$ in (C.8) yields

$$\frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} \sum_{j=1}^{n} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}} = -\lambda_i \sum_{j=1}^{n} p_{ij}. \tag{C.9}$$

and by substituting (C.3) into (C.9), it follows, after simplification, that

$$\frac{1}{|\mathcal{R}_t|}\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} = -\lambda_i. \tag{C.10}$$

Finally, by substituting (C.10) into (C.8), the updating rule

$$p_{ij} = \frac{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} I_{\{\boldsymbol{x}\in\widetilde{\mathcal{X}}_{ij}\}}}{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}}} \tag{C.11}$$

is obtained. The interpretation of this updating rule is as follows. In order to update the probability p_{ij}, count the number of decision vectors in which fuel assembly j is assigned to loading position i, within the set of those decision vectors whose Pareto ranks do not exceed ρ_E, and divide it by the total number of decision vectors whose Pareto ranks are at most ρ_E.

The generation of solutions and the parameter updating for the equivalent problem (C.2) may now be performed as follows. For each decision variable x_i, sample its value independently from the i-th row of the probability matrix $\boldsymbol{P}$ until a full solution is generated. Once a set of solutions has been generated and combined with the elite set, apply the updating rule (C.11) to this combined set in order determine the new probabilities in $\boldsymbol{P}$.

The majority of solutions generated in the aforementioned manner would, however, not be useful in the context of the MICFMO problem, because their components would not form a permutation (*i.e.* a valid reload configuration). Accordingly, their objective function values $\widetilde{\boldsymbol{f}}$ would be $-\boldsymbol{\infty}$. The generation of such undesirable solutions may, however, be avoided by employing an alternative generation procedure in which permutations are created explicitly. This may be achieved by employing Algorithm 7.10 to generate solutions.

It should, however, be noted that the solutions generated by Algorithm 7.10 do not follow the exact same probability distribution as those generated according to the independent, variable-by-variable sampling procedure. The algorithm is simply employed as a heuristic to speed up the solution generation process during optimisation [108]. It is, however, assumed that Algorithm 7.10 generates solutions according to $p(\cdot; \boldsymbol{P})$. As such, the updating rule (C.11) is unaffected by the algorithm and, since only permutations are generated, the solutions correspond to valid reload configurations, *i.e.* $\boldsymbol{x} \in \mathcal{X}$. The updating rule for the probability p_{ij} then becomes

$$p_{ij} = \frac{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}} I_{\{\boldsymbol{x}\in\mathcal{X}_{ij}\}}}{\sum_{k=1}^{|\mathcal{R}_t|} I_{\{\rho_k \leq \rho_E\}}}, \tag{C.12}$$

where $\mathcal{X}_{ij}$ is the set of all reload configurations $\boldsymbol{x} \in \mathcal{X}$ for which $x_i = j$, and $I_{\{\boldsymbol{x}\in\mathcal{X}_{ij}\}}$ is the corresponding indicator function.

APPENDIX D

Additional MICFMO experimental results

This appendix contains additional results obtained during the comparative studies described in Chapter 8 between two constraint handling techniques and eight multiobjective metaheuristics, but which were not presented in §8.4 so as to enhance the exposition of the main text.

D.1 Constraint handling technique comparison

In Figures D.1–D.5, box plots are presented for the converted ΔI_{HVD} and ΔI_{R2} samples, obtained for each problem instance in the test suite of §8.1, using the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms, respectively. The average values of the samples are also included in the graphs as black diamond points. Negative values in these converted samples correspond to superior performance by the MPF technique, whereas positive values correspond to superior performance by the CDP technique.

The two-tailed Wilcoxon signed rank test was applied to the samples obtained for each test problem instance (*i.e.* in the context of a single-problem analysis) in order to determine whether there are statistically significant differences between the MPF and CDP constraint handling techniques. The resulting p-values are presented in Tables D.1–D.5 for the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms, respectively. Bold-faced entries in a table represent a statistically significant difference (for $\tilde{\alpha} = 0.05$). If a significant difference is detected, the box plots in Figures D.1–D.5 may be referred to in order to pronounce on which constraint handling technique outperformed the other.

A multi-problem analysis, *within each class* of problem instances, was also conducted using the results obtained by each metaheuristic. In these analyses, an average indicator value per metaheuristic variant/problem instance pair constitutes the samples. The converted average indicator value samples, $\overline{\Delta I}_{\text{HVD}}$ and $\overline{\Delta I}_{R2}$, obtained by the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms within each problem instance class, are presented in Table D.6. The sample values correspond to the diamond points in Figures D.1–D.5. The two-tailed Wilcoxon signed rank test was applied to the average samples obtained for each class of problem instances. The resulting p-values are presented in Table D.7 and, as before, bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).

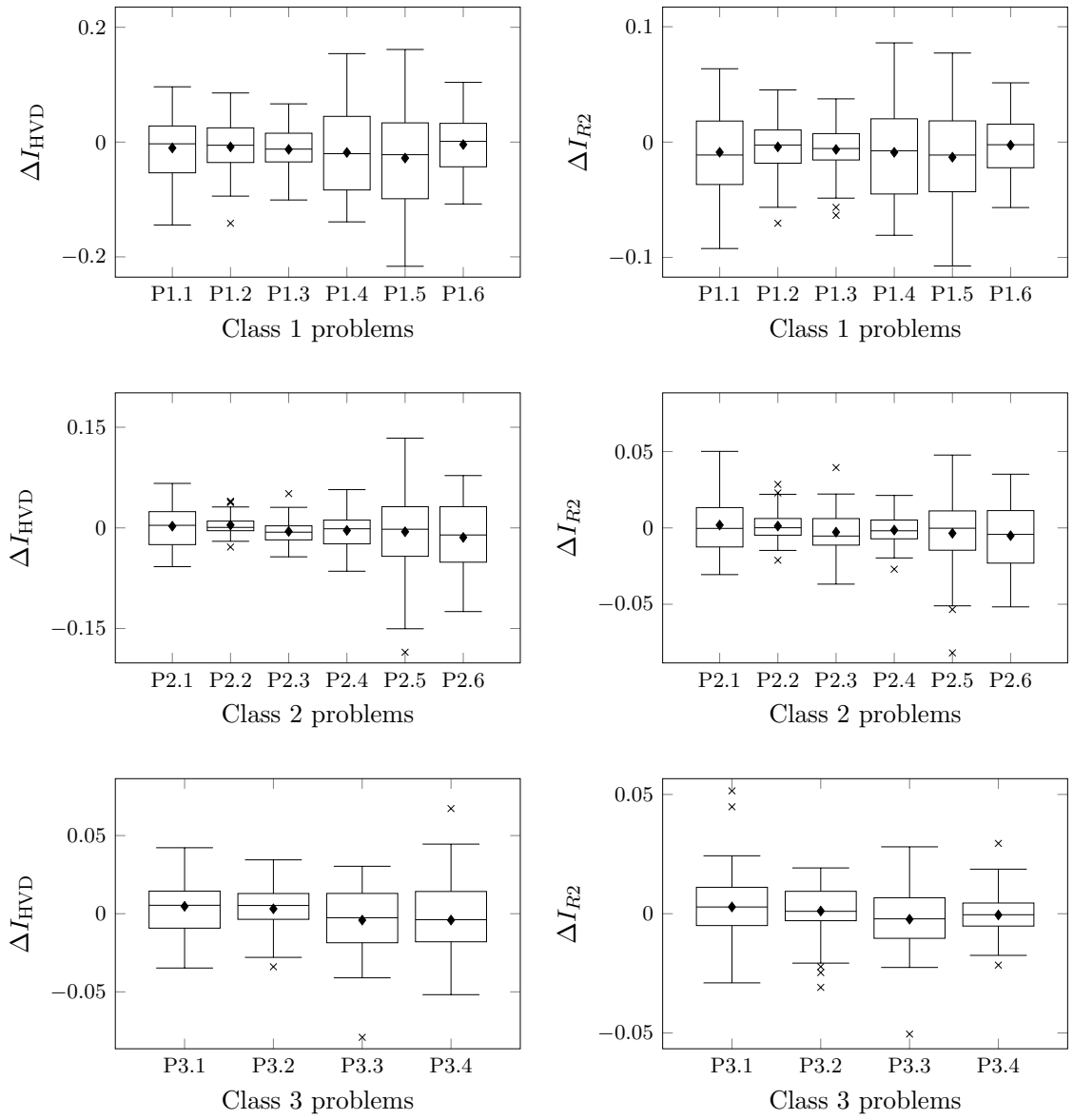


FIGURE D.1: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the SPEA2.*

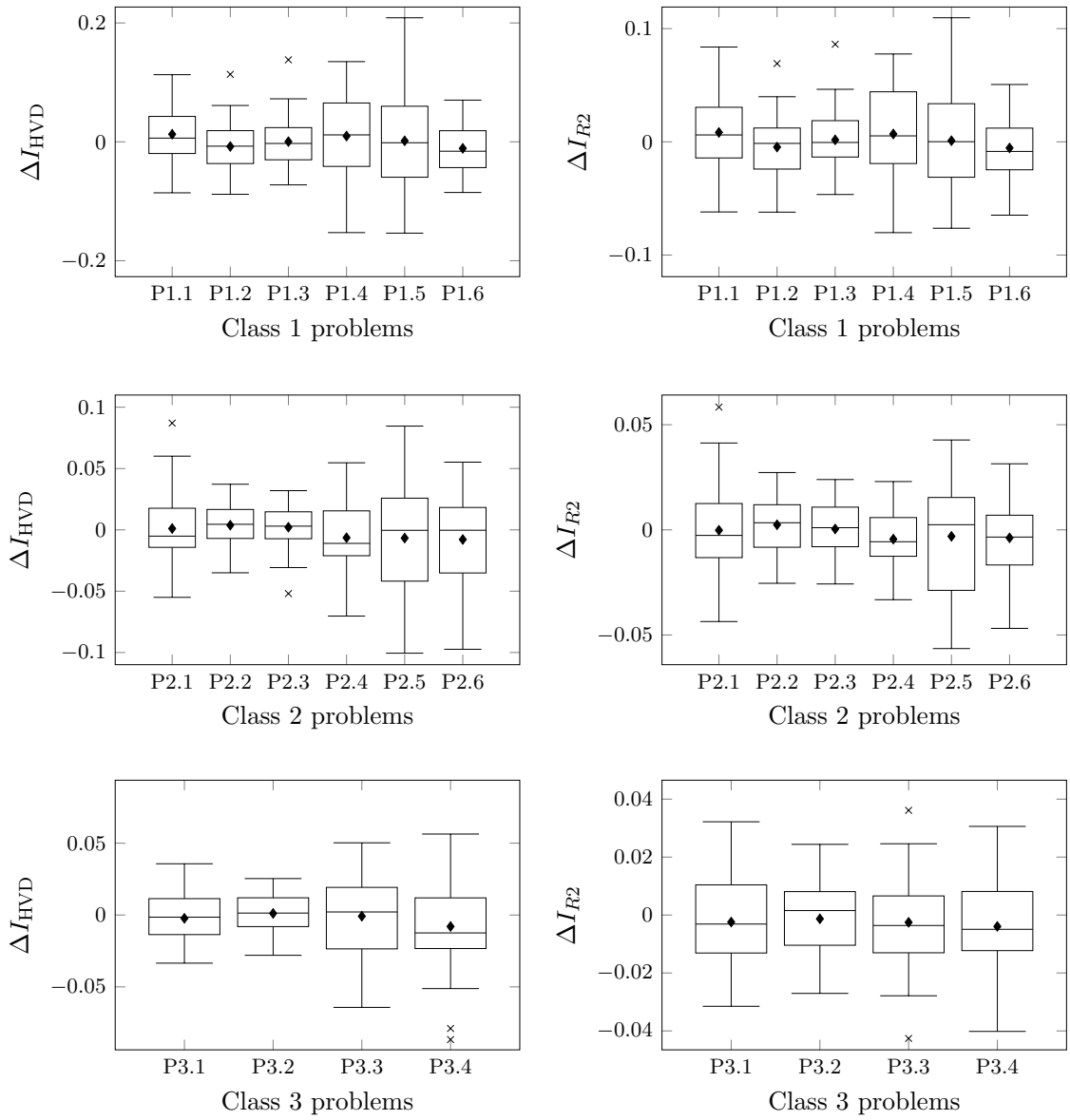


FIGURE D.2: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the OMOPSO algorithm.*

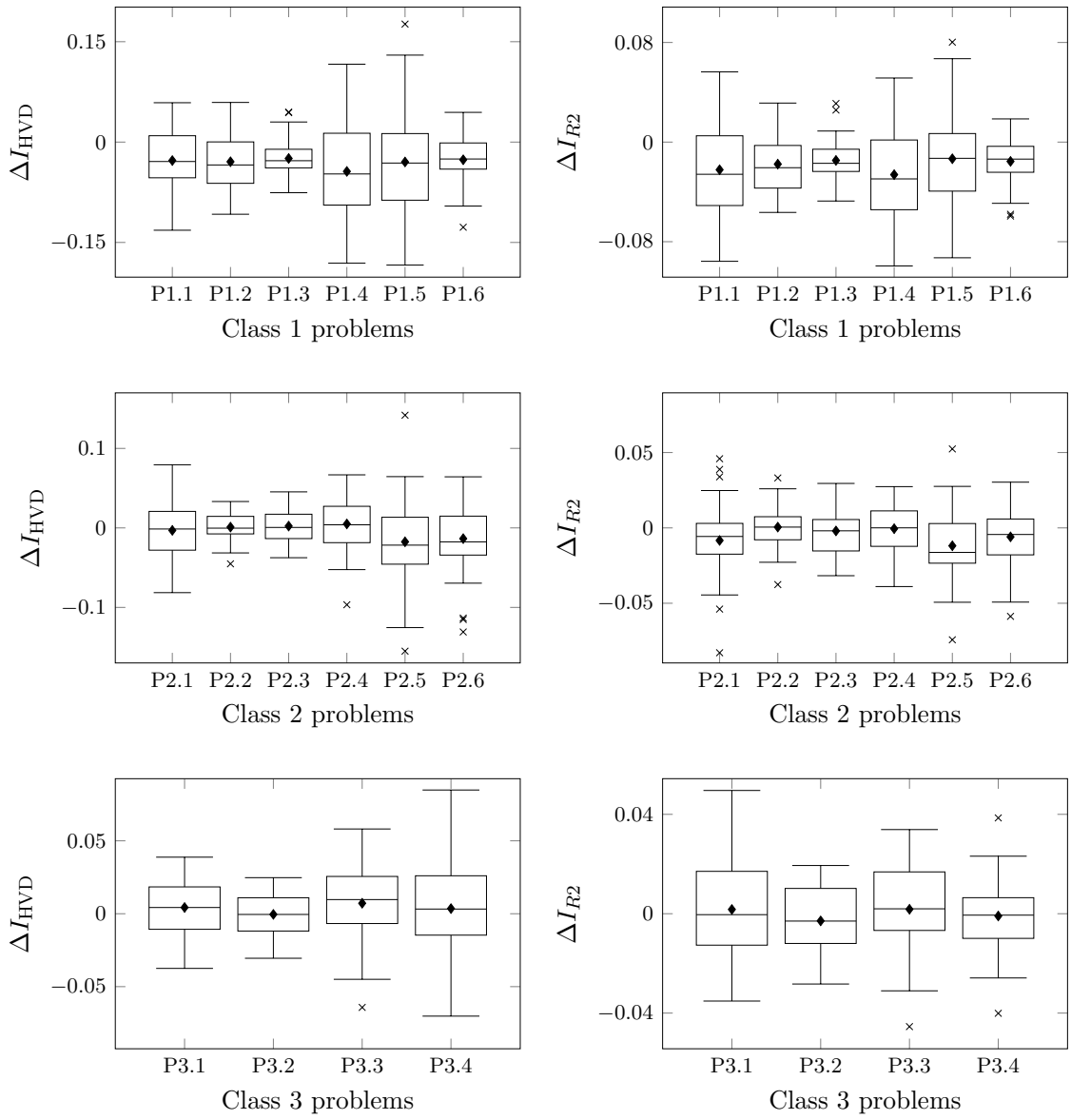


FIGURE D.3: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the AMOSA algorithm.*

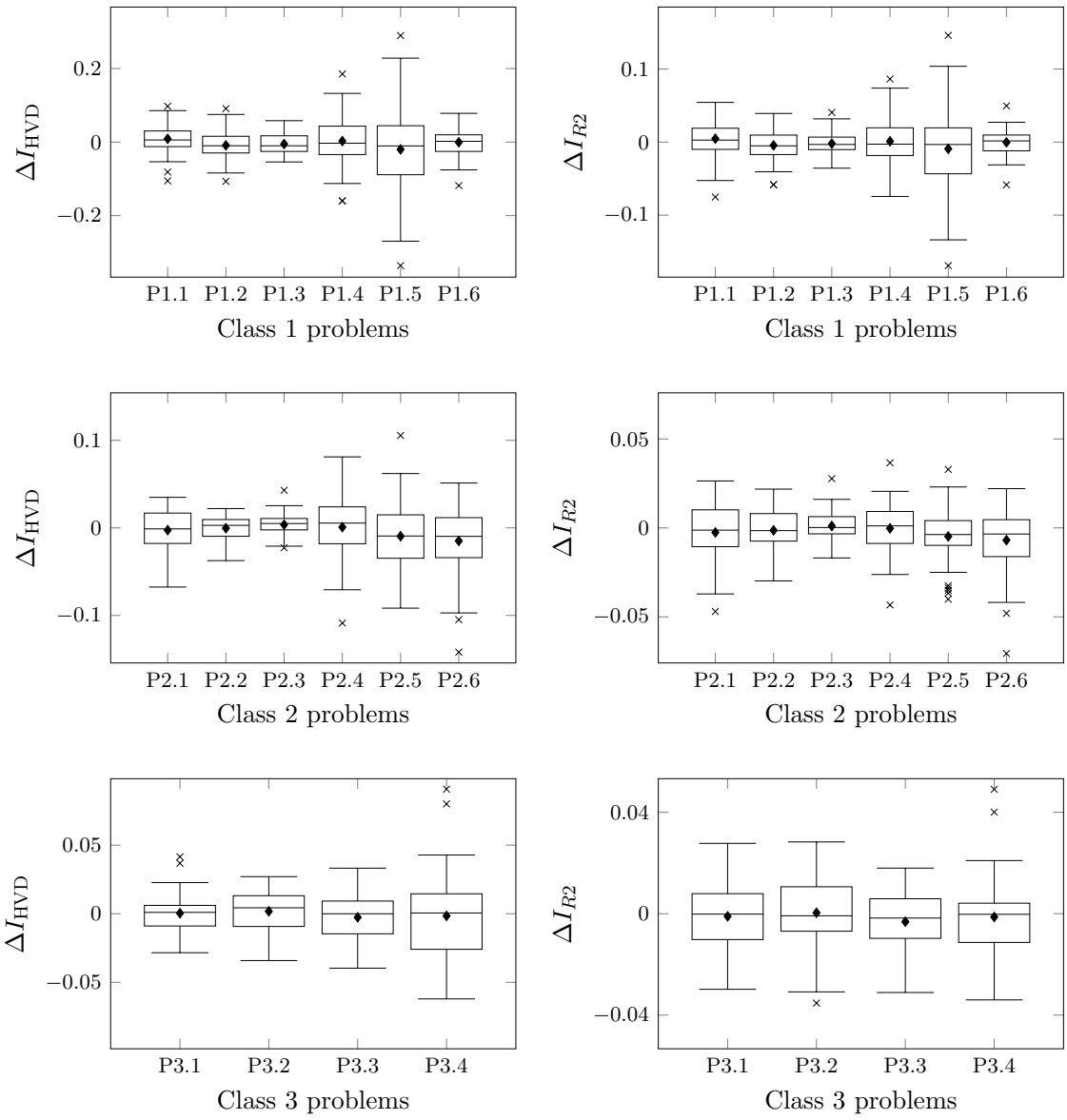


FIGURE D.4: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the MOVNS algorithm.*

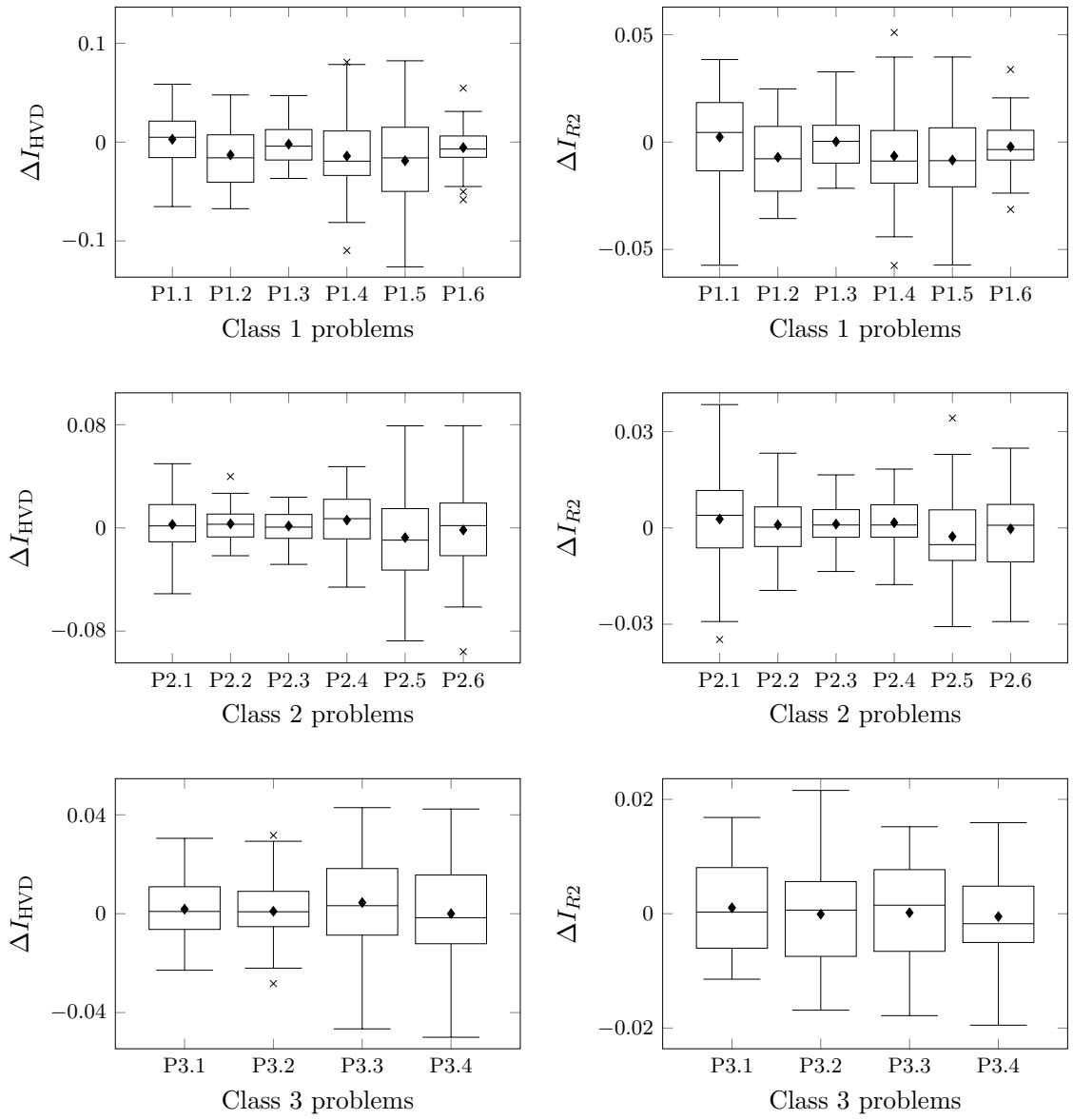


FIGURE D.5: *Box plots of the converted ΔI_{HVD} samples (on the left-hand side) and ΔI_{R2} samples (on the right-hand side) obtained for each problem instance in the test suite during the constraint handling technique comparison within the MOHS algorithm.*

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	0.2649	0.3131	**0.0248**	0.1167	0.0518	0.5988
ΔI_{R2}	0.1410	0.3667	**0.0375**	0.1689	0.0581	0.4840
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.6958	0.1812	**0.0192**	0.4202	0.7684	0.1059
ΔI_{R2}	0.7391	0.6887	0.0849	0.3131	0.4314	0.1167
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.0765	**0.0422**	0.3566	0.2608		
ΔI_{R2}	0.2370	0.2043	0.2949	0.6887		

TABLE D.1: *Single-problem analysis results for comparing constraint handling techniques within the SPEA2. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	0.1284	0.1463	0.7907	0.2567	0.9564	0.0718
ΔI_{R2}	0.1059	0.2220	0.9193	0.2447	0.9643	0.1384
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.9346	0.1145	0.2043	0.1630	0.5921	0.3368
ΔI_{R2}	0.7246	0.1975	0.7174	**0.0325**	0.6191	0.2043
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.4037	0.4147	0.8583	0.0645		
ΔI_{R2}	0.2732	0.5399	0.1844	0.1123		

TABLE D.2: *Single-problem analysis results for comparing constraint handling techniques within the OMOPSO algorithm. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	**0.0003**	**0**	**0**	**0.0004**	**0.0047**	**0**
ΔI_{R2}	**0.0002**	**0**	**0**	**0.0001**	**0.0066**	**0**
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.5463	0.6535	0.4901	0.2184	**0.0072**	**0.0441**
ΔI_{R2}	**0.011**	0.7611	0.3876	0.7981	**0.0003**	0.0506
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.0940	0.8205	**0.0254**	0.4147		
ΔI_{R2}	0.6605	0.1490	0.3417	0.7318		

TABLE D.3: *Single-problem analysis results for comparing constraint handling techniques within the AMOSA algorithm. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	0.0679	0.1630	0.1660	0.7174	0.2732	0.8281
ΔI_{R2}	0.1725	0.2257	0.2818	0.8507	0.3224	0.9423
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.6958	0.8583	**0.0317**	0.5592	**0.0431**	**0.0302**
ΔI_{R2}	0.5399	0.7391	0.5272	0.8130	**0.0375**	**0.0236**
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.9116	0.2257	0.4428	0.5921		
ΔI_{R2}	0.6397	0.9961	0.1145	0.3667		

TABLE D.4: *Single-problem analysis results for comparing constraint handling techniques within the MOVNS algorithm. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

Sample	Wilcoxon signed rank test p-values					
	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6
ΔI_{HVD}	0.4314	**0.0054**	0.4602	**0.0085**	**0.0160**	0.0529
ΔI_{R2}	0.3417	**0.0074**	0.9961	**0.0203**	**0.0173**	0.1545
	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6
ΔI_{HVD}	0.4962	0.0959	0.3929	**0.0366**	0.1018	0.8658
ΔI_{R2}	0.0959	0.4720	0.1781	0.1909	0.0940	0.9730
	P3.1	P3.2	P3.3	P3.4		
ΔI_{HVD}	0.4544	0.5023	0.1436	0.9654		
ΔI_{R2}	0.5463	0.9961	0.6466	0.8056		

TABLE D.5: *Single-problem analysis results for comparing constraint handling techniques within the MOHS algorithm. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the ΔI_{HVD} and ΔI_{R2} samples for each problem instance. Bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$).*

	Problem	SPEA2		OMOPSO		AMOSA		MOVNS		MOHS	
		$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$	$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$	$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$	$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$	$\overline{\Delta I}_{\text{HVD}}$	$\overline{\Delta I}_{R2}$
Class 1	P1.1	−0.0102	−0.0088	0.0128	0.0083	−0.0276	−0.0222	0.0090	0.0047	0.0028	0.0023
	P1.2	−0.0083	−0.0042	−0.0078	−0.0046	−0.0294	−0.0177	−0.0089	−0.0045	−0.0130	−0.0071
	P1.3	−0.0127	−0.0064	0.0004	0.0017	−0.0245	−0.0147	−0.0051	−0.0021	−0.0021	0.0002
	P1.4	−0.0181	−0.0089	0.0097	0.0070	−0.0438	−0.0262	0.0029	0.0011	−0.0141	−0.0065
	P1.5	−0.0279	−0.0131	0.0017	0.0010	−0.0297	−0.0134	−0.0199	−0.0092	−0.0189	−0.0084
	P1.6	−0.0043	−0.0027	−0.0110	−0.0054	−0.0264	−0.0155	−0.0005	−0.0004	−0.0056	−0.0022
Class 2	P2.1	0.0025	0.0019	0.0011	−0.0002	−0.0031	−0.0083	−0.0026	−0.0025	0.0026	0.0027
	P2.2	0.0042	0.0012	0.0038	0.0024	0.0010	0.0005	−0.0003	−0.0014	0.0032	0.0009
	P2.3	−0.0053	−0.0028	0.0021	0.0004	0.0023	−0.0020	0.0037	0.0011	0.0015	0.0012
	P2.4	−0.0038	−0.0014	−0.0064	−0.0044	0.0050	−0.0005	0.0009	−0.0003	0.0061	0.0016
	P2.5	−0.0057	−0.0035	−0.0067	−0.0031	−0.0175	−0.0119	−0.0095	−0.0048	−0.0075	−0.0027
	P2.6	−0.0142	−0.0051	−0.0078	−0.0038	−0.0135	−0.0059	−0.0148	−0.0069	−0.0016	−0.0003
Class 3	P3.1	0.0047	0.0028	−0.0023	−0.0024	0.0043	0.0017	0.0004	−0.0010	0.0018	0.0010
	P3.2	0.0031	0.0011	0.0011	−0.0013	−0.0004	−0.0029	0.0017	0.0004	0.0010	−0.0001
	P3.3	−0.0041	−0.0023	−0.0008	−0.0025	0.0072	0.0018	−0.0026	−0.0032	0.0045	0.0002
	P3.4	−0.0041	−0.0005	−0.0080	−0.0039	0.0035	−0.0009	−0.0016	−0.0013	0.0000	−0.0005

TABLE D.6: *The converted samples of average indicator values, $\overline{\Delta I}_{\text{HVD}}$ and $\overline{\Delta I}_{R2}$, obtained for each problem instance class using the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms.*

Metaheuristic	Sample	Wilcoxon signed rank test p-values		
		Class 1	Class 2	Class 3
SPEA2	$\overline{\Delta I}_{\text{HVD}}$	**0.03125**	0.21875	1
	$\overline{\Delta I}_{R2}$	**0.03125**	0.21875	0.875
OMOPSO	$\overline{\Delta I}_{\text{HVD}}$	0.6875	0.4375	0.375
	$\overline{\Delta I}_{R2}$	0.5625	0.3125	0.125
AMOSA	$\overline{\Delta I}_{\text{HVD}}$	**0.03125**	0.5625	0.25
	$\overline{\Delta I}_{R2}$	**0.03125**	0.09375	1
MOVNS	$\overline{\Delta I}_{\text{HVD}}$	0.5625	0.4375	0.875
	$\overline{\Delta I}_{R2}$	0.5625	0.09375	0.25
MOHS	$\overline{\Delta I}_{\text{HVD}}$	0.09375	0.6875	0.125
	$\overline{\Delta I}_{R2}$	0.21875	0.4375	0.875

TABLE D.7: *Multi-problem analysis results for comparing the constraint handling techniques within the SPEA2 and the OMOPSO, AMOSA, MOVNS and MOHS algorithms. The table contains the p-values obtained by two-tailed Wilcoxon signed rank tests applied to the* $\overline{\Delta I}_{HVD}$ *and* $\overline{\Delta I}_{R2}$ *samples for each problem instance class and corresponding metaheuristic. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

D.2 Multiobjective metaheuristic solution comparison

In the single-problem analysis of the metaheuristic comparison, conducted in §8.4.2, the Friedman test detected a statistically significant difference (for $\tilde{\alpha} = 0.05$) for every problem instance in the test suite with respect to both indicators. Accordingly, the Nemenyi *post hoc* procedure was applied to the results in order to identify the individual differences between pairs of samples (*i.e.* metaheuristics). Since there are eight metaheuristics in the comparative study, the Nemenyi procedure involved $\binom{8}{2} = 28$ pairwise significance tests. The resulting p-values obtained for each of these pairwise tests are presented in Tables D.8–D.10 for all the problem instances in class 1, class 2 and class 3, respectively. As before, bold-faced entries represent a statistically significant difference (for $\tilde{\alpha} = 0.05$). If a significant difference is detected, the box plots in Figures 8.2–8.4 may be consulted in order to pronounce on which metaheuristics outperformed the others.

The Friedman test also detected a statistically significant difference (for $\tilde{\alpha} = 0.05$) between the metaheuristics in all three problem instance classes, for both indicators, during the multi-problem analysis conducted in §8.4.2. The Nemenyi *post hoc* procedure was therefore applied, again, to the average indicator value samples for each class of problem instances. The resulting p-values obtained for each of the pairwise significance tests are presented in Table D.11 for the three problem instance classes, with bold-faced entries representing a significant difference (for $\tilde{\alpha} = 0.05$).

Metaheuristic pairs		P1.1		P1.2		P1.3		P1.4		P1.5		P1.6	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
NSGA-II	SPEA2	0.9473	0.9715	1	1	0.9999	1	1	0.9823	0.9997	0.9960	0.4815	0.3740
NSGA-II	OMOPSO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.0002**	**0.0010**	**0**	**0**
NSGA-II	P-ACO	0.1662	0.9972	0.9863	0.9943	0.0079	1	0.7553	0.1241	**0.0002**	**0**	**0.0079**	0.9715
NSGA-II	AMOSA	0.2369	1	0.6777	1	**0**	**0.0002**	1	0.9248	0.8441	0.8631	**0**	**0**
NSGA-II	MOVNS	1	0.9960	0.7044	0.9367	0.9863	0.9923	0.1121	0.3011	0.0908	0.1010	0.6226	0.6504
NSGA-II	MOOCEM	0.8968	**0.0122**	0.8021	0.9473	1	0.8968	**0.0037**	0.0728	0.9998	1	0.9565	1
NSGA-II	MOHS	**0**	**0**	0.7553	0.8631	**0.0043**	**0.0068**	0.5378	0.6226	0.3740	0.2573	0.0908	0.3245
SPEA2	OMOPSO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SPEA2	P-ACO	0.8441	0.6777	0.9115	0.9565	**0.0314**	1	0.6226	**0.0068**	**0.0014**	**0.0012**	0.7553	0.9473
SPEA2	AMOSA	0.9115	0.9367	0.4266	1	**0**	**0.0012**	1	0.3740	0.9823	0.9981	**0.0003**	0.1241
SPEA2	MOVNS	0.9565	1	0.8968	0.8021	0.9997	0.9998	0.0650	**0.0276**	0.2787	0.4538	1	0.9999
SPEA2	MOOCEM	1	0.2176	0.9473	0.8237	0.9972	0.9823	**0.0016**	**0.0031**	0.9823	0.9896	**0.0404**	0.3488
SPEA2	MOHS	**0.0006**	**0.0043**	0.5095	0.9646	**0.0185**	**0.0242**	0.6777	0.9896	0.7044	0.7303	0.9923	1
OMOPSO	P-ACO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	AMOSA	**0**	**0**	**0**	**0**	**0.0019**	**0**	**0**	**0**	**0**	**0**	**0.0019**	**0.0001**
OMOPSO	MOVNS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.0016**	**0.0016**	**0**	**0**
OMOPSO	MOHS	**0.0001**	**0.0003**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
P-ACO	AMOSA	1	0.9995	0.9923	0.9473	0.1371	**0.0010**	0.8021	0.8237	**0.0404**	**0.0140**	0.1010	**0.0031**
P-ACO	MOVNS	0.1822	0.8441	0.1662	0.9999	0.1241	0.9997	0.9473	0.9999	0.6777	0.4538	0.6226	0.9960
P-ACO	MOOCEM	0.9115	**0.0008**	0.2369	1	**0.0027**	0.9774	0.3488	1	**0**	**0**	**0.0001**	0.9646
P-ACO	MOHS	0.0908	**0**	0.9972	0.3740	1	**0.0212**	**0.0106**	**0.0002**	0.2573	0.2176	0.9943	0.9248
AMOSA	MOVNS	0.2573	0.9863	**0.0161**	0.7793	**0**	**0.0068**	0.1371	0.9646	0.8631	0.8631	**0.0001**	**0.0404**
AMOSA	MOOCEM	0.9565	**0.0068**	**0.0276**	0.8021	**0**	**0.0356**	**0.0050**	0.7044	0.5662	0.8021	**0**	**0**
AMOSA	MOHS	0.0579	**0**	1	0.9715	0.1994	0.9923	0.4815	0.0515	0.9960	0.9774	**0.0091**	0.1511
MOVNS	MOOCEM	0.9115	0.1121	1	1	0.9367	0.9997	0.9646	0.9987	**0.0242**	0.0728	0.0728	0.6226
MOVNS	MOHS	**0**	**0.0014**	**0.0242**	0.1662	0.0814	0.0908	**0.0001**	**0.0012**	0.9981	0.9999	0.9715	0.9997
MOOCEM	MOHS	**0.0012**	0.8968	**0.0404**	0.1822	**0.0014**	0.2787	**0**	**0.0001**	0.1511	0.1994	**0.0023**	0.3011

TABLE D.8: *Single-problem analysis results for comparing the metaheuristics in respect of class 1 problem instances. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which 28 pairwise significance tests are performed for each problem instance in class 1. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

Metaheuristic pairs		P2.1		P2.2		P2.3		P2.4		P2.5		P2.6	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
NSGA-II	SPEA2	0.6504	0.0515	0.5095	0.9943	**0**	**0**	0.1511	0.9473	1	0.9823	0.7793	0.7793
NSGA-II	OMOPSO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
NSGA-II	P-ACO	0.9923	0.9943	0.8968	0.9896	1	1	1	1	0.9981	0.8631	0.4815	0.9473
NSGA-II	AMOSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
NSGA-II	MOVNS	**0**	**0**	**0**	**0**	**0**	**0**	**0.0079**	**0**	**0.0079**	**0**	**0.0031**	**0.0016**
NSGA-II	MOOCEM	1	0.4815	**0.0212**	0.9715	0.6226	0.9774	0.8441	1	**0.0276**	0.5945	0.7793	0.3488
NSGA-II	MOHS	0.5095	**0.0003**	0.8631	0.2176	1	0.4815	0.9998	0.6504	1	0.9863	**0.0012**	**0.0031**
SPEA2	OMOPSO	**0**	**0**	**0**	**0**	**0.0356**	0.3740	**0**	**0**	**0**	**0**	**0**	**0**
SPEA2	P-ACO	0.9823	0.3245	0.9981	0.7303	**0.0002**	**0**	0.2369	0.9923	0.9999	0.9998	0.9998	0.9999
SPEA2	AMOSA	**0**	**0.0006**	**0**	**0**	**0.0122**	0.7793	**0**	**0**	**0**	**0**	**0**	**0.0002**
SPEA2	MOVNS	**0.0031**	**0.0003**	**0**	**0**	0.5662	0.9823	**0**	**0**	**0.0185**	**0.0031**	0.3011	0.2176
SPEA2	MOOCEM	0.4266	0.9715	0.8807	1	**0**	**0**	0.9367	0.8021	**0.0122**	0.1010	1	0.9981
SPEA2	MOHS	1	0.8631	0.9992	0.7044	**0**	**0**	**0.0456**	0.0728	0.9998	1	0.1822	0.3011
OMOPSO	P-ACO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	AMOSA	0.1241	**0.0043**	0.4815	0.8631	1	0.9987	0.3740	0.2176	0.0814	**0.0091**	**0.0014**	**0.0001**
OMOPSO	MOVNS	**0.0003**	**0.0079**	0.3740	0.5662	0.9115	0.9248	**0.0002**	**0.0456**	**0**	**0**	**0**	**0**
OMOPSO	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	MOHS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
P-ACO	AMOSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.0001**	**0**	**0**
P-ACO	MOVNS	**0**	**0**	**0**	**0**	**0**	**0**	**0.0037**	**0**	0.0650	**0.0161**	0.5945	0.0814
P-ACO	MOOCEM	0.9473	0.9248	0.4815	0.5662	0.3999	0.9987	0.9248	0.9972	**0.0027**	**0.0276**	0.9998	0.9646
P-ACO	MOHS	0.9473	**0.0068**	1	**0.0212**	0.9981	0.7303	0.9981	0.4266	0.9896	0.9997	0.4266	0.1241
AMOSA	MOVNS	0.6777	1	1	0.9997	0.7553	0.9987	0.3011	0.9987	0.2176	0.8968	0.0579	0.4538
AMOSA	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.0031**
AMOSA	MOHS	**0**	0.0814	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0.1121	0.3488
MOVNS	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0.3011	0.6226
MOVNS	MOHS	**0.0068**	0.0515	**0**	**0**	**0**	**0**	**0.0356**	**0.0007**	**0.0037**	**0.0027**	1	1
MOOCEM	MOHS	0.3011	0.2369	0.5378	0.8441	0.8237	0.9715	0.5662	0.8631	0.0515	0.1121	0.1822	0.7303

TABLE D.9: *Single-problem analysis results for comparing the metaheuristics in respect of class 2 problem instances. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which 28 pairwise significance tests are performed for each problem instance in class 2. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

Metaheuristic pairs		P3.1		P3.2		P3.3		P3.4	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
NSGA-II	SPEA2	**0**	**0**	0.5945	0.9960	**0.0356**	**0**	0.7793	1
NSGA-II	OMOPSO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
NSGA-II	P-ACO	0.9823	1	1	0.9473	0.5662	0.9473	0.8807	0.5945
NSGA-II	AMOSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
NSGA-II	MOVNS	**0**	**0**	**0**	**0**	**0.0019**	**0**	**0.0001**	**0**
NSGA-II	MOOCEM	1	0.2787	1	0.7044	0.6504	1	0.9987	0.9943
NSGA-II	MOHS	0.9960	0.3245	1	0.0908	0.5945	1	0.9997	0.9646
SPEA2	OMOPSO	1	0.9248	**0**	**0**	**0.0002**	**0.0031**	**0**	**0**
SPEA2	P-ACO	**0**	**0**	0.5945	0.5662	**0**	**0**	1	0.7793
SPEA2	AMOSA	0.6504	0.9646	**0**	**0**	0.1371	0.1662	**0**	**0**
SPEA2	MOVNS	1	0.8021	**0**	**0**	0.9923	0.9981	**0**	**0**
SPEA2	MOOCEM	**0**	**0**	0.4815	0.9823	**0**	**0**	0.9823	0.9646
SPEA2	MOHS	**0**	**0**	0.5945	0.4266	**0**	**0**	0.9646	0.8807
OMOPSO	P-ACO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	AMOSA	0.5095	0.3011	0.9715	0.9987	0.6226	0.9115	0.0515	0.2176
OMOPSO	MOVNS	1	1	0.9896	0.9972	**0.0068**	**0.0314**	**0.0031**	**0.0404**
OMOPSO	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
OMOPSO	MOHS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
P-ACO	AMOSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
P-ACO	MOVNS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
P-ACO	MOOCEM	0.9565	0.1511	1	0.0908	1	0.9923	0.996	0.1511
P-ACO	MOHS	1	0.1822	1	**0.0019**	1	0.9565	0.9896	0.0728
AMOSA	MOVNS	0.3999	0.1662	1	1	0.5945	0.5378	0.9923	0.9981
AMOSA	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
AMOSA	MOHS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
MOVNS	MOOCEM	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
MOVNS	MOHS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
MOOCEM	MOHS	0.9863	1	1	0.9473	1	1	1	1

TABLE D.10: *Single-problem analysis results for comparing the metaheuristics in respect of class 3 problem instances. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which 28 pairwise significance tests are performed for each problem instance in class 3. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

Metaheuristic pairs		Class 1		Class 2		Class 3	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
NSGA-II	SPEA2	1	0.9968	1	0.9382	1	0.8370
NSGA-II	OMOPSO	**0.0217**	**0.0453**	**0.0097**	**0.0003**	0.5673	**0.05**
NSGA-II	P-ACO	0.9918	0.9647	0.9968	0.9998	0.9731	0.9991
NSGA-II	AMOSA	0.6421	0.9818	**0.0453**	**0.0040**	0.5673	0.0752
NSGA-II	MOVNS	0.9990	1	0.3281	**0.0316**	0.7579	0.2157
NSGA-II	MOOCEM	0.9990	1	0.9818	0.9647	0.9999	0.9731
NSGA-II	MOHS	0.7902	0.9006	0.9968	0.4794	1	0.8996
SPEA2	OMOPSO	0.0638	0.2628	**0.0146**	**0.0316**	0.4676	0.7579
SPEA2	P-ACO	0.9998	0.6421	0.9990	0.9968	0.9890	0.4676
SPEA2	AMOSA	0.8511	1	0.0638	0.1584	0.4676	0.8370
SPEA2	MOVNS	0.9818	0.9818	0.4009	0.4794	0.6661	0.9731
SPEA2	MOOCEM	0.9818	0.9968	0.9647	1	1	0.9999
SPEA2	MOHS	0.9382	0.9990	0.9990	0.9918	1	1
OMOPSO	P-ACO	0.2062	**0.0010**	0.0880	**0.0026**	0.0752	**0.0074**
OMOPSO	AMOSA	0.7902	0.4009	0.9998	0.9990	1	1
OMOPSO	MOVNS	**0.0026**	**0.0217**	0.9006	0.9382	1	0.9991
OMOPSO	MOOCEM	**0.0026**	**0.0453**	**0.0002**	**0.0217**	0.2882	0.4676
OMOPSO	MOHS	0.6421	0.6421	0.0880	0.2628	0.3731	0.6661
P-ACO	AMOSA	0.9818	0.4794	0.2628	**0.0217**	0.0752	**0.0125**
P-ACO	MOVNS	0.8511	0.9918	0.7902	0.1193	0.1564	**0.05**
P-ACO	MOOCEM	0.8511	0.9647	0.7196	0.9990	0.9991	0.7579
P-ACO	MOHS	0.9968	0.2628	1	0.7902	0.9964	0.5673
AMOSA	MOVNS	0.2628	0.9382	0.9918	0.9990	1	0.9999
AMOSA	MOOCEM	0.2628	0.9818	**0.0016**	0.1193	0.2882	0.5673
AMOSA	MOHS	1	1	0.2628	0.6421	0.3731	0.7579
MOVNS	MOOCEM	1	1	**0.0316**	0.4009	0.4676	0.8370
MOVNS	MOHS	0.4009	0.7902	0.7902	0.9382	0.5673	0.9444
MOOCEM	MOHS	0.4009	0.9006	0.7196	0.9818	1	1

TABLE D.11: *Multi-problem analysis results for comparing the metaheuristics for each problem instance class. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which 28 pairwise significance tests are performed for each problem instance class. Bold-faced entries represent a statistically significant difference (for* $\tilde{\alpha} = 0.05$*).*

APPENDIX E

Additional hyperheuristic experimental results

This appendix contains additional results obtained during the comparative studies described in Chapter 9 between four variants of the AMALGAM method, and between one of those variants and its constituent sub-algorithms, but which were not presented in §9.5 so as to enhance the exposition of the main text.

In the single-problem analysis for comparing the four variants of the AMALGAM method, conducted in §9.5.1, the Friedman test detected a statistically significant difference (for $\tilde{\alpha} = 0.05$) for several problem instances in the test suite with respect to both indicators. Accordingly, the Nemenyi *post hoc* procedure was applied to the relevant results in order to identify the individual differences between pairs of samples (*i.e.* hyperheuristic variants). Since there are four variants in the comparative study, the Nemenyi procedure involved $\binom{4}{2} = 6$ pairwise significance tests. The resulting p-values obtained for each of these pairwise tests are presented in Table E.1 for the relevant problem instances. As before, bold-faced entries represent statistically significant differences (for $\tilde{\alpha} = 0.05$). If a significant difference is detected, the box plots in Figures 9.4–9.6 may be consulted in order to pronounce on which variant outperformed the others.

The Friedman test also detected a statistically significant difference (for $\tilde{\alpha} = 0.05$) between the four variants of the AMALGAM method in problem instance class 2 (for both indicators) and for problem instance class 3 (for I_{R2}) during the multi-problem analysis conducted in §9.5.1. The Nemenyi *post hoc* procedure was therefore applied, again, to the average indicator value samples for the relevant class of problem instances. The resulting p-values obtained for each of the pairwise significance tests are presented in Table E.2 for the relevant problem instance classes, with bold-faced entries representing significant differences (for $\tilde{\alpha} = 0.05$).

Finally, in the single-problem analysis for comparing the AMALGAM$_{\text{npm}}$ method against its constituent sub-algorithms, conducted in §9.5.2, the Friedman test detected a statistically significant difference (for $\tilde{\alpha} = 0.05$) for every problem instance in the test suite with respect to both indicators. Accordingly, the NWWM *post hoc* procedure was applied to the results in order to identify which sub-algorithm differs from the AMALGAM$_{\text{npm}}$ method. Since there are three sub-algorithms within AMALGAM$_{\text{npm}}$ method, the NWWM procedure involved three one-tailed pairwise significance tests. The resulting p-values obtained for each of these pairwise tests are presented in Table E.3 for all the problem instances in the three classes. As before, bold-faced entries represent statistically significant differences (for $\tilde{\alpha} = 0.05$). Since the pairwise significance tests are one-tailed, if a significant difference is detected, it may be inferred that the AMALGAM$_{\text{npm}}$ method performs significantly better than the associated sub-algorithm.

Hyperheuristic pairs		P1.1		P1.2		P1.3		P1.4		P1.5		P1.6	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npm}}$	—	—	—	—	—	—	—	—	0.1827	0.1320	—	—
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{np}}$	—	—	—	—	—	—	—	—	0.7890	0.8294	—	—
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	—	—	—	—	—	—	0.8660	0.8982	—	—
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{np}}$	—	—	—	—	—	—	—	—	**0.0171**	**0.0134**	—	—
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	—	—	—	—	—	—	**0.0272**	**0.0216**	—	—
$\text{AMALGAM}_{\text{np}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	—	—	—	—	—	—	0.9987	0.9987	—	—
		P2.1		P2.2		P2.3		P2.4		P2.5		P2.6	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npm}}$	—	—	0.2822	0.2127	0.9258	—	—	—	0.6510	0.3635	—	—
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{np}}$	—	—	0.1827	0.7890	0.4080	—	—	—	0.9897	0.7453	—	—
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	0.1110	0.2127	0.2459	—	—	—	0.1558	0.1827	—	—
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{np}}$	—	—	**0.0008**	0.7453	0.1320	—	—	—	0.4547	0.9258	—	—
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	**0.0003**	**0.0006**	0.0633	—	—	—	**0.0063**	**0.0016**	—	—
$\text{AMALGAM}_{\text{np}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	0.9956	**0.0216**	0.9897	—	—	—	0.2822	**0.0134**	—	—
		P3.1		P3.2		P3.3		P3.4					
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}				
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npm}}$	0.0517	**0.0012**	0.5029	**0.0037**	0.2127	0.1320	—	**0.0272**				
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{np}}$	0.7453	0.0768	0.9956	**0.0028**	0.3635	0.0517	—	**0.0063**				
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npv}}$	0.9987	0.8982	0.2822	0.9956	0.5029	0.7453	—	**0.0171**				
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{np}}$	0.4080	0.5521	0.3635	0.9998	0.9897	0.9803	—	0.9667				
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{npv}}$	0.0768	**0.0134**	**0.0081**	**0.0081**	**0.0048**	**0.0081**	—	0.9987				
$\text{AMALGAM}_{\text{np}}$	$\text{AMALGAM}_{\text{npv}}$	0.8294	0.3214	0.4080	**0.0063**	**0.0134**	**0.0021**	—	0.9897				

TABLE E.1: *Single-problem analysis results for comparing the four variants of the AMALGAM method in respect of all problem instances. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which six pairwise significance tests are performed for each problem instance. Bold-faced entries represent statistically significant differences (for* $\tilde{\alpha} = 0.05$*).*

Hyperheuristic pairs		Class 1		Class 2		Class 3	
		I_{HVD}	I_{R2}	I_{HVD}	I_{R2}	I_{HVD}	I_{R2}
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npm}}$	—	—	0.6784	0.1833	—	0.2208
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{np}}$	—	—	0.8078	0.1833	—	0.0655
$\text{AMALGAM}_{\text{npmv}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	0.1833	0.5363	—	1
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{np}}$	—	—	0.1833	1	—	0.9472
$\text{AMALGAM}_{\text{npm}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	**0.0095**	**0.0044**	—	0.2208
$\text{AMALGAM}_{\text{np}}$	$\text{AMALGAM}_{\text{npv}}$	—	—	0.6784	**0.0044**	—	0.0655

TABLE E.2: *Multi-problem analysis results for comparing the four variants of the AMALGAM method in respect of all problem instance classes. The table contains the p-values obtained by the Nemenyi* post hoc *procedure in which six pairwise significance tests are performed for each problem instance class. Bold-faced entries represent statistically significant differences (for* $\tilde{\alpha} = 0.05$*).*

	Problem	NWWM procedure p-values for I_{HVD}			NWWM procedure p-values for I_{R2}		
		NSGA-II	P-ACO	MOOCEM	NSGA-II	P-ACO	MOOCEM
Class 1	P1.1	$\mathbf{1.779\times10^{-2}}$	$\mathbf{7.155\times10^{-7}}$	$\mathbf{1.151\times10^{-4}}$	$\mathbf{1.158\times10^{-2}}$	9.096×10^{-2}	$\mathbf{1.615\times10^{-9}}$
	P1.2	$\mathbf{2.944\times10^{-4}}$	$\mathbf{2.341\times10^{-6}}$	1.058×10^{-1}	$\mathbf{3.032\times10^{-5}}$	$\mathbf{5.978\times10^{-5}}$	$\mathbf{9.472\times10^{-4}}$
	P1.3	$\mathbf{3.978\times10^{-4}}$	$\mathbf{2.058\times10^{-10}}$	$\mathbf{2.944\times10^{-4}}$	$\mathbf{5.012\times10^{-6}}$	$\mathbf{1.358\times10^{-7}}$	$\mathbf{3.985\times10^{-10}}$
	P1.4	$\mathbf{2.140\times10^{-5}}$	$\mathbf{2.138\times10^{-3}}$	5.224×10^{-1}	$\mathbf{5.012\times10^{-6}}$	1.224×10^{-1}	2.599×10^{-1}
	P1.5	$\mathbf{3.570\times10^{-3}}$	9.932×10^{-1}	$\mathbf{2.138\times10^{-3}}$	$\mathbf{1.640\times10^{-3}}$	9.860×10^{-1}	$\mathbf{1.250\times10^{-3}}$
	P1.6	$\mathbf{7.135\times10^{-4}}$	$\mathbf{2.063\times10^{-10}}$	$\mathbf{5.821\times10^{-3}}$	$\mathbf{2.140\times10^{-5}}$	$\mathbf{7.331\times10^{-10}}$	$\mathbf{7.268\times10^{-6}}$
Class 2	P2.1	1.610×10^{-1}	$\mathbf{1.250\times10^{-3}}$	1.831×10^{-1}	6.600×10^{-2}	$\mathbf{8.321\times10^{-5}}$	$\mathbf{7.268\times10^{-6}}$
	P2.2	$\mathbf{2.362\times10^{-8}}$	$\mathbf{1.151\times10^{-4}}$	1.408×10^{-1}	$\mathbf{1.502\times10^{-5}}$	$\mathbf{3.570\times10^{-3}}$	$\mathbf{6.102\times10^{-9}}$
	P2.3	$\mathbf{7.366\times10^{-3}}$	$\mathbf{9.472\times10^{-4}}$	4.873×10^{-1}	5.224×10^{-1}	1.610×10^{-1}	$\mathbf{4.572\times10^{-3}}$
	P2.4	$\mathbf{7.135\times10^{-4}}$	$\mathbf{3.900\times10^{-2}}$	1.610×10^{-1}	$\mathbf{2.771\times10^{-3}}$	$\mathbf{3.234\times10^{-2}}$	$\mathbf{3.978\times10^{-4}}$
	P2.5	$\mathbf{1.584\times10^{-4}}$	$\mathbf{1.586\times10^{-6}}$	6.587×10^{-1}	$\mathbf{4.572\times10^{-3}}$	$\mathbf{1.358\times10^{-7}}$	1.224×10^{-1}
	P2.6	7.771×10^{-2}	$\mathbf{8.321\times10^{-5}}$	$\mathbf{9.472\times10^{-4}}$	$\mathbf{1.158\times10^{-2}}$	$\mathbf{5.978\times10^{-5}}$	$\mathbf{7.155\times10^{-7}}$
Class 3	P3.1	$\mathbf{1.640\times10^{-3}}$	1.610×10^{-1}	$\mathbf{3.978\times10^{-4}}$	$\mathbf{2.138\times10^{-3}}$	1.610×10^{-1}	$\mathbf{1.070\times10^{-9}}$
	P3.2	$\mathbf{4.764\times10^{-7}}$	$\mathbf{1.584\times10^{-4}}$	$\mathbf{2.140\times10^{-5}}$	$\mathbf{1.502\times10^{-5}}$	$\mathbf{7.366\times10^{-3}}$	$\mathbf{2.086\times10^{-10}}$
	P3.3	$\mathbf{9.472\times10^{-4}}$	9.096×10^{-2}	1.224×10^{-1}	$\mathbf{2.771\times10^{-3}}$	$\mathbf{2.138\times10^{-3}}$	$\mathbf{7.268\times10^{-6}}$
	P3.4	$\mathbf{7.135\times10^{-4}}$	$\mathbf{1.440\times10^{-2}}$	$\mathbf{5.343\times10^{-4}}$	$\mathbf{1.584\times10^{-4}}$	7.771×10^{-2}	$\mathbf{4.764\times10^{-7}}$

TABLE E.3: *Single-problem analysis results for comparing the $AMALGAM_{npm}$ method against its constituent sub-algorithms. The table contains the p-values obtained by the NWWM post hoc procedure in which three pairwise significance tests are performed for each problem instance. Bold-faced entries represent statistically significant differences (for $\tilde{\alpha} = 0.05$).*