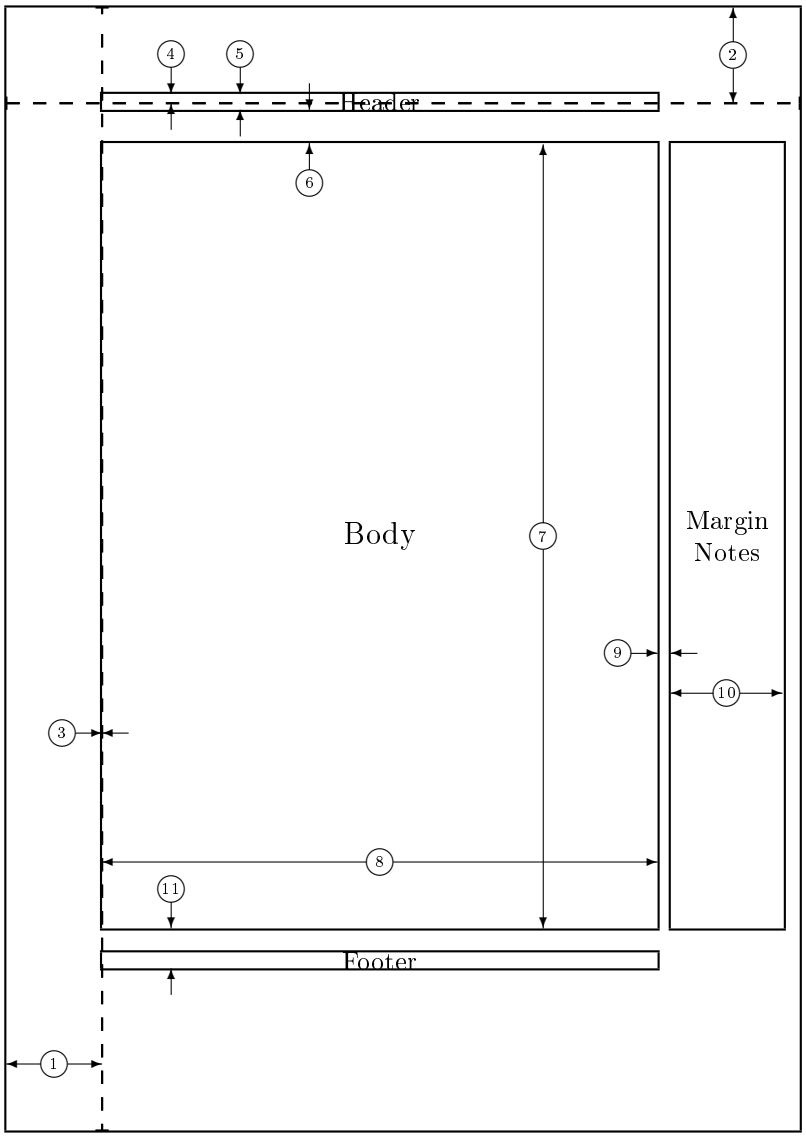
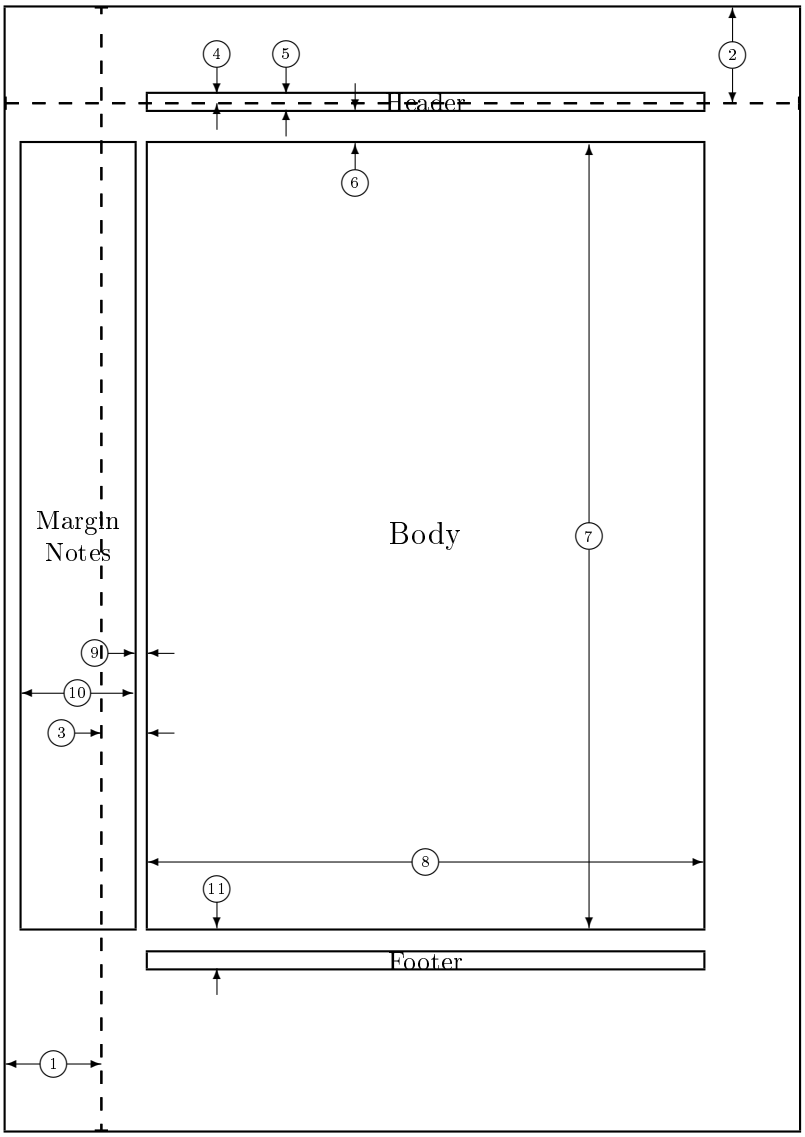



```
 1  one inch + \hoffset      2  one inch + \voffset
 3  \oddsidemargin = 0pt     4  \topmargin = -7pt
 5  \headheight = 12pt       6  \headsep = 25pt
 7  \textheight = 591pt      8  \textwidth = 418pt
 9  \marginparsep = 10pt    10  \marginparwidth = 85pt
11  \footskip = 30pt            \marginparpush = 7pt (not shown)
    \hoffset = 0pt              \voffset = 0pt
    \paperwidth = 597pt         \paperheight = 845pt
```




| | | | |
|---|---|---|---|
| 1 | one inch + \hoffset | 2 | one inch + \voffset |
| 3 | \evensidemargin = 35pt | 4 | \topmargin = -7pt |
| 5 | \headheight = 12pt | 6 | \headsep = 25pt |
| 7 | \textheight = 591pt | 8 | \textwidth = 418pt |
| 9 | \marginparsep = 10pt | 10 | \marginparwidth = 85pt |
| 11 | \footskip = 30pt | | \marginparpush = 7pt (not shown) |
| | \hoffset = 0pt | | \voffset = 0pt |
| | \paperwidth = 597pt | | \paperheight = 845pt |

# Multi-Objective Optimisation Methods Applied to Aircraft Techno-economic and Environmental Issues

C. Tsotskas <c.tsotskas@gmail.com>

26th October 2016

*This page is intentionally left blank.*

Cranfield University, School of Engineering, Department of Power and Propulsion

Dissertation (Full Time PhD)

Christos Tsotskas

PhD
Academic Year: 2010-2014

Supervisors: Prof. A. M. Savill, Dr T. Kipouros

Date of Initial Registration: 7 November 2010

Date of Submission: **7 November 2014**

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

©Cranfield University 2014. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

*This page is intentionally left blank.*

To my parents and other half, for the time spent apart.

Christos Tsotskas
November 2014

Στους γονείς μου και στο άλλο μου μισό, για το χρόνο που τους στέρησα.

Χρηστος Τσοτσκας
Νοεμβριος 2014

*This page is intentionally left blank.*

For the goal is not the last, but the best. Aristoteles (Second Book of Physics, ii, 194a 32-33)

Το ρόλο του τέλους δεν τον διεκδικεί κάθε έσχατο σημείο, αλλά μόνο το βέλτιστον. Αριστοτέλης (Φυσικά, II, ιι, 194α 32-33)

Be content with the present, but seek after what is best. Isocrates, To Demonicus, section 29

Στέργε μέν τά παρόντα, ζήτει δέ τά βελτίω. Ισοκράτης, Προς Δημόνικον, ενότητα 29

*This page is intentionally left blank.*

## Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration. This dissertation contains a total of 92330 words and 83 Figures in 291 pages.

Christos Tsotskas

*This page is intentionally left blank.*

# Acknowledgements

The research has been carried out at Cranfield University between November 2010 and November 2014.

The research project has received funding from the European Union's Seventh Framework Program (FP7/2007-2013) for the Clean Sky Joint Technology Initiative under grant agreement N0 CJSU-GAMSGO-2008-001.

I would like to thank my supervisors Prof. A. M. Savill and Dr T. Kipouros for their patience and guidance throughout my first research steps.

I would also like to thank CLEANSKY System for Green operations team, Dr V. Sethi and his team for providing access to GATAC framework and technical guidance when integrating and using GATAC.

I express my gratitude to CLEANSKY committee at Cranfield University and Bodossaki Foundation, Greece for providing additional support to carry out my research.

I would like to thank the German Academic Exchange Service (DAAD) for giving me the opportunity to meet with Prof. S. Turek and Dr M. Razzaq from TU-Dortumnd, Germany, where we exchanged significant scientific expertise and knowledge.

I would like to thank Dr G. T. Parks and Dr V. Sethi for their feedback.

Last but not least I would like to thank my family and my wife who stayed by my side throughout this journey. Their support has been invaluable.

*This page is intentionally left blank.*

# Abstract

Engineering methods that couple multi-objective optimisation (MOO) techniques with high fidelity computational tools are expected to minimise the environmental impact of aviation while increasing the growth, with the potential to reveal innovative solutions. In order to mitigate the compromise between computational efficiency and fidelity, these methods can be accelerated by harnessing the computational efficiency of Graphic Processor Units (GPUs).

The aim of the research is to develop a family of engineering methods to support research in aviation with respect to the environmental and economic aspects. In order to reveal the non-dominated trade-off, also known as Pareto Front(PF), among conflicting objectives, a MOO algorithm, called Multi-Objective Tabu Search 2 (MOTS2), is developed, benchmarked relative to state-of-the-art methods and accelerated by using GPUs. A prototype fluid solver based on GPU is also developed, so as to simulate the mixing capability of a microreactor that could potentially be used in fuel-saving technologies in aviation. By using the aforementioned methods, optimal aircraft trajectories in terms of flight time, fuel consumption and emissions are generated, and alternative designs of a microreactor are suggested, so as to assess the trade-offs between pressure losses and the micro-mixing capability.

As a key contribution to knowledge, with reference to competitive optimisers and previous cases, the capabilities of the proposed methodology are illustrated in prototype applications of aircraft trajectory optimisation (ATO) and micro-mixing optimisation with 2 and 3 objectives, under operational and geometrical constraints, respectively. In the short-term, ATO ought to be applied to existing aircraft. In the long-term, improving the micro-mixing capability of a microreactor is expected to enable the use of hydrogen-based fuel. This methodology is also benchmarked and assessed relative to state-of-the-art techniques in ATO and micro-mixing optimisation with known and unknown trade-offs, whereas the former could only optimise 2 objectives and the latter could not exploit the computational efficiency of GPUs. The impact of deploying on GPUs a micro-mixing flow solver, which accelerates the generation of trade-off against a reference study, and MOTS2, which illustrates the scalability potential, is assessed.

With regard to standard analytical function test cases and verification cases in MOO, MOTS2 can handle the multi-modality of the trade-off of ZDT4, which is a MOO benchmark function with many local optima that presents a challenge for a state-of-the-art genetic algorithm for ATO, called NSGAMO, based on case studies in the public domain. However, MOTS2 demonstrated worse performance on ZDT3, which is a MOO benchmark function with a discontinuous trade-off, for which NSGAMO successfully captured the target PF. Comparing their overall performance, if the shape of the PF is known, MOTS2 should be preferred in problems with multi-modal trade-offs, whereas NSGAMO should be employed in

discontinuous PFs. The shape of the trade-off between the objectives in airfoil shape optimisation, ATO and micro-mixing optimisation was continuous. The weakness of MOTS2 to sufficiently capture the discontinuous PF of ZDT3 was not critical in the studied examples.

First, the climb phase of a medium-haul aircraft was optimised by employing MOTS2 and NSGAMO, where both optimisers revealed comparable results, and the genetic algorithm was assessed more appropriate because of the span of solutions and the value of hypervolume quality indicator. When optimising the climb phase, the MOTS2' trade-off was narrower than NSGAMO. In terms of fuel-efficiency in the PF, NSGAMO's extreme trajectory outperforms MOTS2 by approximately 80 kg (6.38% fuel saving). Similarly, the fastest trajectory discovered by NSGAMO is 35 seconds shorter (4.07% time efficiency) against the most fuel efficient trajectory. Also, the flight path trends revealed by NSGAMO are closer to the theoretical optimal cruise-climb trajectories.

Second, by using MOTS2, a 3-phase trajectory was optimised to simultaneously minimise the fuel burn, flight time and NOx emissions. When comparing the extreme designs from the discovered PF, the optimal solutions reveal an improvement of up to 1.13 minutes (2.4% shorter travel time), 171.44 kg of consumed fuel (8.87% fuel savings) and 2.15 kg of NOx (4.37% emissions savings). In both ATO applications, specifying the altitude of the first segment of the flight is the most significant factor with respect to the optimum behaviour.

Third, the internal design of a micro-reactor was optimised by altering the geometrical layout of the device and the flow characteristics. When compared to a reference study, the simulation time in the design process was improved by approximately 20 times and the quality of the generated trade-off increased by 5% in terms of hypervolume. The generated trade-off suggested designs that maximised the micro-mixing capability up to a factor of 227% in terms of normalised vorticity magnitude, whereas the normalised difference in total pressure was as low as 1.8%. A selected compromise design achieved more than 30% improvement in both objectives. The performance of the discovered designs in micro-mixing optimisation is very similar with respect to the area of low difference of total pressure, even after the considerable speed up. However, the vorticity magnitude (i.e., the mixing capability) improved by approximately 20%. The suggested compromise design geometrically is very similar to the minimum pressure losses design, but relatively distant in terms of performance (mainly because of the flow speed), which is consistent with experimental observations, where minor changes can bring dramatic changes in performance.

Future directions suggest linking the optimisation methodology to decision support systems and integrate multiple principles, and improve the levels of automation and scope of the research in transport and other fields.

# Contents

**Abstract** xiii

**Table of Contents** xv

**List of Figures** xix

**List of Tables** xxiii

**Listings** xxv

**Acronyms and Nomenclature** xxvii

**1 Introduction** 1
- 1.1 Background . . . 1
  - 1.1.1 Improving Operations by Optimising Aircraft Trajectories . 2
  - 1.1.2 Evolving Aircraft Technology by Optimising Micro-Mixing . 2
  - 1.1.3 The Role of Simulation and Optimisation . . . 5
- 1.2 Aims and Objectives . . . 6
- 1.3 Overall Project Methodology . . . 7
- 1.4 Outline of Thesis . . . 10
- 1.5 Contribution to Knowledge . . . 11

**2 Literature Review** 13
- 2.1 Initial Contributions . . . 13
- 2.2 Computational Methods for Aircraft Technology for Mixing by using CFD . . . 14
- 2.3 Accelerating Computational Methods of Computational Fluid Dynamics . . . 15
- 2.4 Optimisation Applications by using CFD . . . 17
- 2.5 Accelerating Optimisation Methods . . . 18
- 2.6 Optimising Aircraft Trajectories . . . 19
- 2.7 Optimisation and Multi-Objective Optimisation . . . 27
- 2.8 Summary and Linking with next Chapter . . . 30

**3 Technical Description of Computational and Optimisation Tools 33**
3.1 Multi-Objective Optimisation Algorithm . . . 33
3.1.1 Introduction . . . 33
3.1.2 Trends in Multi-Objective Tabu Search . . . 36
3.1.3 The Design and Implementation of MOTS2 . . . 37
3.1.3.1 Describing the Different Types of Memories . . . 40
3.1.3.2 Algorithmic Structure . . . 44
3.1.3.3 Frequently Performed Moves . . . 48
3.1.3.4 Conditionally Performed Moves . . . 52
3.1.4 Constraints and Objectives Handling . . . 54
3.1.5 Unique Features . . . 55
3.1.6 Verifying MOTS2 . . . 58
3.1.6.1 Benchmarking the Family of ZDT Functions for Bi-objective Optimisation . . . 59
3.1.7 Validating MOTS2 . . . 65
3.1.7.1 Airfoil Shape Optimisation . . . 66
3.2 Parallel Multi-Objective Optimisation Algorithm . . . 70
3.2.1 Trends in Parallel Implementations . . . 71
3.2.2 Parallelisation Strategy for External Evaluation . . . 72
3.2.3 Scalable Multi-Objective Optimisation Problems . . . 74
3.2.4 Massive Parallelisation Strategy . . . 74
3.2.4.1 Software and Hardware Perspectives . . . 75
3.2.4.2 Design Decisions and Concerns . . . 76
3.2.4.3 Implementation . . . 78
3.2.5 Assessing GPU-MOTS2 on a Simple Benchmark . . . 79
3.2.6 Investigating the Interactions between the GPU Hardware and Algorithm Configuration on GPU-MOTS2 . . . 84
3.2.7 Discussion and Future Work . . . 87
3.3 Flow Solver Simulation . . . 89
3.3.1 Background . . . 89
3.3.2 Graphics Accelerators Programming Model . . . 90
3.3.3 Lattice Boltzmann Method for D3Q19 . . . 92
3.3.4 Porting LBM to GPU . . . 94
3.3.4.1 Algorithm . . . 97
3.3.4.2 Memory Access . . . 99
3.3.4.3 Execution . . . 100
3.3.5 Interfacing and Configuration Settings . . . 101
3.3.6 Architecture of the Flow Solver . . . 102
3.3.7 Simulating the Fluid Flow in the Microreactor by using GPU-LBM . . . 105

3.4 Summary and Linking with next Chapter . . . 113

**4 Applications** **117**
4.1 Aircraft Trajectory Optimisation . . . 117
4.1.1 Trajectory Optimisation of Climb Phase . . . 117
4.1.1.1 Problem Description . . . 117
4.1.1.2 Methodology for the Climb Phase . . . 118
4.1.1.3 Preparing the Multi-Objective Aircraft Trajectory Optimisation Process for Climb . . . 122
4.1.1.4 Optimisation Algorithms . . . 124
4.1.1.5 Optimising Climb - Results and Discussion . . . 126
4.1.1.6 Identified Issues . . . 133
4.1.2 Trajectory Optimisation of Climb, Cruise and Descent . . . 134
4.1.2.1 Problem Description . . . 134
4.1.2.2 Methodology for the 3-Phase Trajectory . . . 135
4.1.2.3 Preparing the Multi-Objective Optimisation Process of the 3-phase ATO . . . 140
4.1.2.4 Optimisation Progress . . . 151
4.1.2.5 Comparing the Variables and Objectives . . . 154
4.1.2.6 Analysing Aircraft Trajectories . . . 155
4.1.2.7 Identified Issues . . . 161
4.2 Micro-Mixing Optimisation of Microreactor . . . 163
4.2.1 Problem Description . . . 163
4.2.2 Methodology . . . 164
4.2.3 Preparing the Multi-Objective Optimisation Process of the Reaction Mixing . . . 169
4.2.4 Optimising Environmentally-friendly devices - Results and Discussion . . . 173
4.2.5 Identified Issues . . . 196

**5 Concluding Remarks and Discussion** **199**
5.1 Main Research Contributions . . . 199
5.2 Discussion of Findings . . . 200
5.3 Main Conclusions . . . 204
5.4 Assumptions and Limitations . . . 206
5.5 Future Work . . . 210
5.5.1 Improving the Methods . . . 210
5.5.1.1 Improving Optimisation Process and Procedures . 211
5.5.1.2 Improving Process and Procedures for Aircraft Trajectory Optimisation . . . 213

5.5.1.3 Improving Process and Procedures for Multi-Disciplinary Design Optimisation of the Microreactor . . . . . 215
5.5.2 Expanding the Applications . . . . . 215
5.5.2.1 Future Aircraft Trajectory Studies . . . . . 216
5.5.2.2 Future Multi-Disciplinary Design Optimisation Studies . . . . . 218
5.5.3 Improving the Tools . . . . . 220
5.5.3.1 Improving the Optimiser . . . . . 220
5.5.3.2 Improving the Flow Solver . . . . . 224
5.5.4 Long-Term Extensions . . . . . 225

**References** **227**

**Appendices** **257**

**A Scalability of GPU-MOTS2** **258**

**Corrections** **258**

**A Rebuttal - TEMPORARY Chapter** **263**
A.1 Statement of Thesis Deficiencies . . . . . 263
A.2 Common Corrections . . . . . 266
A.3 Corrections suggested by Dr Parks . . . . . 267
A.3.1 Generic Corrections . . . . . 267
A.3.2 Specific Corrections . . . . . 269
A.3.3 Annotated Corrections . . . . . 274
A.3.4 Other rebuttal comments . . . . . 275
A.4 Corrections suggested by Dr Sethi . . . . . 275
A.4.1 Corrections in Abstract . . . . . 275
A.4.2 Corrections in Introduction . . . . . 276
A.4.3 Corrections in Literature . . . . . 278
A.4.4 Corrections in Computational Tools . . . . . 279
A.4.5 Corrections in Applications . . . . . 281
A.4.6 Corrections in Conclusions . . . . . 284
A.5 Other Corrections, Introduced by the Author . . . . . 284

# List of Figures


1.1 Abstract Optimisation Methodology . . . . . 8

2.1 GATAC Top Level Design [167] . . . . . 25

3.1 MOTS2 Architecture . . . . . 38
3.2 Tabu Search Memories . . . . . 41
3.3 MOTS2 Algorithm Flow Diagram . . . . . 45
3.4 An Instance of MOTS2 on a 2 Decision Variables and 2 Objective Optimisation . . . . . 47
3.5 Hooke and Jeeves Move in Decision Space . . . . . 49
3.6 Methodology for Verifying MOTS2 against Test Functions . . . . . 59
3.7 ZDT1 EAF . . . . . 62
3.8 ZDT2 EAF . . . . . 62
3.9 ZDT3 EAF . . . . . 62
3.10 ZDT4 EAF . . . . . 62
3.11 ZDT6 EAF . . . . . 62
3.12 Normalised Hypervolume Boxplots for ZDT Test Functions . . . . . 62
3.13 Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT1 Against State-of-the-art Optimisation Algorithms [220] . . . . . 63
3.14 Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT3 Against State-of-the-art Optimisation Algorithms [220] . . . . . 64
3.15 Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT6 Against State-of-the-art Optimisation Algorithms [220] . . . . . 65
3.16 Methodology for Validating MOTS2 against Low Fidelity Physics . 66
3.17 XFOIL EAF . . . . . 70
3.18 Normalised HV Boxplots for XFOIL . . . . . 70
3.19 Parallel Evaluation Scheme on GPU . . . . . 78
3.20 Methodology for Verifying the Scalability of GPU-MOTS2 against a Single Test . . . . . 81
3.21 Scalability of GPU-MOTS2 when the Number of Variables Increases 83

3.22 3 Dimensional 19 Microscopic Densities - D3Q19 Model . . . . . 92
3.23 GPU-LBM Class Diagram . . . . . 104
3.24 GPU-LBM Sequence Diagram . . . . . 105
3.25 a) Perspective View of the Microreactor Model, b) Profile View of the Microreactor Model in Physical Units (not to scale) [45] and c) Profile View of the Microreactor Model in Lattice Units (not to scale) [46] . . . . . 108
3.26 View of the Baffle Plate in Lattice Units [64] . . . . . 109
3.27 The Flow Pattern between Experimental Measurements and Simulation are Qualitatively Compared . . . . . 110
3.28 Vector Field . . . . . 111
3.29 Vector Field with Scaled Vectors . . . . . 111
3.30 Velocity Profile of the Microreactor, perpendicular to the Flow . . . 112
3.31 Velocity Profile along the Microreactor . . . . . 113

4.1 Methodology to Compare MOTS2 against NSGAMO for ATO in the Climb Phase . . . . . 120
4.2 Trajectory Modelling . . . . . 121
4.3 Outline of Optimisation Process on an Instance of a 3-Segment Flight 122
4.4 Comparing the Trade-Offs and the Compromise Designs . . . . . 126
4.5 Generated Trajectories . . . . . 128
4.6 Design Variables to Objective Functions Relationships . . . . . 131
4.7 Mapping of Decision Space to Objective Space of the PF in Parallel Coordinates Projection . . . . . 132
4.8 Methodology for Optimising a 3-Phase Trajectory by using MOTS2 140
4.9 Engine Model Schematic . . . . . 141
4.10 HERMES Aircraft Performance Model Flow Chart [263] . . . . . 145
4.11 Calculating the Corrected Emission Index of NOx at Altitude by using the Emission Model [272] . . . . . 147
4.12 MOTS2 Search Progress in 3-phase ATO . . . . . 152
4.13 Parallel Coordinates Projection of the Objective Space of 3-phase ATO . . . . . 153
4.14 Comparing the Variability from HISTORY and PF with Reference to Tables 4.5 and 4.8 . . . . . 156
4.15 Relative Objectives' Improvement . . . . . 157
4.16 Comparing the Trajectory Altitude Profiles . . . . . 159
4.17 Comparing the Trajectory Speed Profiles . . . . . 159
4.18 Zooming trajectory altitude profiles . . . . . 160
4.19 Methodology for Comparing MOTS2 and GPU-LBM against the Original Multi-Objective Tabu Search and the Original LBM . . . . 165
4.20 Layout of Optimisation Process on Microreactor Mixing . . . . . 170

4.21 Coverage of Decision Space (from HISTORY) . . . 174
4.22 Optimum Decision Space and Objective Space (from MTM) . . . 175
4.23 Histogram of first Decision Variables of the Optimal and Complete Dataset . . . 177
4.24 Histogram of Second Decision Variables of the Optimal and Complete Dataset . . . 177
4.25 Histogram of Third Decision Variables of the Optimal and Complete Dataset . . . 177
4.26 Comparing the Pareto-Front (PF) between the Current Implementation and the Original one from [64] . . . 179
4.27 Optimisation Search Pattern at 1877 Total Evaluations . . . 182
4.28 Optimisation Search Pattern at 3805 Total Evaluations . . . 182
4.29 Optimisation Search Pattern at 7711 Total Evaluations . . . 183
4.30 Optimisation Search Pattern at 9676 Total Evaluations . . . 183
4.31 Optimisation Search Pattern at 11743 Total Evaluations . . . 184
4.32 Complete Optimisation Search Pattern, PF and Selected Designs (listed in Table 4.13) . . . 184
4.33 PF Progress (Related to Figs. 4.27 to 4.32) . . . 185
4.34 The Progress of the PF at the Beginning of the Search, as an Interim Snapshot of the Trade-Off from Fig. 4.33 . . . 186
4.35 Monitoring the Number of Invocations of the Provided Moves Throughout the Optimisation Search . . . 188
4.36 The Activity of the Optimiser Through the Number of Consecutive Unsuccessful Iterations . . . 188
4.37 Zooming into the Number of Invocations of the Provided Moves Throughout the Optimisation Search . . . 190
4.38 Baffle Plate for the Datum Design . . . 192
4.39 Baffle Plate for the Design with Maximum Vorticity Magnitude . . . 192
4.40 Baffle Plate for the Compromise Design . . . 192
4.41 Baffle Plate for the Design with Minimum Difference in Total Pressure 192
4.42 Trade-off of Selected Designs while Varying Flow Speed (the Flow Speed Increases when Moving from Left to Right) . . . 193
4.43 Velocity Contours in the mid Plane of the Datum Design . . . 194
4.44 Velocity Contours in the mid Plane of the Design with Maximum Vorticity Magnitude . . . 194
4.45 Velocity Contours in the mid Plane of the Compromise Design . . . 195
4.46 Velocity Contours in the mid Plane of the Design with Minimum Normalised Difference in Total Pressure . . . 195

A.1 Scalability of the Elapsed Time of GPU-MOTS2 on ZDT1 when the Number of Variables Increases . . . 258

A.2 Scalability of the Elapsed Time of GPU-MOTS2 on ZDT2 when the Number of Variables Increases . . . . . 259
A.3 Scalability of the Hypervolume Indicator of GPU-MOTS2 on ZDT1 when the Number of Variables Increases . . . . . 260
A.4 Scalability of the Hypervolume Indicator of GPU-MOTS2 on ZDT2 when the Number of Variables Increases . . . . . 261

# List of Tables


3.1 Optimisation Configuration Settings for ZDT Benchmark Functions 60
3.2 Features of ZDT Functions . . . . . 60
3.3 MOTS2 Configuration Settings for Airfoil Shape Optimisation . . . 69
3.4 GPU Hardware Specifications to Benchmark the Performance of MOTS2 . . . . . 77
3.5 GPU-MOTS2 Configuration Settings for Test Functions . . . . . 82
3.6 GPU-MOTS2 Configuration Settings for Test Functions to Study the Interactions of GPU Hardware and Algorithm Configuration . . 85
3.7 GPU Hardware Specifications to Simulate the Microreactor . . . . . 106

4.1 Climb Flight Envelope for 4-Segments Aircraft Trajectory Optimisation (ATO) . . . . . 124
4.2 Optimisation Configuration Settings for Aircraft Trajectory . . . . . 125
4.3 Principal Component Analysis of the Decision Variables that Belong to the Optimal Trade-Off . . . . . 129
4.4 Hypervolume Indicator . . . . . 130
4.5 Flight Envelope for 3-phase Aircraft Trajectory Optimisation (ATO) and Mapping to Optimisation Problem Variables . . . . . 137
4.6 Engine Design Specification . . . . . 142
4.7 Aircraft Model Specifications . . . . . 144
4.8 Specification of Variables for 3-phase Aircraft Trajectory Optimisation (ATO) with Reference to Table 4.5 . . . . . 149
4.9 Multi-Objective Tabu Search 2 (MOTS2) Configuration Settings for 3-phase Aircraft Trajectory Optimisation (ATO) . . . . . 151
4.10 MOTS2 Configuration Settings for Microreactor Optimisation . . . 173
4.11 MOTS2 Configuration Settings to Optimise the Microreactor . . . . 173
4.12 Hypervolume Indicator for Microreactor Mixing . . . . . 179
4.13 Design and Performance Metrics for Selected Designs from Fig. 4.32 192

*This page is intentionally left blank.*

# Listings


3.1 MOTS2 Top Level Pseudocode . . . . . 38
3.2 STM Container Class Pseudocode . . . . . 40
3.3 Container Class Pseudocode . . . . . 41
3.4 LTM Container Class Pseudocode . . . . . 43
3.5 Hooke and Jeeves Move Pseudocode . . . . . 49
3.6 Pattern Move Pseudocode . . . . . 51
3.7 Intensify Move Pseudocode . . . . . 53
3.8 Diversify Move Pseudocode . . . . . 53
3.9 Reduce Move Pseudocode . . . . . 54
3.10 Pseudocode of the Method to Activate Kick . . . . . 56

*This page is intentionally left blank.*

# Acronyms and Nomenclature

**ALT** altitude . . . . . 120

**APM** Aircraft Performance Model . . . . . 118

**ATC** Air Traffic Control . . . . . 19

**ATM** Air Traffic Management . . . . . 2

**ATO** Aircraft Trajectory Optimisation . . . . . 2

**BP** Base Point . . . . . 40

**CAS** Calibrated Airspeed . . . . . 123

**CFD** Computational Fluid Dynamics . . . . . 3

**CPU** Central Processor Unit . . . . . 18

**CUDA** Compute Unified Device Architecture . . . . . 15

**D3Q19** 3 dimensional 19 microscopic densities . . . . . 89

**DACE** Design and Analysis of Computer Experiments . . . . . 212

**EAF** Empirical Attainment Function . . . . . 59

**EPM** Engine Performance Model . . . . . 118

**FMS** Flight Management System . . . . . 21

**FPA** Flight Path Angle . . . . . 120

**GATAC** Green Aircraft Trajectory under ATM Constraints . . . . . 24

**GPU** Graphic Processor Unit . . . . . 7

**HISTORY** History Memory . . . . . 40

**HPC** High Performance Computing . . . . . 15

**ICAO** International Civil Aviation Organisation . . . . . 146

**IM** Intensification Memory . . . . . 40

**LBM** Lattice Boltzmann Method . . . . . 7

**LTM** Long Term Memory . . . . . 40

**MOO** Multi-Objective Optimisation . . . . . 6

**MOTS2** Multi-Objective Tabu Search 2 . . . . . 7

**MTM** Medium Term Memory (full Pareto Front) . . . . . 40

**NS** Navier-Stokes . . . . . 14

**NSGAMO** Nondominated Searching Genetic Algorithm Multi-Objective . . . . 22

**NSGA-II** Nondominated Sorting Genetic Algorithm II . . . . . 22

**PCA** Principal Component Analysis . . . . . 128

**PF** Pareto-Front . . . . . 28

**RBF** Radial Basis Functions . . . . . 212

**SBX** Simulated Binary Crossover . . . . . 22

**SIMD** Single-Instruction Multiple-Data . . . . . 91

**SIMT** Single-Instruction Multiple-Threads . . . . . 75

**SM** Symmetric Multiprocessor . . . . . 72

**SPD** speed . . . . . 120

**SS** Search Step . . . . . 48

**SSRF** Search Step Retain Factor . . . . . 68

**SRS** Software Requirements Specifications . . . . . 34

**STM** Short Term Memory (Tabu) . . . . . 40

$avgD$ average density

$blockId.x$ block index in CUDA

$blockDim.x$ dimension of blocks in CUDA

$CO_2$ Carbon Dioxide . . . . . 5

$did$ domain index

$C_D$ drag coefficient . . . . . 67

$C_L$ lift coefficient . . . . . 67

$e_i$ lattice velocity

$gid$ global thread index

$f_i$ distribution of particles for density $i$

$f_i^{eq}$ distribution of particles in equilibrium for density $i$

$i$ microscopic density index

$K_n$ Knudsen number . . . . . 17

$LX$ size of computational domain on X-axis

$LY$ size of computational domain on Y-axis

$LZ$ size of computational domain on Z-axis

$lu$ lattice unit ..... 101

$NO_x$ Nitrous Oxides ..... 4

$r$ hole radius in the baffle ..... 107

$P_0$ total pressure ..... 171

$Re$ Reynolds number ..... 106

$s$ spacing in the baffle ..... 107

$t$ temporal index

$threadId.x$ thread index in CUDA

$u$ macroscopic velocity

$w_i$ weights for density $i$

$x$ spatial index on the X-axis ..... 112

$y$ spatial index on the Y-axis ..... 112

$z$ spatial index on the Z-axis

$\Delta P_0$ difference of total pressure ..... 171

$\Delta t$ time step

$\rho$ macroscopic density

$\tau$ relaxation time

$\omega$ vorticity . . . . . 171

**SoTD** Statement of Thesis Deficiencies . . . . . 263

**P2** Revised Assessment By Dr Parks . . . . . 263

**S2** Revised Assessment By Dr Sethi . . . . . 263

# Chapter 1

# Introduction

## 1.1 Background

The high demand to transfer more passengers and goods along with expanding global economies have been driving the transport trends [1]. Among others, air transport, as a sector, has been growing very fast over the last 40 years [2]. Despite numerous incidents (e.g., market crash), the volume of passengers and goods to be transferred by air has increased tenfold and fourteenfold, respectively, which has a direct impact on the environment and energy use [3]. Economically, in terms of return on investment, airlines have the lowest rate among many other industries [4]. Environmentally, a four-pillar strategy was set to achieve the emission reduction goals, where the first and second items of the list are related to aircraft technology (i.e., airframe, engines and sustainable biofuels) and flight operations [5]. The faster any solutions become available, the greater the benefit for all the stakeholders and the quicker the recovery for the planet will be [6]. Hence, dedicated efforts to save fuel and costs are required in every phase of an aircraft's life cycle and operations. The main challenges for air transport [7], which highly conflict with each other, are aviation growth, climate change, air quality, aircraft noise, and sustainable development. These problems could be addressed by using technical methods and computational tools [8, 9].

There are various procedures so as to accommodate the aforementioned challenging goals [10]: a) decrease the overall number of operations, b) change aircraft type/technology, c) alter aircraft trajectories. As discussed in [11–15] and by considering the statistics in [12], the first option is unlikely to happen, as traffic tends to continuously increase [16–18]. Therefore, a combination of the last two options seems a viable approach with respect to the initial problem.

The tools and methods developed in this research are expected to assist aviation (and similarly related industries) at the operational and research level. A

multitude of stakeholders could find interest in the products of this research. Airlines could use their existing resources in an optimal way while accomplishing both internal and/or external objectives and/or other criteria. Manufacturing industry and academia will possess an alternative way to perform further research studies and analyses, which will increase their potential and knowledge span at the research level. Individuals and researchers could carry out their own projects, which will enable them to be more productive and will give them additional capability and flexibility to effectively tackle their problems. Ultimately, in the long term, following years of technological improvement, research bodies that focus on the legislation and protection of the environment (e.g., Air Traffic Management (ATM)) could experiment new strategies and design for a better society.

### 1.1.1 Improving Operations by Optimising Aircraft Trajectories

Aircraft Trajectory Optimisation (ATO) could be readily performed as a more promising short-term solution and is expected to reshape the next generation of ATM [19]. As a concept, ATO includes all the means and processes, so as to improve the flight path of an aircraft. ATO is strongly related to ATM because it can enhance the mechanisms to coordinate a range of actions from issues related to the flight of a single aircraft to issues concerning international airspace [6]. Operational improvements include a variety of methods of flying, maintaining and loading the aircraft as well as ATM procedures and systems. Frequently, these solutions attempt to maximise fuel efficiency by minimising flight distance, optimising cruise altitude and speed [20]. This is a straightforward technique that could considerably reduce the effect of aircraft operations on the environment with most minimal change(s) to the stakeholders of the aviation industry (aircraft, airport, human resources, other equipment, etc.). By applying optimisation methods, a range of solutions for real-world problems of ATO can be obtained [21]. Performance-based navigation, a wider concept that uses area navigation as discussed in [22, 23], is expected to be an on-going issue until 2020 [24], towards optimising the airspace in order to meet high demand requirements and future needs, even in pilot trials [25].

### 1.1.2 Evolving Aircraft Technology by Optimising Micro-Mixing

Changing the type of the aircraft is a massive task that requires huge amounts of time, resources and encompasses potential risk; so it seems to be an alternative long-term solution, because of the high economic and temporal cost [20]. Although the current technology is considerably advanced, relative to the beginning of avi-

ation, it has not reached the level of maturity that would allow any scaling in terms of the number of operations. Technological improvements involve changes in the engine and airframe in terms of design and performance [26], the development and use of alternative fuels [27]. As a single cost item, the fuel is responsible for 30% to 40% of total costs [24], where technologies that aim to reduce fuel consumption should also be cost-efficient either in terms of initial investment or maintenance costs. In terms of using alternative fuels, where liquid hydrogen has the largest span for new types of aircraft before 2020, 10% of the fuel used by aircraft is expected to be an alternative fuel by 2017 [24,28]; there is a great uncertainty behind this technology, which ought to be de-risked by carrying out further research. A significant drive in the evolution of aviation is the advance of Computational Fluid Dynamics (CFD) as discussed in [29].

Investigating alternative energy technology for aircraft has been an on-going topic since the beginning of aviation. Several advanced technological alternatives are predicted to improve the environmental performance of commercial aviation between the years 2030 and 2040 [30,31]. As expected, fairly soon after the commercial launch of aircraft for civil flight, several studies have discussed alternative ways of addressing environmental and other concerns, mainly with respect to technology [32]. Similarly, environmental concerns and political pressure concerning the future of aviation have been expressed in [33]. Since then, investigating alternative types of fuel, better combustor design, better materials, better tools and methods, and reduction of noise and emissions have been important topics.

Using hydrogen is a viable solution in the long run in order to address environmental concerns along with energy dependency [34,35]. One of the suggestions in [15] was to use hydrogen-powered aircraft, where hydrogen will be used as a fuel by the combustor of the aircraft engine. In general, combining hydrogen with diesel or using lean mixtures of hydrogen in experimental testings have reduced the emissions, but in extreme cases emissions slightly increased [36,37]. Compared to kerosene, hydrogen weighs less [38]. However, a larger volume is required to store enough fuel for operational purposes, and it is very expensive to produce and the efficiency is reduced [39,40]. More importantly, hydrogen aircraft configurations have a minimal environmental footprint [34,35,39–41]. This concept is further explained in [34], where new designs are presented with very promising results for the future of aviation.

However, in order to use hydrogen on conventional engines, modifications are required to achieve a better performance with respect to the environmental impact [34]. For instance, implementing cryogenic storage on hydrogen tanks will reduce the size of tanks and could enable existing aircraft to use hydrogen. In general, this technology has improved over the years and seems a viable solution for achieving environmentally-friendly combustion [34,42]. Nevertheless, using hydro-

gen effectively depends on the geometry of the combustor as shown in [34, 39, 40]. The shape of the combustor can affect the mixing of fuel and air [39, 41]. High turbulence makes sure that the fuel is well mixed with air giving a high flame speed, which is of major importance when minimising emissions [43]. Hence, environmentally-friendlier combustion depends on micro-mixing, which obviously requires small-scale devices [44].

Fundamentally, in order to enable hydrogen-powered aircraft, small devices that perform mixing at small scale are required, such as microreactors. A *microreactor*, also known as a miniaturised reactor, is a small-scale device that can mix reacting agents by molecular diffusion with applications in energy generation, medical diagnostics, drug discovery, etc. It is a promising solution, whose mixing capability has been investigated experimentally, where it is clearly stated that the geometry of the reactor needs to be optimised to improve the mixing capability [45] and a number of concepts have been explored via simulation [46]. Bringing together computational and experimental methods will be very beneficial for the development process of environmentally-friendly devices [47]. However, specialised tools and methods are needed to design such devices and to study their performance, because of the complexity of natural phenomena that take place at such a small scale [46, 48, 49]. When designing the shape of the microreactor, two important factors to consider are the mixing capability and the pressure losses, which both ought to be optimised at the same time.

In order to reduce gaseous emissions, moving away from stoichiometric combustion (i.e., the ideal combustion that combines the correct amount of air and fuel), which is related to high flame temperature, could be an alternative approach [50, 51]. The challenge in combustor design is to reduce the high flame temperature oxidation of nitrogen [50], which is mainly responsible for Nitrous Oxides ($NO_x$) emissions. By providing a more homogeneous fuel/air mixture while operating at a lean fuel/air ratio, $NO_x$ emissions decrease dramatically because of the low temperature of the flame and the complete removal of hot spots in the combustion area [52]. So, lean or rich combustion is required to reduce the high flame temperature. The air has to be redistributed to minimise the liner cooling flow and up to 70% of the combustor air flow has to be premixed with the fuel. By using premixing, an increasingly homogeneous mixture is formed that prevents high-temperature concentration. Consequently, better mixing requires a combustor design with a high swirl. In addition, in premixing, auto-ignition and flashback might occur when the combustor pressure is high. Different types of combustors have been invented, among which lean premixed prevaporised has the potential to reduce $NO_x$, up to 70% on take-off, compared to a standard reference. In lean premixed prevaporised combustor, flows are separated for fuel atomisation and fuel mixing and evaporation, which achieves high mixture uniformity, but at

the risk of auto-ignition and flashback. Micro-mixing is the technology that can achieve mixing without the aforementioned risks. Hydrogen has wider stability limits and can burn lean without approaching the lean blowout limits, in which case it reduces flame temperature and consequently reduces $NO_x$ emissions. Additionally, hydrogen-based combustion has no Carbon Dioxide ($CO_2$) emissions at aircraft mission level [1]. In such conditions, this may increase $NO_x$ emissions. All this demonstrates that achieving the right balance is very challenging, because of underlying physics. In general, large pressure losses may provide better mixing (for combustor mixture and also affect the size of the combustor). However, any pressure loss is a parasitic loss and will compromise the thermal efficiency and consequently the specific fuel consumption. Clearly, investigating the trade-off by using optimisation techniques is required in order to achieve the best possible mixing and low pressure losses at the same time.

### 1.1.3 The Role of Simulation and Optimisation

Several computational tools are used in aviation in order to make decisions to drive the costs down and minimise environmental impact [8, 9]. Until recently, the aviation industry was focused on safety, time efficiency and low costs. At the moment, noise, emissions and fuel reduction are gaining more ground in research, so as to generate sustainable solutions that cover both short-term and long-term environmental and economic targets [3, 8]. Computational methods are now contributing to the environmental impacts such as $CO_2$ and $NO_x$ emissions and aircraft trajectories. State-of-the art methods and the needs of aviation should be bridged [53], as the scientific progress in the multi-disciplinary analysis of aircraft flight is slow.

Many processes or procedures could be potentially optimised via simulation, so as to assist in a decision making process (or phase) of a problem [54]. In fact, many challenging problems in science and industry could be expressed as optimisation problems [55] and could be tackled by combining them with optimisation following the principles of systems modelling [54, 56]. Unfortunately, the concept of optimisation does not receive the industry's full attention, and, consequently, future extensions are not appreciated [6, 57]. In general, optimisation can lead to innovative solutions [58], followed by a return on investment, which is aligned with the concept of innovization [59] and the need to employ revolutionary solutions to meet demand in order to improve environmental compatibility, safety, affordability and reliability [6, 60]. Revolutionary technologies are anticipated to contribute in approaching aviation zero-emissions in the future [24]. This research is expected to contribute to the establishment and improvement of tools and methods in the field of optimisation via simulation, by suggesting solutions to address the identified

[1] However, the $CO_2$ footprint of the hydrogen production process has to be evaluated, too.

limitations that are described in the next chapter.

In the field of computer modelling and operational research, *optimisation* or *optimisation search* or *optimisation process* (the terms are used interchangeably throughout the text) denotes the search process that discovers alternative solution(s) subject to certain constraints. Frequently, in real-world applications, the optimisation process means to minimise time, cost, risk or to maximise profit, quality and efficiency. The optimisation problem consists of independent (decision) variables, also called parameters, which represent the degrees of freedom, and one or more objective functions, which represent the goals. The *decision point* or *decision vector* is the combination of decision variables, with reference to the optimisation process. The collection of the decision variables and objectives defines the *decision space* and *objective space* of the problem, respectively. In some cases, there are restrictions that define acceptable values of variables, objectives or other relations. These are called *constraints* and should be carefully handled, as they could dramatically increase the complexity of the problem. In the context of optimisation, *solution* or *optimisation solution* denotes the combination of decision variables along with the corresponding objectives/fitness/performance (via an objective function evaluation). In the context of *design optimisation*, the decision space is also called the *design space*.

## 1.2 Aims and Objectives

The aim of this research is to devise a family of engineering methods and benchmark them relative to state-of-the-art methods, so as to be applied to Multi-Objective Optimisation (MOO) studies in aviation with respect to the environmental and economic aspects. The products of this research (i.e., methods and tools) are developed to be freely used in a variety of ways by any number of stakeholders and could link to either research or commercial tools and methods.

Objectives:

1. Devise a methodology with the potential to be used in applications of ATO and micro-mixing optimisation.

2. Develop a prototype method to model the mixing capability of a microreactor that could potentially be used in fuel-saving technologies in aviation.

3. Benchmark and assess the aforementioned developments against state-of-the-art methods of ATO and design optimisation.

4. Demonstrate the optimisation methodology by generating optimal aircraft trajectories in terms of flight time, fuel consumption and emissions.

5. Demonstrate the optimisation methodology by generating alternative designs of a microreactor to improve the micro-mixing capability and environmental impact.

## 1.3 Overall Project Methodology

Principles of Optimal Design [61], Systems Engineering and Systems Modelling [56] are employed in the proposed methodology, as outlined in Fig. 1.1. More specifically, the methodology is an *optimisation methodology*, where the aim is to search for improvements in the performance of aircraft trajectories and micro-mixing by altering characteristic parameters. This includes a number of methods and tools, where a combination of them is employed in an instance/application, as required. Any method or the whole methodology can be repeated as many times as necessary, so to achieve a certain level of confidence. A few of the methods can also be employed during the optimisation process, so as to keep the user as much informed as possible.

The novelty of the methodology lies in the ability to interface with other systems, scalability and use of Graphic Processor Units (GPUs) to accelerate computationally intensive algorithms. The technology readiness level [62] of the methodology is assessed to be between 1 and 2, and it is possible to reach level 3, where feasibility can be proven by tackling numerous applications.

With reference to the objectives above, the following methods are suggested:

1. Multi-Objective Tabu Search 2 (MOTS2) and related tools and methods will be developed, so as to support the aforementioned optimisation methodology with the potential to be applied to applications at scale. As a metaheuristic optimiser, MOTS2 is expected to be able to deal with the requirements of real-world applications [55, 63]. For scalability purposes, the optimiser will be ported on GPU, so as to be able to handle a large number of decision variables in the future.

2. A flow solver based on the Lattice Boltzmann Method (LBM) will be developed and will be ported on GPU, so as to accelerate the execution.

3. The optimiser will be benchmarked against test function and reference problems. Furthermore, it will also be compared against a competitive optimiser of the same class and another, so as to identify cases for which it would be suitable.

4. The methodology will be integrated into representative models to study the performance of the optimiser under 2 and 3 objectives.

5. The methodology will be linked to the aforementioned fluids solver and will be applied to a reference study, where tools of the same class of optimisation and simulation were used, so as to compare and evidence any performance gains.

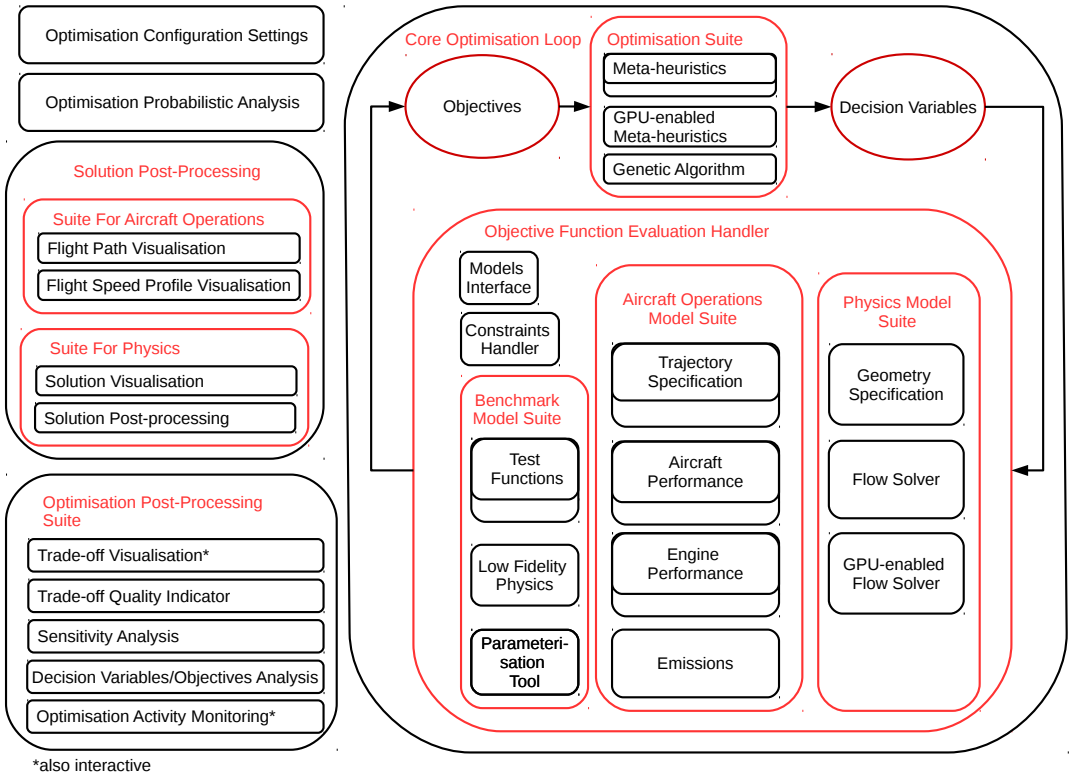



Figure 1.1: Abstract Optimisation Methodology

A description of the modules and methods follows, as depicted in Fig. 1.1:

**Optimisation Configuration Settings:** A collection of settings to specify the optimisation problem and to instruct the behaviour of the optimiser. These are specified to balance between exploration and exploitation of decision space, depending on the complexity of the optimisation problem.

**Optimisation Probabilistic Analysis:** A method to run multiple instances of an optimisation problem and acquire the generated trade-off, so as to visualise and assess the quality of the trade-off over all the runs.

**Optimisation Post-Processing Suite:** A collection of methods to analyse and interpret the optimisation results following the end of the optimisation, where:

**Trade-off Visualisation:** A plotting method to visualise the trade-off, as generated by the optimiser during the execution of the optimisation process.

**Trade-off Quality Indicator:** A method to numerically represent the quality of the trade-off by processing all the points of the trade-off. It is mainly used for automation (i.e., within the optimiser) and comparison purposes.

**Sensitivity Analysis:** A method to numerically represent the importance of decision variables.

**Decision Variables/Objectives Analysis:** A collection of methods to process and visualise the decision variables, and/or the interplay among the objectives, so as to identify relationships, which would be hard to detect otherwise.

**Optimisation Activity Monitoring:** A method to visualise the search progress of the optimiser.

**Core Optimisation Loop:** The main optimisation loop. Depending on the complexity, the computationally intensive parts of the simulation ought to be further restructured, so as to run on a platform that could speed up the simulation, such as GPUs.

**Optimisation Suite:** A collection of optimisers.

**Objective Function Evaluation Handler:** A collection of handlers, interfaces and models (where the decision variables are mapped via simulation of the underlying physics and/or business logic to objectives) for the optimisation problem, where:

**Constraints Handler:** An interfacing method between the optimiser and the Model Suites to detect the operating status of an instance of Model Suite and provide failed values for the objective function(s) in the case of a failure (e.g., when the Model Suite does not return the values of an objective function for a certain combination of decision variables). This is a fail-safe mechanism to protect any unexpected cease of operation of the optimiser.

**Benchmark Model Suite:** A method to verify the behaviour of an optimiser to compare the ability to generate a trade-off. The difference between the Test Functions and the Low Fidelity Physics is that the former have a well-known trade-off, whereas the latter's trade-off might not be known in advance. The Parameterisation Tool is a module that is required by the Low Fidelity Physics method, as explained in 3.1.7.1.

**Aircraft Operations Model Suite:** A collection of methods to carry out ATO.

**Physics Model Suite:** A collection of methods to carry out an instance of design optimisation.

**Solution Post-Processing:** A collection of methods to visualise, process and evidence aspects of an optimisation solution. Normally, this is employed to further analyse a number of solution(s) by using human insight and domain expertise, so as to identify special features that are responsible for the change in objectives and/or assess the quality of the optimisation solutions.

**Solution Post-Processing Suite For Aircraft Operations:** A collection of methods to visualise aspects of the target flight, where:

**Flight Path Visualisation:** A snapshot of the flight path, frequently represented as a 2D plot of range vs altitude.

**Flight Speed Profile Visualisation:** A snapshot of the speed profile of the flight path, frequently represented as a 2D plot of range vs speed.

**Solution Post-Processing Suite For Physics:** A collection of methods to visualise aspects of the target design by visualising a target geometry and flow features and/or post-analysis the former so as to gain a better insight.

## 1.4 Outline of Thesis

The remainder of this document is structured as follows:

Chapter 2 identifies limitations of existing methods by reviewing current state-of-the-art tools and methods in optimisation techniques, ATO and micro-mixing studies.

Chapter 3 describes in detail the development and benchmarking of a MOO algorithm and a prototype fluid solver for the purposes of the research. The architecture of the developed systems, implementation details and design decisions are discussed, too.

Chapter 4 presents three optimisation applications to demonstrate the developed tools and methods. First, an ATO application at the climb phase is performed, so as to minimise fuel consumption and flight time. The same methodology is then applied to a 3-phase ATO, so as to minimise fuel consumption, flight time and gaseous emissions. Third, while investigating for new aircraft technology, the shape of a microreactor is optimised, so as to study its flow mixing and other flow properties that reduce gaseous emissions.

Chapter 5 summarises the main findings of the research and recommends future research directions at various levels.

## 1.5 Contribution to Knowledge

By achieving the aforementioned objectives and identifying current limitations in state-of-the-art methods, the contribution to knowledge follows:

- A flexible optimisation methodology is devised that can be linked and applied to problems related to ATO and micro-mixing optimisation with 2 and 3 objectives. This is also benchmarked and assessed relative to state-of-the-art techniques in optimisation with known and unknown trade-offs. The ability of the methodology to reveal non-dominated trade-offs, also with reference to competitive optimisers, in 2 and 3 objectives is additionally demonstrated.
- The impact of deploying an optimiser on GPUs and carrying out a micro-mixing optimisation study where the flow solver is deployed on GPUs is assessed in terms of computational efficiency and speed up. In the latter case, the quality of the generated trade-off is also compared against a reference study [64].

During this research, software packages (see chapter 3, [65]), and the following technical report and articles were produced:

- C. Tsotskas, T. Kipouros, and A. M. Savill. Fast Multi-objective Optimisation of a Micro-fluidic Device by Using Graphics Accelerators. Procedia Computer Science, 51:2237 - 2246, 2015
- C. Tsotskas, T. Kipouros, and A. M. Savill. Multi-Objective Tabu Search 2: First Technical Report. Cranfield University, 2015, ISBN 978-1-907413-34-6
- C. Tsotskas, T. Kipouros, and A. M. Savill. The Design and Implementation of a GPU-enabled Multi-objective Tabu-search Intended for Real World and High-dimensional Applications. Procedia Computer Science, 29:21522161, 2014
- C. Tsotskas, T. Kipouros, and M. Savill. Biobjective optimisation of preliminary aircraft trajectories. In Evolutionary Multi-Criterion Optimization, volume 7811 of Lecture Notes in Computer Science, pages 741-755, 2013
- M. Razzaq, C. Tsotskas, T. Kipouros, M. Savill, and J. Hron. Multi-objective optimization of a fluid structure interaction benchmarking. Computer Modeling in Engineering & Sciences, 90(4):303337, 2013

*This page is intentionally left blank.*

# Chapter 2

# Literature Review

## 2.1 Initial Contributions

The concept of ATO had been discussed in [66–69], where the trajectory is modelled by using optimal control theory. Significant work in terms of theory, methodology, processes and techniques is presented in [68, 70–78]. All these approaches transform the optimisation problem into another type of problem and employ respective methods, usually called direct and indirect methods. However, a prior and deep insight of the nature of mathematics and physics is required so as to obtain a sensible solution by employing the optimal control theory, which cannot be taken for granted when dealing with real-world cases [69], especially in diverse environments. Obviously, the first action to reduce emissions is to cut down fuel consumption, as was also mentioned in [79], which, as expected, has reduced the negative impact of greenhouse gases.

Optimisation was employed very early to address real-world problems when critical decision(s) had to be made. The nature of real-world cases is such that very frequently multiple and conflicting criteria are involved, which leads to using specialised tools and methods. One of the earliest references in MOO can be found in [80], where cost, sensitivity, stability, risk, and irreversibility and other metrics were considered as separate objectives. Following MOO, the concept of Multi-Objective Decision Making was also discussed in [81], where a decision maker is called to attain more than one goal or objective in selecting the path of action, subject to a variety of constraints forced by the environment, resources and processes.

## 2.2 Computational Methods for Aircraft Technology for Mixing by using CFD

Over the years CFD has been proven as a tool of paramount importance in aviation [82]. It has enabled users to design, to analyse and to support high-performing and cost-efficient commercial transports. The ability to model (both aerodynamic and reactive) flows by using CFD has progressed considerably and changed the aerospace design process, while reducing testing requirements and physical designs at lower cost and risk. Moreover, it is foreseen that using CFD will contribute critically to performing more environmentally-friendly studies in the next few years [29].

Although there are numerous approaches to improving the technology of existing gas turbine engines [83], there are studies that investigate alternative ways to generate energy such as [27, 35, 41, 84–91], which depend on specialised software. The studied devices consist of many different components with very complicated geometrical arrangements that challenge conventional tools from the fields of CFD. Apart from new conceptual devices, conventional CFD tools are invaluable [92,93], but new tools will be required to make considerable progress [60, 93]. These tools are required to perform studies within acceptable time frames, as the current mechanisms are very time and resource consuming. When a new technology of energy generation in aviation is researched, usually standard tools and methods from the fields of CFD are employed. However, these tend to take considerable wall-clock time. In addition, setting up the case to handle complicated boundary conditions, multi-phase or multi-component flows is usually a very time-consuming task, which requires further processing time to finish the simulation.

When engineering applications require resolving fluid interactions in high accuracy, the LBM offers an alternative method for CFD instead of using the Navier-Stokes (NS) equations [94–96]. More specifically, it is a class of cellular automata that can approximate the NS equations to second order with an explicit collision-streaming scheme [97]. It is particularly suitable for problem instances that involve low Mach number flow, mesoscopic flows, complex geometrical arrangements and particular boundary conditions. Further studies considerably improve LBM's potential and capabilities making it a practical tool for engineering applications [98–103]. However, when it comes to LBM, the majority of the implemented solutions attempt to simulate a few of the benchmark problems, such as lid-driven cavity flow and others, for instance [104]. LBM is expected to be the next evolution in computational sciences [105]. It is not going to replace traditional CFD methods based on NS, but it can be considered as a competitive alternative that can occasionally (e.g., [106]) influence the generation of new approaches and needs to be further matured.

## 2.3 Accelerating Computational Methods of Computational Fluid Dynamics

Multi-disciplinary approaches that involved CFD did not change as rapidly as they could, while High Performance Computing (HPC) has rapidly evolved [29]. Physical modelling could not accurately predict the important flow features, as the geometries became more complex and the software became even more sophisticated, which gave rise to multi-disciplinary simulations. Furthermore, conducting high-precision studies is of paramount importance when researching on combustors that use hydrogen, but they are also computationally intensive [107]. Consequently, the time required to run a CFD simulation will impede to scale up the problem to simulate a more detailed aircraft model as the computational workload would increase and more complicated geometrical arrangements should be handled.

In general, adopting alternative software and hardware is still at an early stage before it becomes well established and is an ongoing topic widely discussed in many scientific conferences [108, 109]. Big vendors and institutions from the fields of computer modelling either have focused their efforts to develop tools that make use of that technology or they have already acquired them [110, 111]. Others have integrated them with their processes or have adopted them accordingly, such as [112].

The advance of recent GPUs in terms of higher processing throughput, larger memory bandwidth and their faster in-between inter-communication capabilities greatly contributes to the evolution of new tools and methods. The hardware includes highly parallel, many-core and multi-threaded processors and many hierarchical memories. In addition, Compute Unified Device Architecture (CUDA) is a programming language that has the ability to access GPUs' hardware and has reached a level of maturity, where programming is more flexible and robust than ever before. Hence, computationally intensive applications can run more efficiently even on very large scale problems [113]. Simply, it provides flexibility to directly and easily manipulate various levels and types of fine computational resources on GPUs. Due to the low overall cost and high computational efficiency, they suggest a decent alternative computational architecture that can sufficiently cope with the requirements of scientific applications, which increasingly require even more computational power [29, 114, 115]. Computationally, GPUs also have great cost-benefit: Their low energy consumption for the attained computational speed inspired the creation of the performance metric proposed in [116], where the computational power is combined with the energy required for performing a simulation.

The LBM is one of the alternatives with great potential for the future of aviation and a great fit for computational platforms based on GPUs, because it is

an algorithm whose logic is very well aligned with hardware operation of GPUs, as explained below and in greater detail in subsection 3.3.1. By definition, the concept of LBM is a memory-bound algorithm, which provides a good fit for the GPU architectures. Given a computational environment with many processors, GPU-enabled LBM codes could potentially run considerably faster without compromising accuracy [104, 117–129]. The execution of the LBM algorithm can be significantly accelerated when running on GPUs, for instance [123]. In general, the GPU-enabled LBM is a competitive counterpart in terms of both accuracy and execution speed. So far, significant execution speed-up and faster convergence rate have been achieved. Furthermore, for a relatively small computational domain it has the potential to perform a real-time simulation.

Available computational power and low cost will be a considerable factor in the evolution of aviation [93]. Employing alternative computational architecture on CFD-based applications can be an intermediate step towards the next generation of HPC and computational methods [29]. As already mentioned, there are several applications where GPUs have been proven useful in accelerating the execution of sub-system(s) for the purposes of aviation and many more are expected in the future. However, only a few of them are compared against real-world data and even fewer are coupled with other systems (e.g., [130]).

As usual, when implementing an HPC application, the most challenging issues are instruction-level-parallelism, memory management, power management, algorithmic dependencies, hardware dependences, data dependences, and other dependencies. Therefore, it is critical to design and develop the application following the principles of object-orientation to minimise any complications and allow for future extensions. However, as pointed out in [131], porting and upgrading every possible application to CUDA does not pay off for the attained speed-up. Another challenge lies in the field of intercommunication with other systems, which depends on legacy software and impedes coupling.

The performance of GPUs on many applications has inspired the creation of supporting tools and projects for even greater achievements. Other projects attempted to simplify and unify the implementation of applications for sustainable development [132]. Two applications in the fields of structural mechanics [131] and fluid mechanics [133] not only demonstrated increased computational performance, but they also introduced novelty in memory management (for instance, memory-fetching) and mixed-precision arithmetic, respectively. Similar projects are required so as to establish the use of GPUs on engineering applications.

For the purposes of advancing applications of aviation technology by using CFD methods, high-fidelity simulation technologies ought to evolve, which could potentially contribute to the achievement of future environmental goals [29]. More specifically, CFD methods should evolve along with HPC and should take advantage

of modern HPC infrastructure, where GPUs are very strong candidates. Several studies attempt to adapt applications of CFD with finite volumes to run on GPUs, such as [130, 134], but compared to LBM, the obtained speed-up is not very satisfactory. Successful work in this direction has been presented in [117, 119], where significantly high performance was achieved. Even in more advanced applications, such as [135], LBM can be computationally faster. From a theoretical point of view, when studying devices at micro-scale Knudsen number ($K_n$) rises and the NS equations cannot be used [136, 137]:

$$K_n = \frac{\lambda}{L} \tag{2.1}$$

where $\lambda$ denotes the mean free path of a molecule, and $L$ is the characteristic length. Moreover, the effect of $K_n$ will be very important when the application will be capable of resolving the phenomena of heat transfer, as discussed in [138, 139]. Simply, accurately predicting the characteristics of any micro-device that depend on temperature is a very challenging task. Hence, using the NS equations is not the best approach.

## 2.4 Optimisation Applications by using CFD

When it comes to designing technology for the future of aviation, there are several studies that carry out optimisation by using tools from the field of traditional CFD. Nevertheless, using LBM was first introduced in [64, 137] and these studies extend this work. However, the elapsed wall-clock time and resources required to complete such an application are not satisfactory for engineering applications. Specialised and expensive infrastructure is required, which is accessible to very specific organisations and users. Hence, the utilisation of resources, the cost of delay (with respect to the time required to complete one case) and the accessibility ought to be improved so as to scale up and be more practical. To the best of the author's knowledge, this is the first time such operational constraints are considered and these should be combined with a flexible software architecture so as to link with other applications, too.

An instance of computational intelligence in optimisation methodology can be found in [133], where a two-layer hierarchical optimisation code operates with variable precision. The lower level handles computations in single precision while exploring design space. Following a finite threshold, the optimisation algorithm switches to the double precision mode for the most promising regions, where significant accuracy is required. This strategy follows the fundamental principles of CUDA, where single precision operations are executed faster, so as to effectively utilise the computational efficiency of GPUs. As a compromise, mixed precision

could be used: On each equation, the left-hand-side and right-hand-side parts are calculated with a single and double precision, respectively. The latter presented the highest ratio of precision for computations, a very reasonable metric for engineering applications. The optimisation algorithm can also operate in both precision modes. It is important to mention that a few scientific references were found that combine design optimisation along with GPUs-enabled CFD tools and methods (e.g., [133]). However, none was found to be applicable to micro-mixing optimisation studies.

## 2.5 Accelerating Optimisation Methods

Based on the author's experience, multi-objective optimisers might degrade in performance because of the number of decision variables, the number of objective functions, the number and type of constraints. Also, in terms of process life cycle, the elapsed time to evaluate an objective function is also important, as computationally intensive objectives might need to be modelled [140], so as to speed up the optimisation. As pointed out in [93], the large number of variables (more than 4000) involved in a single CFD simulation while employing multi-disciplinary objectives dramatically increases the problem size. In addition, it is clearly mentioned in [141] that alternative parallelisation techniques (e.g. using GPUs) should be employed in the future. Hence, employing GPUs to accelerate the execution of the algorithm is a viable approach and further developed in subsection 3.2.

The potential of GPUs has also been demonstrated in optimisation applications. First, an instance of single-objective tabu search in Mixed Integer Programming was presented in [142]. Specifically, the quadratic assignment problem for more than 30 items was optimised, achieving higher throughput. The greatest gain was noticed for the pair-wise exchange and initial cost calculation routines, which are the most computationally intensive parts. Comparative to Central Processor Unit (CPU) implementation, 20 to 45 times faster results derived by porting the same code in CUDA, while the same results were obtained as if the code ran on a computational cluster. Algorithmic-wise, neighbourhood exploration and concurrent regions searching are mainly favoured. Still, minimising communication between CPU and GPU is a critical factor, which is alleviated in most recent generations. Secondary problems concerning the parallel tabu search are: unique search space allocation, threads operational overlapping, promising regions exploration and memory control. A variant of native multi-objective tabu search that operates on GPUs has been designed and has been implemented with the intention to be used on real-world applications with increased problem size. Many cost functions for different combinations of decision variables can be evaluated at the same time because of the fast throughput of GPUs. This capability will enable

users to run cases with thousands of variables within a reasonable time frame.

## 2.6 Optimising Aircraft Trajectories

The environmental impact can be minimised considerably by revising the ATM procedures such as lower cruise speed, arrival management and departure management, the continuous descent approach, noise preferential routes, noise abatement procedures and others described in [143]. A similar concept to ATO was presented in [144, 145]. The objective was to design an aircraft to fly on pre-specified routes. This is the opposite of this research, where the aircraft specification is given and the optimum route is to be discovered.

The exact sequencing and spacing required for maximum take-off and landing rates in peak hours cannot be manually maintained by Air Traffic Control (ATC) [10]. Introducing automatic procedures comprises a two-fold advantage: a) the noise impact and the performance of aircraft can be predicted, b) the controller can determine and maintain adequate sequencing and spacing. In this direction, the work carried out in [146] is helpful as the following methodology could be used for various aircraft. In this case, simulation data from a Boeing 737-200 were used in order to build a noise pollution model for ATO cases. Given a specific combination of aircraft orientation, thrust setting and altitude, the model delivers the footprint on the ground, which is exposed to noise levels at or above 70 dB. This work could be coupled with optimisation methods, so as to generate more environmentally-friendly plans to operate the airspace, which is heavily congested close to the airport.

The Ames Research Center of National Aeronautics and Space Administration developed the En route Descent Advisor, which is described in [147–149]. This is a decision support computer application that manages complex en route traffic subject to metering restrictions. Ultimately, the En route Descent Advisor will control procedures based on trajectory management, whereas this is currently based on sector management. According to the authors, both horizontal and vertical arrival trajectory optimisation can be performed, which results in more fuel-efficient arrivals. In general, a variety of concepts are supported, as described in [150], such as top of descent optimisation, user preferred routing and relaxed static metering fix restrictions [1]. The first improves the vertical descent of the flight profile by shifting the top of descent location further downstream, which tends to eliminate the endurance of less efficient (i.e., lower) altitudes. The second, independent of the path, facilitates a flow-rate conformance. The last introduces the horizontal and vertical anchor point. As a high-level tool, it could be combined with optim-

[1] During arrival, current conservative, static, restrictions can be relaxed so as to improve the vertical descent profile or horizontal arrival trajectory.

isation methods, so as to further improve the automation, which could also have a positive effect on the environment by minimising the flight time.

Several approaches were reviewed to performing ATO at the ATM level, where the objectives were to keep aircraft separated and to avoid dangerous weather conditions [151]. In the comparisons, 2D trajectories (i.e., vertical flight profile) were used and the ideas examined were part of the *free flight* concept. Again, the principles of control theory are employed. As expected, different methods would be used if airspace was continuous or discrete for resolving the objective, where one or more aircraft were considered. It was also shown whether the problem could be resolved by a centralised authority, usually ATC, or cooperatively by distributing this task among the aircraft flying in airspace (i.e., in multiple aircraft cases). Using a 4D description of the trajectory would be beneficial and it is highlighted that new optimisation approaches are required.

As already described above, ATO problems are governed by multiple disciplines [58, 152]: overhead, various types of costs, multiple performance metrics, further desired properties, etc. Besides academia and research centres, industrial sectors show high interest in trajectory optimisation as stated in [153]. Several studies focus on flight efficiency factors such as advanced noise and fuel consumption, advanced trajectory technologies such as prediction and management, and other aero-transport economics. Ultimately, all these technologies aim at mitigating environmental impact by relieving overcrowded airspace and airports [11,143]. In terms of computational modelling, the system consists of many different components with complex interactions among them.

When it comes to trajectory optimisation, the majority of research work takes for granted that the equations of motion, which describe the motion of an aircraft, would be available and a range of techniques from control theory is employed [68, 70], and more recently [154]. Simply, the analytic expression is given, which is very cheap to compute. Although this had a number of advantages, it cannot cope well with the modern needs of aviation projects, where multiple principles and stakeholders are involved, modularity is highly desirable, and the number of important factors/dimensions is very large.

Because this is expected to be challenging [155], coupling the systems as an assembly of black boxes is the suggested approach for a more flexible and extensible architecture. The development of such a system would allow one to experiment with a variety of other systems, occasionally provided by third parties, and will advance towards a more modular design, where several modules could be combined in a mix-and-match fashion, without requiring any previous knowledge of the existing system(s). This is expected to considerably advance multi-disciplinary studies very rapidly to a new level, where many different models could be coupled together without knowing anything about the architecture of the host system.

To the best of the author's knowledge, there is not any efficient way to generate at the tactical (and possibly operational) level the trajectory of an aircraft, given a number of constraints from ATM in a black box manner, ideally with up-to-date information from aeronautical information services. Usually, a number of trajectories are pre-planned and ranked, which are provided to the operator to select. Of course, there are many factors that could influence the trajectory, but these are not considered, as there is not any system to generate optimal trajectories within an acceptable time frame before the flight [156]. So, a unified and flexible system is required that would encompass a range of subsystems related to aircraft and ATM procedures. In that system, a number of variables could be fixed (such as the fuel load), and others could be set according to additional specifications that could be provided a few minutes ahead of the flight (such as payload weight, which contributes to the overall weight). Furthermore, industrial stakeholders are very reluctant to employ optimisation methods, mainly because the former have a limited understanding [57].

Selecting an optimal trajectory to satisfy a number of criteria for each of the millions of commercial flights that take place every year just in the EU [157] is a very challenging task. Obviously, the complexity, the frequency and the size of the problem are greatly augmented when considering a common worldwide airspace. Again, in the future, there will be a greater need to optimise trajectories in the heavily congested international airspace. Ideally, this should be performed, before the take-off with the most up-to-date information from aeronautical information services. Moreover, trajectories should be recalculated in the middle of the flight as an update for the most optimal route(s) because of the occasional changes in the airspace. This is for the benefit of all the stakeholders and for better coordination, which will lead to a more optimal management of the entire international air traffic. In addition, more information should be exchanged among the stakeholders, which will be used to precisely formulate the optimisation problem. This technology could also be part of the Flight Management System (FMS) and could even run in the background, while continuously providing the user optimal solutions. The modularity could have an extra effect: when the trajectory deviates, the remaining part of flight could be optimised as a sub-problem, similar to the recalculation of the route when a driver fails to follow satellite navigation instructions from a dedicated system. Of course, the problem size is more complicated, but the same idea applies. From the side of ATM, this will allow more effective air traffic flow management while achieving separation of aircraft and hitting environmental and operational targets. Eventually, more tailored trajectories could be generated.

Although many studies have used optimisation algorithms for ATO purposes, most of the approaches are intended for a research level and cannot be used in an industrial environment and rarely consider operational constraints. Regarding the

nature of optimisation problems, most of the times, single-objective optimisation (where very often a weighted composition of multiple objectives is used) problems are studied and the objective functions are expressed in an analytical form, e.g., [158, 159], which is far from real conditions. Among other studies, the work presented in [160] meets the aforementioned requirements, where aircraft trajectories are optimised by using genetic algorithms. However, it only considers two objectives. In an industrial environment, black-box models are provided, which cannot be expressed analytically, as will be further described in chapter 4. To the best of the author's knowledge, metaheuristic optimisers, as an alternative to genetic algorithms, have never been used on ATO applications with more than two objectives, which will allow their use to optimise for many principles.

Nondominated Searching Genetic Algorithm Multi-Objective (NSGAMO) was originally developed in [161] and is considered as a state-of-the-art genetic algorithm for MOO problems, appropriate for trajectory optimisation, as demonstrated in [160, 162, 163]. It is designed for lower computational complexity during non-dominated sorting using the concept of Elitism, which improves the ability to retain satisfactory solutions. Compared to the popular Nondominated Sorting Genetic Algorithm II (NSGA-II) [164], NSGAMO uses [161]:

- A different selection process, so as to form the mating pool,
- An entirely different sequence of genetic operators for mutation (i.e., dynamic vector mutate method and creep mutate with decay) and crossover (i.e., tri linear crossover and Simulated Binary Crossover (SBX) [165]).

The main flow of the algorithm consists of the following stages:

1. The algorithm begins with an initial population of N individuals and multiplies it with an initialisation ratio for the 1st generation.
2. This population is sorted based on the principle of constrained non-dominated sorting to form the initial generation $P_t$.
3. If after sorting, individuals exceed N, then N individuals are selected based on crowding distance from the final front.
4. Using constrained crowded tournament selection, a mating pool is created from $P_t$.
5. The genetic operators (mutation and crossover) are then used to form an offspring population $Q_t$.
6. On the merged set $R_t = (Q_t + P_t)$, steps 2 to 3 are performed to form the next generation.

Despite its suitability for ATO, fully understanding the meaning of the configuration parameter(s) of the optimiser (e.g., see Table 4.2 for demonstration and comparison purposes) is not a trivial task and requires familiarity with complex concepts, such as SBX. More importantly, based on the author's experience, selecting which (combination of) configuration parameter(s) to alter when facing a new challenge or just adjusting the settings of a previous application, so as to achieve/explore any better performance, is not a straightforward task. For instance, when the mutation operators are set to a high value, the optimiser mainly performs exploration and the information of strong individuals is lost, which could decrease the performance of the optimisation search. This steep learning curve could discourage new users from using the tool/method and could consume a lot of time to master, without any obvious benefit. Furthermore, carrying out any software development extensions can be even more challenging.

A brief description of the configuration settings of NSGAMO follows:

**% for creep mutate with decay** percentage of the genes range to apply creep mutation with decay, which shrinks the window of mutation as the optimisation progresses (the larger the value, the more aggressive the search)

**% for dynamic vector mutate** percentage of the genes range to apply dynamic vector mutation (the larger the value, the more aggressive the search)

**% covered dynamic vector mutate** bias to force the optimiser to apply dynamic vector mutation

**% covered for vector mutate** bias to force the optimiser to apply vector mutation

**% covered for element mutate** bias to force the optimiser to apply element mutation

**Convergence fitness tolerance** threshold for convergence criterion, if the difference between two consecutive generations is less than this value, then the optimisation search terminates

**Initialisation factor** is the number of random samples when starting the optimisation search

**Maximum generations** represents the number of iterations and the optimiser will stop if the aforementioned convergence is not achieved

**Population size** denotes the number of individuals within an iteration, which is related to the number of objective function evaluations

**Element mutation probability** probability to mutate a single gene

**Creep mutate probability** probability to apply creep mutation to a single gene

**SBX distribution coefficient** controls the spread of SBX crossover operator, it is suggested to be 1.0 from the original version of NSGA-II

**Selection pressure** preserves the genetic algorithm within the failure boundaries

In a collaboration between Cranfield University and the University of Malta, a framework called Green Aircraft Trajectory under ATM Constraints (GATAC) has been developed under CLEANSKY [166, 167], where the author contributed in providing the optimisation algorithm, described in section 3.1. This is a framework for ATO that includes a variety of models related to aircraft performance, engine-gaseous emissions, noise emission, optimisation algorithm, end-user front-end and supporting tools for addressing the goals of management of trajectory and mission. It has also the potential to be used in multidisciplinary design optimisation applications and its top-level structure is depicted in Fig. 2.1 The main goal was to produce a standard and easy-to-use tool to resolve ATO problems on simplistic computational infrastructure, such as commodity computers. This initiative, which started from CLEANSKY, brings together several monolithic [2] applications in a black box interfaced fashion, where the user needs to specify and to link inputs and outputs. GATAC was developed under the System for Green Operations and will be used in this work, too. The development was initially discussed in [166] and related work was demonstrated in [160, 162, 163]. This is an ongoing project and gradually more models will be integrated into the future versions. By the moment of writing of this document, the features (as described in [168]) related to information management and architectural design were not available.

[2] also referred to as stand-alone

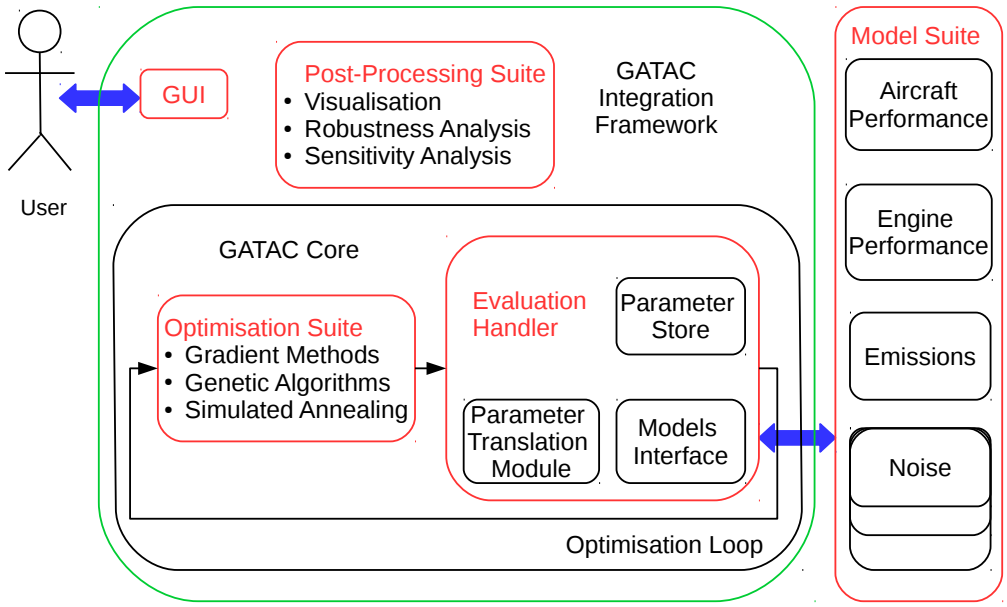



Figure 2.1: GATAC Top Level Design [167]

The integration of trajectory simulations along with optimisation algorithms under GATAC has been presented in [160, 162, 163]. At the time of the research, the framework provides a variety of trajectory simulation models but only one optimiser. This work expands the portfolio of optimisation algorithms available to users by adding a new optimiser. This is common practice and more flexibility is provided to the end-user to choose between the two optimisers for different cases, because no optimiser is equally good for every possible scenario [169].

At the time of writing this work, NSGAMO was already part of GATAC [167]. In order to improve the portfolio of optimisers GATAC, the author contributed by providing a new optimiser (presented in section 3.1), from the class of metaheuristic optimisers, which are known for their performance in real-world cases.

An optimisation framework was developed in [170, 171], to resolve aircraft performance and engine performance, and pollutants formation. Robust and feasible flight paths are generated by running full-scale simulations. The multi-objective case optimised time, fuel consumption, and pollutants emissions. Eco-friendly trajectories and engine cycles were finally obtained. The results were cross-checked against competitive optimisers and demonstrated minimal variations. Two more studies are available in [172, 173], where genetic algorithms optimise the flight segments.

The optimisation of noise abatement trajectories for departures and arrivals was studied in [174–178]. Both generic indices (e.g., noise footprints) and site-specific criteria, which consider the population distribution around the airports, were included in the studies. First, noise models and a geographic information system were integrated into a dynamic trajectory optimisation code [176]. Given any airport, the tool delivers the analysis and design of noise abatement procedures. Two objectives were considered (i.e., noise and fuel consumption) for the departure of a Boeing 737-300 from Schiphol (i.e., Amsterdam Airport). In the optimal trajectories, the noise affected about 35% fewer people compared to the reference fuel optimal trajectory, while requiring only 1% additional fuel. Following the previous case studies, the same tools and scenarios were also employed for the arrival of the aircraft in [177]. The aim was to produce noise optimal arrival trajectories. As a consequence, about 50% fewer people were disturbed compared to the reference, fuel optimal trajectory at the expense of about 15% more fuel and 10% longer endurance. In general, the shape of the descent profiles is very similar. However, an operational limitation in both of the aforementioned cases forced modifications as described in [178]: by slightly modifying the flight path, similar results were obtained. Similarly, it would be useful to combine noise indices with other principles. In [174], the previous tool was extended by including other noise performance criteria. Two extra parameters were added: population and area. By considering the arrival trajectories, the author performed parametric analysis of a single composite noise objective. Consequently, the single goal optimisation case illustrates unsatisfactory performance, because it is not possible to demonstrate a trade-off. This indicates one must use individual (conflicting) objective functions, so as to demonstrate a trade-off, which could be used by the decision makers. However, these studies focused on noise, whereas more objectives should be considered so as to acquire a trade-off.

So far, the previous studies expressed population awakening as a function of sound exposure level. This is expanded in the studies carried out in [179], where it is revealed that the objectives of the number of disturbed people against the maximum noise levels vary in harmony. Besides including the energy equivalent noise level, which is commonly used in such cases, the number of aircraft events is equally severe regarding nocturnal annoyance. It is noteworthy that the noise exposure qualitatively changed over the last two decades. Although the traffic in airspace increased, the emission levels per aircraft were reduced. This justifies people's complaints about an increase in noise throughout the same period of time, which is also cross-stated in [180]. Again, a multi-objective approach would provide a deeper insight.

A method to plan an environmentally-friendly trajectory for commercial aircraft by using ATO was presented in [154], where the considered objectives were

noise against emissions of $CO_2$. By following the original ATO style, the problem is transformed to an optimal control problem, where direct methods are applied. A differential algorithm was developed for the purposes of that project so as to generate fast non-dominated solutions between conflicting objectives. Many trajectories were produced that minimise environmental costs, which provides several options to the decision maker to make informed decisions. Two practical extensions would be to increase the complexity of the model, so as to improve the accuracy, and to increase the number of objectives, which would require changing the methodology so as to cope with the increased computational workload.

A methodology that includes the time dimension and generates 4D trajectories with multiple time restrictions was presented in [181]. Similar to other studies, the original problem is transformed to an optimal control problem and by using dynamic programming the operating cost of the flight as a single compound objective is minimised for the cruise phase of a flight. The authors suggest that manipulating precisely the length of the endurance of aircraft will significantly increase the capacity of airspace while preserving the current safety levels. In addition, neural networks are employed to decrease computing time by eliminating the number of cost calculations related to each decision step throughout the optimisation process. The direct operating cost is a combination of fuel and endurance objectives used for the optimisation. The results reveal a solution to the 4D cruise optimisation problem that significantly prevents high fuel consumption. This approach could also be used for on-line 4D trajectory optimisation as part of the capabilities of the aircraft's FMS and could potentially include window type constraints. In any case, increased computational power is required so as to carry out more complicated scenarios and allow higher precision tools to model the environmental impact.

## 2.7 Optimisation and Multi-Objective Optimisation

Optimisation problems are classified in a number of ways. The most frequent cases will be considered, as they are sufficient for the ATO cases. More details can be found in [182, 183]. As a suggestion, several computationally based design optimisation methods and techniques are outlined in reference [184]. Firstly, one can find problems in which the nature of input variables is partially or fully non-numeric. These problems fall into the category of optimal selection problems and can be resolved by applying a systematic computational mix-and-match procedure also called *combinatorial optimisation*. Hence, they are part of an entirely distinct set of methods which will not be discussed further because different types

of algorithms and strategies are required to tackle these problems.

The choice of the most appropriate optimisation algorithm depends on the nature of the optimisation problem, and often human experience and deep understanding are required. Therefore, it is obvious that there is not a single algorithm which could tackle every case equally well. This indicates hybrid algorithms that will have to be composed combining features from various classes [185], such as [186]. In addition, quick algorithms are always required and desired at the earliest possible stage of a design process. This affects the optimisation process directly. In order to obtain a satisfactory solution quality or even just a solution, an appropriate optimisation algorithm should be selected. Certain types of problems should be tackled by employing specific types of optimisers [169].

Metaheuristics have been effectively adapted to a variety of problems with continuous variables, nonlinear and noisy objectives, and highly complicated interactions between decision variables and objectives. These have demonstrated satisfactory performance on a variety of engineering problems and can yield satisfactory solutions within acceptable time frames. They are a special class of optimisers that among others include tabu search, evolutionary algorithms and particle swarm optimisation. The first is a local-search-based optimiser and belongs to single-solution-based metaheuristics, whereas the latter are global-search-based optimisers and belong to population-solution-based metaheuristics. Nevertheless, metaheuristics could rapidly resolve problems of high complexity, and this is a critical factor in their overall performance [55, 187, 188].

In contrast to the optimisation algorithms and iterative methods, metaheuristics could not necessarily find a globally optimal solution to certain classes of problems [55, 187, 189], for instance, where the parameters need to change and the number of permitted evaluations is limited [190]. Several metaheuristics implement a form of stochastic optimisation so that the discovered solution depends on the set of random variables generated [191]. Although combinatorial optimisation problems are not considered here, the stochastic nature of metaheuristics allows them to easily expand to this class of problems [190]. Hence, they can be considered for large-scale optimisation problems, and they are selected for this research.

By the definition of the MOO problem, selecting the final solution(s) among the ones that lie on the Pareto-Front (PF) is not an easy task and requires human intuition, and this belongs to the decision-making process. Depending on the problem's specification either a single or a set of solutions must be resolved. For example, in the gas turbines industry, a single high-performance engine needs to be manufactured for several types of aircraft, whereas in the automotive industry a variety of car models from the same firm needs to be produced. Telling the difference among the best solutions cannot be automated and needs a good

understanding of the problem; this is true because the PF represents the boundary beyond which any improvement of the overall performance can be achieved only by degrading one or more of the objective functions. Because the shape of the trade-off does not always reveal the optimum practical solution(s), a decision maker is required in order to find them among the discovered Pareto-equivalent ones. Therefore, the MOO process necessitates both an optimisation algorithm and a man-in-the-loop in order to resolve the ultimate solution.

Alternatively, constructing the optimal set by using genuine MOO algorithms is an advisable and viable way [183]. These algorithms manipulate either single solutions or populations of solutions and discover multiple compromise solutions within a single run, irrespective of the features of the objective functions such as modality and convexity. Usually, the sets of solutions are processed in parallel and evolve towards the final PF. During algorithmic iterations, the currently discovered solutions are compared to each other according to domination criteria [192]. Newly discovered solutions with better fitness form the current optimal set and the process continues until stopping criteria are met.

Unearthing the concepts of multi-objective optimisation, there are several features to be discussed. Throughout the iterations, the best solutions are selected and copied into the next population. This is called elitism and usually yields satisfactory results [193]. Some algorithms auto-adjust the population size they investigate at the end of every iteration so as to explore only the area(s) of objective space with the most promising solutions. Moreover, systematically managing available resources is important; this leads to higher throughput (i.e., delivering the same results within shorter time intervals) or higher efficiency (i.e., exploring a larger fraction of the search space within a certain time). Incorporating a user's preference during the search is an upcoming feature, which could be particularly useful in certain applications; after a number of iterations, monitoring the entire solution set becomes difficult so the user focuses on certain regions of interest[3] and evolves the local population. This has been proven very useful for real-world applications [194]. Frequently, the quality of optimisation algorithm is assessed by the performance of the PF for the total number of objective function evaluations. Obviously, the evaluation of an objective function is the most computationally intensive part of the optimisation process and the maximum benefit for the least computational cost is sought. Thus, algorithms that capture the target PF with as few evaluations as possible are considered better. When various well-known MOO algorithms were compared to each other, it was shown that none of them was capable of dealing with any class of problems sufficiently [195].

When assessing the quality of multi-objective optimisers, two important factors are the number of solutions and their diversity [196, 197]. Ideally, the final PF

[3] e.g., where the best solutions reside or the most innovative solutions appear

should demonstrate richness and span features. In other words, it contains as many points as possible closer to the target, which are as evenly spaced as possible. In fact, the method should discover and present to the end user the solution(s) they are interested in, and this partly depends on the tools used along with their settings. Then, several solutions should be available, but they should be different to each other. Of course, the number of required solutions at the end should be predefined as well; this will help the user to select a few solutions from the final presented trade-off, which was generated by the optimisation process. For example, on the one hand, the automotive industry wants a number of models for different market segments without competing with each other. On the other hand, for aerospace applications, manufacturers produce a single aircraft engine for target classes of aircraft. Hence, different needs dictate the respective settings for the problem set-up which will affect the whole process.

## 2.8 Summary and Linking with next Chapter

Based on the literature, the following limitations were identified:

- Computational methods for aircraft technology for mixing by using CFD tend to take considerable wall-clock time. Setting up the case to handle complicated boundary conditions, multi-phase or multi-component flows is usually a very time-consuming task, which requires further processing time to finish the simulation.
- A few computational methods of CFD, which are accelerated by GPUs, are compared against real-world data and even fewer are coupled with other systems (e.g., [130]).
- Porting and upgrading every possible application to CUDA does not pay off for the attained speed-up [131]. Frequently, linking with other systems and methods is very challenging [29] because of the discontinued support, which depends on legacy software and impedes coupling.
- Regarding the software design and implementation of optimisation applications by using CFD, to the best of the author's knowledge, this is the first time operational constraints are considered and these should be combined with a flexible software architecture, so as to link with other applications, too.
- A few scientific references were found that combine design optimisation along with GPUs-enabled CFD tools and methods (e.g., [133]). However, none was found to be applicable to micro-mixing optimisation studies.

- When accelerating optimisation methods, alternative parallelisation techniques (e.g., using GPUs) should be employed in the future [141].

More specifically:

- Gaps in methods:
  - Local-search-based metaheuristic optimisers could be employed in ATO.
  - The key to enabling hydrogen-based fuels lies on micro-mixing, but specialised methods are required to capture the complexity of a device at micro-scale.
  - The ability of LBM to manage complicated geometries could be used to study micro-mixing.
  - CFD methods could be accelerated by employing LBM that is deployed on GPU.
  - There are a few applications that combine optimisation techniques and CFD, so as to study micro-mixing.
  - In optimisation studies, it is not clear how optimum solutions are compared and selected.
- Gaps in tools:
  - CFD simulations are computationally intensive and require considerable resources.
  - GPU-enabled tools are expected to be an alternative approach to accelerated CFD and other computational methods, if the latter are restructured accordingly.
  - A new experimental feature could be added to the optimisation algorithm, so as to explore an alternative strategy that could potentially increase its effectiveness.
  - A few of the tools presented some interfacing capability with other tools (only via specific frameworks, such as GATAC).
- Gaps in applications:
  - It was demonstrated that acceptable solutions in ATO can be suggested by using optimisation techniques.
  - At the time of this research, most of the ATO studies are limited to general mission tasks and limited to two objectives optimisation.

- Mainly implementations of genetic algorithms are used in ATO.
- Combining GPU-enabled applications with optimisation techniques can accelerate the computational design optimisation cycle for real-world applications.
- Most of the LBM applications have been applied to benchmark studies, but they could be used to research micro-mixing.
- Tools ported on GPUs have achieved a high computational power at a reasonable cost.
- None had investigated the scalability of the presented tools and methods.

# Chapter 3

# Technical Description of Computational and Optimisation Tools

## 3.1 Multi-Objective Optimisation Algorithm

### 3.1.1 Introduction

As a single contribution, a new multi-objective optimisation algorithm for direct search problems with continuous decision variables, called MOTS2, has been developed and the source code can be found in [65]. Direct search is the class of optimisers that considers the evaluation of objective function(s) as a black box procedure. This is a multi-objective variant of the original tabu search [63], which had been deployed on a range of aerodynamic optimisations [198–201]. It can be considered as a hybrid that combines approaches from the fields of numerical analysis and artificial intelligence optimisation methods and techniques [202]. This variant contains several of the advanced features of tabu search, such as multiple strategies and various types of memories, as described in [203] by its creator, which is closer to the concept of adaptive memory programming [204]. Here, the main purpose is to introduce and to describe MOTS2 without compromising understanding and technical detail. Throughout this document, the terminology from [55,182] is used.

As identified by its developers, tabu search [205] could be employed in several real-world applications such as: scheduling, design, location and allocation, logic and artificial intelligence, technology, telecommunications, production, inventory and investment, general combinatorial optimisation, etc. Tabu search constitutes a powerful direct search method, which could be applied to many problem instances and is particularly strong in problems where classical solution methods are

infeasible [1].

Although MOTS2 is by nature a direct search optimiser, it can be used on a variety of other problems. As a local-search-based optimiser, its best performance can be obtained in cases where:

- the feasible decision space is too big to thoroughly investigate,
- the feasible decision space is highly complex to explore because of the constraints,
- there are multiple conflicting objectives,
- the evaluation procedure of a set of decision variables is considered as a black box.

In general, it was demonstrated that its performance is satisfactory on hard problems [190]. As a software package, it was developed by using best practices from the fields of software engineering, whereas many other software packages can be easily linked to other systems and it is hard to expand. More specifically, the object-oriented structure of MOTS2 allows to encapsulate its operation as part of a much larger system. Its modular structure allows it to scale up the portfolio of strategies. It also supports a number of interfacing mechanisms to be used either by human users or by other automated systems because of an embedded communication protocol, which is very similar to web-based applications. Finally, from the author's experience, it is very intuitive (even for less experienced users) to interpret and understand the results from the optimiser compared to other famous optimisers such as (mutation rate of) genetic algorithms and (thermal jump of) simulated annealing.

The Software Requirements Specifications (SRS) were created by considering portability, easy access, integration flexibility, computational efficiency and continuous expansion. The practices described in [206, 207] have been followed. First and foremost, a number of strategic decisions were taken into account, so as to re-design and re-implement the structure of the optimiser, in order to be part of an Engineering Design Process, as described in [208]. Originally, it was efficiently designed to be used in examples of real-world and continuous optimisation problems with multiple objectives (i.e., more than two). The optimiser attempts to imitate a systematic and organised search process that resembles human intuition, and high-quality solution(s) are expected to be discovered. It was also carefully designed to use as little computational time as possible, as suggested in [55, 209], by optimally utilising the available computational resources. Given an appropriate distributed environment or a computationally parallel framework, it can operate in

[1] for instance, no knowledge, such as derivatives, of the problem domain is available

parallel to save elapsed time because of its asynchronous [2] evaluation of objective functions in certain strategies, otherwise it is in a sequential mode. During the conceptual phase and implementation, principles from scientific computing and software engineering were combined to expand the algorithm to monitor the progress of the search and to visualise the solution; this will keep the user as informed as possible throughout the overall process. Additionally, integrated memories are manipulated and designed as databases, which makes the implementation very flexible and sustainable. The structure is such that it allows it to be configured and to be linked to other processes by using external files. For the validation phase, practices described in [182, 196] are considered.

The ultimate goal for the optimiser is to imitate an enhanced human cognition for a certain range of optimisation problems. The range is limited by the classes of problems. A number of integrated features are inspired by the fields of artificial intelligence, where knowledge-based methods usually perform more efficiently [202]. Of course, it is well known that the algorithm cannot be equally good in every class of problems [169]. However, the intention is to provide a flexible algorithm that adapts to the problem and can deliver very competitive results, even compared against specific algorithms. The comparative advantage is based on its artificial intelligence features: The algorithm on-the-fly analyses decision space, interrogates its current knowledge base and employs the most appropriate modules from a provided portfolio of strategies. As expected, proper integration of the tools into the portfolio in terms of interfacing is a separate issue, which is worth exploring. Ultimately, an ideal optimiser will be given a certain computational budget, will sufficiently cover decision space, will manage computational resources, will quickly identify the most promising regions with the best performance in as few iterations as possible and will focus the search and effort to discover a wide and rich range of solutions that capture the globally optimal performance of the target system.

The structure of this section follows. Firstly, the progress of the related developments of tabu search as a MOO algorithm is presented. Next, the algorithmic structure and strategic decisions during the development of the optimiser are described in terms of data structures, algorithmic logic and functionality. The behaviour of the optimiser in relation to the optimisation problem is explained, followed by special features and additions. Thereafter, the optimiser is verified against a family of benchmark tests and is validated against an instance of design optimisation in aerospace development.

[2] In contrast, it is possible to implement an optimiser to evaluate one objective function after another in a certain order, which cannot be accelerated by any computationally parallel framework

### 3.1.2 Trends in Multi-Objective Tabu Search

In the last 15 years, several studies have explored and investigated various instances of tabu search. Many variations have come to the surface, where different features of the optimiser are modified, introduced and expanded in the hope of improving the code. Great spread and depth of information about tabu search can be found in [55, 187, 188]. Performing a survey on tabu search is out of the scope of this subsection. Here, the intention is to present important contributions that inspired the development of the current optimiser.

Tabu search methods belong to the class of metaheuristic optimisers [55, 202]. They perform stochastic and local search or, alternatively, intensive local search. Originally, they were designed to manage the heuristics of hill climbing, but they were adopted to manage the heuristics of neighbourhood exploration (e.g., see Hooke and Jeeves Move in 3.1.3.3). In fact, they can be considered as a strategy that controls a collection of embedded heuristic techniques. The former family of optimisers first introduced the concept of memory in metaheuristics, where the reactive and parallel derivatives emerged [210–212]. This algorithm is based on the single-objective tabu search, introduced in [205, 213], and on its multi-objective variant (the first MOTS), as described in [63], where a detailed explanation can be found. Moreover, the current implementation of MOTS2, which is used in the next chapter, also includes the improvements (i.e., local search enhancements for the Diversification Move) discussed in [198] and extra features described in subsection 3.1.5.

An interesting approach that attempts to group several metaheuristics in a single category was demonstrated by introducing the concept of adaptive memory programming [204], as a unified problem-solving approach. It is thought of as a general scheme that assembles the essence of the good ideas from many metaheuristics. This class of optimisers has several advantages and it is suggested they be used on difficult problems. Tools based on adaptive memory programming could be easily adapted to operate in parallel with computationally distributed environments. They could be particularly efficient on dynamic problems, since their method could adapt to slightly modified data [214]. The program could demonstrate several different high-quality solutions, which will enable the decision maker to select an appropriate one to solve their problem. The latter can be particularly useful in real-world applications, where frequently a few features of the problem that could affect the final solution are disregarded or are not included so as to have the option to choose a solution with higher importance among several alternatives. Hence, it was highly advisable to adopt tools to metaheuristics with memory for future developments.

### 3.1.3 The Design and Implementation of MOTS2

Until the moment of writing up this document, there was not any available version of multi-objective tabu search. So, a new version was designed and developed from scratch based on the technical description of the original version in [63,198] and author's experience in developing software packages for industrial applications. This code is also intended to be released as an open source project and the top-level design is depicted in Fig. 3.1, whose individual features are described below. At the top, the different types of memories are listed, which are described in greater detail in the following subsubsection. At the bottom left side, the four modules related to the objective function provide interfacing mechanisms to evaluate the objective function(s) of the optimisation process either externally or internally of the optimiser. These can also delegate the evaluation on a GPU, as will be described in subsection 3.2.4. At the bottom right side, the settings modules were separated to allow it to carry out performance analysis by injecting various configuration settings in the optimisation algorithm, as will be demonstrated below. For more details, the pseudocode is illustrated in Listings 3.1. MOTS2 was implemented in two stages: Firstly, the memories were created as the core data structure of the algorithm. Then, the algorithmic structure was built around them. Since the concept of memories that manipulate problem-specific data is so important, it is sensible to employ object-oriented techniques and database design. Because of the diversity of cases, the optimiser will be called on to perform, the software is implemented in C++ programming language, which was selected for performance, structural, linking and interfacing purposes. The code is portable, cross-platform and can be deployed on any computational architecture. This subsection can be used as a design pattern for the development of a wider range of optimisers based on adaptive memory programming.

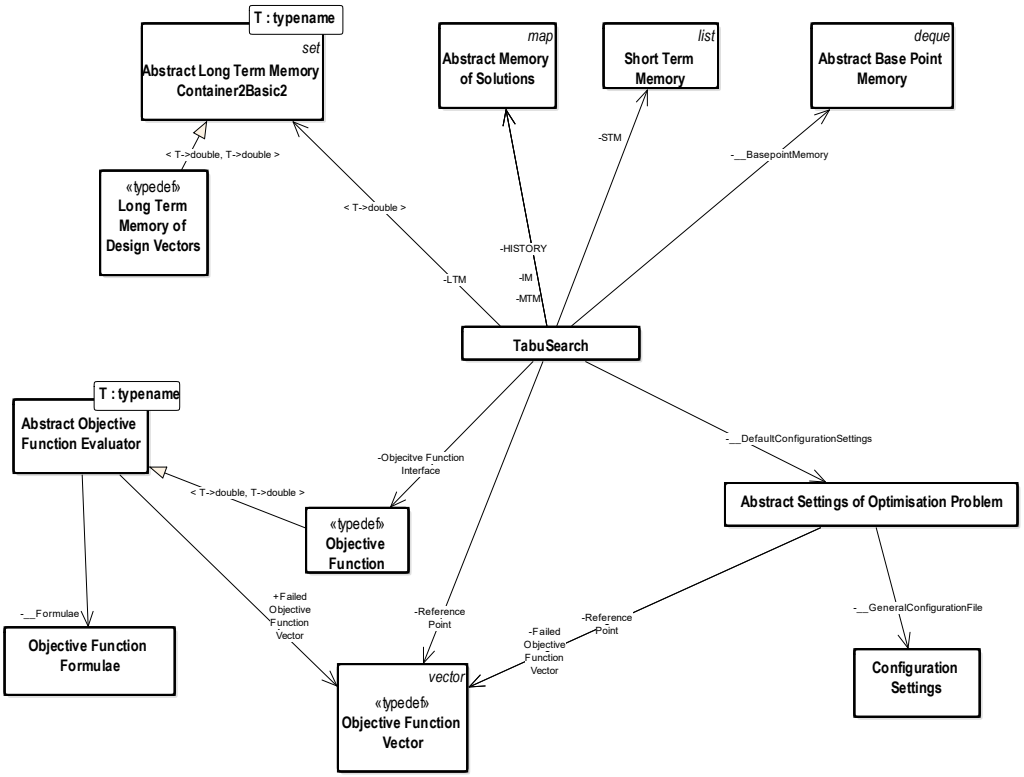



Figure 3.1: MOTS2 Architecture

Listing 3.1: MOTS2 Top Level Pseudocode

```
1  double TabuSearch(){
2      int current_loop;
3      int previous_current_loop;
4      int iLocal;
5      int previous_evaluations;
6      int trade_off_consequetive_improvements;
7      int maximum_improvements;
8      int iLocal=0;
9      int success=0;
10     Point2 newPoint(nVar, 0.0);
11
12
13     int stoppingCriteriaTrigger = stoppingCriteriaNotMet(
14           current_loop-previous_current_loop,
15           HISTORY.count_evaluations(TS_ObjFunc.
                 get_penalty_objectives())
16              -previous_evaluations,
17           trade_off_consequetive_improvements);
18     while (stoppingCriteriaTrigger){
```

```
19          if (nextMoveHookeJeeves){
20              HookeJeevesMove_parallel(current_loop) ;
21          }else {
22              success = PatternMove() ;
23              if (success)
24                  nextMoveHookeJeeves = 1 ;
25              else{
26                  HookeJeevesMove_parallel(current_loop) ;
27              }
28          }
29
30          line18:
31          UpdateMemories();
32
33          int kickTrigger = (MTM.activate_kick("./memories/
                MTMfrequencies.txt",
34                  maximum_duplicates,
35                  TS_ObjFunc.failedObjectiveFunctionVector) &&
36                  trade_off_consequetive_improvements>
                        maximum_improvements*0.1)
37                  || trade_off_consequetive_improvements>
                        maximum_improvements;
38          if (iLocal==diversify) {
39              newPoint=DiversifyMove2() ;
40              Push_Base_Point(newPoint, "DiversificationMove");
41              goto line18 ;
42          }else if (iLocal==intensify) {
43              newPoint=IntensifyMove2() ;
44              Push_Base_Point(newPoint, "IntensificationMove");
45              goto line18 ;
46          }else if (iLocal==reduce) {
47              newPoint=ReduceMove2() ;
48              Push_Base_Point(newPoint, "ReduceMove");
49              goto line18 ;
50          }else if (kickTrigger){
51              newPoint=ReduceMove2() ;
52              Push_Base_Point(newPoint, "kickMove");
53              trade_off_consequetive_improvements=1;
54              goto line18 ;
55          }
56
57          update_trade_off_progress();
58
59          ++current_loop;
60      }
61  }
```

#### 3.1.3.1 Describing the Different Types of Memories

Memories have first been explicitly used in tabu search and this section describes the concept of hierarchical memories, which are believed to have a considerable impact on the ability of the optimiser to solve problems [203]. In total, there are 5 types of memories within the optimiser (but totally 3 types of memory in terms of software architecture), each with a slightly different purpose. These are the Short Term Memory (Tabu) (STM), Medium Term Memory (full Pareto Front) (MTM), Long Term Memory (LTM), Intensification Memory (IM), and History Memory (HISTORY) containers, or, alternatively, memory banks. All memories share a common feature: Their attributes are user-defined and remain constant throughout the optimisation search. Moreover, in MTM, IM and HISTORY duplicate decision vectors are not allowed, but identical objective function values (if any) might exist. The different types of memories and their relations among themselves with regard to an example 2D objective space are depicted in Fig. 3.2. There is another simplistic type of memory, called Memory of Base Points that registers the order of the selected Base Points (BPs), explained later. This will not be covered here and exists for monitoring purposes and future development. All these memories are used to assist critical decisions during the optimisation process. The pseudocodes of the signature of the three different classes are illustrated in Listings 3.2- 3.4. The actual implementation of the individual methods of the class is identical to the original version of MOTS [63]. All the other methods of the class are just helper methods that perform sanity checks and utility operations (etc., storing a snapshot of the memory) for analysis purposes.

The first, STM, is the collection of Tabu points and is the simplest structure, as shown in Listing 3.2. These are black-listed points that the optimiser is not allowed to select during the search, when searching for a new decision vector as long as they remain in the memory. The size of STM is selected before the optimisation starts and remains fixed until the end. In principle, it implements a stack data structure and contains information about the decision variables only. During every iteration, a new point is pushed to the top of the memory and, if the memory is full, the last point pops out. Therefore, the newly inserted points are located at the top, and the older ones are at the bottom of the memory.

Listing 3.2: STM Container Class Pseudocode

```
1 class STM_Container : public std::list< Point >{
2 public:
3     unsigned int STM_size;
4     unsigned int local_nVar;
5     std::string container_name;
6     std::string __save_path;
7
8     STM_Container(const int &input_STM_size, int const n, std::
```

```
        string given_name, std::string save_path);
    void import(Point const &P);
    void showTabu();
    bool isTabu(const Point &P);
    void save_stm_container(std::string case_qualifier, int
        evaluations_counter);
    int load_stm_container(const char* load_path);
};
```

In fact, MTM, IM and HISTORY are instances of the same class of memory and the corresponding pseudocode is shown in Listing 3.3. Following the principles of object-oriented design, a single class was implemented, called memory container, which is instantiated three times, one for each conceptual memory. The implemented memories are different in two ways: Firstly, the insertion policy is different for each memory, where combinations of decision variables and objective function values are inserted into the memory container under different occasions, as described below. Secondly, different functions are applied to memories at different stages, as explained in sub-subsection 3.1.3.2. This was done for ease of implementation and for a more sustainable design.

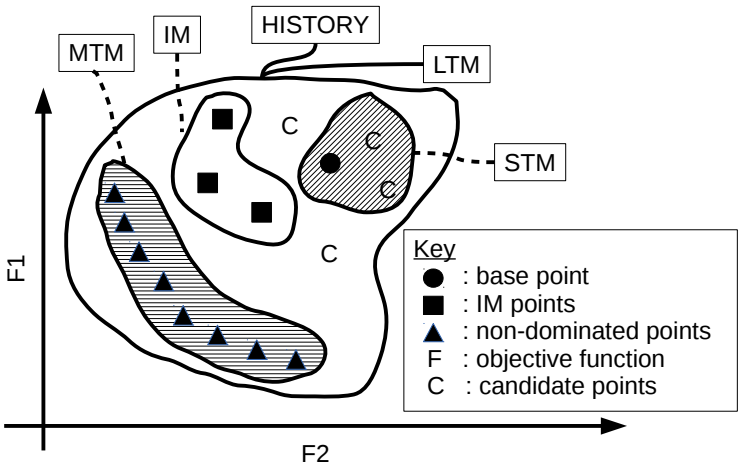



Figure 3.2: Tabu Search Memories

Listing 3.3: Container Class Pseudocode

```
class Container : public std::map< Point, ObjFunction >{
    unsigned int local_nVar;
    unsigned int local_nObj;
```

```
4     std::string container_name;
5     std::string __save_path;
6     typedef std::pair<Point,ObjFunction> tuple;
7     int dominates(const ObjFunction &O1, const ObjFunction &O2);
8     void displayContainer ();
9     int report_optima_plot(char const *save_path);
10    int report_optima_current(char const *save_path);
11    double calculate_hypervolume2D(const ObjFunction &
          reference_point);
12 public:
13    void add_point(const entry __entry);
14    int load_container(char const *load_path);
15    double calculate_quality_indicator(const ObjFunction &
          reference_point);
16    int activate_kick(char const  *save_directory, int const
          kick_limit, ObjFunction failedObjectives);
17    int count_evaluations(ObjFunction given_penalty_vector);
18    Point check_memory();
19    int removeDominatedPoints();
20    int addIfNotDominated(const Point &V1,const ObjFunction &O1);
21    Point selectRandom();
22    Point find_extreme(int objective);
23    int save_container_snapshot(std::string case_qualifier, int
          evaluations_counter);
24    Container(unsigned int n, unsigned int m, std::string
          given_name, std::string filename);
25    ~Container();
26 };
```

These three types of memories implement a dictionary (i.e., hash table) data structure, which holds a number of entries but are populated differently. Each entry is defined as an assembly of decision vectors and their corresponding objectives. Moreover, all the entries are listed in ascending order, based on the decision variables only. The collection of the optimal non-dominated points, which are usually called PF points, is stored in MTM. This memory holds the final output of the optimiser, which is a trade-off among the selected objectives and presents their interplay. All the points required for a tactical random selection (see the Intensification move in sub-subsection 3.1.3.4) reside in IM, which is used for the local-search features. Lastly, HISTORY logs all the evaluated points (i.e., both feasible and infeasible). This gives one the big picture of the optimisation search, which is frequently combined with data mining techniques, so as to extract information that will guide the search further, as suggested in [215]. In addition, HISTORY is used as a look-up table that holds the already evaluated decision vectors for post-processing and other purposes.

The last type of memory, LTM, is used for the global-search features and is the

most complicated memory to implement, as shown in Listing 3.4. This holds the visiting frequency of various regions of decision space for each decision variable. Each variable range is equally split into a number of regions, whose frequency is registered, monitored and used for certain occasions (see the Diversification Move in sub-subsection 3.1.3.4). The number of regions is the same for all the decision variables, is set beforehand and remains the same until the end. This simply informs the optimiser and the user which areas are the most and least frequently visited. It also forces the optimiser to explore the unknown regions in order to diverge the course of the search to sparsely explored areas. Whenever it is required, the memory calculates the frequency per decision variable per region of decision variable and finds out the least visited region per decision variable. Then, a new decision vector will be formed and each of its decision variables would belong to that region.

Listing 3.4: LTM Container Class Pseudocode

```
1  class LTM_Container :  public std::set< Point >{
2     unsigned int local_nVar;
3     unsigned int local_nRegions;
4     Point local_lower_bound, local_upper_bound;
5     Point tempVector;
6     T offset;
7     Point range_points;
8     std::vector<Point > variable_range_points;
9     std::string container_name;
10    std::string __save_path;
11    LTM_Container & operator= ( LTM_Container const &other);
12 public:
13    std::deque< Point > Region;
14    LTM_Container(const int &nVar,const int &nRegion, const Point &
          lower_bound, const Point &upper_bound, std::string
          given_name, std::string filename );
15    ~LTM_Container();
16
17    T RandomNumber(T min, T max);
18    Point generate_Random_Point_From_Least_Visited_Region2();
19    int leastVisitedRegion();
20    void showContents();
21    int getSize();
22    int getSizePerRegion();
23    std::vector<int> getRespetiveRegion(const Point &P);
24    void save_ltm_container(std::string case_qualifier, int
          evaluations_counter);
25    void load_ltm_container(char const *save_path);
26 };
```

#### 3.1.3.2 Algorithmic Structure

The optimiser depends on memories and performs a search by combining a systematic local search along with stochastic elements so as to intelligently search the entire decision space. The flow of the aforementioned procedures is shown in Fig. 3.3. Following the categories of [55], MOTS2 implements single-solution-based metaheuristics, where a single solution is improved in every iteration and the search looks like a *walk* through a neighbourhood. The search starts from a BP and three memory containers (STM, MTM, HISTORY). All these are used in every iteration, whereas the other parts are called on when special conditions are met. It starts with blank memories, and as the optimisation search progresses, it gradually builds a knowledge base about discovered decision space and objective space. Then, the search is guided by the performance of the decision vectors in objective space, so as to discover even better performance.

The search is an iterative process, as demonstrated in Fig. 3.3. During an iteration one heuristic procedure takes place every time, another heuristic occurs every other time and a special one (from a group of specials, described in 3.1.3.4) is conditionally triggered at certain stages. In any case, all the stages are guided by the BP. Around the BP, a few adjacent candidate decision points are investigated and evaluated. Then, the corresponding fitness values are sorted according to domination criteria of multi-objective optimisation [192], the appropriate memories for the current iteration are updated and the next BP is resolved. The previous BP and all the recently generated points are inserted into the appropriate memory containers. Hence, MOTS2 progresses effectively through objective space, while not wasting precious computational time. This is one of the basic operations of single-solution-based optimisers. An analogy to population-based optimisers is to create many solutions at every iteration. The population is generated based on mechanisms such as genetic operations between different genes (in genetic algorithms) or looking for food (in particle swarm algorithms), where each of these mechanisms can be configured by altering certain parameters such as mutation rate (in genetic algorithms) or travelling speed (in particle swarm algorithms). An analogy to global-search optimisers is that it is possible to visit any point of the decision space, whereas local-search optimisers explore a fraction of the decision space within a relatively close distance.

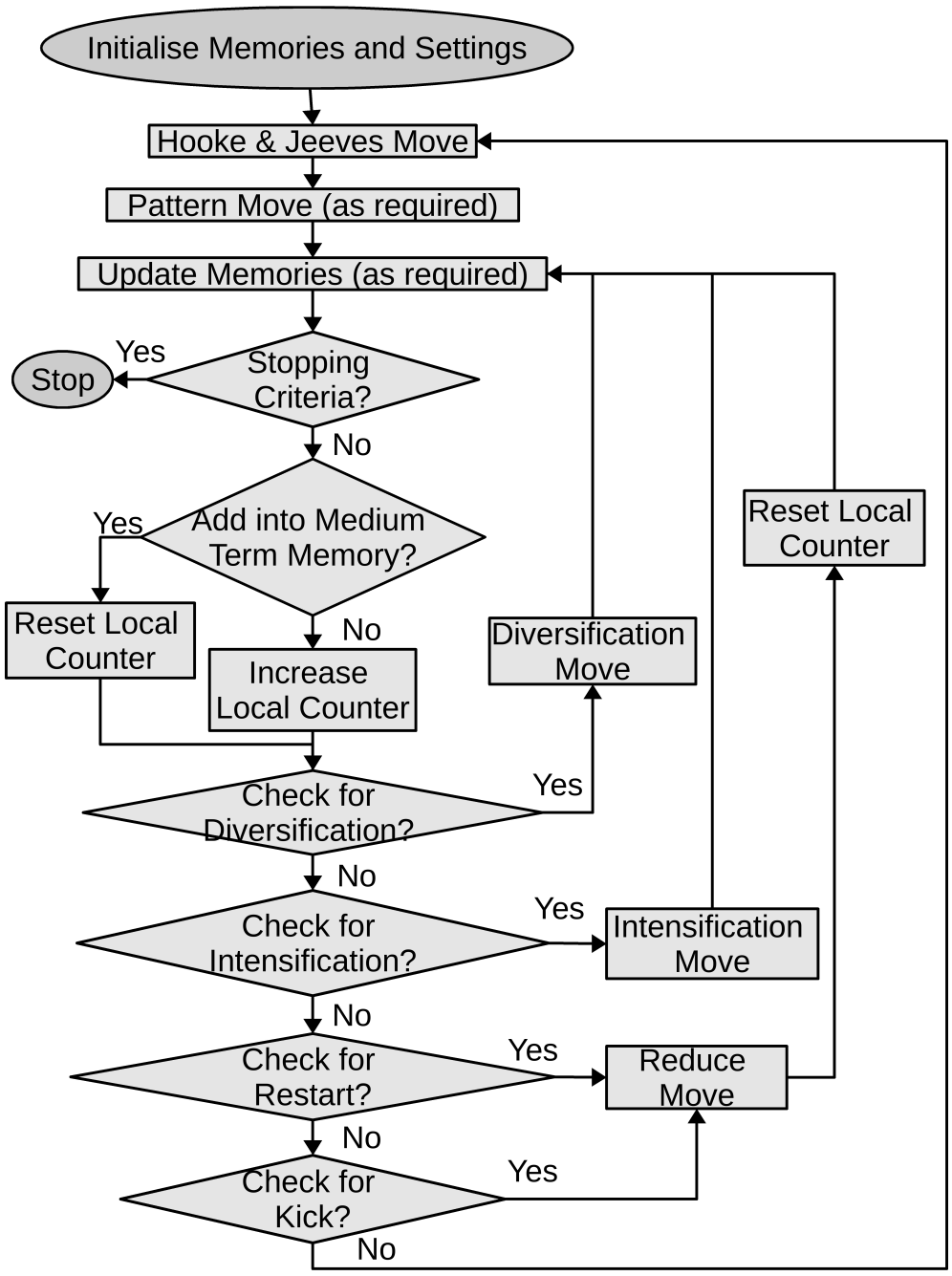



Figure 3.3: MOTS2 Algorithm Flow Diagram

In general, decision space is explored in a stochastic way, while recently visited

points called Tabu points, are avoided, so as to guarantee more exploitation of the unknown decision space. More importantly, this type of optimiser explicitly uses memory (and history to some broader extent) throughout the optimisation search. Initially, this seems complicated and taking advantage of it is not very straightforward. However, this can be particularly beneficial in terms of computational budget when the number of iterations increases and/or when a direct search is the only possible way. In fact, the local search scheme, which implements Hooke and Jeeves [216] that is particularly efficient for continuous parameters, is combined with stochastic elements and other enhancements.

The optimiser also keeps track of statistics during the optimisation process, which direct the search according to the discovered landscape of decision space. At the top level, the optimiser employs a mechanism for local-search and global-search. Thus, it could be considered as a hybrid, since it expands the original definition [55, 188]. In practice, the local search part is performed more frequently than its global counterpart. However, the global part can be considered as an enhancement to cover more general cases and assists the optimiser to escape from local optima. The statistics determine the progress of the optimiser by activating and deactivating certain mechanisms that change the behaviour of the optimiser when special conditions are met; they are mostly used to detect decision points around the current BP within a relatively short distance, whereas the search mechanisms attempt to discover good decision points in the entire decision space. This implements a form of self-adaptation to the fitness landscape of the problem. Consequently, the functionality of MOTS2, as depicted in Fig. 3.3, results in better performance throughout the optimisation process.

At a glance, the search starts without any information or from a previous check point, and it gradually becomes more familiar with decision space, while applying various heuristic mechanisms to move through. Aggregated information will be used in future steps to guide the search when certain conditions are triggered. This procedure keeps repeating until the user-defined stopping criteria are met. Depending on the nature of the application, these are usually the elapsed time, the number of evaluations , the number of consecutive failures to find a better point and the number of iterations or a combination of them. In a production environment, time restrictions, computational costs and financial costs could be considered. The number of evaluations required to finish the optimisation mainly depends on the complexity of the problem, which is not frequently known a priori. However, a suggested approach is to consider the quality indicators, as discussed in [182], mentioned in subsection 3.1.5 and used in subsection 3.1.6. Because it is assumed that the process can stop at any given moment at the user's discretion, there is no termination mechanism based on the actual time. During every iteration, a fraction of the flow of the algorithm is executed every time, and the rest runs when certain

conditions are met. The core of the optimiser consists of the Update Memories, Hooke and Jeeves-Move, Intensify-Move and Reduce-Move; the remaining parts are algorithmic enhancements, which speed up the search.

In the simplest approach, the optimisation starts off from a datum decision point, which is a combination of parameters of well-known performance and must be a valid point. Unless this is true, then there is no point in trying to improve any solution. Following the operation of local-search optimisers, the selection of the datum decision is the most crucial part before actually launching the optimisation process, as this could potentially trap the algorithm at an early stage. Then, several new points are generated and evaluated. One of them will be selected as the BP, throughout the current iteration loop, and the rest are accordingly forwarded to the memories. The selection process of the BP depends on a mixture of strategies that combine the contents of STM and dominance criteria. Thereafter, the optimisation keeps iteratively going on. An example is illustrated in Fig. 3.4 with a very low diversification threshold (see Diversification Move in 3.1.3.4), where a special move is performed once to assist the algorithm to escape a local minima situation. This was specified for demonstration purposes only. In reality, diversification should happen after several evaluations that have not discovered a satisfactory solution, as will be explained later. The algorithm keeps checking the statistics to determine whether any special functionality is appropriate to be called on.

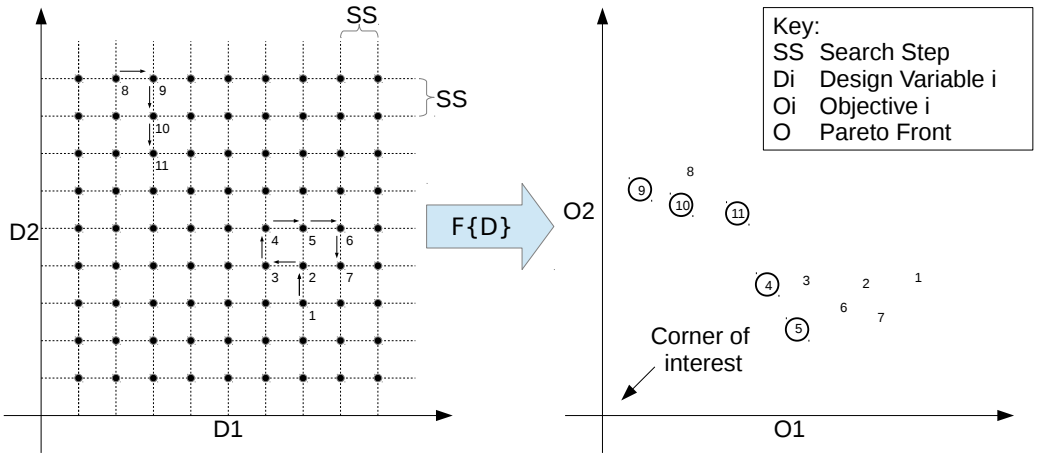



Figure 3.4: An Instance of MOTS2 on a 2 Decision Variables and 2 Objective Optimisation

#### 3.1.3.3 Frequently Performed Moves

The following parts take place on a regular basis:

**Hooke and Jeeves Move:** It was originally described in [216] and is the most important part of the local search scheme, as it occurs on every iteration and requires plenty of evaluations of the objective function(s), which can be expensive. This is the most time-consuming part of the optimisation process and a few options are provided in order to minimise spending long periods at a single part of an iteration. Starting from the BP, a couple of valid and non-tabu points are generated by combining the current BP and the current Search Step (SS), as illustrated in Fig. 3.5. In theory, the number of points that can be generated is up to twice the number of the decision variables, but since some of them could either belong to STM or be invalid, they are excluded from the following stages of Hooke and Jeeves Move. Then, a few of the recently created points are randomly selected and evaluated in order to save some computational budget and are added into the appropriate memory banks. These points are within the close vicinity of the BP. This is the local search phase of the optimiser. In order to save computational budget, a lesser value than the theoretical maximum should be selected, but not too low because the optimiser will compromise local exploration. Among the recently evaluated points, one of them will be selected as the BP and the search continues.

The steps for selecting the BP in Hooke and Jeeves Move are described below:

1. Remove all the tabu points from the procedure and create candidate points.
2. Split the remaining candidate points into sets, based on the number of samples per set. If there are not any candidate points and IM contains any points, one point from IM is randomly selected, otherwise the algorithm stops.
3. Apply the following steps in the first set of candidate points and repeat if it is impossible to discover at least one point that is not dominated by the current MTM:
   (a) Evaluate all the points of the current set in parallel, so as to map them against their corresponding objectives.
   (b) Compare the objectives of each point against the objectives of the currently optimal trade-off. If the point is not dominated by the points in the MTM, it is inserted in the set of dominant points, otherwise it is inserted into the set of dominated points.

(c) If the set of dominant points contains any elements, randomly select one of them as the next BP and exit the selection procedure, otherwise use the next set of candidate points and repeat.

4. If there are no points that are not dominated by the current MTM, then one of them is randomly selected as the next BP.

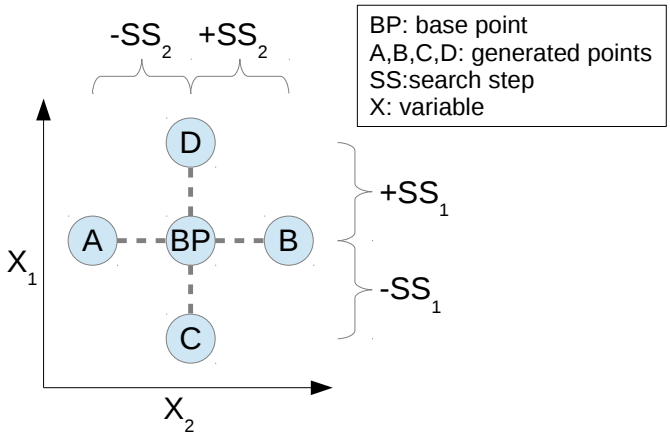



Figure 3.5: Hooke and Jeeves Move in Decision Space

The pseudocode in this implementation is illustrated in Listing 3.5.

Listing 3.5: Hooke and Jeeves Move Pseudocode

```
1  void HookeJeevesMove(unsigned int loop){
2      unsigned int i;
3      Container::iterator it;
4      int remaining_sample_sets;
5      temp_Container bestCandidatePoints;
6      Container temp_sampled(nVar,nObj,"HJ_P1","temp_sampled.txt");
7      Container temp_sampled_dominant(nVar,nObj,"HJ_P2","
           temp_sampled_dominant.txt");
8      Container temp_sampled_dominated(nVar,nObj,"HJ_P3","
           temp_sampled_dominated.txt");
9      Container C1(nVar,nObj,"HJ_P4","C1.txt");
10     Container buffer_container(nVar,nObj,"HJ_P5","buffer_container.
           txt");
11     Point newPoint(nVar,0.0);
12     Point tempPoint(nVar,0.0);
13     Point nextPoint(nVar,0.0);
14     Point currentPoint(__BasepointMemory.get_current_base_point());
           each designVariable
```

```
15        newPoint_incr=increase(currentPoint,i) ;
16
17        if ( !STM.isTabu(newPoint_incr) &&  TS_ObjFunc.isValid(
              newPoint_incr)  ){  isNotTabu(newPoint) and isNotInvalid
              (newPoint)
18            bestCandidatePoints.insert(newPoint_decr);
19        }
20    }
21
22    remaining_sample_sets=(int)(bestCandidatePoints.size()/n_sample
          );
23    if(bestCandidatePoints.size()%n_sample!=0)
24       ++remaining_sample_sets;
25
26    if(bestCandidatePoints.size()==0){
27        if(IM.size()>0){
28            Point tempPnt=IM.selectRandom();
29            IM.erase(tempPnt);
30            ++remaining_sample_sets;
31            bestCandidatePoints.insert(tempPnt);
32        }else{
33            std::cout << "algorithm trapped, neither good candidate
                  point was generated, nor IM has any candidate point"
                  << std::endl;
34            save_external_memories(" ",loop);
35            exit(-3000);
36        }
37    }this contains the points to be sent for evaluation and then
          its entries are  updated
38
39 evaluate Points
40            tempPoint=bestCandidatePoints.selectRandom();
41            bestCandidatePoints.erase(tempPoint);
42            evaluation_buffer.insert(entry( tempPoint , TS_ObjFunc.
                  penalise(tempPoint) ) );
43        }
44
45        evaluate_point_parallel(evaluation_buffer);
46        for(Container::iterator it=evaluation_buffer.begin(); it!=
              evaluation_buffer.end(); ++it)
47            temp_sampled.insert(entry( it->first , it->second ) );
48
49        evaluation_buffer.clear();
50
51        for(Container::iterator it1=temp_sampled.begin(); it1!=
              temp_sampled.end(); ++it1){
52            if(     MTM.addIfNotDominated(    it1->first , it1->second
                      )==1   )
```

```
53              temp_sampled_dominant.insert( entry (  it1->first , it1
                    ->second  ) );
54          else
55              temp_sampled_dominated.insert( entry ( it1->first , it1
                    ->second ) );
56      }
57
58      switch ( temp_sampled_dominant.size() ) {
59      case 0 : end of case 0
60      case 1 : end of case 1
61      default :
62          nextPoint=temp_sampled_dominant.selectRandom();
63          keep_sampling=0;
64          break;
65      }
```

**Pattern Move:** This is just an enhancement of the Hooke and Jeeves Move, where the next BP will be quickly resolved via a single evaluation. Whenever the iteration number is even, the following BP is generated by combining information from the last two BPs. It takes place every other iteration (once in every two iterations). Thereby, the search may be accelerated along known downhill directions, because two BPs will have been resolved in a single iteration. In fact, it calculates the difference in terms of decision variables between the last and penultimate BP and applies the same change to generate a new point that will be evaluated. It seems like following the search direction of the recently executed Hooke and Jeeves Move. The pseudocode in this implementation is illustrated in Listing 3.6.

Listing 3.6: Pattern Move Pseudocode

```
1  int PatternMove(){
2      Point lastPoint, lastMove, newPoint;
3      Point currentPoint;
4      ObjFunction tempObjFunc;
5      ObjFunction currentObjFunc;
6      currentPoint =_BasepointMemory.get_current_base_point();
7      currentObjFunc = HISTORY[currentPoint];
8      lastPoint =_BasepointMemory.get_previous_base_point();
9      lastMove = currentPoint-lastPoint;
10     newPoint = currentPoint+lastMove;
11
12     if(!TS_ObjFunc.isValid(newPoint) ){
13         return 0;
14     }
15
16     tempObjFunc=evaluate_point(newPoint);
17
```

```
18      if ( MTM.addIfNotDominated(   newPoint, tempObjFunc )==1 ){
19          IM.insert(entry ( newPoint, tempObjFunc ));
20          ++iLocal;
21          Push_Base_Point(newPoint, "PatMove");
22          return 1;
23      }else{
24          return 0;
25      }
26  }
```

**Update Memories:** This is the only non-metaheuristic procedure and is employed for operational purposes. At the end of every iteration, the newly resolved BP (i.e., either the decision vector along with its corresponding fitness or just the decision vector) is inserted into the Memory of Base Points, STM, MTM, LTM, IM and HISTORY should it fulfil their corresponding conditions. Each memory is populated according to its type, as described above. New points are popped and pushed into STM following the principles of a data queue. All the points within IM are filtered based on Pareto dominance [192], whereas HISTORY keeps track of all the records. The dominated points from MTM are also discarded.

As the algorithm runs, the memory containers accumulate information that will be exploited at later stages. Therefore, a zero-knowledge search starts and as the optimiser runs, it learns about the intrinsic features of decision space from the containers iteration-by-iteration. This results in a knowledge base and according to the principles of artificial intelligence, this is the best method for a heuristic search to be performed [202–204]. As the optimisation process progresses a more informed selection takes place, whenever required.

#### 3.1.3.4 Conditionally Performed Moves

Besides the moves mentioned above, a number of complementary moves is carried out when certain conditions are met, one at a time. These moves can escape MOTS2 from local minima by performing local search, by performing global-search and by performing step refinement, respectively. Although the calling frequency is defined by the user in order to be consistent with the original definition of tabu search, the numerical value of frequency should increase in the order of appearance, otherwise unexpected behaviour may occur.

**Intensify Move:** By definition, contrary to a single-objective optimisation, during a multi-objective optimisation several points form the PF. However, during every iteration, only one of them might be the BP. Therefore, the

remaining points that dominate the current trade-off, but have not been selected as BPs, are inserted into IM. Whenever the search can neither discover any new nor non-tabu points for a number of iterations, another point from that back-up container is selected randomly as the following BP. Therefore, the search returns back to the vicinity of the most promising points discovered so far and picks up the search thereafter from that location of decision space. This should be the most frequently performed move to assist the optimiser in discovering any improvement(s). In a similar fashion, if more than one new points improve the fitness, they will be stored for future use, should they not be discarded at some later stage. The pseudocode in this implementation is illustrated in Listing 3.7.

Listing 3.7: Intensify Move Pseudocode

```
1  Point IntensifyMove(){
2      Point newPoint(nVar,0.0);
3      ObjFunction newObjFunc(nObj,0.0);
4
5      do{
6          newPoint = IM.selectRandom();
7          IM.erase(newPoint);
8      }while( STM.isTabu(newPoint) && IM.size()>0);
9
10     ++iLocal;
11     ++intensification;
12     nextMoveHookeJeeves = 1 ;
13     IM.removeDominatedPoints();
14     return newPoint;
15 }
```

**Diversify Move:** Instead of finding a better point within a short range, a new non-Tabu point is randomly and globally generated from the least explored region of decision space. Here, the information stored in LTM is utilised in order to generate the new BP. Decision space is explored as uniformly as possible. This is the global-search phase of the optimiser, and its frequency depends on the problem. The pseudocode in this implementation is illustrated in Listing 3.8.

Listing 3.8: Diversify Move Pseudocode

```
1  Point DiversifyMove(){
2      Point newPoint(nVar,0.0);
3      ObjFunction newObjFunc(nObj,0.0);
4      int patience=20;
5      do{
6          newPoint=LTM.
               generate_Random_Point_From_Least_Visited_Region();
```

```
7         --patience;
8     }while ( (STM.isTabu(newPoint) || (!TS_ObjFunc.isValid(
          newPoint))) && (patience!=0) );
9
10    if(patience==0)
11        newPoint=TabuSearch::IntensifyMove2();
12
13    evaluate_point(newPoint);
14    ++iLocal;
15    ++diversification;
16    nextMoveHookeJeeves = 1 ;
17    return newPoint;
18 }
```

**Reduce Move:** Whenever the search fails to discover a new point with the current SS for a large number of iterations, the next BP is randomly selected from the MTM and must not be a tabu point. Then, the SS is refined accordingly. This should be the rarest performed move because SS is only reduced. By performing this move the optimiser gradually narrows down the exploration range and focuses on certain regions, where it is expected to find a better optimum. The pseudocode in this implementation is illustrated in Listing 3.9.

Listing 3.9: Reduce Move Pseudocode

```
1 Point RestartMove(){
2     Point newPoint(nVar,0.0);
3     ObjFunction newObjFunc(nObj,0.0);
4
5     do{
6         newPoint = IM.selectRandom();
7     }while( STM.isTabu(newPoint) );
8
9     for(unsigned int i=0; i<nVar; ++i)
10        CurrentStep[i]*=SSRF;
11
12    iLocal=0;
13    ++reduction;
14    nextMoveHookeJeeves = 1 ;
15    return newPoint;
16 }
```

### 3.1.4 Constraints and Objectives Handling

Throughout this document, only minimisation of objectives will be considered, due to the duality between maximisation and minimisation. So, for maximisation

problems, one simply needs to reformulate the original problem: the quantities to be maximised should be multiplied by (-1). Hence, minimising the negative of any objective equals maximising the same objective. In addition, any number of objectives (i.e., above two) can be minimised. It is worth mentioning that multi-objective optimisers can also perform single objective optimisation just by using the same numerical value for two objectives.

Furthermore, MOTS2 can deal with both soft and hard constraints. The former have to be programmed within the source code and the optimiser needs to be compiled again. Although the range of variability also belongs to the soft constraints, the range of each parameter can be set individually without the need for compiling. Whenever there is a violation of constraints, the optimiser should automatically assign a very large penalty value to the set of decision variables that triggered the constraint without evaluating the respective decision vector on the actual evaluation tool/method. If some small value is specified as a penalty value, this will trip and/or trap the optimiser, and it should be avoided at all costs. The hard constraints are related to the objective function evaluation tool/method, and they are set independently of the optimiser [3]. Since the optimiser performs minimisation of the objectives, it is highly advisable for the evaluation tool to assign a different penalty value at the same order of magnitude as for the soft constraints. This will prevent confusing the optimiser and will assist in the analysis of the results. Hence, the optimiser can deal with any type of optimisation problem that involves real parameters, both constrained and unconstrained.

### 3.1.5 Unique Features

This version of MOTS2 has a number of features that differentiate it from the original variation, presented in [63], so as to explore more efficiently decision space and to tackle a wider range of problems by providing extra handles and settings. A few of the suggestions in [203] were considered, too. As usual, the purpose is to extract additional information and to cover a larger fraction of the promising areas of decision space by being more intelligent. Hence, the optimiser stands greater chances to be used on diverse cases and becomes a very powerful and flexible engineering tool for ATO and design optimisation problems. Firstly, they all intend to make the structure of the optimiser more flexible in order to link with external and internal tools and software. Secondly, they serve as the basis for other tools to be developed on top of these. Thirdly, more configuration settings enable the optimiser to adjust to the project requirements.

At the data representation level, the object-oriented decision allows all the

[3] The penalty values of the invalid decisions should be of the same order of magnitude as the soft constraints. However, both should be set to a numerically large value.

parts of the optimiser to be developed in isolation, while the data are being manipulated through mechanisms similar to databases for improved access, better organisation and ease of applying functions (for instance, structured queries). The latter will be particularly useful for large problems. By using this modular approach, more future modular extensions have a great potential to be developed. From the fields of data mining, the data can be interrogated and extra information can be extracted [215]. Furthermore, there is an extra memory, the Memory of Base Points, that has not been exploited yet, but it exists in the algorithmic structure. It simply logs all the BPs and can be used to monitor the evolution of the BP over the iterations. In addition, this information could be used so as to predict new points without employing any of the other moves.

An extra move has been added to the portfolio of moves, following one of the requirements in section 2.8. It is called *kick* and depends on the number of consecutive improvements and the number of duplicates in the MTM. In the background, *kick* calls on Reduce Move, when a certain combination of less likely to happen conditions are met. This addition was inspired by observing the progress of the optimiser in a few cases (e.g., when investigating the reasons for discrepancies in the performance assessment of the algorithm, shown in Fig. 3.12), where MOTS2 seemed to spend unexpected more computational budget than usual. Inspecting this behaviour was possible because of the existence of memories and logged data, which justify their purpose. Hence, a sensible way to help the optimiser to escape from local optima was to count any of the improvements of the assessment of the MTM, by comparing the change of a quality/assessment indicator such as hypervolume [4] [217]. Thereafter, the duplicates of the objective values for each entry from the MTM are found and counted. Whenever the number of improvements and the number of duplicates exceed certain levels, then the BP is *kicked* in the hope to escape from a local minimum. Both of these threshold numbers are at the user's disposal and can be artificially disabled when they are set to a very high value. Of course, this special functionality cannot solve any possible problem; it only exists as an additional tool for cases where all the other mechanisms fail. It was deliberately developed to be used even less frequently than Reduce Move, as an experimental addition; it was an idea tried but there is not sufficient evidence to judge its effectiveness.

Listing 3.10: Pseudocode of the Method to Activate Kick

```
1 int Container::activate_kick(char const *save_directory, int
      const kick_limit, ObjFunction failedObjectives){
```

[4] Briefly, hypervolume is one of the most commonly used metrics to assess the quality of the trade-off of MOO algorithms; given a reference point to a trade-off, the union of volumes between the reference point and each point of the trade-off defines hypervolume. By definition, wide trade-offs are favoured.

```
2     std::ofstream kick_file;
3     kick_file.open ( save_directory ,  std::ios::out  );
4     int max=0;
5     std::vector<int> count(size(),0);
6     int counter;
7     ObjFunction temp;
8     std::map<ObjFunction, int> occurances;
9     std::set<ObjFunction> counter_table;
10    ObjFunction tempv1, tempv2;
11
12    for(iterator it=begin(); it != end() ; ++it ){
13       counter_table.insert(it->second);
14    }
15
16    for( std::set<ObjFunction>::const_iterator it2=counter_table.
          begin(); it2 != counter_table.end() ; ++it2 ){
17       counter=0;
18       copy_contents(tempv2, *it2);
19
20       for(iterator it=begin(); it != end() ; ++it ){
21          copy_contents(tempv1, it->second);
22          if( compare_error (it->second, *it2) && tempv1!=
                failedObjectives)
23             ++counter;
24       }
25       occurances[*it2]=counter;
26    }
27
28    max=0;
29    for ( std::map<ObjFunction, int>::iterator it3=occurances.begin
          (); it3!=occurances.end(); ++it3){
30       if( it3->second > max )
31          max=it3->second;
32
33       kick_file << std::fixed <<  it3->first << " appears " << it3
             ->second << "times" << std::endl;
34    }
35
36    kick_file.close();
37
38    if(max>kick_limit)
39       return 1;
40
41    return 0;
42 }
```

Another practical extension is the ability of the optimiser to resume the search. This feature is very important when running large scale problems for two reasons.

Firstly, as was shown in [199], restarting tabu search from an already shaped PF and based on previously acquired data can very quickly advance the discovery process of a better PF. Secondly, for operational purposes, the search can be interrupted and start again without any loss of precious data. The implementation is also very straightforward: all the memories and monitoring quantities are stored at a regular pace in external files. In addition, a file called *checkpoint* traces the current state of the search, which stores the current BP, problem description (for instance, the range of variability) and the current number of iteration. Whenever the optimisation process needs to restart, the optimiser picks up the most recent memory files and the additional checkpoint file and continues as usual. Restarting should be used when the problem specifications have not changed.

In order to be applicable in a production environment, the source code of optimiser should not be changed whenever a new case arrives. Hence, several parts can be configured externally from several human readable text files. This approach has the advantage that any user or system can create or edit these files and they can be automated, too. Currently, there is a main configuration file that affects the behaviour of the optimiser for the functionality described above. In addition, there are several other text files that allow the user to provide further information about the optimisation problem. By default, most of them are hard-coded in the application; otherwise, if any of the files are present, the respective settings are overridden.

### 3.1.6 Verifying MOTS2

Verifying MOO algorithms, like MOTS2, is challenging because they frequently employ stochastic mechanisms. Hence, it is not possible to predict the search path since this is expected to be random and different every time one performs the optimisation. The optimiser makes decisions based on the information contained in the memories and from the statistics it gathers. Therefore, the performance of the optimiser will be assessed by carrying out statistical analysis [218]. This is a standard and consistent way that can compare any optimiser on a common basis. The turnaround times are not considered as a performance metric. The specific instance of the methodology is depicted in Fig. 3.6.

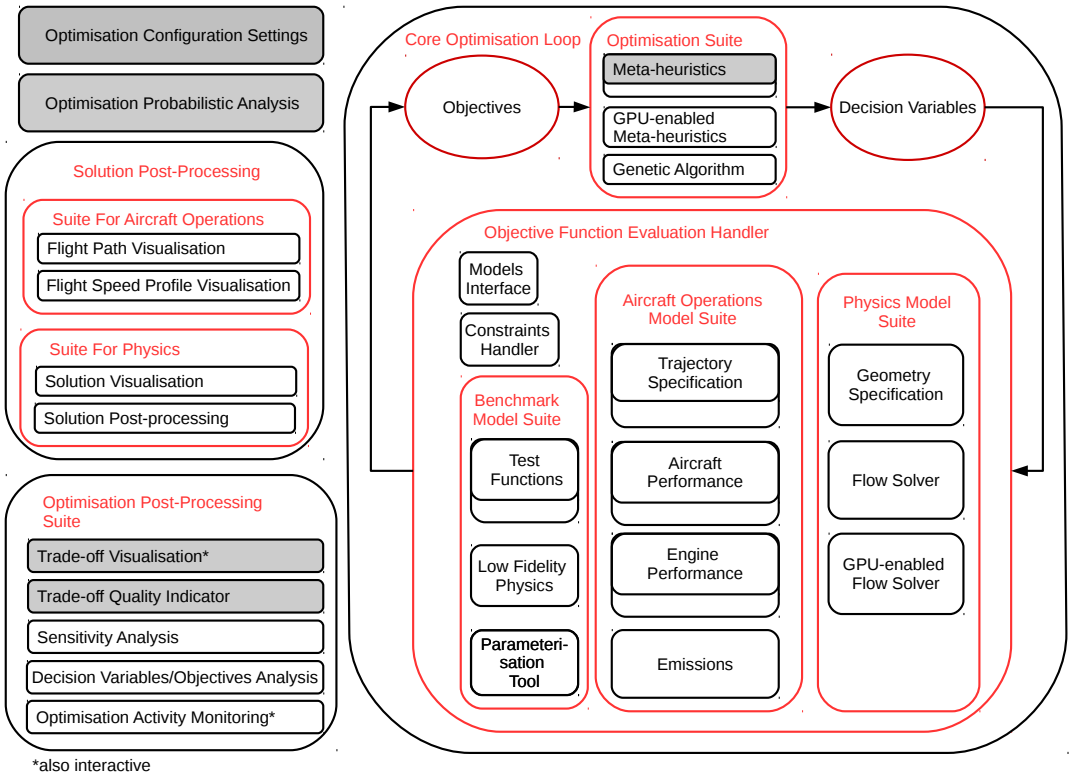



Figure 3.6: Methodology for Verifying MOTS2 against Test Functions

#### 3.1.6.1 Benchmarking the Family of ZDT Functions for Bi-objective Optimisation

The verification methodology described in [219] was followed. Also, as suggested in [193], each optimisation scenario with test functions was executed fifty times. For each test function, the configuration settings listed in Table 3.1 were used, based on the experience of the author with the test functions. The optimisation search starts from the middle (i.e., 0.5) of the range of each variable in decision space, a principle from design optimisation, where the process starts from a well-known design. The combinations of configuration settings used for the verification phase are based on [63]. Although these are neither unique nor optimal, they can satisfactorily achieve the target goals of delivering the target trade-off. It is common practice to test the family of ZDT [193] benchmark functions, as they can expose a number of challenges to the optimisers, listed in Table 3.2, that reflect possible features of the PF of real-world applications. Among them, the ZDT4 test function is the most difficult to resolve for extremely multi-dimensional landscapes, as stated in [193]. First, the collection of PFs for each case was used to produce the Empirical Attainment Function (EAF), applied to the data.

Table 3.1: Optimisation Configuration Settings for ZDT Benchmark Functions

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 20 |
| Call intensification move after # non-improvements | 10 |
| Reduce the search step size after # non-improvements | 50 |
| Initial step sizes (as % of variable range) | 0.1 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | 6 |
| # of variables | 30 (ZDT1, ZDT2, and ZDT3)<br>10 (ZDT4 and ZDT6) |
| # of objectives | 2 |
| # of objective function evaluations | 20000 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | 20 |

Table 3.2: Features of ZDT Functions

| **Test Function** | **Challenge for MOO Algorithms** |
|---|---|
| ZDT1 | convex PF |
| ZDT2 | nonconvex PF |
| ZDT3 | noncontiguous convex PF |
| ZDT4 | large number of local optima PFs |
| ZDT6 | nonuniformly distributed global PF and sparse solutions near PF |

The EAFs are depicted from Fig. 3.7 to Fig. 3.11. For each of the separate test functions, all the discovered optimal trade-offs from the aforementioned fifty consecutive executions were combined by concatenating all the points of the PF. The respective figures were generated by using the external plotting tool described in [196]. In these figures the user can realise the effectiveness of the optimiser on certain well-known problems. Firstly, the user is informed about the quality of the trade-off; best curve denotes the combined set of the PFs with the best performance, whereas the worst curve combines the worst discovered performance.

All the other levels are between the extremes. If the best and worst trade-offs are very close to each other, it means there is not any noticeable difference between the best and worst cases. As this is the verification phase, the PFs are known and well defined in [193], but it is different for each test function. In Fig. 3.7, 3.8 and 3.10, the captured PF is very close to the ideal case for ZDT1, ZDT2, ZDT4, respectively. In Fig. 3.11, 75% of the times the optimal trade-off was found, but there is a big gap until the worst case, which reflects the complexity of ZDT6. However, regarding ZDT3, MOTS2 managed to find the optimal trade-off, but it is consistent only 25% of the times. Considering the worst case for ZDT3, it is very far from the true PF, which partly depends on the selected configuration settings. Overall, it seems that the optimiser demonstrated very good performance for ZDT1, ZDT2, ZDT4 and ZDT6, whereas the discontinuities of ZDT3 caused problems for the search.

An additional metric that involves the hypervolume indicator was created and depicted in Fig. 3.12 in the form of boxplots, for ease of comparisons. This is a more compact representation of the performance of the optimiser in the same test functions and is illustrated in Fig. 3.12, where the hypervolume indicator is applied to each of the PFs and the hypervolume values are normalised to the maximum value, for each test function separately. Simply, the aforementioned performance of the trade-off between two objectives is cumulated into a single value. Regarding the boxplots, the spread of the box and its associated whiskers represent the variation between the best case and the worst case. If the whiskers are close to each other, i.e., the boxplot is thin, then the variation is lower, which reflects the consistency of the optimiser. Therefore, as shown in Fig. 3.12, ZDT1, ZDT2, ZDT4 and ZDT6 demonstrate comparable performance, and they practically confirm that for these test functions the algorithm behaved decently. Although ZDT3 presents high variation, which is another way to illustrate the performance depicted in Fig. 3.9, this is still acceptable because the minimum PF was also captured, which demonstrates the ability of the optimiser to find out the true PF. Nevertheless, achieving the true PF with fewer evaluations is related to finding out a better combination of configuration settings. An immediate improvement would be to implement a feature that automatically adapts the configuration settings. Finally, as already mentioned above, the performance of the optimiser is satisfactory in ZDT1, ZDT2, ZDT4 and ZDT6, whereas the PF of ZDT3 was decently captured 25% of the times.

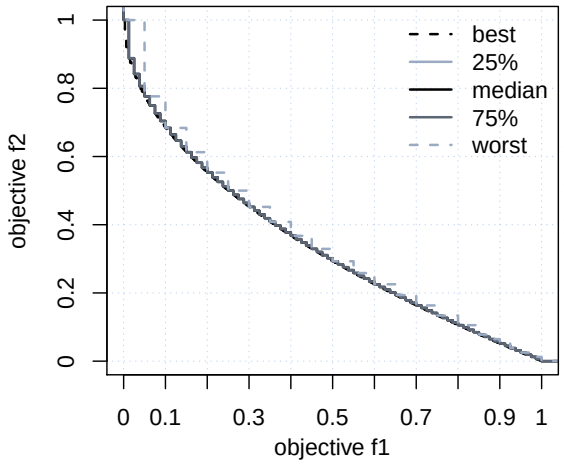



Figure 3.7: ZDT1 EAF

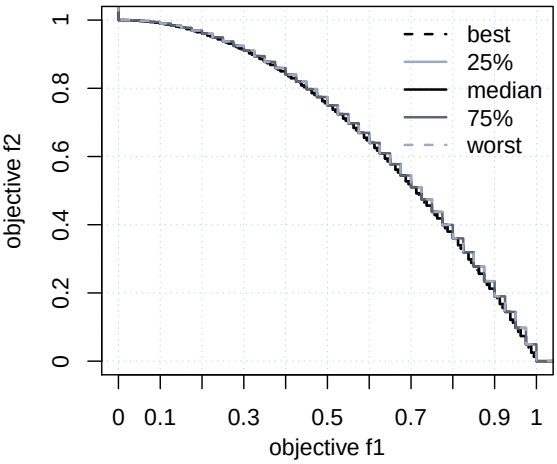



Figure 3.8: ZDT2 EAF

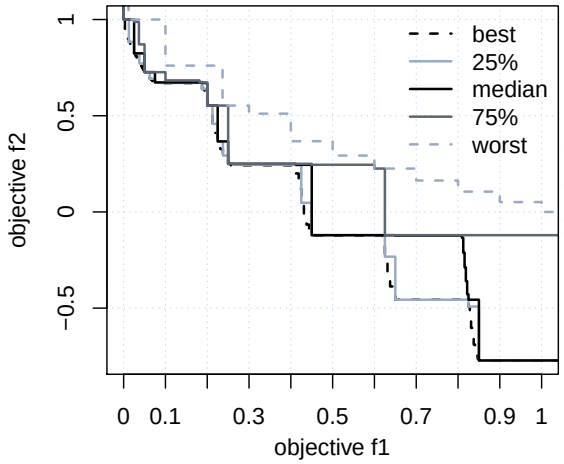



Figure 3.9: ZDT3 EAF

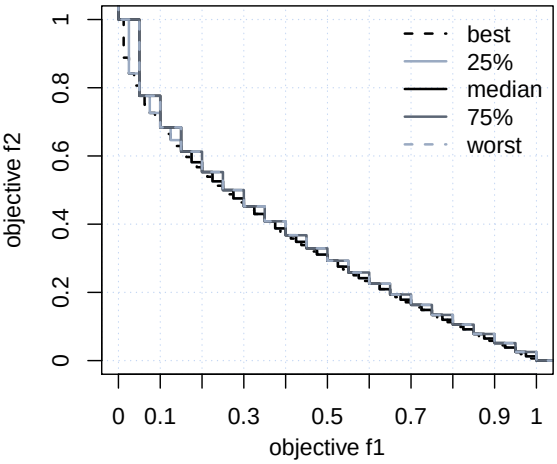



Figure 3.10: ZDT4 EAF

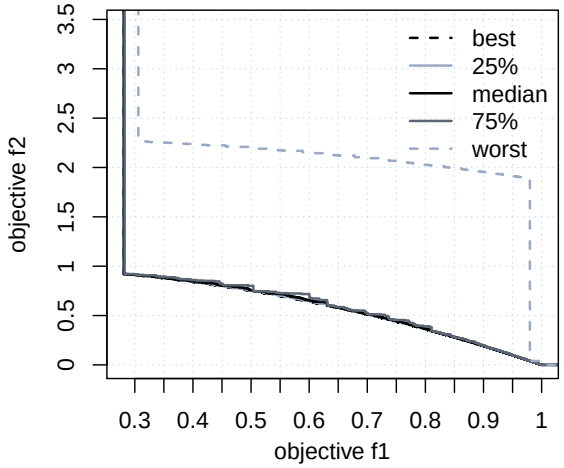



Figure 3.11: ZDT6 EAF

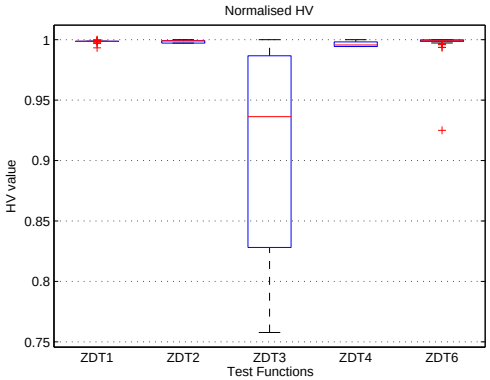



Figure 3.12: Normalised Hypervolume Boxplots for ZDT Test Functions

For comparison/reference purposes and linking with the following chapter, the ability of NSGAMO (in the plots referred to as NSGAMO3) to capture the target trade-off after 22000 objective function evaluations for three of the ZDT functions is demonstrated in Figs. 3.13- 3.15. This is part of a comparison performance study of NSGAMO against state-of-the-art MOO algorithms, and was conducted in [220].

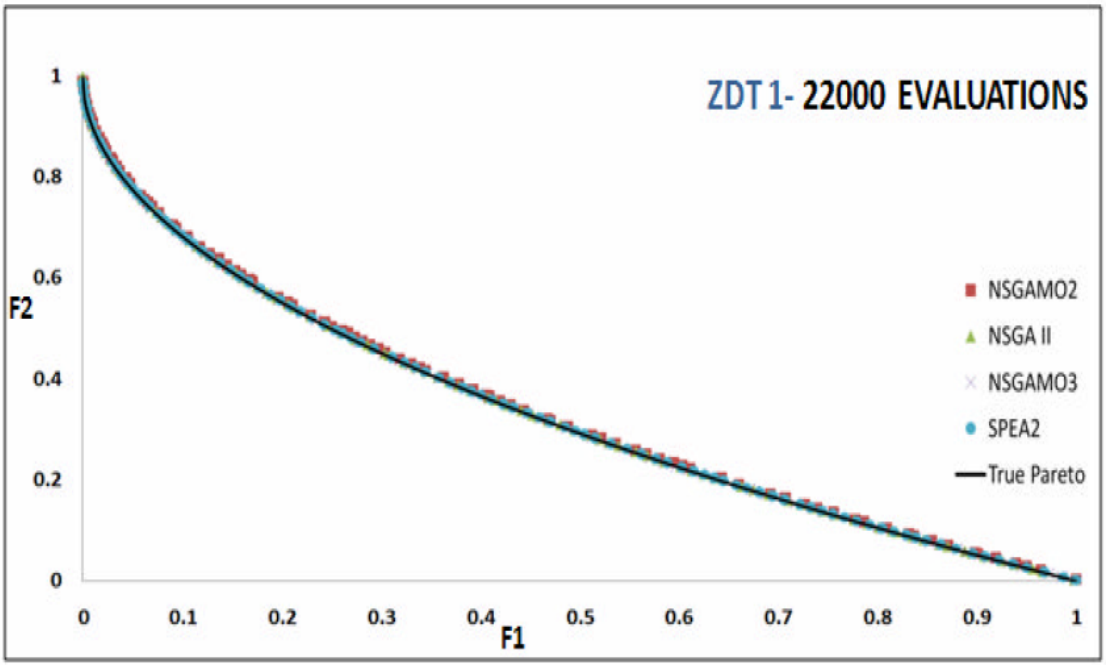



Figure 3.13: Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT1 Against State-of-the-art Optimisation Algorithms [220]

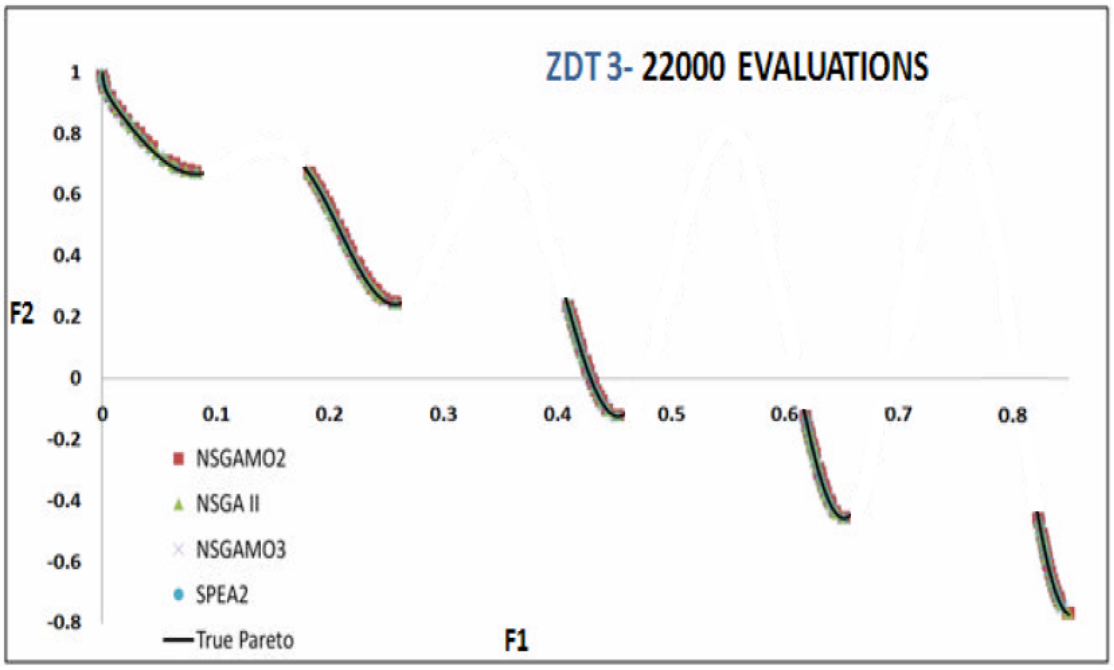



Figure 3.14: Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT3 Against State-of-the-art Optimisation Algorithms [220]

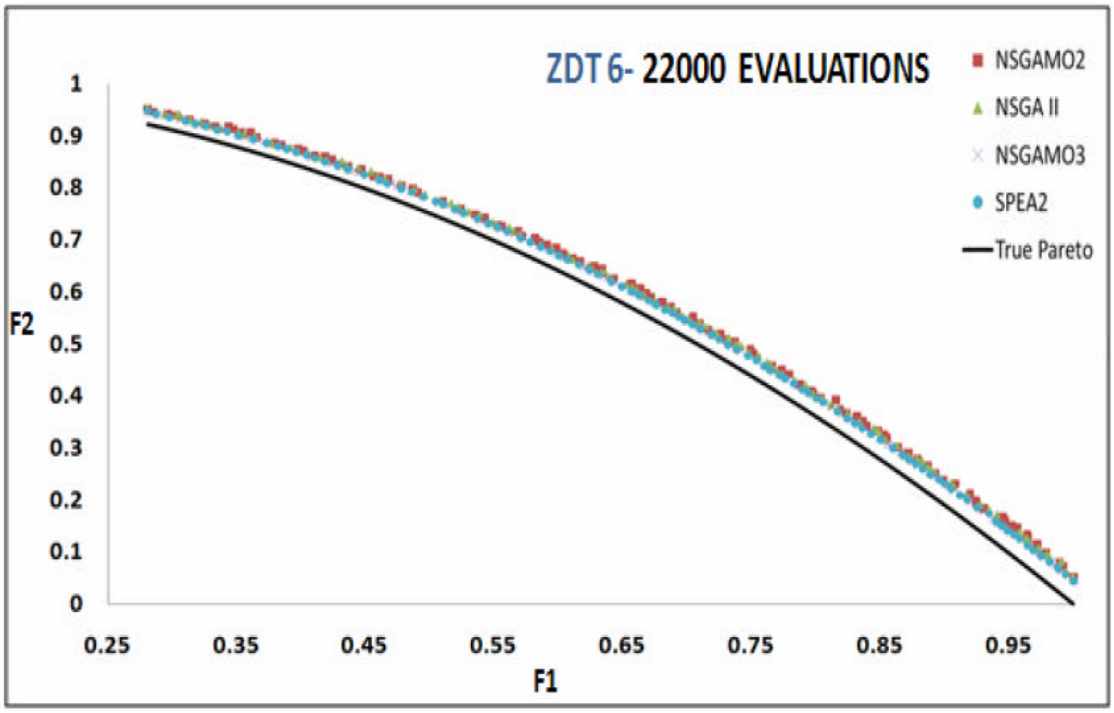



Figure 3.15: Demonstrating the Performance of NSGAMO (in the Plots Referred to as NSGAMO3) on ZDT6 Against State-of-the-art Optimisation Algorithms [220]

### 3.1.7 Validating MOTS2

The test functions used in the previous subsection can be used for verifying the functionality of the optimiser, but they do not reflect the needs of real-world applications. The former are simple and computationally cheap to evaluate, whereas the latter can be of arbitrary complexity and extremely costly to use. In addition, this high abstraction level can lead to a severe loss of information concerning the nature of the original problem. Therefore, it is important to validate MOTS2 by demonstrating that it can satisfactorily capture the behaviour of objectives that are conflicting by nature in a computational physics problem. The specific instance of methodology is depicted in Fig. 3.16.

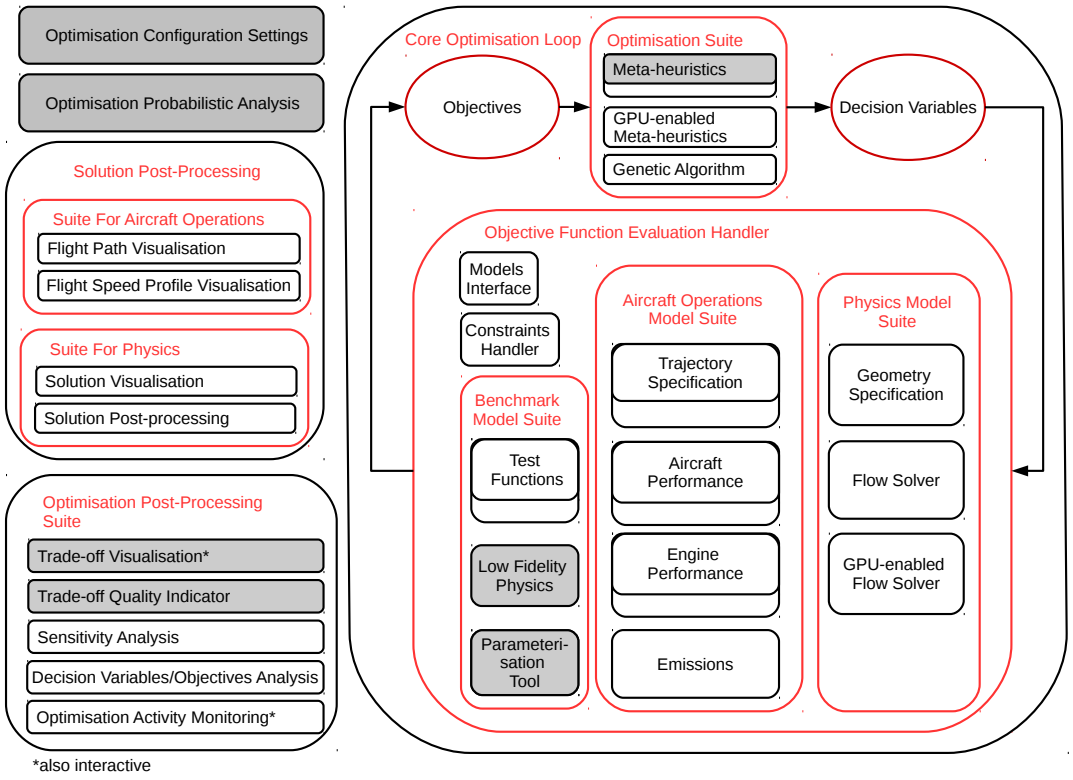



Figure 3.16: Methodology for Validating MOTS2 against Low Fidelity Physics

#### 3.1.7.1 Airfoil Shape Optimisation

A relatively simple scenario that proves the effectiveness of the optimiser in the shape/profile optimisation of an airfoil. This is an aerodynamic application, which was originally proposed in [221], which measures the performance of an airfoil by changing its shape (a combination of upper and lower surface). More specifically, the objectives are to maximise the lift and to minimise the drag of a 2D airfoil, which are conflicting in nature. In the open literature there are several techniques which can derive these two quantities, but the choice of the most appropriate one, in terms of accuracy, is out of the scope of this research. Herein, the goal is to validate MOTS2 and demonstrate its linking with external software, as a black box.

The physical representation of the airfoil surfaces is a very challenging task. The points should be distributed mostly along the leading and trailing edge because the velocity changes rapidly in these regions. By using exact 2D coordinates of the upper and lower airfoil surface, the velocity distribution is calculated. Imposing additional structural and shape constraints could help without mitigating completely the generation of irregular shapes. For instance, cusp-like configura-

tions that will alleviate drag due to formation of normal shock waves cannot be proposed. However, this should be investigated in the future. In order to minimise the complexity of the problem, a representation scheme, called Free Form Deformation [222], is used to parametrise the different geometrical airfoil profiles, which is a technique that manipulates any shape in a free-form manner. By specifying a few control points smooth curves are generated, which is to be used to represent the shape of the surfaces of the airfoil.

The two objectives are derived by using a software called XFOIL [223, 224]. It belongs to the category of panel methods for low-speed inviscid flow, where the body is discretised in terms of singularity distribution on the surface. Compared to conventional CFD methods, where the flow is calculated around the test object over a large surrounding field, panel computations around the surface are only required. So, the evaluation time is significantly low. This method has certain advantages for specific cases. The only prerequisite is to discretise the surface, while the computational cost is relatively low. The downside is that only subsonic flows of low Mach number can be studied. Consequently, inaccurate skin friction estimations, wave drag predictions and lift coefficient estimation will eventually lead to poor results in the optimisation process. In any case, the application is fit for the purpose of validating the optimiser and that is the focus here.

A top level description of the system to be optimised follows. Essentially, the system is a wrapper that couples Free Form Deformation and XFOIL together, which are combined with additional structural constraints on the shape of the airfoil. Initially, a decision vector of 8 real-valued decision variables is received as input, which is a set of control points for the Free Form Deformation model. A series of exact 2D coordinates are generated for the upper and lower surface of the airfoil, and these are fed into XFOIL. These coordinates are be used to calculate the pressure distribution over the airfoil surfaces through XFOIL and will deliver the respective lift coefficient ($C_L$) and drag coefficient ($C_D$) for a target airfoil configuration. This airfoil is simulated on a series of different angles of attack, and the average $C_L$ and $C_D$ are obtained. These are the 2 objectives, as real numbers. This parametrisation intelligently suppresses the originally high dimensionality of the problem. A new geometrical arrangement is generated by adjusting the control points that manipulate the whole surface. Regarding the constraints, only hard ones are applied here, which are part of the wrapper. More specifically, two thickness limits of the airfoil are specified at 25% and 50% of its length. These are hard-integrated into the objective function evaluation model. If the constraints are violated, the calculated objectives will be set to a very large numerical value, as a penalty. Regarding the optimisation search, this is considered as a single evaluation of the fitness of a certain decision vector and, thus, will be repeated many times. This is the most computationally intensive part of the

whole optimisation search and can be considered as a modular black box. After all, this application can deliver good results for low-velocity cases and suggests to use MOTS2 in more demanding applications. Using less accurate but faster methods (such as XFOIL) in the early stages of the design process is preferred. Thereafter, more accurate simulation models should be considered and, ultimately, computationally expensive CFD tools could be employed. Clearly, optimising the behaviour of aerodynamic problems via a high fidelity CFD tool is of paramount importance in the whole engineering design process, as described in [92, 93].

The optimisation problem will use the aforementioned system as a black box. As a starting point for the optimisation process, the airfoil profile of NACA [5] 0012 blade is used. This is a well known geometrical arrangement whose performance is known in advance. The shape of NACA 0012 has been constructed by using 8 control points via Free Form Deformation. When these control points are set to default values, that is 0.0 for decision variable, the profile of the aforementioned airfoil will be generated. Any change in the control points is a relative modification to the position of the original control points and the default value is zero. Therefore, a new airfoil shape is uniquely represented by using 8 decision variables, whose range lies between -0.4 and 0.3. Consequently, a number of different combinations of the 8 decision variables will be proposed by the optimiser so as to maximise the $C_L$ and minimise the $C_D$. In the figures below, the $C_L$ maximisation is equal to the minimisation of the negative $C_L$. It is to be reminded that the constraints are hard-coded to the black box system. So, the optimiser does not need to know anything about them. Moreover, the obtained objectives for each new design will be normalised against the objectives of the datum point. Hence, the results of the optimiser are expressed in terms of relative improvement. For minimisation purposes, the lower the objective value is below 1.0, the better the design is from the initial performance. This will be repeated for a fixed number of objective function evaluations, which is set to 3000.

The configuration settings of the optimiser are listed in Table 3.3 and follow the same logic as described in [63]. Based on the author's experience with this particular application, the problem is not expected to be very complicated and the threshold of the *kick* is specified to a very large value, so it is never going to be activated. The first three values are related to the search strategy. The smaller the number is, the more frequently the respective move will be performed. The termination criteria for the optimisation process is the number of objective function evaluations, which is set to 3000, and the obtained PFs are depicted below. The initial step was resolved by previous studies, which carried out a sensitivity analysis on XFOIL. The Search Step Retain Factor (SSRF) and random sampling were set arbitrarily. According to the developer's experience, halving the SS and acquiring

[5] US federal agency to undertake, to promote and to institutionalise aeronautical research

about as many samples as the number of variables turns out to be a decent choice. In this specific case, the turnaround time of one objective function evaluation can take up to 1 minute. This number is not very large, but in more complex problems it could be of the order of magnitude of hours or days. So, a relatively low upper limit of evaluations was used. The ability of the optimiser to reveal a PF as close to the (unknown) true PF is the most important aspect for efficient and reliable multi-objective optimisation, when validating an optimiser [6]. Whenever the true PF is not known, delivering an approximation set as close to the corner of interest (bottom left in the trade-off figures) is satisfactory. Again, the number of individual regions and STM size were chosen based on previous experience and following previous studies. Although this strategy seems to be satisfactory here, this is not always the case. In fact, deeper sensitivity analysis should be performed in order to resolve the most important variables or a combination of variables and the search space should be divided according to this information.

Table 3.3: MOTS2 Configuration Settings for Airfoil Shape Optimisation

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 25 |
| Call intensification move after # non-improvements | 15 |
| Reduce the search step size after # non-improvements | 45 |
| Initial step sizes (as % of variable range) | 0.07 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | 6 |
| # of variables | 8 |
| # of objectives | 2 |
| # of objective function evaluations | 3000 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | 15 |

Similarly to the previous subsection, the same statistical testing (i.e., fifty instances) was applied again. The EAF plots were produced by using the tool described in [196] and the hypervolume boxplots are generated as before. Both

[6] As shown previously, in the verification, the optimal PF is always well known and is frequently easy to represent via mathematical expressions.

Fig. 3.18 and 3.17 were produced by following exactly the same process. Obviously, the discovered curves of the PF are very similar, and they demonstrate the conflicting nature of $C_L$ against $C_D$, where the corner of interest is at the bottom left. It is to be reminded that maximising $C_L$ is equivalent to minimising negative $C_L$ in terms of optimisation methodology. Furthermore, the normalised hypervolume seems extremely robust with very slight variation between the worst and the best case. Hence, in terms of software engineering/development, it was demonstrated that MOTS2 is successfully validated on a real-world problem, where it discovered the expected behaviour. It is now ready to be used on other problems, and its performance can be compared against other optimisers [7], see section 4.1.1.

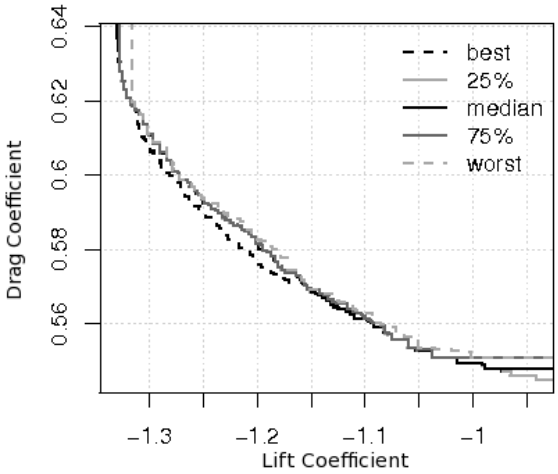



Figure 3.17: XFOIL EAF

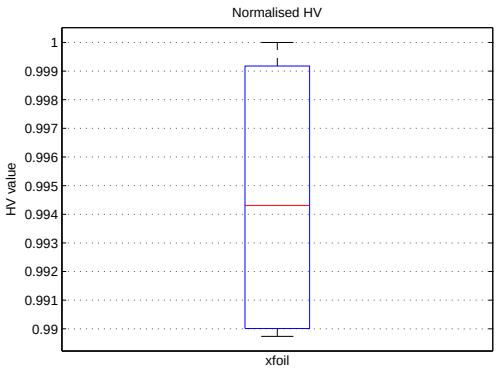



Figure 3.18: Normalised HV Boxplots for XFOIL

## 3.2 Parallel Multi-Objective Optimisation Algorithm

So far, all the discussion in the previous section mainly described the standard operation of the algorithm, which is in a sequential mode. However, the ability to operate in parallel is a frequently required feature for any application in order to fully exploit the available computational power of modern computers and to enable users to carry out their studies in more reasonable time frames, also possibly at a lower cost. Hence, this section presents the parallel perspective of the algorithm for regular computers and for specialised hardware. The design and implementation described in the above section are expanded by providing further implementation details that were not covered before, by introducing parallel evaluation on any external framework, and by porting the algorithm into CUDA. The

[7] validation in terms of optimisation process

high computational cost due to the problem complexity is addressed by employing GPUs, which alleviate the computational intensity. The main challenges of the re-implementation are effective communication and transparent integration with the optimisation procedures. This version was originally created to cope with the high-dimensionality of real-world applications.

As an extension of the previous work, this section attempts to enable the optimiser to operate in parallel. Standard practices in parallel implementations are revisited in the first subsection. Additional details regarding the parallel version are presented in the following sections. A simplistic expansion of the optimiser, shown next, allows one to evaluate several decision vectors in parallel through any external system. Later, practices in MOO for problems of increased scalability are explored. The development of the extension of MOTS2 that performs optimisation by using GPUs is presented below, which is also verified. Preliminary results and future work are discussed in the final subsection.

### 3.2.1 Trends in Parallel Implementations

There are several advantages when running an optimiser in parallel [55, 209]. The first improvement is the reduction of elapsed time, which is very crucial in real-world applications. Second, the robustness of the algorithm can be enhanced by searching a larger fraction of decision space, which could potentially result in a more efficient search scheme. Third, the quality of the obtained solutions in terms of reduced search time and convergence will be improved by using parallel models for metaheuristics. Fourth, large-scale problems could be tackled, either the ones that are impractical to solve with sequential optimisers or the ones that require more computations to achieve a certain level of accuracy (usually system models that represent a process [54]). Furthermore, the effort to calibrate the optimiser to obtain better results could be reduced, which increases the consistency of the optimiser on various sets of problems. The main trends are methods with asynchronous cooperative multi-threading capability and hybrids. Using either a *pure* tabu search or some hybridised version requires realising the subtle interactions caused by employing complex cooperative strategies, where statistics and performance measures will help, such as the execution status of threads. This is the point where design patterns and object-oriented designs will be of great assistance, as higher programming skills are required.

The most common requirement of real-world processes is to deliver sensible results in practical time intervals. This is the first reason to expand any existing method to operate in parallel in the hope of shrinking the turnaround time from the start of the problem to the final solution. As the number of dimensions increases the computational time rises exponentially and this is where metaheuristics can make a significant contribution; they cope well with the dimensionality

and the cost of highly dimensional problems. In addition, if they are enhanced with parallelisation features, such as direct parallelisation and hybridisation that are described in [55, 225], then their overall value is even more considerable; even if sometimes they cannot deliver the ultimately best global solution. This implementation was inspired by [212].

Two important criteria of parallel implementation are: how fast they can deliver a solution and then how far that solution is from the most optimal. The combined evolution of the network, protocols and hardware (computational, intercommunication, storage) is a significant drive for the future of such applications and they should be considered for sustainability. The latter cannot be known in real-world cases, but the distance covered from the start (the first base point) of the search can be used as an indication of search progress. Empirical and ad hoc evaluation of any metaheuristics should be avoided. Rather, any experimental analysis has to be supported by statistical analysis and parallel performance metrics [226]. Furthermore, any changes or additions to metaheuristics should be reflected in terms of improvement of the aforementioned metrics of parallel performance.

The algorithms could evolve by improving either the software or the hardware or both. Hardware is related to the parallel architectures and software is linked to the parallel programming models [227]. Focusing on single-solution based metaheuristics, the most common parallelisation techniques are parallel multi-start, parallel moves and move acceleration. However, parallelisation of computation can always be used to speed up a single evaluation of one or more objectives. In multi-objective cases, parallelisation could be even more productive. Algorithms that take into consideration the heterogeneity of allocated computational resources, especially cluster of workstations, will be of particular help in many cases [228].

The underlying hardware equally contributes to the overall performance of parallel optimisers. By default, running the optimiser on different GPUs could affect the execution time, as the number of threads, the memory size and speed of Symmetric Multiprocessors (SMs) vary. Additional factors that can affect the performance, as experimentally tested in a variation of particle swarm optimisation (as a type of metaheuristic optimiser) in [229], are the selected data structure (partly influenced by CUDA) the number of kernels and the memory management techniques, which also depend on the specifications of the underlying GPU. This is going to have a notable impact on applications where real-time performance is required, such as pattern recognition and computer vision. Hence, a change in performance is expected when running MOTS2 on GPUs.

### 3.2.2 Parallelisation Strategy for External Evaluation

As part of the GATAC framework, the optimisation algorithm had to transparently communicate with all the other available modules and should exploit the

ability of the framework that brings together many heterogeneous resources. The most straightforward way to do this is by exchanging simple human readable text files. These could be generated by any user or system and could be automated. Following GATAC's architecture, the files are collected by the main system, they are processed and then the computational workload is distributed to the pool of resources. As expected, this approach can be applied to cases when the number of computational resources is large.

As part of the software design pattern, the optimiser has a single instance of a class called Evaluation Manager, which is responsible for evaluating any number of group(s) of decision variables that correspond to the decision vector(s). It does not make any difference to the optimiser how this happens, as long as the corresponding objective function values are generated and assigned to the originally asked decision vectors. As usual, extra statistics are generated such as the number of feasible objective function evaluations and failed evaluations. An instance of a class Constraints Manager is part of the Evaluation Manager and checks the feasibility of decision vectors and objective function values. In addition, the Evaluation Manager interrogates the HISTORY to save computational budget if any requested decision vector has been evaluated before. So, all the decision vectors are evaluated in a single way, either as a group or individually. This very flexible structure allows one to expand the evaluation of objective functions in many possible ways, both within the compiled code or externally by any other system.

The object-oriented design of MOTS2 allowed the optimiser to transparently expand its capability without changing its algorithmic structure or the memories. As described above, the Evaluation Manager could operate in many different ways. Therefore, if the parallel evaluation through GATAC is required, the Evaluation Manager can assist. The only place in the optimiser that it is sensible to use parallel evaluation is the Hooke and Jeeves Move, where a number of decision vectors is proposed for evaluation. The rest of the algorithm requires, at most, a single evaluation. In any case, when the objectives are going to be evaluated externally, the Evaluation Manager generates a list, a simple text file, that contains all the necessary decision vectors to be evaluated. Obviously, if any of the decision variables violate any constraint or the decision vector has been evaluated before, an appropriate penalty value or the corresponding objective function value will be assigned, respectively, in order to avoid any unnecessary computational overspend. Then, the optimiser waits until another list appears that contains the corresponding objective values in a 1-1 fashion. Any external system, GATAC in this case, can pick up that list and produce the list with the objective values. Following that, the optimiser collects the response of the system and internally matches the decision vectors with the objective values. The list of the decision vectors to be evaluated can contain many entries, and it is up to the external

system whether these will be evaluated sequentially or in parallel in any order. Here, GATAC, after receiving the list of decision vectors, separates the entries and allocates each entry to separate resources. Then, it gathers back all the answers and creates the interface file for the optimiser. This objective function evaluation is transparent, external, and parallel. Moreover, it does not affect the operation of the overall system that performs the evaluation. Of course, the evaluation can be asynchronous.

### 3.2.3 Scalable Multi-Objective Optimisation Problems

The performance of MOO algorithms is usually assessed by using benchmark problems. These are mathematical functions expressed in an analytic formula and can be calculated quickly. Their characteristics are well-known and they can demonstrate a number of features that replicate situations met in real-world cases, such as discontinuous PF and multi-modality. The most typical ones are test instances in [230], ZDT [193], DTLZ [231] and WFG [232, 233]. All these benchmark suites are scalable in the number of decision variables, whereas the last two are also scalable in the number of objective functions, too. Usually, the optimisers are given a limited computational and time budget, in which they are expected to discover the optimal trade-off of the problem. Significant contributions with an industrial impact that also serve as benchmarks have been addressed in [234–236].

The scalability of the problem in terms of the number of decision variables and the number of objective functions is a very active field. Although the aforementioned problem instances have a relatively small number of variables, usually less than 30, in real-world applications the number of decision variables can grow from the hundreds to thousands. The capability of metaheuristic optimisers to scale in the multi-objective domain has been studied in [237–240].

When the problem scales up and more variables are involved, due to the curse of dimensionality, the standard methods soon struggle to deliver acceptable solution(s). Capturing the target PF becomes increasingly more difficult. Validating the optimisers becomes very challenging and, frequently, the computational resources do not suffice. Therefore, alternative methods and tools are required.

### 3.2.4 Massive Parallelisation Strategy

Here, the main contribution is to illustrate an approach that can handle multi-objective continuous problems with an increased number of decision variables, as frequently occur in real-world applications. Furthermore, CUDA was selected as a very promising technology for the future. To the best of the author's knowledge, it is the first time that a multi-objective tabu search is implemented by using CUDA. This new version is an in-house development and is based on MOTS2, which

has been used in two other real-world applications, as listed in 1.5. Important issues related to high-dimensional data parallelism on GPU architectures for multi-objective optimisers are addressed, which are also associated with the selected implementation.

#### 3.2.4.1 Software and Hardware Perspectives

In parallel implementations, GPUs are often considered due to their performance improvements and programming accessibility, which result in greater processing power, storage capability and very high-cost effectiveness [241, 242]. In order to harness the computing cycles and to get the maximum potential out of the hardware, algorithms ought to be re-designed and re-implemented bottom-up, starting from the chip. By implementation, MOTS2 is a local-search-based optimiser and it is believed that it would be very beneficial to run on GPUs because both software and hardware are strongly based on locality and memories. The ideas and concerns discussed below were strategically considered for a sustainable design, appropriate for engineering applications. The general and specific parts of the development of the new version of the optimiser are presented in the following pages.

Starting from the software perspective, it is important to understand the features of the selected optimiser (MOTS2) by following the classification of metaheuristics [55]. First of all, MOTS2 is nature-based in the sense that it attempts to mimic human intelligence when searching through decision space. It uses a variety of memories, each for slightly different purposes. At the top level, the majority of the steps are deterministic, but some minor parts are stochastic, so in that sense it is mixed. It performs a single-solution-based search. It is an iterative optimiser, because every search starts from a single solution [8] that is transformed in an iterative manner. Simply, this list characterises the performance of MOTS2 and attempts to describe an appropriate environment to be adapted to CUDA.

From the hardware perspective, the characteristics of the selected hardware (GPU) will indicate what software features are to be implemented, simply because only the behaviour of software can change. By production, GPUs contain a large number of cores that can execute the same instruction in parallel, but the algorithmic logic must follow the principles of the Single-Instruction Multiple-Threads (SIMT) model in order to be effective. This makes GPUs ideal for applications, where different data are similarly processed in large batches, where each batch represents a set of solutions of the optimisation problem. In addition, the storage capability of GPUs consists of hierarchical memories with variable size and access speed, usually the larger the memory, the lower the access speed is and vice versa. However, high-end models combine both these features at a much higher

[8] one of the principles of design optimisation

cost. In practice, this means that the right amount of the data of interest should be at the right memory level and at the right moment. Moreover, since groups of cores can access the same memory location, in order to avoid latencies, this has to be implemented in a pre-scheduled manner. The latter should be structured to be as asynchronous as possible because the hardware decides the execution order. So, the local memories should be manipulated in a way that the decomposed data could be used independently but very precisely planned as if they were part of a factory pipeline that processes batches of products. In addition, transferring data between CPU and GPU has to be scheduled in a way to hide data latencies; one part of the card will carry out calculations and the other will send/receive data, preparing the outbound/inbound batches. The major point is to keep everything local with separate concerns at the very fine level of data decomposition. Consequently, this will simplify the design, will minimise the execution branching and will suppress any other delays. These features ought to be used so as to fully exploit the underlying infrastructure, as suggested in [243].

#### 3.2.4.2 Design Decisions and Concerns

The key requirement of an application when deployed on GPU architectures is to maximise the utilisation of the available resources. This is possible by balancing the ratio of processing, storage and data transfers. MOTS2 was mainly re-designed to match the capabilities of the hardware. Nowadays, computational infrastructure is very heterogeneous and this is a considerable challenge since the overall performance depends on individual specifications of the modules and how they communicate. In addition, the design should transparently scale by adding more powerful hardware, in terms of specifications, and this is an absolute requirement since the number of required objective function evaluations will exponentially increase as the problem size grows. In any case, it is not possible to come up with a design that would be equally good in every case.

In order to be sustainable, the functionality of the optimiser is split so that each part of the optimiser should be carried out in the computational environment that can perform better. Following the CUDA computational model, the execution starts from the host and occasionally the device is called to support the optimisation process. It is not possible to fully deploy an implementation on GPU and dismiss the host. Also, a principle from the field of HPC dictates that a fraction of the optimiser is not worth parallelising, but this has to be kept to a minimum so as to increase the overall computational efficiency.

When designing a metaheuristic optimiser, the challenge is to combine exploration (diversification) and exploitation (intensification) features at the right balance. However, the nature of real-world applications makes it difficult to precisely decide the balance. The effective operation of the optimiser lies within a

margin of confidence and the user's experience. Since MOTS2 is local-search-based and single-solution-based, the intensification features are going to be used more frequently and this will have a great impact during the runtime.

There are two common conceptual points between MOTS2 and a GPU's operation. Fundamentally, MOTS2 manipulates and is based on memories to guide the optimisation search. Partly, the great performance of GPUs depends on memories. In addition, MOTS2 is a local-search-based optimiser and GPUs access data by following local patterns. In contrast, recently investigated solutions are not considered by MOTS2, since they might be tabu, whereas an ideal GPU application should reuse as much data as possible located in the (local) shared memories. The new design ought to take advantage of the concepts of locality and memories, which will be a competitive advantage over other (global-search-based) optimisers and strategies. Therefore, the objective function evaluation for different decision vectors will take place on the device.

Currently, the high level (conceptual) parts of MOTS2, which employ practices from the fields of artificial intelligence, run on the host, whereas the device only evaluates the batches (see Fig. 3.19). Within the device, either all the cores together are used to evaluate the objective function (similar to data decomposition) or each core is responsible for a single objective function evaluation (or even part of it - like function decomposition). Certainly, there is not a clear answer here, as it heavily depends on the intrinsic features of the problem; more importantly, the specifications of a single core are relatively weak, and it is advisable (by the vendor) to manage the cores in groups so as to make use of the data locality of hierarchical memories. In terms of parallelisation techniques, GPUs seem to be ideal for more acceleration. In any case, the host would transparently delegate only the computationally intensive evaluation part to the device, as was also suggested in [244, 245].

Table 3.4: GPU Hardware Specifications to Benchmark the Performance of MOTS2

| **Quadro 1000M** | |
|---|---|
| CUDA Driver Version / Runtime Version | 5.5 / 5.5 |
| CUDA Capability | 2.1 |
| Total Amount of Global Memory (MB) | 2048 |
| CUDA Cores | 96 |
| Stream Processors Rate (MHz) | 1400 |
| Memory Clock Rate (MHz) | 900 |
| Power Consumption (W) | 45 |

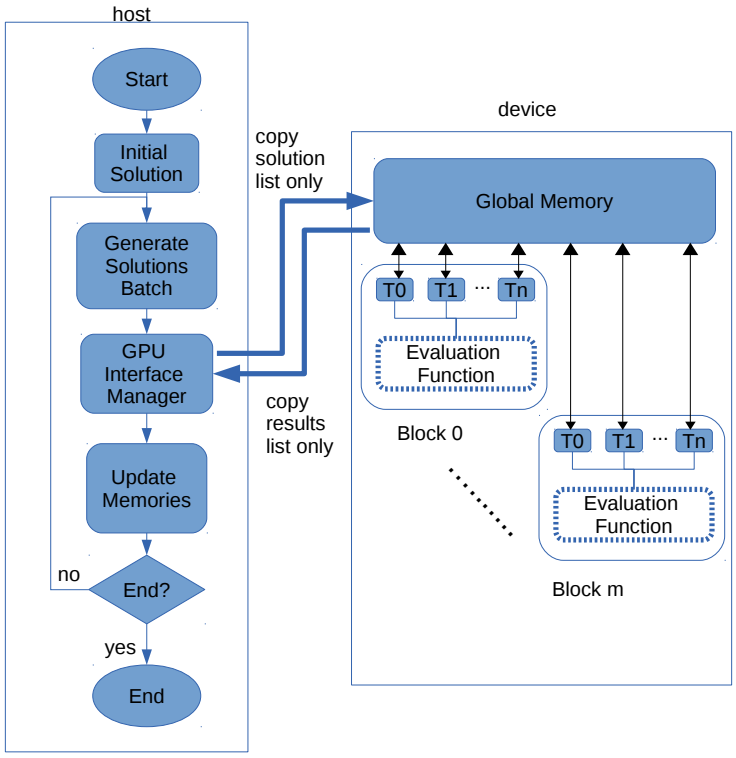



Figure 3.19: Parallel Evaluation Scheme on GPU

#### 3.2.4.3 Implementation

The idea is to gradually reform parts of the optimiser (i.e., both algorithmic and data structures) in a way to align with the SIMT model, which also aligns with a more object-oriented approach, where the concerns have to be separated and ought to be as modular as possible. More importantly, within the device, the optimiser should execute without branching and should exploit the local features of local-search (such as the generation of permitted neighbourhood of decision space) so as to match the locality of data that GPUs operate on.

Currently, there are two approaches: firstly, the appropriate parts should be moved directly onto the device code and should be modified accordingly. For instance, the local neighbourhood could be generated within the device, while the host will not be aware of this and will have no access to it. Secondly, the logic of the optimiser should be altered in a more SIMT-friendly way. For example, although the Pattern Move is an enhancement to the logic of a sequential execution (on CPU), it performs a single evaluation, which leaves all the rest of the GPU cores

idle whenever it occurs.

As already explained above, the high level and complex parts of the code will be performed by the host, whereas the evaluations will be carried out in batches by the device. At a certain stage, the optimiser will generate the candidate solutions for evaluation and will aggregate all the feasible and new decision vectors in lists. Only these will be sent to the device, where each solution will be asynchronously evaluated in the most efficient way, managed by the CUDA scheduler. Thereafter, only the results will be returned back to the host, where they will be matched with the initial solutions. This scheme will be repeated until the stopping criteria are met and is illustrated in Fig. 3.19. It is important to note that due to the weakly-ordered parallelism, see [246], extra care is required to secure that the reads and writes do not interfere and they have to be synchronised within the GPU.

All the CUDA specific execution settings are configured by the optimiser. The number of unique solutions that have never been evaluated before, also called the size of the solution list, will determine the amount of parallelism. More specifically, the CUDA kernel configuration parameters are automatically adjusted from the size of the problem; the number of CUDA threads is decided by the number of solutions, which, in turn, defines the number of CUDA blocks. For simplicity, a single dimension grid is employed in CUDA. This approach guarantees that the right number of threads will be spawned in order to achieve high-performance throughput and maximum utilisation of the other resources.

In terms of access patterns, the standard practices were followed from [243]. At the host side, the generated candidate solutions are checked, sampled, ordered, filtered for existing solutions and sent to the device. Most likely, the solutions in adjacent rows will have some degree of similarity, especially when the number of variables is large enough. The similarity of the candidate solutions could be combined along with a different representation of solutions within the GPU so as to gain more performance out of the data locality, as suggested in [242]. Even if the number of samples remains the same, moving an additional part into the GPU, will speed up the overall optimisation search. The global thread identification number, which uniquely distinguishes a thread within the overall grid domain, was used to load the received solutions. Finally, the results are sent back to the host in a coalesced manner.

### 3.2.5 Assessing GPU-MOTS2 on a Simple Benchmark

In this subsection the preliminary results of a feasibility study are demonstrated, for verification purposes only. The goal is to further develop MOTS2 to deal with problems with a great number of parameters. As the elapsed time is expected to increase, it is sensible to employ alternative technology to save time and also to demonstrate where it is suitable to use the variant that runs on CPU and

where it is advisable to employ the processing power of GPUs. The developed optimiser has already been tested against the ZDT functions (see subsection 3.1.6) at the standard size of variables and delivered the expected trade-off. Here, ZDT functions were selected because the evaluation time is negligible, and the focus is only on the number of parameters. Normally, in real-world cases, if the number of parameters increases, so does the evaluation time of the objective functions, as they could represent more complicated processes. This time, the same process is applied to the GPU variant of MOTS2 with the test function ZDT2 on problem instances with 30, 120, 270, 480 and 750 variables. The numbers were selected so as to progressively increase the size of the problem. Following the completion of the optimisation search, the wall-clock time has been measured. In addition, the performance of the GPU, whose specifications are listed in Table 3.4, is compared against the host's CPU, which is a regular Intel i7-2720QM at 2.2 GHz. The range for each decision variable is between 0.0 and 1. Moreover, the configuration settings for MOTS2 are listed in Table 3.5 and most of them have been preserved from the previous verification phase (see subsection 3.1.6). However, some of the values were chosen as a function of the number of decision variables for convenience. Similarly to the benchmarks carried out in sub-subsection 3.1.6.1, the optimisation search starts from the middle (i.e., 0.5) of the range of each variable in decision space. The intention is to assess what the performance of the optimiser is and how it behaves when the problem size increases because the problem complexity increases dramatically. The scale factor and the type of test function were chosen arbitrarily, but the scope is to replicate real-world conditions. Here, it is assumed, without loss of generality, that there is a class of real-world problems with a concave and continuous PF. Currently, the focus is to locate any weakness and potential improvements. Furthermore, the intention is to check that the optimiser works well and can manage the number of decision parameters. Without loss of generality, this capability is expected to hold in complicated real-world cases. The specific instance of the methodology is depicted in Fig. 3.20.

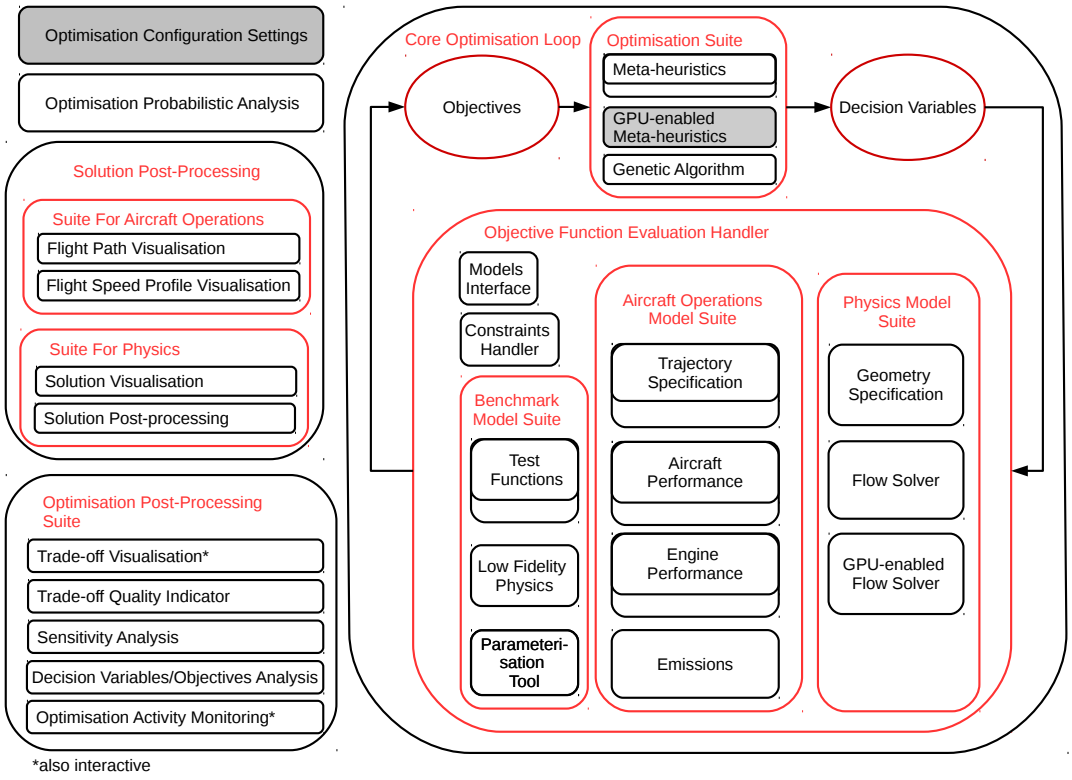



Figure 3.20: Methodology for Verifying the Scalability of GPU-MOTS2 against a Single Test

Table 3.5: GPU-MOTS2 Configuration Settings for Test Functions

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 20 |
| Call intensification move after # non-improvements | 10 |
| Reduce the search step size after # non-improvements | 35 |
| Initial step sizes (as % of variable range) | 0.005 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | # of variables / 5 |
| # of variables | 30, 120, 270, 480, 750 |
| # of objectives | 2 |
| # of objective function evaluations | # of variables * 3000 / 5 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | # of variables * 2 / 3 |

The performance gap between the two versions of MOTS2, namely MOTS2 (when running on CPU) and GPU-MOTS2 (when running on GPU), can be better demonstrated by comparing their speed-up. This is a very standard method in HPC, where the elapsed time of each application for the same scenario is measured. Here, the speed-up is expressed as the ratio of the elapsed time of MOTS2 over the elapsed time of GPU-MOTS2 when both are launched for the same number of variables. Since the core of MOTS2 has been verified and validated in the previous section, there is no point presenting EAF and hypervolume boxplots. At the end of the execution, each and every instance have accurately captured the target trade-off. The goal here is to appreciate what the performance gain is between these two versions, which is illustrated in Fig. 3.21. For values lower than 1, it is more beneficial for the application to run on the CPU, whereas for larger values GPU-MOTS2 is suggested.

It is obvious from the results shown in Fig. 3.21 that there are three cases of performance, based on the achieved speed-up, which in the end prove why CUDA is a viable alternative architecture. By carefully looking at the performance plot, it seems that there are three distinct cases: running on CPU is better, running either CPU or GPU does not make any difference and running on GPU is better. In the first case, the CPU version executes faster than the CUDA version. This is expected, since the computational load is low and an additional time overhead is required in order to communicate with the GPU. The situation is similar when

the number of variables becomes 4 times larger than the initial problem size (120 variables), but this time the performance gap is closer. Again, the use of GPUs is not justified because the workload is very intensive and the size of exchanged data between the host and the device is not sufficiently large. In the second case, when 270 variables are used, the elapsed time between the two variations is almost identical, which reveals that the performance has been matched.

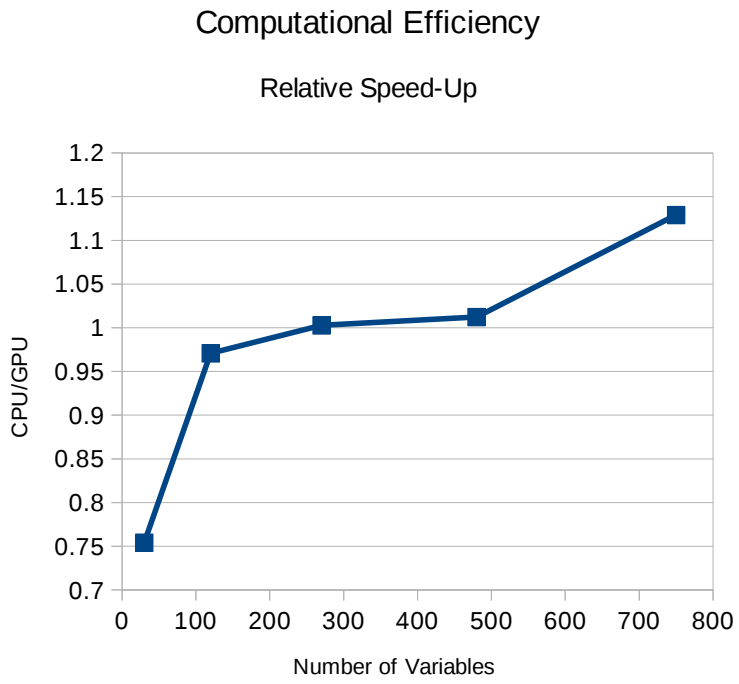



Figure 3.21: Scalability of GPU-MOTS2 when the Number of Variables Increases

The third case includes problem instances above 270 variables, where the CUDA variant starts to gradually outperform the CPU version. Initially, for 480 variables, there is only a 1% speed-up, which reveals a relatively flat region of performance gain between 270 and 480 variables. In this situation, additional information will be required to determine which infrastructure behaved better. The power consumption could be a metric [116], which gives the advantage to the CPU version because the CUDA variant requires both CPU and GPU. However, above 480 variables, there is a clear performance improvement for the CUDA version, which is 12% faster. Exploring what the performance gain is for larger problem instances will be part of future work.

It is important to appreciate that the behaviour of the optimisation process is highly related to the combination of the inherent operation of CUDA and MOTS2

configuration settings. More importantly, the problem size, expressed in the number of required parallel evaluations, dictates what technology is suitable to be employed, so as to save elapsed time. As specified in the configuration of MOTS2 in Table 3.5, on every algorithmic iteration up to 1/5 of the number of variables will need to be evaluated in parallel. So, the performance illustrated above makes perfect sense. In the current arrangement, an actual performance improvement would occur when 96 (cores) * 5 (candidate points per iteration) = 480 evaluations. Simply, when the complete number of GPU cores is utilised, the CUDA enabled version of MOTS2 becomes effective. It is important to note again that this is a transparent feature and the performance of the optimisation search will change if the GPU is replaced by a more powerful counterpart, as also discussed in subsection 3.2.1. A simple rule of thumb is to compare the computational power (product of the number of cores and the frequency per core) of CPU against the GPU. This is true under the condition that the evaluation of an objective function is an analytical formula that can be computed by a single GPU thread, otherwise more sophisticated schemes that balance evaluation time and communication time will be required, a subject to explore in the future. In the end, it is up to the user to fully appreciate and realise the potential of the application and to use the appropriate settings for the provided tool(s).

### 3.2.6 Investigating the Interactions between the GPU Hardware and Algorithm Configuration on GPU-MOTS2

In order to investigate the impact of the hardware and the configuration settings on the execution of the algorithm, scalability comparisons were performed as simulation experiments, where the overall performance is illustrated in Figs. A.1- A.4. These settings mainly follow the original settings from MOTS [63]. However, the sampling size varied, so as to investigate how this also affects the efficiency of the hardware; based on the author's experience, this is expected to affect the performance of the optimiser. Again, the methodology described in the previous subsection and illustrated in Fig. 3.6 was used. In addition, the conditions to call the diversification move, intensification move, reducing the step size, the size of the divided search space and the size of tabu memory were factored in as a function of the number of variables, for consistency purposes (e.g., it would be unrealistic to specify 4 tabu points in an optimisation problem with 1920 variables). The structure of these experiments follows:

- Two test functions were trialled (i.e., ZDT1 and ZDT2), originally described in sub-subsection 3.1.6.1, where MOTS2 delivered the best performance, as demonstrated above.

- The number of variables for each test function varied: 30, 60, 120, 240, 480, 960, 1920, 3840, 7680.
- All the simulations ran with the configuration settings listed in Table 3.6, but the sampling size varied: number of variables/5, number of variables/9, number of variables/11 and number of variables/13.
- All the above ran for 10 times, where the performance metrics (see point below) were averaged and grouped by the number of variables for each test function and each sample size.
- The performance was measured in terms of execution time and the hyper-volume indicator from reference point 20,20.

Table 3.6: GPU-MOTS2 Configuration Settings for Test Functions to Study the Interactions of GPU Hardware and Algorithm Configuration

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | # of variables * 20 / 30 |
| Call intensification move after # non-improvements | # of variables * 10 / 30 |
| Reduce the search step size after # non-improvements | # of variables * 35 / 30 |
| Initial step sizes (as % of variable range) | 0.005 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | # of variables / 5<br># of variables / 9<br># of variables / 11<br># of variables / 13 |
| # of variables | 30, 60, 120, 240, 480, 960, 1920, 3840, 7680 |
| # of objectives | 2 |
| # of objective function evaluations | 20000 |
| Divide search space into # regions | # of variables * 2 / 3 |
| Size of Tabu Memory | # of variables * 4 / 30 |

After running all the cases, the following consistent trends were discovered:

- Figs. A.1 and A.2 demonstrate that if the memory of the GPU is not fully utilised, the performance of the optimiser drops. This is shown where the elapsed time for 30 variables is larger than for 60 variables, but for ZDT1 this trend continues up to 120 variables. The elapsed time for ZDT1 does not change significantly between 120 and 960 variables. Similarly, the elapsed time required to run 20000 evaluations for ZDT2 is less than 50 seconds up to 960 variables. Thereafter, the time requirements increase fast for both test functions.

- The test function clearly affects the execution and the quality of the optimisation process.

- It is important to mention that the computational requirements of ZDT1, in terms of processing power and memory capacity, cannot be accommodated by the GPU for 7680 variables, which will be further investigated in the future.

- As expected, since the simulations ran for a fixed number of objective function evaluations, the quality of the trade-off reduces as the number of parameters increases. From 240 variables, the quality is very low, which suggests investigating alternative strategies that could improve the discovered trade-off. However, the quality of the trade-off for ZDT2 is higher than ZDT1 for 30 and 60 variables, before lower levels of performance are obtained for larger problem sizes.

- Regarding the different configuration settings of the optimiser, all the points are in very good agreement, except from the case with 30 variables and 6 points for sampling size (i.e., nVar/5) in Fig. A.1. The former suggests that sampling size does not severely affect the performance of GPU- MOTS2 in terms of time. In conjunction with the sampling size, the latter recommends that when 6 points are randomly sampled in a 30 variables problem the trialled performance is the worst and it improves as the sampling size reduces. This could be related to the memory size of the specific GPU.

- By observing Fig. A.1 and A.2, one can notice that most of the points lie in the zone between 0 and 50 seconds, where the performance of optimiser, in terms or elapsed time, slightly increases as the number of variables increases. This practically suggests the best zones to operate GPU- MOTS2. However, the first (this is more in Fig. A.1) and last point seem to move beyond that zone. For problems with computational complexity and computational resource requirements similar to ZDT1 and ZDT2, the problem size (in terms of resource requirements) is either too small or too big for the computational

model of the selected hardware. In particular, the last case is more than 3 times slower than the previous. The fact that GPU- MOTS2 could not run ZDT1 for 7680 variables is an indication of the hardware's upper bound. A deeper investigation would be required to precisely capture this performance drop.

- By comparing the shape of the two curves, ZDT1 presents a more concave and narrower shape than ZDT2. This could be interpreted that GPU- MOTS2 on the specific hardware has a more stable performance for problems that are similar to ZDT2, as for up to 960 variables, the time performance is relatively stable.

As more evidence would be required to support the last three statements, this can be the basis for further research steps, so as to find the limitation where the combination of hardware specifications and optimisation settings could impede the execution of the optimisation. This is only commented here to point out future research avenues towards a deeper understanding of the operation of the optimiser while maturing its architecture and implementation.

### 3.2.7 Discussion and Future Work

Important issues from software and hardware can have a significant impact on further developing an algorithm. In addition, heterogeneity of computational infrastructure has to be addressed and, more importantly, considered. GPU-MOTS2 attempts to give an answer to all these issues and hopefully will serve as the basis for future improvements. The ability of optimisers and methods to make use of a cluster of workstations will become even more important. Several (idle) machines already exist and have the potential to be utilised in a different way to solve hard problems.

A flexible and sustainable design of MOTS2 for the GPU architecture was presented. The new variant allows the user to optimise multi-objective high-dimensional problems within more acceptable time frames in a cost-efficient manner, a very desirable requirement when dealing with real-world applications. It was demonstrated through scalability studies that the hardware specifications of the GPU and the configuration settings of the optimiser affect the performance in terms of elapsed time and hypervolumne. Although the development is at a preliminary stage, promising results were delivered.

Employing GPUs reduced the elapsed time, which complements the operation of MOTS2, especially at high dimensionality. In the proposed implementation, the GPU acts as a co-processor that supports the CPU logic by evaluating big batches of solutions at higher rates. The evaluation procedures of MOTS2 were

modified so as to transparently couple with any GPU. The main challenges of porting were the synchronisation of data transfers and the mapping of solutions to CUDA threads by using the hierarchical memories. The ultimate goal is to use the heterogeneous computational infrastructure by combining low-level SIMT and high-level CPU approaches for higher efficiency.

The performance of high-dimensional multi-objective optimisation by using ZDT1 and ZDT2 was demonstrated on two different architectures and multiple different problem instances. The performance comparison of MOTS2 on CPU and CUDA show that by the moment all the available GPU cores are utilised, the performance of the optimisation search increases and the elapsed time decreases. The correct combination of hardware (GPU cores) and configuration settings on a problem instance indicate which version of MOTS2 is more appropriate to use and where the user should expect a performance gain. It was found that initially CPU behaves better for small problem instances, whereas the CUDA version should be preferred when the number of parallel evaluations is equal or greater than the number of the available CUDA cores. When considering only GPUs reasonable results in the optimisation of ZDT1 and ZDT2 can be obtained, otherwise alternative strategies ought to be implemented, so as to increase the effectiveness of the search.

The obtained speed-up is highly dependent on the configuration settings of the optimiser and can affect the overall performance when the application is deployed on GPUs. More precisely, the configuration settings can affect the operation of the optimiser in such a way that the size of the memories of the algorithm would not be enough to fit into the memories of the GPU, so as to evaluate the objective functions. The combination of number of variables, number of objectives and sampling size is related to the memory capacity of the GPU. This design risk could potentially raise an issue in the current implementation of MOTS2, which could be mitigated by querying the GPU capability before optimisation starts and forwarding all computational load to the CPU instead.

From the hardware perspective, the capability of the GPU (clock speed, memory size and bandwidth rate) could also cause a performance bottleneck. Sometimes, the GPU clock frequency could be half of the CPU's. This could execute a single calculation at a slower pace, but still, the number of cores on a GPU will be large enough to compensate for this. The conditions under which the difference would cause notable delays are going to be investigated in the future. The impact of the memory size has already been described above. The GPU stream that transfers data between the host and the device could be used by additional applications while carrying out an optimisation. This could cause a delay in the execution of the applications, as the hardware scheduler would attempt to serve all the applications. Similarly, through the CUDA programming interface (provided by the vendor),

this could be prevented by applying appropriate load balancing algorithms.

Future extensions will investigate more use cases and will accommodate further enhancements. First, more models will be included from the ZDT suite, and other benchmark suites, such as [230]. Second, a range of numbers of decision variables will be tested, such as [238]. Third, comparative studies will be conducted against sequential optimisation instances and against other optimisers. Fourth, larger problem instances will be tested so as to discover the upper limit of performance gain. Finally, more capability and functionality will be added in the next versions.

## 3.3 Flow Solver Simulation

A brief description of the programming model on GPUs and the implementation of a 3D LBM flow solver are presented in the following subsections. The subsequent sections illustrate the development cycle of an in-house LBM code that uses nineteen discrete densities, also known as the 3 dimensional 19 microscopic densities (D3Q19) model with the Bhatnagar Gross Krook collision operator on GPUs. The development phase of porting an earlier version of LBM onto GPUs by using CUDA programming language is presented in the following pages. The algorithmic design and challenges when developing the code are analysed, followed by discussions about scalability for the future.

### 3.3.1 Background

The advance of GPUs greatly contributes to the evolution of new tools and methods. The LBM offers an alternative method to approximate the NS equations, instead of using finite volumes that are frequently used in most CFD tools. By definition, LBM is a memory-bound algorithm and has the potential to be significantly accelerated when implemented on GPUs. The implemented model supports 3D flows by using the D3Q19 model.

The advance of recent GPU, in terms of higher processing throughput, larger memory bandwidth and faster in-between inter-communication capabilities greatly contributes to the evolution of new tools and methods. The hardware includes highly parallel, many-core and multi-threaded processors and many hierarchical memories. In addition, CUDA is a programming language that can access GPUs' hardware and has reached a level of maturity, where programming is more flexible and robust than ever before. Hence, computationally intensive applications can run more efficiently even on very large scale problems [113]. Due to the low overall cost and high computational efficiency, they suggest a satisfactory alternative computational architecture that can sufficiently cope with the requirements of scientific applications, which increasingly require even more computational power [114,115].

The LBM offers an alternative method for CFD instead of using the finite volumes numerical scheme to approximate NS equations [94,95]. More specifically, it is a class of cellular automata that can approximate the NS equations to second order with an explicit collision-streaming scheme [97]. It is particularly powerful for problem instances that involve low Mach number flow, mesoscopic flows, complex geometrical arrangements and particular boundary conditions. Further studies considerably improve LBM's potential and capabilities making it a practical tool for engineering applications [98–103]. By definition, the concept of LBM is a memory-bound algorithm, which provides a good fit for the GPU architectural arrangements. Given a computational environment with many processors, GPU-enabled LBM codes could potentially run considerably faster without compromising accuracy [104, 117–129].

The great challenge is to reformulate the algorithmic structure of the original code in a way that smoothly aligns to the GPU computing cycle of the underlying hardware. That means rearranging the execution and storing phase of data in different memory levels and on various processors. The new version is easily expandable so as to accommodate more information and capability such as new types of physics (e.g., magneto-dynamics) energy models, turbulence models, etc. Instances for 2D and 3D cases are described in [122, 123].

In general, the GPU-enabled LBM is a competitive counterpart in terms of both accuracy and execution speed. So far, significant execution speed-up and faster convergence rate have been achieved. For a relatively small computational domain, it has the potential to perform real-time simulation, as demonstrated in [247]. The aim is to present the basis of a flexible and extensible code that can be coupled with other engineering applications in a modular and transparent way. Although considerable speed-up was obtained, memory and computational optimisations will be investigated in the future.

The structure of this section follows. A brief description of the programming model on GPUs and the implementation of a 3D LBM are presented in sections 3.3.2 and 3.3.3, respectively. The subsequent sections present the development cycle of an in-house LBM code on GPUs, which is intended for environmentally-friendly applications and extends the work carried out in [45, 46, 48, 49, 64, 248]. First, the development of the code is presented in section 3.3.4. Second, the flow behaviour of a fluid in mixing phase is simulated within the geometrical arrangement of a microreactor device in section 3.3.7, so as to compare the simulated data against the experimental measurements.

### 3.3.2 Graphics Accelerators Programming Model

Originally, GPUs have been used for rendering complicated computer graphics onto a computer screen, but an alternative way is to carry out calculations (in-

stead of visualisation) by using CUDA. The developers can harness the computational power of the GPUs by using the SIMT parallel programming model, where elements of short vectors are processed by many SMs at the same time. The capability of the model to execute a single instruction on multiple register sets, multiple address and multiple flow paths differentiates SIMT from Single-Instruction Multiple-Data (SIMD), which was used in past vector computers. Simply, CUDA's SIMT model retains SIMD's efficiency with more flexibility.

When running an application there are two distinct entities that participate, each with its own memory and processing units. The first is called the host, which includes CPU and other units of the same machine, besides GPU. The second entity is the device, which is actually the GPU. Effectively, GPU is a simplistic form of host and can be accessed via CUDA. From the host's side, the device serves as a co-processor that executes many threads in parallel and depends on the host but not vice-versa. In practice, the device(s) should off-load the computationally intensive parts of an application.

The most important concepts of GPUs and CUDA will be mentioned here. CUDA is based on the standard C programming language and recently implemented C++ extensions. A detailed introduction to CUDA can be found in [113] and further development directions are included in [243, 246]. In terms of hardware, each GPU has a number of individual processing units, which operate in parallel and are called SMs, and a separate storage area, which is called global memory. Moreover, each SM has a number of stream processors - where many threads, the basic computational unit in CUDA, can run - and various levels of cache memory, called shared memory. Both threads and cached memory can run and be accessed in parallel, respectively. The number and speed of SMs and the size of storage determine GPU's performance and cost. Memories have different access speed and size. Shared memory is fast and small, whereas global memory is the opposite. Although there are other types of memory such as texture memory, constant memory and registers, these will not be considered here because their specifications significantly vary from GPU to GPU and any performance improvement would be highly dependent on the implemented LBM model. Effectively, transferring data between shared and global memory is one of the major concerns when programming in CUDA and achieves high levels of performance [243]. Regarding the computational model, sets of threads are organised in blocks and sets of blocks form the whole parallel computational potential of GPU, where each block runs on a distinct SM. The order of execution is decided by the hardware's scheduler. The developer can only prescribe the threads' instruction and the blocks' organisation in a special CUDA function called a *kernel*, which is executed on a CUDA block. During application launch, the user has to specify as many threads as possible; this allows the GPU's execution scheduler to optimise the execution and access.

Therefore, it is of paramount importance to properly design the data structures and to organise the access and layout of data in a CUDA-friendly way so as to achieve shorter elapsed times.

### 3.3.3 Lattice Boltzmann Method for D3Q19

LBM has been described in [94–96], where the equations are derived, too. Briefly, the Boltzmann equation, which describes the distribution of molecules of a physical domain, is discretised by lattice nodes. These nodes are uniformly spaced and can represent obstacles and free space. At each node a set of microscopic densities is defined, which are used for propagating fluid molecules. The implemented D3Q19 model is depicted in Fig. 3.22. The flow simulated here is incompressible at low Mach number. For completeness, the final form of the equations used follows:

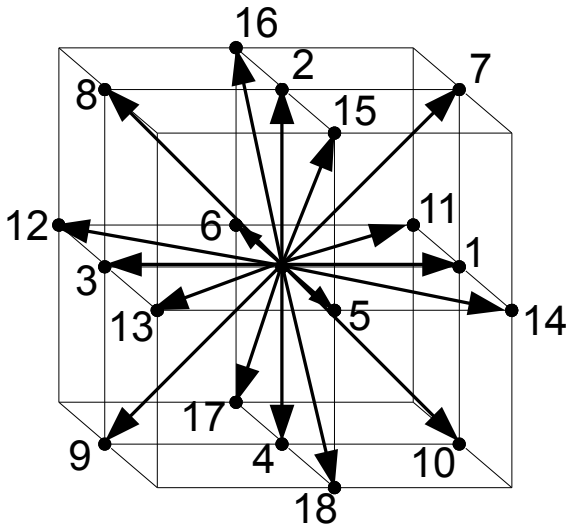



Figure 3.22: 3 Dimensional 19 Microscopic Densities - D3Q19 Model

$$f_i(\overrightarrow{x}+\delta t\cdot e_i, t+\delta t) - f_i(\overrightarrow{x}, t) = \frac{1}{\tau}\left[f_i^{eq}(\overrightarrow{x}, t) - f_i(\overrightarrow{x}, t)\right], \quad i = 0, 1, \ldots, 18 \tag{3.1}$$

$$f_i^{eq} = w_i\rho\left[1 + 3(e_i\cdot u) + \frac{9}{2}(e_i\cdot u)^2 - \frac{3}{2}(u\cdot u)\right] \tag{3.2}$$

$$w_0 = \frac{1}{3}; \quad w_i = \frac{1}{18}, \quad i = 1, 2, \ldots, 6; \quad w_i = \frac{1}{36}, \quad i = 7, 8, \ldots, 18 \tag{3.3}$$

where $f_i$ represents the distribution of particles of physical domain for each microscopic density $i$. Similarly, $f_i^{eq}$ describes the same quantity for the equilibrium of particles, $x$, $t$ denote the spatial and temporal index for each domain respectively, $w_i$ are the corresponding weights for each microscopic density $i$, $\tau$ denotes the relaxation time and $e_i$ represents the velocity along each of the directions of the microscopic density $i$ shown in Fig. 3.22.

The left part of (3.1) denotes the streaming phase of LBM, where particles advance to their nearest neighbours along their velocity direction. The right part of (3.1) represents the collision phase. Here, the distribution of particles at each node is described by the Bhatnagar Gross Krook collision operator [95, 249].

The macroscopic fluid quantities of density ($\rho$) and velocity ($u$), also called local fluid velocity, are obtained by:

$$\rho = \sum_{i=0}^{18} f_i \tag{3.4}$$

$$u = \frac{1}{\rho} \sum_{i=0}^{18} f_i e_i \tag{3.5}$$

The fluid viscosity is controlled via the relaxation time $\tau$:

$$\nu = \left[ \frac{(2\tau - 1)}{6} \right] e^2 \delta t \tag{3.6}$$

$\Delta x / \Delta t$. Here, the time step $\Delta t$ and lattice spacing $\Delta x, \Delta y, \Delta z$ are set to one. Hence:

$$e_0 = 0; \quad |e_i| = 1, \quad i = 1, 2, \ldots, 6; \quad |e_i| = \sqrt{2}, \quad i = 7, 8, \ldots, 18 \tag{3.7}$$

This simplifies further (3.6), where the last two terms equal one. Obviously, $\tau$ has to be larger than 0.5 units.

The calculation of (3.1) for the whole computational domain is resolved in two stages: collision and streaming. First, the collision phase takes place as described by:

$$f_i^{new} = f_i(\overrightarrow{x}, t) - \frac{\Delta t}{\tau} \left[ f_i^{eq}(\overrightarrow{x}, t) - f_i(\overrightarrow{x}, t) \right], \quad i = 0, 1, \ldots, 18 \tag{3.8}$$

where $f_i^{eq}$ is obtained from (3.2). Then, the streaming phase is described by:

$$f_i(\overrightarrow{x} + \delta t \cdot e_i, t + \delta t) = f_i^{new}, \quad i = 0, 1, \ldots, 18 \tag{3.9}$$

The explicit form of (3.8) and (3.9) reveals the locality feature of the calculations and aligns with the operation of GPUs. This proves why LBM is a good fit for GPU architecture.

### 3.3.4 Porting LBM to GPU

The main challenge when porting an application onto GPUs is to identify the parts of the code that can be best parallelised by GPUs. Following the principles of HPC, there will be a fraction of the code that will always be sequential and has to be minimised. Consequently, the remaining part has to be as parallel as possible. However, not all of the parts can be fully parallelised for certain reasons, which lie out of the scope of this work. Considering the inherent operation of SIMD architectures, the operations (calculations) have to be as independent and systematic as possible, whereas the data required have to be arranged in a collective layout. This fully exploits the potential of GPUs, while independent calculations are carried out by each computational thread and data are effectively cached. An additional challenge is to hide data latencies by doing calculations at the same time, to load data in parallel streams, to reduce the amount of branching of the execution workflow (even if this means performing calculations that do not affect the final solution) and to hierarchically decompose and exchange sets of data in a way that can be effectively processed by GPU's SMs [243]. Hence, most of the effort is to re-arrange the existing code so that it can be compatible with all these requirements. Then, CUDA can transparently scale from a single machine to a cluster of GPUs, cluster of workstations and beyond.

The implementation presented here adopts the original parallel FORTRAN code, which was successfully used in earlier studies [46, 48, 64]. The first step was to reduce the code to a sequential C++ code, by using appropriate data structures and indexing, as also suggested in [250]. Then, the arrays of data were modified so as to be accessible in a linear way, which is identical with the way the memory lies onto hardware and resembles the programming-style of vector machines. This approach provides a better understanding of the actual layout of data and reveals the required data access patterns. Consequently, this allows for more portability, control and other future improvements. Following these steps, porting onto CUDA is very straightforward, simply by replacing the majority of sweeps throughout the computational domain with threads and by (re)using cached data as much as possible to take advantage of the hierarchical memories and access speed. More specifically, the code is based on an earlier parallel variation that runs on multiple

CPUs, and it has been significantly improved. The code can simulate the flow in 3D with 19 distinct densities, also known as D3Q19 (Fig. 3.22), and implements the Bhatnagar Gross Krook collision operator [249].

Scheduling data transfers is the most important issue when porting an algorithm onto SIMT model. In contrast to the conventional HPC applications that scale on many distributed CPUs, SIMT-enabled applications are just the opposite. The data should be organised in a Structure of Arrays, whereas an Array of Structures is preferred when dealing with CPU-based distributed environments. Hence, as a single development decision/strategy, this is the most important part that could determine the overall performance [250]. Due to particularities of CUDA, special care is required for memory access pattern, execution and memory layout.

On top of translating the code from FORTRAN to C++, the development of GPU- LBM was particularly challenging for the following reasons:

- As explained before, the data structures of the FORTRAN code (i.e., two multi-dimensional shared-access arrays and a few small utility arrays) had to be converted to many individual local-access single-dimension data structures, where each of them would be processed independently, in asynchronous mode, so as to accelerate the execution flow. This also requires the memory access pattern to be changed in a such way that consecutive locations are accessed by a single thread that should not coalesce with other threads.

- In the original FORTRAN code, a few subroutines covered all the functions of LBM. The part of the code that runs on GPU has to be organised in kernels. Here, the same functionality (e.g. for loops) was restructured, so as to access the aforementioned data structures time-efficiently. So, the computational load has to split into multiple CUDA kernels that are allocated on SMs, whose computational speed, memory capacity and number significantly varies from one GPU to other. If this was not resolved, then the code could only run on limited models of GPUs (i.e., with certain specifications).

- Duplicates of the data structures to hold the data have to live in the host and the device, so as to be accessible by the CPU and the GPU, respectively. However, this requires the data structures to be synchronised at regular intervals. In order to save in synchronisation time, the communication channels of GPU have to be used and the kernels for memory transfer are staggered.

- The original FORTRAN code had a number of execution branches that cannot be used in the CUDA implementation, as branching severely penalises the performance. So, additional kernels were developed, so as to isolate the branching in very small sections of the code. These kernels are staggered with the larger kernels, so as to hide the latency.

- Debugging a multi-threading application (with many hundreds of threads, where each thread updates a particular area in the computational domain) requires additional utility kernels and data structures to be created. This is the only piece of code, which does not participate in the final version, where branching takes place, so as to check for data consistency. Although this code is not used in the actual simulation, it considerably accelerated the code development phase, as the generated data were continuously checked from one iteration to the other, without interrupting other tests. At the moment of writing this document, the CUDA toolkit was not mature enough to support this.

- A series of unit tests, integration tests, system tests and stress tests [207] were implemented, so as to continuously check the consistency/quality of the code when incrementally adding new features. Although these do not contribute to the computational performance, they are invaluable in terms of development effort, as the number of GPU threads and the volume of processed data can only be handled in an automated way. Normally, in a production environment, more than 50% of the overall development effort is put against testing [251, 252]; however, in this research, testing took approximately 33% of the overall time.

- All the above kernels and data structures were developed to be as generic as possible from a centralised point in the code, whereas the FORTRAN code could only run at a certain precision mode and on a very specific parallel arrangement of 4 parallel threads. Here, the GPU-LBM code can support many precision levels and can be deployed on many combinations of GPU and CUDA driver. In fact, even during the code development phase, the actual code development and the tests/execution took place on different machines, with different underlying hardware. Whenever possible, the most primitive libraries of CUDA were used, so as to ensure as much backwards compatibility as possible with many configurations of GPU and CUDA driver.

- Based on the author's experience with engineering applications, the architecture of the code used several design patterns, so as to easily extend and link to other applications, whereas the original FORTRAN code required significant development effort to add new features (for integration purposes). This enables the following behaviour:

  - The code has been successfully tested on multiple operating systems (i.e., unix-based and Windows-based) and various GPUs.

- The solution of the LBM code can be exported on proprietary (TECPLOT dat) and open (VTK) file formats, so as to be post-processed by a wider number of stakeholders and on other post-processing tools.
- The code has been tested on a single CPU and GPUs with CUDA compute capability 1.3, 2.0, and 2.1. In theory, it should be possible to run on higher compute capability, but this was not tried.
- The code was designed to be configured externally in terms of the number of threads, by using external configuration files, so as adjust the desired maximum execution speed, as required.
- Certain parameters of the application were decoupled from the algorithm and can be edited externally, so as enable more parametric studies. By using the same geometrical arrangement (i.e., a baffled microreactor), the size of the microreactor, the layout and location of the baffle can be specified externally. It is also possible to provide a much more complicated geometry by directly providing a new geometry file. In both of the aforementioned cases, the flow conditions can change and run without compiling the code again. All these are expected to be used by another system or users (e.g., an optimiser, a Monte Carlo algorithm, a testing server, graduate researchers, etc.) to run a wider-in-scope application with minimal effort.
- Several other parameters in the code can be adjusted from a central point, for greater flexibility. More importantly, the arithmetic precision mode, the execution mode (i.e. sequential or parallel) and the boundary conditions can easily change.

All the above features were tested individually.

#### 3.3.4.1 Algorithm

The selection of data structures is based on the formulae of (3.8) and (3.9), where the second shows that LBM is of first order in time and second order in space. Hence, two data structures are required to implement D3Q19 of LBM. Each structure contains 19 individual arrays, one for each density. Moreover, an array of obstacles is required for qualifying the type of each node, as described in subsection 3.3.3. This allows one to alter the shape of the geometry of interest within the computational domain, while the size of memories is kept constant. The remaining quantities will remain constant. Because host and device have their own memory space, the aforementioned structures need to be allocated on both ends.

The most frequently called functions of the algorithm are check density, streaming, bounceback, relaxation and convective boundary conditions. The first one is

basically used for calculating the convergence. It is a two-step reduce-sum over each density. Then, the results for all densities are averaged over the total number of nodes that were used. In general, checks do not take place very frequently so as to speed up the application. Usually, the frequency of carrying out density checks is set to a relatively small value if there is much uncertainty about the shape of the target geometry.

The layout of Structure of Arrays significantly speeds up streaming and bounce-back because of the continuous access of data both for reading and writing purposes. In streaming, fluid densities are propagated to their next neighbour nodes. Effectively, data are transferred from non-occupied nodes along the lattice connection lines to their next neighbours. Every streaming is applied to each density separately, which makes it ideal for CUDA streams [243]. Similarly, bounceback makes use of the additionally allocated array for obstacles for each density, separately. Essentially, fluid densities are inverted if any obstacle is present. Thereafter, fluid densities are sent back to the node where they were located before the last propagation step but with opposite velocity vector.

Relative to the other function calls, relaxation is more data intensive and is linked to convective boundary conditions. It is a one-step density relaxation process, where all 19 microscopic densities are used at the same time for each node, before moving to the next. Then data are prepared for the next step. Relaxation also requires knowing the obstacle nodes. During convective boundary conditions, macroscopic velocities are computed at the end of the domain. The velocities at the end of the computational domain from the previous iteration are combined with velocities of the current iteration at the penultimate and final end of the computation. After this conversion, the microscopic densities are distributed to their domain and all the data are ready for the next iteration.

The code iterates until it reaches convergence. Here, the average density ($avgD$) is used to decide when the flow solution has converged and is calculated as:

$$avgD(f) = \frac{\sum_{i=0}^{18}\left(\sum_{x=0}^{LX}\left(\sum_{y=0}^{LY}\left(\sum_{z=0}^{LZ} f_i\left(x,y,z\right)\right)\right)\right)}{18 \cdot LX \cdot LY \cdot LZ} \tag{3.10}$$

where $x$, $y$, $z$ denote the spatial indices, $i$ is the density index, $LX$, $LY$, $LZ$ are the total number of nodes in the $x$, $y$, $z$ directions, respectively, and $f_i$ represents the distribution of densities. The convergence is calculated between two consecutive iterations as follows:

$$\frac{\left|avgD\left(f_{current}\right) - avgD\left(f_{previous}\right)\right|}{\left|avgD\left(f_{current}\right)\right|} \leq 10^{-4} \tag{3.11}$$

Since this function actually performs a massive averaging of all the nodes of the domain, it seems that it can be performed less frequently in order to speed up the

overall execution. So, this check can take place after a certain period, which is expressed in a number of iterations and is specified by the user before running the application.

#### 3.3.4.2 Memory Access

The aforementioned data structures were selected so as to fit with CUDA's computational model and operation. In turn, this means being able to access data in a coalesced and continuous manner, as strongly suggested in [243]. Simply, both shared and global memory have to be read and written in this way, because this achieves higher parallelisation and increases computational efficiency.

Following the fundamentals of CUDA presented in subsection 3.3.2, each node of the computational domain is uniquely accessed by a single thread. As expected, this continuous access is grouped in blocks so that each block takes over an independent fraction of the domain. A large number of threads is spawned, enough to cover the whole domain. The global thread identification number is used as an index for accessing the arrays described above. In fact, each thread is matched with only one memory location with the following formulae:

$$gid = blockId.x \cdot blockDim.x + threadId.x \tag{3.12}$$

$$\begin{aligned} did = \lfloor gid/\left(LY \cdot LX\right) \rfloor + \\ \lfloor \left(gid - \lfloor gid/\left(LY \cdot LX\right) \rfloor\right)/LX \rfloor + \\ \left(gid - \lfloor gid/\left(LY \cdot LX\right) \rfloor\right) \mod LX \end{aligned} \tag{3.13}$$

where $blockId.x$, $threadId.x$, $blockDim.x$ are CUDA's reserved variables for identifying blocks, threads and the total number of blocks, respectively. The unique global identification number of thread and data are called $gid$ and $did$, respectively. This approach reduces the range of generated threads to the actual computational domain, secures a 1-1 mapping and preserves the linear access fashion.

Since the number of threads has to be equal to or larger than the domain size, a few remain idle. This is the only part of the code where branching appears. Nevertheless, it can be considered as a minimal overhead, which is negligible in practice. This will eventually produce a very small number of idle threads in a single block that will not take place in the computations. However, this approach guarantees that the number of threads will be kept to a minimum, which will, in turn, increase the computational efficiency. For ease of use, the C++ *#define* directives can bridge the gap between the programming model and human interpretation.

The uniform handling of the computational domain has another implication: a fraction of the total number of threads will be calculating and/or carrying zeroes. However, they will not have any implication for the final results. This was

done for two reasons. First, for simplicity, as fewer checks would be required, thus maximising the amount of computational power required to obtain the target information. It is also suggested by NVIDIA to carry calculations in bulk even by repeating some of them in order to reduce branching because threads run in parallel batches. However, the most important reason is to keep the whole domain *active*. Doing that serves a very special reason: different shapes of geometries can be used without the need to modify the code (or any of the settings). This feature is particularly useful in engineering cases, where LBM could be a module of a larger system that needs to communicate transparently with the rest of its environment.

#### 3.3.4.3 Execution

The most important factor when porting a code onto CUDA, besides the underlying hardware and the accompanying drivers, is to resolve the number of blocks and threads spawned that will actually perform the calculations. This will determine the communication between adjacent memories and other access patterns. As mentioned in [243, 246], the challenge is to design the data structures in a way that naturally fit with the hierarchical memories and the computational model of CUDA, while (re)utilising data as much as possible. Of course, there is a trade-off between full saturation of the SM and under-utilisation of computing cycles, because of execution flow branching and data transfers.

The elapsed time of a simulation heavily depends on the number of threads and blocks employed. On the one hand, the total number of threads will be known from the size of the lattice nodes. However, the ratio between threads and blocks is also important because it is related to the capability of GPUs' SMs, which has a finite and relatively small memory, where the data can be cached before being transferred elsewhere. So, there is an upper limit to the amount of data that can be processed effectively before the performance drops due to caching effects. On the other hand, in CUDA, threads are processed in groups of 32 (in the most recent GPUs), which is called a warp. Then, a multiple of warps can be part of a block. Therefore, in the current implementation, the user has to specify the number of warps (as a multiple of 32 threads) in a block before launching a simulation so as to satisfy the following criterion:

$$threads \cdot blocks \geq LX \cdot LY \cdot LZ \tag{3.14}$$

Then, the number of blocks is automatically calculated as the ceiling of the equality. However, if (3.14) is not true or the number of threads and blocks reaches CUDA's limit, then the simulation does not run.

### 3.3.5 Interfacing and Configuration Settings

It was decided to make the application as generic as possible so as to link to other tools in order to perform general engineering studies. The easiest way to achieve this is to provide the user and another system a number of different handles for interfacing and configuration purposes. This is a very straightforward approach, where all the files for input to the application and for configuring its behaviour are simple external files in a human readable format. Hence, editing and creating any of these files is very easy and can be automated. In addition, this requires fewer changes to the actual application, which could be considered as another black box either for a stand-alone project or as a part of a larger system.

The shape of the geometry of interest is provided via a combination of two external files, which are read once by the host at the beginning of the simulation and initialise all the other memories. One file describes the number of points (i.e., lattice units (*lu*s)) on each axis and another is the actual representation of the computational domain, which has to agree in dimension with the previous file. The target geometry is simply represented within an assembly of points in a 3D coordinate system. The external file contains as many binary values as the points of the 3D space, where 0 denotes an unoccupied space, whereas 1 represents an obstacle. In fact, this can be considered as a simple stream file that describes the whole computational domain. Therefore, it is very easy to change the shape of the geometry.

Then, all the data are transferred to GPU's memories and the iterations start until the convergence criteria are met. At the end, the results are sent from the device to the host. This is a constant overhead that is not worth parallelising and is used for interfacing purposes.

Because the code is intended to be used on engineering applications, it is advantageous to be able to carry out calculations either in single precision or double precision or mixed [9]. Therefore, all the functions of the code can operate in either a single or a double precision mode. This will be particularly useful on multi-fidelity studies. As expected, double precision will be more computationally intensive than single precision.

There are two external files classified as configuration files. These control the behaviour of the application and implicitly determine the overall performance, such as the execution time. More specifically, in the first file, the quantities that can be specified are the number of algorithmic iterations, the frequency of checking for convergence, flow features, output name, the number of computational threads and special attributes of the hard-coded geometry to simulate, which could be deactivated. The size of the computational domain is specified on a separate file.

[9] One type of precision on the one hand side and another type of precision on the other hand side of a computational instruction line.

All these options are provided because they have a computational impact on the execution of LBM applications, while, by definition, they affect certain boundary conditions and, consequently, change the computational workload.

In addition, a number of files related to the target project are provided to the application and are generated by that. In the presence of an external file that contains the full definition of geometry, which agrees with the dimensions of the computational domain, as specified above, the previous configuration settings for the hard-coded geometry are overridden. Hence, the externally provided geometry, which is expressed as a sequence of obstacle nodes in the array of obstacles in LBM, is used instead. At certain time intervals, the application performs a check in the computational domain for convergence, which is registered in an external file for monitoring purposes. Two prescribed flow features, vorticity magnitude and difference of total pressure (between two locations in the dataset), are automatically calculated at the end of the simulation and are stored on a separate file. These characterise the flow in the specified geometry and can be considered as flow quality indicators to measure the performance of either the flow or the geometry or both. Finally, the flow domain expressed in macroscopic scale units is exported at the end of the simulation, too.

### 3.3.6 Architecture of the Flow Solver

This section aims to present to the reader (and the future user of the code) the static and dynamic perspective of the flow solver simulation, where most of the development effort was spent. Here, the principles of object-oriented design were employed to conceptually create the architecture of the software package. Eventually, it will be a ready-to-use implementation, where in-depth knowledge and expertise in HPC are not required. In addition, it helps the interested reader to individually compare certain parts of the code with other competitive implementation(s) and to apply other adjustment(s). The structure of the software package is such that it can easily and transparently interface with a variety of other computational system(s). The modular architecture provides more flexibility and allows more modification and enhancements to take place with minimal effort. In the author's experience, this is the most important feature of the software development phase because potential risks can be revealed and devising a recovery strategy is more straightforward. To the best of the author's knowledge, this is the first time that such a software architecture is released and is expected to evolve into a distinct design pattern for a class of problems. The presented design is expected to be reused and to save precious time and resources when the Flow Solver from the previous section is employed.

The software package provides as much control and flexibility as possible. The majority of the architectural details are language-independent, which makes the

design pattern more practical. This is the top-level perspective of the code and provides all the handles and mechanisms to interface with other systems. However, whenever higher levels of computational efficiency are required, changing the lowest levels of the architecture will affect the overall performance. This is one of the merits of the object-oriented design, where the implementation details related to the computational performance are encapsulated very deep in the structure of the code. Although these details are language-dependent, they represent a very small fraction of the code. This practically means that they can be easily replaced without affecting the higher levels and interrupting the communication with other internal or external systems.

Initially, a list of the most important participating modules is illustrated along with their attributes and operations in Fig. 3.23. This can be considered as the top-level map of the system that performs the simulation of fluid flow by using LBM, where individual entities that serve a single purpose are shown. Each of the entities is an abstract class, which promotes code re-usability instead of a collection of individual routines. The interaction(s) among the described modules is shown in Fig. 3.24. This reflects how the system behaves and demonstrates how the participating modules communicate.

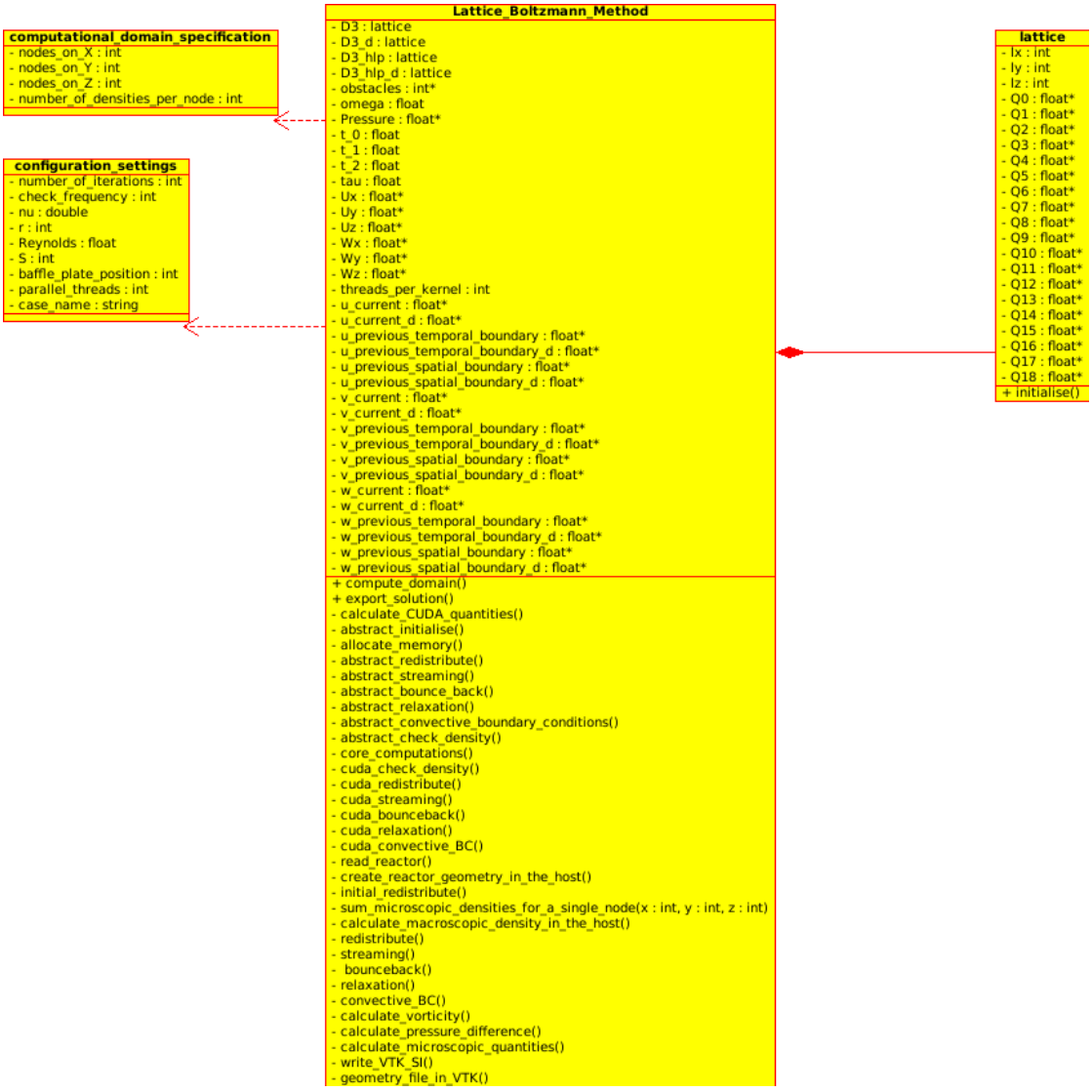



Figure 3.23: GPU-LBM Class Diagram

Starting from the simplest to analyse classes, the solver needs to know certain attributes about the geometry of interest. These attributes are grouped in two distinct objects. The first object, called computational domain specification, contains information about the arrangement of nodes and densities, as they are defined in LBM. The second object, called configuration settings, lists the settings related to the problem description and the low-level settings of the computational method. These were joined in a single object because (for a fixed geometry in terms of a number of nodes and densities) their combination determines the wall-clock time of the simulation. Since, the application aims to save elapsed time, this

is considered a sensible approach.

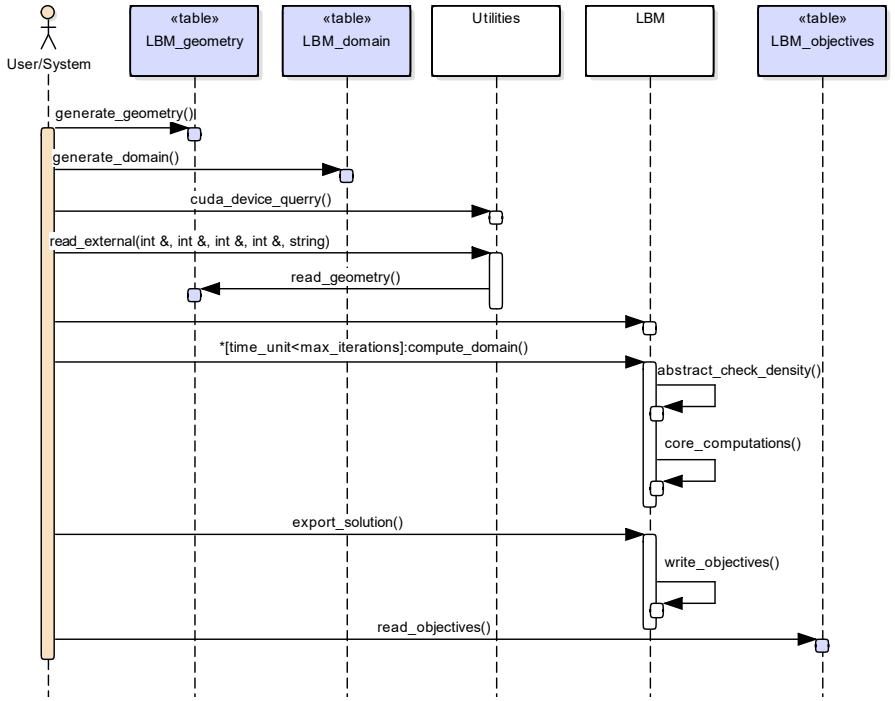



Figure 3.24: GPU-LBM Sequence Diagram

### 3.3.7 Simulating the Fluid Flow in the Microreactor by using GPU-LBM

For demonstration purposes, the developed flow solver is used to simulate the flow in the microreactor. By running applications on GPUs, the flow can be resolved in significantly shorter time-intervals and in conjunction with the features and architecture described before, the overall application is considerably accelerated. By upgrading the GPU hardware, the methodology can transparently scale to many cores, whereas expanding on more CPU-based computational arrangements has higher risk and requires more effort; in fact, computational arrangements based on CPUs require much more space and economic resources and so does all the supporting infrastructure. Compared to standard CFD applications, fewer processes and tools are required for simulating the fluid flow, which makes the method simpler to employ and easier to understand and to link with others. Because of the microscopic representation of the domain, more memory is required and, consequently, more operations have to take place during a single iteration. Although it is mainly

a memory-bounded application, the code gains more when it is deployed on parallel hardware.

In this research, LBM is employed to simulate the flow of a micro-mixing device for use in an environmentally-friendly context, where the dimensions of the target device and the areas of interest are so small that using the model of NS equations to describe the flow is not practical. Further attributes of the computational domain and solver settings are provided in other external files. This is used to specify geometrical features of the device and to set flow characteristics. Other settings affect the operation of the flow solver, where the number of threads necessary to run the simulation is specified. The specifications of the employed GPU are listed in Table 3.7. The number of spawned threads per block on the GPU is a very important factor in order to achieve high computational efficiency because of the occupancy of SMs with threads and the available memory to store the results. Here, the flow solver is instructed to use 512 threads when it simulates the flow. This is the fastest setting provided by the current GPU.

| **Tesla M2070** | |
|---|---|
| CUDA Driver Version / Runtime Version | 4.0 / 4.0 |
| CUDA Capability | 2.0 |
| Total Amount of Global Memory (MB) | 5375 |
| CUDA Cores | 448 |
| Stream Processors Rate (MHz) | 1150 |
| Memory Clock Rate (MHz) | 1566 |
| Power Consumption (W) | 225 |

Table 3.7: GPU Hardware Specifications to Simulate the Microreactor

The initial and boundary conditions will attempt to replicate the experimental conditions of [45], as follows. This was specified so as to demonstrate that the geometrical arrangements of the microreactor could be accurately simulated and represent real-world behaviour. It is important to mention the following points for clarity:

- At the beginning, both of the chambers, before the baffle plate, are initialised to constant and different density values, which are expressed as a function of Reynolds number ($Re$). This assumes that this flow feature will remain constant through the operation of the device and does not degrade over a period of time or other material properties.

- The initial conditions are part of the code and all the unoccupied nodes of the computational domain are assigned a certain value. This assumes that it is possible to achieve this type of flow.

- The walls of the domain are considered obstacles and implement the bounce-back operator. It is assumed that the obstacles will not move or deform throughout the simulation.

- At the outlet of the domain, a zero streamwise (x component) velocity gradient was applied:

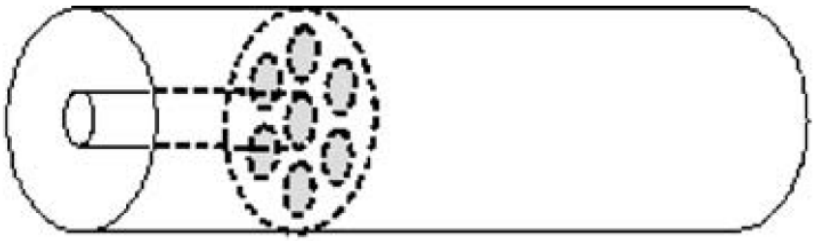
$$\frac{\partial u}{\partial x} = 0 \tag{3.15}$$

  This boundary condition is applicable when the flow is fully developed, as described in [253]. In LBM, the microscopic domain is converted on the fly to the macroscopic domain, which is straightforward to implement and had also been applied to the original reference [46] (where 3 different geometrical concepts were investigated to study the mixing capability of a microreactor). In the performed simulation, the velocity profile results indicate that the flow is fully developed (see Fig. 3.30). By inspecting Fig. 3.31, the velocity gradient appears to be constant after $x = 180$ $lu$.

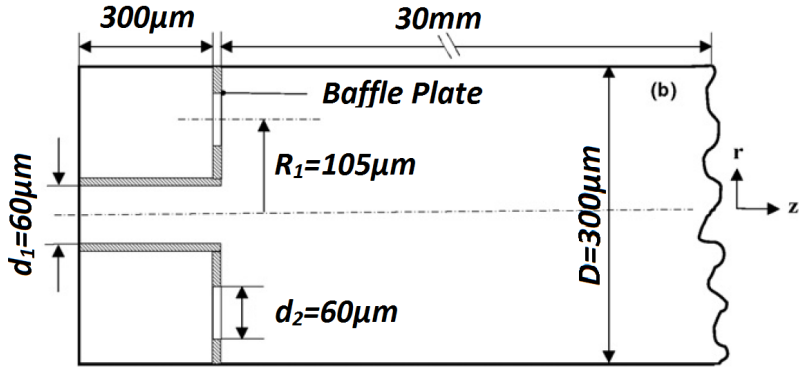
The above points represent a minimal set of specifications that imply that the flow features and the device will remain as described, which could potentially affect the costs of a manufacturing process to make sure that these will always be met. By definition, in LBM, the complexity of boundary conditions might affect the execution time because of the data access pattern. This happens because the code is fitted to cope with the GPU's operation; boundary conditions might not take into consideration the data structure layout, which is optimal for the core operation of GPU-LBM. In fact, the required access is in a domain of 3D coordinates, whereas the data structures of the LBM model are expressed in 19 distinct densities. So, the requested 3D information has to be translated into access patterns of sparsely located data in memory, which will certainly increase latency because of the sequential access. Hence, this is the only part when running a case where the user needs to carefully select the orientation of the geometry of interest in combination with the required boundary conditions, because a different orientation is expressed differently in the linear hardware memory.

The geometry of interest is a baffled microreactor, the seventh experiment from [45], as depicted in Fig. 3.25, and was selected because it demonstrates the recirculation of the flow downstream of the baffle plate. The dimensions of the lattice are 680 $lu$, 73 $lu$, 73 $lu$ (additional nodes are used for the wall) in the $x$, $y$ and $z$ directions, respectively. Following the geometrical parameterisation scheme of the baffle plate, depicted in Fig. 3.26 where the location and size of the outer holes are specified, the hole radius in the baffle ($r$) and spacing in the baffle ($s$) are specified to 7 and 24.5 $lu$, respectively. As described above, the flow speed is inferred from the $Re$=100. Although it takes slightly more effort to configure a case, this is going to be particularly useful in the next chapter, in the optimisation.

a)



b)




c)

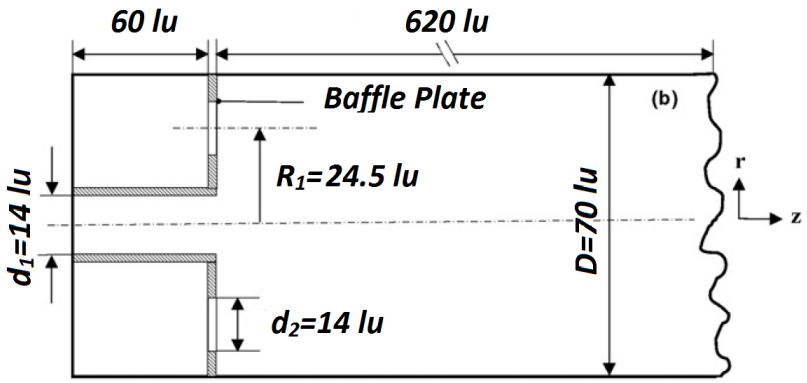



Figure 3.25: a) Perspective View of the Microreactor Model, b) Profile View of the Microreactor Model in Physical Units (not to scale) [45] and c) Profile View of the Microreactor Model in Lattice Units (not to scale) [46]

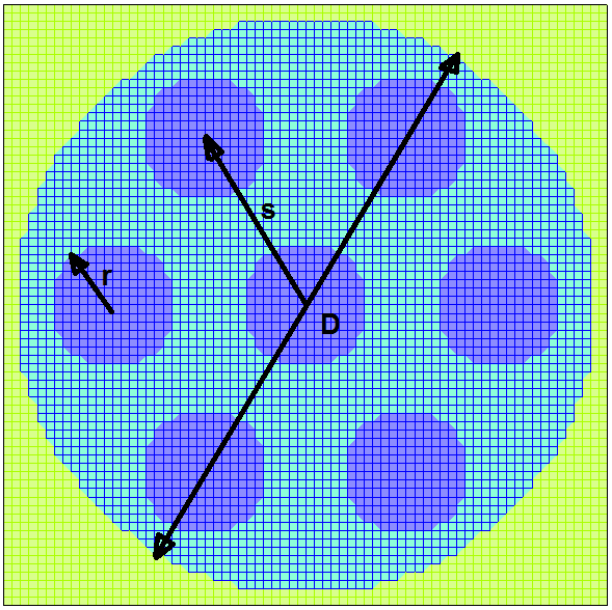



Figure 3.26: View of the Baffle Plate in Lattice Units [64]

The effectiveness of the developed flow solver is demonstrated qualitatively, because of the lack of the experimental dataset from [45]. Certain flow features, which were interpreted from the experimental visualisation, are going to be depicted below by using the simulated results. First, a simple (side-by-side) comparison in flow pattern between the experimental measurements and the simulated dataset is shown in Fig. 3.27. In the slice ($x$-$z$) of data from the middle of the $y$-axis, an eddy is formed downstream of the baffle plate between the outer tube and the inner tube, because the velocity of the outer flow is greater, as was set up in the experiment. The top part of the figure is the vector field of the seventh experiment, whereas the bottom part is a visualisation technique that shows the streamlines of the flows, which is similar to the technique (i.e., micro-particle image velocimetry) used in the experiments to create the aforementioned vector field. Although the flows between simulation and experimental measurements are not identical, the flow patterns are similar. This discrepancy was expected, as the simulation attempts to approach the natural behaviour that was experimentally measured. The precision of the simulation could be enhanced by employing a different numerical scheme (e.g., [254]), but additional computational resources are required. Here, the attained precision and accuracy are adequate for prototyping purposes only.

The same feature is also shown in Figs. 3.28 and 3.29 in the form of a vector field. Downstream of the baffle plate (at $x$=59), the difference in flow speed between the outer holes and the middle of the baffle forms an eddy, which will

actually contribute to the mixing capability of the device. This can be observed in the areas between 80 $lu$ and 90 $lu$ in $x$, or between 10 $lu$ and 20 $lu$ in $y$. In Fig. 3.28 all the vectors are identical, so as to clearly present the eddies. However, in Fig. 3.29, the size of the vectors represents their actual magnitude, which is used to represent the intensity of the vector field in the outer holes.

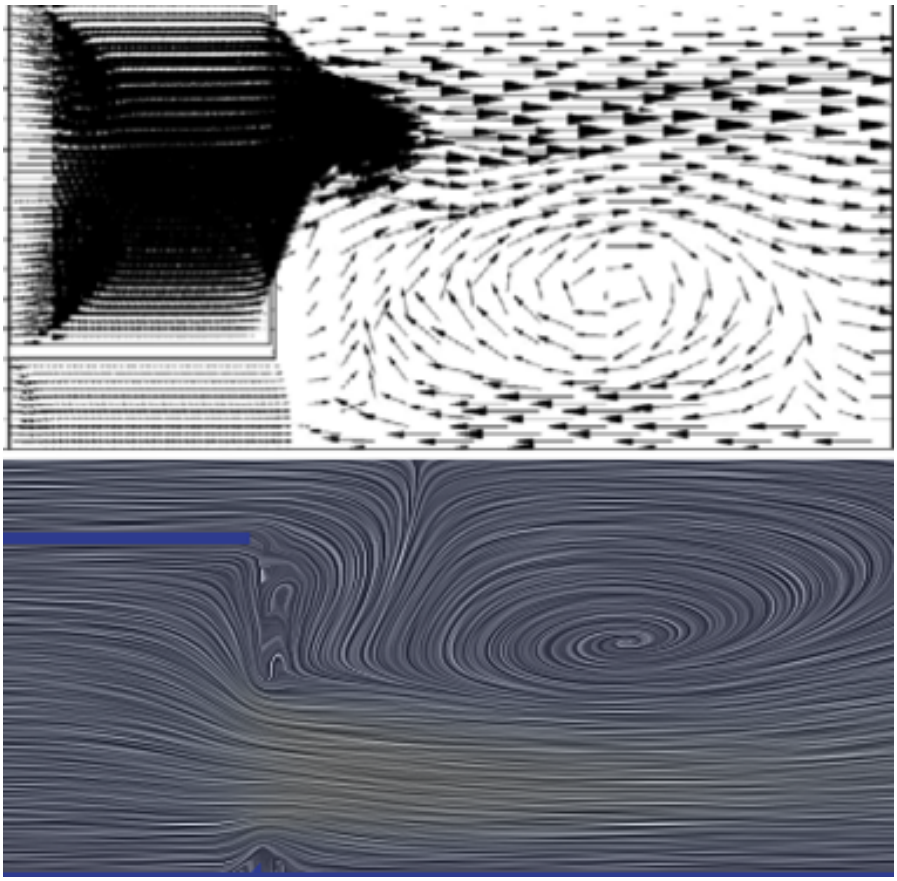


Figure 3.27: The Flow Pattern between Experimental Measurements and Simulation are Qualitatively Compared

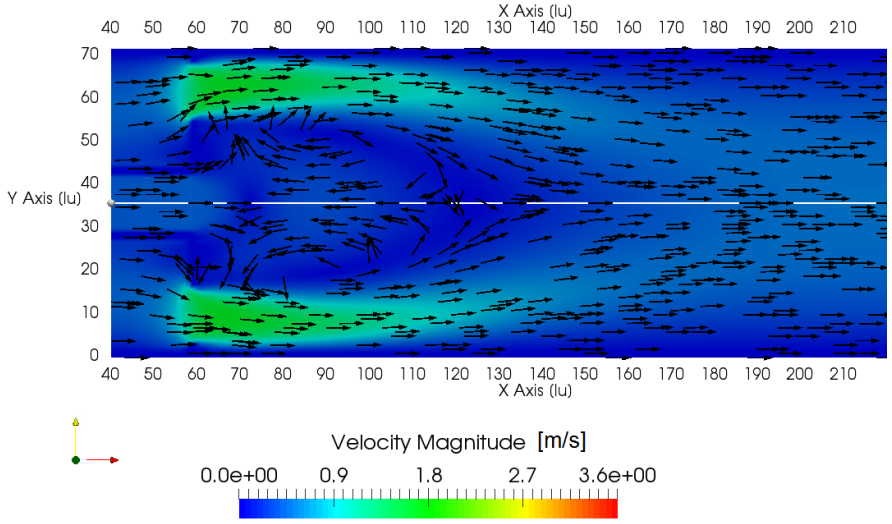



Figure 3.28: Vector Field

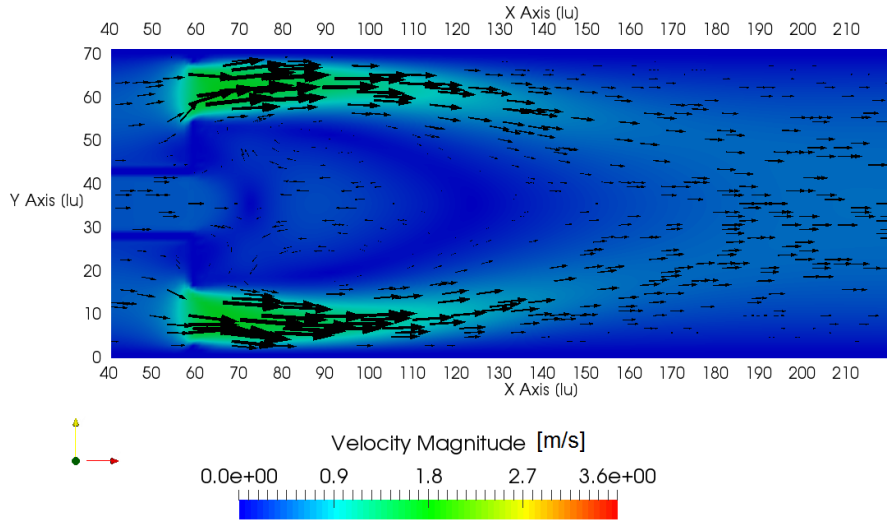



Figure 3.29: Vector Field with Scaled Vectors

Two more features are presented in Figs. 3.30 and 3.31. First, in Fig. 3.30, by taking snapshots of the x component of the velocity before the baffle plate, after

the baffle plate and far from the baffle plate, it can be seen that the profiles of the x component of the velocity are very different. At a spatial index on the X-axis ($x$)=40 $lu$ the flow profile was expected, because it is part of the initial conditions of the flow. At $x$=67 $lu$, the flow speed is higher in the outer holes, because the constant flow rate (from the initial conditions) passes through a smaller cross-section. The decrease in flow speed in the inner hole is caused by the recirculation of the flow explained above, further analysed below. Far from the baffle plate, the flow has been fully developed and follows the theory of flow in pipes, where the velocity profile is parabolic. This is higher than the velocity at $x$=40, because of the aforementioned acceleration of flow in the outer holes.

Second, by using the x component of the velocity in a line in the middle of the microreactor (along a spatial index on the Y-axis ($y$)=36 $lu$), one can see in Fig. 3.31 that the velocity significantly drops before the baffle plate and will increase smoothly. Following the recirculation of flow, described above, the flow from the outer holes creates a secondary slower flow that counters the flow from the inner hole. This is obvious in Fig. 3.28, where between 40 $lu$ and 100 $lu$ in $x$, and along $y$=36 $lu$, the vectors seem to conflict with each other. However, because of the strong flow of the outer holes, flow starts to accelerate and after $x$=140 $lu$ the flow speed has exceeded the initial flow speed.

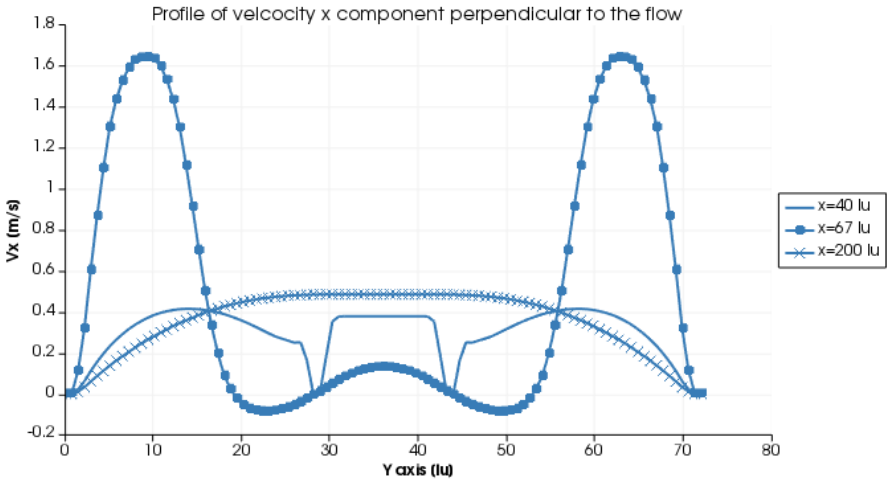



Figure 3.30: Velocity Profile of the Microreactor, perpendicular to the Flow

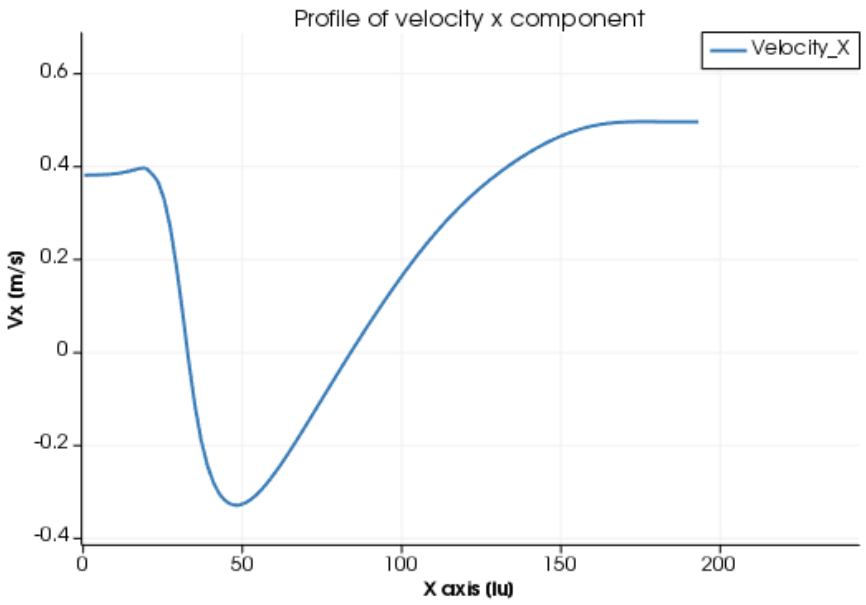



Figure 3.31: Velocity Profile along the Microreactor

Simulating a single configuration via GPU-LBM takes on average 145 seconds, whereas the code of [46] that was used in [64] requires approximately 3,000 seconds. Hence, this is approximately 20 times faster. The most common metric to measure the computational efficiency of the code is *Mega Lattice Updates Per Second*, as follows:

$$Mega\ Lattice\ Updates\ Per\ Second = \frac{LX \cdot LY \cdot LZ \cdot 1000}{elapsed\ time} \tag{3.16}$$

This version of GPU-LBM delivers slightly less than 25 Mega Lattice Updates Per Second. However, this figure is about an order or magnitude less than the performance gains reported in [118], where an inferior GPU was used. Although the shape of the computational domain (in terms of number of obstacles) affects the computational performance, there is still considerable room for improvement. More approaches to improve the computational efficiency are explained in sub-subsection 5.5.3.2.

## 3.4 Summary and Linking with next Chapter

The contribution of this research, in terms of developed tools, follows:

1. A native MOO algorithm, called MOTS2, has been developed from scratch, based on [63, 198] with additional features (subsection 3.1.5). MOTS2 can

handle continuous variables and is intended for real-world applications with many (more than two) conflicting objectives. The selected data structures, algorithm and one extra strategy (called kick) have been discussed and analysed, which are the main competitive features compared to other optimisers.

2. MOTS2 has been verified on well-known test-functions and has been validated on an airfoil shape optimisation problem, so as to demonstrate its suitability for real-world problems.

3. Compared to a state-of-the-art optimiser, NSGAMO, MOTS2 can handle the multi-modality of the trade-off of ZDT4, whereas there is no evidence of NSGAMO in this particular test function. Nevertheless, MOTS2 demonstrated the worst performance on ZDT3, but NSGAMO successfully captured the target PF

4. MOTS2 has also been expanded to operate in parallel and integrated with external systems, so as to minimise elapsed time and increase the quality of solutions. By investigating the scalability on GPUs in a simple benchmark, it was found that the combination of hardware (GPU cores) configuration settings on a problem instance dictates which version of MOTS2 is more appropriate and where the user should expect a performance gain, based on the saturation of threads. When two benchmark functions were used for scalability studies on a specific GPU hardware, up to 7680 variables, a relatively stable zone of performance, in terms of time, was discovered.

5. The above demonstrate that MOTS2 can be used in larger problems that involve tens (or even hundreds) of variables and will be employed to optimise real-world cases in the next chapter. However, on the specific GPU hardware, the performance, in terms of time, considerably drops for more than 960 variables, which suggests an upper operational limit.

6. A new CFD flow solver that implements the LBM has been developed from scratch and deployed on CUDA, based on [64], so as to reduce the utilisation of computational resources and the overall wall-clock time to set up and simulate 3D flows in complicated geometrical devices. Compared to traditional tools and methods in CFD, it has great potential for the next generation of CFD studies as a robust method for engineering application.

7. The architecture, interfacing mechanisms and the configurations of LBM have been presented and explained, so as to enable other users to link this to their systems and processes.

8. The LBM package has been compared against the experimental data of a single case only, so as to prove that it can capture the target physics of complicated geometrical arrangements at small scale/size, for prototyping purposes. More comparisons against different cases would be required, in order to increase the confidence in the computational method. GPU- LBM was used as the underlying system to simulate the mixing capability of a microreactor, which will evaluate the objectives in the micro-mixing optimisation in section 4.2. Further verification of the results would be appropriate before drawing conclusions This is expected to affect the progress of the optimisation search and the simulation results obtained by using GPU- LBM should be handled with caution.

*This page is intentionally left blank.*

# Chapter 4

# Applications

## 4.1 Aircraft Trajectory Optimisation

A number of performance criteria have been set by the European Union and other large-scale projects with respect to the environment, citizens and passengers. The most important aims suggest reducing emissions by decreasing fuel consumption, minimising noise and shortening flight time. Different aircraft models are used in the applications below, so as to demonstrate the possibilities of integrations. The ultimate goal is to help in shaping the future of aviation by assessing the trade-offs among, fuel burn, flight time and emissions.

Modelling the evaluation of a trajectory as a black box is suggested, because critical information is frequently not available and it is easy to expand and to modify the system or any sub-system transparently while being consistent all the time. The approach suggested in [144, 145] mainly affects the decision parameters that relate to the performance of an aircraft and a few other parameters that change, whereas this methodology only affects the trajectory.

Although real trajectories are in 3D, the trajectories studied here are in 2D, for demonstration purposes. In fact, the third dimension would be useful when considering turns and would require a different model, which was not available at the time of writing this document, but it will be investigated in future studies. Without loss of generality, the same method would be applied again.

### 4.1.1 Trajectory Optimisation of Climb Phase

#### 4.1.1.1 Problem Description

The flight path of a medium-haul flight in the climbing phase is optimised. This is expected to scale up so as to cover a wide range of cases that will meet the future tactical and operational needs of air transport. Optimising the climb phase is very

important, in terms of $CO_2$ emissions fuel consumption [255]. In order to optimise the energy requirements clearly stated in [256], best practices suggest to follow the ATM procedures of Standard Instrument Departure and Standard Terminal Arrival Route, where possible [255]. Because fuel efficiency is related to duration, and vice-versa, it seems clear that saving fuel and shortening the duration (i.e., the maximum length of elapsed time) are conflicting objectives. Therefore, the objectives of the MOO problem are to minimise fuel consumption and elapsed time. More than 55% of the aircraft in European airspace are medium-haul aircraft [157]. It seems sensible to devise a methodology based on medium-haul aircraft because this will have the greatest environmental and economic impact.

Following the integration of MOTS2 into GATAC, this research is expected to demonstrate, via an application, the integration and the suitability of MOTS2 to address ATO cases. To the best of the author's knowledge, it was the first time that MOTS2, as a local-search-based optimiser, is applied to ATO problems. Equally, it is also important to demonstrate the capability of MOTS2 to deliver sensible results. Moreover, in order to assess its effectiveness in this class of problems, NSGAMO, the default optimiser that GATAC comes with, is going to be employed, for comparison purposes. In an application of bio-fuel combustion optimisation, MOTS2 was an order of magnitude (approximately 12 times) faster in delivering competitive trade-offs compared to NSGAMO [257].

#### 4.1.1.2 Methodology for the Climb Phase

This method requires a system that can simulate the trajectory of an aircraft, an optimisation algorithm and ATM constraints. The trajectory simulation software is a package that combines an Aircraft Performance Model (APM) and an Engine Performance Model (EPM) along with a flight path. Direct-search MOO algorithms are suggested, because of the number of principles involved and they can be interchanged for others with minimal effort. Regarding ATM constraints, additional information will be required for the flight procedures, such as Standard Instrument Departure and Standard Terminal Arrival Route.

A series of steps is proposed in order to perform ATO for the main phases of flight, for standard aircraft models, subject to a set of constraints, either in 2D or 3D.

1. Define trajectory and specify the number of segments.
2. Specify aircraft attributes that are required by APM.
3. Specify engine attributes that are required by EPM.
4. Define a stand-alone system that combines the above two models and a trajectory, which is specified as a multitude of points, and calculates the

elapsed time and fuel consumption, which are the cost functions for this problem.

5. Specify all the settings of the system as a single unit. This will be the black box that will evaluate the provided trajectory.
6. Formulate the optimisation problem, where ATM constraints should be considered; these will indicate speed and altitude bounds, if any.
7. Specify the configuration settings for the optimiser(s).
8. Execute the whole system either pro-actively (i.e., while monitoring the progress of the optimisation process) or as a stand-alone application.
9. Discuss the results of the optimisation process by revealing the interplay between objectives, other optimiser information and flight paths.
10. Assess the shape and performance of the revealed PF by using a compromise design and by using quality indicators.
11. Assess the contribution of each decision variable to the optimal trade-off.
12. Identify the mapping trend of decision variables to the objectives.

The specific instance of the methodology is depicted in Fig. 4.1.

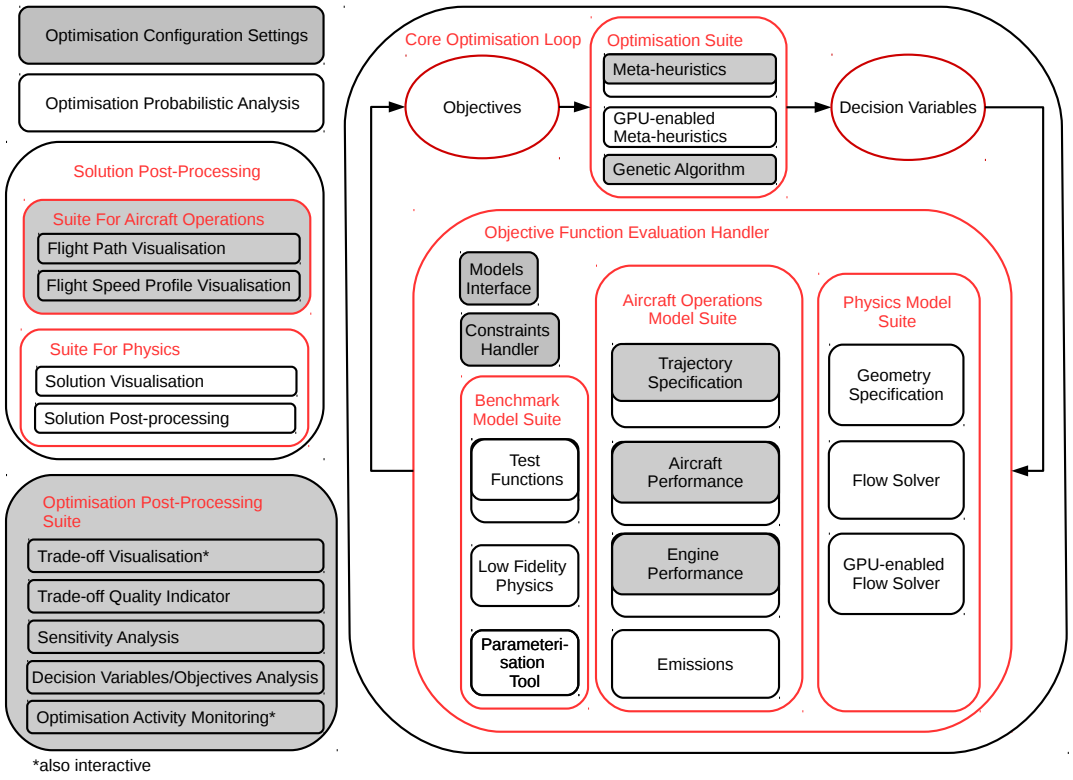



Figure 4.1: Methodology to Compare MOTS2 against NSGAMO for ATO in the Climb Phase

The trajectory consists of two types of parameters: control and state parameters. Here, it is modelled as a sequence of 2D points, as shown in Fig. 4.2, where $\gamma$ denotes the Flight Path Angle (FPA) of the flight, $V$ is the velocity magnitude, $Z$ represents the altitude and $R$ denotes the range. Hence, any point is a combination of these 4 components and many points form a trajectory. This is the absolute minimum of information to form a trajectory in 2D space. By following the modelling mechanism, as provided by GATAC, the state variables (such as aircraft weight, range number of segments) are pre-defined by the user and the control variables (altitude and airspeed) at each point are systematically handled by the optimiser, which is external to the evaluation system. The trajectory is modelled as a multitude of straight line segments, where each segment is defined by 2 points. At each point, the altitude (ALT) and speed (SPD) of the aircraft are specified. The ground projection of the segment is called *range*. Obviously, the flight path is a combination of straight line segments. An assembly of all the points of the flight path forms the decision vector.

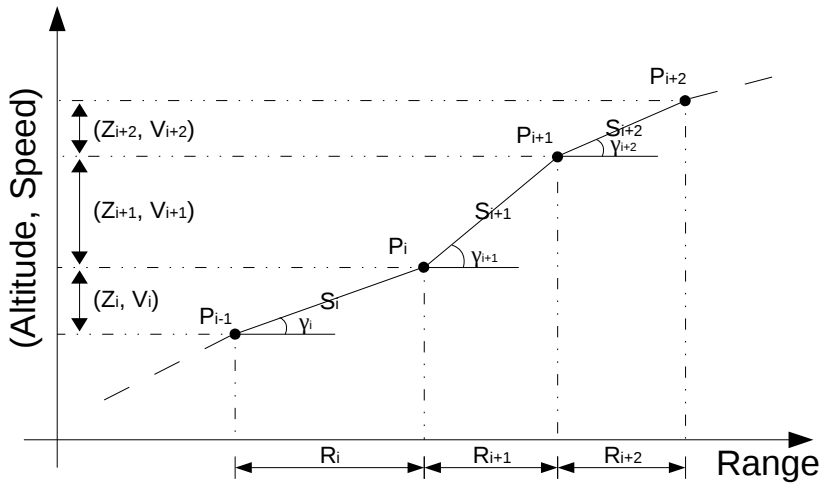



Figure 4.2: Trajectory Modelling

A brief description of the black box system that evaluates a trajectory follows. The APM and EPM from GATAC are going to evaluate each segment, by calculating the flight time and fuel consumption, which are the two objectives. In the following figures, they are represented as TIME, FUEL, respectively. APM and EPM are linked together through a simple file exchanging scheme and are applied to every segment, where parts of the objectives are calculated. Thereafter, the performance of a complete trajectory is iteratively calculated and the corresponding indices are aggregated for the whole phase. At the end of the simulation of a single segment, the APM calculates the exiting FPA and the EPM computes the mass of the consumed fuel. Thereafter, these values will be used as input for the simulation of the following segment; the exiting FPA of the previous phase will be the entering FPA of the current segment, and the mass of the fuel consumed in the previous segment will be subtracted from the overall aircraft mass of the current segment. Obviously, the overall flight time and fuel consumed for a particular flight path are going to be the sum of these values for each segment. The above procedure will be called the evaluation of the trajectory, or simply evaluation. A trajectory evaluation can take up to 2 minutes, which is prohibitive for real-time optimisation for this type of application, but it is satisfactory for a proof-of-concept application. An example of this process is shown in Fig. 4.3. It will be repeated several times under different ALT and SPD values in order to obtain the optimum behaviour.

Here, the 4-segment trajectory is parametrised by setting 6 variables: 3 for the altitude (ALT1, ALT2, ALT3) and 3 for the speed (SPD1, SPD2, SPD3), where the end points are fixed. The range of altitude and speed depend on the combination of operational limitations, the aircraft's structural limitations and

ATM constraints, where the latter are usually stricter. The range of each segment is constant and predefined. During each segment, the speed remains constant at the originally set value. This method considers only altitude and speed variables.

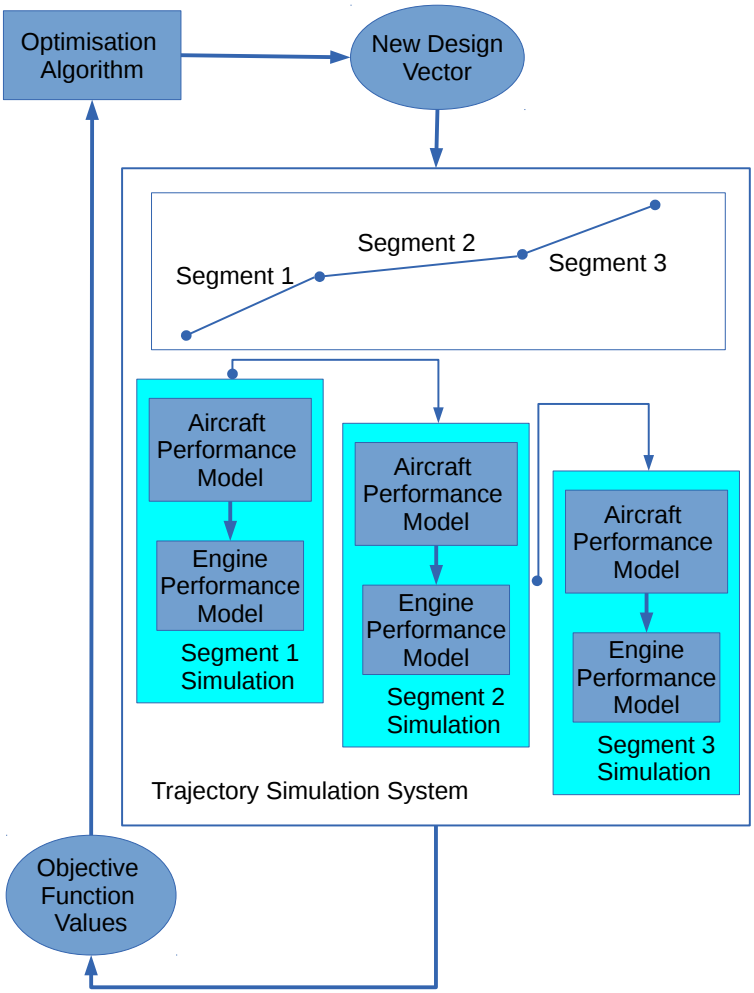



Figure 4.3: Outline of Optimisation Process on an Instance of a 3-Segment Flight

#### 4.1.1.3 Preparing the Multi-Objective Aircraft Trajectory Optimisation Process for Climb

The configuration settings for the APM and a system level description follow below. The simulated aircraft belongs to the category of medium-haul, A320 family, and carries 2 gas-turbine engines, which are further described below. The input is a

pair of points for a certain segment. The flight time and propulsion requirements are computed by the APM at the end-points of a segment. The FPA is derived from ground range and altitude intervals of a single trajectory segment. A standard mid-sized, single-aisle aircraft with two turbofan engines is modelled, where the maximum take-off weight is about 72,000 kg and the seating capacity is set to approximately 150 passengers. Based on the aircraft performance characteristics (size, lift, etc.), the APM used in this study (provided by Cranfield University and the University of Malta [258]) calculates the required thrust throughout the target segment and the respective elapsed time. Therefore, the calculated objective is the elapsed time.

The configuration settings for the EPM follow below. The input is the duration of the flight segment, elapsed time, the required thrust settings and current aircraft weight from the APM above. Then, the EPM (provided by Cranfield University [259]) is invoked to calculate the fuel consumption of the engine over the same flight period. The used engines are typical two-spool, high bypass turbofan air-engines with separate exhaust, model CFM56-5B4. Therefore, the calculated objective is fuel consumption.

The aircraft is subject to a number of constraints regarding its structural limitations (e.g., maximum travel speed, expressed in Calibrated Airspeed (CAS) [1] and Mach), operational (e.g., the maximum angle of attack) limitations and ATM restrictions (e.g., flight within certain altitude margins). All these are constraints that affect the range of variability of the components of the decision vector. In addition, hard constraints are imposed by the APM and EPM whenever the design vector produces irregular trajectories in terms of unrealistic FPA, thrust requirements, etc.

The flight envelope of the climb phase is presented in Table 4.1 and reflects the range of variability in the optimisation problem. Furthermore, for the climbing phase, a continuous ascending altitude must be used. The lower and upper bounds for both altitude and speed delimit decision space, wherein the optimiser should locate the best trajectories based on the objective values.

The communication time among the participating modules is an important factor, too. The interface between the optimisers and the trajectory simulation is handled by GATAC, as depicted in Fig. 4.3. Part of the evaluation time is spent in exchanging files among the modules, which will be improved by employing direct communication methods. The black box communication is achieved by using special dictionaries of extensible mark-up language and via directly exchanging files.

[1] is indicated airspeed after correction for instrument error(s), position error(s) and installation error(s)

Table 4.1: Climb Flight Envelope for 4-Segments ATO

| node | Range [m] | Altitude [m] | | Speed | |
|---|---|---|---|---|---|
| | | Min | Max | Min | Max |
| 0 | 0 | 1,500 | 1,500 | 130 m/s CAS | 130 m/s CAS |
| 1 | 40,000 | 3,000 | 5,500 | 130 m/s CAS | 190 m/s CAS |
| 2 | 80,000 | 3,500 | 6,000 | 130 m/s CAS | 190 m/s CAS |
| 3 | 120,000 | 5,500 | 6,000 | 140 m/s CAS | 190 m/s CAS |
| 4 | 160,000 | 6,000 | 6,000 | 0.8 Mach | 0.8 Mach |

#### 4.1.1.4 Optimisation Algorithms

In this work, two native MOO algorithms will be used in the same case and their performance will be analysed and assessed. Handling all of the objective functions at the same time without using any other kind of transformation is of paramount importance since this can deliver an unbiased trade-off [183]. The first optimiser is NSGAMO, described in subsection 2.6 (page 22), a variant of NSGA-II and the second is MOTS2, described in section 3.1. Both optimisers can operate on constrained and unconstrained problems, and they will run for 20,000 objective function evaluations. The configuration settings (Table 4.2) were chosen based on the author's experience, so as to explore decision spaces sufficiently and to generate feasible trajectories. For comparison purposes, 20,000 evaluations are specified (the initialisation factor of NSGAMO actually samples the decision space before operating as a genetic algorithm), which is anticipated to be long enough to reveal any discrepancies in the final results. These settings are not unique and are specified to be as neutral as possible, so as to allow the optimisers to explore and exploit decision space. The configuration settings used are listed in Table 4.2.

Table 4.2: Optimisation Configuration Settings for Aircraft Trajectory

(a) MOTS2

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 15 |
| Call intensification move after # non-improvements | 10 |
| Reduce the search step size after # non-improvements | 30 |
| Initial step sizes (as % of variable range) | {0.3,0.3,0.3,0.1,0.1,0.1} |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | 6 |
| # of variables | 6 |
| # of objectives | 2 |
| # of objective function evaluations | 20,000 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | 15 |

(b) NSGAMO

| **Parameter Description** | **Value** |
|---|---|
| % for creep mutate with decay | 0.01 |
| % for dynamic vector mutate | 1.01 |
| % covered dynamic vector mutate | 0.75 |
| % covered for vector mutate | 1 |
| % covered for element mutate | 0.6 |
| Convergence fitness tolerance | 1.0E-6 |
| Inflationary scheme population limit | 3 |
| Inflationary scheme starting point | 1.6 |
| Initialisation factor | 50 |
| Maximum generations | 150 |
| Population size | 100 |
| Element mutation probability | 0.05 |
| Creep mutate probability | 0.1 |
| SBX distribution coefficient | 1.0 |
| Selection pressure | 2.0 |

#### 4.1.1.5 Optimising Climb - Results and Discussion

Both optimisers ran for 20,000 evaluations under their individual settings listed in Table 4.2 and the finally discovered PF is shown in Fig. 4.4. Among the discovered solutions one representative decision vector from each trade-off is selected approximately from the middle of the plane and the corresponding trajectory, also known as the compromise design, is visualised. The solutions in the PF revealed by NSGAMO spread almost uniformly over objective space, whereas the ones discovered by MOTS2 lie in two narrower regions, but their population is larger. Hence, depending on the requirements of the application, it is impossible to choose one optimiser from the other. In general, NSGAMO delivered a large range of designs, which informs the user about the performance of the aircraft in many different scenarios. In contrast, the designs revealed by MOTS2 are denser and very close to each other. So, on different settings of altitude and speed, there are trajectories that demonstrate very similar performance in terms of fuel burn and flight time.

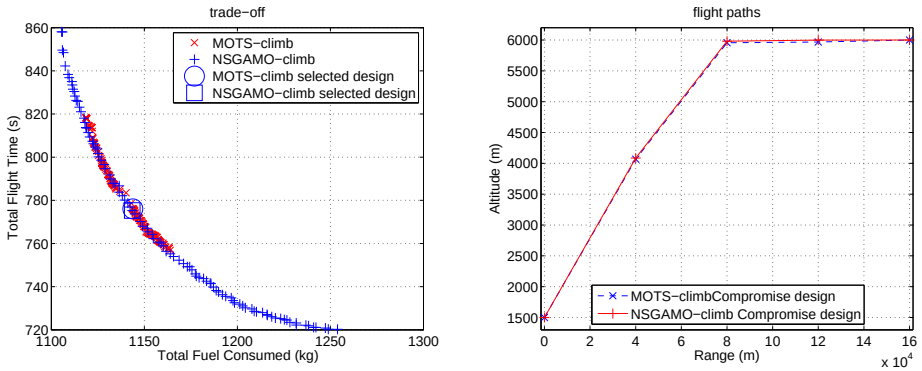


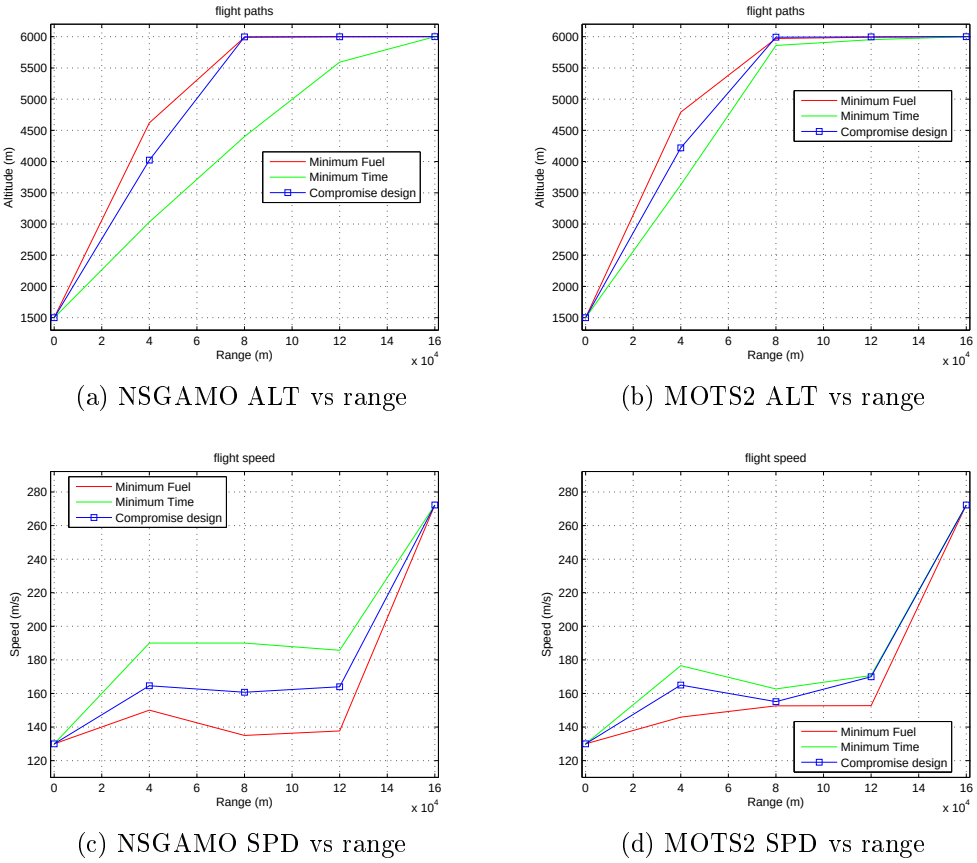

Figure 4.4: Comparing the Trade-Offs and the Compromise Designs

Despite the difference in the shape of PF, the trajectory of the compromise design generated by MOTS2 is slightly lower in altitude and quicker than NSGAMO's, but the difference in objective values is negligible, as shown in Fig. 4.5. Hence, the compromise design behaves equally well in both cases. It is obvious that the gap between the extreme designs for NSGAMO is larger. This is expected since the variation of the variables is wider and the extrema are quite distant.

A common practice in order to assess the performance of optimisers is to compare the quality of the two extreme solutions against the previously selected compromise point from the PF. Studying the difference from the compromise design to the other two gives deeper insight into the optimisation problems. The target is to compare the different designs and understand the factors that increase/decrease performance. All these solutions are depicted in Fig. 4.5. First, both optimisers

discovered similar trends. As expected, NSGAMO's flight envelope is wider, because it has discovered a wider trade-off. Both optimisers suggest that the aircraft should climb as quickly as possible to reach the top of climb of the permitted range. Thereafter, the aircraft should maintain level flight. In both cases, the top of climb should be reached at half of the specified range. The speed profiles can be divided into three zones: first, the aircraft increases speed for the first segment. Second, during the second and third segments, the speed is relatively constant, where NSGAMO's profile presents smaller changes in speed compared to MOTS2, which slows down at the third point. For the last zone, which is also the last segment of the flight, the aircraft accelerates to reach the end speed of 0.8 Mach before entering the cruise phase, as defined in the problem description in Table 4.1.

Following the previous analysis, there is a pattern in the solutions of Fig. 4.5 when moving from the fastest trajectory to the most environmentally-friendly. More specifically, in the minimum-fuel trajectory the aircraft climbs relatively slower and, in terms of altitude, the transition from the first segment to the second seems very smooth and the speed is as low a possible. Conversely, then the aim is to minimise the flight time, the aircraft tries to achieve the top of climb as fast as possible at a faster speed than the minimum-fuel trajectory.




(a) NSGAMO ALT vs range

(b) MOTS2 ALT vs range

(c) NSGAMO SPD vs range

(d) MOTS2 SPD vs range

Figure 4.5: Generated Trajectories

The significance of decision variables from the PF will be assessed by using Principal Component Analysis (PCA) [260]. This method can detect which components of the decision vector are the most energetic by calculating the variance of each parameter in the dataset. Effectively, the contribution of each decision variable to the discovered optimal trade-off is quantified. The higher the variance of a decision variable is, the more important that is to the set. Conversely, a smaller variance means that the set is less sensitive to that decision variable. In the context of optimisation, this means that any change to the decision variables with high variance will result in a considerable change in the objectives. Furthermore, decision variables with low contribution could be neglected from the set to reduce the problem complexity. The numeric PCA scores for each decision variable are listed in Table 4.3. These scores are mainly responsible for the current instance of the trade-off. If the PF changes, then the values should be recalculated. Therefore, ALT1 contains by far the highest percentage of variance for both cases, and it is considered the most significant parameter. Hence, the optimisation process should

mainly focus the search based on this parameter. This has a double implication: either the search step could be very fine for the specific parameter or the optimisation could be performed again with fewer variables, which would suppress the dimensionality and would speed up the process. Because the number of iterations is very large, setting ALT1 as the most important variable seems to be a sensible choice, but this is not guaranteed to be true until the whole search space is explored. Here, the user is informed about the importance of the discovered decision vectors that belong to the PF.

Table 4.3: Principal Component Analysis of the Decision Variables that Belong to the Optimal Trade-Off

| | **ALT1** | **ALT2** | **ALT3** | **ALT4** | **ALT5** | **ALT6** |
|---|---|---|---|---|---|---|
| NSGAMO | 0.9316 | 0.0658 | 0.0024 | 0.0001 | 0.0000 | 0.0000 |
| MOTS2 | 0.8958 | 0.1011 | 0.0028 | 0.0003 | 0.0001 | 0.0000 |

The hypervolume indicator [217] is employed to assess the performance of the discovered trade-off. The numerical values in Table 4.4 are used to quantitatively compare the performance of the PF. As for the reference point, the combination of the worst objectives was used, which is common for both trade-offs. More specifically, the richness and the span of the PF are combined in one metric: the higher the value is, the better the trade-off should be. According to hypervolume, NSGAMO achieved a higher value against MOTS2. First, by definition, the indicator favours the trade-off that spreads over decision space. Second, NSGAMO, as a genetic algorithm in combination with the configuration settings and the sampling feature, performed a global-search-based optimisation that has searched the entire decision space, before narrowing down to promising/good locations in terms of objective function fitness. Contrary, MOTS2's local-search-based behaviour along with the optimisation settings focused on specific areas, which resulted in a larger number of discovered points in a relatively smaller area. Therefore, the user is informed about the overall performance of the revealed PF. Combining the information from Tables 4.3 and 4.4, on one hand, NSGAMO discovered a better PF and identified ALT1 as the most important design variables. On the other hand, relative to NSGAMO the performance of MOTS2 with respect to hypervolume was lower, the importance of ALT1 and ALT2 was slightly lower and higher, respectively.

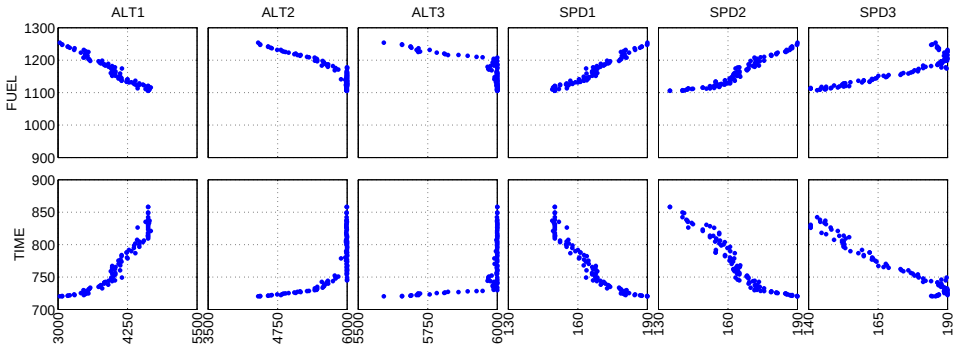
The pairwise relationship between each variable against each objective for each optimiser is depicted in Fig. 4.6. The situation for NSGAMO is straightforward. Whenever the altitude increases, the fuel consumption is reduced, and the elapsed time is longer and vice versa, not in a linear way. The same statement is also true for speed but with a smoother response. It is noteworthy that since ALT1 was identified as the most significant variable, for less than 1200 kg of consumed

Table 4.4: Hypervolume Indicator

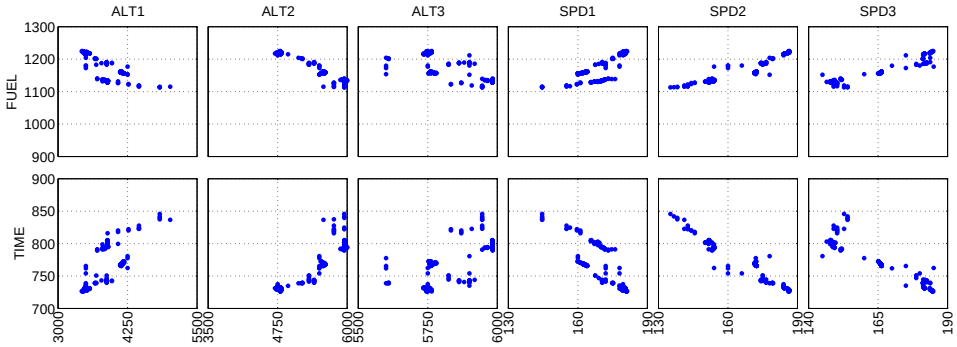
| | **Hypervolume Indicator** |
|---|---|
| | Reference Point: {1253.8, 858} |
| NSGAMO | 15109.666 |
| MOTS2 | 12579.178 |

fuel the second and third altitude parameter are almost constant. Similarly, if the elapsed time is longer than 750 s the same components remain constant. Interpreting the results from MOTS2 is less intuitive. Although the PF is denser, the mapping of the decision space to objective space presents two separate zones of performance. Again, as ALT1 increases the fuel consumption drops and the elapsed time increases. Then, as speed increases, more fuel is consumed and the flight is shorter. However, trends appear smoother. This is partly true because the trade-off is not as wide as with NSGAMO. Hence, the reduced distance between the extrema explains this behaviour. In both cases, the response of speed parameters against fuel consumption seems to follow a linear pattern. The fact that many points for ALT2 and ALT3 live at the end of the permitted range suggests that it will be sensible in the future either to perform the same study with increased upper bounds or to deploy a second study to focus on these highly visited regions of decision space. In most of the instances, speed seems to vary in harmony with fuel consumption. In any case, both optimisers revealed similar trends, where NSGAMO results are clearer to interpret.

The revealed PFs are further investigated by using the Parallel Coordinates Projection [261], which is illustrated in Fig. 4.7. It is confirmed again that the trade-off of NSGAMO is wider and equally spaced, whereas MOTS2 discovered two rich and different zones of performance. Based on the fuel axis, the domain is clustered in high and low consumption regions. Moreover, the user is informed about which regions each optimiser explored.

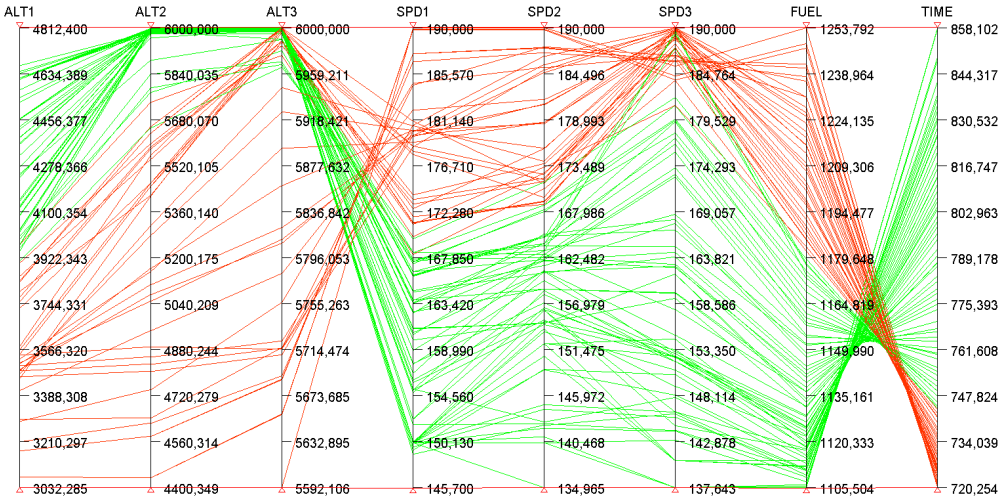



(a) NSGAMO

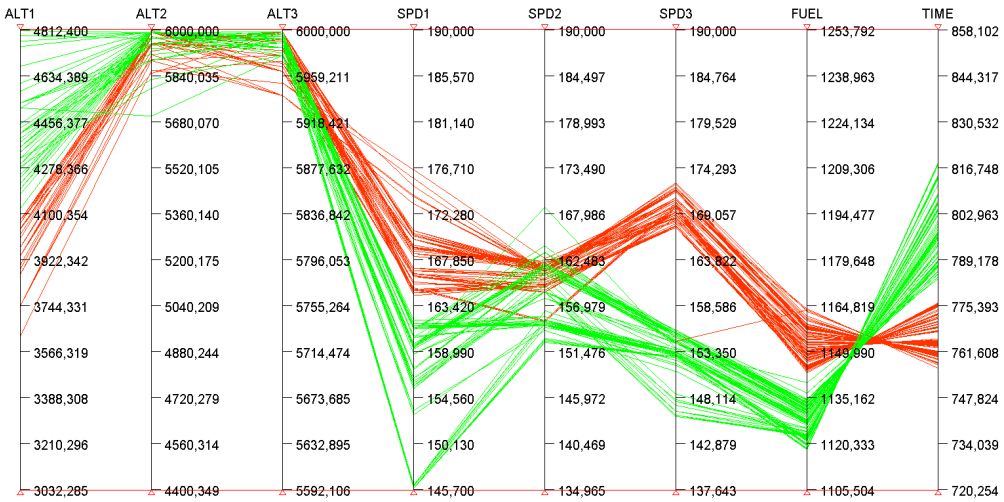



(b) MOTS2

Figure 4.6: Design Variables to Objective Functions Relationships




(a) NSGAMO




(b) MOTS2

Figure 4.7: Mapping of Decision Space to Objective Space of the PF in Parallel Coordinates Projection

Parallel Coordinates could interpret the behaviour of the optimal profiles. Initially, for NSGAMO, the population (which is 36% of the whole set) of trajectories that belongs to the upper-half of fuel consumption corresponds to a very thin range for elapsed time. This means that increasing the fuel consumption does not equally improve the elapsed time. Besides the ALT3 axis, the majority of designs do not mix and thus the flight performance could be easily separated. NSGAMO discovered designs at a larger fraction of the axes, which also justifies the span of the PF.

Two distinct zones of performance were recognised by MOTS2 with complicated interactions between the variables. In terms of the range of objectives, Tabu-Search is more balanced. The local-search scheme also affects the mixed shape of the Parallel Coordinates plot, where the concentration of coordinates is more intense from certain regions of the axes.

In general, the ratio of the number of trajectories with high fuel efficiency over the number of trajectories with low fuel efficiency for NSGAMO is 64/36 whereas the same ratio for MOTS2 is 112/217. Although the designs seem scrambled, the classification of fuel consumption can be directly resolved by ALT1. In addition, both cases revealed that fuel-efficient trajectories (i.e., the lower half of the fuel consumption axis) present very small variation in ALT2 and ALT3. Also, a similar pattern between ALT1 and ALT2 axis is presented.

#### 4.1.1.6 Identified Issues

Regarding the ATO methodology, the performance quantities of the flight should be mapped with the quantities from optimisation problems. First, the flight performance metrics represent the objective of the problem. In fact, each objective is generated by a separate principle. A number of subsystems, one for each principle, are required to provide accurate metrics. Here, the behaviour of the aircraft and its engines is resolved by the APM and the EPM on a trajectory that consists of a series of points. All these are linked into a black box system which is then linked with an optimisation algorithm to study the trade-off of the objectives and to reveal the speed profile and the altitude profile of environmentally friendly trajectories. The trajectory shape should be flexible enough and should generate sensible/flyable flight paths. Any optimisation algorithm that combines both global and local search can traverse into the highly complicated decision spaces. Ideally, the ability of the whole system to operate in parallel and on a large number of dimensions will prove it useful for real-world applications.

This work demonstrated that the newly developed MOTS2 was successfully integrated into GATAC and is suitable for ATO. When compared against NSGAMO in a 4-segment climb of a commercial passenger aircraft of the medium-haul family, different zones of performance were highlighted, all of which suggests further

exploring this case in higher dimensions. In terms of the hypervolume indicator, NSGAMO was more superior and identified a wider trade-off that is clear to interpret. Contrary, MOTS2's revealed PF, whose designs are more focused within a short range and provide many more alternatives. Compromise and extreme designs were discovered and discussed, by using both optimisers. It was found that the first variable (ALT1) is by far the most significant parameter for 4-segment climb trajectory and severely affects the performance of the flight. The findings are in agreement with flight physics and provide additional insight into the control parameters and the objective values.

### 4.1.2 Trajectory Optimisation of Climb, Cruise and Descent

#### 4.1.2.1 Problem Description

Following the previous case, climb, cruise and descent were selected for optimisation for their impact on the environment. As discussed above, although climb and descent are the most energy consuming and cruise is the longest flight phase, achieving an optimum cruise level is also very important [255]. They are combined together to demonstrate the capability of the methodology to handle more complicated cases. For the purposes of demonstrating the flexibility of the introduced methodology to operate stand-alone, GATAC will not be used. A different combination of models will be used to evaluate the objectives on a different type of aircraft.

In this section, the trade-off of the performance characteristics of a commercial aircraft for a 3-phase trajectory on a typical 200 nautical miles mission is investigated. The goal is to minimise fuel consumption, flight elapsed time and pollutants' emissions, which are conflicting by nature and closer to a real-world study, by changing the shape of the flight path. The aircraft, the engine and their respective settings will be unaltered. These values are obtained by using a combination of well-established models and by using the same methodology, as described in the previous subsection. The optimal trade-off surface is derived by employing MOTS2. Since only MOTS2 is used, all the constraint handling is hard coded in the Evaluation Manager of the optimiser. In this application, the Evaluation Manager of MOTS2 is used, so as to link all the modules together by exchanging external files. Following the black-box approach, a single wrapper was created that is called by MOTS2, when a trajectory needs to be evaluated. To the best of the author's knowledge, this is the first time a study that involves three conflicting objectives is carried out. The results provide deeper insight into understanding how the trajectory profile affects the interplay among the objectives and how this knowledge should affect the future of aviation.

#### 4.1.2.2 Methodology for the 3-Phase Trajectory

In this approach, each objective comes from a different model, which is considered as a black box. This has two advantages; it permits different models of various fidelity to be altered interchangeably without interrupting the others, and different optimisers can be applied to combinations of models. The developed framework orchestrates the information exchange by capturing and processing data before the execution of each model and finally feeding information back to the optimiser. The process starts from HERMES (the APM in this case), then feeds information to TURBOMATCH (the EPM) and finally comes to P3T3 (the emission prediction model), in this order, as described below. This is repeated whenever the optimiser requires the evaluation of a given set of parameters.

The 3-phase trajectory of a single aircraft, without diversion, is resolved at once. All three flight phases are calculated one after the other. The take-off, early climb, approach and landing phases are not considered for the optimisation. They are very specific and subject to a number of conditions and parameters that cannot be modelled and/or controlled, such as weather, and also depend on the pilot's judgement. Also, the aircraft congestion will not affect the results at the current stage.

Due to noise restrictions, the speed of the aircraft near the airport area should be preserved under a certain threshold. In fact, this type of constraint affects the range of variability of the parameters. For the climb phase this is 250 knots calibrated airspeed and for the descent phase the upper limit is initially 250 knots calibrated airspeed and then drops to 220 and 160 knots calibrated airspeed. Of course, the lower bound of variability is the minimum value under which the aircraft cannot fly. In the cases where it is not possible to precisely achieve the requested flight speed, small safety margins have been added, which slightly widens the range of variability for the respective parameters. These constraints were extracted from official Standard Instrument Departure and Standard Terminal Arrival Route diagrams from the source and destination airport.

The ATM constraints are imposed to increase/secure minimum separation among aircraft. By problem definition, during the cruise phase level flight is performed. Speed values during the descent phase should be continuously decreasing. In addition, following ATC regulations, there are two main restrictions for the cruise phase. First, an aircraft can fly within a zone of 1,000 ft. However, if it needs to move to another zone, this should be (multiples of) 2,000 ft either higher or lower than the current one. Hence, all of the proposed trajectories can be considered as flyable.

The decision space is composed of 44 parameters, which is a combination of the trajectory altitude and speed values at various points. The 3 objectives to be minimised are total block time (in minutes), total block fuel (in kg of burnt fuel)

and NOx (in kg of the emitted pollutant), each calculated by a different model. In the following figures, they are represented as TIME, FUEL and NOx, respectively.

The trajectory is decomposed into a number of segments, listed in Table 4.5. Originally, HERMES models aircraft performance by receiving a wide number of parameters such as aircraft specifications (e.g., geometry), engine specifications, aircraft taxiing specifications and flight path details. Here, only the parameters that describe the flight path are modified, because the focus is on generating alternative flight paths by using existing configurations of aircraft. Only the three basic flight phases will be considered, since they represent more than 90% of flight duration. The specification of the flight envelope is slightly different from Table 4.1, presented above, as the structure of input files for HERMES is different from APM, so as to provide a flyable trajectory. More specifically, 18 segments for the climb phase and 10 segments for the descent phase are defined. A long cruise phase is used, where only a single pair of speed and altitude values is specified. The overall range of the flight is specified (and fixed) but the individual ranges of segments are automatically resolved by HERMES. When the aircraft reaches the top of climb, it continues on the cruise phase until it reaches the top of descent. The duration of climb is an assembly of multiple fixed-time-length segments. The descent phase performs a continuous descent approach for the altitude values, which is the most optimal arrival way for aircraft to approach the runway. For this phase, the altitude is automatically resolved and only speed values vary. Although this modelling approach is different because of the combination of software packages that resolve the flight physics, fewer variables are involved, which simplifies the problem.

Table 4.5: Flight Envelope for 3-phase ATO and Mapping to Optimisation Problem Variables

| **Phase/Node** | **Altitude [m]** | | | **Speed** | | |
|---|---|---|---|---|---|---|
| | **Min** | **Max** | | **Min** | **Max** | |
| Climb fixed node | 475 | 475 | | 250 m/s CAS | 250 m/s CAS | (variable 10) |
| Climb node 0 | 475 | 1500 | (variable 1) | auto-specified | | |
| Climb node 1 | 1000 | 2500 | (variable 2) | 250 m/s CAS | 350 m/s CAS | (variable 18) |
| Climb node 2 | 2000 | 3048 | (variable 3) | 250 m/s CAS | 350 m/s CAS | (variable 19) |
| Climb node 3 | 3048 | 12000 | (variable 4) | 250 m/s CAS | 350 m/s CAS | (variable 20) |
| Climb fixed node | 3048 | 3048 | | 250 m/s CAS | 250 m/s CAS | (variable 11) |
| Climb node 4 | 3048 | 12000 | (variable 5) | 250 m/s CAS | 350 m/s CAS | (variable 21) |
| Climb node 5 | 3048 | 12000 | (variable 6) | 250 m/s CAS | 350 m/s CAS | (variable 22) |
| Climb node 6 | 3048 | 12000 | (variable 7) | 250 m/s CAS | 350 m/s CAS | (variable 23) |
| Climb node 7 | 3048 | 12000 | (variable 8) | 250 m/s CAS | 350 m/s CAS | (variable 24) |
| Climb node 8 | 3048 | 12000 | (variable 9) | 250 m/s CAS | 350 m/s CAS | (variable 25) |
| Climb node 9 | 3048 | 12000 | (variable 10) | 250 m/s CAS | 350 m/s CAS | (variable 26) |
| Climb node 10 | 3048 | 12000 | (variable 11) | 250 m/s CAS | 350 m/s CAS | (variable 27) |
| Climb node 11 | 3048 | 12000 | (variable 12) | 250 m/s CAS | 350 m/s CAS | (variable 28) |
| Climb node 12 | 3048 | 12000 | (variable 13) | 250 m/s CAS | 350 m/s CAS | (variable 29) |
| Climb node 13 | 3048 | 12000 | (variable 14) | 250 m/s CAS | 350 m/s CAS | (variable 30) |
| Climb node 14 | 3048 | 12000 | (variable 15) | 250 m/s CAS | 350 m/s CAS | (variable 31) |
| Climb node 15 | 3048 | 12000 | (variable 16) | 250 m/s CAS | 350 m/s CAS | (variable 32) |
| Climb node 16 | 3048 | 12000 | (variable 17) | auto-specified | | |
| Cruise | 12000 | 12800 | (variable 33) | 0.8125 Mach | 0.875 Mach | (variable 34) |
| Descent node 0 | auto-specified | | | 150 knots | 250 knots | (variable 35) |
| Descent node 1 | auto-specified | | | 150 knots | 250 knots | (variable 36) |
| Descent node 2 | auto-specified | | | 150 knots | 250 knots | (variable 37) |

Continued on next page

**Table 4.5 – continued from previous page**

| Phase/Node | Altitude [m] | | Speed | | |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| Descent node 3 | auto-specified | | 150 knots | 250 knots | (variable 38) |
| Descent node 4 | auto-specified | | 150 knots | 250 knots | (variable 39) |
| Descent node 5 | auto-specified | | 150 knots | 250 knots | (variable 40) |
| Descent node 6 | auto-specified | | 150 knots | 250 knots | (variable 41) |
| Descent node 7 | auto-specified | | 135 knots | 145 knots | (variable 42) |
| Descent node 8 | auto-specified | | 135 knots | 145 knots | (variable 43) |
| Descent node 9 | auto-specified | | 135 knots | 145 knots | (variable 44) |

The suggested methodology steps follow:

1. Define trajectory and specify the number of segments for climb and descent.
2. Specify aircraft attributes that are required by APM.
3. Specify engine attributes that are required by EPM.
4. Specify engine attributes that are required by the emission prediction model.
5. Define the Evaluation Manager of MOTS2 that combines the above three models and a trajectory, which is specified in an external file (which should be generated by MOTS2), and calculate the elapsed time, fuel consumption and emissions, which are the cost functions for this problem.
6. Specify all the settings of the system as a single unit. This will be the black box that will evaluate the provided trajectory.
7. Formulate the optimisation problem, where ATM constraints should be considered; these will indicate speed and altitude bounds, if any.
8. Specify the configuration settings for the optimiser(s).
9. Execute the whole system either pro-actively (i.e., while monitoring the progress of the optimisation process) or as a stand-alone application.
10. Discuss the results of the optimisation process by revealing the interplay between objectives, other optimiser information and flight paths.
11. Since 3 objectives are involved, the parallel coordinates projection will be used to analyse the interplay among the objectives.
12. Assess the contribution of each decision variable to the optimal trade-off.
13. The trajectories will be visualised and information from the flight path will be extracted.
14. The discussion will focus on the variables that correspond to the most extreme objectives.
15. In order to demonstrate the merits of the optimisation process, for each objective, the parameters that correspond to each minimum objective will be compared. This serves in understanding how the shape of the trajectory alters depending on which performance criterion is considered as the most important.

The specific instance of the methodology is depicted in Fig. 4.8.

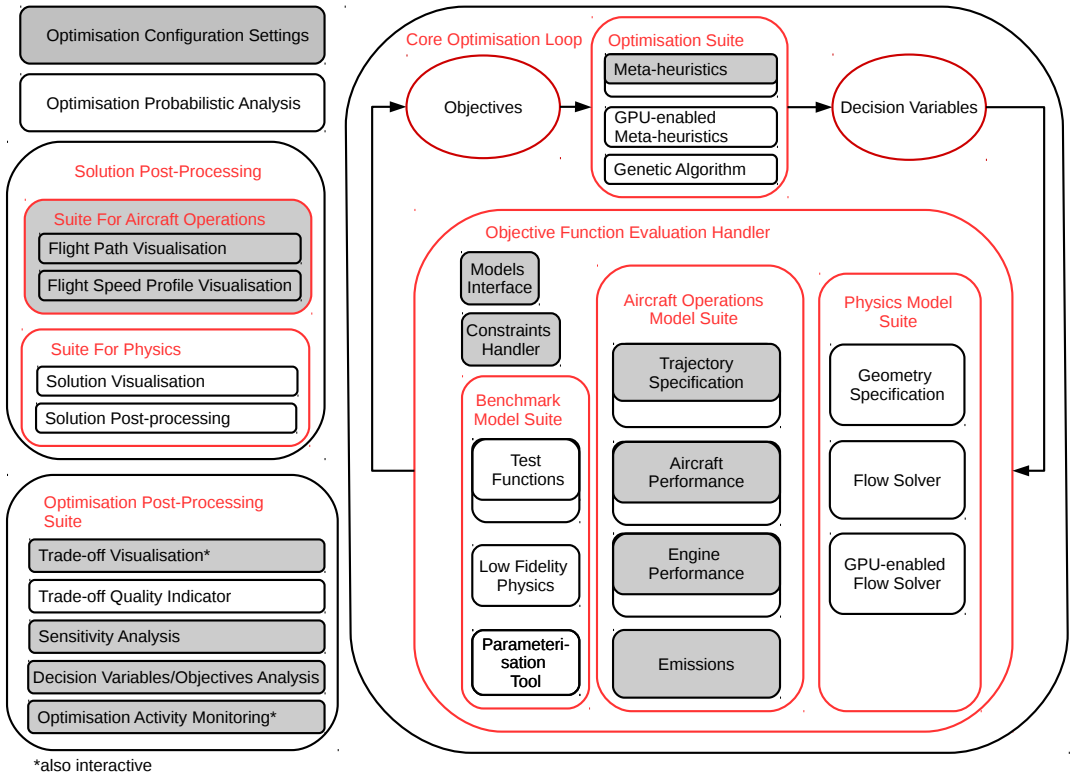



Figure 4.8: Methodology for Optimising a 3-Phase Trajectory by using MOTS2

#### 4.1.2.3 Preparing the Multi-Objective Optimisation Process of the 3-phase ATO

The simulated aircraft is a Boeing 737-800 with Engine CFM56-7B27 and flies from Heathrow (London) to Schiphol (Amsterdam). This is a very frequent flight, carried out daily by the KLM airline with the same aircraft. Although the combination of aircraft, engine and city-pair is very specific, the results can provide a trend for short-haul flights and the methodology can equally be applied to other combinations, too. The airports of London and Amsterdam were chosen not only because they are very strategic airports for serving all the range of flights, but also the city pair is one of the most frequently operated. So, they serve as good demonstrators of the proposed methodology. The results not only can affect the shape of the trajectory and cost indices assigned to trajectories, but also, can result in a change of the current trajectories. The distance between the two airports is 427.65 km and, on average, the phases of climb, cruise and descent are estimated to be 60.5 km, 294.9 km and 72.25 km, respectively.

The ground distance of the visualised altitude and speed profiles, see Fig. 4.16 and 4.17, are automatically resolved by HERMES according to the flight speed. So, the user (and to some extent the optimiser) cannot directly specify it. The main reason is that HERMES always delivers a flyable trajectory, as opposed to other approaches, where a point mass model is used and the user needs to specify this information, too. Hence, the overall range slightly differs for each trajectory. Compared to the other aircraft performance models, HERMES is more appropriate, as it provides the means to reduce the complexity of specifying each point of a trajectory.

For the purpose of this study, the simulated engine was developed using the Cranfield University in-house engine performance code TURBOMATCH [262,263]. The engine under consideration in this study is based upon a CFM56 type engine, which is modelled as shown in Fig. 4.9 and is currently used to power the Boeing 737-800 aircraft. The baseline engine is a two-spool turbofan with a booster stage including a high bypass ratio, separate exhausts, customer bleeds and cooling bleed off-takes. A summary of the main parameters for the baseline specification is given in Table 4.6.

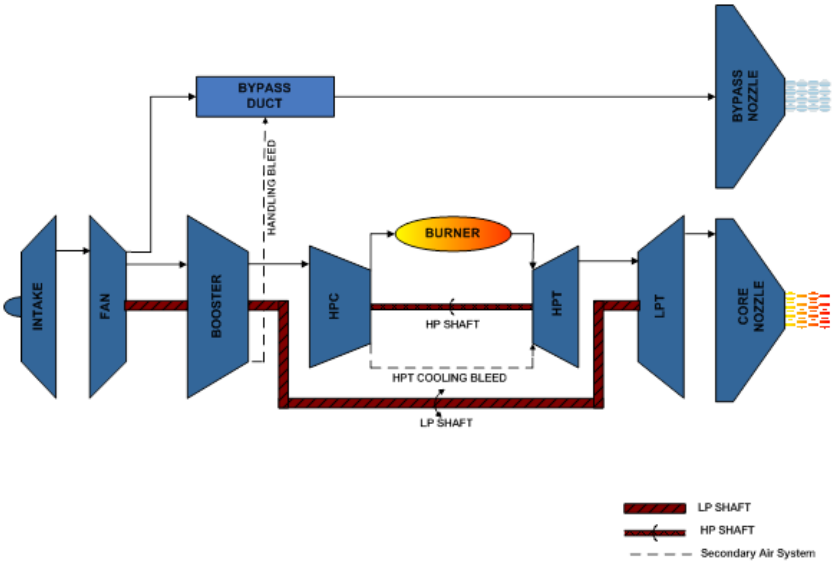



Figure 4.9: Engine Model Schematic

| Engine Parameter Model | Specifications |
|---|---|
| Engine design point altitude (m) | 10668 |
| Design flight Mach no | 0.8 |
| Top of climb thrust (N) | 26600 |
| Top of climb specific fuel consumption (mg/N s) | 16.18 |
| Top of climb turbine entry temperature (K) | 1560 |
| Top of climb mass flow (kg/s) | 145 |
| Take-off thrust (N) | 121400 |
| Take-off fuel flow (kg/ s) | 1.09 |
| Take-off turbine entry temperature (K) | 1660 |
| Take-off bypass ratio | 5.1 |
| Fan pressure Ratio | 1.8 |
| Booster pressure ratio | 1.85 |
| High-pressure compressor pressure ratio | 1.85 |
| Overall pressure ratio | 32.8 |
| Isentropic compressor efficiencies | 0.85 |
| Isentropic turbine efficiencies | 0.91 |
| Combustor efficiency | 0.99 |

Table 4.6: Engine Design Specification

The design point of the engine model was selected at top of climb, i.e., altitude: 10668 m, Mach number 0.8, and the pressure recovery of 0.99 under International Standard Atmospheric conditions. Several iterations were performed using the model at design and off-design conditions to match the performance of the model with the data obtained from the public domain for the engine on which the design was based [264].

The mass flow rate of the engine intake was estimated based on the measured nacelle area and assuming an average inlet Mach number of 0.55 – 0.65. The design point (at the top of climb) bypass ratio and the turbine entry temperature were determined based on the overall pressure ratio and the net thrust at the top of climb. The optimum fan pressure ratio corresponding to the calculated turbine entry temperature, overall pressure ratio and bypass pressure ratio were also specified. In addition to the above, compressor pressure ratios, component efficiencies, and compressor bleeds for turbine cooling, and other parameters, were initially randomly set and then iterated to match the required engine performance at design point and off-design (maximum take-off and cruise) conditions [265, 266].

Finally, the model has been tested and validated against different off-design conditions such as several thrust ratings and corresponding fuel flow rates available in the public domain. The summary of the main parameters for the specification

is given in Table 1 of [267]. The validated engine model has been used to simulate many off-design conditions required by the aircraft performance model and emission model to calculate fuel burn and emissions for each flight segment as well as for the full mission.

The software package that has been used to simulate the integrated aircraft engine performance is called HERMES. It has been developed at Cranfield University in order to assess the performance of conventional aircraft and potential benefits of novel aircraft configurations [268, 269]. The code consists of the following modules:

1. Aircraft configuration module: The required input data comprise the basic information used to define the aircraft shape and the geometry, atmospheric data and finally the information of required mission profile [263].

2. Mission profile module: The user specifies the climb schedule, cruise speed and altitude (including any stepped cruise requirements) and descent schedule of the aircraft.

3. Atmospheric and aerodynamic module: The above information is passed to the atmospheric module and aerodynamic module to calculate the aerodynamic performance of the complete aircraft [270].

4. Engine data module: The mission profile data is also used by the engine data module to determine the off-design operational conditions of the engine to calculate the engine performance required for various segments of the mission profile defined by the user.

5. Aircraft performance module: The information from the rest of the modules is passed to the aircraft performance module where the detailed figures are produced and the overall performance of the aircraft is computed.

The output includes total fuel required to complete the given mission, flight duration, and distance covered for each flight segment. In addition, the model is capable of producing component level engine performance parameters such as temperatures, pressures and mass flows along with the overall engine thrust, and specific fuel consumption [263]. The baseline aircraft in this study is a single aisle short range twin engine aircraft similar to the Boeing 737-800. A summary of the main characteristics is given in Table 4.7. The complete flow chart of the HERMES aircraft model is shown below in Fig.4.10. Whenever the system yields the aforementioned indicator, it produces a flyable aircraft trajectory, as explained in greater detail next. Here, HERMES is configured to simulate the operation of an aircraft Boeing 737-800, by specifying the number of points.

| **Main Model Parameters** | **Specifications** |
|---|---|
| Passenger capacity | 150 |
| Mission range (km) | 3000 |
| Maximum take-off weight (kg) | 79242 |
| Maximum landing weight (kg) | 66349 |
| Payload (kg) | 20540 |
| Operating empty weight (kg) | 41145 |
| Maximum fuel capacity (l) | 26020 |
| Maximum operating Mach no | 0.785 |
| Cruise altitude (m) | 10675 |
| Fuselage length (m) | 38.08 |
| Fuselage diameter (m) | 4.01 |
| Wing span (m) | 34.3 |

Table 4.7: Aircraft Model Specifications

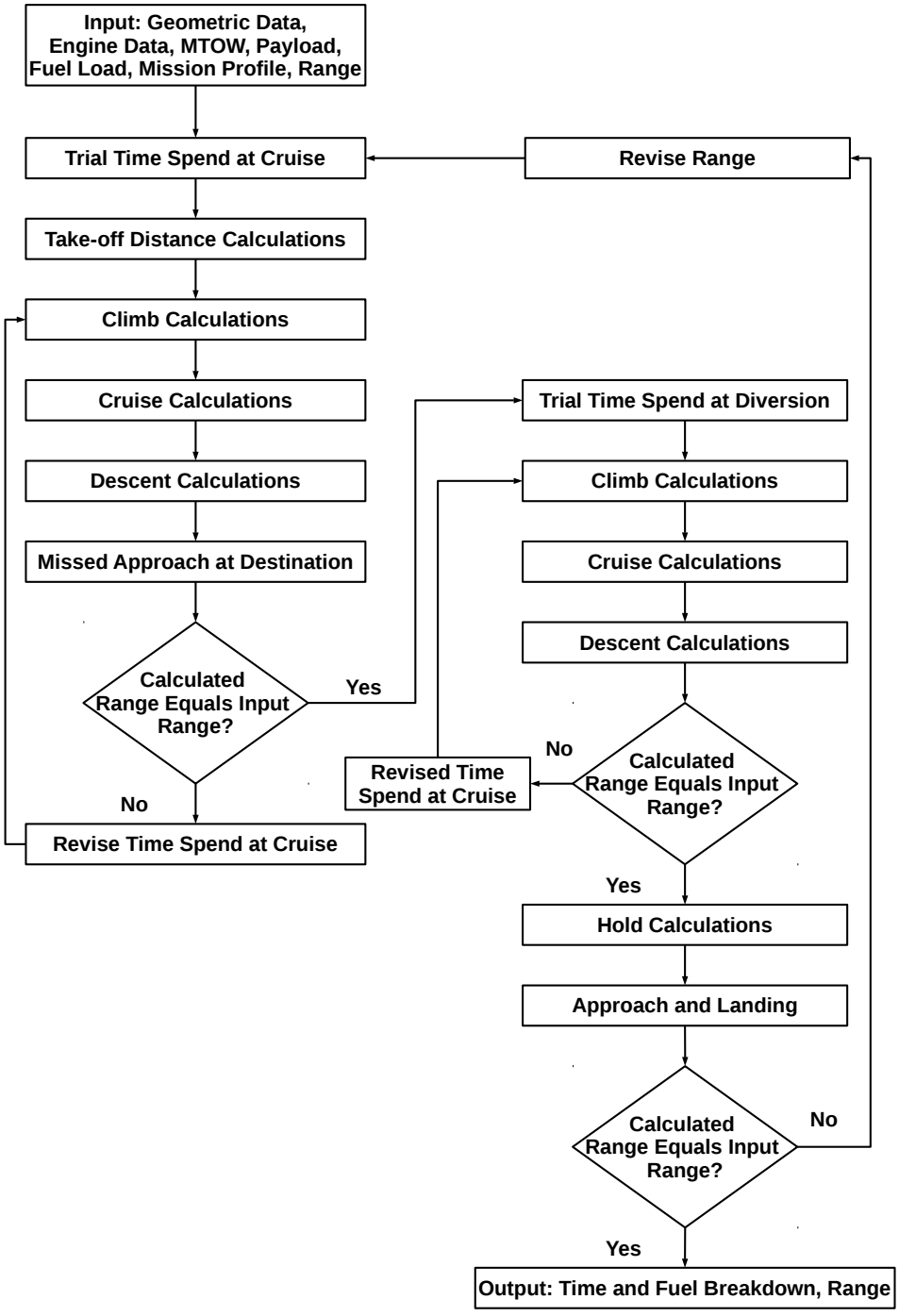



Figure 4.10: HERMES Aircraft Performance Model Flow Chart [263]

The emission prediction model used in this work is the P3T3 empirical cor-

relation model. This model estimates the level of emissions at altitude using a correlation with the emissions measured at ground level [271, 272]. This methodology is straightforward. Firstly, during the certification test of the engine the emission indices are measured. Then, it is required to correct them to take into account the variation of altitude and flight speed. In order to do that, it is necessary to know the combustion parameters for the operating conditions at both ground level and at altitude. These parameters are burner inlet pressure (P3) and temperature (T3), fuel and air ratio and fuel flow. In addition, the model takes into account the variation of humidity from the sea level to the altitude [272]. The model is capable of predicting all the emissions, but here the focus is given to the NOx emissions only [220].

Emission index NOx measurements at ground level are interpolated for different combustor inlet temperatures. Moreover, as explained above, in order to calculate the emissions at certain flight altitude and speed, the combustor inlet temperature, inlet pressure and air mass flow have to be known (by the aircraft performance model). Even if these values are not measured during the International Civil Aviation Organisation (ICAO) tests they can be assessed using the gas turbine simulation software package (TURBOMATCH). At this point, similarly to emission index NOx, burner inlet pressure and fuel air ratio are interpolated for different burner inlet temperatures, as shown in Fig. 4.11.

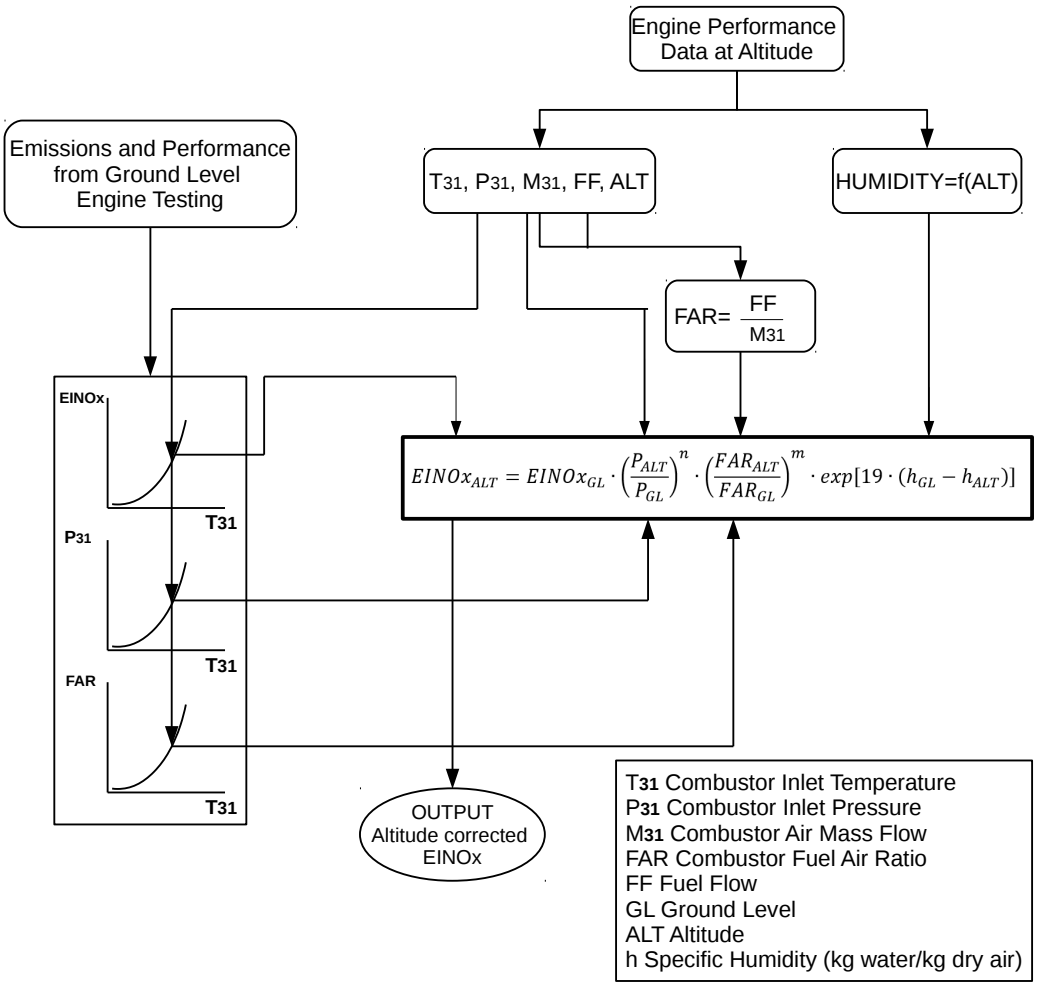



Figure 4.11: Calculating the Corrected Emission Index of NOx at Altitude by using the Emission Model [272]

Then, by using the combustor inlet temperature it is possible to obtain the respective value of emission index NOx at ground level. This value of emission index NOx is then corrected to take into account the differences in fuel and air ratio and inlet combustor pressure between ground level and altitude. The values of exponent "n" and "m" establish the severity of emission index NOx correlation. Finally, a correlation for the humidity influence is also taken into account [272]. Having calculated the value of emission index NOx, the emitted NOx in kilogrammes is given by:

$$NOx = (fuel\,flow) \cdot time \cdot (emission\,index\,NOx)$$

where the fuel flow is in [kg/s], and the time is in seconds. The model is based on correlations and the main advantage of using the P3T3 model with respect to other models such as multi-stirred reactor emissions models is the low computational time, which is very important when performing optimisation studies. The required computational time is a key feature for a model to be used in a multi-objective ATO study considering the large number of calculations.

Table 4.8: Specification of Variables for 3-phase ATO with Reference to Table 4.5

| **Variable ID** | **Type (Node)** | **Min NOx** | **Min FUEL** | **Min TIME** | **Units** |
|---|---|---|---|---|---|
| 1 | Climb ALT | 776.937 | 775.784 | 781.55 | m |
| 2 | Climb ALT | 1360 | 1360 | 1045 | m |
| 3 | Climb ALT | 1632.34 | 1656.59 | 1211.92 | m |
| 4 | Climb ALT | 2626.08 | 2764.12 | 2902.16 | m |
| 5 | Climb ALT | 2922.2 | 2946.85 | 3119.4 | m |
| 6 | Climb ALT | 3001.4 | 3001.4 | 3494.4 | m |
| 7 | Climb ALT | 3277.8 | 3425.7 | 3869.4 | m |
| 8 | Climb ALT | 3850 | 3830.65 | 4159.6 | m |
| 9 | Climb ALT | 3870.2 | 3870.2 | 4225 | m |
| 10 | Climb ALT | 3990.4 | 3952.3 | 4600 | m |
| 11 | Climb ALT | 4975 | 4136.8 | 4975 | m |
| 12 | Climb ALT | 5045.2 | 5045.2 | 5350 | m |
| 13 | Climb ALT | 5420.2 | 5382.1 | 5725 | m |
| 14 | Climb ALT | 5947.6 | 6100 | 6100 | m |
| 15 | Climb ALT | 6475 | 6322.6 | 6475 | m |
| 16 | Climb ALT | 6545.2 | 6354.7 | 6850 | m |
| 17 | Climb ALT | 8749 | 9053.8 | 8444.2 | m |
| 18 | Climb SPD | 220.75 | 221.562 | 224 | knots |
| 19 | Climb SPD | 266.25 | 246.75 | 224 | knots |
| 20 | Climb SPD | 272.75 | 272.75 | 276 | knots |
| 21 | Climb SPD | 272.75 | 273.562 | 276 | knots |
| 22 | Climb SPD | 289 | 277.625 | 289 | knots |
| 23 | Climb SPD | 289 | 289 | 315 | knots |

Continued on next page

**Table 4.8 – continued from previous page**

| Variable ID | Type (Node) | Min NOx | Min FUEL | Min TIME | Units |
|---|---|---|---|---|---|
| 24 | Climb SPD | 289 | 289 | 315 | knots |
| 25 | Climb SPD | 307 | 294 | 320 | knots |
| 26 | Climb SPD | 320 | 321.625 | 320 | knots |
| 27 | Climb SPD | 320 | 323.25 | 320 | knots |
| 28 | Climb SPD | 326.5 | 326.5 | 320 | knots |
| 29 | Climb SPD | 336.25 | 336.25 | 320 | knots |
| 30 | Climb SPD | 339.5 | 336.25 | 346 | knots |
| 31 | Climb SPD | 349.25 | 341.125 | 346 | knots |
| 32 | Climb SPD | 349.25 | 341.125 | 346 | knots |
| 33 | Cruise ALT | 8828.8 | 8828.8 | 7000 | m |
| 34 | Cruise SPD | 0.708 | 0.724 | 0.796 | Mach |
| 35 | Desent SPD | 248.1 | 248.1 | 243.1 | knots |
| 36 | Desent SPD | 246.5 | 246.5 | 241.5 | knots |
| 37 | Desent SPD | 227.9 | 230.4 | 232.9 | knots |
| 38 | Desent SPD | 227.5 | 225.625 | 225 | knots |
| 39 | Desent SPD | 220.6 | 223.1 | 213.1 | knots |
| 40 | Desent SPD | 187.2 | 188.45 | 184.7 | knots |
| 41 | Desent SPD | 165.9 | 173.4 | 160.9 | knots |
| 42 | Desent SPD | 145 | 145 | 144 | knots |
| 43 | Desent SPD | 138 | 137.125 | 137 | knots |
| 44 | Desent SPD | 137 | 135 | 137 | knots |

The optimiser settings, as listed in Table 4.9, are based on the experience of the author and pre-processing, where sensitivity analysis has also been performed beforehand in order to resolve all the optimisation settings related to each parameter individually. The optimiser settings are specified to combine exploration and exploitation of the decision space. The optimisation search keeps running until there is not any significant improvement of the PF (i.e., more than 200 objective function evaluations without any change of the hypervolume).

Table 4.9: MOTS2 Configuration Settings for 3-phase ATO

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 20 |
| Call intensification move after # non-improvements | 10 |
| Reduce the search step size after # non-improvements | 60 |
| Initial step sizes (as % of variable range) | 0.1 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | 6 |
| # of variables | 44 |
| # of objectives | 3 |
| # of objective function evaluations | 20000 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | 20 |

#### 4.1.2.4 Optimisation Progress

The optimiser carried out 1581 iterations and its progress is depicted in Fig. 4.12. Initially, MOTS2 behaves as a local search optimiser, since it only performs the intensification move for the first third of its progress. Then, it diversifies the search and reduces the search step a couple of times, since finding a better design was not possible with the current search settings. Thereafter, it keeps searching again locally with sporadic calls to diversification and reduce move until the 1200th iteration, where diversification and reduce were consecutively called for a number of times to discover new designs. The latter means that improving the PF was not possible and a change to the search settings was required, which seems to be a correct choice because of the number of better designs discovered. For the remainder of its progress, the local-search scheme was used until the end of the

computational budget. Primarily, by conducting local search most of the optimal results were obtained, which proves the suitability of MOTS2 for this case. This was inferred because the diversification move was used a few times.

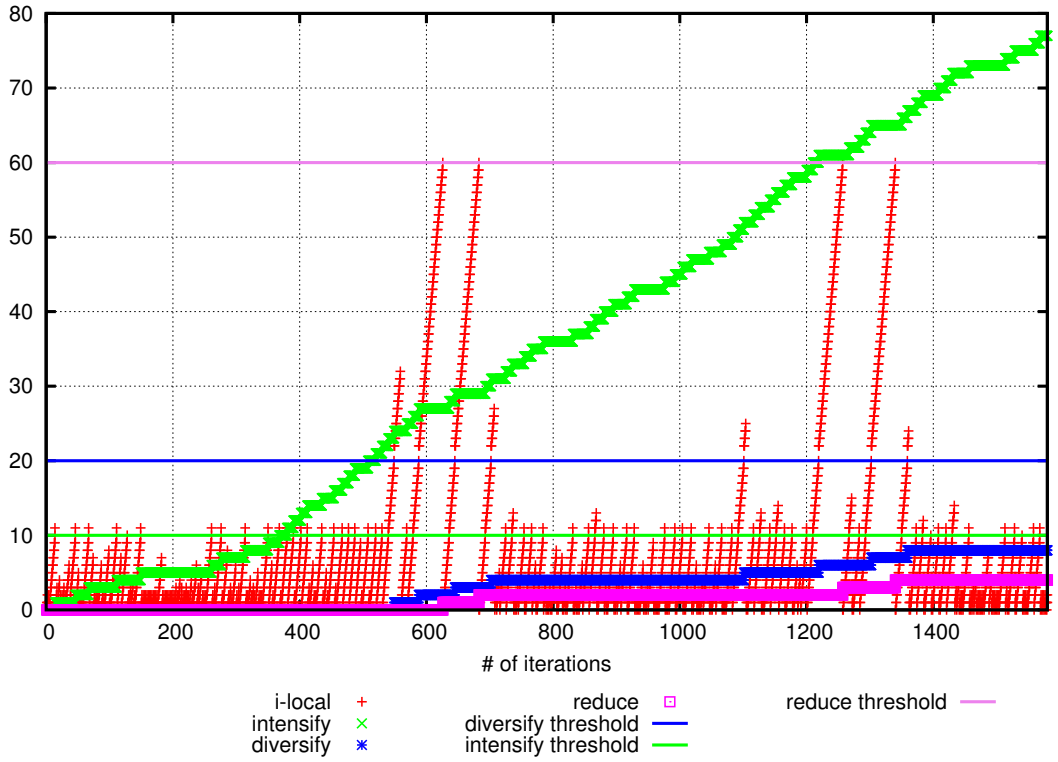
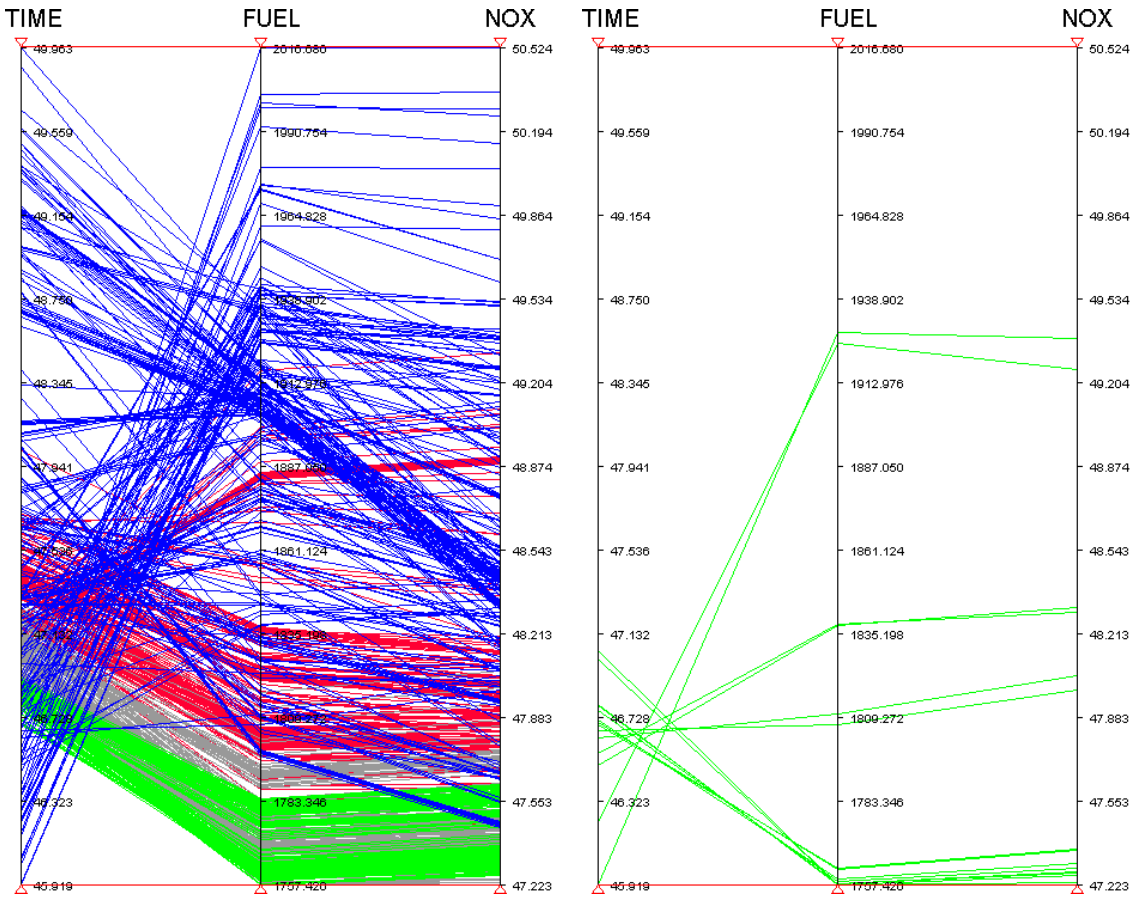



Figure 4.12: MOTS2 Search Progress in 3-phase ATO

The results of the optimisation are illustrated in Fig 4.13. For completeness, data since the start of the optimisation process, i.e., HISTORY, and the optimal ones are presented. By performing all the possible permutations between the axes that represent the objectives it is proven that all the objectives are negatively related to each other, and hence they are conflicting. Although this statement is more obvious in Fig. 4.13a, it is not always true, which means that the objectives are conditionally conflicting in nature and it is interesting to notice under what circumstances they vary in harmony. As will be discussed later, the more the optimiser approaches the optimal set, the less conflicting the objectives will be. This is demonstrated by the non-intersecting lines that connect adjacent axes and by the scarcity of designs in the non-dominated set.




(a) Incremental history progress of valid designs, blue-13608 evaluations, red-31692 evaluations, gray-49821 evaluations, green-67851 evaluations

(b) Optimal set

Figure 4.13: Parallel Coordinates Projection of the Objective Space of 3-phase ATO

Understanding how the optimiser advances through the objective space, as shown in Fig. 4.13a, indicates the complexity of the problem. This figure presents the distinct performance (objective-wise) of all the valid designs explored. By nature, all the objectives are conflicting, since the parallel coordinates projection informs the user that axis-parameters are negatively related, where lines cross each other. For ease of understanding the progress, HISTORY is linearly split into four mutually exclusive sets based on the number of evaluations, coloured differently. The blue set comes first, which is the most scattered, then the other colours incrementally form the history progress. There is a wide range of designs discovered across a relatively large region of the objective space that is not within the optimal set. However, interestingly, several time-optimal solutions (as depicted

in Fig. 4.13b) were found from the early stages of the optimisation, which means that it is relatively easier to minimise the elapsed time. As the search step is refined, certain regions of the design space have been intensively explored, which yields a few thick bands of performance in the objective space. Gradually, the following performance areas are thinner than their predecessors and also lower, which means that the optimiser converges to the optimal region. Therefore, the last region, coloured in green, is significantly low and contains most of the non-dominated designs in terms of fuel and NOx.

Another metric of importance for the optimal objectives can be their interplay, see Fig. 4.13b. More specifically, a little change in the time axis yields significant performance difference in the other objectives. For instance, less than two minutes flight time can result in more than 170 kg of consumed fuel and 2 kg of NOx emitted in the atmosphere. This observation can be integrated into the optimiser's logic so as to speed-up and/or affect the whole process. First, understanding which objectives are easier to optimise, that means their minimum can be reached within a relatively small number of objective function evaluations, can quickly advance the optimisation process. Second, the optimiser can focus on improving the performance of the objective that presents the larger gap of performance among the extrema. Finally, this can be an indication about the ranking of importance of the objectives and this information can be particularly useful at the decision-making stage.

Via using parallel coordinates interactively an interesting relationship among the objectives has been discovered. It was found that for the optimal designs both fuel and NOx objectives mostly live in harmony, as was already demonstrated in Fig. 4.13b. They are not related linearly, but they increase and decrease together. However, during the initial and middle phase of the optimisation process all of the objectives conflict with each other. Therefore, it is suggested to start a 3 objectives optimisation to guide the search and after a large number of iterations (more than 2/3 of the computational budget), the problem should switch to 2 objectives when the objectives start varying in harmony. This functionality, which could potentially reduce the problem's complexity, should be carried out within the optimiser's core.

#### 4.1.2.5 Comparing the Variables and Objectives

Finding out which variables drive the optimisation process is crucial and certainly affects the speed and quality of the optimiser. Here, the same methodology is applied both to HISTORY and the PF, since it was commented that they are both equally important. The Principal Component Analysis will be used for all the valid and optimal designs, separately. This is done in order to reduce the dimensionality of 44 parameters, listed in Table 4.8, while capturing more than

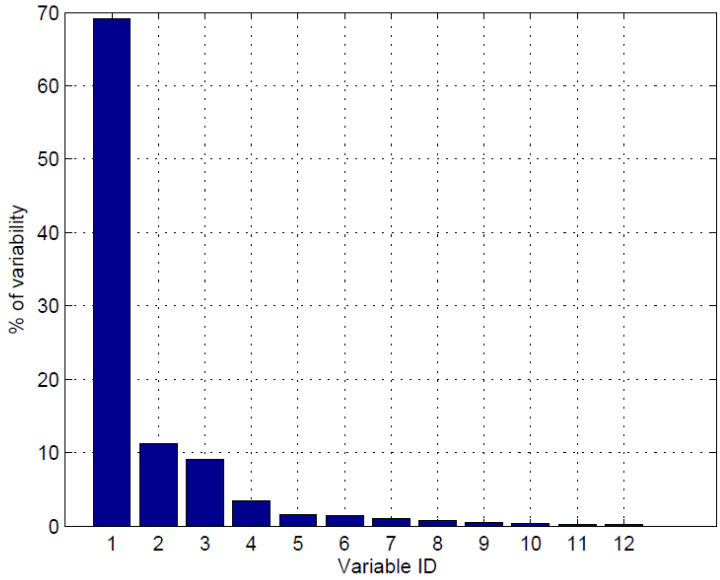
99% of the variability. The results are depicted in Fig. 4.14. Obviously, the first component of the set (i.e., the altitude of the first segment in the climb phase) is by far the most significant since it accounts for more than 65% of the problem's variability. This parameter corresponds to the first altitude value and it contributes to the first and second segment of the climb phase. The rest of the parameters are less important in decreasing order. More specifically, the first 12 variables from HISTORY account for 99% variability, whereas the top 5 of the optimal set account for 99.9%. Conversely, for the PF, the second parameter gained importance, but the importance of the third one was reduced, as shown in Fig. 4.14b. So, resolving accordingly the first two points during the climb phase will heavily affect all of the objectives.

Among the valid and optimal solutions, a number of them were selected in order to demonstrate the practical progress of the optimisation process. This informs the user how each performance criterion affects the shape of the trajectory. The arbitrary trajectory represents the initial solution, where the optimiser started from, but will not be used in the discussion, as it was only specified to start the optimisation process. Since three objectives are optimised, one set of designs that includes the minimum of each objective will be selected, too. For the NOx objective, three solutions were found that correspond to the same performance set, but they only differ at the last the climb phase altitude parameter and the 14th climb phase speed parameter. Without loss of generality, by sorting these solutions in ascending order, the median was chosen. The performance of solutions is depicted in Fig. 4.15, so as to demonstrate that at least one objective of each solution behaves better than the other solutions, which proves that employing optimisation techniques is successful. Then, the minimum time solution is the only solution that improves time by 7%, but at the same time delivers worse performance for fuel and NOx. All the other solutions improve all of the objectives, especially fuel, followed by time and then NOx. Practically, the optimisation process delivered environmentally-friendly solutions.

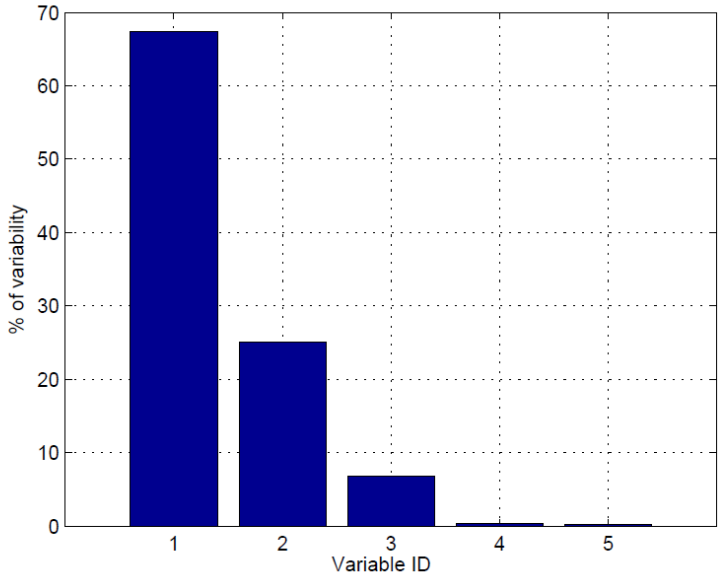
#### 4.1.2.6 Analysing Aircraft Trajectories

The last part will visualise and discuss the revealed (and optimal) trajectories, whose specification is listed 4.8. Effectively, a trajectory is a combination of points, but for ease of illustration, the altitude and speed components are separated. So, each trajectory is represented by an altitude and a speed profile that combine all the three main phases of the flight, see Fig. 4.16 and 4.17. Also, for comparison purposes, all different trajectories are illustrated in the same figure.

The altitude profiles present a lot of similarities. First, within the terminal manoeuvring area, that is an airspace control area that surrounds the airport, the trajectories are almost identical. This is because the departures and arrivals flight




(a) Parameters from HISTORY accounting for 99.2513% variability (first 12 parameters)




(b) Parameters from the PF accounting for 99.9250% variability (first 5 parameters)

Figure 4.14: Comparing the Variability from HISTORY and PF with Reference to Tables 4.5 and 4.8

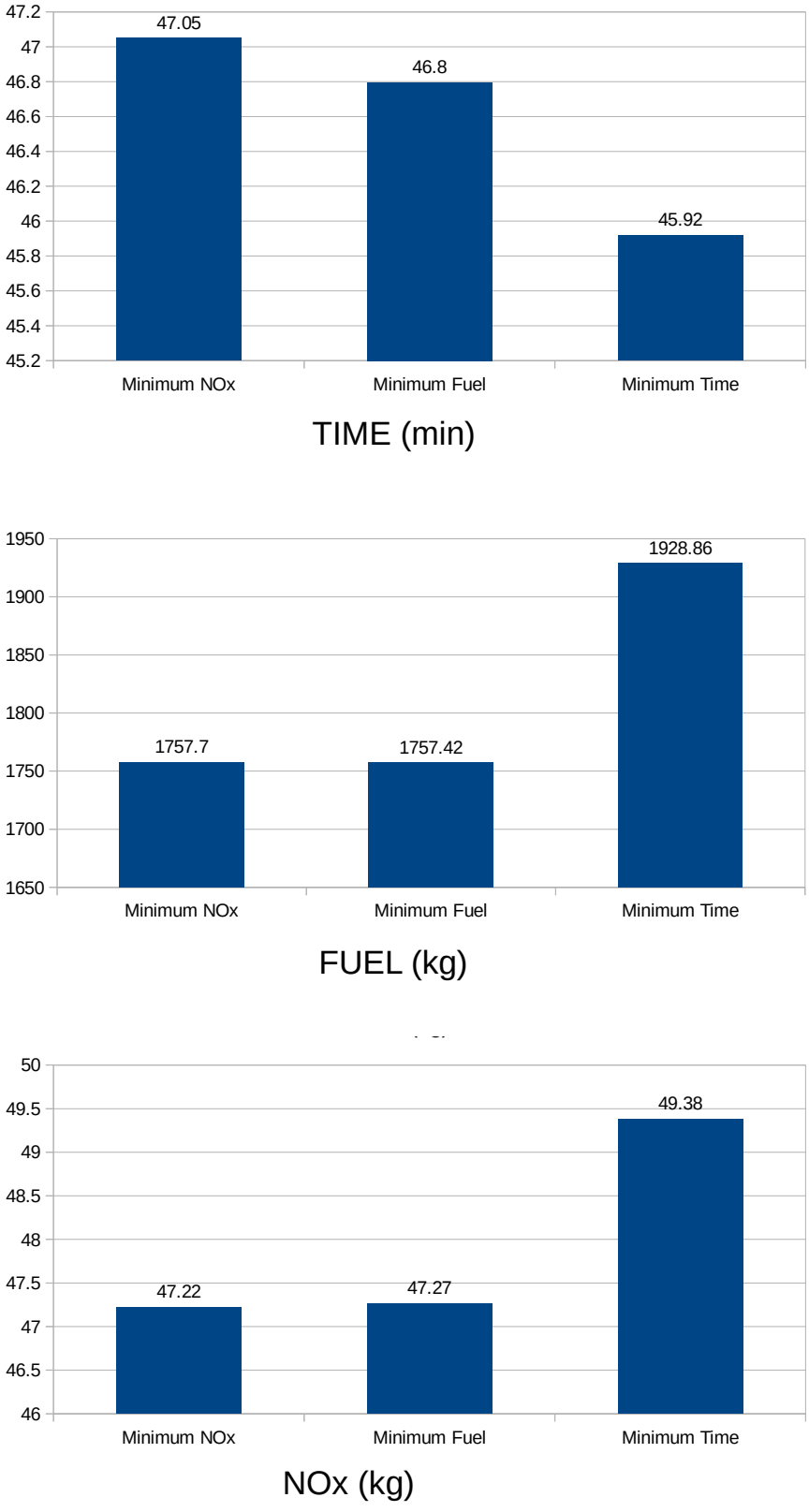



Figure 4.15: Relative Objectives' Improvement

instruction charts given to aircraft operators, called standard instrument departure and standard terminal arrival route, respectively, have very strict bounds. Hence, there is less flexibility for any modifications and much similarity is expected at both ends of the trajectory. Sometimes, the aircraft must pass exactly from a certain point at the right speed. Since each trajectory has a combination of different points, the ground distance travelled will not be exactly the same.

The altitude profile description follows. No stepped climb phase was observed in any trajectory. Right after the end of the terminal manoeuvring area, the aircraft slightly descents, and then keeps climbing until it reaches the top of climb. However, this would violate ATC restrictions, because the altitude should continuously increase. Thereafter, it maintains the same flight level and speed throughout the cruise phase until the top of descent, where it starts to descend. The rest of the climb phase, up to 42 nautical miles of ground distance, is almost the same. However, depending on the position of the top of climb altitude, some aircraft fly longer during the climb phase. The length of the climb phase mode is almost the same for every case. Interestingly enough, the trajectories for minimum NOx and minimum fuel consumption share the same cruise phase altitude. Only the minimum time trajectory has a top of climb later than the other two extreme trajectories, who also have the same top of climb, but a different top of descent. In fact, the minimum fuel trajectory is very similar to the minimum NOx trajectory. Finally, none of the discovered trajectories follows the cruise-climb practice trend, which is supposed to be the most optimal way according to modern aviation practices.

There is an obvious diversity in the speed profiles, as shown in Fig. 4.17. Only within the terminal manoeuvring area and between 25 and 40 nautical miles, the speeds are about the same. Lower emissions for minimum NOx are achieved by flying at a slower speed during the cruise phase and the descent phase. Following its definition, the minimum time trajectory has the highest speed in the cruise phase, which exceeds the second fastest by 0.068 Mach and it is the only one that does not increase speed after the end of the cruise phase. Besides the minimum time trajectory, another common trend is at the end of each cruise phase, where a surge in speed is noted. Minimum fuel seems to behave well in terms of fuel consumption, it maintains the speed from the top of climb throughout the cruise. It only increases the speed before starting to descend. Minimum NOx follows the same trend, but travels at a slower speed and remains at the cruise phase slightly longer than the minimum-fuel flight path.

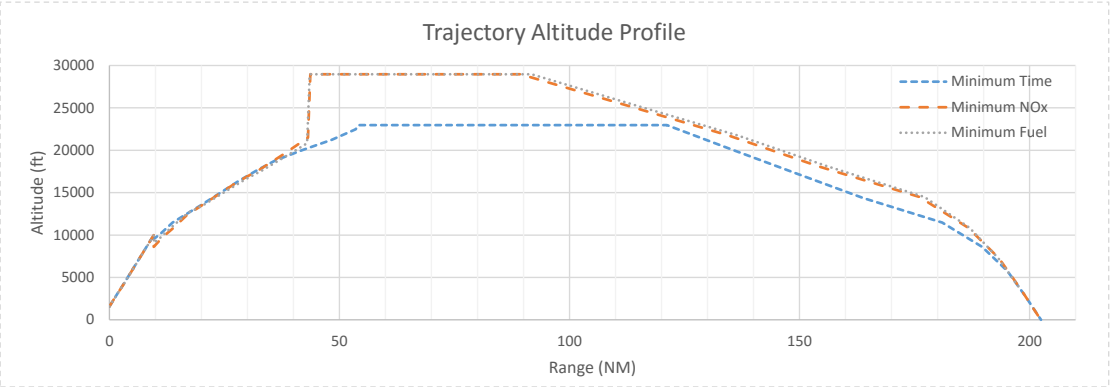



Figure 4.16: Comparing the Trajectory Altitude Profiles

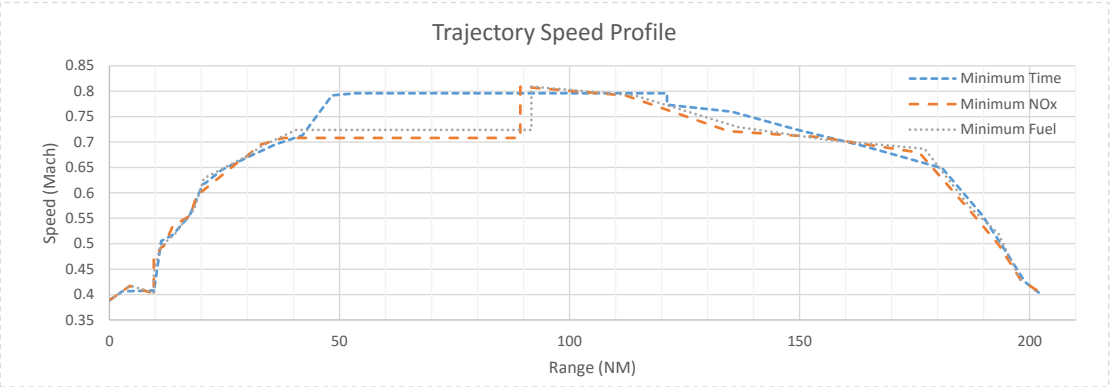



Figure 4.17: Comparing the Trajectory Speed Profiles

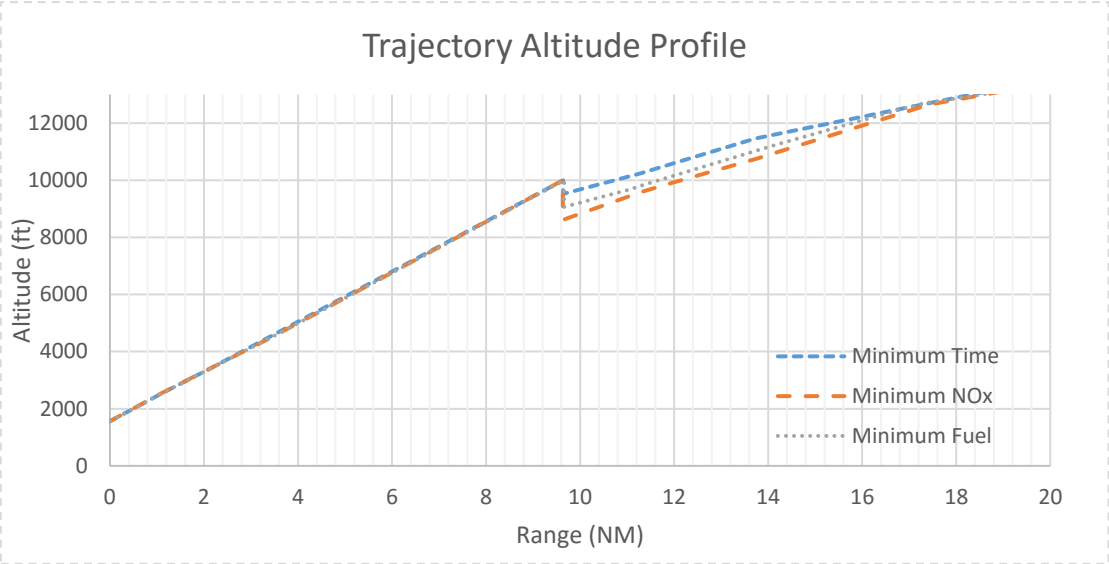



Figure 4.18: Zooming trajectory altitude profiles

The shape of the revealed optimal trajectories is in satisfactory agreement with state-of-the-art studies. More specifically, by investigating the optimal trajectories in terms of fuel, time and in [273] for the climb phase, similar trends were discovered. The minimum time trajectory flies at a higher speed and following the same altitude profiled with the other optimal trajectories during the climb phase, until the top of climb. Then, during the cruise phase, the aircraft flies at a higher speed, but at a lower altitude. The minimum fuel and minimum $NO_x$ trajectories fly higher, because this is related to the specific fuel consumption of the engine, which is better at a higher altitude, for this particular flight envelope. The shape of the descent phase obtained is similar to the optimal vertical flight paths under continuous descent approach that were discovered in [274], where a detailed mathematical formulation can also be found. Also, in the same reference, in the mathematical formulation, it is demonstrated that the objectives of $NO_x$ emissions and fuel consumption are related with respect to the speed of the aircraft during the cruise phase, which is the same in this case, too. Hence similarities in the profiles of fuel-optimal and $NO_x$-optimal trajectories are expected, and they are slightly different because of the coefficients involved in the mathematical formulation. The same trends were also discussed in [171], where it was also mentioned that the minimum $NO_x$ trajectories specify a constant turbine entry temperature, which is also related to the specific fuel consumption and will be further investigated in the future.

#### 4.1.2.7 Identified Issues

The methodology and results for optimal trajectories in terms of time, fuel consumption and NOx emissions were presented, where reductions (see Fig. 4.13) in time, fuel and NOx have been achieved, towards reducing the environmental impact of a medium-haul aircraft, too. The optimiser searched through a highly constrained decision space with 44 variables that describe climb-cruise-descent, due to the operational and ATM constraints, only 1.35% valid trajectories were found out of 68,000 evaluations. Starting from an arbitrary trajectory, among the optimal solutions, the improvement is 1.13 minutes, 171.44 kg of fuel and 2.15 kg of NOx. However, for prototyping purposes, the revealed trajectories did not consider strict climb rate rules with respect to ATC restrictions, where the aircraft should continuously ascend during climbing. This could be resolved by appropriately specifying a tighter flight envelope and by introducing appropriate constraints for the climb rate such as the altitude of the previous point in the climb phase should be lower than the altitude of the next point. Methods for speeding-up the optimisation process either by changing the configuration settings, algorithmic behaviour or problem description have been discussed. Differences among optimal trajectories were highlighted.

As the optimisation search continues the problem changes from 3-objectives to 2-objectives. Initially, all the objectives were conflicting with each other. While the optimiser was discovering better decision points, the fuel and NOx objective started to follow a similar trend.

More specifically, the knowledge extraction mechanisms should consider both optimal sets and HISTORY. Although, it is proven that all of the objectives generally conflict with each other, by inspecting the HISTORY, just by observing the optimal trade-off this relation is not revealed. The most important parameters have been identified, too. The first altitude value of the initial segments heavily affects the performance of the trajectory, the progress of optimisation search and, hence, the shape of the optimal trade-off. This was expected since all the following segments depend on the first one. In fact, altitude values affect the overall performance of the trajectories.

Trends for optimal trajectories have been identified. By analysing the extreme trajectories for each objective, two common trends were revealed; first, regarding the altitude profile, the climb phase is very similar to all the extreme optimal solutions before entering the cruise phase. Second, at the beginning of the descent phase, all the types of trajectories reach their top speed for a short period of time before entering into the continuous descent mode. The speed during the cruise phase was different for each of the trajectories, but only two levels of altitude were revealed to be optimal. The optimal trajectories could be simulated with tools of higher fidelity for increased accuracy, which will lead to multi-fidelity optimisation

case studies.

## 4.2 Micro-Mixing Optimisation of Microreactor

### 4.2.1 Problem Description

The microreactor is a device that could be used for a hydrogen micromix combustor, as described in [34]. One of the problems associated with facilitating lean combustion, to reduce flame temperatures (in an effort to reduce $NO_x$ emissions) is the relatively narrow stability limits of kerosene, which could result in problems with lean blowout and combustion instabilities. Hydrogen as a fuel is a promising candidate in this context, as it has much larger stability limits and therefore lean combustion is possible without approaching lean blowout limits. It has been demonstrated that it may be possible to eliminate the dilution zone and control the combustor outlet radial and circumferential ("clocking") profiles by customising the fuel distribution in the injectors. This will yield benefits not only for turbine life and performance but also result in a shorter combustor. When designing a microreactor, the shape of devices directly affects the efficiency of the device in terms of mixing capability and pressure losses, as also studied in [137, 275]. The optimisation of the trade-off between these two objectives will be investigated in this section. By investigating alternative geometrical configurations, it is expected to assist towards maturing this technology.

It was demonstrated experimentally that even a relatively small change in the geometry (i.e., baffle plate) of a microreactor, can cause dramatic changes in the structure of the flow [153]. Following the demonstration of simulating the flow in a microreactor, in section 3.3.7, the flow solver is integrated into MOTS2, so as to carry out micro-mixing optimisation. Also, this application aims at demonstrating the suitability of MOTS2, as an optimiser, to be used in design optimisation cases. In terms of technology integration, via this application, the ability of MOTS2 to interface with applications that run on GPU is demonstrated below.

Being able to perform micro-mixing at the minimum possible pressure losses reflects the performance of the device, as less energy would be required to be used (because of reduced losses) to generate energy (through the mixing of reactant agents). The chemical reactions are not currently resolved.

The technological means to study devices at the micro-scale level for future applications in air transport are presented below. This is the long-term approach to minimise the impact on the environment, as suggested in section 1.1. The aim is to demonstrate an alternative method that will enable the users to carry out research on devices that will minimise gaseous emissions.

Here, MOTS2 is combined with the flow solver based on LBM, which were presented in sections 3.1 and 3.3, respectively. By using the LBM flow solver, several geometrical arrangements of a microreactor are simulated to study the mixing capability of the device and to investigate related flow features. Furthermore, the

process will be accelerated by employing the high computational power of GPUs in order to speed up the computational engineering design process with respect to environmental issues. This is expected to integrate with computational methods and experimental methods across many development processes for environmental issues.

The original geometry presented in [45] has been used here, too. It is a microreactor that can mix two different liquids coming from separate chambers into a common chamber. The purpose of the device is to mix these reactants by generating turbulence. A multi-holed baffle plate controls the speed of the fluid and creates turbulence, which increases the mixing capability. As was noted in the same reference, the shape of the holes is extremely important for the mixing. Although the design of the geometry is relatively simple, it could be used as a simple well-established benchmark test case that can be used in real-world applications, as it was experimentally tested in [45].

### 4.2.2 Methodology

The specific instance of the methodology is depicted in Fig. 4.19.

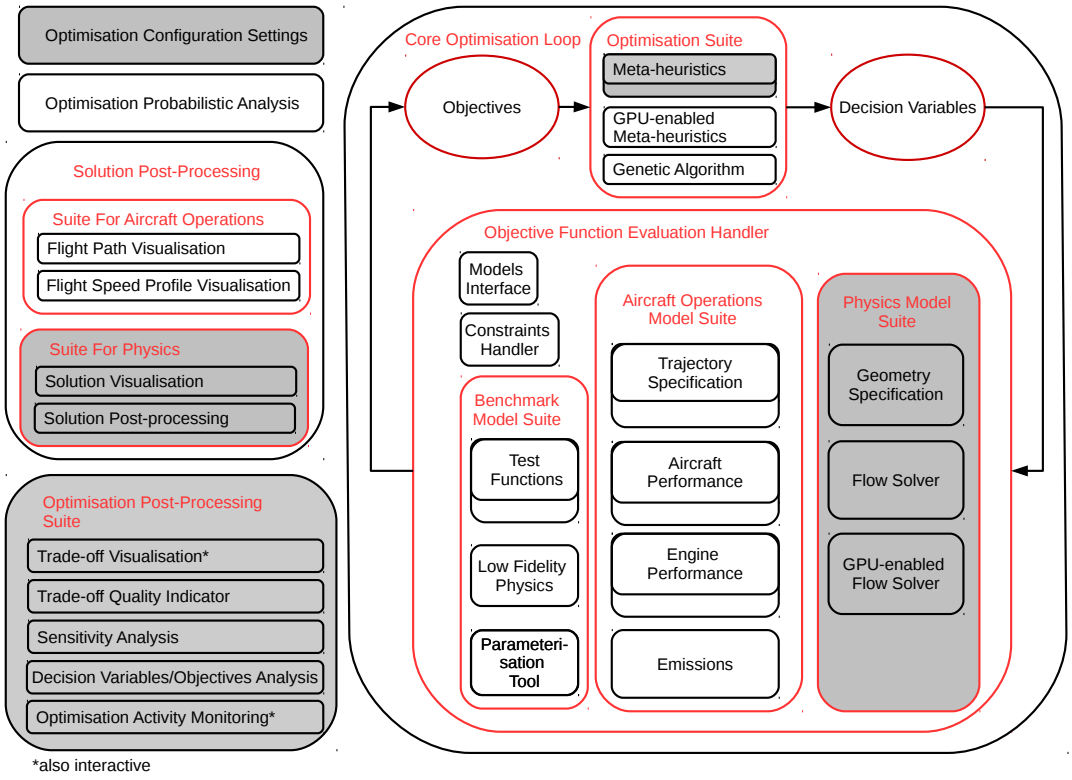



Figure 4.19: Methodology for Comparing MOTS2 and GPU-LBM against the Original Multi-Objective Tabu Search and the Original LBM

In micro-scale problems the lattice Boltzmann equation is a more effective simulation technique than the NS equations for a series of reasons [94, 95]:

- First and foremost, the NS equations model a fluid's behaviour at the macroscopic level under the hypothesis of the continuum, whereas LBM is applied at the microscopic level, thus closer to the dimensions of the problem. This is related to the $K_n$, which is introduced later, see (2.1) (on page 17), where the diameter of the device is selected for the characteristic length.

- Implementing complex solid surfaces and handling boundary conditions [2] is easy. LBM is mainly used in porous media, where very sophisticated geometrical arrangements occur. In the open literature, instances of the latter range from advective and diffusive transport to realistic cases. Moreover, there are several applications such as multiphase flows and/or multi-fluid problems, investigating the effects of cavitation, too.

[2] especially in cases where solid nodes reside within the media, the overall computational cost can be decreased significantly

- By definition, LBM does not involve highly dependent calculations; thus, computations could be carried out very efficiently in parallel and many schemes and techniques could be easily applied, delivering various levels of performance and fidelity.

As presented in [64], the concept of combining a Multi-Objective Tabu-Search optimiser and a LBM code has been tried and has delivered satisfactory results. However, here, the intention is to present an alternative approach to combining tools of the same class to speed up the computational engineering design process and to provide stakeholders higher benefit with lower cost. In addition, the results from this approach will be compared against the previous study, for convenience purposes. However, as the code was compared against a single case, the results of the optimisation process should be handled with caution.

This work combines MOTS2 and GPU-LBM to optimise the flow within a device that performs mixing at micro-scale. This could be used in environmentally-friendly applications and extends the work carried out in [45, 46, 48, 49, 64, 248]. Micro-mixing is fundamental when generating energy with alternative fuels, such as hydrogen, as discussed in section 2.1. In addition, LBM could be extended to simulate reacting flows [276, 277] and combusting flows [278, 279], so as to capture more physics of real-world applications. Resolving the mixing capability of the device within relatively short time-intervals is the first step to start designing environmentally-friendly devices through design optimisation, as was carried out in [137]. The proposed methodology can be used to revisit the technology of using hydrogen or other fuels for energy generation purposes on aircraft and to influence, in general, the design process of new devices for future needs, too. This is expected to enable more stakeholders to study similar devices and to prompt more research.

The optimisation of the reaction mixing is approached in two stages, as follows. First, a black box system has to be generated in order to evaluate the decision variables to specify the geometrical arrangement of the microreactor and to characterise its environmental performance. Second, a direct-search MOO algorithm is used to optimise the aforementioned black box system. The optimiser and the flow solver are linked together via a simple interfacing mechanism, where files are exchanged in both directions. Geometrical constraints are imposed to adhere to the concept of the microreactor as presented in [45]. The optimisation process will begin from the geometry specified in the same reference; it will be considered the baseline and any findings will be expressed relative to that one. The revealed trade-offs and selected designs will be compared against the ones from the earlier study in [64]. The modular structure of the methodology will allow one to scale the original problem in terms of the size of the computational domain, to employ more powerful HPC infrastructure, to expand the physics of the code, to integrate with other system(s) and to effortlessly alter either the target geometry or the

optimiser or both. This is a very flexible strategy, which is expected to positively influence CFD and design optimisation.

Although several combinations of software and hardware could be used, as nowadays there are many different tools that serve the same purpose, the ones used here serve the purpose of accelerating the computational engineering design process. Regarding the part of the application where the flow is resolved, there are several CFD tools for either commercial or research purposes. However, very few of them harness the computational speed of GPUs at the lowest possible economic cost and have been proven to work on real-world applications. The majority of CFD tools solve the NS equations by using a finite volumes numerical scheme, which is computationally very expensive and requires several CPUs in order to deliver results within sensible time frames. When thinking of a bigger system with multiple principles, conventional CFD tools that solve the NS equations via a finite volumes numerical scheme take more effort to prepare and to run a case. For instance, generating a computational mesh for a target geometry and setting up complicated boundary conditions could potentially be very time-consuming tasks that significantly affect the duration of the whole process, whereas both are easily resolved when employing LBM. Carrying out simulations on the previously mentioned platform has been done for many years; it is common knowledge that the computational and economic cost(s) of running and maintaining such platforms increase very quickly and tend to be impractical for large scale projects and/or when finer detail and precision are required. Regarding the optimisation algorithm, several types of optimisers could be employed here. However, features of metaheuristics, as discussed in section 2.7, seem to be more appropriate for complicated real-world cases and have the potential to further shorten the duration of a project, as will be explained in sub-subsections 5.5.1.3 and 5.5.2.2. Given the relatively short time frame, metaheuristics might not be able to find the global optimal solution, but they can discover a competitive solution [55]. Furthermore, the combination of Multi-Objective Tabu-Search and LBM has been demonstrated in [64] and this section builds on that work. This methodology aims to exploit alternative software and hardware to bridge the gap between academic research and industrial production needs so as to provide the means of developing greener applications.

Since the intention is to provide the means to optimise and to study environmentally-friendly devices, data will be collected in two stages. First, all the data related to the optimisation process will be used to comment on the effectiveness of the optimiser and how it moved in decision space. These will mainly involve configuration data, trade-off analysis, monitor data and sensitivity analysis of the decision variables that participate in the PF. Second, after identifying optimal results, their corresponding data from the LBM simulation will be used to analyse

the features that contribute to their optimal environmental performance. More importantly, the shape of the baffle plate and the flow field will characterise different geometrical arrangements that adhere to the concept of the microreactor.

A few components are required for this application. First, the concept of the geometric design is expressed as an array of obstacles for each LBM node. Second, physical GPU devices should be available to run the simulation via LBM. MOTS2 is used to optimise the concept of the geometric design of a microreactor, whose configuration settings are specified before running. For the optimisation process, software that can plot 2D and multi-dimensional data will be required so as to assess the progress and overall performance. Additional fluid properties should be specified in the configuration settings. Boundary conditions should be set before running the simulation. Then, the simulation will run on any GPU and the results will be exported. Any software that can read the resolved flow field will also be required for post-processing purposes. This could be used to analyse and to visualise the final results.

The methodology to optimise the environmental performance of a microreactor suggests the following steps:

1. Prepare the flow solver to be automatically used on different configurations.
   (a) Prepare the target geometry by specifying in 3D the empty and occupied areas of the domain in binary representation as an external file.
   (b) Configure the solver with information about the flow and geometry, as described in 3.3.5. This will be generated automatically before every simulation.
   (c) Set boundary conditions within the code.
2. Link MOTS2 to GPU-LBM so as to start a new simulation for every different decision vector, as proposed by the optimiser.
3. Configure MOTS2 by specifying configuration settings and by setting up the optimisation problem.
4. Run the optimisation process starting from the design that has been experimentally tested.
5. The flow field of every design will be simulated by running GPU-LBM.
6. Monitor the progress of the optimisation process.
7. Obtain and analyse the PF, memories and monitor data from the optimisation process.

8. Select optimal designs from the finally revealed PF.
9. Assess the performance of the PF.
10. Visualise the simulated flow field of the selected designs.
11. Compare and explain the geometrical and flow features of the selected designs.

### 4.2.3 Preparing the Multi-Objective Optimisation Process of the Reaction Mixing

The MOO problem has two objectives and two structural constraints. Both objectives are additionally computed by using the GPU-LBM flow solver, described in section 3.3, after the flow has been resolved. They determine the performance of the micro-device and reflect the conflict between environmental targets and operational targets. The structural constraints exist to instruct the optimiser to search for sensible arrangements that adhere to the concept of the microreactor.

For ease of assessment, the same problem formulation and practices from [64] are followed. In total, three parameters are going to alter the performance (i.e., the fitness of the objectives, in terms of optimisation) of the microreactor and are specified externally, as described in subsection 3.3.5. Following subsection 3.3.7, the case is modelled based on the geometry shape in Fig. 3.25 and the speed of the flow for inner and outer holes by altering $Re$. The geometry of the baffle plate is geometrically parameterised, as shown in Fig. 3.26, by varying its inner hole radius ($r$) and spacing between two holes ($s$). For consistency, the same name convention is used. An additional module was implemented in the LBM code to care for the shape of the baffle plate. The middle hole has a constant size.

A brief description of the black box system that evaluates the environmental performance of the microreactor follows; again, the evaluation of the decision variables is treated as a black box for convenience through a very simplistic file exchanging scheme. The GPU-LBM works as a monolithic application that requires a number of configuration files that manage the behaviour of the flow solver and describe the computational domain. The file that describes the geometrical arrangement of the baffle plate and the tube size of the microreactor are required by the black box. The other files required by the GPU-LBM remain constant. All these are read by the code, which also constructs the same computational domain within the GPU memory. Then, the LBM is executed on the GPU and the final converged solution is saved, which is also converted to meaningful units. At the end, the aforementioned solution of the above computational domain is exported to a single file. The environmental performance indices are further computed from the solved computational domain and the calculated quantities are exported to a separate file.

Here, the black box evaluation is encapsulated within the Evaluation Manager of MOTS2, which actually acts as a wrapper. More precisely, whenever a new decision vector is to be evaluated, the Evaluation Manager module prepares the configuration file for GPU-LBM, launches the flow solver under new settings and then collects the final computed objectives. Obviously, this will be repeated for every evaluation to simulate different geometrical arrangements of the microreactor that do not violate any of the constraints. The layout of this process is depicted in Fig. 4.20. These objectives are indications of the losses of the device and of the mixing capability of the device.

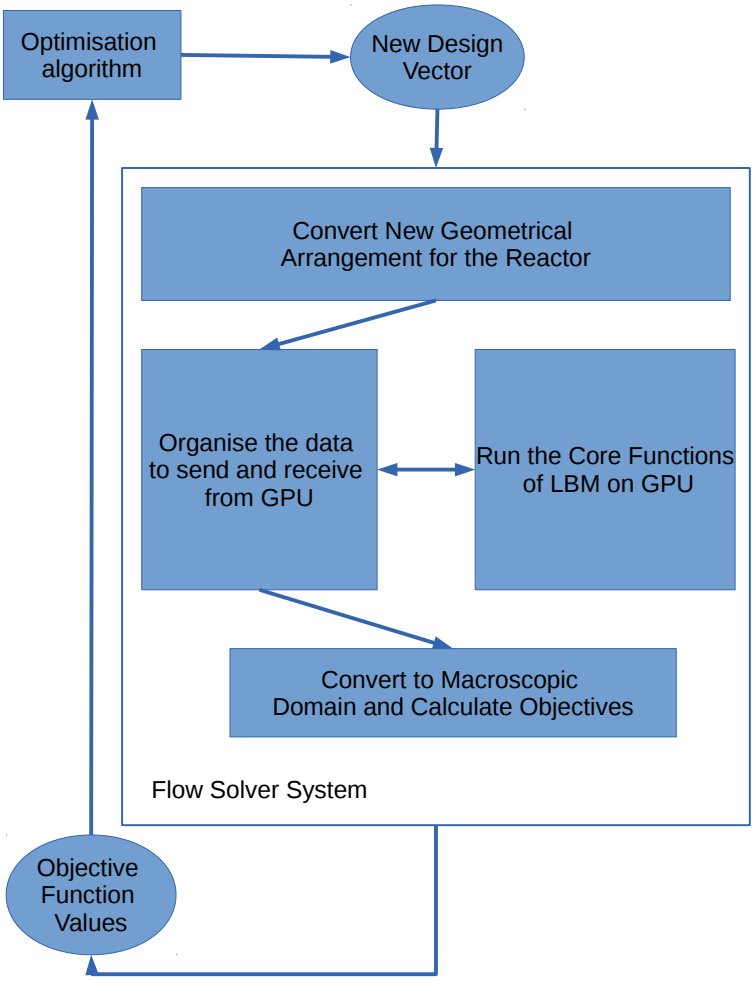



Figure 4.20: Layout of Optimisation Process on Microreactor Mixing

The first objective is the difference of total pressure ($\Delta P_0$) between the inlet and the outlet of the domain. This is the difference between two regions of the geometry, where the total pressure ($P_0$) at the downstream end of the device is subtracted from the $P_0$ at the inlet. The lower the $\Delta P_0$ is the lower the losses of the microreactor are. In practice, the lower the pressure difference, the less the losses of the microreactor will be, which means the system will lose less energy for the same quantity of fuel. Therefore, this quantity has to be minimised. The numerical value of $\Delta P_0$ is simply obtained by calculating the macroscopic density from the microscopic densities of the data structures of LBM, as shown in (3.4).

The second objective is the magnitude of vorticity ($\omega$), which is a vector field that describes the tendency of a fluid to rotate. This is related to the mixing capability of the device, where better mixing could be related to higher levels of energy released, and could reduce the amount of fuel required to generate target levels of energy. In fact, it is a vector whose components are defined by using the components of the velocity defined in (3.5) as follows:

$$\omega_x = \frac{\partial u_z}{\partial y} - \frac{\partial u_y}{\partial z} \tag{4.1}$$

$$\omega_y = \frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x} \tag{4.2}$$

$$\omega_z = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \tag{4.3}$$

Moreover, the magnitude of $\omega$ is defined as:

$$\|\omega\| = \sqrt{{\omega_x}^2 + {\omega_y}^2 + {\omega_z}^2} \tag{4.4}$$

The higher the magnitude of vorticity is the better the mixing capability of the micro-device. Therefore, this quantity needs to be maximised.

The optimisation problem is formulated as follows:

$$\begin{aligned} \underset{r,Re,s}{\text{minimise}} \quad & -\frac{\|\omega(r,Re,s)\|}{\|\omega_{datum}\|}, \frac{\Delta P_0(r,Re,s)}{\Delta P_{0,datum}} \\ \text{subject to} \quad & 2 \cdot r + 1 < s < 70 - r - 1 \\ & 5 \leq r \leq 11 \\ & 1 \leq Re \leq 200 \\ & 11 \leq s \leq 29 \end{aligned} \tag{4.5}$$

The objectives are normalised so as to more effectively guide the optimisation search, as the numerical accuracy will be the same for both of them. In addition,

it is easier for the user to appreciate any improvement from the baseline geometry and to maintain direct comparison between this study and [64], where this concept was originally introduced. It should be remembered that the objective is to maximise the normalised vorticity magnitude, which is equivalent to minimising the normalised negative vorticity magnitude. The double inequality is a geometrical constraint, where the radius of the outer holes of the baffle plate should be kept small in order to propose a feasible geometrical arrangement; this is partly for manufacturing purposes and to adhere to the original concept of six holes on the baffle plate for one chamber and one centre hole for the other chamber. The range of variability of $Re$ is specified in order to search for devices that operate in the laminar flow regime. This is due to the fact that laminar flow will introduce lower friction (viscous) losses compared to turbulent flow.

The configuration settings were specified based on the experience of the author. The number of evaluations was set to a very high value, because the computational cost of running on GPUs is now affordable, which is not the case in [64], and will increase the confidence in the final PF. In general, the exploration and exploitation settings have been set to a very slow pace (compared to the other optimisation configuration settings in earlier cases); despite the small number of variables and constraints, the MOO problem is expected to be complicated because of the nature of the objective functions. Gradually, the optimiser will converge to promising regions in terms of objectives and will employ any of the additional strategies if it is trapped in local minima. Furthermore, the starting point of the optimisation search was selected to be the decision vector whose decision variables describe the geometrical arrangement of the device, whose performance was experimentally tested in [45]. In fact, this is good practice because the optimisation process will start from a point of well-known performance. The step for each decision variable has been set following the suggestion in [64]. More precisely, the search step of the variables for $r$, $Re$ and $s$ is 1 $lu$, 10 and 1 $lu$, respectively, but in the optimisation settings express the same values as a percentage of the range of variables, as defined in equation (4.5). Then, the reference point for the hypervolume has been set to a high value that will capture any wide span between the extreme points. All these are listed in Tables 4.10 and 4.11 with reference to equation (4.5).

Table 4.10: MOTS2 Configuration Settings for Microreactor Optimisation

| **Parameter Description** | **Value** |
|---|---|
| Call diversification move after # non-improvements | 10 |
| Call intensification move after # non-improvements | 5 |
| Reduce the search step size after # non-improvements | 15 |
| Initial step sizes (as % of variable range from the constraints in (4.5)) | r: 0.1666666<br>Re: 0.050251256<br>s: 0.055555556 |
| Step sizes are multiplied by this factor at restart | 0.5 |
| Number of points randomly sampled | 3 |
| # of variables | 3 |
| # of objectives | 2 |
| # of objective function evaluations | 3000 |
| Divide search space into # regions | 4 |
| Size of Tabu Memory | 6 |
| Maximum Improvements | 200 |
| Maximum Duplicates Allowed | 300 |

Table 4.11: MOTS2 Configuration Settings to Optimise the Microreactor

| | |
|---|---|
| Datum Decision vector | r:7, Re:100, s:24.5 |
| Reference Points for hypervolume Indicator | 20, 20 |

### 4.2.4 Optimising Environmentally-friendly devices - Results and Discussion

By applying again the Parallel Coordinates Projection on the decision space of the problem one can discover the areas of decision space that the optimiser has visited. The projection of all the feasible decision points of HISTORY is depicted as shown in Fig. 4.21, where the user can realise where the search was focused. The feasible decision space was covered decently. By focusing on the points visited (especially between integer numbers on the $r$ axis), it possible to see that the search step was reduced once, which is clearer in Fig. 4.37. However, although a wide range was designated for $s$, the optimisation search focused on the region above 17.5 $lu$, and the refined search step covered an even narrower range (above 21.5). Regarding

$r$, more points towards the upper bound were investigated and the search step refinement started from 7.5 $lu$ onwards. Relatively to the other two variables, $Re$ seems to be traversed more uniformly (including the step refinement, which is visible across the axis), but slightly more points were visited in the lower half of the range, closer to the lower bound.

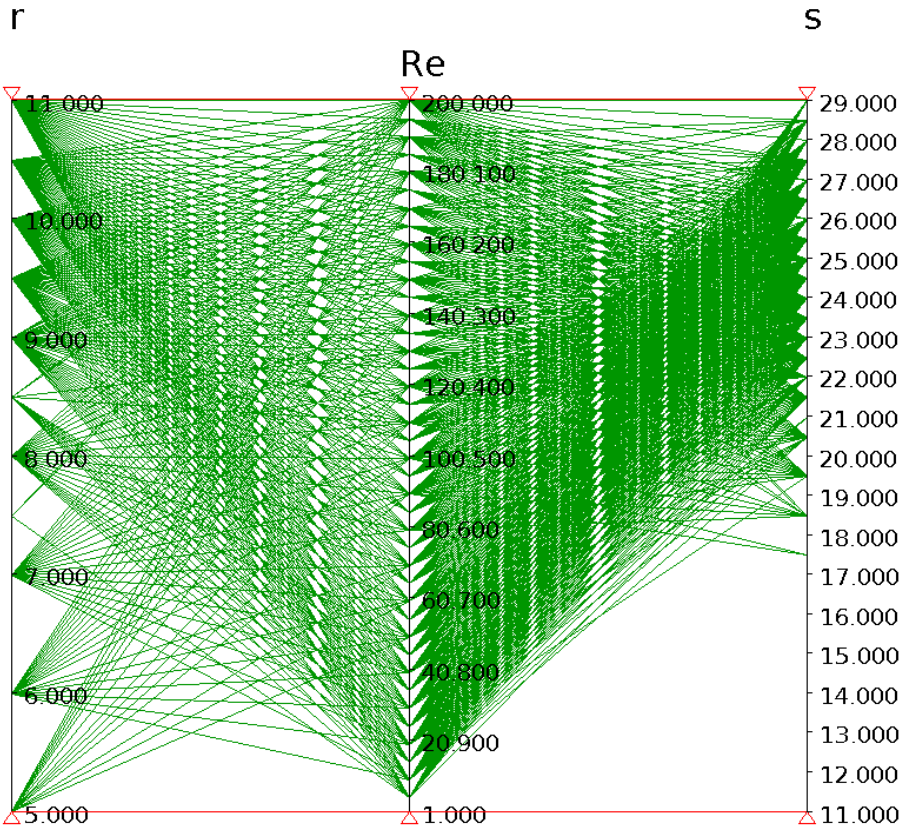



Figure 4.21: Coverage of Decision Space (from HISTORY)

The relationship between the decision variables and the objective space is illustrated in Fig. 4.22. This is an extension of Fig. 4.21, where the axes of objectives were added so as to provide further insight into the relationship. In addition, the axis of $s$ was rescaled so as to focus only on the discovered area of the optimum decision space. Here, it is easier to notice that the area between 23 and 26 $lu$ in the $s$ axis was visited more, which is also shown in Fig. 4.25. Again, the lines between the axes of objectives cross, which means that they are conflicting. In addition, the spread of designs along the axis of normalised vorticity magnitude is more uniform compared to the axis of normalised difference in total pressure, where most

of the designs are concentrated at the lower one-fourth. As is also explained next in greater detail, when it comes to the optimal behaviour, as discovered so far, the changes in total pressure of the microreactor present a larger gap. Consequently, it is expected to be more difficult to achieve the target behaviour in total pressure.

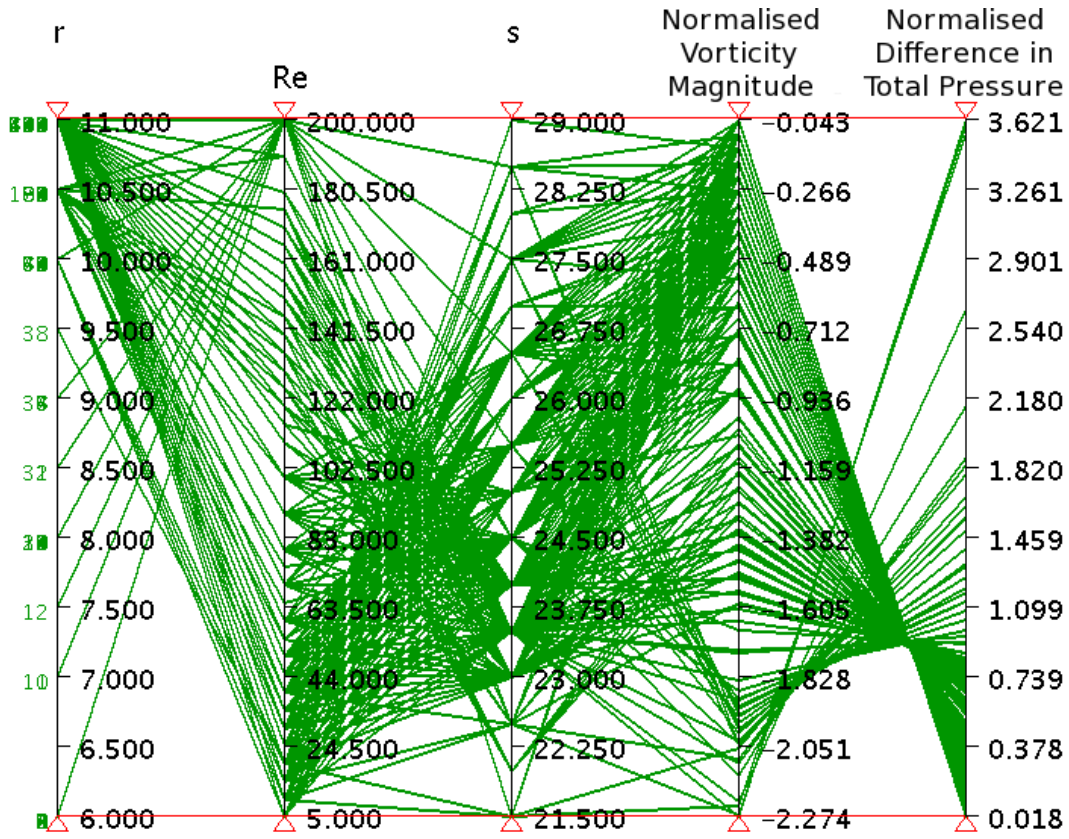



Figure 4.22: Optimum Decision Space and Objective Space (from MTM)

Further statistics are provided in Fig. 4.23, Fig. 4.24 and Fig. 4.25 for each decision variable separately. It is clear that most of the evaluated designs had big outer holes on the baffle plate, which can be considered as a trend. Regarding *Re*, more extreme values were visited; because of this trend, its impact on objective space will be investigated next. However, the region for the decision variable that described the radius of the outer holes of the baffle plate was never investigated for values lower than 16 *lu*s. Thus, Fig. 4.23 and 4.24 have been produced to complement this observation. The vast majority of the points live in the area where the distance of the outer holes from the centre of the tube is larger than 21 *lu*s. Moreover, in the PF the concentration of the decision points belong to the class 24-26 *lu*s. Regarding the radius of the outer holes, it is clear that there is a

significant high number of points in the range above 9 *lu*s. All this suggests that a new case could be relaunched starting from the zones of each decision variable where the concentration is the highest.

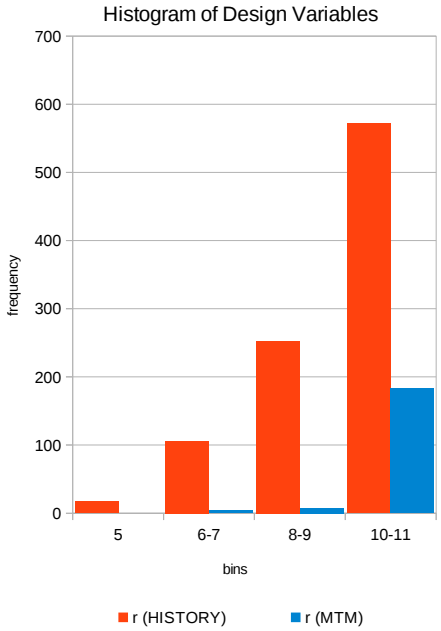



Figure 4.23: Histogram of first Decision Variables of the Optimal and Complete Dataset

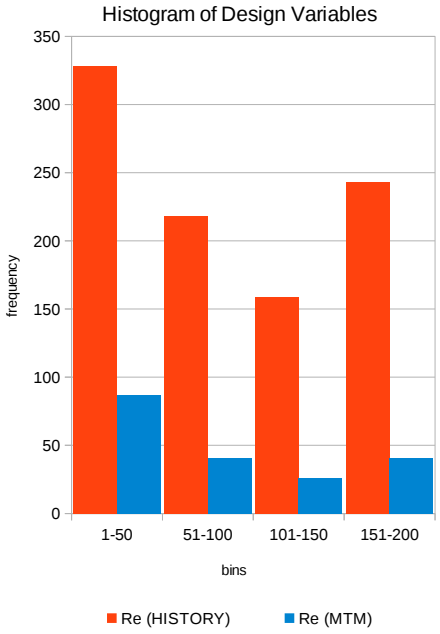



Figure 4.24: Histogram of Second Decision Variables of the Optimal and Complete Dataset

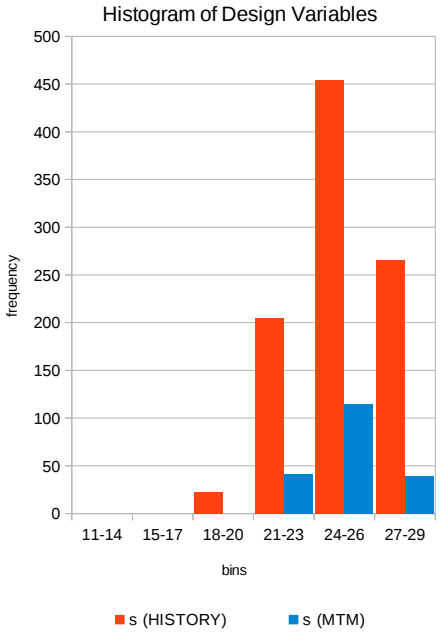



Figure 4.25: Histogram of Third Decision Variables of the Optimal and Complete Dataset

The progress of the optimiser in objective space from the beginning of the process till the end is illustrated from Fig. 4.27 to Fig. 4.32 by using the configuration settings listed in Table 4.10. First of all, it is clear that the objectives conflict with each other and, hence, this method attempts to give further insight in this regard. The evolution of the discovered objective space suggests that there were not any local minima to trap the optimiser. It is also important to mention that very few of the feasible designs are dominated by the datum design; this means that the optimiser effectively used the computational budget to find solutions that improve at least one of the datum's objectives. However, among the proposed different designs, approximately 17% were feasible and were actually evaluated through GPU-LBM simulations. These indicate the complexity of the problem and prove that employing MOTS2 was a sensible decision. By using all the available computational budget 195 designs were found in the final trade-off, as shown in Fig. 4.32.

Compared to [64], many more designs have been discovered, because the computational cost was more affordable, as described in subsection 3.3.7. The comparison between the two trade-offs is illustrated in Fig. 4.26. In order to perform a fair comparison, regarding the problem set-up (for instance, the size of the baffle plate, the lattice size etc.), exactly the same settings were used, and no sensitivity studies were performed, as this work extends the previous. The trade-off is wider, denser and further away from the datum design, again for the same reason. Clearly, many more designs were discovered that improve the mixing capability of the device (less than $-1.8$ in the horizontal axis). The new PF is also more continuous and almost reached the absolute extreme for normalised difference in total pressure. Although it is obvious (because of the span of the trade-off) in Fig. 4.26, in terms of hypervolume, the PF revealed by MOTS2 with GPU- LBM is better, as listed in Table 4.12. The reference point was specified by using the worst combination of points in the PF.

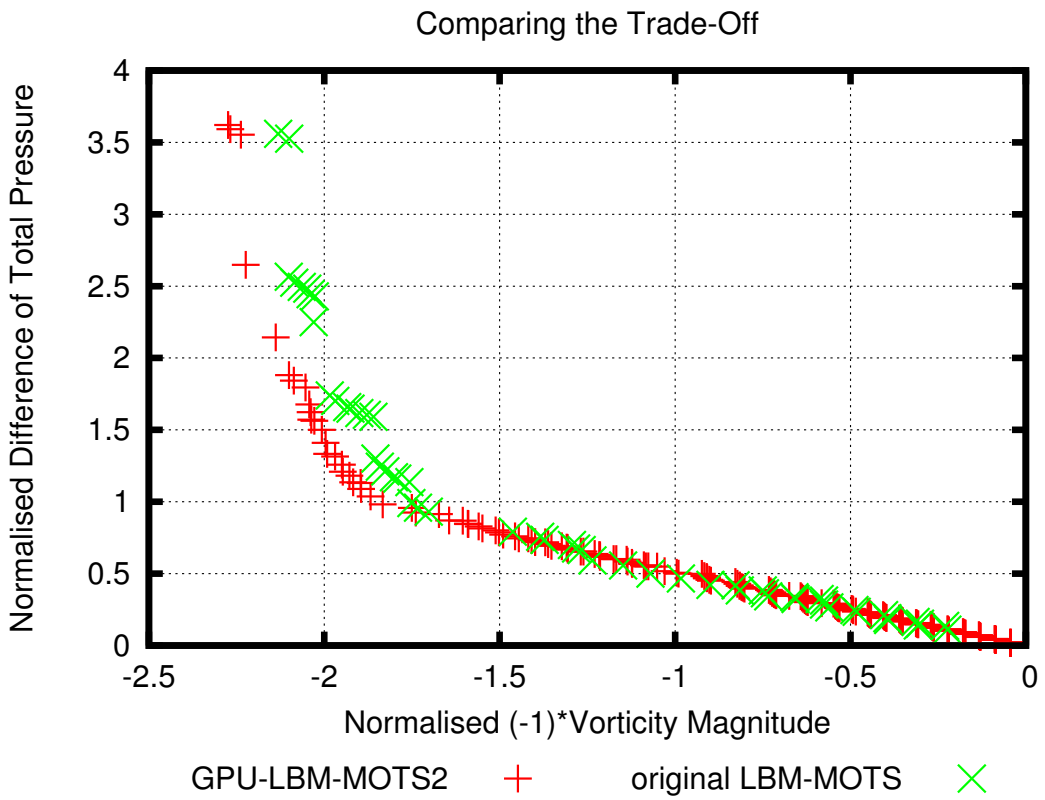



Figure 4.26: Comparing the PF between the Current Implementation and the Original one from [64]

Table 4.12: Hypervolume Indicator for Microreactor Mixing

| | **Hypervolume Indicator** |
|---|---|
| | Reference Point: { -0.0430896 , 3.62125} |
| Original multi-objective tabu search with LBM | 6.006 |
| MOTS2 with GPU- LBM | 6.338 |

The discovered designs cover a wide range of the relative performance, compared to the datum design and this could have two interpretations. Firstly, the revealed PF is very wide, which provides the user with a plethora of solutions to select from, by following a decision-making process. The range of normalised difference in total pressure is larger compared to the other objective. It means that there are more opportunities to reduce losses without changing the other principle much. This is a very useful feature because it can be directly affected by the flow

speed. Therefore, it seems easy to change and to control the losses of the device by changing a non-geometrical factor. Secondly, from one extreme to the other the normalised vorticity magnitude can be improved by a factor of approximately 2.2, whereas the normalised difference in total pressure can reach an almost perfect level with approximately less than 2% losses. This means that (in this case) aggressive mixing comes at the cost of higher losses and almost lossless operation cannot do any mixing. In a production environment, the user would be practically interested in the designs that dominate the datum design. These are the designs with a normalised negative vorticity magnitude less than -1 and with a normalised difference in total pressure less than 1. Moreover, this range appears to change linearly, which will be further discussed below.

The final PF presents two different types of behaviour. It can be separated at a normalised negative vorticity magnitude of -1.8 (or 0.98 of the normalised difference in total pressure). Above that threshold, the trade-off in this region changes linearly. Below that point, it follows an exponential trend. The latter means that for small changes in the normalised negative vorticity magnitude, towards the extreme of -2.2, the normalised difference in total pressure greatly increases, which means that solutions in that range should be avoided because the cost rapidly increases. In contrast, the linear change in the other half is very favourable because the slope is small, which means that one can select designs with very similar performance. Moreover, it was found that in the linear region of the trade-off *Re* varies in harmony with the objective of normalised vorticity magnitude. This suggests that just by changing the flow the user can achieve various levels of performance even for the same device.

Further information can be inferred by observing the sequence from Fig. 4.27 to Fig. 4.32. First of all, *Total Evaluations* encompass both feasible and infeasible designs. Within the currently considered computational budget, the more time the optimiser spent on the problem, the more the PF was improved in terms of span and richness. It also moved closer to the corner of interest (bottom left in all figures); as the number of evaluations increased, more points appeared near the PF with more emphasis on the region where the normalised difference in total pressure became better than the one of the datum design. It would be interesting to investigate how the optimiser would behave if there was more available computational budget. Several good points that are part of the final PF were revealed at an early stage of the search; more importantly, the designs with normalised negative vorticity magnitude below -1.8 and normalised difference in total pressure below 1.8 are part of the final PF. However, these correspond to different geometrical arrangements. Although this area has also been investigated, the performance did not improve. This suggests the case should be relaunched under a different formulation and settings. On a separate front, a couple of designs have

been found, whose performance is very similar to the datum design's. These could be considered as an alternative for achieving the same target performance. It is noteworthy to mention that only one design has been evaluated which is worse than the datum in both objectives and with very different geometry. Hence, because of the formulation of the MOO, non-promising designs were avoided.

In order to better understand the trade-off, three points were selected in Fig. 4.32 along with the datum design for the decision-making process. Here, the datum is included because this is the only configuration that has actually been tested and serves as the baseline for the comparisons. The other points attempt to improve on the datum in at least one objective. Moreover, the extreme designs for both objectives were chosen in order to demonstrate how the performance changes from one end of the trade-off to the other. Then, the compromise design lives between the extrema and is intended to be an improved candidate design over the datum design in both principles; in a production environment, this could be considered as a prototype, which is worth testing experimentally.

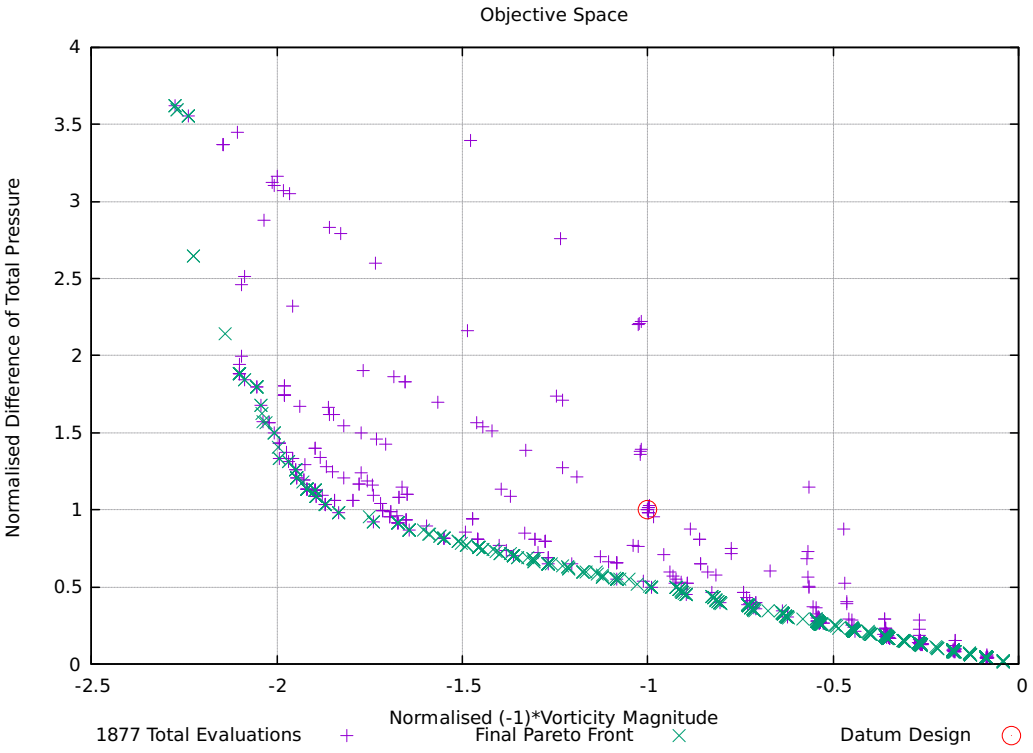



Figure 4.27: Optimisation Search Pattern at 1877 Total Evaluations

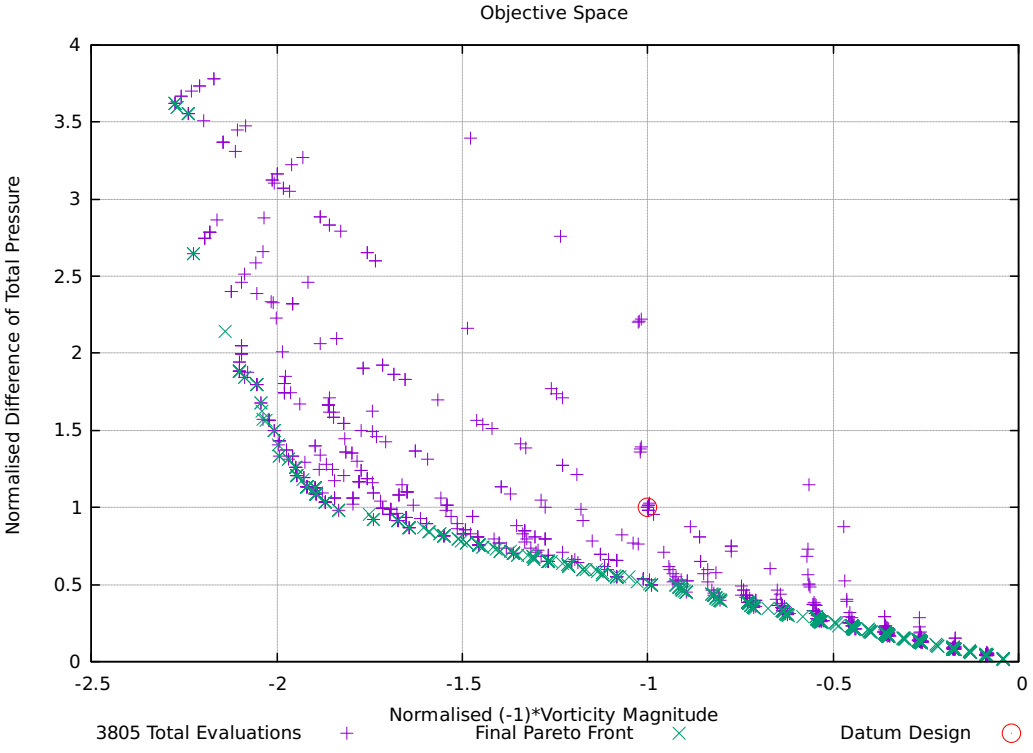



Figure 4.28: Optimisation Search Pattern at 3805 Total Evaluations

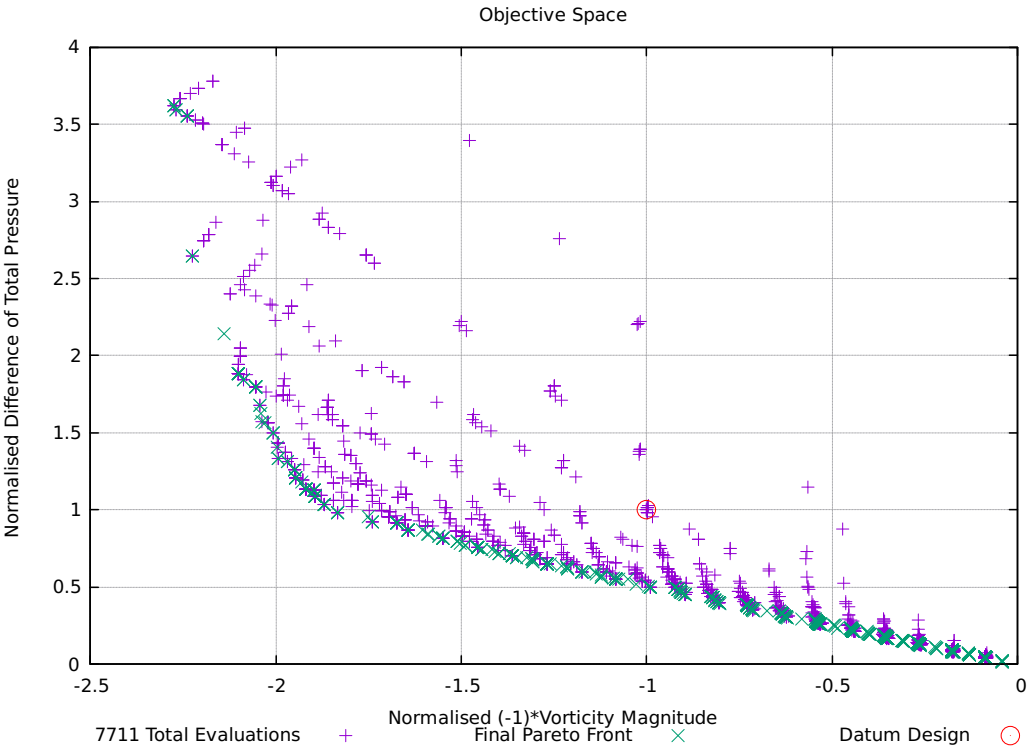



Figure 4.29: Optimisation Search Pattern at 7711 Total Evaluations

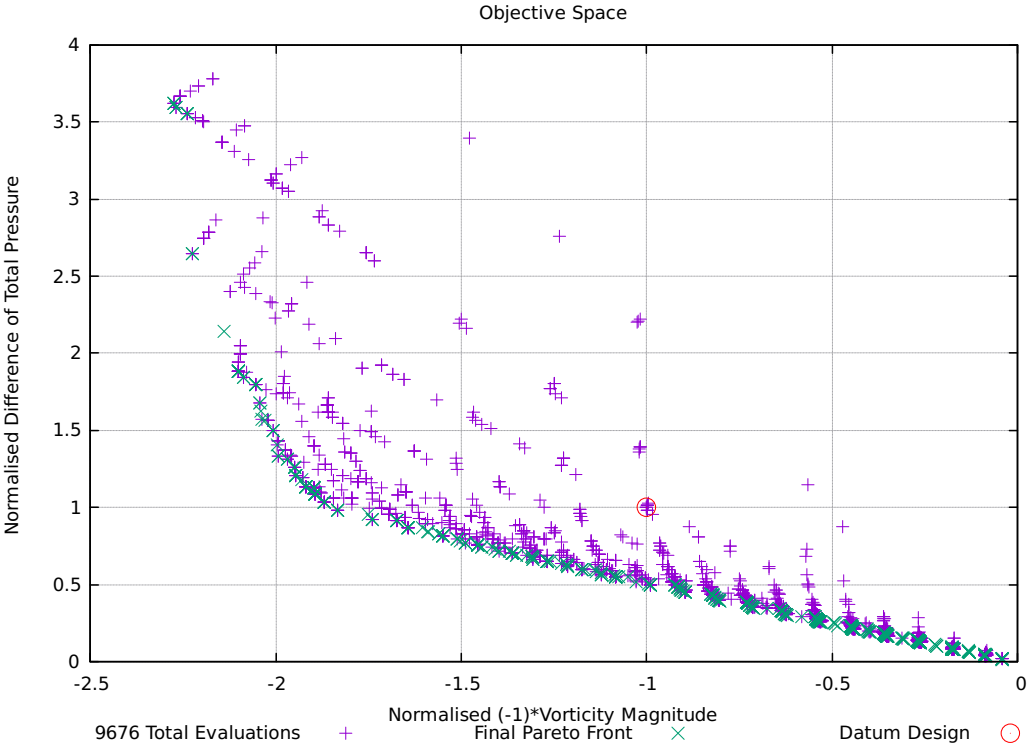



Figure 4.30: Optimisation Search Pattern at 9676 Total Evaluations

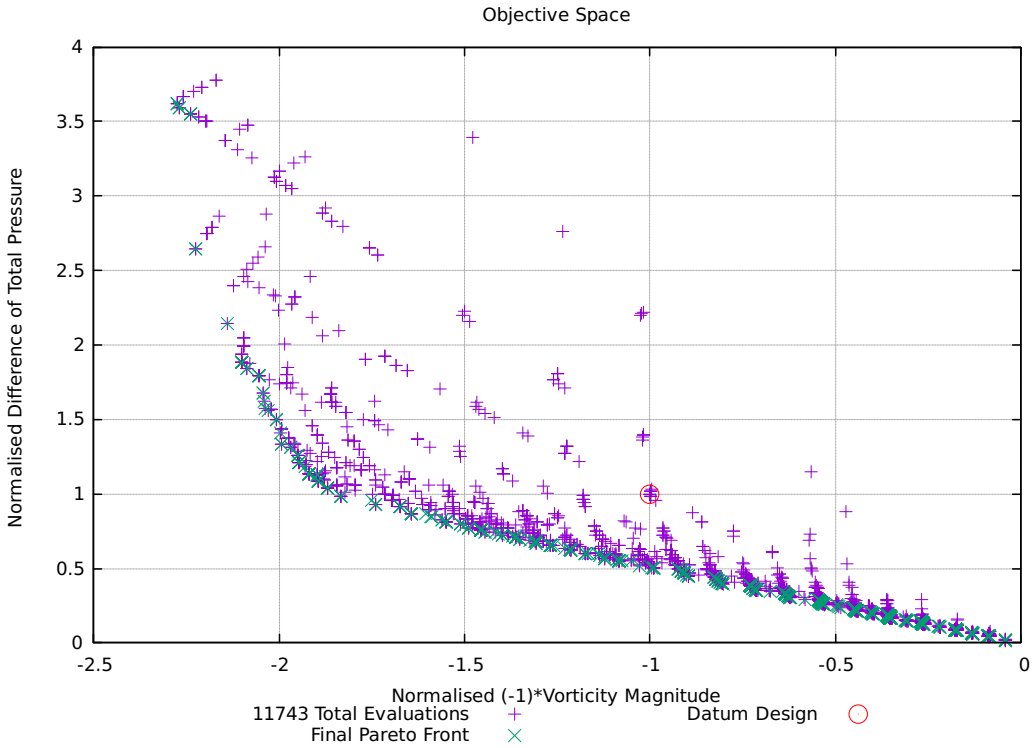



Figure 4.31: Optimisation Search Pattern at 11743 Total Evaluations

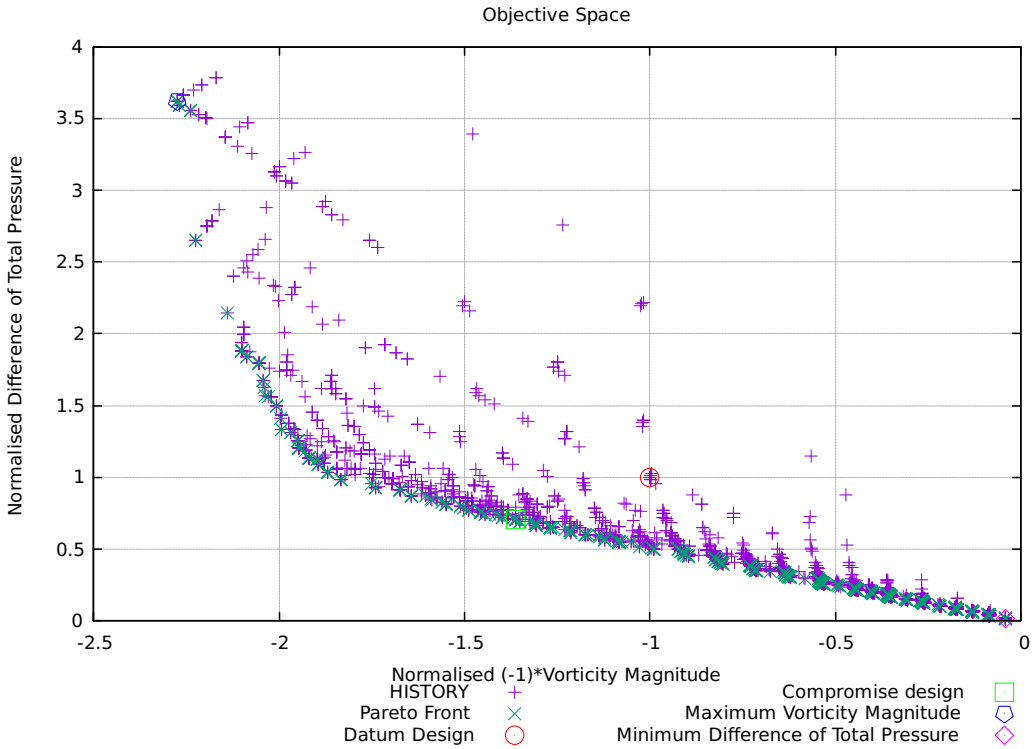



Figure 4.32: Complete Optimisation Search Pattern, PF and Selected Designs (listed in Table 4.13)

The PF of the optimisation process had progressed as depicted in Fig. 4.33. The generic optimum area had been found early relative to the overall number of evaluations. The majority of the trade-off does not change shape significantly in terms of span and quality. On the one hand, near the end of the process, any change is very small. On the other hand, the area of the PF where the normalised difference in total pressure is above 2.0 and the normalised vorticity magnitude becomes less than -2.2 (i.e., top left corner) presents the biggest change. As also discussed above, the complexity of decision space pushed the optimiser to discover fewer points towards the extreme of maximum vorticity magnitude compared to the other objective. This seems to be the upper limit of the current concept of the microreactor, which should be revisited if the mixing capability is very important for the application domain or the business logic.

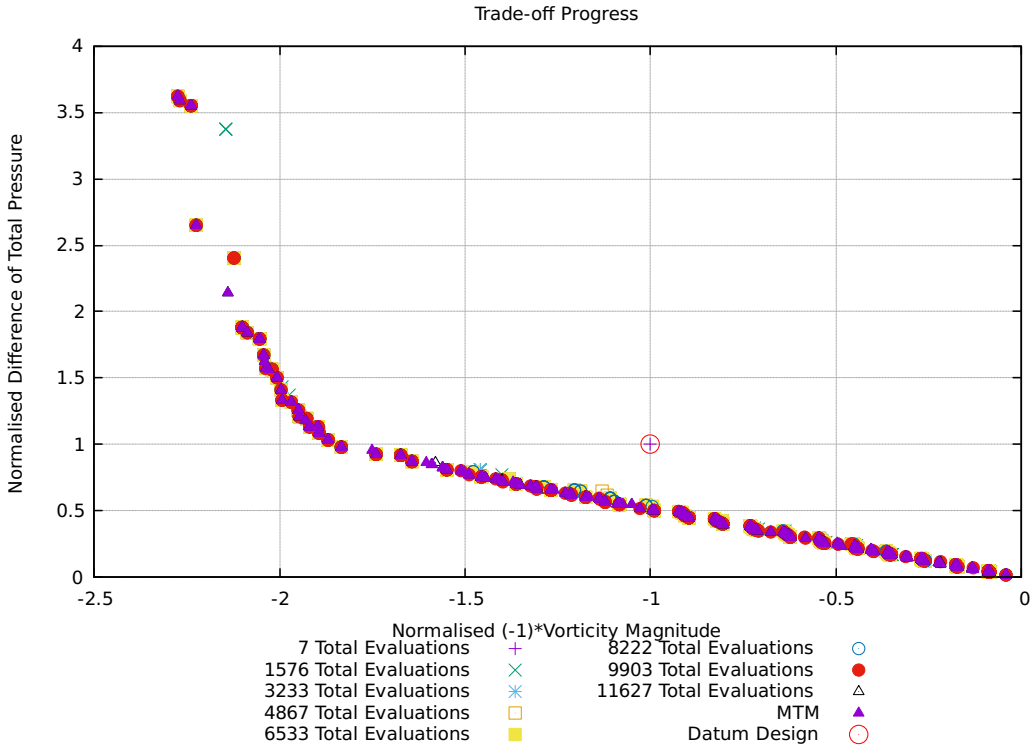



Figure 4.33: PF Progress (Related to Figs. 4.27 to 4.32)

Nevertheless, the initial progress of the optimisation process, as depicted in Fig. 4.34, is also studied in order to identify the performance of the optimiser at the beginning of the search. Comparing the progress of the PF in Fig. 4.33 and 4.34, it seems that the optimiser has very quickly located the area of global optimum performance, as discussed above. Then, it kept improving the PF in

terms of span and richness; this is more obvious in Fig. 4.34. On the one hand, the progress from the datum design up to 737 total evaluations was slow with a few points. Then, it seems that MOTS2 discovered good performance and used the remainder of the computational budget to expand that. On the other hand, it is important to mention that the trend of the PF for the improvement of normalised vorticity magnitude between 1.65 and 2.0 times relative to the base line has been captured very early. Thereafter the PF kept improving until it reached the final shape, as presented in Fig. 4.33.

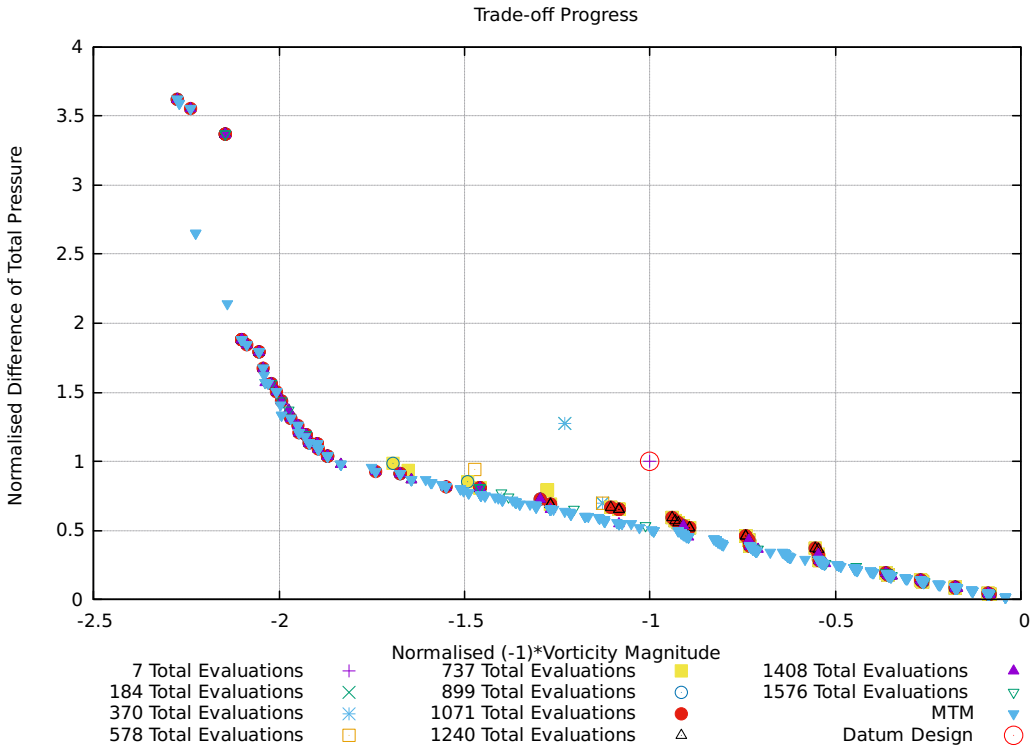



Figure 4.34: The Progress of the PF at the Beginning of the Search, as an Interim Snapshot of the Trade-Off from Fig. 4.33

The user could be informed by Fig. 4.35 whether MOTS2 used any of the available strategies presented in subsection 3.1.3.4 to navigate through the decision space. It should be remembered that these were provided to assist the optimiser to escape from local optima. It is clear that the Intensify Move was employed many times, and it was not required to diversify the search. This means that the optimiser made all the progress depicted in the trade-off figures above by using the local-search features of the optimiser, which could be an indication of the complexity of the decision space. Hence, the additionally evaluated points that

were inserted into the IM during Hooke and Jeeves Move were very useful in order to advance the search.

Similar information from Fig. 4.36 is present in an alternative way when the intensification was performed. Whenever the i_local index exceeds the intensification threshold, the Intensification Move from subsection 3.1.3.4 was employed. In general, the IM had been increasing in size through the search; occasionally its size slightly dropped because dominated points were filtered out. There is not any point to illustrate the complete progress of the size of IM, as it will be difficult to notice calls to the Intensify Move. It is sufficient to say that there was enough information in that memory for the remainder of the search unless any important point in terms of Pareto-dominance was discovered. On the one hand, this means that current configuration settings for MOTS2 were satisfactory to guide the search. On the other hand, restarting the optimisation with finer configuration settings and finer SS is expected to discover additional behaviour.

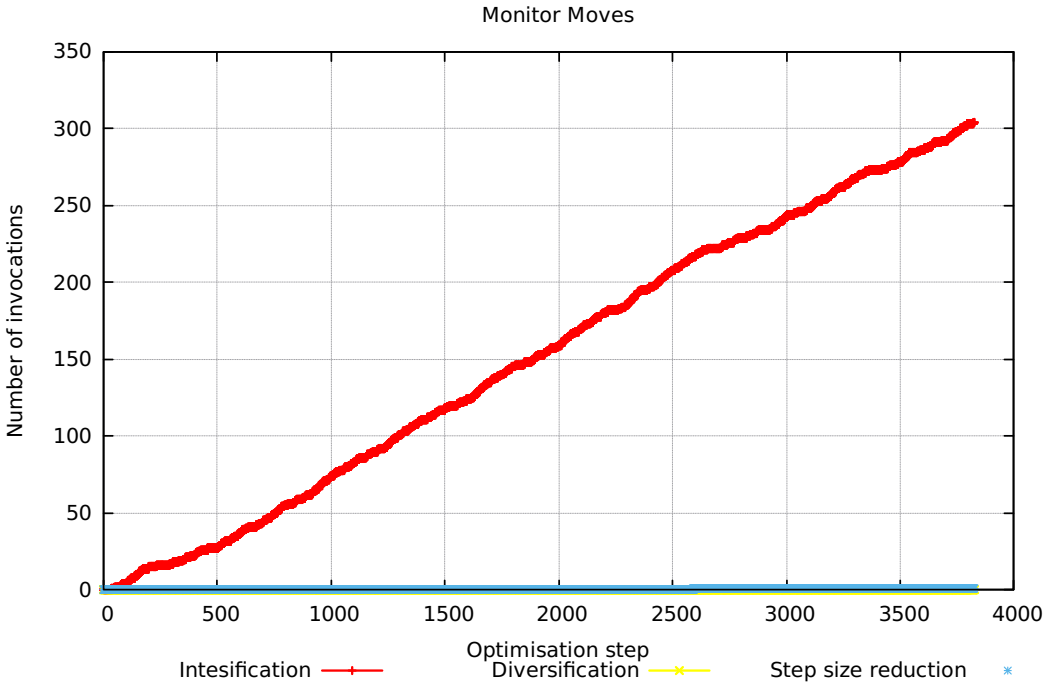



Figure 4.35: Monitoring the Number of Invocations of the Provided Moves Throughout the Optimisation Search

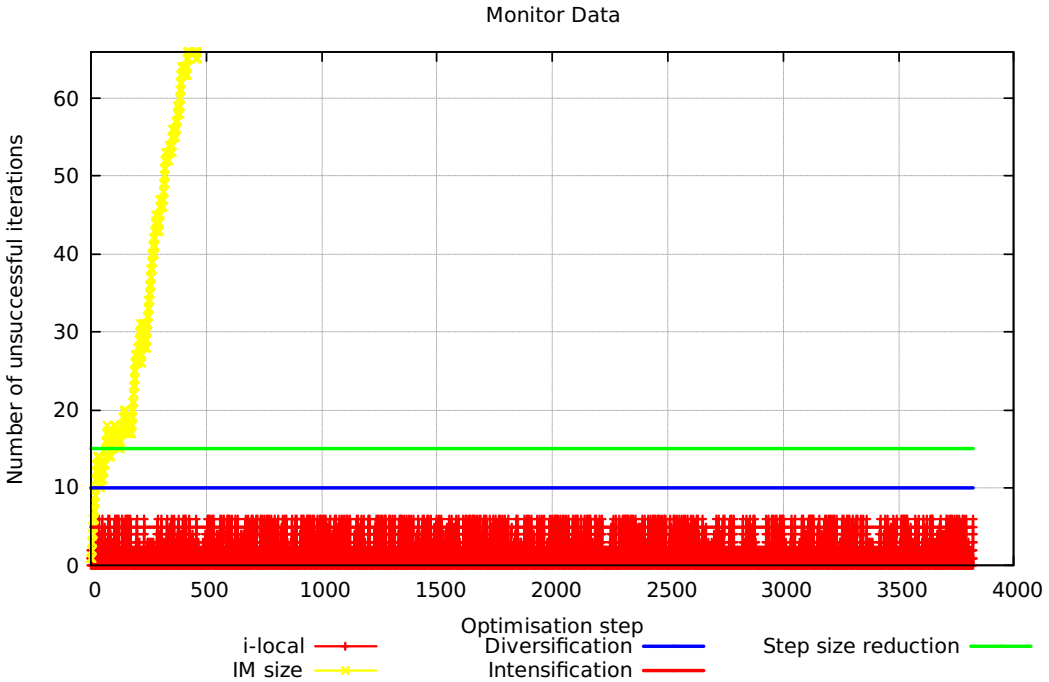



Figure 4.36: The Activity of the Optimiser Through the Number of Consecutive Unsuccessful Iterations

In combination with Fig. 4.36, Fig. 4.37 demonstrates one of the unique features of the optimiser, the kick, as discussed in section 3.1.5. In the previous figure, the user was informed that mainly Intensification Moves were called throughout the search. However, at the 2,529th optimisation step, Reduce Move was also called, but this is not shown in Fig. 4.36, because i_local never reached the threshold of step size reduction. This is explained along with the configuration settings from Table 4.10, and more specifically the *Maximum Improvements*, which participates in the conditions to perform the kick. The kick does not appear in the monitor moves, because it is expected to be called very rarely and would clutter the figure. Because the hypervolume indicator of the PF did not change for a long period, the conditions for activating the kick were met and, hence, the kick was called, which implicitly performs Reduce Move on rare occasions. From that moment, the search step for all the decision variables was halved, which led to the discovery of more points. This explains why the values of 7.5, 8.5, 9.5 and 10.5 for the first decision had been visited in Fig. 4.21. Likewise, many more values were also selected in the third decision variable. Non-visiting values 5.5 and 6.5 for the first decision variables and below 17.5 for the third decision variables means that the optimiser focused the search with the refined step in the regions with the most promising performance.

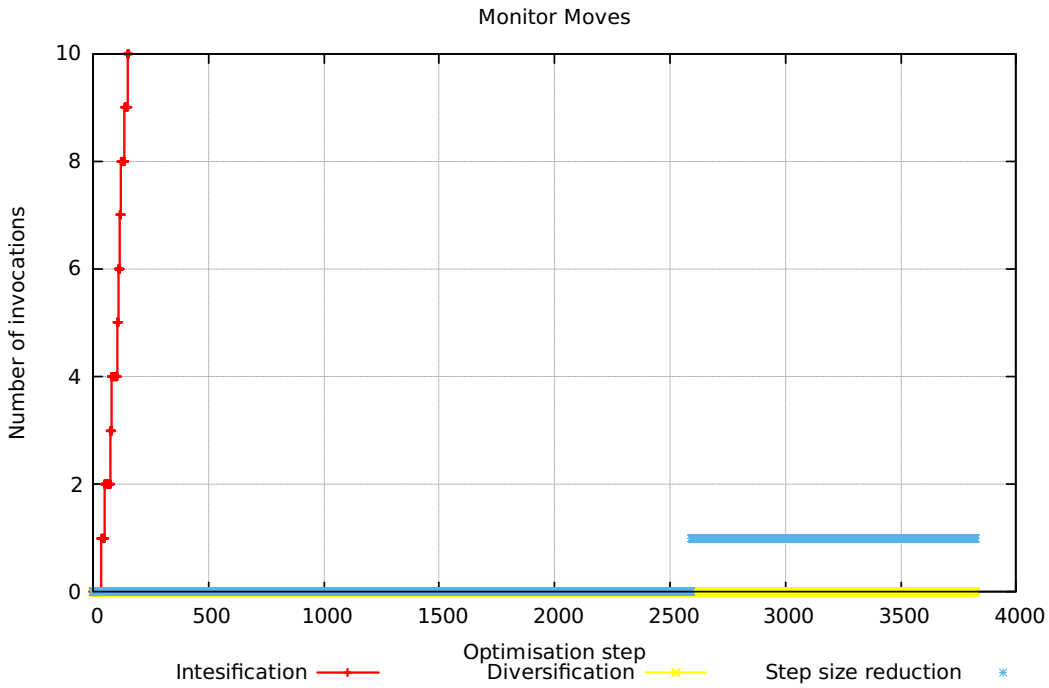



Figure 4.37: Zooming into the Number of Invocations of the Provided Moves Throughout the Optimisation Search

Another way to appreciate the performance of the optimiser is to illustrate the performance of the selected designs that live in the PF against the datum design. The specifications of the datum design and optimised designs from the MOO process are listed in Table 4.13, where the numerical values of each decision variable and the corresponding objective values are presented. The numerical values of the objectives in conjunction with the optimal results discovered in [64] confirm the observation in [45], where the performance of the microreactor is very sensitive to the geometrical arrangement. Tiny discrepancies compared to the selected designs in [64] have a considerable impact on the objectives.

The geometrical arrangements from designs presented in Table 4.13 are shown from Fig. 4.38 to Fig. 4.41. Again, all the designs are compared to the datum design. Achieving the maximum level of mixing is possible by increasing the flow speed and by slightly reducing the radius of outer holes and their distance from the center of the tube, as shown in Fig. 4.39. In contrast, increasing the radius of the outer holes and slowing down the flow can result in minimum possible losses, as illustrated in Fig. 4.41. It was noted that the performance of the design with minimum normalised difference in total pressure belongs to the optimal set only for $Re$ up to 15, which is very slow flow. Finally, significantly increasing the radius of

outer holes, slightly bringing them closer to the centre of the tube while increasing the flow, as demonstrated in Fig. 4.40, achieves more than a 30% improvement in both objectives.

The impact of varying the flow speed in selected designs (when the geometrical parameters remain fixed) is illustrated in Fig. 4.42. For each of the aforementioned five designs the corresponding objectives are depicted, as recorded in the HISTORY during the optimisation search, which explains the different number of points. It is obvious that the datum design is different from the other designs. Although the datum design is not part of the PF, just by changing the flow speed, it has some impact on the objectives, as shown in Fig. 4.42. The performance of the different geometrical arrangements varies linearly by adjusting $Re$. For each set of points of the selected designs the leftmost point has the greatest flow speed and the rightmost point has the lowest flow speed. Therefore, the objective space is expected to be an assembly of such linear trade-offs. It is important to mention that several points with the geometrical arrangement of the compromise design belong to the PF. Moreover, the performance between the compromise design and the design with the minimum normalised difference in total pressure is very similar, despite the different baffle plate: as the flow speed decreases, so does the normalised difference in total pressure.

Table 4.13: Design and Performance Metrics for Selected Designs from Fig. 4.32

| | | **Datum** | **Maximum Mixing** | **Compromise** | **Minimum Pressure Losses** |
|---|---|---|---|---|---|
| decision variable 1 | r | 7 | 6 | 11 | 10 |
| decision variable 2 | Re | 100 | 200 | 150 | 5 |
| decision variable 3 | s | 24.5 | 23.5 | 23.5 | 24.5 |
| objective function 1 | Normalised Vorticity Magnitude | 1 | 2.27427 | 1.361878 | 0.0430896 |
| objective function 2 | Normalised Difference in Total Pressure | 1 | 3.62125 | 0.706433 | 0.018133 |

Figure 4.38: Baffle Plate for the Datum Design

Figure 4.39: Baffle Plate for the Design with Maximum Vorticity Magnitude

Figure 4.40: Baffle Plate for the Compromise Design

Figure 4.41: Baffle Plate for the Design with Minimum Difference in Total Pressure

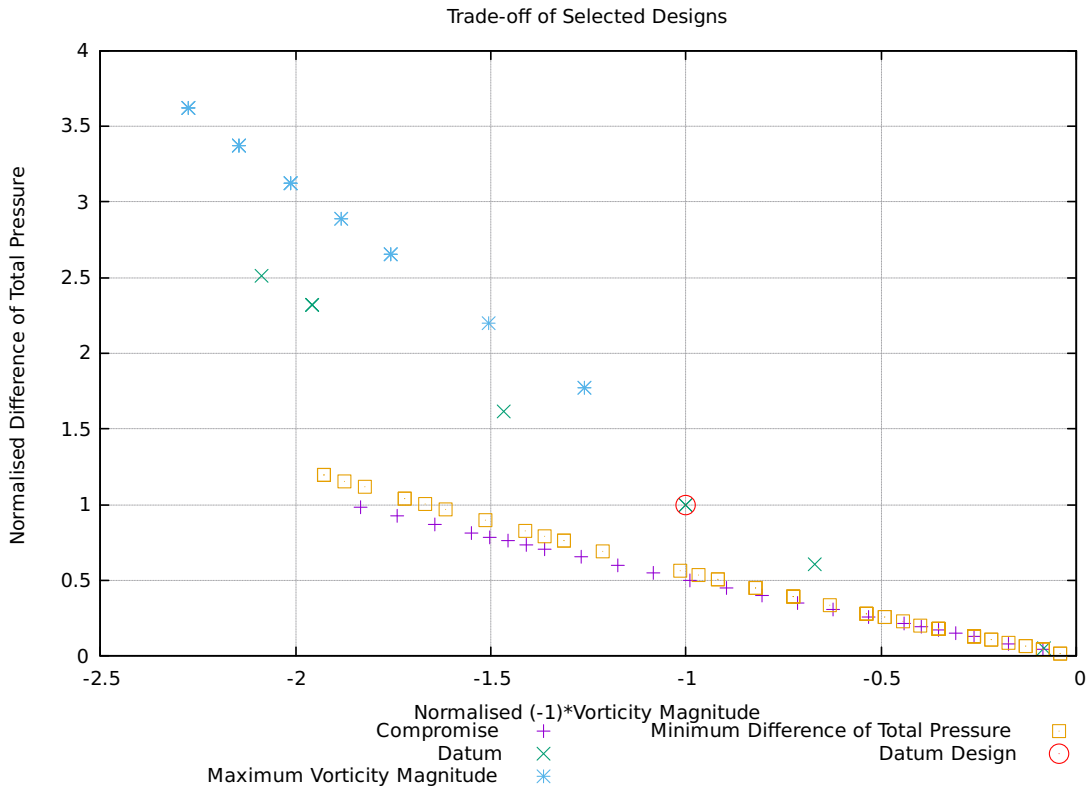



Figure 4.42: Trade-off of Selected Designs while Varying Flow Speed (the Flow Speed Increases when Moving from Left to Right)

Following the analysis of the impact of the considered geometrical and flow characteristics on the performance of the microreactor, it seems important to demonstrate what the corresponding flow is within the device. The flow for each design from Table 4.13 can be found from Fig. 4.43 to Fig. 4.46. As expected from the discussion above, the geometrical arrangement for the design with the maximum relative vorticity presents the fastest flow, as shown in Fig. 4.44. Then, the flow in the other extreme design moves very slowly, as illustrated in Fig. 4.46. Finally, the compromise design, depicted in Fig. 4.45, combines the aforementioned characteristics, where it is clear that the flow in general is faster than the datum design throughout the tube. There are also less discrepancies in the velocity after the baffle plate and at the end of the tube.

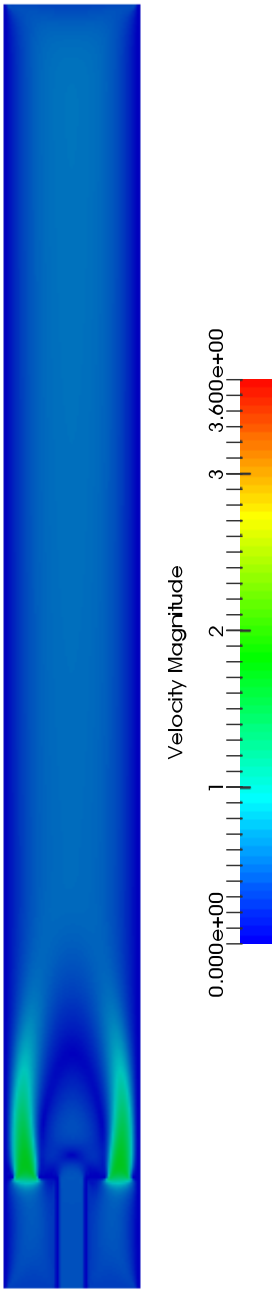



Figure 4.43: Velocity Contours in the mid Plane of the Datum Design

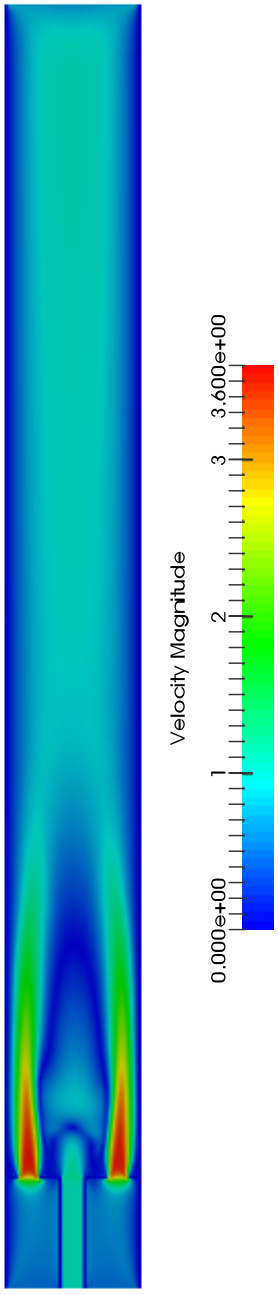



Figure 4.44: Velocity Contours in the mid Plane of the Design with Maximum Vorticity Magnitude

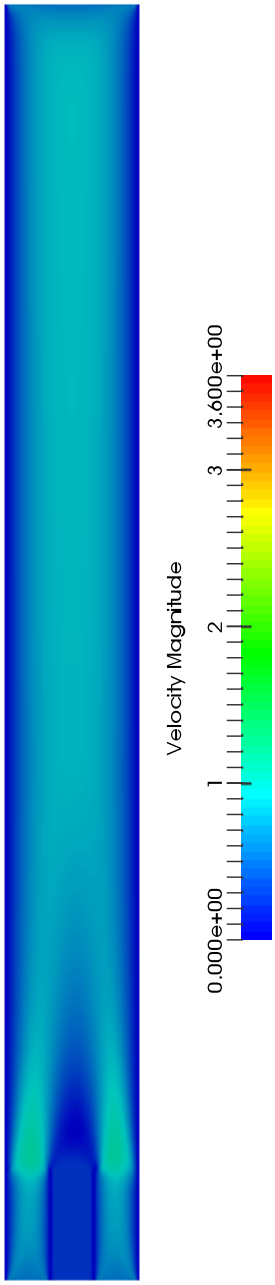



Figure 4.45: Velocity Contours in the mid Plane of the Compromise Design

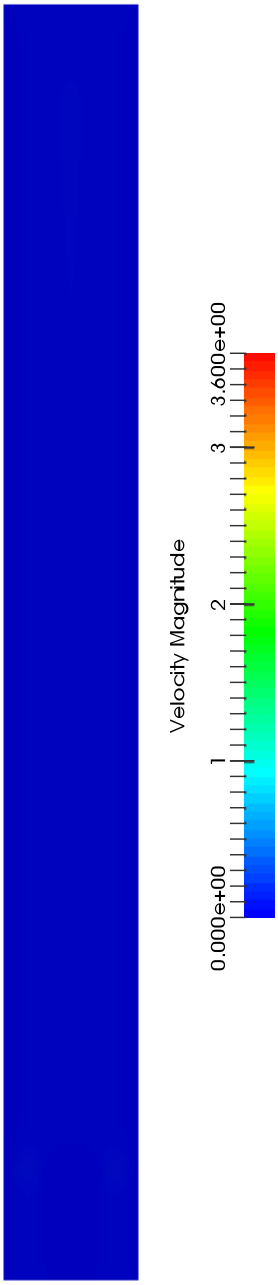



Figure 4.46: Velocity Contours in the mid Plane of the Design with Minimum Normalised Difference in Total Pressure

### 4.2.5 Identified Issues

A fast computational engineering design process for environmentally-friendly studies has been presented, for prototyping purposes, based on the models presented in the previous chapter. In order to optimise two operational characteristics of a microreactor, i.e., the mixing capability and pressure losses, MOTS2 was linked to GPU- LBM, where the optimiser changed the internal geometrical layout of the microreactor and the fluid solver evaluated the objectives. The process took advantage of the high computational efficiency of GPU and the ability of MOTS2 to drive the optimisation search and was significantly accelerated by employing GPU technology, which carried out the most computationally intensive part of the process, i.e., the simulation of the flow via LBM. By using the local-search features of MOTS2 a wide and rich PF has been revealed that improves both of the aforementioned principles of the baseline, which was an experimentally tested microreactor. This study could be carried out on standard commodity machines and could deliver results approximately 20 times faster compared to an earlier study [64].

By applying the aforementioned method 3,000 different designs were simulated. Relative to the baseline, the normalised vorticity magnitude has seen an improvement by a factor of 2.2, whereas the normalised difference in total pressure, and consequently losses, were less than 2%. The range of the discovered normalised vorticity magnitude is less than the range of the normalised difference in total pressure and the response of the device in the objective space is linear when the normalised difference in total pressure is below the datum's. A selected compromise design achieved more than 30% improvement in both objectives. When the normalised difference in total pressure increases more than the datum's, the PF seems to have an exponential shape, before that it follows a linear trend. It was also found that varying the flow speed on fixed geometrical arrangements has a linear response to the performance of the device. For all the selected designs, the lower the flow speed, the lower the normalised difference in total pressure is. Compared to the datum design, maximum normalised vorticity magnitude is achieved with small outer holes in the arrangement of the device, relatively closer to the tube centre and at higher speed flow; conversely, bigger holes at the same distance from the centre tube (as the datum design), and a very low flow speed are required to operate the device at the minimum normalised difference in total pressure. The above compromise design stands between the aforementioned extreme designs. Nevertheless, the optimisation process was driven by a version of GPU-LBM that was compared against a single experimentally measured case. Although the results appear to be reasonable, once GPU-LBM is validated, it could be used to confirm the validity of the optimisation. It is important to mention that this is the prototyping phase of the method and more validations of the optimum and

near-optimum designs should be carried out against real-world data, in order to increase the confidence in the results and employ this method in a production environment.

Regarding the effectiveness of kick, this is the first time where the move actually affected the course of the search, as discussed in subsection 4.2.4 (on page 189). Although the conditions to activate kick were met very late in the optimisation search, a few designs were discovered and the optimiser explored the corresponding area. However, further investigation is required on the conditions of activating the kick move, in order to demonstrate it as a valuable strategy that could assist the optimiser to identify more promising areas in other real-world applications.

*This page is intentionally left blank.*

# Chapter 5

# Concluding Remarks and Discussion

## 5.1 Main Research Contributions

The most important points of this research are listed below:

- An optimisation methodology has been introduced and demonstrated in ATO and design optimisation studies, so as to suggest alternative solutions for real-world applications.
- A new optimisation algorithm, called MOTS2, has been developed from scratch to support the computational engineering design process for real-world applications. During the development, it was verified against benchmark cases and validated against a simple real-world application. It was further developed to scale up the optimisation process when the number of variables increases by harnessing the computational efficiency of GPUs, a very competitive feature that many optimisers (including NSGAMO) cannot demonstrate. The newly introduced move (kick) requires further investigation, as there is not enough evidence to judge its effectiveness. MOTS2 was used in three prototypes applications of real-world cases. First, the trajectory of a commercial aircraft in climb phase with 2 objectives was optimised under operational (ATM) constraints. Second, a 3-phase trajectory (44 variables to describe climb-cruise-descent) with 3 objectives was optimised under operational (ATM) constraints. Third, the geometrical layout of a microreactor with 3 design variables and 2 objectives was optimised under geometrical constraints.
- A GPU-LBM code has been developed to resolve flow simulations in micro-scale within relatively shorter time intervals, so as to enable studying micro-mixing for environmentally-friendly aircraft technology.

- Two ATO applications were performed, where the flight path of a medium-haul aircraft has been optimised to minimise its environmental impact and its fuel burn at the climb flight phase and for a climb-cruise-descent flight. First, by employing two native MOO algorithms, it was found that the aircraft should reach the top of climb by the middle of the range of the flight phase and should fly at a relatively low speed before accelerating to the next flight phase. As a local-search-based optimiser MOTS2 discovered many more optimum points within a relatively closed area of the trade-off, which complements the findings of a global-search-based optimiser that found fewer points that spread over a longer area. In the 3-phase flight the interplay among time, fuel burn and emissions was demonstrated, a clear advantage for MOTS2. In both cases, it was shown that determining the altitude of the initial segment had the greatest impact on the performance of the trajectory.
- The performance of a microreactor has been optimised by altering the geometrical shape and the flow characteristics of the device. In a design optimisation application, where MOTS2 was combined with GPU-LBM, a wide range of solutions were generated among which a compromise design improved both objectives by 30%, outperforming an earlier study.

## 5.2 Discussion of Findings

In a nutshell, this thesis briefly introduced important environmental issues caused by aviation and a two-stage approach was proposed, for short-term and long-term solutions. Related work from the fields of aviation, aerospace research, computational engineering sciences, HPC and CFD was presented. In order to accelerate the computational engineering design cycle for environmentally-friendly applications three tools were developed to tackle the needs of real-world applications. Firstly, a native MOO algorithm called MOTS2 was developed to be used in ATO and design optimisation studies. Secondly, MOTS2 was extended to harness the computational efficiency of HPC infrastructure with the intention of using it on MOO problems of higher dimensionality. Thirdly, a LBM code has been created in order to simulate the flow within a microreactor; this code was ported on GPU architecture to carry out simulations within very short time intervals. All the tools were developed to operate either as a monolithic application or as a component of a larger framework, and are interoperable and reusable. The aforementioned tools were applied to two real-world problems. Firstly, the flight path of existing aircraft was optimised based on operational and environmental performance metrics. MOTS2 was combined with tools that simulate the performance of aircraft and engines in order to optimise the environmental impact and the operating cost.

Secondly, as part of design optimisation studies, the GPU-LBM code was combined with MOTS2 to optimise the geometrical and flow characteristics of a microreactor so as to maximise its mixing capability while reducing its environmental impact. The ability of a device to perform efficient mixing at a microscopic level is key when the fuel is based on hydrogen, which could be used as an alternative fuel in aircraft gas turbines so as to generate energy, as described in [41] and used in [280].

It was demonstrated that the tools and methods proposed can deal with the complexity of examples of ATO and design optimisation, and deliver sensible results within a reasonable time frame, where a large number of innovative and improved solutions (compared to baselines) was discovered. Nonlinear and highly constrained decision space can be effectively searched more rapidly. The trade-off of the considered systems is revealed, more data are produced to assist the decision-making process, the user gains a deeper insight into the problem and the decision maker could make more informed decisions.

Regarding the ATO system, the factors that affect the efficiency of an aircraft flight were investigated and analysed. An APM and an EPM were used on a variety of flight paths and flight speed profiles to determine the performance of the trajectory, where the objective was to minimise the fuel consumption of the aircraft and to minimise the flight time. This is a genuine bi-objective problem, which should be addressed by using native MOO algorithms that can deal with the complex decision space and constraints of the climb phase of an aircraft.

In order to find optimal flight paths that comply with ATM constraints, a method was proposed that combines MOO algorithms and specific APM and EPM models. In subsection 4.1.1, monolithic tools were linked together and then were coupled in a modular way with MOTS2 and NSGAMO, where it was shown that MOTS2 is a competitive optimiser against NSGAMO. Because of the nature of each optimiser, two different trade-offs were discovered for a large number of evaluations, which complement each other in terms of information and performance, while showing a different perspective on the considered problem. Similar trends were found by both optimisers for the decision variables that belong to the common section of PF. Processing and understanding the contribution of each decision variable of the high dimensional decision space was demonstrated, too. Next, in subsection 4.1.2 in order to demonstrate the generality of the method and the flexibility of MOTS2, three software packages were linked together and coupled with MOTS2 so as to perform ATO in 3-phase trajectory with 3-objectives. The second application could be considered an extension of the first, as 44 design variables were used to calculate 3-objectives in a 3-phase trajectory. The flight time, fuel burn and emissions were simulated by combining different software packages than before. Without loss of generality, this method could be used to study even more complicated cases with a greater level of detail, for higher accuracy and

fidelity.

The optimal flight paths share the feature that their performance is governed by the altitude of the first segment of the flight. Thereafter, the importance of the remaining decision variables is greatly reduced by the order of occurrence, as listed in Table 4.3 and Fig. 4.14. This attribute was identified by both optimisers, in both applications, and were used to guide the optimisation search. Further cases should be carried out with more decision variables to determine the global efficient behaviour of the climb phase.

By analysing the profile of altitude and speed of the optimum designs that belong to the common area of the PFs, the optimal performance is related to a simple pattern, as illustrated in Fig. 4.5. In general, the aircraft should reach the top of climb as soon as possible and then it should maintain a level flight; it should increase speed for the first segment, then it should fly at an almost constant speed for the next two segments, and it should accelerate to the maximum allowed speed at the end of the climb phase. From the multi-dimensional analysis, the fact that many decision variables are gathered at the maximum allowed altitude suggests the ATM constraints should be revisited so as to balance legislation constraints against environmental gain. A wider flight envelope would allow the optimisers to further improve the considered objectives.

This study has revealed that employing two optimisers, one global-search-based and another local-search-based, was beneficial to understanding the complexity of the problem. Of course, the PFs are different in terms of span and richness but comparable in terms of hypervolume quality indicator, see Table 4.4. They can be used to either emphasise a certain region of the trade-off, where MOTS2 discovered many optimal designs in a relatively short range or to appreciate the big picture of the problem with less information between certain objective values. It is expected that if the configuration settings of the optimisers (see Table 4.2) were specified to balance between exploration and exploitation, then a similar trade-off would be revealed.

Relative to NSGAMO, a narrower and richer PF was revealed by MOTS2, whereas the other optimiser found a wider trade-off with fewer points, as shown in Fig. 4.4. The optimisers discovered fractions of the PF that are common to both, where the objectives behave similarly and their corresponding decision variables follow similar trends. The wider range between extreme trajectories found by NSGAMO in Fig. 4.5 is expected, since the range of its PF is also wider. In any case, the selected compromise trajectory found by both optimisers is almost identical. Although the PFs are not identical, they have the same shape at similar ranges, which implies that the optimal behaviour of the considered system has been reached.

Regarding the design optimisation system, the factors that affect the efficiency

of the microreactor were investigated and analysed. The GPU-LBM was used on many different combinations of geometrical arrangements and flow speed to determine the performance of the device, where the objectives were to maximise its mixing capability and to minimise its environmental footprint. This is a genuine bi-objective problem, which should be addressed by using native MOO algorithms that can deal with the constrained decision space of the operational characteristics of a microreactor that is simulated by using the LBM.

The GPU-LBM flow solver was combined with MOTS2 in a modular way so as to find the optimum operational characteristics subject to geometrical constraints that adhere to the concept of a microreactor, which was originally tested experimentally. GPU-LBM greatly reduced the simulation/evaluation time of a single design point to approximately 145 seconds, which is approximately 20 times faster than the parallel CPU implementation of [46]. By extending a previous study [64] many more designs were investigated in shorter time intervals. The discovered trade-off is wider, richer, and closer to the ideal target. Similar designs have been found, but as mentioned in [45] minor changes can have a great impact on the performance of the device. Analysing the contribution of each decision variable to the PF and the performance of the optimisation process was presented, too. The computational design cycle has been significantly accelerated. This can be integrated with experimental processes for the complete design and testing of products, as was also suggested in [47].

Optimal geometrical shape and flow speed were found. Similar trends compared to [64] were found, but now the discovered performance was better because it was computationally inexpensive to evaluate many more designs within acceptable time frames. More layout concepts (i.e., triangle shaped holes instead of circular ones) of the baffle plate could yield even better behaviour.

By analysing the specification of the optimal designs (i.e., the shape of the baffle plate and the flow speed), as shown on page 192, a simple pattern was identified: For a fixed combination of the geometry decision variables, any change to the flow speed results in a linear response in the objectives. Moreover, big outer holes in the baffle plate are related to minimum normalised difference of total pressure, which means fewer losses and, consequently, a better environmental impact. Conversely, small outer holes maximise the mixing capability at the expense of large normalised difference in total pressure. A compromise design can improve both objectives by more than 30% relative to the datum design.

Using a local-search-based MOO algorithm like MOTS2 for optimising a microreactor was very beneficial in the design optimisation process. All the optimum points were discovered by using the local-search scheme to navigate in constrained decision space.

The revealed PF is wider, richer and closer to the ideal targets. This was expec-

ted, since it was possible to evaluate many more designs. It was demonstrated that even the geometrical factors considered in the optimisation case have a significant impact on the performance of the microreactor. This had also been confirmed experimentally. Consequently, the extreme designs look very similar to the ones presented in [64], but their performance is better as their gap in the objective space is larger, see Fig. 4.32. The flow speed is the actual difference between the compromise designs.

Parallel Coordinates has been undoubtedly a useful technique to appreciate multi-dimensional data. This had been used to visualise the complexity of the studied systems. It was particularly important to understand the impact of the decision variables on the PF and to identify patterns based on the target performance. Moreover, they can reveal hidden interaction(s) among the considered designs and objectives. Hence, the user gains a deeper insight into the problem and can make more informed decisions.

## 5.3 Main Conclusions

Findings from the benchmarks and applications are summarised below:

- It was demonstrated that MOTS2 can handle the multi-modality of the PF of ZDT4, whereas there is no evidence of NSGAMO in this particular test function [273]. Nevertheless, MOTS2 demonstrated the worst performance on ZDT3, but NSGAMO successfully captured the target PF. Comparing their performance, if the shape of the trade-off is known, MOTS2 should be used in problems with a multi-modal trade-off, whereas NSGAMO should be employed in discontinuous trade-offs.

- The shape of the trade-off between the objectives in airfoil shape optimisation, ATO and micro-mixing optimisation was continuous. The weakness of MOTS2 to sufficiently capture the discontinuous PF of ZDT3 was not critical in the studied examples.

- When optimising the climb phase, MOTS2's trade-off was narrower than NSGAMO; the gap between the most fuel-efficient trajectory discovered by MOTS2 and NSGAMO's is approximately 80 kg over a time period of approximately 35 s. The flight path trends revealed by NSGAMO are closer to the theoretical optimal cruise-climb trajectories.

- In the 3-phase ATO, the trends of the altitude profile during the climb are very similar for all the trajectories with a notable difference in the speed profile for the same phase. With respect to performance, the most notable gap in performance among the extreme designs is in the fuel consumption.

- In both ATO applications, the altitude of the first segment of the flight is the most significant factor with respect to the optimum behaviour.
- The combination MOTS2 with GPU- LBM delivered a trade-off approximately 5% better in terms of hypervolume compared to reference study. More importantly, the performance of the discovered designs is very similar with respect to the area of low difference of total pressure, even after the considerable speed up. However, the vorticity magnitude (i.e., the mixing capability) improved by approximately 20%. The suggested compromise design geometrically is very similar to the minimum pressure losses design, but relatively distant in terms of performance (mainly because of the flow speed), which also confirms the experimental observations of [45], where minor changes can bring dramatic changes in performance.

With reference to the objectives:

1. Six instances of the devised methodology were applied to test and benchmark functions, ATO studies and micro-mixing, where a satisfactory trade-off was obtained, which in some cases is comparable to other state-of-the-art methods (i.g., MOTS2 against the original Multi-Objective Tabu Search).
2. A version of GPU- LBM was developed and qualitatively compared against experimental measurements, where similar flow features in the geometry of a microreactor were identified, which also follow theoretical observations of the flow in a pipe (i.e, parabolic speed profile after the baffle plate). This version was applied to a micro-mixing optimisation, where a compromise design was suggested that could improve both objectives by approximately 30%.
3. MOTS2 was compared against NSGAMO in the climb phase optimisation, where the performance of the latter was better in terms of hypervolume indicator and trade-off span. Compared to the original implementation of Multi-Objective Tabu Search, the integration of GPU- LBM into MOTS2 revealed a trade-off that is approximately 5% better in terms of hypervolume and expanded the trade-off discovered in an earlier study.
4. When MOTS2 was applied to a 3-phase ATO, a number of extreme trajectories were identified that are Pareto-equivalent, so as to demonstrate the effectiveness of the methodology.
5. The compromise design in the micro-mixing optimisation improved both objectives by approximately 30%.

## 5.4 Assumptions and Limitations

This section includes a list of limitations and assumptions, where each entry has the following elements:

- Risk description
- Impact
- Importance (Low/Medium/High)
- Probability to raise issue (Low/Medium/High)
- Mitigation plan

Considering the technology readiness level of this research, the following assessment is the view of the author and it is expected to vary as the methods and tools mature.

1. So far, when selecting the compromise solution(s) (for implementation), they were selected manually and deliberately from the middle of the PF, so as to be as fair as possible. This could be biased by the personal preference of the stakeholder, but potentially not acceptable in real-world cases, depending on short-term/long-term strategies or otherwise. (Low, Low). Multi-criteria decision making processed could be employed, as detailed in sub-subsection 5.5.1.1.

2. For simplicity and proof-of-concept, the climb rate of the aircraft was not considered, which accounts for the discontinuous flight level (as shown in Fig. 4.18 for greater detail) and the immediate transition from the climb phase to cruise phase (as shown in Figs. 4.16 and 4.17). (High, High). Although the remainder of the trajectory is sensible, a new case should run again with explicit ATC constraints for continuously ascending climb in conjunction with a tighter flight envelop, so as to produce a more realistic trajectory plan. Nevertheless, for the purposes of demonstrating the capability of the optimisation methodology, this is acceptable. This limitation can be addressed by introducing a climb rate constraint.

3. For methodology demonstration purposes, it was assumed that climb-cruise-descent accounts for 90% of the flight. However, other flight phases also contribute to the objectives. (Medium, Low). In order to produce a more realistic trajectory, all the phases should be considered, block-to-block.

4. HERMES does not allow to specify a flight range (i.e., it is auto-specified). How could it handle waypoints? By design, HERMES is a point-mass model,

which is appropriate for prototyping purposes. (Medium, Low). When higher levels of precision and/or fidelity are required, models that can model 6 degrees of freedom should be used, as demonstrated in subsection 4.1.1. However, this is expected to increase the computational workload and the time required to solve a problem.

5. Because of the complexity and the number of principles involved, the generation of ATO will not cover a very wide range of variables and parameters, but only enough in order to be sensible to employ it as a method. In the current form of GATAC, according to the author's experience, setting up a big case on GATAC is expected to be a time-laborious task. Additional programming skills might be required to set up a new case, similar to the application in 4.1.2. (Medium, Low). A number of strategies is suggested. Variables could be grouped, so as to reduce dimensionality. An additional application programming interface should be provided by GATAC. Alternatively, the architecture of the optimiser should support this, but this would invalidate the use of GATAC.

6. By using GATAC, when small cases are considered in terms of the time needed to carry out a single evaluation, the overhead of the interfacing mechanism could cause small delays and might be affordable. However, when the problem scales up, the communication overhead could cause significant delays comparable to the simulation time and using the proposed method is not be advisable. Moreover, the interface between modules by exchanging files is expected to be a performance bottleneck if communication is required very frequently. (Medium, Low). An event-based communication protocol will be expected to be more effective and should be handled by a framework like GATAC or a similar one. Alternatively, the 3-phase ATO did not depend on GATAC.

7. The total number of threads depends on the CUDA compute capability of the device. The computational efficiency is subject to the underlying hardware. (Medium, Low). In the absence of CUDA the original LBM could be used.

8. Because the geometry is represented in nodes, it is not possible to model a general curvilinear surface. For the purposes of the current technology demonstrator, a stepped approximation is expected to be sufficient. (Low, Low). In fact, this problem will always persist. A finer $lu$ could be used, that could yield higher levels of precision. However, this could increase the computational workload.

9. The environmental variables will always be known, will be provided and will remain constant throughout the simulation. This can drastically alter the

performance of aircraft and, consequently, the trajectory. (Low, Low). A profile of the environment could be developed, that would alter the environmental variables accordingly.

10. For the purposes of this study, multiple APMs, the EPMs and GATAC were required. Lower fidelity models could be developed from scratch, but this would require considerable resources to test, verify and validate. (Medium, Medium). They could be substituted by other modules of the same class or even ones of lower fidelity. It was demonstrated that MOTS2 could also operate as a stand-alone optimisation algorithm. In fact, it was designed to cope with this risk, too.

11. Currently, only 2D trajectories were considered. Considering a single aircraft in an empty airspace, the effects of 3D trajectories could affect the validity of certain discovered solutions and could affect additional objectives (e.g., stability). (Medium, Medium). Appropriate constraints should be modelled and implemented in the models. This is expected to increase the complexity of the models that simulate the performance of an aircraft and could reduce the feasible area of the decision space. However, a 3D representation is out of the scope of this research.

12. It is assumed that the aircraft pass from the exact point. For current technological level, this is not expected to raise any issue. (Low, Low). Normally, any deviation from the exact point is important because this would specify what air navigation procedures should be followed and it affects the ATM to a greater extent. Here, for simplicity, it is assumed that the operator and the FMS will make sure that the aircraft would follow a strictly prescribed path. Otherwise, in the formulation of the model, stochastic noise should be introduced, which would increase the overall complexity.

13. During the simulation of a segment on APMs, the flight conditions are assumed to be constant throughout the segment. Both as an assumption and a limitation, straight line segments were used. Otherwise, the modelling of the trajectory would be harder to resolve and a finer level of detail might be required. (Low, Low). Additional modules on GATAC or any framework could be developed, but this would increase the computational load.

14. Any other modification(s) to the APM, except for the quantities related to the trajectory, such as the configuration of runways and other airport's intrinsic features were not considered. This would only increase the complexity of the optimisation problem, as a range of constraints should be considered such as the structural and operational constraints of the aircraft,

and the ATM procedures and regulations. (Low, Low). MOTS2 was originally developed to be deployed on such highly constrained cases, where global-search-based optimisers are expected to be trapped or (even worse) not to be able to discover any feasible solutions. This is also suggested in sub-subsection 5.5.2.1.

15. The number of threads should be a power of 2 and has to be specified following CUDA guidelines, so as to saturate the GPU with enough threads and enough memory per SM. Then, the computational efficiency can be appreciated. However, as suggested by the manufacturer, the speed up would be better if the previous condition is met. (Low, Low). By design, the architecture of GPU- LBM is flexible enough to run on any number of threads.

16. The simulation speed of the package described in section 3.3 depends on the combination of the hardware of GPU and the corresponding hardware driver. By design, LBM, as a computational method, is known for its ability to run fast on GPUs. (Medium, Low). Alternatively, the original flow solver based on LBM could be deployed on CPU. It has been tested on Linux-based environments, and it is expected to perform equally well on other platforms.

17. In traditional CFD studies, a computational grid is generated around the geometry of interest and grid dependence studies are carried out, as described in [281], so as to study the accuracy of the flow-solver. However, this mechanism is not accessible here. By definition, LBM is a different computational method and the precision partly depends on the collision operator (and the lattice size, which could be an analogous to the grid). In the current implementation of GPU- LBM only one type of operator was implemented. (Medium, Medium). The collision operators in LBM are fundamentally part of the solver and contribute to the speed up and the precision. In this case, appropriate memory management patterns should be employed, so as to achieve satisfactory performance on GPUs. Currently, it is only possible to change the lattice size, but this would require modifying any software/system that generates the geometry of interest, so as to consider a finer computational domain.

18. The configuration settings of MOTS2 were specified based on the author's experience, but might not be applicable in other cases. The optimiser would still be able to run, but (possibly) more objective function evaluations would be required to achieve a satisfactory PF. (Low, Low). By design, the configuration settings of MOTS2 are easier to understand, which is expected to assist (even novice) users to carry out any changes with greater confidence. Even in the verification of MOTS2, in sub-subsection 3.1.6.1, the settings

listed in Table 3.1 are not unique; there might be a better combination that yields the target PFs with fewer objective function evaluations.

19. Currently, the underlying simulation of the microreactor by using GPU- LBM needs to be enhanced, in order to increase the confidence in the simulation method and in the results of the optimisation process. Consequently, many more different geometrical arrangements could be simulated with higher precision. (High, High). Different boundary conditions and the numerical scheme of the relaxation method (described in [254, 282]) could be implemented. In addition, more comparisons against experimental measurements, for validation purposes, should be carried out, so as to accurately drive the optimisation process towards solutions whose actual performance can be obtained in their physical design.

## 5.5 Future Work

This section describes work in progress, which is based on the foundations of the research presented in the previous pages. A number of future extensions are suggested below, starting from the improvements from the top level to enhancements at lower levels. The majority of these proposals can be implemented independently, unless stated otherwise. They all aim to enhance the methods and tools presented in this research; they are mainly expected to reduce the elapsed time of certain engineering processes or procedures while increasing their quality and providing more data. This will allow the decision maker to perform a more informed decision and will increase their confidence.

The importance, complexity (in terms of development/research effort) and abstraction (in terms of the level of detail) of the listed recommendations vary from short-term tasks to very strategic tasks. Depending on the level of seniority of the reader, the former tasks target individual researchers (perhaps at an early stage) that would decide to immediately carry on existing work, whereas the latter tasks aim at managers and/or head of departments who might be looking for a vision and future trends.

### 5.5.1 Improving the Methods

Below, the reader can find the most abstract upgrades to carry out and their merit can be easily appreciated. Implementing and testing these takes longer, but they are anticipated to pay off in the long term.

#### 5.5.1.1 Improving Optimisation Process and Procedures

A powerful extension would be to integrate the optimisation process with multi-criteria decision-making methods [283] after performing an optimisation search (or in a proactive approach) so as to select a number of solutions from the PF. The merits of linking MOO with multi-criteria decision making were reported in [284]. Among the methods mentioned in [283], using the family of methods *ELimination and Choice Expressing REality* [1] can be a very promising step. This family of methods has the ability to handle data sets that are characterised by a high degree of uncertainty, which seems appropriate for environmental applications. It is less sensitive to changes in the data set and considers more aspects than other methods. In addition, discrete criteria of either quantitative or qualitative nature could be manipulated.

Essentially, the methodology attempts to present a series of steps that reveal optimal solutions to ATO problems. The optimiser conducts an intelligent search over complicated decision space and learns from it as long as the operation goes on. This systematic gathering of information is mostly automated, while the decision making is left to the decision maker. The ultimate goal is to harmoniously combine the computational and intellectual parties while appreciating and complementing the contribution of each side. Hence, more mechanisms should be provided for more interactive operation between the decision maker and the system.

If the time required to evaluate a decision vector is no longer important or it is improved by using alternative methods and tools, it will be worth increasing the number of evaluations to discover solutions of a higher quality. An important feature of many objective(s) is their robustness to slight variations of the variables. The robustness can be studied for the environmental conditions, the internal attributes of the system of interest or both. Obviously, the latter is more complicated to resolve. Hence, from a single objective, a robustness metric can be derived by evaluating similar decision points in the local neighbourhood in decision space around a candidate decision vector for evaluation. Then, the robustness can be calculated by assembling all these surrounding evaluations either as a simple averaging or another more sophisticated process; this will capture additional behaviour around a certain decision point which could be used in later stages of the search by other memories and processes.

Frequent and recurring information should be stored and recovered as a wrapper of the actual module that performs an objective function evaluation. Therefore, unnecessary evaluations of decision vectors will be kept to a minimum. On the one hand, this technique looks similar to HISTORY, but can be linked with any type of optimiser, giving them a memory feature that can be used as shown

[1] in french it translates to *ELimination Et Choix Traduisant la REalité*, or ELECTRE

in the previous chapter. In the event of a restart, this information could remain there, saving even more time and computational budget. On the other hand, if the evaluation consists of an assembly of partial solutions, such as in ATO where the same sub-modules are called on a separate flight segment, then this could also assist MOTS2 to advance faster. As part of the evaluation manager, an external look-up table could be created, where partial solutions are saved after a successful execution. In latter stages of the optimisation process this is expected to save considerable wall-clock time because the table would already be populated; if the decisions of the PF share similar decision variables, most likely any partial solutions will already exist in the look-up table. In general, this feature is expected to be a good enhancement in local-search-based optimisers; the more computationally time consuming the evaluation is, the more beneficial this approach is.

The existence of memories could further be used to extract even more information by applying data mining techniques, as suggested in [215]. Because the nature of the decision variables of the problem is continuous, in combination with a previous look-up table, certain conditions could be predicted by using Kriging interpolation [285, 286], Support Vector Machines [287, 288], design of experiments [289] or similar methods. This interpolation could be further accelerated by using the ideas described in [290, 291] and by developing them on GPUs, because of the number of highly intensive computational procedures applied to structured data. If the computational cost increases because of employing higher fidelity processes, this approach will be even more beneficial because considerable time will be saved. This improvement will also assist and/or enable other processes, such as calculating the robustness of the system, calculating sensitivities and creating performance maps. Surrogate models [140] can add an extra perspective of intelligence to the optimisation process, as demonstrated in [292], by predicting the evaluation of the objective function without actually executing the resource intensive computational procedure. Of course, this comes with a range of confidence, and specifying certain thresholds could automate the decision of either evaluating an objective or predicting its value. This would be particularly beneficial when either the number of variables increases, or the number of objectives increases, or more objective functions are required, or all of the above. Alternatively, having such a system could either be linked to existing optimisation processes or could be used independently to conduct performance-based optimisation with expected improvements, as described in [140]. Normally, there should be a single surrogate model per objective. Hence, this will lead to the development of multi-surrogate modelling. Nevertheless, the designer's experience and insight are mandatory to determine the ongoing stages. As pointed out in [293] Design and Analysis of Computer Experiments (DACE) and Radial Basis Functions (RBF) behave particularly well for low-order non-linear functions. The differences between design

of experiments and DACE are explained in [294]. However, for the same type of functions, the accuracy is not improved significantly by using a large sample size. This behaviour depends on the expensive fitness function calculation, the lack of an explicit model for calculating the fitness function, the noisy environment of Evolutionary Algorithms and multi-modal objective space, as presented in [295]. As observed, applying RBF and DACE in cases of high dimensionality might be prohibitively costly compared to exact calculation, as also stated in [296].

Selecting a better initial point is an important feature in many single-solution-based and local-search-based optimisers. By using more effective sampling on decision space, the sensitivity of decision space can be extracted. Thereafter, the optimisation algorithm should start from the most promising area of the decision space.

Another important feature of any optimisation process is to understand the importance and contribution of the decision variables to objective space. Hence, the optimisers could navigate more effectively and could converge to the global optimal behaviour faster. One standard method to calculate the sensitivity of decision variables is to apply PCA, as suggested in [297]. This could be performed as a procedure of MOTS2 for higher effectiveness in carefully selecting the decision variables while considering their contribution to objective space. It is described in greater detail for MOTS2 in sub-subsection 5.5.3.1. Finally, based on the calculated sensitivity, procedures could also be launched proactively by the decision maker. These procedures will discover an optimal behaviour within a relatively small fraction of decision space and will feed the results into a top-level optimisation process so as to give the flexibility to the user to intervene whenever it is considered appropriate.

#### 5.5.1.2 Improving Process and Procedures for Aircraft Trajectory Optimisation

Following subsection 4.1.1.6, the first immediate study would be to investigate the performance difference between MOTS2 and NSGAMO, so as to increase the effectiveness of MOTS2. Among others, different problem instances (i.e., a combination of flight paths and underlying models) and/or various optimiser parameters could be trialled. Next, additional moves for MOTS2 could be implemented specifically for ATO.

The main idea is to gradually integrate all the different principles from strategic planning, to operational planning, up to the lowest levels of simulation. Most of these can be implemented for the current aircraft and infrastructure. Then, they could serve as the basis for frameworks to design future aircraft that will use the current infrastructure and would deal with the future political, environmental, social, technological and economic challenges. The increased computational cost

could be alleviated by employing tools and methods such as the ones described in section 3.2, which could deal with high dimensionality. The vision of commercial aviation is described in [298, 299].

The methods presented above were originally developed to modify the trajectories in which existing aircraft would fly, but involving new aircraft and alternative routes would give deeper insight into how the flight path changes. Following the process of reducing the aircraft contrails [300], several other forms of environmental impact could be improved such as emissions and noise, where the contribution of the weather should be considered, too. Additional measured quantities from [301] should be considered for future developments, as they are very important in the environmental sciences. Moreover, researching and developing any of the following modules will give a more pragmatic view of the problem: a legislation module will handle ATM constraints and taxation modules will calculate flight costs when crossing international airspace and when using airport infrastructure, respectively. All of these could be part of a module with constraints related to economics. One level below, fundamental additions would be to compute structural dynamics of the aircraft during flight and to carry out CFD simulations.

Of course, MOTS2 could be specifically improved for ATO in a number of ways. First of all, it should be able to perform PCA on each memory during each iteration. This will enable the optimiser to realise the importance of each component of the decision vector and to focus the search accordingly, for instance by readjusting the SSRF for each decision variable separately, as described in sub-subsection 5.5.3.1. Then, it should be modified to manipulate integer variables, because the big picture of ATM for free flight will require searching for discrete decisions such as the number of aircraft flying at a certain flight level.

Although the method was effective, higher levels of fidelity will be required for higher precision solutions. Since straight line segments seem a crude approach, refining the trajectory with more points will produce more accurate results. Trajectories could also be generated through curves, similar to the parametrisation technique used in airfoil optimisation, described in sub-subsection 3.1.7.1. Furthermore, high-fidelity CFD tools could be used to feed information to the APM about the aircraft's attributes, because they are currently too computationally intensive to employ frequently as the black box evaluations tools.

Industrial data could be integrated into the process, on aircraft that fly on a regular basis and whose trajectories are recorded for a variety of reasons. Hence, trajectory data could be used in two ways. Firstly, this information could be combined to infer trends of the trajectories like a swarm of aircraft, similar to swarm optimisation. Then, this would help to form common practices and serve as a starting point of the ATO process. In addition, it should be possible to compare how much the generated trajectory approached the real one and to assess

any benefit. Thereafter, more feedback could be provided to the process to take more information and operational needs into consideration. Certainly, it would be beneficial to integrate the computational processes with experimental tests, should they be available. Secondly, this could influence the design of FMSs so as to assist the operator to stick to the optimal course; the FMSs could also automatically perform minor corrections to the course. This will be one step closer to the integration of optimisation practices with flight control and closer to performance-based navigation.

#### 5.5.1.3 Improving Process and Procedures for Multi-Disciplinary Design Optimisation of the Microreactor

Considering the software as a system from the field of CFD, it can be combined with other principles to be used in more complicated applications. More specifically, as with most of CFD tools, LBM could be coupled with tools that can perform structural analysis so as to resolve any structural deformation, which should be provided to CFD processes, as part of fluid-structure interaction studies. Another addition would be to carry out computational aeroacoustic studies by following the method described in [135]. One step further, all of the above could be coupled with optimisation practices so as to design products with higher performance as part of multidisciplinary design optimisation studies.

Smart systems can be created by combining this application with control theory, as follows. Further expanding the idea of using a nozzle for the shape of tube downstream, another concept could change that shape dynamically, as this is relatively easy to implement with LBM. Thereafter, control theory could be employed to dynamically control the shape of the whole nozzle considering again the same objectives. Of course, this would require further developing the model of the code for unsteady flows, which could be developed because of irregular shapes. Similarly, a few of the outer holes of the baffle plate could open and close, which could also be enhanced by implementing control features for better management of energy and performance. Since the flow has to be laminar, a suggestion for the future would be to investigate more mechanisms and arrangements to create laminar flow and/or to maintain a low $Re$ as long as possible under a variety of conditions.

### 5.5.2 Expanding the Applications

The previously described enhancements could be used on new cases, too. The ultimate goal is to help shape the future of aviation. The future developments are to improve the framework and other participating modules.

#### 5.5.2.1 Future Aircraft Trajectory Studies

Using a parameterisation scheme to generate the shape of trajectories is expected to further speed up the optimisation process. Also, the discovered trends will be integrated within the new trajectories. A more realistic scenario will involve trajectories in 3D and more objectives, by including additional emission values, engines' life expectancy, and contrails path.

A promising research path is the integration of additional modules to enhance ATO. The following modules presented in [302–306] are suggested, each of which comes from different principles and could be part of a unified framework:

- Environmental externalities in air transport [303]
- Model for risk assessment [304]
- Risk and safety models for ATM operations were presented in [305], and they seem to be ideal for the concept of ATO
- At a higher level, a system that assesses global emissions from aviation was demonstrated in [306]

The authors of [155] predicted that it will be challenging to integrate modules. All this will lead to the development of a framework for aircraft conceptual design and environmental performance studies such as the one presented in [307]. However, it is important to mention that all these are expected to dramatically scale up the optimisation problem in terms of decision variables, constraints and objectives.

When it comes to studying an aircraft trajectory, more segments and flight phases could be considered so as to optimise for more realistic flight paths. By introducing additional points, more realistic trajectories could be generated. Currently, the trajectories were modelled by using a relatively small number of points. However, the dimensionality will increase and appropriate tools would be required in order to deliver sensible results within reasonable time intervals. This leads to the development of optimisation algorithms that can manage a large number of decision parameters, like GPU-MOTS2. Since the ultimate goal is to carry out optimisation in real-time, further developments that eliminate the evaluation cost are required. This can be achieved either by developing/adapting new algorithms or by employing an alternative computational infrastructure. The tools and methods employed are intended to be further improved and integrated into the FMS so as to carry out trajectory optimisation in real-time while flying.

Determining the aircraft speed in the cruise flight phase is not a trivial task because of the nature of the flight. Due to the total drag (profile and induced drag) flying the aircraft at the most optimal speed (where total drag is minimum) can be very challenging. The balance between the different types of drag in the total

drag can easily change because of variations in speed. An increase in drag causes the performance to drop and, hence, more power is required to increase aircraft speed. However, aircraft regularly fly at cruise speed higher than the speed of minimum drag curve to ensure a safe operating margin [308, 309]. Hence, special mechanisms should be included to care for speed variations when the cruise phase is encountered. Similar studies, such as [310], investigated how to reduce the environmental impact at cruise phase. Combining this along with the proposed method in this research could complement the investigation of aircraft performance.

In addition to the aforementioned points and according to the author's experience, the following suggestions will further enhance the effectiveness and applicability of the framework:

- Searching, gathering and administrating available resources
- Continuous monitoring of the allocated resources and the progress of the optimisation process
- The framework should be self-adjusted when interruptions occur
- Quick interfacing between participating modules via dedicated data exchange channels
- Operating in a batch mode
- More constraints will be inserted into the optimisation problem

An important addition to the process would be to individually modify the black box evaluation system to take advantage of the locality of partial solutions by inserting extra memory, which is separated from the optimiser. In addition, because of the nature of the problem, by using the concept of look-up tables described above, local solutions are more likely to save computational budget, independently of the optimiser. Since the trajectory consists of individual segments, whose individual performance is accumulated with some extra information exchange, it would be beneficial to store the calculated performance for each segment in a look-up table. The latter will be queried at a latter stage to avoid unnecessary re-evaluation under the same conditions, where partially stored performance for each segment could be combined with others. Local optimisers and robustness studies could be assisted by this enhancement because evaluating (and exploring) local neighbourhoods will be faster. In conjunction with the local-search features of MOTS2 this will result in a faster optimisation process, where more information could be discovered in shorter time-intervals.

The authors of [311] introduced a new way of performing robust optimisation. The initial problem was successfully transformed in optimising the system's

decision parameters subject to probabilistic criteria by employing Monte Carlo methods. The calculation technique of the Monte Carlo method and the technique to optimise the parameters determine the solutions' efficiency. Herein, a direct search was combined with the aforementioned approximations in a parallel and multilevel fashion proposing decision points that are more likely to be manufactured. The same method produced satisfactory results in aircraft trajectories, too [312].

Secondly, following the optimisation of the climb flight phase, it is sensible to optimise the cruise and the descent flight phase of the same type of aircraft for the same pair of objectives, as part of a flight within European airspace. The same compilation of tools will be used, too. The former case will continue and will stop at the last point of the climb phase and will cover twice the previous range, where the aim is to investigate what the optimum flight path and speed are. Similar trajectory modelling will be applied to a wide flight envelope, which will allow the optimiser to select either level flight or stepped cruise. The descent flight phase will continue from the end of the cruise flight phase over the same range as the climb flight phase. The reverse flight envelope from climb phase will be used so as to find optimum trajectories, where the optimiser will be allowed to select among a stepped descent, a continuous descent approach or a combination of these. The trends from the individual flight phases will be used for the next application.

Thirdly, the lessons learned from the previous applications will be used to perform ATO of a complete flight that will include climb, cruise and descent for the same type of aircraft considering the same objectives. This will be implemented in two ways: on the one hand, the same compilation of tools will be used on the scaled problem so as to investigate whether it is sensible to break the trajectory in individual phases or to simulate all of them in a single case. On the other hand, the same study for the same trajectory will be carried out on alternative tools to evaluate the performance of a trajectory because the computational cost is expected to increase exponentially. More specifically, a combination of an engine and aircraft performance model, described in [268], will be used instead. The results will be used to compare the trends of the optimal trajectories and will be used to study how the computational process could be accelerated in order to become part of an operational FMS.

#### 5.5.2.2 Future Multi-Disciplinary Design Optimisation Studies

In order to further understand the optimised environmentally-friendly trajectories that can be deployed by airlines, it is important to investigate the impact of degraded engine performance on these trajectories at a multi-disciplinary level assessing trade-offs among fuel burn, flight time, emissions and direct operating cost. This will bring environmentally sustainable and economically feasible solutions to

the operator.

From a multidisciplinary design optimisation perspective, more complicated cases could be created to design a higher quality microreactor that could combine financial modelling, structural dynamics modelling, fluid dynamics modelling, etc. As expected, adding more objectives will enable the stakeholders to analyse and to understand how the microreactor behaves on different principles. This would require expanding the existing LBM code. In addition, more decision parameters could be introduced to simulate a wider range of geometrical arrangements and flow conditions.

The current implementation can be improved in many different ways at various levels, whose significance decreases by their order of appearance. First, the most important and challenging task is to improve the computational model itself, because the other enhancements depend on this. The immediate target is to extend current LBM by adding an extra lattice for thermal properties as described in [98, 99, 102, 103]. However, the lattice size will be smaller so as to mitigate the additional data load and processing for the information to be obtained. Another worthwhile improvement, closer to the approach of finite volumes, is to extend the model as described in [100] so as to obtain more accurate results. In addition, the capability of resolving more complicated boundary conditions will strongly demonstrate the suitability of LBM to real-world applications. Thereafter, the approaches discussed in [313] and after [106] should be considered in order to increase the accuracy of the model and to make it suitable for many more applications.

Since a wide range of results has been quickly discovered, just by changing three variables, the next step would be to take advantage of the ability of LBM to easily change the computational domain of interest. Again starting from the same datum design, an optimisation process that changes additional geometrical features and flow features would be more beneficial in order to study alternative concepts. More specifically, the overall size of the reactor could be changed up to the level where $K_n$ would be practical for using the LBM. For instance, larger tube diameters could be tested. An extreme approach would be to manipulate the tube as a nozzle with convergent and divergent sections. In addition, the orientation of the baffle plate could also change at different angles, not perpendicular to the flow. Different numbers of holes can be tried, at different diameters each. Even the shape of the holes can change. Another extension would be to consider more separated chambers to simulate the flow from three or more sources; furthermore, each of the simulated fluids could be different in nature and/or in multiple phases. Perhaps the impact of adding another baffle plate or introducing an orifice plate on the interplay of objectives could be investigated. Under the current framework, all these require minimal programming effort and could be fully automated and pre-configured.

Next, the code is going to be applied to more complicated geometrical arrangements and mathematical models. An immediate application is that of a micro-combustor, where the thermal extension will also be used. Then, the performance of the current code and a future extension based on [100] will be compared against standard turbulence mathematical models from conventional CFD: Reynolds Averaged Navier-Stokes and Large Eddy Simulation. Alternatively, it could be more easily coupled to another system as part of fluid-structure interaction applications, which study the interaction between a deformed or movable structure with a fluid, and could be integrated with design optimisation applications as is the last item in section 1.5.

### 5.5.3 Improving the Tools

#### 5.5.3.1 Improving the Optimiser

As the optimiser is the core of the optimisation process that will generate data from the top level, closer to the user's understanding, the following improvements are suggested.

Obviously, MOTS2 should be compared to the most recent version of the original multi-objective tabu search. This could include a range of comparisons, where possible, so as to compare the impact of configuration parameters on the quality of the revealed trade-off, the computational efficiency/quality of the revealed trade-off when testing against benchmark when running the same applications, and any acceleration when running against GPU- MOTS2. It could also be useful to investigate what would be the benefit of integrating both versions.

Introducing the principle of robustness is easy to implement in MOTS2. Following the current structure of MOTS2, only the originally proposed decision points would be candidates for the role of BP for the next iteration. This evaluation of local decision points smoothly links with the local features of the optimiser and can be considered as an enhancement, which also links with other approaches that take advantage of the locality and similarity of solutions.

Failed objectives should be manipulated differently. Currently, all the infeasible decision points are assigned a penalty value that is a very large number, and it is almost impossible to select these points anymore during the remainder of the optimisation process. An alternative would be to locate the closest feasible decision point to the failed decision point and to investigate a few points within this distance by inserting them into the list of proposed decision points. These decision points could be flagged differently and should be treated with caution when analysing the final results.

One of the biggest challenges of optimisers is to asynchronously evaluate points. Here, it is suggested that the structure of the code be modified to create a batch

memory that will contain a number of points to be evaluated from one iteration to the other before selecting the BP. In the first implementation, this batch memory should be populated by the Hooke and Jeeves Move and the Pattern Move. Then, different methods could be integrated to propose new decision points for evaluation. The local-search functionality of MOTS2 can be maintained if the decision points in the batch memory are inserted by a mechanism that generates local points. The balance between exploration and exploitation can change if a few global decision points are inserted into the memory; if the number of global decision points is significantly lower than local decision points, the chances of selecting a global decision point as the new point are significantly slimmer. To some extent, this is another mechanism to escape from local optima.

When the number of variables becomes so big that they cannot be managed by the user, sensitivity analysis will be employed to assist the user to select the best possible axes in Parallel Coordinates that capture the behaviour of the studied system. Here, PCA would play an important role in identifying the most sensible axis to include in the analysis, based on the variability of each axis.

GPUs can assist MOTS2 by being part of a separate system that implements surrogate modelling techniques, such as Kriging, RBF, Support Vector Machines. These methods require highly intensive number crunching applications and can be significantly accelerated by GPU. GPU-MOTS2 can be particularly useful when the number of dimensions and objectives is so large that an even larger number of objective function evaluations is required. The practicality is more obvious when GPU-MOTS2 has to find solutions within very tight time restrictions; it could also be part of systems for tactical decision making. These applications vary from controlling airspace, to controlling the power distribution of utility services, to gambling companies, just to name a few. They all consist of too many variables, and there are conflicting interests between the participating entities and stakeholders. In general, it is believed that GPU-MOTS2 is good for tactical optimisation problems, where decision space is massive and the user is bound by time limits. Then, MOTS2 assisted by GPUs will work either to provide the process additional information about the objective function (for instance, by using surrogate modelling) or to calculate faster objective function(s), or (ideally) both. The optimisation process will understand more from meta-data such as the sensitivity of variables and the prediction of the objective function value(s). Nevertheless, as pointed out in [229] and demonstrated in 3.2.5, the implications of the underlying GPU in the architecture/structure of the algorithm and the overall performance of the application should be considered before making any conclusive or investment decisions.

Hybrid optimisers will have to be considered for future applications that combine features from various classes of optimisers. An instance of tabu search was

combined with genetic algorithms [314], but more generic cases, such as [186], should be investigated, too. In [315] automatic switching between 6 classical optimisers was presented. More generic moves can be created and be part of MOTS2 as part of its portfolio of moves. This will allow the optimiser to tackle different types of problems more effectively by using these problem-specific moves. This case involved both gradient-based and non-gradient-based algorithms operating in an interchangeable mode for local and global search, respectively. Another optimiser that can deal with many objectives was described in [316].

Introducing sampling techniques could have a number of advantages. In conjunction with greedy sampling schemes [317], fewer computations were needed, simulation results were predicted successfully, and the dimensionality was reduced [317]. The techniques described in [318,319] could also be used. Even effective sampling can be computationally intensive, where advanced computational systems could assist. This is true because of the nature of the techniques, which require performing operations on matrices and other types of single-instruction multiple data functionality. Hence, it is an ideal fit for GPU-based hardware and code. Different techniques from Design of Experiments could be employed for sampling purposes so as to configure more effectively an optimisation process [289,293]. Furthermore, Game Theory could be combined with intelligent sampling. This could be used to propose better points to sample (during the sampling phase). It could also assist in generating new points, where an agent will be proposing new points to be evaluated; this could either replace the Hooke and Jeeves Move, or work along with that, or be another module in the overall optimisation process to assist the optimiser. In addition to the previous method, supposing that a sampling method is provided, the user can also select more appropriate configuration settings by considering the sensitivities of decision space. This can certainly affect MOTS2's SS and SSRF. Also, by performing a statistical analysis more settings can be configured, such as the frequency of intensification, diversification and step size reduction. Furthermore, the above method can also be used to self-adapt a few of the configuration settings of the optimiser as more areas of decision space are explored. This self-adaptation can take place at any number of consecutive iterations, predefined by the user.

Two of the existing moves could be improved. Diversifying could generate a random point by using biases, from PCA applied to decision space. Alternatively, it could generate a new point within an expanded area around the BP, at a user-defined distance either by complementing or replacing STM. Very remote regions either will have less probability or will not be selected at all, which means that the next BP will not be very different than before. This links with other proposals, too.

As many local-search-based optimisers, the main disadvantage of tabu search is

the selection of the initial point. As a single decision, this will determine the overall performance of the optimiser. In addition, because the size of the decision space will be increasingly larger, this weakness has to be addressed. Therefore, it was proposed to combine the optimiser with methods from statistics and multivariate analysis, so as to enhance its functionality. The three-fold idea is to obtain a global preliminary understanding of problem space to resolve its sensitivity and then to appropriately deploy an optimiser at the most promising region so as to explore and capture the optimal behaviour.

Therefore, the whole optimisation process consists of three individual modules that are called in a sequential fashion in order to deliver the desired PF at the end phase. The first step aims at creating a sampling plan that uniformly samples decision space and will be used to obtain global information about the problem. This will help to decide the initial point and the search step before launching the optimiser. A sampling method is employed: a variant of the Latin Hypercube Sampling is enhanced with orthogonal criteria. This method evenly samples the entire decision space so as to efficiently capture the complexity of the problem. The number of points to sample is predetermined and is a fraction of the computational budget considered for the case. Usually, it should vary between 10% and 20% of that. By definition, the Latin Hypercube Sampling remembers the sampled points; an idea that naturally fits with the concept of memories presented in MOTS2.

Following the sampling plan provided by the Latin Hypercube Sampling, each of the points will be evaluated through the objective function. Then, all of them will be ranked for Pareto-dominance, where the non-dominated set will be formed. Among the fitness points of the PF, one of them will be selected randomly, and its corresponding decision variables will serve as the initial point for the next phase. If more than one set of decision variables has the same fitness, one of them will also be selected randomly. At this point, all the pairs of decision variables and fitness points will be inserted into the memories of MOTS2.

The second phase uses the PF discovered so far in order to estimate the sensitivity of the current optimal set. At the beginning of this phase, the covariance matrix of the optimal set is calculated, because the data are at a similar scale. Thereafter, PCA is applied to the covariance matrix. This returns the Principal Component Analysis, where the order of components variance reflects the importance of each component to the data set; the higher the variance of a decision variable is, the more sensitive the optimal set is when altering that variable. Hence, depending on the search state a larger or smaller step should be selected appropriately. Nevertheless, because the current data set is the optimal set so far, then the result of the second step represents the global importance of each decision variable to the problem. It is noteworthy that this is not a satisfactory estimate, because it is highly dependent on the sampled decision points. The significance

could be used either to reduce the dimensionality of the problem or to assist in the selection of the initial search step for the third phase. The latter will be applied before launching the optimisation algorithm to increase the chances of the optimiser discovering the global optimal performance of the system of interest. Several approaches have been introduced in the past, such as reducing the number of objectives [320,321] or the number of decision variables [322,323]; however, to the best of author's knowledge, none of them was applied in a MOO context to conduct environmentally-friendly studies that demonstrate the impact on the environment.

Before the start of the last step, a number of important elements are determined, which will make the optimisation search more effective. In particular, the initial point is selected to start from one of the extremes of the PF. The search step for each decision variable is individually specified and the remainder of the optimisation search is performed as normal.

#### 5.5.3.2 Improving the Flow Solver

The flow solvers could be validated and computationally improved. First, given appropriate experimental measurements with a variety of flow conditions and geometries, the same results ought to be generated via simulation. Under this condition, the code structure of GPU-LBM will be modified in order to achieve better performance, closer to the device's theoretical performance. This means optimising the memory manipulation so as to take full advantage of every computing cycle. Memory strategies could be employed [114, 122, 123] and all the levels of memory mentioned in subsection 3.3.2 could be used. Mixed precision could also be tested following the idea from [133]. Although these suggestions require more effort to be integrated by harnessing every computing cycle, they are subject to a further investigation without obstructing the aspects above.

The code should also be flexible to use. The configuration parameters could dynamically self-adapt to the problem in order to make the method easier to employ. Statistics from the resources and their operation could be used to self-adjust the important configurations (e.g., number of spawned threads, etc.); higher computational efficiency could be obtained by identifying the most active regions of the lattice, which is a feature provided directly from the hardware in the most recent GPUs. Although this seems to introduce a small overhead, it could pay off in terms of computational time required performing a converged simulation. Implementing models with less or more discrete densities will also allow one to perform trade-off studies, where the number of densities will be altered in order to appreciate the computational cost for the information benefit at desired levels of accuracy.

From the lowest practical perspective, the impact of hardware has to be con-

sidered, especially in cases where the number of evaluations per time unit and/or computational size is very large. Replacing one GPU with a more powerful one in terms of memory size and speed is expected to directly speed up the application, because the codes have been designed to take advantage of the features of the provided GPU. Therefore, this is a very straightforward change, which can be immediately appreciated. The next step would be to take advantage of the most recent features of GPU technology, where two or more GPUs can have a common address space and could transparently exchange data. This will considerably increase the performance of a single simulation.

Finally, the application of fluid-structure interaction of the work mentioned in section 1.5, could be re-implemented on GPU- LBM. The velocity magnitude and the location of three inlets on an experimental device could be optimised, so as to minimise the tangential velocity of the flow at the outlet. MOTS2 could be integrated with the same simulation code to study a more complicated geometrical arrangement with a master inlet and two secondary inlets, whose purpose is also to control the flow.

### 5.5.4 Long-Term Extensions

Based on the outcome of this work, an additional list of research questions follow so as to demonstrate the potential of extending this research:

- What is the cost of using the recently developed technology?
- What business models could be employed to enable this technology to be used by a number of stakeholders for a variety of purposes, while balancing between a sustainable development and standards to integrate systems from diverse principles?

Questions regarding the social impact are:

- What is the level of research-community trust in modelling, simulation and optimisation technology?
- What is the potential for linking different platforms and simulation systems together?

Questions regarding the economic and policy impact are:

- What would the impact of ownership model be on funding? (e.g., changes to tax/fuel revenues)
- How can one create a clear vision and policy?

- How can we link to other policies?

Questions regarding the technology are:

- How do we assess the impact of different levels of adoption within the operations of aviation?
- How do we assess the impact of different levels of integration on the aviation industry?
- How would differing performance specifications be assessed?
- Would there be a requirement for intelligent management?
- What is the threshold for manual against autonomous control?

Questions regarding the environmental impact are:

- What is the impact on noise?
- Will we be moving towards zero emissions aircraft with this technology?

# Bibliography


[1] OECD. *Globalisation, Transport and the Environment.* OECD Publishing, 2010.

[2] IATA Vision. 2050: Shaping Aviation's Future, 2011.

[3] International Energy Agency. Transport Energy and CO2. https://www.iea.org/publications/freepublications/publication/transport2009.pdf, 2009. [Online; accessed November-2014].

[4] B. Pearce. Profitability and the Air Transport Value Chain. IATA Economics Briefing No 10, 2013.

[5] IATA. The IATA Technology Roadmap Report, 2013.

[6] ATAG. The Economic and Social Benefits of Air Transport 2008. www.atag.org/component/downloads/downloads/61.html, 2008.

[7] B. Daley. *Air Transport and the Environment.* Ashgate, 2010.

[8] D. Knörzer and J. Szodruch. *Innovation for Sustainable Aviation in a Global Environment: Proceedings of the Sixth European Aeronautics Days.* IOS Press, 2012.

[9] CLEANSKY. Innovating Together, Flying Greener. [Online] Available at http://cleansky.eu/ [Accessed November 2012].

[10] J. P. Clarke. The Role of Advanced Air Traffic Management in Reducing the Impact of Aircraft Noise and Enabling Aviation Growth. *Journal of Air Transport Management*, 9(3):161–165, 2003.

[11] EUROCONTROL. Performance Review Report an Assessment of Air Traffic Management in Europe During the Calendar Year 2011. Technical Report PRR2011, EUROCONTROL, 2012.

[12] ACARE. 2008 Addendum to the Strategic Research Agenda. http://www.acare4europe.com/sites/acare4europe.org/files/document/ACARE_2008_Addendum.pdf, 2008. [Online; accessed November-2014].

[13] E. Nygren, K. Aleklett, and M. Höök. Aviation Fuel and Future Oil Production Scenarios. *Energy Policy*, 37(10):4003–4010, 2009.

[14] K. Mayor and R. S. J. Tol. Scenarios of Carbon Dioxide Emissions from Aviation. *Global Environmental Change*, 20(1):65–73, 2010.

[15] D. S. Lee, D. W. Fahey, P. M. Forster, P. J. Newton, R. C. N. Wit, L. L. Lim, B. Owen, and R. Sausen. Aviation and Global Climate Change in the 21st Century. *Atmospheric Environment*, 43(22):3520–3537, 2009.

[16] EUROCONTROL. Network Operations Report. Technical report.

[17] EUROCONTROL. An Assessment of Air Traffic Management in Europe During the Calendar Year 2012. Technical report, EUROCONTROL, 2013.

[18] ICAO. ICAO Predicts Continued Passenger Traffic Growth and Rebound in Freight Traffic Through 2016 . http://www.icao.int/Newsroom/NewsDoc2014/COM.21.14.EN.pdf, 2014. [Online; accessed November-2014].

[19] M. S. Nolan. *Fundamentals of Air Traffic Control*. Cengage Learning, 5th edition, 2010.

[20] J. E. Penner, D. H. Lister, D. J. Griggs, D. J. Dokken, and M. McFarland. Aviation and the Global Atmosphere. A Special Report of IPCC Working Groups I and III in Collaboration with the Scientific Assessment Panel to the Montreal Protocol on Substances that Deplete the Ozone Layer, Cambridge, 1999. https://www.ipcc.ch/pdf/special-reports/spm/av-en.pdf [Online; accessed November-2014].

[21] C. Barnhart, H. Jiang, and L. Marla. Optimization Approaches to Airline Industry Challenges: Airline Schedule Planning and Recovery. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[22] ICAO. Performance-Based Navigation (PBN) Manual. Technical report, ICAO, 2013.

[23] ICAO. Performance-Based Navigation (PBN) Operational Approval Manual. Technical Report 9997, ICAO, 2013.

[24] IATA. The IATA Technology Roadmap Report, 2009.

[25] AIRBUS. Sustainable Aviation - Aviation Environmental Roadmap, 2013.

[26] C. Thomas and D. Raper. The Role of Aero Engineering in the Sustainable Development of the Aviation Industry. *Aeronautical Journal*, 104(1037):331–333, 2000.

[27] K. K. Gupta, A. Rehman, and R. M. Sarviya. Bio-fuels for the Gas Turbine: A Review. *Renewable and Sustainable Energy Reviews*, 14(9):2946–2955, 2010.

[28] IATA. The IATA 2015 Report on Alternative Fuels, 2015.

[29] J. P. Slotnick, A. Khodadoust, J. J. Alonso, D. L. Darmofal, W. D. Gropp, E. A. Lurie, D. J. Mavriplis, and V. Venkatakrishnan. Enabling the Environmentally Clean Air Transportation of the Future: A Vision of Computational Fluid Dynamics in 2030. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2022), 2014.

[30] R. A. Wahls, M. M. Rogers, and K. M. Taminger. Airframe Technologies for Aerodynamic and Structural Efficiency of N+3 Subsonic Transport Aircraft. Presentation in 3rd UTIAS International Workshop on Aviation and Climate Change, electronic version at http://goldfinger.utias.utoronto.ca/IWACC3/IWACC3/Program_files/Wahls_IWACC_2012.pdf, 2012. [Online; accessed November-2014].

[31] R. A. Wahls. N+3 Technologies and Concepts. Presentation in Green Aviation Summit Integrated Solutions for Fuel, Noise and Emission Reduction, electronic version at http://www.aeronautics.nasa.gov/pdf/wahls_2_green_aviation_summit.pdf, 2010. [Online; accessed November-2014].

[32] R. A. Craig, H. O. Pritchard, F. Pompei, J. B. Heywood, R. C. Flagan, J. P. Appleton, P. Breisacher, R. J. Nichols, W. A. Hicks, M. R. Khan, and H. K. Newhall. Preprints of Conference in Fuel Composition and Its Effects on Combustion and Emission Problems, 1972.

[33] W. L. Stewart, D. L. Nored, J. S. Grobman, C. E. Feiler, and D. A. Petrash. Preparing Aircraft Propulsion for a New Era in Energy and the Environment. *Astronautics & Aeronautics*, 18(1):18–37, 1980.

[34] B. Khandelwal, A. Karakurt, P. R. Sekaran, V. Sethi, and R. Singh. Hydrogen Powered Aircraft: The Future of Air Transport. *Progress in Aerospace Sciences*, 60:45–59, 2013.

[35] S. Blakey, L. Rye, and C. W. Wilson. Aviation Gas Turbine Alternative Fuels: A Review. *Proceedings of the Combustion Institute*, 33(2):2863–2885, 2011.

[36] N. Saravanan, G. Nagarajan, G. Sanjay, C. Dhanasekaran, and K.M. Kalaiselvan. Combustion Analysis on a DI Diesel Engine with Hydrogen in Dual Fuel Mode. *Fuel*, 87(17):3591–3599, 2008.

[37] N. Saravanan and G. Nagarajan. An Experimental Investigation of Hydrogen-Enriched Air Induction in a Diesel Engine System. *International Journal of Hydrogen Energy*, 33(6):1769–1775, 2008.

[38] H. Nojoumi, I. Dincer, and G. F. Naterer. Greenhouse Gas Emissions Assesment of Hydrogen and Kerosene-Fueled Aircraft Propulsion. *International Journal Of Hydrogen Energy*, 34(3):1363–1369, 2009.

[39] G. L. Juste. Hydrogen Injection as Additional Fuel in Gas Turbine Combustor. Evaluation of Effects. *International Journal Of Hydrogen Energy*, 31(14):2112–2121, 2006.

[40] M. J. Sefain. *Hydrogen Aircraft Concepts and Ground Support*. PhD thesis, Cranfield University, UK, 2005.

[41] P. Murthy, B. Khandelwal, V. Sethi, and R. Singh. Hydrogen as a Fuel for Gas Turbine Engines with Novel Micromix Type Combustors. In *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2011.

[42] A. Contreras, S. Yiğit, K. Özay, and T. N. Veziroğlu. Hydrogen as Aviation Fuel: A Comparison with Hydrocarbon Fuels. *International Journal of Hydrogen Energy*, 22(10):1053–1060, 1997.

[43] E. Sher. *Handbook of Air Pollution From Internal Combustion Engines Pollutant Formation and Control*. Academic Press, 1998.

[44] G. Dahl and F. Suttrop. Engine Control and Low-$NO_x$ Combustion for Hydrogen Fuelled Aircraft Gas Turbines. *International Journal of Hydrogen Energy*, 23(8):695–704, 1998.

[45] B. Moghtaderi, I. Shames, and L. Djenidi. Microfluidic Characteristics of a Multi-Holed Baffle Plate Micro-Reactor. *International Journal of Heat and Fluid Flow*, 27(6):1069–1077, 2006.

[46] L. Djenidi and B. Moghtaderi. Numerical Investigation of Laminar Mixing in a Coaxial Microreactor. *Journal of Fluid Mechanics*, 568:223–242, 2006.

[47] E. M. Kraft. *After 40 Years Why Hasn't the Computer Replaced the Wind Tunnel?* Defense Technical Information Center, 2010.

[48] M. A. Ali and L. Djenidi. Lattice Boltzmann Simulation of Pulsed Jet in T-Shaped Micromixer. In *IUTAM Symposium on Advances in Micro- and Nanofluidics*, pages 167–174. Springer, 2009.

[49] A. Agrawal, L. Djenidi, and R. A. Antonia. Simulation of Gas Flow in Microchannels with a Sudden Expansion or Contraction. *Journal of Fluid Mechanics*, 530:135–144, 2005.

[50] J. Yan, editor. *Handbook of Clean Energy Systems*, volume 5. Wiley, 2015.

[51] M. Huth and A. Heilos. Fuel flexibility in gas turbine systems: impact on burner design and performance. In *Modern Gas Turbine Systems*, pages 635 – 684. Woodhead Publishing, 2013.

[52] A. H. Lefebvre and D. R. Ballal. *Gas Turbine Combustion: Alternative Fuels and Emissions*. CRC Press, third edition, 2010.

[53] A. Filippone. Comprehensive Analysis of Transport Aircraft Flight Performance. *Progress in Aerospace Sciences*, 44(3):192 – 236, 2008.

[54] M. Pidd, editor. *Systems Modelling Theory and Practice*. John Wiley & Sons, 2004.

[55] E.-G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley, 2009.

[56] J. Liu, X. Jin, and K. C. Tsui. *Autonomy Oriented Computing From Problem Solving to Complex Systems Modeling*. Kluwer Academic Publishers, 2010.

[57] M. E. Balázs, G. T. Parks, and P. J. Clarkson. Optimization in Industry-What Does Industry Really Need? In *Optimization in Industry*, pages 15–24. Springer, 2002.

[58] I. Kroo. Innovations in Aeronautics - AIAA Dryden Lecture. In *42nd AIAA Aerospace Sciences Meeting*, pages 1–11. AIAA, 5-8 January 2004.

[59] K. Deb and A. Srinivasan. Innovization: Innovative Design Principles Through Optimization. Technical Report 2005007, KanGAL, 2005.

[60] A. K. Sehra and W. Whitlow Jr. Propulsion and Power for 21st Century Aviation. *Progress in Aerospace Sciences*, 40(4–5):199–235, 2004.

[61] P. Y. Papalambros and D. J. Wilde. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, 2000.

[62] NASA. Technology Readiness Level. https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html, 2012. [Online; accessed November-2014].

[63] D. M. Jaeggi, G. T. Parks, T. Kipouros, and P. J. Clarkson. The Development of a Multi-Objective Tabu Search Algorithm for Continuous Optimisation Problems. *European Journal of Operational Research*, 185(3):1192–1212, 2008.

[64] A. D' Ammaro, T. Kipouros, S. D. Saddawi, A. M. Savill, and L. Djenidi. Computational Design for Micro Fluidic Devices Using Lattice Boltzmann and Heuristic Optimisation Algorithms. In *Joint OCCAM/ICFD Lattice Boltzmann Workshop*, 2010.

[65] C. Tsotskas. MOTS2 Code Repository. https://github.com/christos-tsotskas/MOTS2, 2015. [Online; accessed November-2014].

[66] C. R. Hargraves and S. W. Paris. Direct Trajectory Optimization Using Nonlinear Programming and Collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.

[67] D. G. Hull. Conversion of Optimal Control Problems into Parameter Optimization Problems. *Journal of Guidance, Control and Dynamics*, 20(1):57–60, 1997.

[68] J. T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[69] O. Von Stryk and R. Bulirsch. Direct and Indirect Methods for Trajectory Optimization. *Annals of Operations Research*, 37(1):357–373, 1992.

[70] J. T. Betts and E. J. Cramer. Application of Direct Transcription to Commercial Aircraft Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 18(1):151–159, 1995.

[71] J. T. Betts. Trajectory Optimization Using Sparse Sequential Quadratic Programming. In *Optimal Control*, pages 115–128. Springer, 1993.

[72] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010.

[73] J. T. Betts and W. P. Huffman. Trajectory Optimization on a Parallel Processor. *Journal of Guidance, Control, and Dynamics*, 14(2):431–439, 1991.

[74] J. T. Betts and W. P. Huffman. Application of Sparse Nonlinear Programming to Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 15(1):198–206, 1992.

[75] J. T. Betts and W. P. Huffman. Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming. *Journal of Guidance, Control, and Dynamics*, 16(1):59–68, 1993.

[76] J. T. Betts and W. P. Huffman. Large Scale Parameter Estimation Using Sparse Nonlinear Programming Methods. *SIAM Journal on Optimization*, 14(1):223–244, 2003.

[77] H. Seywald. Trajectory Optimization Based on Differential Inclusion. Technical report, NASA, 1993. NASA Contractor Report 4501.

[78] H. Seywald. Trajectory Optimization Based on Differential Inclusion (Revised). *Journal of Guidance, Control, and Dynamics*, 17(3):480–487, 1994.

[79] M. Janić. Aviation and Externalities: The Accomplishments and Problems. *Transportation Research Part D: Transport and Environment*, 4(3):159–180, 1999.

[80] Y. Y. Haimes and W. A. Hall. Sensitivity, Responsivity, Stability and Irreversibility as Multiple Objectives in Civil Systems. *Advances in Water Resources*, 1(2):71–81, 1977.

[81] C. L. Hwang, S. R. Paidy, K. Yoon, and A. S. M. Masud. Mathematical Programming with Multiple Objectives: A Tutorial. *Computers & Operations Research*, 7(1–2):5–31, 1980.

[82] F. T. Johnson, E. N. Tinoco, and N. J. Yu. Thirty Years of Development and Application of CFD at Boeing Commercial Airplanes, Seattle. *Computers & Fluids*, 34(10):1115–1151, 2005.

[83] W. Horn, S. Ardey, F. Grauer, K.-J. Schmidt, and S. Staudacher. Opportunities and Challenges for More Intelligent Gas Turbine Engines. In *Deutscher Luft-und Raumfahrtkongress*, 2008.

[84] S. Dooley, S.H. Won, J. Heyne, T.I. Farouk, Y. Ju, F.L. Dryer, K. Kumar, X. Hui, C.-J. Sung, H. Wang, M.A. Oehlschlaeger, V. Iyer, S. Iyer, T.A. Litzinger, R.J. Santoro, T. Malewicki, and K. Brezinsky. The Experimental Evaluation of a Methodology for Surrogate Fuel Formulation to Emulate Gas Phase Combustion Kinetic Phenomena. *Combustion and Flame*, 159(4):1444–1466, 2012.

[85] P. Ponticel. EADS Aims to Electrify the Skies. http://articles.sae.org/12452/, 2013. [Online; accessed November-2014].

[86] I. Yilmaz, M. Ilbaş, M. Taştan, and C. Tarhan. Investigation of Hydrogen Usage in Aviation Industry. *Energy Conversion and Management*, 63:63–69, 2012.

[87] C. G. Moniruzzaman and F. Yu. A 0D Aircraft Engine Emission Model with Detailed Chemistry and Soot Microphysics. *Combustion and Flame*, 159(4):1670–1686, 2012.

[88] J. Cain, M. J. Dewitt, D. Blunck, E. Corporan, R. Striebich, D. Anneken, C. Klingshirn, W. M. Roquemore, and R. Vander Wal. Characterization of Gaseous and Particulate Emissions from a Turboshaft Engine Burning Conventional, Alternative, and Surrogate Fuels. *Energy and Fuels*, 27(4):2290–2302, 2013.

[89] J. Burguburu, G. Cabot, B. Renou, A. Boukhalfa, and M. Cazalens. Comparisons of the Impact of Reformer Gas and Hydrogen Enrichment on Flame Stability and Pollutant Emissions for a Kerosene/Air Swirled Flame with an Aeronautical Fuel Injector. *International Journal of Hydrogen Energy*, 36(11):6925–6936, 2011.

[90] G. Marsh. Biofuels: Aviation Alternative? *Renewable Energy Focus*, 9(4):48–51, 2008.

[91] J. Jupp. Greener by Design Annual Report 2012-2013. Technical report, Royal Aeronautical Society, 2013.

[92] P. R. Spalart and D. R. Bogue. The Role of CFD in Aerodynamics, Off-Design. *The Aeronautical Journal*, 107(1072):323–329, 2003.

[93] A. Jameson and J. C. Vassberg. Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact. *AIAA*, 538:1–26, 2001.

[94] M. C. Sukop and D. T. Thorne. *Lattice Boltzmann Modeling*. Springer, 2010.

[95] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, 1st edition, 2001.

[96] A. A. Mohamad. *Lattice Boltzmann Method*. Springer, 2011.

[97] S. Chen and G. D. Doolen. Lattice Boltzmann Method for Fluid Flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[98] Y. Peng, C. Shu, and Y. T. Chew. A 3D Incompressible Thermal Lattice Boltzmann Model and its Application to Simulate Natural Convection in a Cubic Cavity. *Journal of Computational Physics*, 193(1):260–274, 2004.

[99] Y. Peng, C. Shu, and Y. T. Chew. Simplified Thermal Lattice Boltzmann Model for Incompressible Thermal Flows. *Physical Review E*, 68(2):026701–026708, 2003.

[100] C. Van Treeck, E. Rank, M. Krafczyk, J. Tölke, and B. Nachtwey. Extension of a Hybrid Thermal LBE Scheme for Large-Eddy Simulations of Turbulent Convective Flows. *Computers & Fluids*, 35(8-9):863–871, 2006.

[101] H. Huang, L. Wang, and X. Lu. Evaluation of Three Lattice Boltzmann Models for Multiphase Flows in Porous Media. *Computers & Mathematics with Applications*, 61(12):3606–3617, 2011.

[102] C.-H. Liu, K.-H. Lin, H.-C. Mai, and C.-A. Lin. Thermal Boundary Conditions for Thermal Lattice Boltzmann Simulations. *Computers & Mathematics with Applications*, 59(7):2178–2193, 2010.

[103] P. Lallemand and L.-S. Luo. Theory of the Lattice Boltzmann Method: Acoustic and Thermal Properties in Two and Three Dimensions. *Physical Review E*, 68(3):036706–036730, 2003.

[104] F. Kuznik, C. Obrecht, G. Rusaouen, and J. J. Roux. LBM Based Flow Simulation Using GPU Computing Processor. *Computers & Mathematics with Applications*, 59(7):2380–2392, 2010.

[105] C. K. Aidun and J. R. Clausen. Lattice-Boltzmann Method for Complex Flows. *Annual Review of Fluid Mechanics*, 42:439–472, 2010.

[106] H. Yu, S. S. Girimaji, and L.-S. Luo. DNS and LES of Decaying Isotropic Turbulence With and Without Frame Rotation Using Lattice Boltzmann Method. *Journal of Computational Physics*, 209(2):599–616, 2005.

[107] H. Wang, K. Luo, S. Lu, and J. Fan. Direct Numerical Simulation and Analysis of a Hydrogen/Air Swirling Premixed Flame in a Micro Combustor. *International Journal of Hydrogen Energy*, 36(21):13838–13849, 2011.

[108] Conference Review of 2nd International Conference on Advances in Computational Modeling and Simulation. In R. Guo, Y. Zeng, J. Zhang, Y. Yang, Z. Li, and T. Guo, editors, *Applied Mechanics and Materials*, volume 444–445. Trans Tech Publications, October 2014.

[109] Conference Review. In *20th AIAA/CEAS Aeroacoustics Conference*, 2014.

[110] J. Beisheim. Speed Up Simulations with a GPU. http://www.ansys.com/staticassets/ANSYS/staticassets/resourcelibrary/article/AA-V4-I2-Speed-Up-Simulation-with-GPU.pdf, 2010. [Online; accessed November-2014].

[111] R. Strzodka. Accelerated ANSYS Fluent: Algebraic Multigrid on a GPU. http://on-demand.gputechconf.com/supercomputing/2012/presentation/SB030-Strzodka-Accelerated-ANSYS-Fluent.pdf, 2012. [Online; accessed November-2014].

[112] EXA. The Technology Behind Power FLOW: Exa's Lattice Boltzmann-Based Physics. http://www.exa.com/core-technology.html, 2014. [Online; accessed November-2014].

[113] NVIDIA Corporation. What is CUDA?, 2012.

[114] J. Dongarra, G. Peterson, S. Tomov, J. Allred, V. Natoli, and D. Richie. Exploring New Architectures in Accelerating CFD for Air Force Applications. In *DoD HPCMP Users Group Conference*, pages 472–478. IEEE, 2008.

[115] S. Tomov, M. Dongarra, and M. Baboulin. Towards Dense Linear Algebra for Hybrid GPU Accelerated Manycore Systems. *Parallel Computing*, 36(5–6):232–240, 2010.

[116] C. Bekas and A. Curioni. A New Energy Aware Performance Metric. *Computer Science - Research and Development*, 25(3):187–195, 2010.

[117] A. A. Davydov, B. N. Chetverushkin, and E. V. Shil'nikov. Simulating Flows of Incompressible and Weakly Compressible Fluids on Multicore Hybrid Computer Systems. *Computational Mathematics and Mathematical Physics*, 50(12):2157–2165, 2010.

[118] P. R. Rinaldi, E. A. Dari, M. J. Venere, and A. Clausse. A Lattice-Boltzmann Solver for 3D Fluid Simulation on GPU. *Simulation Modelling Practice and Theory*, 25(0):163–171, 2012.

[119] J. Habich, T. Zeiser, G. Hager, and G. Wellein. Performance Analysis and Optimization Strategies for a D3Q19 Lattice Boltzmann Kernel on NVIDIA GPUs Using CUDA. *Advances in Engineering Software*, 42(5):266–272, 2011.

[120] J. Linxweiler, M. Krafczyk, and J. Tölke. Highly Interactive Computational Steering for Coupled 3D Flow Problems Utilizing Multiple GPUs. *Computing and Visualization in Science*, 13(7):299–314, 2010.

[121] C. Obrecht, F. Kuznik, B. Tourancheau, and J. J. Roux. A New Approach to the Lattice Boltzmann Method for Graphics Processing Units. *Computers & Mathematics with Applications*, 61(12):3628–3638, 2011.

[122] J. Tölke. Implementation of a Lattice Boltzmann Kernel Using the Compute Unified Device Architecture Developed by NVIDIA. *Computing and Visualization in Science*, 13(1):29–39, 2009.

[123] J. Tölke and M. Krafczyk. TeraFLOP Computing on a Desktop PC with GPUs for 3D CFD. *International Journal of Computational Fluid Dynamics*, 22(7):443–456, 2008.

[124] J. Myre, S. D. C. Walsh, D. Lilja, and M. O. Saar. Performance Analysis of Single-Phase, Multiphase, and Multicomponent Lattice-Boltzmann Fluid Flow Simulations on GPU Clusters. *Concurrency and Computation: Practice and Experience*, 23(4):332–350, 2011.

[125] J. Habich, C. Feichtinger, H. Köstler, G. Hager, and G. Wellein. Performance Engineering for the Lattice Boltzmann Method on GPGPUs: Architectural Requirements and Performance Results. *Computers & Fluids*, 80(0):276–282, 2013.

[126] Y. Zhao, L. Wang, F. Qiu, A. Kaufman, and K. Mueller. Melting and Flowing in Multiphase Environment. *Computers & Graphics*, 30(4):519–528, 2006.

[127] H. Yu, R. Chen, H. Wang, Z. Yuan, Y. Zhao, Y. An, Y. Xu, and L. Zhu. GPU Accelerated Lattice Boltzmann Simulation for Rotational Turbulence. *Computers & Mathematics with Applications*, 67(2):445–451, 2014.

[128] K. R. Tubbs and F. T. C Tsai. GPU Accelerated Lattice Boltzmann Model for Shallow Water Flow and Mass Transport. *International Journal for Numerical Methods in Engineering*, 86(3):316–334, 2011.

[129] C. Feichtinger, J. Habich, H. Köstler, G. Hager, U. Rüde, G. Wellein, and G. Hager. A Flexible Patch-Based Lattice Boltzmann Parallelization Approach for Heterogeneous GPU–CPU Clusters. *Parallel Computing*, 37(9):536–549, 2011.

[130] T. Tomczak, K. Zadarnowska, Z. Koza, M. Matyka, and Ł. Mirosław. Acceleration of Iterative Navier-Stokes Solvers on Graphics Processing Units. *International Journal of Computational Fluid Dynamics*, 27(4-5):201–209, 2013.

[131] D. Göddeke. *Fast and Accurate Finite-Element Multigrid Solvers for PDE Simulations on GPU Clusters*. Logos Verlag, 2011.

[132] M. Giles. OP2 Project Page. http://www.oerc.ox.ac.uk/projects/op2, 2011. [Online; accessed November-2014].

[133] I. C. Kampolis, X. S. Trompoukis, V. G. Asouti, and K. C. Giannakoglou. CFD-Based Analysis and Two-Level Aerodynamic Optimization on Graphics Processing Units. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):712–722, 2010.

[134] T. Tomczak, K. Zadarnowska, Z. Koza, M. Matyka, and Ł. Mirosław. Complete PISO and SIMPLE Solvers on Graphics Processing Units. *CoRR*, abs/1207.1571, 2012.

[135] S. Marié, D. Ricot, and P. Sagaut. Comparison Between Lattice Boltzmann Method and Navier-Stokes High Order Schemes for Computational Aeroacoustics. *Journal of Computational Physics*, 228(4):1056–1070, 2009.

[136] J. D. Anderson. *Hypersonic and High-Temperature Gas Dynamics*. AIAA, 2006.

[137] A. D' Ammaro. Computational design for micro fluidic devices using lattice boltzmann and heuristic optimisation algorithms. Master's thesis, Cranfield University, UK, 2010.

[138] J. Baker, M. E. Calvert, D. J. Power, E. T. Chen, M. L. Ramalingam, and T. R. Lamp. On the Role of the Knudsen Number with Respect to Heat Transfer in Micro-Scale Flows. In *Proceedings of the 31st Intersociety Energy Conversion Engineering Conference*, volume 2, pages 1396–1401. IEEE, 1996.

[139] S. S. Mehendale, A. M. Jacobi, and R. K. Shah. Fluid Flow and Heat Transfer at Micro- and Meso-Scales with Application to Heat Exchanger Design. *Applied Mechanics Reviews*, 53(7):175–193, 2000.

[140] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane. Optimization Using Surrogate Models and Partially Converged Computational Fluid Dynamics Simulations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 462, pages 2177–2204. The Royal Society, 2006.

[141] A. Arias-Montano, C. A. C. Coello, and E. Mezura-Montes. Multiobjective Evolutionary Algorithms in Aeronautical and Aerospace Engineering. *IEEE Transactions on Evolutionary Computation*, 16(5):662–694, Oct 2012.

[142] W. Zhu, J. Curry, and A. Marquez. SIMD Tabu Search for the Quadratic Assignment Problem with Graphics Hardware Acceleration. *International Journal of Production Research*, 48(4):1035–1047, 2010.

[143] A. Cook. *European Air Traffic Management : Principles, Practice and Research.* Ashgate, 2007.

[144] G. C. Bower and I. M. Kroo. Multi-Objective Aircraft Optimization for Minimum Cost and Emissions Over Specific Route Networks. In *The 26th Congress of ICAS and 8th AIAA ATIO*, 2008.

[145] N. E. Antoine and I. M. Kroo. Aircraft Optimization for Minimal Environmental Impact. *Journal of Aircraft*, 41(4):790–797, 2004.

[146] A. Barkana and C. Gerald. An Aircraft Noise Pollution Model for Trajectory Optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 12(2):109–116, 1976.

[147] R. Coppenbarger, R. Lanier, D. Sweet, and S. Dorsky. Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering. In *AIAA Guidance Navigation and Control Conference and Exhibit*, pages 1–19, 2004.

[148] S. M. Green and R. A. Vivona. En Route Descent Advisor Concept For Arrival Metering. In *AIAA Guidance, Navigation and Control Conference*, 2001.

[149] K. T. Mueller, R. Bortins, D. R. Schleicher, and D. Sweet. Effect of Uncertainty on En Route Descent Advisor (EDA) Predictions. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, 2004.

[150] T. J. Weidner. Benefits Assessment Compilation for the En Route Descent Advisor (EDA) AATT Decision Support Tool. *Seagull TR98188. 52f, NASS AATT TO52.(December 2000)*, 2000.

[151] M. P. Fanti, G. Pedroncelli, G. Stecco, and W. Ukovich. Modeling and Optimization of Aircraft Trajectories: A Review. In *7th International Conference on System of Systems Engineering*, pages 235–240, 2012.

[152] R. P. Henderson, J. R. R. A. Martins, and R. E. Perez. Aircraft Conceptual Design for Optimal Environmental Performance. *The Aeronautical Journal*, 116(1175):1–22, 2012.

[153] F. Escarti. Boeing Research and Technology Europe: Can Europe Compete as a Research Location? http://locomotive-project.org/cms/Content/download/05_LOCOMOTIVE_Escarti_070605.pdf, 2007. [Online; accessed November-2014].

[154] Q. McEnteggart and J. Whidborne. A Multiobjective Trajectory Optimisation Method for Planning Environmentally Efficient Trajectories. In *UKACC International Conference on Control*, pages 128–135, 2012.

[155] L. F. Vismari and J. B. Camargo Jr. Evaluation of the Impact of New Technologies on Aeronautical Safety: An Approach Through Modelling, Simulation and Comparison with Legacy Systems. *Journal of the Brazilian Air Transportation Research Society*, 1:19–30, 2005.

[156] AIRBUS. *Getting to Grips with Aircraft Performance*. AIRBUS Customer Services, 2002.

[157] Eurostat. Database by Themes: Air Transport Measurement-Traffic Data by Airports, Aircrafts and Airlines. Table Aircraft Traffic Data by Reporting Country from http://epp.eurostat.ec.europa.eu/portal/page/portal/transport/data/database, 2012. [Online; accessed November-2014].

[158] A. Gardi, R. Sabatini, S. Ramasamy, and T. Kistan. Real-Time Trajectory Optimisation Models for Next Generation Air Traffic Management Systems. In *Applied Mechanics and Materials*, volume 629, pages 327–332, 2014.

[159] C. Jun and S. Polite. Planning Aircraft Taxiing Trajectories via a Multi-Ojective Immune Optimisation. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, volume 4, pages 2235–2240. IEEE, 2011.

[160] H. Pervier, D. Nalianda, R. Espi, V. Sethi, P. Pilidis, D. Zammit-Mangion, J. M Rogero, and R. Entz. Application of Genetic Algorithm for Preliminary Trajectory Optimization. *SAE International Journal of Aerospace*, 4(2):973–987, 2011.

[161] J. M. Rogero. *A Genetic Algorithm based Optimisation tool for preliminary design of Gas Turbine Combustors*. PhD thesis, Cranfield University, UK, 2002.

[162] M. Sammut, D. Zammit-Mangion, and R. Sabatini. Optimization of Fuel Consumption in Climb Trajectories Using Genetic Algorithm Techniques. In *AIAA Guidance, Navigation, and Control Conference*, 2012.

[163] K. Chircop, W. Camilleri, V. Sethi, and D. Zammit-Mangion. Multi-Objective Optimisation of a Constrained 2000 km Trajectory Using Genetic Algorithms. In *3rd CEAS Air & Space Conference*, 2011.

[164] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[165] K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9(2):115–148, 1995.

[166] K. Chircop, M. Xuereb, D. Zammit-Mangion, and E. Cachia. A Generic Framework For Multi-Parameter Optimization of Flight Trajectories. In *27th Congress of the International Council of the Aeronautical Sciences*, 2010.

[167] M. Xuereb, M. Sammut, K. Chircop, and D. Zammit-Mangion. GATAC–Version 3 of the Generic Framework for Multi-Parameter Optimization of Flight Trajectories. In *AIAA Modeling and Simulation Technologies Conference*, 2013.

[168] A. O. Salas and J. C. Townsend. Framework Requirements for MDO Application Development. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 4740, page 1998. AIAA, 1998.

[169] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[170] C. Celis. *On Evaluation and Optimisation of Environmentally Friendly Aircraft Systems.* PhD thesis, Cranfield University, UK, 2010.

[171] C. Celis, V. Sethi, R. Singh, and P. Pilidis. On Optimisation of Environmentally Friendly Aircraft Engine Cycles. *Journal of Aerospace Technology and Management*, 7(2):143–156, 2015.

[172] H. Zolata. Development of Multidisciplinary Aircraft Trajectory Optimisation Simulation Framework. Master's thesis, Cranfield University, UK, 2009.

[173] R. M. Espi. Benchmarking and Testing of Different Genetic Algorithms for Multidisciplinary Aircraft Trajectory Optimisation Studies. Master's thesis, Cranfield University, UK, 2010.

[174] H. G. Visser. Generic and Site-Specific Criteria in the Optimization of Noise Abatement Trajectories. *Transportation Research Part D: Transport and Environment*, 10(5):405–419, 2005.

[175] H. G. Visser. Environmentally Optimized Resolutions of In-Trail Separation Conflicts for Arrival Flights. *Journal of Aircraft*, 45(4):1198–1205, 2008.

[176] H. G. Visser and R. A. A. Wijnen. Optimization of Noise Abatement Departure Trajectories. *Journal of Aircraft*, 38(4):620–627, 2001.

[177] H. G. Visser and R. A. A. Wijnen. Optimisation of Noise Abatement Arrival Trajectories. *Aeronautical Journal*, 107(1076):607–615, 2003.

[178] R. A. A. Wijnen and H. G. Visser. Optimal Departure Trajectories with Respect to Sleep Disturbance. *Aerospace Science and Technology*, 7(1):81–91, 2003.

[179] J. Quehl and M. Basner. Annoyance from Nocturnal Aircraft Noise Exposure: Laboratory and Field-Specific Dose-Response Curves. *Journal of Environmental Psychology*, 26(2):127–140, 2006.

[180] W. Babisch, D. Houthuijs, G. Pershagen, E. Cadum, K. Katsouyanni, M. Velonakis, M.-L. Dudley, H.-D. Marohn, W. Swart, O. Breugelmans, G. Bluhm, J. Selander, F. Vigna-Taglianti, S. Pisani, A. Haralabidis, K. Dimakopoulou, I. Zachos, and L. Jarup. Annoyance Due to Aircraft Noise has Increased Over the Years-Results of the HYENA Study. *Environment International*, 35(8):1169–1176, 2009.

[181] P. Hagelauer and F. Mora-Camino. A Soft Dynamic Programming Approach for On-Line Aircraft 4D-Trajectory Optimization. *European Journal of Operational Research*, 107(1):87–95, 1998.

[182] C. A. C. Coello, G. B. Lamont, and D. A. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition, 2007.

[183] K. Deb. *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[184] A. J. Keane and P. B. Nair. *Computational Approaches to Aerospace Design: The Pursuit of Excellence*. J. Wiley & Sons, 2005.

[185] D. F. Jones, S. K. Mirrazavi, and M. Tamiz. Multi-Objective Meta-Heuristics: An Overview of the Current State-Of-The-Art. *European Journal of Operational Research*, 137(1):1–9, 2002.

[186] M. J. Colaço, H. R. B. Orlande, G. S. Dulikravich, and T. J. Martin. Hybrid Optimization with Automatic Switching Among Optimization Algorithms. *Evolutionary Algorithms and Intelligent Tools in Engineering Optimization*, pages 92–118, 2005.

[187] F. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. Springer, 2003.

[188] M. Gendreau and J.-Y. Potvin. *Handbook of Metaheuristics*, volume 2. Springer, 2010.

[189] C. Blum and A. Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[190] J. Dréo, A. Petrowski, P. Siarry, and E. Taillard. *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer Science & Business Media, 2006.

[191] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. A Survey on Metaheuristics for Stochastic Combinatorial Optimization. *Natural Computing: An International Journal*, 8(2):239–287, 2009.

[192] M. Geilen, T. Basten, B. Theelen, and R. Otten. An Algebra of Pareto Points. *Fundamenta Informaticae*, 78(1):35–74, 2007.

[193] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[194] L. Bui, K. Deb, H. Abbass, and D. Essam. Interleaving Guidance in Evolutionary Multi-Objective Optimization. *Journal of Computer Science and Technology*, 23(1):44–63, 2008.

[195] F. Freschi, C. A. C. Coello, and M. Repetto. *Multiobjective Optimization and Artificial Immune Systems: A Review*, pages 1–21. Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies. IGI Global, 2009.

[196] M. López-Ibáñez, L. Paquete, and T. Stützle. *Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization*, pages 209–222. Experimental Methods for the Analysis of Optimization Algorithms. Springer, 2010.

[197] J. Arora. *Introduction to Optimum Design.* Academic Press, 3rd edition, 2011.

[198] T. Kipouros, D. M. Jaeggi, W. N. Dawes, G. T. Parks, A. M. Savill, and P. J. Clarkson. Biobjective Design Optimization for Axial Compressors Using Tabu Search. *AIAA Journal*, 46(3):701–711, 2008.

[199] T. Kipouros, D. M. Jaeggi, W. N. Dawes, G. T. Parks, A. M. Savill, and P. J. Clarkson. Insight into High-quality Aerodynamic Design Spaces Through Multi-Objective Optimization. *Computer Modeling in Engineering and Sciences (CMES)*, 37(1):1–44, 2008.

[200] T. Ghisu, J. P. Jarrett, and G. T. Parks. Robust Design Optimization of Airfoils with Respect to Ice Accretion. *Journal of Aircraft*, 48(1):287–304, 2011.

[201] T. Ghisu, G. T. Parks, J. P. Jarrett, and P. J. Clarkson. Robust Design Optimization of Gas Turbine Compression Systems. *Journal of Propulsion and Power*, 27(2):282–295, 2011.

[202] S. Russell. *Artificial Intelligence : A Modern Approach.* Prentice Hall, 3rd edition, 2010.

[203] F. Glover. Tabu Search and Adaptive Memory Programming-Advances, Applications and Challenges. In *Interfaces in Computer Science and Operations Research*, pages 1–75. Springer, 1997.

[204] É. D. Taillard, L. M. Gambardella, M. Gendreau, and J.-Y. Potvin. Adaptive Memory Programming: A Unified View of Metaheuristics. *European Journal of Operational Research*, 135(1):1–16, 2001.

[205] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, 3rd printing edition, 1999.

[206] IEEE. Systems and Software Engineering-Life Cycle Processes-Requirements Engineering. International Standard ISO/IEC/IEEE 29148:2011(E), IEEE, 2011.

[207] I. Sommerville. *Software Engineering.* Addison-Wesley, 9th edition, 2010.

[208] A. Ertas and J. C. Jones. *The Engineering Design Process*. Wiley, 2nd edition, 1996.

[209] T. G. Crainic, M. Gendreau, and J.-Y. Potvin. *Parallel Metaheuristics: A New Class of Algorithms*, chapter 13 Parallel Tabu Search, pages 289–313. John Wiley & Sons, 2005.

[210] J. Brownlee. *Clever Algorithms Nature-Inspired Programming Recipes*. Lulu Enterprises, 2011.

[211] R. Battiti and G. Tecchiolli. The Reactive Tabu Search. *ORSA Journal on Computing*, 6(2):126–140, 1994.

[212] D. M. Jaeggi, C. Asselin-Miller, G. T. Parks, T. Kipouros, T. Bell, and J. Clarkson. Multi-Objective Parallel Tabu Search. In *Parallel Problem Solving from Nature VIII*, pages 732–741. Springer, 2004.

[213] A. M. Connor and D. G. Tilley. A Tabu Search Method for the Optimization of Fluid Power Circuits. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 212(5):373–381, 1998.

[214] M. Gendreau, P. Badeau, F. Guertin, J. Y. Potvin, and E. Taillard. A Solution Procedure for Real-Time Routing and Dispatching of Commercial Vehicles. In *Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport Systems*, 1996.

[215] L. Jourdan, C. Dhaenens, and E.-G. Talbi. Using Datamining Techniques to Help Metaheuristics: A Short Survey. In *Hybrid Metaheuristics*, pages 57–69. Springer, 2006.

[216] R. Hooke and T. A. Jeeves. Direct Search Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[217] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the Hypervolume Indicator: Optimal $\mu$-Distributions and the Choice of the Reference Point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, pages 87–102. ACM, 2009.

[218] O. Mersmann, H. Trautmann, B. Naujoks, and C. Weihs. Benchmarking Evolutionary Multiobjective Optimization Algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[219] J. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2005.

[220] D. Nalianda Karumbaiah. *Impact of Environmental Taxation Policies on Civil Aviation-A Techno-Economic Environmental Risk Assessment*. PhD thesis, Cranfield University, UK, 2012.

[221] B. R. Jones, W. A. Crossley, and A. S. Lyrintzis. Aerodynamic and Aeroacoustic Optimization of Rotorcraft Airfoils via a Parallel Genetic Algorithm. *Journal of Aircraft*, 37(6):1088–1096, 2000.

[222] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *SIGGRAPH Computer Graphics*, 20(4):151–160, 1986.

[223] M. Drela. XFOIL Subsonic Airfoil Development System. http://web.mit.edu/drela/Public/web/xfoil/, 2012. [Online; accessed November-2014].

[224] M. Drela. XFOIL-An Analysis and Design System for Low Reynolds Number Airfoils. In *Low Reynolds Number Aerodynamics*, pages 1–12. Springer, 1989.

[225] E. Alba. *Parallel Metaheuristics: A New Class of Algorithms*, volume 47. John Wiley & Sons, 2005.

[226] E. Alba and G. Luque. *Parallel Metaheuristics: A New Class of Algorithms*, chapter 2 Measuring the Performance of Parallel Metaheuristics, pages 43–62. John Wiley & Sons, 2005.

[227] E. Alba and A. J. Nebro. *Parallel Metaheuristics: A New Class of Algorithms*, chapter 3 New Technologies in Parallelism, pages 63–78. John Wiley & Sons, 2005.

[228] E. Alba, E.-G. Talbi, G. Luque, and N. Melab. *Parallel Metaheuristics: A New Class of Algorithms*, chapter 4 Metaheuristics and Parallelism, pages 79–103. John Wiley & Sons, 2005.

[229] L. Mussi, F. Daolio, and S. Cagnoni. Evaluation of Parallel Particle Swarm Optimization Algorithms within the CUDA™ Architecture. *Information Sciences*, 181(20):4642–4657, 2011.

[230] Q. Zhang, A. Zhou, S.-Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-887, University of Essex, Singapore

and Nanyang Technological University, Singapore, 2008. Special Session on Performance Assessment of Multi-Objective Optimization Algorithms.

[231] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation*, pages 825–830. IEEE Press, 2002.

[232] S. Huband, L. Barone, L. While, and P. Hingston. A Scalable Multi-Objective Test Problem Toolkit. In *Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer, 2005.

[233] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[234] M. M. Woolard and J. E. Fieldsend. On the Effect of Selection and Archiving Operators in Many-Objective Particle Swarm Optimisation. In *Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pages 129–136. ACM, 2013.

[235] T. Van Luong, N. Melab, and E.-G. Talbi. GPU-Based Multi-Start Local Search Algorithms. In *Learning and Intelligent Optimization*, pages 321–335. Springer, 2011.

[236] C. Schulz. Efficient Local Search on the GPU-Investigations on the Vehicle Routing Problem. *Journal of Parallel and Distributed Computing*, 73(1):14–31, 2013.

[237] F. Luna, D. L. González-Álvarez, F. Chicano, and M. A. Vega-Rodríguez. On the Scalability of Multi-Objective Metaheuristics for the Software Scheduling Problem. In *11th International Conference on Intelligent Systems Design and Applications*, pages 1110–1115. IEEE, 2011.

[238] J. J. Durillo, A. J. Nebro, C. A. Coello Coello, J. García-Nieto, F. Luna, and E. Alba. A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems. *IEEE Transactions on Evolutionary Computation*, 14(4):618–635, 2010.

[239] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[240] S.-Y. Ho, L.-S. Shu, and J.-H. Chen. Intelligent Evolutionary Algorithms for Large Parameter Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 8(6):522–541, 2004.

[241] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra. Graphics Processing Unit (GPU) Programming Strategies and Trends in GPU Computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13, 2013.

[242] T. van Luong, N. Melab, and E.-G. Talbi. GPU Computing for Parallel Local Search Metaheuristic Algorithms. *IEEE Transactions on Computers*, 62(1):173–185, 2013.

[243] NVIDIA Corporation. *CUDA C Best Practices Guide*. NVIDIA, 5.0 edition, 2012.

[244] Y. S. G. Nashed, R. Ugolotti, P. Mesejo, and S. Cagnoni. libCudaOptimize: An Open Source Library of GPU-Based Metaheuristics. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion*, pages 117–124. ACM, 2012.

[245] N. Melab, T. Van Luong, K. Boufaras, and E.-G. Talbi. ParadisEO-MO-GPU: A Framework for Parallel GPU-Based Local Search Metaheuristics. In *Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pages 1189–1196. ACM, 2013.

[246] NVIDIA Corporation. *CUDA C Programming Guide*. NVIDIA, 5.0 edition, 2012.

[247] N. Delbosc, J. L. Summers, A. I. Khan, N. Kapur, and C. J. Noakes. Optimized Implementation of the Lattice Boltzmann Method on a Graphics Processing Unit Towards Real-Time Fluid Simulation. *Computers & Mathematics with Applications*, 67(2):462–475, 2014.

[248] L. Djenidi. Lattice-Boltzmann Simulation of Grid-Generated Turbulence. *Journal of Fluid Mechanics*, 552:13–35, 2006.

[249] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review*, 94(3):511–525, 1954.

[250] G. Wellein, T. Zeiser, G. Hager, and S. Donath. On the Single Processor Performance of Simple Lattice Boltzmann Kernels. *Computers & Fluids*, 35(8-9):910–919, 2006.

[251] K. Naik and P. Tripathy. *Software testing and quality assurance: theory and practice*. John Wiley & Sons, 2011.

[252] S. McConnell. *Rapid development: taming wild software schedules*. Pearson Education, 1996.

[253] Y. A. Çengel, R. H. Turner, J. M. Cimbala, and M. Kanoglu. *Fundamentals of thermal-fluid sciences*, chapter 8.3. McGraw-Hill New York, NY, 2005.

[254] F. Gray and E. Boek. Enhancing Computational Precision for Lattice Boltzmann Schemes in Porous Media Flows. *Computation*, 4(1):11, 2016.

[255] ICAO. ICAO Circular 303-AN/176. http://www.icao.int/environmental-protection/Documents/Publications/Chapter6.website.v3.en.pdf, 2004. [Online; accessed November-2014].

[256] ICAO. *Environmental Protection*, third edition, 2008. Volume II Aircraft Engine Emissions.

[257] N. M. Mazlan. *Assessing/Optimising Bio-Fuel Combustion Technologies for Reducing Civil Aircraft Emissions*. PhD thesis, Cranfield University, UK, 2012.

[258] W. Camilleri, K. Chircop, D. Zammit-Mangion, R. Sabatini, and V. Sethi. Design and Validation of a Detailed Aircraft Performance Model for Trajectory Optimization. In *AIAA Modeling and Simulation Technologies Conference*, pages 13–16. AIAA, 2012.

[259] J. R. Palmer. The Turbomatch Scheme for Gas-Turbine Performance Calculations, User's Guide, Cranfield University, UK, 1999.

[260] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2nd edition, 2010.

[261] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, 2009.

[262] J. R. Palmer and V. Pachidis. The Turbomatch Scheme for Aero/Industrial Gas Turbine Engine Design Point/Off Design Performance Calculation. *SME, Thermal Power Group, Cranfield University*, 1990.

[263] P. Giannakakis. Hermes V5 and TmatchCalls V3. User Manual, 2009.

[264] CFM. The Technology Behind the CFM56-5B Turbofan Engine, 2011. [Online; accessed September-2014].

[265] W. Gu, R. Navaratne, D. Quaglia, Y. Yu, I. Madani, V. Sethi, H. Jia, K. Chircop, R. Sabatini, and D. Zammit-Mangion. Towards the Development of a Multi-Disciplinary Flight Trajectory Optimization Tool: GATAC. In *ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, pages 415–424. American Society of Mechanical Engineers, 2012.

[266] R. Navaratne, W. Camilleri, E. Najafi, V. Sethi, and P. Pilidis. Preliminary Aero Engine Life Assessment via Techno-Economic Environmental Risk Analysis. In *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition*, pages V06BT43A012–V06BT43A012. American Society of Mechanical Engineers, 2013.

[267] European Aviation Safety Agency. ICAO Engine Exhaust Emissions Data Bank Subsonic Engines CFM56-5B4 Unique ID Number: 2CM014, 2013. [Online; accessed September-2014].

[268] P. Laskaridis, P. Pilidis, and P. Kotsiopoulos. An Integrated Engine-Aircraft Performance Platform for Assessing New Technologies in Aeronautics. In *Proceedings of the 17th International Symposium on Air Breathing Engines*, 2005.

[269] P. Laskaridis. *Performance Investigations and Systems Integration for the More Electric Aircrafts*. PhD thesis, Cranfield University, UK, 2004.

[270] L. R. Jenkinson, P. Simpkin, D. Rhodes, L. R. Jenkison, and R. Royce. *Civil Jet Aircraft Design*, volume 7. Arnold London, 1999.

[271] European Aviation Safety Agency. ICAO Engine Exhaust Emissions Data Bank Subsonic Engines CFM56-5B4 Unique ID Number: 3CM034, 2013. [Online; accessed September-2014].

[272] P. D. Norman, D. H. Lister, M. Lecht, P. Madden, K. Park, O. Penanhoat, C. Plaisance, and K. Renger. Development of the Technical Basis for a New Emissions Parameter Covering the Whole Aircraft Operation: NEPAIR, Final Technical Report. (NEPAIR/WP4/WPR/01, and CAEP/6-IP/17 Appendix B), September 2003.

[273] H. Pervier. *Emissions Modelling for Engine Cycle and Aircraft Trajectory Optimisation*. PhD thesis, Cranfield University, UK, 2013.

[274] S. G. Park and J.-P. Clarke. Vertical Trajectory Optimization to Minimize Environmental Impact in the Presence of Wind. *arXiv preprint arXiv:1508.01829*, 2015.

[275] A. Degroote. Assessment of the Feasibility of Controlling the Outlet Temperature Distribution of a Novel Micromix Combustor. Master's thesis, Cranfield University, UK, 2014.

[276] S. P. Dawson, S. Chen, and G. D. Doolen. Lattice Boltzmann Computations for Reaction-Diffusion Equations. *The Journal of Chemical Physics*, 98(2):1514–1523, 1993.

[277] S. Chen, S. P. Dawson, G. D. Doolen, D. R. Janecky, and A. Lawniczak. Lattice Methods and Their Applications to Reacting Systems. *Computers & Chemical Engineering*, 19(6):617–646, 1995.

[278] K. Yamamoto, N. Takada, and M. Misawa. Combustion Simulation with Lattice Boltzmann Method in a Three-Dimensional Porous Structure. *Proceedings of the Combustion Institute*, 30(1):1509–1515, 2005.

[279] K. Yamamoto, X. He, and G. D. Doolen. Simulation of Combustion Field with Lattice Boltzmann Method. *Journal of Statistical Physics*, 107(1-2):367–383, 2002.

[280] European Commission. H2Aircraft - CRYOPLANE and the Future of Flight. http://ec.europa.eu/research/transport/projects/items/h2aircraft___cryoplane_and_the_future_of_flight_en.htm, 2000. [Online; accessed November-2014].

[281] J. W. Slater. Tutorial on CFD Verification and Validation. http://www.grc.nasa.gov/WWW/wind/valid/tutorial/tutorial.html, 2008. [Online; accessed November-2014].

[282] O. Malaspinas. Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization. *arXiv preprint arXiv:1505.06900*, 2015.

[283] J. R. San Cristóbal. *Multi Criteria Analysis in the Renewable Energy Industry*. Springer Science & Business Media, 2012.

[284] S. Greco, K. Klamroth, J. D. Knowles, and G. Rudolph. Understanding Complexity in Multiobjective Optimization (Dagstuhl Seminar 15031). *Dagstuhl Reports*, 5(1):96–163, 2015.

[285] W. C. M. Van Beers and J. P. C. Kleijnen. Kriging Interpolation in Simulation: A Survey. In *Proceedings of the 2004 Winter Simulation Conference*, volume 1. IEEE, 2004.

[286] J. P. C. Kleijnen. Kriging Metamodeling in Simulation: A Review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[287] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support Vector Machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[288] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[289] F. Kai-Tai. *Design and Modeling for Computer Experiments*. Chapman and Hall CRC, 1st edition, 2006.

[290] K. E. Kerry and K. A. Hawick. Kriging Interpolation on High-Performance Computers. In *High-Performance Computing and Networking*, pages 429–438. Springer, 1998.

[291] L. Pesquer, A. Cortés, and X. Pons. Parallel Ordinary Kriging Interpolation Incorporating Automatic Variogram Fitting. *Computers & Geosciences*, 37(4):464–473, 2011.

[292] I. Loshchilov, M. Schoenauer, and M. Sebag. Dominance-Based Pareto-Surrogate for Multi-Objective Optimization. In *Simulated Evolution and Learning*, pages 230–239. Springer, 2010.

[293] T. W. Simpson, D. K. J. Lin, and W. Chen. Sampling Strategies for Computer Experiments: Design and Analysis. *International Journal of Reliability and Applications*, 2(3):209–240, 2001.

[294] A. J. Booker. Design and Analysis of Computer Experiments. In *Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 2–4, 1998.

[295] Y. Jin. A Comprehensive Survey of Fitness Approximation in Evolutionary Computation. *Soft Computing*, 9(1):3–12, 2005.

[296] W. Song and A. J. Keane. Surrogate-Based Aerodynamic Shape Optimization of a Civil Aircraft Engine Nacelle. *AIAA Journal*, 45(10):2565–2574, 2007.

[297] T. Ghisu, G. T. Parks, D. M. Jaeggi, J. P. Jarrett, and P. J. Clarkson. The Benefits of Adaptive Parametrization in Multi-Objective Tabu Search Optimization. *Engineering Optimization*, 42(10):959–981, 2010.

[298] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous Formation Flight. *Control Systems*, 20(6):34–44, 2000.

[299] P. K. Menon, G. D. Sweriduk, and B. Sridhar. Optimal Strategies for Free-Flight Air Traffic Conflict Resolution. *Journal of Guidance, Control, and Dynamics*, 22(2):202–211, 1999.

[300] V. Williams and R. B. Noland. Variability of Contrail Formation Conditions and the Implications for Policies to Reduce the Climate Impacts of Aviation. *Transportation Research Part D: Transport and Environment*, 10(4):269–280, 2005.

[301] J. S. Fuglestvedt, K. P. Shine, T. Berntsen, J. Cook, D. S. Lee, A. Stenke, R. B. Skeie, G. J. M. Velders, and I. A. Waitz. Transport Impacts on Atmosphere and Climate: Metrics. *Atmospheric Environment*, 44(37):4648–4677, 2010.

[302] H.-J. Shyur. A Quantitative Model for Aviation Safety Risk Assessment. *Computers & Industrial Engineering*, 54(1):34–44, 2008.

[303] Y. Schipper, P. Rietveld, and P. Nijkamp. Environmental Externalities in Air Transport Markets. *Journal of Air Transport Management*, 7(3):169–179, 2001.

[304] W.-K. Lee. Risk Assessment Modeling in Aviation Safety Management. *Journal of Air Transport Management*, 12(5):267–273, 2006.

[305] F. Netjasov and M. Janić. A Review of Research on Risk and Safety Modelling in Civil Aviation. *Journal of Air Transport Management*, 14(4):213–220, 2008.

[306] B. Y. Kim, G. G. Fleming, J. J. Lee, I. A. Waitz, J.-P. Clarke, S. Balasubramanian, A. Malwitz, K. Klima, M. Locke, C. A. Holsclaw, L. Q. Maurice, and M. L. Gupta. System for Assessing Aviation's Global Emissions (SAGE), Part 1: Model Description and Inventory Results. *Transportation Research Part D: Transport and Environment*, 12(5):325–346, 2007.

[307] N. E. Antoine and I. M. Kroo. Framework for Aircraft Conceptual Design and Environmental Performance Studies. *AIAA Journal*, 43(10):2100–2109, 2005.

[308] S. Stewart. *Flying the Big Jets*. The Crowood Press, 4th edition, 2002.

[309] J. D. Anderson. *Aircraft Performance & Design*. McGraw-Hill, 1999.

[310] V. Williams, R. B. Noland, and R. Toumi. Reducing the Climate Change Impacts of Aviation by Restricting Cruise Altitudes. *Transportation Research Part D: Transport and Environment*, 7(6):451–464, 2002.

[311] I. N. Egorov, G. V. Kretinin, I. A. Leshchenko, and S. V. Kuptzov. Multi-Objective Robust Design Optimization Using IOSO Technology Algorithms. *Evolutionary Algorithms and Intelligent Tools in Engineering Optimization*, pages 224–247, 2005.

[312] I. N. Egorov, G. V. Kretinin, I. A. Leshchenko, and S. V. Kuptzov. Multi-Objective Approach for Robust Design Optimization Problems. *Inverse Problems in Science and Engineering*, 15(1):47–59, 2007.

[313] J. G. M. Eggels. Direct and Large-Eddy Simulation of Turbulent Fluid Flow Using the Lattice-Boltzmann Scheme. *International Journal of Heat and Fluid Flow*, 17(3):307–323, 1996.

[314] L. P. Khoo, Z. Zhong, and H. Y. Lim. A Rough Sets Based Tabu-Enhanced Genetic Algorithm Approach to Rule Induction. In *Evolutionary Algorithms and Intelligent Tools in Engineering Optimization*, pages 141–166. 2005.

[315] G. S. Dulikravich, T. J. Martin, B. H. Dennis, and N. F. Foster. Multidisciplinary Hybrid Constrained GA Optimization. In *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances and Industrial Applications*, pages 1–25, 1999.

[316] H. Ishibuchi, Y. Hitotsuyanagi, H. Ohyanagi, and Y. Nojima. Effects of the Existence of Highly Correlated Objectives on the Behavior of MOEA/D. In *Evolutionary Multi-Criterion Optimization*, pages 166–181. Springer, 2011.

[317] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[318] D. J. J. Toal, N. W. Bressloff, and A. J. Keane. Geometric Filtration Using POD for Aerodynamic Design Optimization. In *Collection of Technical Papers–26th AIAA Applied Aerodynamics Conference*, pages 18–21. AIAA, 2008.

[319] D. J. J. Toal, N. W. Bressloff, A. J. Keane, and C. M. E. Holden. Geometric Filtration Using Proper Orthogonal Decomposition for Aerodynamic Design Optimization. *AIAA Journal*, 48(5):916–928, 2010.

[320] D. Brockhoff and E. Zitzler. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In *Parallel Problem Solving from Nature-PPSN IX*, pages 533–542. Springer, 2006.

[321] D. Brockhoff and E. Zitzler. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In *Operations Research Proceedings 2006*, pages 423–429. Springer, 2007.

[322] Y. Zhang and P. I. Rockett. Multiobjective Genetic Programming Feature Extraction with Optimized Dimensionality. In *Soft Computing in Industrial Applications*, pages 159–168. Springer, 2007.

[323] D. Kimovski, J. Ortega, A. Ortiz, and R. Baños. Parallel Cooperation for Large-Scale Multiobjective Optimization on Feature Selection Problems. In *Applications of Evolutionary Computation*, pages 693–705. Springer, 2015.

[324] D. S. Lee and D. W. Raper. *Towards Sustainable Aviation, Chapter: The Global Atmospheric Impacts of Aviation*, pages 77–96. Routledge, 2003.

[325] D. S. Lee. *Transport and the Environment, Chapter: The Impact of Aviation on Climate*, pages 1–23. Royal Society of Chemistry, 2004.

*This page is intentionally left blank.*

# Appendices

# Appendix A

# Scalability of GPU-MOTS2

The following figures depict the performance discussed in subsection 3.2.6.

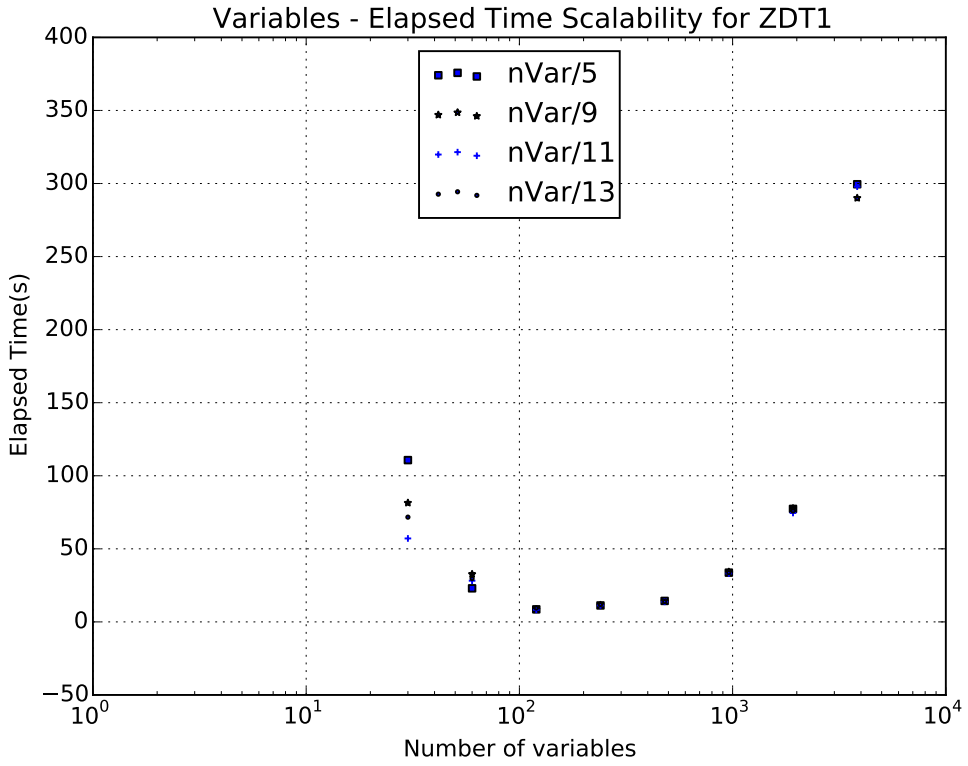



Figure A.1: Scalability of the Elapsed Time of GPU-MOTS2 on ZDT1 when the Number of Variables Increases

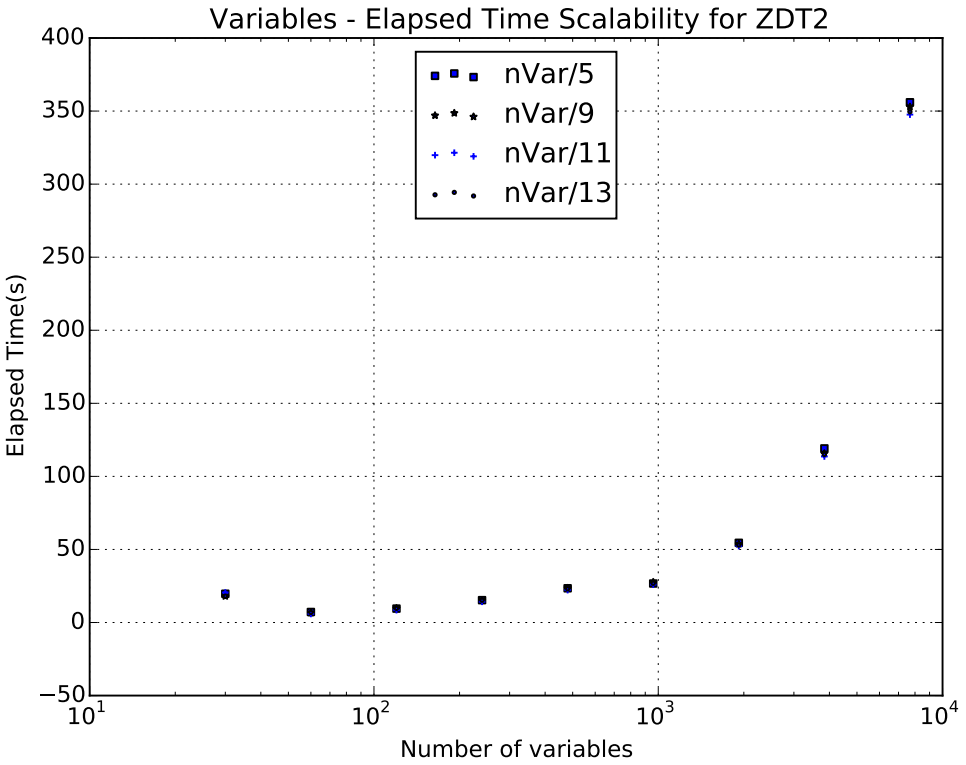



Figure A.2: Scalability of the Elapsed Time of GPU-MOTS2 on ZDT2 when the Number of Variables Increases

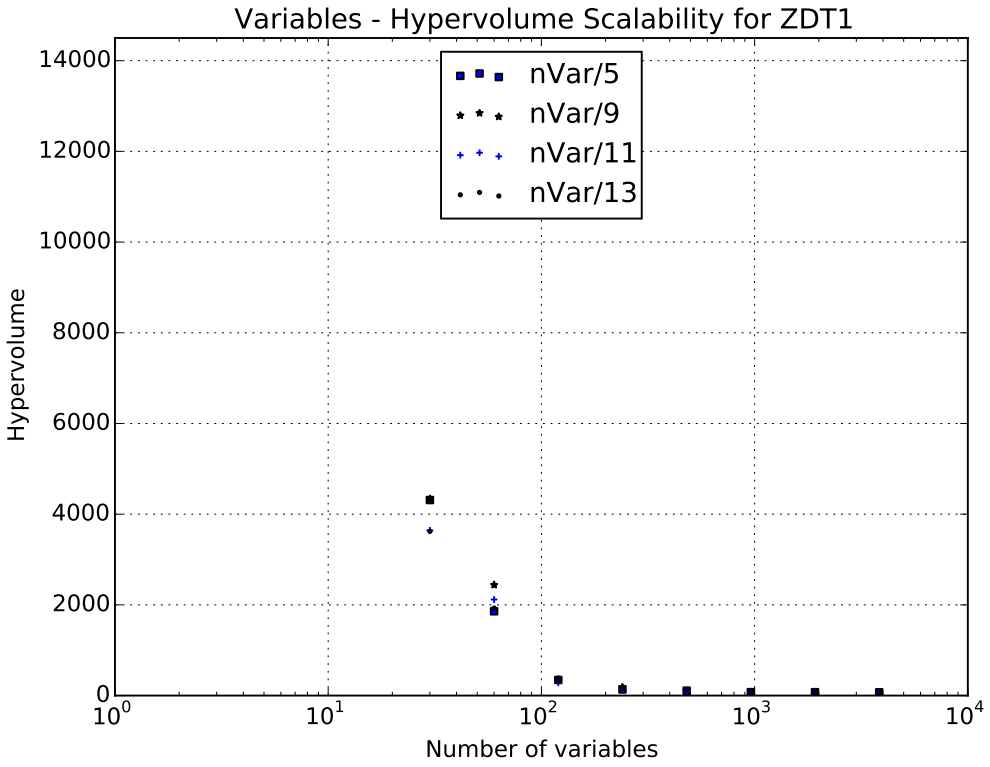



Figure A.3: Scalability of the Hypervolume Indicator of GPU-MOTS2 on ZDT1 when the Number of Variables Increases

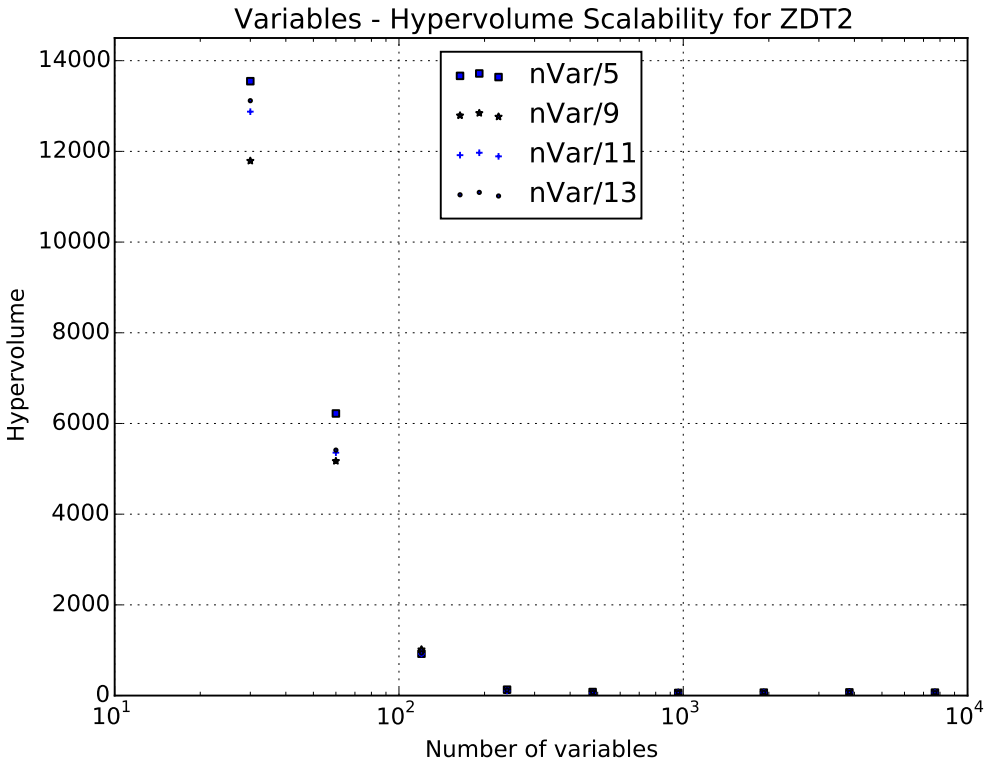



Figure A.4: Scalability of the Hypervolume Indicator of GPU-MOTS2 on ZDT2 when the Number of Variables Increases

*This page is intentionally left blank.*

# Appendix A

# Rebuttal - TEMPORARY Chapter

Format:

1. all the numbered items are the suggested corrections.

- The bullet point(s) that follow each numbered item is my reply.

Indices:

- Statement of Thesis Deficiencies (SoTD): Statement of Thesis Deficiencies
- Revised Assessment By Dr Parks (P2): Corrections suggested by Dr Parks
- Revised Assessment By Dr Sethi (S2): Corrections suggested by Dr Sethi

## A.1 Statement of Thesis Deficiencies

1. SoTD.The title abstract and presentation of the thesis imply that there is a strong focus on design for climate change mitigation. In reality the principal contributions of the work done by the candidate lie in the development of new tool and methodologies for computational design optimization. Although case studies presented in the thesis can be related to climate change mitigation, this does not alter the fact that the research undertaken is really about methodology rather than application; the applications merely illustrate the methodologies developed. In revising the thesis the candidate should ensure that this balance is more appropriately presented.

    - The abstract adheres to a new structure, suggested by Dr Sethi, and is expected to be adequate.

- The text balance of the text changed to emphasise on the developed tools and methods, which could be used to address the problem described in introduction.

2. SoTD.In the oral examination it became apparent that some of the candidate's most significant contributions are not reported in the thesis at all. The thesis therefore needs substantial revision not only to change its emphasis but also to report these contributions properly.

   - The emphasis in section 1.2 (on page 6) and the contribution in section 1.5 (on page 11) are expected to be adequate.
   - The case of 3-objective aircraft trajectory optimisation was added in section 4.1.2, on page 134.
   - More information regarding the development of GPU-Lattice Boltzmann code was added in subsection 3.3.4 (on page 94) and 3.3.7 (on page 105).

3. SoTD.The project context is vague. It is not clear exactly what the problem/limitations of state-of-the-art optimization methods and computational design methodologies are and what new methods are required.

   - Following the above, the first chapter reduced in size (to a few pages), where it is explained that the conflicting challenges of aviation can split into two categories. By using optimisation techniques and alternative computational technology it is possible to accommodate these challenges.

4. SoTD.The candidate should make clear what the research questions are, what the aims and objectives are and what approaches and methodologies were adopted to achieve these aims and objectives and to answer the research questions. None of these are clear.

   - The contribution to knowledge in section 1.5, page 11, explains what are the most important reasons one to use the products of this research. Following the advice by Dr Sethi, the objectives are re-formed to describe what this research will do. Then, each item is linked to a method, which describes how to correspond to the objectives. Following this, the structure of the second chapter changed, so as to report progress against the items in the list of methods and how my research contributes to knowledge.

5. SoTD.Within the abstract, the outcome and conclusion of the research are also presented in a very vague way. What were the main limitations of the research? What are the contributions to knowledge?

- Starting from the paragraph that starts with "MOTS2 can handle the multi-modality...", the findings are presented.
- The new abstract is expected to be adequate.

6. SoTD.The introduction and Literature Review are much too wide in scope. The candidate has made an effort to capture (poorly in places) the complete context of civil aviation. The introduction should be more focused on the research topic. Appropriate revision should result in a substantial reduction in length of the opening two chapters of the thesis.

   - Following the above, the first two chapters were modified, to be shorter and more focused. The emphasis is on developing the tools and methods up to the point where it is possible to demonstrate that the products can be applied in applications of ATO and design optimisation of a microreactor for mixing.

7. SoTD.The candidate should include additional aircraft trajectory optimization case studies. We understand that studies of three-dimensional trajectories with three objective functions have already been completed. These studies should be incorporated. In comparing the performance of the state-of-the-art (NSGAMO) with MOTS2, the candidate should be more dispassionate and recognize that a finding that in some cases MOTS2 does not add value is perfectly acceptable research outcome. The candidate should also include discussions of the underlying physics to explain why certain trajectories are optimal.

   - The case of 3-objective aircraft trajectory optimisation was added in section 4.1.2, on page 134.
   - It is clearly mentioned in sub-subsection 4.1.1.6, on page 133, 2nd paragraph, that the performance of MOTS2 is inferior to NSGAMO. In future work, in section 5.5.1.2, on page 213, first paragraph, recommendations are made, so as to increase/explore the effectiveness of MOTS2.
   - Underlying physics of optimal trajectories are discussed in 4.1.1.5, on page 127.

8. SoTD.The candidate should include a new chapter describing the substantial effort apparently undertaken in implementing Lattice Boltzman Method (LBM) on GPU-based hardware. Evidence of the process of validation of the GPU-LBM implementation should also be presented.

   - More information regarding the development of GPU-Lattice Boltzmann code was added in subsection 3.3.4 (on page 94) and 3.3.7 (on page 105).

9. SoTD.The case study of the design optimization of a micro-reactor should be presented as an illustration of the capabilities of the GPU-LBM implementation combined with MOTS2. The “emission mixing” descriptor used for the application is highly misleading and should be changed. The results produced by previous researchers from Cranfield (in reference [3]) should be included for purposes of comparison, illustrating more clearly the benefits of the new methods developed. The candidate should ‘tone down’ or properly evidence claims about the complexity of the design space for this problem. In presenting this case study the candidate should compare the computational costs of his new approach with the existing state-of-the-art (in terms of other LBM implementations and alternative [i.e. CFD] simulation tools)

    - The subsection 4.2, on page 163, was renamed.
    - Table 4.26, on page 179, was added, for comparison purposes. Discovering more design points and faster execution time are described in the text in subsection 4.2.4, on page 178, paragraph that starts with "Compared to [64], many...".
    - All references to "complex design space" were removed.
    - Regarding the computational costs, there is room for improvement (as mentioned at the last paragraph of subsection 3.3.7, page 113) and more future work is detailed in sub-subsection 5.5.3.2, on page 224 first paragraph.

## A.2 Common Corrections

1. P2 and S2.The candidate has made some efforts to rewrite the thesis so that it is more focused on the development of new tools and methodologies for computational design optimization, but unfortunately not at the expense of the excessive widely scope of the Introduction and Literature Review. Although some material appears to have been moved around in the revision of the thesis, it is not obvious that any substantial cuts have been made; indeed, the number of references has increased by 10%, to nearly 400. Chapters 1 and 2 are in need of further revision to satisfy the concerns expressed in the Statement of Thesis Deficiencies. In my view, the Appendices are completely unnecessary and symptomatic of the candidate’s reluctance to "let go".

    - Chapter 1 and 2 reduced in size (about half) and focused more on the objectives and related work about the objectives, as suggested in the Statement of Thesis Deficiencies.

- The original appendices were completely removed, as they add no value. Four figures were placed in the Appendix A, to support the findings about the scalability of GPU-MOTS2, described in 3.2.6(on page 84.
- The summary of chapter 3 was improved and reduced in size, on page 113.
- For clarity, the configuration settings of the optimisers for ATO, in secion 4.1, were moved to each subsection. In each subsection, under the sub-subsection "Preparing...", it is clearly mentioned which tools had been used. The reason(s) for selecting the particular tools can be found under the respective "Methodology...", in the same subsection.
- The parameterisation of the optimisation for the microreactor, in subsection 4.2.3, on page 169, was simplified and linked to the simulation description in section 3.3.7.
- Unnecessary references (in the sense that do not add value to the messages in the text) were removed.
- The Overall Project Methodology in section 1.3, page 7, was modified so as to give at a glance (visually and briefly) everything that has been used in the research. All the benchmarks and applications are going to refer to this, in derived instances (e.g., Fig. 3.6 and Fig. 4.1).

## A.3 Corrections suggested by Dr Parks

### A.3.1 Generic Corrections

1. P2.Although macroscopically the thesis is well presented, there are still a fairly large number of minor typographical errors and formatting infelicities. I have taken advantage of the fact that the candidate has chosen to submit a soft-bound thesis to mark typographical corrections required on my copy.

   - All identified typographical errors and formatting infelicities were addressed.
   - The document was professionally proof read twice.

2. P2.I regret to report that issues of inconsistency in the presentation of references persist. In particular, there continues to be no consistency in the use of capital letters in paper and journal titles. Some references are incomplete and at least one appears twice.

   - All references were corrected, so as to be consistent.

- Fixed references that were identified inconsistent in terms of using capital letters in paper and journal titles: [39, 108, 156, 220, 264, 267, 271].
- Fixed references that were identified as incomplete: [93, 113, 159, 206, 314, 324, 325].
- All duplicates removed.

3. P2.The writing style is still very 'flabby' and in places extremely repetitive.

   - Repetitive points within individual paragraphs were discovered and merged. Paragraphs with similar messages were also merged. All the identified (in the annotated thesis) duplication was addressed.

4. A consideration is that is not clear that the limitations of the state-of-the-art optimisation methods and computational design methodologies have been clearly identified:

   - The last two periods of the paragraph that starts with "Although there are numerous approaches to improve..." in section 2.2 (on page 14) demonstrate time and process limitations of the computational methods.
   - The last period of paragraph that starts with "Available computational power and low..." in section 2.3 (on page 16) demonstrates the limitations of CFD in real-world problems.
   - (in the following paragraph) The last period of paragraph that starts with "As usual, when implementing an..." in section 2.3 (on page 16) mentions the limitations of integration
   - The paragraph that starts with "For the purposes of advancing applications of aviation..." in section 2.3 (on page 17) describes the limitation in computational efficiency. (The top two sentences of this paragraph were also toned down).
   - The last period of the paragraph that starts with "When it comes to designing technology..." in section 2.4 (on page 17) refers to a gap in software design.
   - The last two periods of the following paragraph mention the gap to combine GPUs with CFD applications for micro-mixing optimisation.
   - The period "In addition, it is clearly mentioned in [141] that alternative parallelisation techniques (e.g. using GPUs) should be employed in the future." was added near the end of the first paragraph of section 2.5 on page 18, to further support the argument for using GPUs.

- For clarity, all the above are also listed at the beginning of the last section of chapter 2, on page 30.

### A.3.2 Specific Corrections

1. P2.On page 12 the candidate defines the acronym MDO to mean "Multi-objective Design Optimisation". This is a perfectly reasonable definition in light of the work presented in the thesis. However, at multiple places later in the thesis the candidate uses the same acronym to mean "Multidisciplinary Design Optimisation" (the more conventional meaning of the acronym) and, more problematically, implies that his work is an example of multidisciplinary design optimisation. It is not. His failure to appreciate the difference between multi-objective design optimisation and multidisciplinary design optimisation is deeply concerning. The fact that many MDO (M for multidisciplinary) papers are referenced is particularly troubling, because, if the candidate has read and understood these papers, he should be in no doubts that his own case studies are not multidisciplinary design optimisation studies.

   - P2.It is clearly understood that this work is not an example of multidisciplinary design optimisation (MDO). All references to the acronym of MDO were removed. There are two minor references to multidisciplinary design optimisation (appears in full wording) in the literature, to highlight that GATAC has the potential to be used for MDO studies, and in the conclusion, to mention that MOTS2 could be used in such studies. All the cited papers (from MDO) are included to demonstrate applications of multi/many-objective optimisation.

2. P2.On page 67, the candidate identifies the so-called kick move to be a unique feature of his MOTS2 implementation, and it is elsewhere claimed as a contribution of the research. I have two concerns about this. First, the circumstances in which this move is made are not very clearly explained — a pseudo-code presentation of the MOTS2 algorithm and ALL its features would be helpful. Second, I believe that there is only one instance of the kick move being made in all the MOTS2 runs presented in the thesis. A single occurrence in one run cannot conceivably provide a sufficient evidence base on which to make any claims about the effectiveness of this move.

   - All the material of the kick move can be found, as follows:
     - Material to describe kick in 3.1.5 on page 56, paragraph that starts with "An extra move has been added in the portfolio of moves,..."

  - Material to describe kick in 3.1.7.1 on page 68, paragraph that starts with "The configuration settings of the optimiser..."
  - Material to describe kick in 4.2.4 on page 189, paragraph that starts with "In combination with Fig. 4.36,..."
  - A discussion on the effectiveness of kick can be found at the last paragraph in 4.2.5 on page 197, paragraph that starts with "Regarding the effectiveness of kick,..."
  - Kick is explicitly mentioned as non-effective on page 199 (key point that starts with "A new optimisation algorithm, called MOTS2,...")
  - This is also mentioned in section 5.1 on page 199, point that starts with "A new optimisation algorithm, called..."

- Added Pseudocode/Listings:
  - Listing 3.1 (page 38) was added (also Fig. 3.3(page 45) was modified accordingly to reflect the pseudocode, for consistency).
  - STM Container: Listing 3.2 (page 40) was added.
  - Container: Listing 3.3 (page 41) was added.
  - LTM Container: Listing 3.4 (page 43) was added.
  - Hooke and Jeeves Move: Listing 3.5 (page 49) was added.
  - Pattern Move: Listing 3.6 (page 51) was added.
  - Intensify Move: Listing 3.7 (page 53) was added.
  - Diversify Move: Listing 3.8 (page 53) was added.
  - Restart Move: Listing 3.9 (page 54) was added.
  - Kick Move: Listing 3.10 (page 56) was added.

- Along with a pseudo-code, a UML class diagram of the top level was added in sub-subsection 3.1, on page 38. This is more useful for developers and describes the structural connections among the depicted modules.
- It is explained in the definition of the move, in subsection 3.1.5, on page 55, that kick was an experimental move and is expected to be used infrequently.

3. P2.The parameter values used in the tests presented in subsection 3.2.6.1 are not reported.

- At the beginning of the section 3.1.6.1, on page 59, it is mentioned that all the tests start from the middle of range of the search space. Also, the configuration settings are listed in Table 3.1.

4. P2.The performance testing of GPU-MOTS2 appears to be limited to tests on a single benchmark optimisation problem (ZDT2). That scale-up performance will be similar on other problems seems to be accepted as an article of faith. The one set of tests conducted is limited to examining how performance scales with the number of design variables for a fixed set of MOTS2 control parameters. No consideration is given to possible interactions between the algorithm configuration and the available hardware. This is a serious omission in my view.

    - On page 72 (paragraph that starts with "The underlying hardware equally contributes to the overall..."), the implications of different hardware are discussed. This was originally commented on page 83 and is also included in the future work on page 221.
    - The implications are also discussed in subsection 3.2.6 (page 84), which also links to Appendix A (page 258).
    - Two paragraphs were added on page 88, the first starts with "The obtained speed-up is highly ..." and the second with "From the hardware perspective..."
    - The subsection 3.2.6 was added on page 84
    - The discovered knowledge from the scalability studies was accommodated in subsection 3.2.7 on page 87.
    - two points: 3 (page 114) and 5 (page 114) were updated.

5. P2.At the original oral examination, the examiners were told that the development of the GPU-LBM implementation had presented considerable challenges. Although the development is now reported in section 3.4, it is not apparent from what is written that this was particularly difficult.

    - Subsection 3.3.4, on page 94 was further extended to highlight the differences between the first version of LBM and the GPU-enabled LBM. Also, on page 102, the architecture of the flow solver is used to support the porting.

6. P2.On page 100, a single validation case is presented for the GPU-LBM implementation. No details of the case (geometry, flow conditions) are actually provided (the reader apparently being expected to look these up in one of the references), and the validation is limited to a visual comparison of flow patterns – with the images themselves appearing to be mirror images. This evidence is wholly inadequate for validation purposes in my view. Full details of the case presented must be provided and other cases, encompassing

different flow conditions, presented before the GPU-LBM implementation can be claimed to have been properly validated.

- The concept of this subsection 3.3.7, on page 105, has changed. It is oriented towards proof-of-concept simulation, to be used in an optimisation study. The details of the case were added. All references/wording of validation have been removed. The validation could be carried out in the future, as described on page 224.
- Handling the optimisation results with caution has been mentioned at the following points:
  - it was mentioned in the second paragraph of the abstract that the GPU-based solver is a prototype. The penultimate paragraph also mentions that MOTS2 + GPU-based flow solver is a prototype combination.
  - The point in 7, on page 114 that starts with "The LBM package has been..." clearly mentions that more comparisons should be performed so as to validate the solver. This is expected to affect the micro-mixing optimisation.
  - Last period of the paragraph that starts with "As presented in..." on page 166.
  - The first period of subsection 4.2.5 (on page 196) mentions that the method is for prototyping purposes.
  - The last two periods of the paragraph that starts with "By applying the aforementioned" on page 196 mention that more
  - it is also mentioned as a limitation at the last point on page 210.
  - In the micro-mixing optimisation, in 4.2.2, on page 4.2.2, the paragraph that starts with "As presented in [64], the concept..." also mentions that the results of the optimisation should be handled with caution.
  - The last two sentences of the paragraph that starts with "By applying the aforementioned method 3,000 different designs..." in 4.2.5 on page 196 also mentions that the results should be handled with caution and the same study could be repeated again when the code is validated.
  - This limitation/risk is also described in risk 19 on page 210 and a mitigation plan is suggested, too.

7. P2.Subsection 3.4.7.2 appears to be incomplete – it is only three lines long.

- On page 102, the architecture of the flow solver is presented.

8. P2.The description of NSGAMO is too superficial to understand the significance of some of the parameters specified in Table 4.3.

   - NSGAMO is described in sub-subsection 2.6 (page 22), as also described in [220, 273].
   - NSGAMO is described in section 2.6, on page 22. Each of the settings from Table 4.2 (page 125) are also described in the text.
   - NSGAMO was also selected for comparison with MOTS2. MOTS2 demonstrated better performance in [257] and this is also mentioned in the last two sentences in sub-subsection 4.1.1.1, on page 118, paragraph that starts with "Following the integration of MOTS2 ...".

9. P2.The three-objective ATO case study presented lacks the equivalents of Tables 4.3 and 4.4 specifying the optimiser settings and control variable bounds (although some of the former can be deduced from Fig. 4.10).

   - The text in sub-subsection 4.1.2.2 (paragraph that starts with "The trajectory is decompose..." ) , on page 136, changed to be more specific.
   - Tables 4.5 (on page 137) and 4.8 (on page 149) introduce the variables and link them to flight segments.
   - Table 4.9 was inserted on page 151, so as to list the settings for MOTS2.

10. P2.Fig. 4.12 needs to identify which parameter is which.

    - The aforementioned figure is Fig. 4.14, on page 156, and it is linked to the tables mentioned below.
    - Table 4.8, on page 149, was inserted to help identifying the variables in Fig. 4.14, on page 156. For reading convenience, a reference to the list of variables is also mentioned in the text: in sub-subsection 4.1.2.5 (on page 154), in sub-subsection 4.1.2.6 (on page 155, first period of the sub-subsection. In the same paragraph it is explicitly mentioned that the altitude of the first segment in the climb phase is the first variable that accounts for 65% variability).
    - Table 4.5, on page 137, was modified, so as to explicitly declare which variable is which. This is expected to assist the reader to map the variables in Figure 4.14, on page 156.

11. P2.I would question whether many of the “Immediate Extensions” presented on pages 194 and 195 are remotely “immediate”. Many of them seem like difficult questions for a Business School, rather than straightforward extensions of the candidate’s work.

- The originally called section "Immediate Extensions"(section 5.5.4, on page 225) was removed, as the listed items are too complicated to be resolved in the short term or medium term. They were inserted into the future work, towards the end. The most strategic one can be found in subsection 5.5.4, on page 225.

### A.3.3 Annotated Corrections

- the paragraph in the abstract that starts with "MOTS2 can handle the ..." was amended to explain ZDT functions.
- Following the point above, illustrations of the performance of NSGAMO in subsubsection 3.1.6.1 (on page 63) were added to support the statement in the abstract along with figures 3.13- 3.15 (from page 63 to 65). In addition, a point was raised in section 2 (on page 114) that starts with "Compared to a state-of-the-art optimiser...", so as to highlight the strength and weakness of MOTS2 against NSGAMO in the benchmark functions.
- the potential user base shrank in section 1.1 (on page 1 )
- The boundary conditions described in subsection 3.3.7 (on page 107) were supported by the referencing the original scientific journal and appropriate text book. Text starts with "At the outlet of the domain...". Also added reference [253].
- the scaling in figure 3.25 (on page 108) was corrected and the corresponding section in the text (on page 107, paragraph that starts with "The geometry of interest is a..." ) was updated with the correct number.
- Units were added in figures 3.28 (on page 111) and 3.29 (on page 111)
- figures 3.30 (on page 112) and 3.31 (on page 113) were swapped, for consistency with the text
- clarified that the new strategy in section 1 (on page 114) refers to kick (which was reinstated)
- it is clearly mentioned in the last point of section 7 (on page 114) that a single case was compared, which is also linked to the limitations (last point of the section of limitations 19 (on page 210). (two references were added [254, 282] to complement the limitation/future direction)

- The text in section 3.3.7 (on page 109, paragraph "The effectiveness of the developed..." was clarified (last 4 periods, starting with "Although the flows between ..."), so as to explain that images are not mirrored, but a close match for prototyping purposes, only.
- Software testing level (more than 50%) is backed up by citations in subsection 3.3.4 (on page 96, "A series of unit tests.." ). Two references( [251,252]) were added.
- Since the kick move was reinstated, it is reasonable to preserve the bottom two configuration parameters in table 4.10 (on page 173)

### A.3.4 Other rebuttal comments

- the F in SSRF is indeed retain factor, as suggested by Dr Kipouros during the implementation of MOTS2

## A.4 Corrections suggested by Dr Sethi

- More rationale of reducing NOx by mixing was added on page 4, paragraph that starts with "In order to reduce...".
- Following the skype session, the author removed any reference to datum design and compromise design from the text, starting from page 155, as the focus is on demonstrating that 3 competitive solutions can be discovered by applying MOTS2.
- A weakness in the trajectory, with respect to ATC regulations, is captured in the 'identified issues' and 'limitations', and could be addressed by introducing new altitude constraints.
- Additional evidence was added on page 160, paragraph that starts with "The shape of...", in order to support why the discovered optimal trajectories are sensible.

### A.4.1 Corrections in Abstract

- The abstract is created following the structure below:
  - (from)project context
  - (from)aims and objectives WITH methods
  - (from)contribution to knowledge

- (from)results / summary of results
- (from)main findings
- (from)main recommendations(from future work)

### A.4.2 Corrections in Introduction

1. Following the meeting on 13/10/2016, it was advised to add a sentence to link the introduction to the next chapter. At the end of the paragraph that starts with "Many processes or procedures could be...", the following period was appended: "This research is expected to contribute to the establishment and improvement of tools and methods in the field of optimisation via simulation, by suggesting solutions to address the identified limitations that are described in the next chapter.", which can be found in subsection 1.1.3 (on page 5).

2. S2.Adding methodology at beginning of the document

   - Subsection 1.3 can be found at page 7. It describes the overall methodology to answer the research questions.

3. S2.Again emphasis is placed on design for climate change and modelling environmental impact – as discussed during the viva, the main contributions lie in the development, refinement and verification of the methodology with appropriate case studies to illustrate the value of the new methodologies relative to existing tools/frameworks.

   - Section 1.2, on page 1.2, is expected to adequate and the technological readiness level is estimated in section 1.3, on page 7.

4. S2.What is the relevance of delays, UAVs in the introduction? These are not assessed within the scope of the thesis.

   - UAVs are not relevant in this study. So this paragraph was removed.

5. S2.1.2.3 – there are other asset management strategies, smart operations on ground, alternative fuels, emissions taxation etc. that may contribute to reducing the environmental foot print of aviation. Poor.

   - The second paragraph on page 1 was modified to state air transport challenges. The subsection 'Means of Reducing Impact of Aviation on the Environment' was removed. Section 1.1.2 (in page 3) is expected to be adequate (and more focused) towards hydrogen-based energy and CFD (and LBM, as an extent) to analyse such technology.

6. S2:1.3 – very disjointed. Methodology? Case studies?

   - The aforementioned 1.3 (Research Questions and Hypothesis) has been removed, as it was very vague and unsuitable.
   - The new form of Aims and Objectives (in Section 1.2, on page 1.2) is expected to be adequate.

7. S2:Past tense

   - present tense is used to highlight current problems/gaps.

8. S2:Research questions do not relate to aims and objectives and CTK statement.

   - the objectives were modified to be more measurable and link directly to the developments. The benefits of using/having the developed tools are reflected in the CTK. The items in the CTK are related to the structure of literature review.

9. S2:CTK – this is a bit better but it is necessary to provide a summary/overview of the limitations of existing frameworks/methodologies – e.g. we have done a lot of work on traj. Optimisation at CU alone – how and why is your framework more superior. The CTK need to be justified/substantiated. Section 1.5.1 attempts to do this but a lot more evidence is needed. You also need to provide some evidence of this in our case studies. Again it appears that the claims made are more than what is actually delivered (e.g. ...their aircraft will satisfy all the requirements...)

   - Section 1.5, on page 11, is expected to be adequate.
   - Identified limitations of other tools/methods, based on literature, are listed in 2.8.
   - Strong arguments were removed.

10. S2:Environmental performance indications have reached their limit and government is artificially trying to constrain this effect??

    - This period was removed.

11. S2:There is a lot of repetition – (in my opinion waffle) in the introduction section.

    - The size of the Chapter was reduced, to minimise any repetition.

12. S2:I feel an overall schematic of the framework which was developed with a clear indication of the specific novel features/capabilities developed by the author would help if include in this chapter.

    - An abstract schematic is presented in section 1.3 on page 7, more specifically: 1.1 on page 8. This is further refined before each benchmark and application.

### A.4.3 Corrections in Literature

1. S2:The literature review is fair although I still have some concerns with respect to the contributions and claims for CTK. These should be substantiated by the literature review. Limitations of existing methods for trajectory optimisation should be clearly and concisely summarised bringing out the main contributions of this research, beyond SOA (MOTS, framework, GPUs etc.). I suggest a summary section is added to the end of the literature review section to elaborate on these points.

    - Limitations of methods in trajectory optimisation are listed in section 2, on page 13.
    - As advised, summary of the key points from literature can be found in section 2.8, on page 30. These are effectively requirements that describe gaps of knowledge/missing features, so as to back up, why the new tools were developed, new methods were devised. Consequently, all these are demonstrated in the applications.

2. S2:Refer to the rebuttal chapter point 2c. This point should be made in the main part of the literature review and the value of this should be elaborated.

    - In the copy that was submitted on 8th of January, I have put in the rebuttal chapter that "to the best of the author's knowledge, it was the first time that MOTS2, as a local search optimiser, was applied to ATO problem". This gap is demonstrated in section 2, on page 13. It is also mentioned as a requirement in section 2.8, on page 30, and is demonstrated in an application in subsections 4.1.1 (page 117) and 4.1.2 (page 134).

3. S2:No literature is provided w.r.t. the "reactor" which is used for one of the case studies. It is still not clear what this "reactor" is and why this was deemed important for the case study to substantiate the claims.

    - Micro-reactor is defined in section 1.1.2, on page 4.

- Section 1.1.2, on page 3, explains why a microreactor is important to study (i.e., it can do micro-mixing, which is one of the main features in order to use hydrogen as an alternative fuel).
- Related literature can be found in section 1.1.2, on page 4.

### A.4.4 Corrections in Computational Tools

1. S2:As before I feel this chapter is significantly better than the previous two. It is well written, thorough and relevant.

2. S2:I feel a table/matrix which summarises the main characteristics of optimisation problems (either in general or more specific to ATO) (e.g. ability to handle multiple objectives and constraints, large number of variables, discontinuities, computational time, convergence and diversity etc.) should be provided with a qualitative assessment of different types of optimisation techniques. The strengths and weaknesses of these techniques based on each of the criteria should be discussed (qualitatively) and this will help provide a justification for selecting MOTS. A similar analysis would also be useful for the framework in general (relative to other frameworks available). These assessments will help justify the CTK claims. This can perhaps be included in the previous chapter.

   - At the end of Chapter two, a list of high-level requirements and gaps are presented.

3. S2:These claims then perhaps need to be quantitatively proved in some of the assessments and case studies (e.g. benchmarking against standard mathematical test functions).

   - At the end of Chapter two, a list of high-level requirements and gaps are presented.
   - Could this requirement be removed?

4. S2:The description and functionality of MOTS is good and thorough. The section on "unique features" is good – brings out the contributions but the value of these features need to be demonstrated in the case studies i.e. MOTS vs MOTS2.

   - To the best of my knowledge, MOTS can only run on linux and the source code is not available. Hence, it can only be used on linux environment. In the first optimisation of microreactor, MOTS was used along with the original version of LBM, as explained in section 2.4, on

page 17. However, this is not a fair comparison. This is was added in the future work, sub-subsection 5.5.3.1, on page 220.

5. S2:Benchmarking – it would be good to explain what specific features the ZDT functions are trying to assess (large number of local maxima, discontinuous PFs, convex PF etc.) What does the HV value represent? MOTS2 did not perform well for ZDT3 but NSGAMOII I believe does. What are the implications of this for the problems you are looking at (e.g. ATO)?

   - The features of ZDT functions are listed in Table 3.2, on page 60, and this is linked to the text in sub-subsection 3.1.6.1, on page 59, paragraph that starts with "The verification methodology...".
   - HV (i.e., hypervolume) is one of the standard quality indicators to assess the performance of multi-objective optimisers and is briefly explained in the footnote in subsection 3.1.5, on page 56, paragraph that starts with "In the portfolio of moves an extra has been added...".
   - In this research, NSGAMOII was not tested on ZDT functions, but the results from another study were used for comparison purposes.
   - In order to provide a reference to the performance of NSGAMO, the paragraph "For comparison/reference purposes and linking with the following chapter, the ability of NSGAMO (in the plots referred as NSGAMO3) to capture the target trade-off after 22000 objective function evaluations for three of the ZDT functions is demonstrated in Figs. 3.13-3.15. This is part of a comparison performance study of NSGAMO against state-of-the-art MOO algorithms and was conducted in [220]." was added in subsection 3.1.6.1 on page 63. This was also added for compliance with Dr Parks' suggestion to add material to demonstrate NSGAMO's performance, for completeness/convenience.
   - Risk( 18), on page 209, was also recorded.

6. S2:The air-foil shape optimisation validation is good but it does not bring out the value of MOTS2 relative to other optimisation techniques.

   - At the last period of the paragraph in subsection 3.1.7, on page 65, it is mentioned that the aim is to reveal a non-dominated trade-off in an application, where the evaluation of the objective functions is more complicated and computationally intensive compared to ZDT functions. A comparison against another optimiser (NSGAMO II) is demonstrated in subsection 4.1.1, on page 117.

7. S2:It would be worth showing the architecture/structure of GATAC.

- A brief description of GATAC can be found in section 2.6, on page 24, paragraph that starts with "In a collaboration between Cranfield...". Following that, a figure of the top-level structure of GATAC can be found in Fig. 2.1, on page 25.

### A.4.5 Corrections in Applications

1. S2:Environmental impact?
   - This is discussed in sections "Identified Issues" in Chapter 4, and important impact is also mentioned in the conclusions, under the section "Discussion on Findings".
2. S2:Elaborate on importance for optimising climb phase.
   - This is explained in the first paragraph of subsection 4.1.1.1, on page 117, paragraph that starts with "The flight path of...". It is advised in the guidelines of ICAO for Standard Instrument Departures and Standard Terminal Arrival Route.
3. S2:Pg. 110 there have been a number of studies at CU looking at 3 objectives (e.g. KP).
   - Appropriate references were added.
4. S2:Assumptions and limitations (and implications) of models used are not discussed.
   - A risk register has been generated in 5.4, on page 206. It appears as a text, as a matrix (the most common) is very sparse and hard-to-read.
   - This particular issue was registered as a risk( 4), on page 206.
   - For all the applications, the limitations and assumptions have been moved to the conclusions, section 5.4. This was formed as a risk register that also includes the importance of each element and the development/research effort required to satisfactorily address it.
5. S2:Hermes - Waypoints? Hermes (point mass) or APM (6DOF).
   - This is clarified in the specific sub-sections of the applications section.

- It was explained in the two sub-subsections of Methodology why a method is required, which led to using each of these tools. Mainly for diversity and proof-of-concept. Of course, there are discrepancies wrt the complexity, fidelity, usefulness/practicality and computational need of each. Should this be elaborated further?
- Their use is also highlighted in the risks 4(page 206), 10(page 208), and 13(page 208).

6. S2:From the ATO case study, NSGAMO appears to offer superior performance (diversity).

   - In subsection 4.1.1.6, it clearly mentioned that NSGAMO outperformed MOTS2 in terms of diversity.

7. S2:On what basis is the compromise design selected? How is this a fair comparison?

   - In all applications, all the compromise designs were chosen manually, from the middle of the PF. This was registered as a risk( 1) in section 5.4.

8. S2:What does the HV Indicator represent? Convergence, diversity?

   - It is explained above.

9. S2:What is meant by ratio of high to low fuel efficiency?

   - This was rephrased in sub-subsection 4.1.1.5, on page 133. It means the ratio of the number of trajectories with high fuel efficiency over the number of trajectories with low fuel efficiency.

10. S2:What is a micro-reactor?

    - It is explained above.

11. S2:What is the specific application?

    - It is explained above.

12. S2:What existing component(s)/ system(s) can it potentially replace and what is the justification behind the claim that it may yield “a greener future in aviation”? This is not clear and needs to be explained!

    - It is explained above.

13. S2:Why was this considered to be a relevant case study to demonstrate the capabilities of the developed methodology?

    - It is presented in 2.3, on page 15 that MOTS + LBM could perform micro-mixing optimisation.
    - As also explained in sub-subsection 4.2.1(page 163), MOTS2, as an optimiser, should be able to manage this, so as to be used in other design optimisation cases, in the future.

14. S2:What were the criteria to assess performance relative to other frameworks/optimisation algorithms?

    - Again, hypervolume (Table 4.12, on page 179) was used to numerically compare the quality of the trade-off. This was commented on page 178, at the end of the paragraph that starts with "Compared to [64],...".

15. S2:The problem set up is not well defined.

    - The problem of optimising micro-mixing is defined in sub-section 4.2.1, on page 163, is expected to be adequate.

16. S2:What are the objectives and what do they represent? What are the variables? What are the constraints? It would be nice to summarise these in a table.

    - The optimisation problem is presented in subsection 4.2.3, on page 170.
    - As fewer variables are considered, it is clearer/concise to demonstrate the formulation of the optimisation problem in the form of equation (4.5), on page 171. This was set up to match with the previous work and for comparison purposes.

17. S2:Identified issues – fast relative to what?

    - As explained in the identified issues of the design optimisation study in subsection 4.2.5, page 196, the speed comparison was between the combination of MOTS+LBM vs MOTS2+GPU-LBM.

18. The risk 2 on page 206 highlights the non-flyable nature of the trajectories and how this can be mitigated. It is also described in 4.1.2.7 on page 161 at the end of the paragraph that starts with "The methodology and results for optimal trajectories ...".

19. The risk 3 on page 206 was rephrased for clarity.

### A.4.6 Corrections in Conclusions

1. S2:A thorough description about what was done but what are the conclusions? Recommendations are through but in many cases too generic.

    - Following this comment, it was highlighted in section 5.5, on page 210, paragraph that starts with "The importance, complexity...", that the complexity and abstraction of the listed recommendations varies from short-term tasks to very strategic tasks. Depending on the level of seniority of the reader, the former target individual researchers that would prefer to immediately carry on existing work, whereas the latter later aim managers and head of departments that might be looking for a vision and future trends.

## A.5 Other Corrections, Introduced by the Author

- Section 4.1 was restructured in such a way to keep the common/general parts at the the same level as the applications (subsections 4.1.1 and 4.1.2). Then, individual information/parts were pushed into these subsections, so as to minimise confusion and inconsistencies.