Robust Evolutionary Methods for Multi-objective and Multidisciplinary Design Optimisation in Aeronautics

BY

LUIS F. GONZALEZ

A Dissertation submitted to the School of Aerospace,
Mechanical and Mechatronic Engineering
in fulfilment of the requirements for the Degree

Doctor of Philosophy

The University of Sydney

Australia

2005

Abstract

ROBUST EVOLUTIONARY METHODS FOR MULTI-OBJECTIVE AND MULTIDISCIPLINARY DESIGN OPTIMISATION IN AERONAUTICS

Luis Felipe Gonzalez	Doctor of Philosophy
The University of Sydney	2005

This thesis is concerned with the development of robust evolutionary methods and a framework in which different multi-objective and multidisciplinary problems in aeronautics can be designed, analysed and optimised. The methods couple a robust evolutionary optimiser, parallel computing, asynchronous evaluation strategies and a hierarchical topology of fidelity solvers that reduce the computational expense of multi-objective and multidisciplinary design optimisation processes. The methods and framework are applicable to single and multi-objective, inverse or direct, complex engineering problems that can be highly non-linear, multi-modal, involve approximations, be non-differentiable, with convex, non-convex, discontinuous Pareto optimal fronts**.**

The methods are integrated in a framework that comprises a Graphical User Interface (GUI), a robust evolutionary optimisation algorithm and several modules for parallel computing, post-processing, design and optimisation.

The reason for their development is the fact that in general, real-world problems in engineering involve three types of complexities:

- Instead of a single optimal solution, a trade-off between conflicting objectives arises due to manufacturing, technical and human constraints. Without preference information none of these trade-off solutions can be said to be better than the others.
- The search space for an optimal solution is often complex and involves non-linearities and multi-modalities and traditional search methods often encounter difficulties or altogether fail.

- Real-world problems usually involve multiple disciplines.

The designer or team of designers is usually presented with a problem that involves several complexities, objectives and multi-physics, therefore a systematic approach, referred to as Multidisciplinary Design Optimisation (MDO), that accounts for these complexities is required.

The conventional approach in aeronautical design and MDO has been the use of traditional deterministic optimisers. These methods are efficient in finding optimal global solutions if the objective and constraints are differentiable. However, robust alternative numerical tools are required if a broader application of the optimiser is desired, or if the problem is multi-modal, involves approximations, is non-differentiable or involves multiple objectives and physics, as is usually the case in the design of complex systems in aeronautics.

A relatively new technique for optimisation is the use of Evolutionary Algorithms (EAs). EAs are based on Darwinian theories of evolution, where populations of individuals evolve over a search space and adapt to the environment through the use of different mechanisms such as mutation, crossover and selection. EAs require no derivatives or gradients of the objective function and have the capability of finding globally optimum solutions amongst many local optima. They are easily executed using parallel computing techniques and can be adapted to arbitrary analysis codes without major modification. Another major advantage of EAs is that they can tackle multi-objective problems directly. Together these characteristics give EAs substantial advantages over more conventional deterministic approaches.

The main idea is to identify and develop robust methods and algorithms. The methods are implemented in a generic framework by integrating several components so that the designer or team of designers can focus on the problem itself. These methods and framework are developed as a tool that helps to automatically evolve and refine a candidate design such as an aerofoil, nozzle, wing or aircraft shape by changing the design variables which characterise what might be called an individual. At the end of the optimisation process, the optimal set of individuals reflects suitable designs from which the designer can select and progress into further steps of the design process. The challenge was to make the concept genuinely useful so it can be generic and adapted to different design problems. The key to this is the way the methods and framework integrate the different components and the manner in which the design variables, constraints and fitness functions (for example, takeoff weight, fuel-weight, aerodynamic performance) are encoded.

The main contributions of this thesis are:

- The development of several methods and a framework that can be used to solve single, multi-objective, multi-disciplinary conceptual and preliminary design optimisation problems relating to manned and unmanned aircraft systems. These problems can be highly non-linear, multi-modal, involve approximations, be non-differentiable or involve multiple objectives and physics, with convex, non-convex or discontinuous Pareto optimal fronts.

- To illustrate the benefits and practicality of the methods for representative types of problems in aeronautical systems design. These problems fall into the categories of single and multi-element aerodynamic shape optimisation, multidisciplinary and multi-objective wing design and multidisciplinary and multi-objective aircraft design optimisation
- The development, integration and testing of a robust generic framework. This framework comprises a GUI using object-oriented principles, a series of robust multi-objective and multidisciplinary design optimisation algorithms and coupling of several modules for parallel computing, post-processing, and aeronautical systems design.

The thesis presents a literature survey, covering the three main areas of this research, namely optimisation methods, multidisciplinary design optimisation and evolutionary algorithms. Next, the main requirements and implementation of evolutionary methods are described and mathematical test cases are considered. Then, real-world applications to single, multi-objective aerofoil, wing and aircraft design optimisation problems are presented. Finally conclusions and future areas of research are outlined.

The research undertaken in this thesis has gone a long way towards realising the benefits of several evolutionary methods for multi-objective and multidisciplinary design optimisation of aircraft systems. In particular, practical applications to aerofoil, wing and aircraft design using a combination of low-to-medium fidelity solvers have been elucidated. Key challenges in the future will be to extend these ideas to other systems and incorporate a higher number of design variables, Design of Experiment (DOE) studies and the use of higher fidelity models.

Acknowledgments

First and foremost, I must thank my family and Isabel, for their love, constant support and encouragement. Being apart and away from home has not been easy but they continued to believe and encouraged me to go forward in all aspects of life.

My adviser, Dr K Srinivas was a constant source of advice through my graduate studies. He gave me numerous opportunities to co-author technical papers and present results to engineering journals and at scientific conferences. These experiences have done a lot to enhance my professional growth and I am grateful for all his efforts. I am also deeply indebted to Dr Eric J. Whitney who, on occasions too numerous to mention, provided helpful insights and discussions into the research topic. For the past three and a half years Eric has not only been a great friend but a constant adviser and critic to many aspects of my research.

Professor Jacques Périaux provided invaluable inputs and advice for this work. For the past three years Professor Périaux has remained active in providing challenging topics and outlined the connection between what is expected from academia and industry. He has remained active via tele-communications, fax and email while being located at the Pole Scientifique in France and through several visits to The University of Sydney. His extraordinary efforts are greatly appreciated.

Thanks also to academics and staff at the School of Aerospace, Mechanical and Mechatronic Engineering at The University of Sydney, especially to Dr K. C. Wong, Dr H. Stone and Dr Doug Auld. Through working with them I have become more aware of the aircraft design process, multidisciplinary design optimisation, computer science in engineering and the tools and benefits of computational aerodynamics. Thanks to Professor Steve Armfield and Dr Patrick Morgan for their advice on topics related to Computational Fluid Dynamics (CFD) and parallel computing. Contributions to this research and to my developing career come also from my experience as a tutor for Professor John Kent's and Paul McHugh's courses. I am thankful for their advice and support, also to Stuart B., Duncan S., Greg C and Greg R at the aero workshop for their help and cheerful times.

Part of the learning process is through interaction with other postgraduate students in the department; special thanks to Michael W., S. Narayana and Chris for providing interesting and challenging questions that helped me to advance and revise some concepts.

Thanks you to Mourad Sefrioui of Dassault Aviation and K. Deb for their valuable input during their short visits to The University of Sydney in 2001 and 2003 respectively. Fruitful advice on the topics of hierarchical and multi-objective evolutionary approaches was obtained during these visits.

Thanks to Alice for all her care, support and sweet treats, to Juan N, Maria, Juan A, Ricardo, Samuel, Pabon, STUCCO and IH friends for happy times, support and encouragement during these four years.

On the topic of analysis tools used in this research, my thanks go to Arnie McCullers at NASA Langley Research Center, who kindly provided the FLight OPtimisation System code, NASA Langley and Prof Shigeru Obayashi for providing facilities on using the potential flow solver, Mark Drela for the Euler + Boundary Layer and B. Mohammadi for the Navier Stokes code.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Alt	altitude, *ft*
AR_w	wing aspect ratio
AR_{ht}	horizontal tail aspect ratio
a_i	discipline analysis outputs
b	wing span, *ft*
c_p	pressure coefficient
c_l	coefficient of lift (2D)
C_L	coefficient of lift (3D)
c_d	drag coefficient (2D)
C_D	drag coefficient (3D)
c_m	moment coefficient (2D)
C_M	moment coefficient (3D)
CSP	Constraint Satisfaction Problems
CMA	Covariance Matrix Adaptation
DACE	Design and Analysis of Computer Experiments
DDM	Distance Dependent Mutation
DOE	Design of Experiments
E	Endurance, *nm*
EA	Evolutionary Algorithm
FEA	Finite Element Analysis
f_i	objective/fitness function
g_i	system or disciplinary constraints
GA	Genetic Algorithms
GUI	Graphical User Interface
HAPEA	Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA)
M	Mach Number
MDO	Multidisciplinary Design Optimisation
MPI	Message Passing Interface
$MTOL$	maximum take-off length, *ft*

NN	Neural Networks
PVM	Parallel Virtual Machine
R	Range, *nm*
RSM	Response Surface Modelling
Re	Reynolds number
S_f	Safety factor
SGA	Simple Genetic Algorithm
SW	reference wing area, *sq ft*
S_{ht}	horizontal tail area, *sq ft*
t_{max}	maximum wing thickness
t/c	wing thickness-to-chord ratio
T	maximum take-off thrust per engine, *Lbf*
UAV	Unmanned Aerial Vehicle
$UCAV$	Unmanned Combat Aerial Vehicle
ULF	Ultimate Load Factor
WE	Aircraft weight, *Lbs*
y_i	state variables
x_i	local input variables
z	global or shared variable
Γ_w	wing dihedral, *deg*
Γ_{ht}	horizontal tail dihedral, *deg*
Λ_{ht}	horizontal tail 1/4 chord sweep, *deg*
Λ_{vt}	vertical tail 1/4 chord sweep, *deg*
Λ_w	wing 1/4 chord sweep, *deg*
α	angle of attack
ε_{allow}	strain
σ	local stress
σ_{ult}	ultimate stress
λ	offspring
λ_{ht}	horizontal tail taper ratio
λ_{vt}	vertical tail aspect ratio
λ_w	wing taper ratio
μ	assembled parent population

Chapter 1

Introduction

> *"Individual innovations imply, by virtue of their nature, a "big" step and a "big" change. A railroad through new country, i.e., country not yet served by railroads, as soon as it gets into working order upsets all conditions of location, all cost calculations, all production functions within its radius of influence; and hardly any "ways of doing things" which have been optimal before remain so afterward."*
>
> Joseph A. Schumpeter.

In today's world of advancing technology, engineers are faced with the problem of designing increasingly complicated multidisciplinary systems. This is a difficult task, as these systems not only involve coupling amongst the different physics involved, but also a large number of variables and a series of objectives and constraints. At the same time, engineers need to optimise and address several requirements which include the reduction of the time spent on design and reducing the cost of research and development, while improving the performance, reliability, quality and safety of the product or process under consideration.

These complex interactions have generated a growing interest in the area of multi-objective and Multidisciplinary Design Optimisation (MDO). In multi-objective optimisation the designer is interested not only in a single global optimal solution but in a set of solutions that represent the trade-off between the different objectives. MDO refers to an approach to formalise a design process which accounts for the interaction amongst the different physics involved, while optimising for a number of objectives and constraints.

The purpose of this thesis is to support the complex task of multi-objective and multidisciplinary design optimisation in aeronautics by developing methods and a framework that use robust numerical evolutionary optimisation algorithms.

When applied to aeronautics, the necessity of optimisation and MDO is clear, given that even a very small improvement in weight or a reduction in aerodynamic drag will have a tremendous impact on the overall performance of the design.

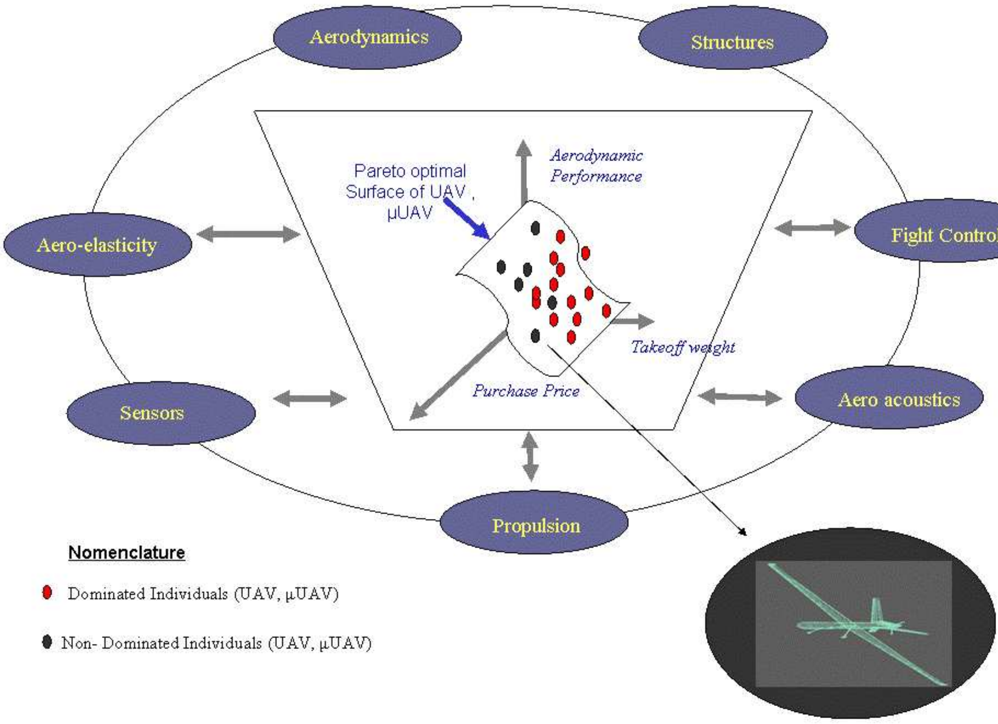

Figure 1.1: Multidisciplinary aircraft/UAV design and optimisation.

The application of MDO to the design and optimisation of aerospace vehicles can be represented as in Figure 1.1. In MDO a number of disciplines (aerodynamics, structures, propulsion, aero-acoustics, etc) are present and interact. The task of the designer or team of designers is to develop a solution that conforms to all disciplines while guaranteeing the requirements and constraints. When optimisation is intended, the different objectives (for example, aerodynamic performance, purchase price, take-off weight) need to be considered in order to find an optimal solution or set of non-dominated solutions.

A common approach for optimisation is the use of aggregating functions in which different weights are assigned to each objective. The problem with this approach is that the weight for each objective needs to be known in advance. Another approach is to compute or produce a set of solutions in what will be referred to in this thesis as a Pareto optimal front or surface. This Pareto optimal front represents the optimal set of non-dominated solutions and the trade-off between the objectives and disciplines involved.

The process of MDO involves the use of several analysis tools such as Computational Fluid Dynamics (CFD) software or Finite Element Analysis (FEA) and also optimisation tools. Analysis tools are under constant development and have reached a point where a confident application to aeronautical design in conjunction with MDO is possible [3, 103, 168]. However, there are limitation on their application within MDO at an industrial level, due to the computational expense involved. A single high-fidelity Navier-Stokes CFD computation around an aircraft wing, for example, might take several hours on a supercomputer. Therefore, the continuing challenge has been to develop methodologies such as Design of Experiments (DOE), approximation methods and variable fidelity models that combine and use different fidelity analysis tools during the design and optimisation process to minimise the computational expense.

While the area of traditional optimisation tools for a single discipline is quite mature, the area of robust optimisation tools and approaches for MDO is still at the initial stages of development [4, 5, 6, 13, 162, 161]. The conventional approach in aeronautical design and MDO has been the use of traditional deterministic optimisers. These optimisers are efficient for finding optimal global solutions if the objective and constraints are differentiable. However, robust alternative numerical tools are required if a broader application of the optimiser is desired, or the problem is multi-modal, involves approximations, is non-differentiable or involves multiple objectives and physics, as is usually the case in the design of complex multidisciplinary systems in aeronautics.

A relatively new technique for optimisation is the use of Evolutionary Algorithms (EAs). EAs are based on Darwinian theories of evolution, where populations of individuals evolve over a search space and adapt to the environment through the use of different mechanisms such as mutation, crossover and selection. EAs require no derivatives or gradients of the objective function and have the capability of finding globally optimum solutions amongst many local optima. They are easily executed using parallel computing techniques and can be adapted to arbitrary analysis codes without major modification. Another major advantage of EAs is that they can tackle multi-objective problems directly. Together these characteristics give EAs substantial advantages over more conventional deterministic approaches.

Interest in EAs for problems in engineering and aeronautics has grown substantially in the past fifteen years. These methods have been successfully applied to different aeronautical design problems including airframe, wing, aerofoil and rotor blade design [94, 119, 126, 132, 142].

The application of EAs for MDO problems has been limited. This is mainly because one of the drawbacks of EAs is that they are slow when compared to traditional deterministic methods, as they require a larger number of function evaluations to converge to an optimal solution. Hence the continuing challenge in evolutionary optimisation has been to reduce the number of function evaluations and the computational expense. To achieve this, several approaches have been proposed; these include a combination of variable fidelity models, parallelisation strategies and hybridisation techniques [28, 42, 81].

This thesis develops multi-objective multidisciplinary evolutionary methods and a framework in which different aeronautical problems can be designed, analysed and optimised. The methods are based on a robust evolutionary optimiser, parallel computing, asynchronous evaluation and a hierarchical topology of fidelity solvers that reduce the computational expense for multi-objective and MDO problems. The methods are applicable to single and multi-objective, inverse or direct complex engineering problems that can be multi-modal, involve approximations, non-differentiable, with convex, non-convex or discontinuous Pareto optimal fronts.

These evolutionary methods simplify the task of the designer or design team by integrating several components so that they can focus on the engineering problem itself. The methods are developed in a sequence of steps consisting of: defining the requirements, formulating the methods, identifying several promising robust analysis and optimisation methods, creation of algorithms and testing with mathematical functions and practical real-world problems in aeronautics.

The methods were implemented in a single framework that integrates a Graphical User Interface (GUI), a robust multi-objective evolutionary algorithm, a series of modules for mathematical test functions, aerofoil, nozzle, wing and aircraft/airframe design and optimisation, parallel computing, post-processing and a Design of Experiments (DOE) capability. The framework was developed in a way that different applications and modules could be easily developed and implemented. These modules include real-world applications using analysis tools with variable fidelities, benchmarking of mathematical test functions, DOE studies, and parallel computing implementations.

It should also be noted that although the development of software for MDO has been an active field of research, the common limitation is that these architectures are proprietary or developed by universities and industries with restrictions on their use, and are sometimes specialised and difficult to expand for other applications. The framework was developed from scratch by progressively selecting, evaluating and assembling different software components in order to obtain detailed insight into the complexities involved in multi-objective optimisation and the process of MDO.

As will be detailed in this thesis, the framework has the following characteristics:

- **Modularity:** Designed with purpose, the system uses object-oriented principles in a modular approach. When a new mathematical test case or a new aeronautical design and optimisation problem needs to be studied, the user only has to develop a few input/output files and lines of code.
- **Scalability and Parallelisation:** New and more complex analysis tools and design modules can be implemented. Also, studies in parallel computation can be performed.
- **Post-processing:** Simple intuitive tools for visualisation of intermediate or final results.
- **Robustness:** The framework uses a robust evolutionary tool for optimisation. The framework and its methods have been shown to be robust and have been successfully coupled and validated for different single- and multi-objective mathematical test functions and coupled to a series of design and analysis tools.

- **Equivalence of Formulations:** Different multi-objective and multidisciplinary analysis formulations are evaluated directly with evolutionary methods. There are no derivatives or transformations on the mathematical formulation of the problem that introduce noise or require fine-tuning of the solution.

Of particular interest in this thesis is the application to Unmanned Aerial Vehicles (UAVs) and Unmanned Combat Aerial Vehicles (UCAV) systems. This vehicles are increasingly becoming important topics of aerospace research. As the complexity of these UAV systems arises, robust tools for multidisciplinary and multi-criteria analysis and optimisation are required. UAVs are usually designed for cruising at single flight condition while meeting other design constraints. The penalty for operating at off-design conditions can be significant. UAV aerodynamic performance might be improved if a multi-criteria multidisciplinary optimisation can be developed that considers several design points. Therefore, these vehicles provide a good environment in which to test the application of the methods to real-world problems in aeronautics.

1.1 Motivation

The goal of the work described in this thesis is to demonstrate the practical use of robust and efficient evolutionary algorithms (EAs) for multi-objective and multidisciplinary design problems in aeronautics, and to develop several application methods for this task. The intention is not to exclude human interaction in the process, but rather to facilitate the otherwise cumbersome exploration of the search space and exploit the method to find optimal solutions. Sound engineering definition and judgement of the problem are fundamental conditions provided by engineers; the tool and the benefit of computer evaluation present the engineers with a broader exploration of design space and non-intuitive designs that could be out of their scope.

At an academic research and industrial level, the methods and framework provide scientists and engineers with a platform on which numerical analysis and optimisation tools can be implemented and complex engineering problems can be solved.

1.2 Objectives and Unique Aspects of this Research

1.2.1 Objectives

The specific objectives of this thesis are:

- To demonstrate the need for, and develop an innovative, robust and efficient multidisciplinary and multi-objective design optimisation methods and a framework that uses evolutionary techniques that is applicable to the design and optimisation of several aircraft systems.

- To review the current state of research in the field of evolutionary computation and its applications to multidisciplinary design optimisation and multi-objective problems in aeronautics.
- The development of a comprehensive mathematical test-suite for multi-objective, goal programming and constrained optimisation problems.
- To conduct studies on parallel computations and hierarchical topologies of evolutionary algorithms with a scalable number of computers for real-world problems in aeronautics.
- The application of the methods to a set of real-world engineering problems ranging from conceptual to complex detailed design studies. These include conceptual and detailed studies for evolutionary optimisation of single aerofoil, multi-element aerofoil, wing and aircraft design.
- The specific application to Unmanned Aerial Vehicles (UAVs) and Unmanned Combat Aerial Vehicles (UCAV) systems.

1.2.2 Unique Aspects of this Research

The contributions of this thesis are:

- The development and application of robust evolutionary methods for multi-objective and multidisciplinary design optimisation problems in aeronautics and specifically their application to aerofoil, wing and aircraft design.
- The development of a unique framework in which different single, multi-objective and multidisciplinary design optimisation problems in aeronautics can be analysed and optimised.
- The development of a comprehensive and expandable test-suite for design, analysis and optimisation of mathematical multi-objective and constrained optimisation problems.
- The application of a hierarchical topology of evolutionary algorithms for aircraft multidisciplinary design optimisation with different fidelity models.
- The introduction of a subspace optimisation embedded within each node of a hierarchical topology of evolutionary algorithms.
- The implementation of the Nash equilibrium approach for aircraft conceptual multidisciplinary design optimisation which allows a rapid exploration of the design space.

1.3 Outline

This thesis considers several evolutionary methods, algorithms and framework developed and implemented by the author, that can be used to evaluate, optimise and solve single, multi-objective, multi-disciplinary conceptual and preliminary design optimisation problems relating to manned and unmanned aircraft systems. These problems can be highly non-linear, multi-modal, involve approximations, be non-differentiable or involve multiple objectives and physics and/or with convex, non-convex or discontinuous Pareto optimal fronts.

This thesis is organised as follows: Chapter 2 reviews the aircraft design process and provides background on theory, applications and limitations of different optimisation methods and multidisciplinary design optimisation (MDO). The requirements, formulation and implementation of the proposed methods are considered in Chapter 3. Chapter 4 analyses the performance of some of the methods for mathematical test functions; this highlights the benefit of the methods for multi-objective and constrained optimisation problems. Chapter 5 details studies in parallel evolutionary computation and hierarchical topologies of fidelity models; this provides an indication of the complexities, coupling and performance of the methods with parallel computing evolutionary techniques and analysis tools. Real-world applications in aeronautics with increasing levels of complexity are discussed in Chapters 6, 7 and 8; Chapter 6 applies evolutionary methods to aerodynamic shape optimisation problems. Chapter 7 is concerned with the application of methods for multi-objective and multidisciplinary wing design and Chapter 8 applies methods for aircraft design and optimisation problems. Chapter 9 discusses key conclusions and proposes avenues for research. Appendix A lists the papers arising or benefiting from this research, Appendix B summarises the methods and software developed and Appendix C details some of the analysis tools considered in this research.

The research undertaken in this thesis has made significant contributions to realising the benefits of robust evolutionary methods and algorithms for multi-objective and multidisciplinary design optimisation. In particular, practical applications to aerofoil, aircraft and wing design using a combination of low-to-medium fidelity solvers have been explored. Key challenges in the future will be to extend these ideas to incorporate a higher number of design variables, Design of Experiments (DOE) studies and the use of higher fidelity models.

Chapter 2

Design and Optimisation Methods

> *"The traditional code of science – that is, the objectives sought and the methods of investigation – cannot satisfy the requirements of our critical time, and this is why science has failed to measure up to the opportunities and obligations before it. The generally accepted ideas of what science is and what it is for are out of date and need radical revision."* C. J. Herrick.

2.1 Introduction

In this chapter, related work on the state-of-the-art of the aircraft design process, the principles of Multidisciplinary Design Optimisation (MDO) and Evolutionary Algorithms (EAs) are outlined and basic concepts are defined.

For contextual purposes, Section 2.2 gives an overview of the design process in aeronautics, Section 2.3 discusses different optimisation methods, their classification, benefits and limitations. Section 2.4 describes a multi-objective optimisation problem and different approaches for the solution, while Section 2.5 discusses theory, developments, formulations and applications of MDO. Section 2.6 provides the foundations of EAs and their application to problems in aeronautics. Section 2.7 describes a robust evolutionary optimisation technique. The chapter concludes by providing a summary, highlighting the limitations of current approaches for optimisation and the continuing need for the development of numerical tools for multi-objective and multidisciplinary optimisation.

2.2 Aeronautical Design Process

Even though the concepts described in this thesis can be extended and used in other engineering areas, this research will focus on the field of aeronautical engineering. Therefore, before proceeding with additional details, it is important to describe the traditional design process in aeronautics.

The aircraft design process can be divided into three major phases: conceptual, preliminary and detailed design. All these phases can benefit from the application of optimisation methods.

In *conceptual design*, basic questions such as size, weight, configuration arrangement and performance are answered. In this phase, a large number of possible alternatives are evaluated together with trade studies. In conceptual design, the design requirements are evaluated and studied to guide and evaluate different aircraft configurations. Conceptual design is a very fluid process and the configuration layout is changed constantly to incorporate new concepts and re-evaluate potential improvements. Trade studies and sophistication of design are improved as the concept design progresses in time. As the level of complexity increases, the level of detail increases and new phases of the design are considered.

In *Preliminary Design*, the level of understanding of the concept defined at the end of the conceptual design phase matures and the mathematical modelling of the outside surface of the aircraft is detailed with accuracy. Different experts in aerodynamics, control and structures perform complex analyses in their areas, usually with the aid of CFD and FEA codes. The use of sophisticated CAD tools, that accurately model the aircraft, are part of the preliminary design process, allowing different disciplines to work with a general configuration that follows the requirements validated in the conceptual design phase and allows for the introduction of new analysis of the current design phase. The final output of the preliminary design is an integrated proposal of the model for entering a full-scale development in the detailed design process.

Detailed Design includes specific details such as the proper location of holes for fasteners, and design of components such as hinges, brackets and racks. Some considerations on production design, whereby specialists determine how the aircraft will be fabricated, are also part of the detailed design. Actual structures are tested and a simulation of the control laws for the system is performed during this phase.

This thesis will focus on developing methods and a framework that are applicable to the conceptual and early stages of preliminary and conceptual design.

2.3 Traditional Methods and Emerging Techniques for Optimisation

Optimisation can be defined as a method of increasing the value of a numerical quantity by manipulating a series of design variables while satisfying a number of constraints. In engineering, for example, these quantities can take the form of speed, aerodynamic performance, weight and cost. In general, optimisation methods can be divided into enumerative, deterministic and stochastic methods [28, 59, 107]. Figure 2.1 illustrates examples of each type.

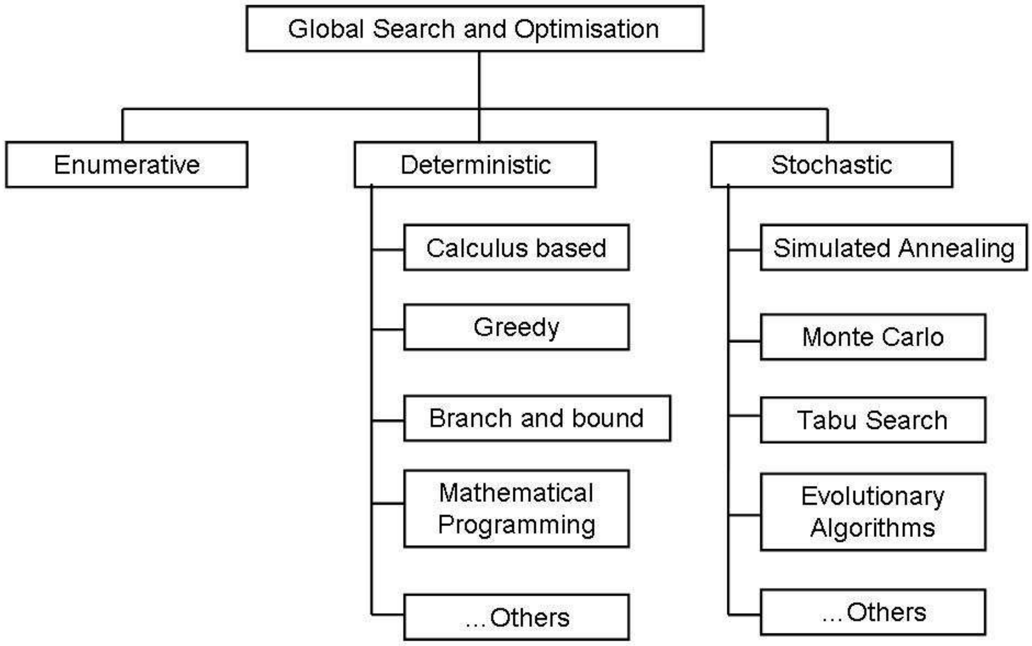

Figure 2.1: Optimisation approaches.

This thesis will focus on stochastic techniques and specifically on Evolutionary Algorithms (EAs), therefore it is not intended to describe other techniques in detail, but a brief description and overview is necessary in order to illustrate their merits and limitations.

Enumerative techniques comprise the most simple optimisation strategy, in this case, each possible solution in the search space is evaluated. It is easily seen that when applied to real world problems in engineering where the search space can be large, the technique is inefficient and computationally prohibitive.

Amongst the *deterministic* methods, the most common are the calculus-based gradient methods. These methods use information not only from the objective function but also from its gradient with regard to the design variables. These methods use the first and sometimes the second derivative sensitivity information to make a decision on the steps to take in the design process that lead to the optimum. The benefit of gradient-based methods is that they are able to find optimal solutions with fewer function evaluations. Some of these methods include the steepest descent, where the optimisation step is taken from the direction of the gradient vector, and Newton methods, which rely on second derivative information (Hessian Matrix) and exhibit a better rate of convergence. Extensive literature on the theory and real-world applications of these methods exists [177]. Greedy algorithms and Branch and Bound techniques also belong to this class but have not been extensively used for aeronautical design optimisation, therefore they will not be discussed here, but their theory and applications can be found elsewhere [177].

The third category, *stochastic*, contains methods such as Random walk, Simplex, Neural Networks, Simulated Annealing, Monte-Carlo, Tabu Search and Evolution Algorithms that include the subsets of: genetic algorithms, evolution strategies and particle swarm optimisation (PSO). Stochastic methods do not rely on any information apart from the value of the objective function. These methods require many function evaluations and also suffer from an expense in computational time as the number of design variables is increased. Simulated Annealing (SA) is a type of stochastic method which is based on the annealing of materials as they go through different cooling temperatures to stabilise their properties [176]. Tabu search methods have not seen broad application in aeronautics, hence discussion of these methods will be omitted here, but can be found elsewhere [177]. Monte Carlo techniques have been explored and exploited to some extent; theory and applications of these techniques can be found in Wenter [177].

Evolutionary Algorithms (EAs) are other stochastic methods which have been shown to be robust to solve many engineering problems and will be explained in detail in Section 2.6.

2.3.1 Traditional Optimisation Methods in Aeronautics

Since the conception of the aeronautical industry, engineers realised the benefits of introducing optimisation during the design process and used conventional deterministic optimisers such as steepest descent or conjugate gradient for this task [2, 75, 78]. As described by Jameson [75, 78], Malone [96] and Thomas [168], the introduction and advances in CFD and its coupling with optimisation tools opened a new range of developments and possibilities. The application of optimisation methods has allowed manufacturers to reduce drag and improve the performance of the aircraft, which in turn reduces operational costs and provides fuel savings [18, 144]. Examples of the application of optimisation methods for aerofoil, wing, aircraft design include:

2.3.1.1 Direct and Inverse Aerofoil Optimisation

Literature on aerofoil shape optimisation using traditional methods is extensive; some examples can be found in work by Pittman on supersonic aerofoil optimisation [137] and by Reuther *et al.* [147] on aerofoil optimisation using adjoint techniques. The use of a hybrid deterministic-adjoint optimisation technique for aerofoil design was studied by Nadarajah *et al.* [113]. Bernard *et al.* [19] and Kim *et al.* [82] used a viscous solver with an adjoint method to optimise the flow on a two-dimensional high-lift system.

2.3.1.2 Wing Optimisation

Wing design and optimisation has also been an active field of research; Martins *et al.* [102] for example, performed an aero-structural optimisation on a transport wing using sensitivity analysis information and high-fidelity analysis tools. The concept of different variable complexity analysis tools for the aerodynamics optimisation of a high-speed civil transport wing was studied by Hutchison *et al.* [73].

2.3.1.3 Aircraft Optimisation

Full aero-structural optimisation of a business jet was studied by Martins [101]. In his research Martins used an adjoint method to compute sensitivities, with considerable reduction in computational cost. Nemec *et al.* [116] studied the use of full Navier-Stokes equations for aerodynamic design and Vassberg illustrated an aerodynamic analysis and optimisation on a complete aircraft configuration [172]. Jameson *et al.* [77] performed a complete aircraft aerodynamic shape optimisation using conventional optimisation techniques and Jameson and Vassberg [78] studied a series of numerical methods and algorithms for aerodynamic analysis and design.

These examples show the broad applicability of optimisation to aeronautical engineering problems, but as will be progressively discussed through this chapter, the applicability of conventional optimisers is reduced when the complexity of the problem increases and involves multiple physics and objectives.

2.4 Multi-objective Problems

Often aeronautical design problems require a simultaneous optimisation of conflicting objectives which need to be maximised or minimised while satisfying a number of equality/inequality constraints. In general, a multi-objective optimisation problem can be formulated as:

Maximise/ Minimise:

$$f_i(x), \;\; i = 1, ...N, \tag{2.1}$$

subject to constraints:

$$g_j(x) \leq 0 \; j = 1, ..., M, \; h_k(x) = 0 \; k = 1, ..., K$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)}, \, i = 1, ..., n$$

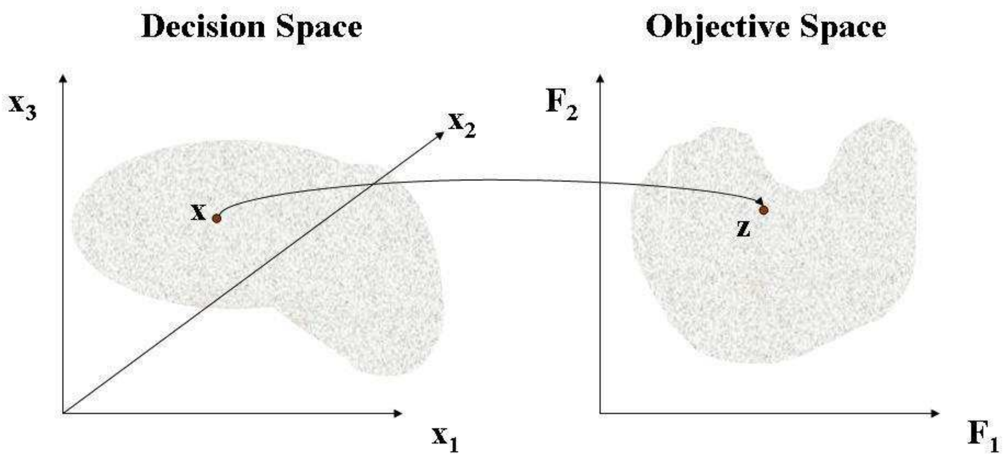

Figure 2.2: Representation of decision variables and objective space.

where f_i are the objective functions, N is the number of objectives, x is an n–dimensional vector where its arguments are the decision variables $(x_1, x_2, ...x_n)$, $g_j(x)$ and $h_j(x)$ are the inequality and equality constraints and the last set of constraints restrict each design variable to take a value between a lower $x_i^{(l)}$ and an upper bound $x_i^{(u)}$. These bounds constitute the decision variable space D, or simply the decision space. A solution x that does not satisfy all the M and K constraints and all the variable bounds is called an infeasible solution. On the other hand, any solution x that satisfies all constraints and variable bounds is known as a feasible solution. The N objective functions considered in the above formulation can be minimised or maximised.

Also, as described by Deb [42], one of the characteristics of multi-objective problems is that the objective functions constitute a multi-dimensional space which is called objective space. That is, for each solution x in the decision variable space, there is a point in the objective space denoted by $f(x) = z = (z_1, z_2, ..., z_N)$. Figure 2.2 illustrates the concept. This is of significant importance as the decision space might be well defined and easy to visualise, but the objective space might be quite complex, with discontinuous, convex, non-convex Pareto fronts.

A common way to represent the solution to a multi-objective problem is by the use of the concept of Pareto optimality or non-dominated individuals [131]. Figure 2.3 shows the Pareto optimality concept for a two conflicting objectives problem. A solution to a given multi-objective problem is the Pareto optimal set, found using a cooperative game which computes the set of non-dominated solutions. This spans the complete range of compromised designs between the two objectives. Formally, the Pareto optimal set can be defined as the set of solutions that are non-dominated with regard to all other points in the search space, or that they dominate every other solution in the search space except fellow members of the Pareto optimal set. For a minimisation problem, a vector x_1 is said to be partially less than x_2 if $\forall_i : \ f_i(x_1) \leq f_i(x_2)$ and $\exists_i : \ f_i(x_1) < \ f_i(x_2)$. For a problem in M objectives, this is called the relationship operator. In

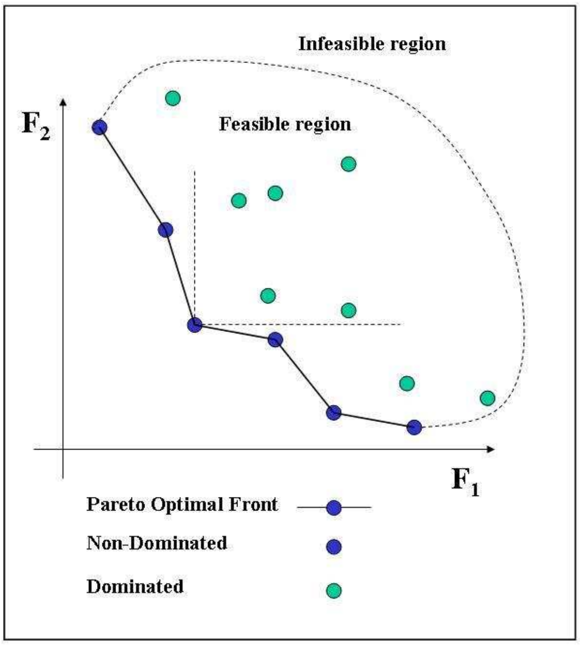

Figure 2.3: Pareto optimality.

practice, an approximation to the continuous set, by assembling $ParetoSet = [x_1^*, x_2^*, ...x_\mu^*]$ is computed.

Figure 2.4 also shows the Pareto set for four different scenarios for the minimisation and/or maximisation of two objectives. The solid curves represent the optimal Pareto sets. In all cases the Pareto set consists of a particular edge of the feasible search region. As can be seen in this figure, the Pareto front could be convex, non-convex or discontinuous. Another characteristic of multi-objective problems is on finding global and local Pareto sets. This concept is illustrated in Figure 2.5. Ideally it is desirable to find the global Pareto front, but in a practical sense this is not always possible, due to computer resource limitations; the search for a solution might be stopped because the time allocated or the number of functions evaluations permitted has been completed.

There are many variants and developments of multi-objective approaches; these include the lexicographic approach, traditional aggregating functions and Pareto and Nash approaches [100]. A good review of these approaches can be found in [28, 42]. These concept will be extended and applied in combination with MDO and EAs in the following sections.

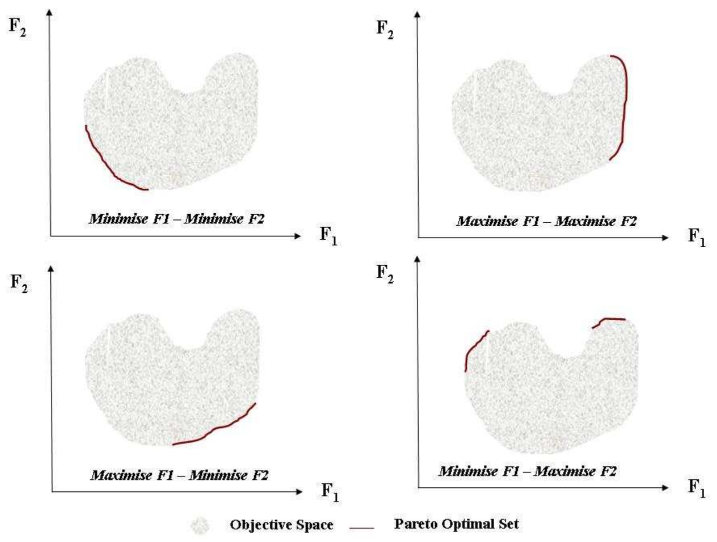

Figure 2.4: Pareto optimal solutions for four possible scenarios.

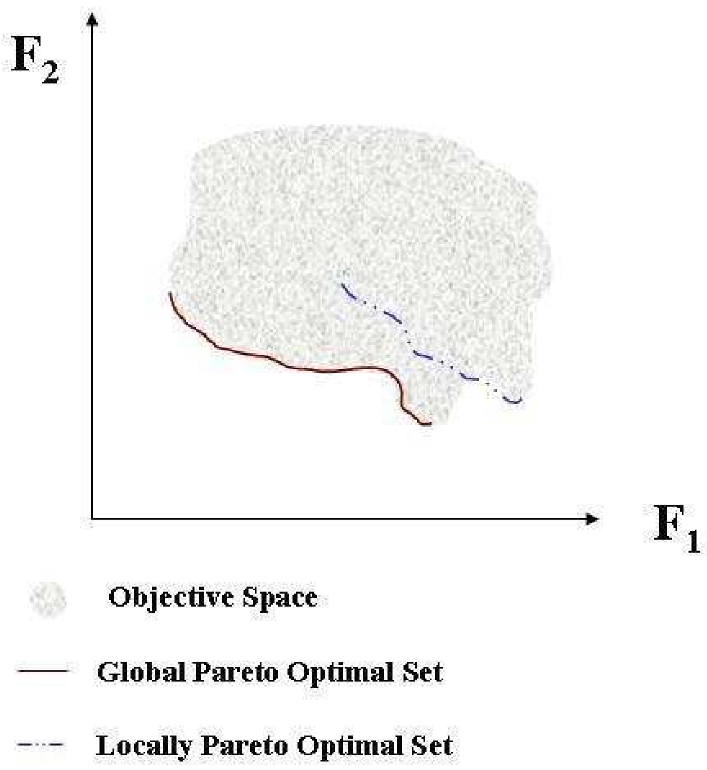

Figure 2.5: Locally and global Pareto optimal solutions.

2.5 Multidisciplinary Design Optimisation (MDO): Overview and Methods

2.5.1 Definition

In aerospace engineering the designer is usually presented with a problem which involves considering not only one objective but numerous objectives and multi-physics. A systematic approach that accounts for the coupling between the disciplines and variables regarded as Multidisciplinary Design Optimisation (MDO), is required. In aircraft design, for example, the multi-physics include aerodynamics, structures, propulsion and control. These multi-physics are interrelated and interdisciplinary constraints must be satisfied to solve the problem. The area of MDO has matured as a separate discipline now with different journals, specialised technical committees, conferences and publications devoted solely to the topic.

The need for, and benefits of MDO are clear, given that a small improvement in the performance of the aircraft can be significant. An MDO process also allows an evaluation of the constraints on multiple disciplines from the early stages of the design, thus the expense of re-designing an aircraft system is reduced [13].

2.5.2 Challenges and Needs for MDO

The aircraft design industry has realised the benefits of optimisation and of MDO during the design process, but as traditional concepts are revised, new configurations are studied and market pressures and industrial requirements emerge, the need for developing alternative numerical methods tools and their application to current designs is a requirement.

A comprehensive survey of MDO methods, their development and limitations is provided by Sobieski *et al.* [161]. The research classified the different methods and highlighted some important needs, including a multi-platform operation, the use of parallel computation to improve computational expense and space visualisation as the designer might be interested in the space around the optimum rather than the optimum itself. In most of the methods described in the survey, the optimisation algorithms for MDO use traditional gradient methods for the solution, but as has been discussed, these methods have some limitations.

Sobieski and Hafka [162] evaluated recent developments in multidisciplinary aerospace design and optimisation and identified several categories of problem formulations and also two other main challenges for MDO, the computational expense and organisational (architectural) complexity. A more recent survey by Giesing and Barthel [56] identified several industrial applications and summarised some of the needs for MDO. Most of the applications are related to detailed design; few applications are developed for conceptual or preliminary MDO studies. On the classification of needs, their research also describes how a MDO framework should be flexible to accept whatever function is needed and should address the issue of low- and high-fidelity models, but not to compromise the optimisation. It also points out the need for efficient models that describe the physics to keep computing time at a reasonable level. A critical aspect mentioned in the paper is the need for accurate realistic design by identification of the constraints, mechanisms and underlying physics of the various disciplines involved.

An open issue on MDO studies is the fact that many high-fidelity processes such as Navier-Stokes, CFD or FEA are complex to couple MDO as many of them are not automated, robust or fast enough. Also, the need for approximation techniques, such as RSM or DACE, arises because of the computational expense of using an analysis code for all the evaluations during the optimisation process and also from the fact that some analysis codes cannot be directly integrated with the MDO architecture.

On MDO framework architectures, there is a requirement for more efficient, robust flexible framework architectures and methods with industrial strength codes. These codes should be easily coupled and reconfigurable, and adaptable to commercial solvers. One of the problems with current MDO architectures is that these are usually developed at universities and in industries with restrictions on their use, and sometimes these are specialised and difficult to expand for other applications.

On the topic of optimisation tools for MDO, it is important to determine the presence of multi-modalities, non-linearities and multiple objectives that might cause a traditional deterministic method to fail. The computational cost of the gradients and the presence of multiple disciplines is also an important consideration. Therefore, the continuing challenge has been on developing and improving numerical optimisation techniques and enhancing their speed and robustness for their use within MDO. One of the emerging optimisation techniques for MDO is EAs, but they have found limited applications in MDO due to the computational burden associated with them.

There are increasing applications of EAs for MDO problems; Giesing and Barthel [56], for example, presented a short discussion on supporting design space search methods such as Evolutionary Algorithms (EAs), and explained how they are gaining popularity for MDO because they are simple to couple with analysis modules and do not incur the cost of computing derivatives. A detailed discussion on the developments of EAs for optimisation and MDO will be presented in the following sections.

2.5.3 Multidisciplinary Design Analysis (MDA) and Optimisation

When considering an MDO problem several multidisciplinary design analyses (MDA) have to be performed. A multidisciplinary analysis is the manner in which different disciplines interact and conform to a solution. A typical multidisciplinary analysis is depicted in Figure 2.6. The solution to an MDA requires the solution to satisfy simultaneously all disciplinary analyses. Following the nomenclature in Renaud and Shi [146], an MDA considers three types of variables, *global* or *shared* variables z, that are variables required by more than one discipline, *disciplinary* or *local* input variables x_i that are used in the calculations concerned with the i^{th}discipline and finally the *state* variables y_i, as for an N-discipline problem:

$$y_i = y_i\left(x_i, y_j, z\right),\ j = 1, ..., N,\ \ j \neq i. \tag{2.2}$$

The analysis of a multidisciplinary system consists of determining the disciplinary state parameters y that correspond to a specific set of local and shared input parameters x and z. The traditional solution involves a sequence of iterations between the different disciplinary analyses. When the convergence is reached, all state variables are compatible and the system is considered to be the so-called state of multidisciplinary feasibility. The solution to this system of equations is complex and has motivated active research in the field of MDO, with several approaches having been developed.

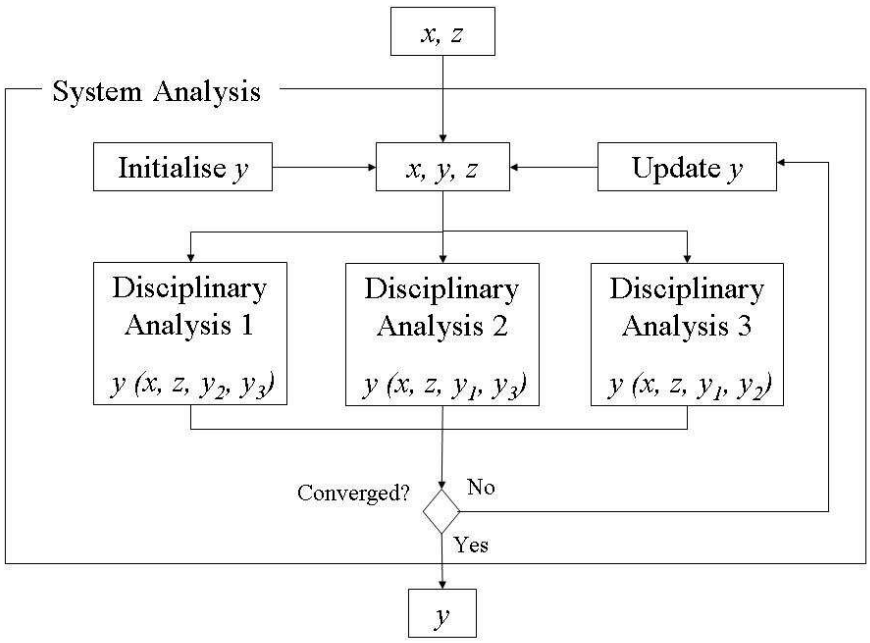

Figure 2.6: Multidisciplinary Analysis.

2.5.4 Approaches to MDO

The selection of the appropriate optimisation architecture is of great importance for an efficient solution of MDO problems, as one MDO architecture may find a feasible solution without high computational expense while other solutions might be too slow or even fail [23, 48]. There are several approaches for MDO, the most common being: the Multidisciplinary Design Feasible (MDF) method [4], the Individual Discipline Feasible (IDF) method [31], the Collaborative Optimisation (CO) method [23], the Concurrent SubSpace Optimisation with Response Surfaces (CSSO/RS) [146] and the Bi-Level Integrated System Synthesis with Response Surfaces (BLISS/RS) method [85]. This thesis develops algorithms for the first two and a variant of CO.

2.5.4.1 Multidisciplinary Design Feasible (MDF)

This formulation has been extensively used in the field of optimisation and can be considered as the standard approach. It is also known as single-level optimisation or All-in-One (A-i-O). The MDF can be stated as:

$$\begin{array}{rl} Minimise: & f\left(z,\ y\left(x,y,z\right)\right) \\ Subject\ to: & g\left(z,\ y\left(x,y,z\right)\right) \leq 0 \end{array} \tag{2.3}$$

where f is the objective function and g represents all systems and/or disciplinary constraints. This method is illustrated in figure 2.7.

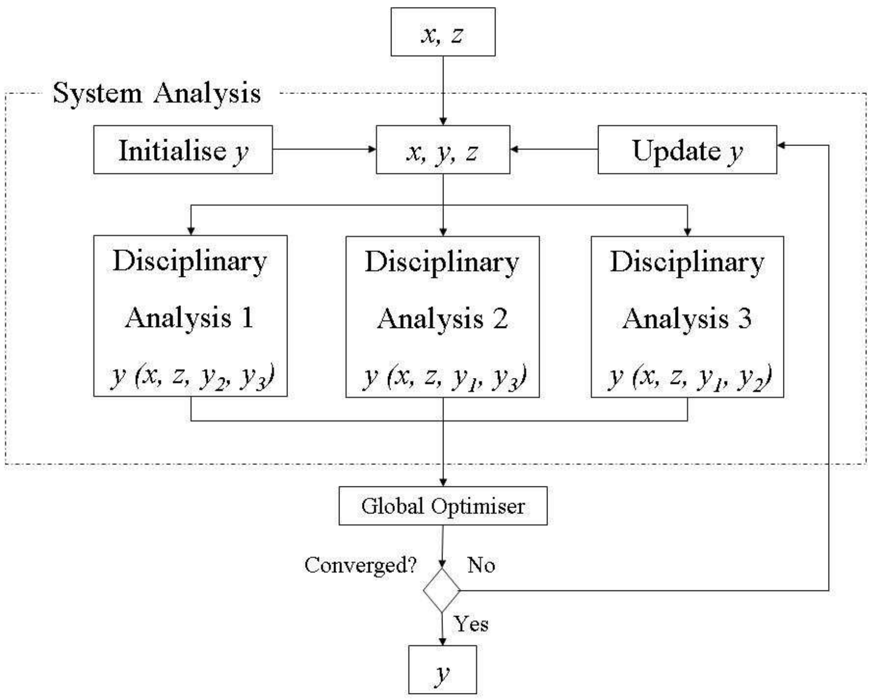

Figure 2.7: Multidisciplinary Design Feasible (MDF).

In this approach the set of design variables z is evaluated by an integrated analysis. This analysis evaluates the coupled system of equations, guarantees interdisciplinary constraints and returns the objective function that is manipulated by the optimiser in search of the optimal solution. To achieve a design that is physically feasible, this approach usually requires a computationally expensive iterative process during each integrated analysis call.

As an example, minimisation of wing drag in a two-discipline problem: structures and aerodynamics, may be considered. The design variables can be, for example, wing sweep, wing span, twist or taper. With an integrated approach the optimiser sends the values of these variables to an integral system of equations that represent the aerodynamics and structures analysis coupling. These analyses are iteratively evaluated to conform and converge on each discipline to a consistent solution. The objective function and constraints are then evaluated and manipulated by the optimiser to improve the design. This approach is conceptually very simple. Once all disciplines are coupled to form a single multidisciplinary analysis module, the same techniques that are used for a single-discipline optimisation can be used. The disadvantage of this approach is that the solution of a single system could be very expensive and does not enable the potential decoupling of the individual disciplines into analysis modules that can be computed in parallel.

As will be illustrated in Chapter 3, the benefits of parallelisation can be exploited by using an Evolutionary algorithm which sends candidate individuals to different processors where they are evaluated by the complete MDA analysis and returned to the optimiser. There is no bottleneck; once a solution is available it is reincorporated back into the optimisation process.

2.5.4.2 Individual Discipline Feasible (IDF)

This method belongs to the distributed analysis architectures and differs from the standard analysis by the way in which the different disciplines are decoupled. The disciplinary analyses are decoupled but it keeps the optimisation at the system level to form a single-level decomposed optimisation problem. The subsystems are individually analysed and the optimisation is performed for the system as a whole. Constraints that impose multidisciplinary feasibility are introduced by using extra coupling variables. In this approach the disciplines are individually feasible but the complete system may not be feasible until the optimisation process converges. The IDF optimisation statement is:

$$
\begin{array}{rc}
Minimise: & f\left(z,\ y\left(x,\ y',z\right)\right) \\
Subject\ to: & \begin{array}{c} g\left(z,\ y\left(x,\ y',z\right)\ \right) \leq 0 \\ y' - y\left(x,\ y',z\right) = 0 \end{array}
\end{array}
\tag{2.4}
$$

where y' are the auxiliary disciplinary input variables. The second constraint ensures multidisciplinary feasibility when y is equal to y'. The IDF approach is illustrated in Figure 2.8.

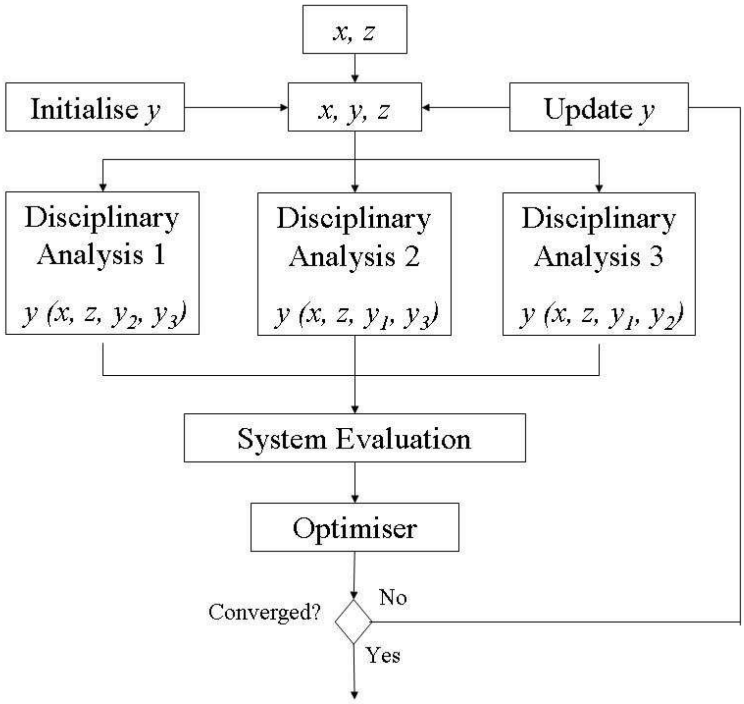

Figure 2.8: Individual Discipline Feasible (IDF).

In this case the design variables $x,\ y,\ z$ are sent to each discipline; these are analysed by each discipline block (that is, aerodynamics-structures). These analyses return an evaluated solution that satisfies the governing equations and domain-specific constraints on each discipline, but that might not be interdisciplinary compatible. The system-level optimiser is responsible for ensuring compatibility of the overall solution, guaranteeing satisfaction of interdisciplinarily constraints and minimising the objective function f.

A distributed architecture has several organisational and computational advantages over a traditional integral standard approach.

Computational advantages:

- A reduction in re-iteration and communication requirements.
- The use of parallel heterogeneous platforms to evaluate the different disciplinary analyses.
- Removal of iteration loops.
- A reduced level of disciplinary requirements.

Organisational advantages:

- Increased autonomy and disciplinary expertise in the design process.
- The benefits of being able to alter a part of the analysis without having to re-pose the whole problem.
- Resembles the division into areas of expertise in most design companies.

2.5.4.3 Collaborative Optimisation (CO)

The CO approach is a two-level hierarchical scheme for MDO which decomposes and reformulates the problem as a bi-level optimisation, with the top level being the system optimiser that optimises the multidisciplinary variables or system level targets z to satisfy interdisciplinary compatibility constraints c while minimising the objective f. The problem is decomposed in the analysis and subspace optimisers that are integrated within each analysis block.

The system is optimised at the coordination level by determining the target values for subsystem responses and shared design variables with compatibility constraints that ensure multidisciplinary feasibility. The optimisation objective at each subsystem is to match as closely as possible these target values while satisfying local disciplinary constraints. Through subspace optimisation each block is given control over its own set of design variables and changed to satisfy its own domain-specific constraints. Specific knowledge of each of the other groups or design variables is not required. The objective of each subspace analysis is to agree upon the values of the interdisciplinary variables with the other groups. Again, a system-level optimiser is used to ensure compatibility at the overall solution, guarantee satisfaction of interdisciplinary constraints and minimise the objective function.

Similar to the IDF approach, the multidisciplinary feasibility in CO is reached at the end of the process. If the target values corresponding to the shared and state variables are z and y', respectively, the system level optimisation can be written as:

$$\begin{array}{rl} Minimise: & f\left(z,\, y'\right) \\ Subject\ to: & c\left(z,\, z^{*},\, y',\, y\left(x_i^{*},\, y',\, z_i^{*}\right)\right) = 0 \end{array} \tag{2.5}$$

where c represents the compatibility constraints, one for each subsystem, and takes the form:

$$c_i = \left(z - z_i^{*}\right)^2 + \left(y' - y\left(x_i^{*},\, y',\, z_i^{*}\right)\right)^2 \tag{2.6}$$

where the asterisks (*) indicate optimal subsystem values.

At the disciplinary level i^{th}, the subsystem optimisation can be stated as:

$$\begin{array}{rl} Minimise: & c_i\left(z,\ z_i,\ y',\ y\left(x_i,\ y',\ z_i\right)\right)=0 \\ Subject\ to: & g\left(x,\ z,\ \ y\left(x,\ y,\ z\right)\right)\leq 0 \end{array} \tag{2.7}$$

where the objective c_i is of the same form as the constraints at the global level. The CO approach is illustrated in Figure 2.9.

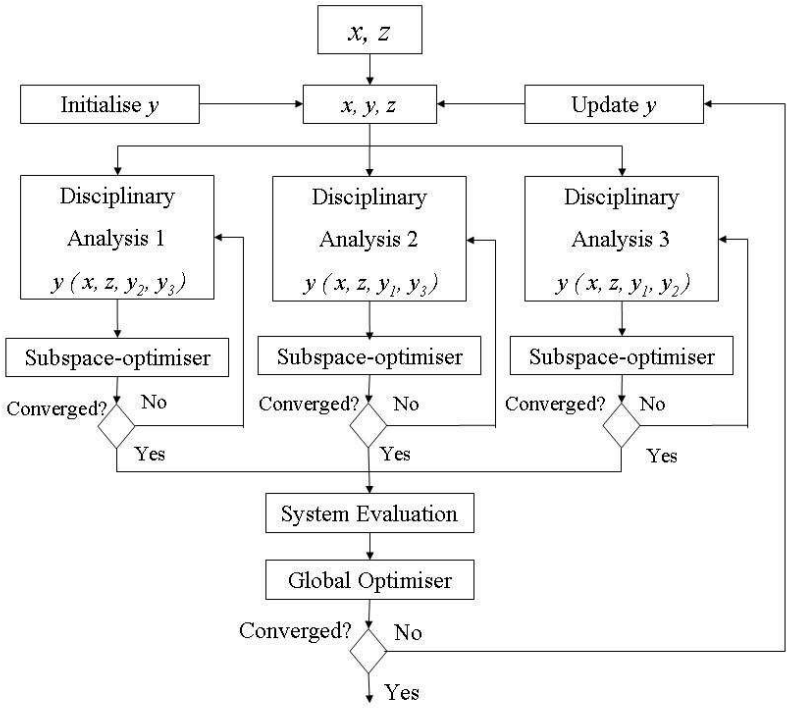

Figure 2.9: Collaborative Optimisation (CO).

2.5.5 Criteria and Performance of MDO Implementations

Alexandrov and Lewis [5] classified different MDO implementations and identified several criteria to classify, analyse and evaluate approaches to MDO. They state the distinction between formulating the MDO problem and solving the resulting computational problem. This technique involves two major elements: posing the problem *formulation* as a set of mathematical statements amenable to a solution and then defining a procedure *algorithm* to solve the problem once it is posed. As explained in the same paper, this distinction is crucial and is often blurred in presentations of new MDO approaches.

According to this research, the main attributes for an MDO *formulation* are consistency, well-posedness, equivalence of formulations, optimality conditions and sensibility to solutions, while *algorithm* considerations are on local convergence rates, global convergence properties and iterative cost. Their research also provides guidelines and poses several questions for new formulations and analysis for MDO methods. Examples of these questions are: how is the original MDO problem formulated? and does the formulation lead to an optimisation procedure that is not amenable to solution by existing optimisation algorithms?

Another concern about the performance and solution of MDO problems is on the use of different fidelity analysis tools for MDO, due to the computational expense involved in the process. For this reason, continuing research has been devoted to developing MDO formulation and in combination with variable fidelity analysis, approximation techniques and Design of Experiments (DOE) theory. Results indicate that these are possible avenues for research, as they lower the computational expense while maintaining the robustness of the solution [58]. In Chapter 3 the method proposed in this thesis will be checked against these criteria to determine its benefits and drawbacks.

2.5.6 Parallel Computing Strategies and MDO

There are different parallelisation strategies that can be exploited by MDO methods to reduce the computational expense. These include the master-slave concept whereby different candidates are sent to remote nodes. The master is responsible for distributing and assembling the information while satisfying constraints. Another strategy for large-scale systems with multiple disciplines is parallelisation of the different discipline blocks [4, 86].

A concept that can also be extended and exploited by MDO is the asynchronous evaluation concept described in Section 2.7.2. As will be described in that section, the parallelisation can be achieved by sending candidate individuals to remote nodes that evaluate them with each disciplinary analysis. The evaluated designs are returned out of order and stored in a temporary buffer from which the optimiser selects. There is no bottleneck; computers do not remain idle while others finish their work, once a solution is ready it is returned to the optimiser [179].

2.5.7 Approximate/Surrogate Models and MDO

The use of high-fidelity analysis CFD and FEA software for aeronautical system design is now a common practice at university and industrial levels. However, their use for optimisation and MDO is limited by the time and cost of performing a number of function evaluations during the optimisation process. One alternative is to construct an approximate model. When compared to the full analysis using high-fidelity tools, the approximate model requires very little time to evaluate the objective function. Therefore, there are savings in computational time. In order to build an approximation for optimisation, a series of steps have to be followed:

1. Define the experimental design sample.

2. Define a model to represent the data.

3. Fit the model to the observed data.

This process is illustrated in Figure 2.10.

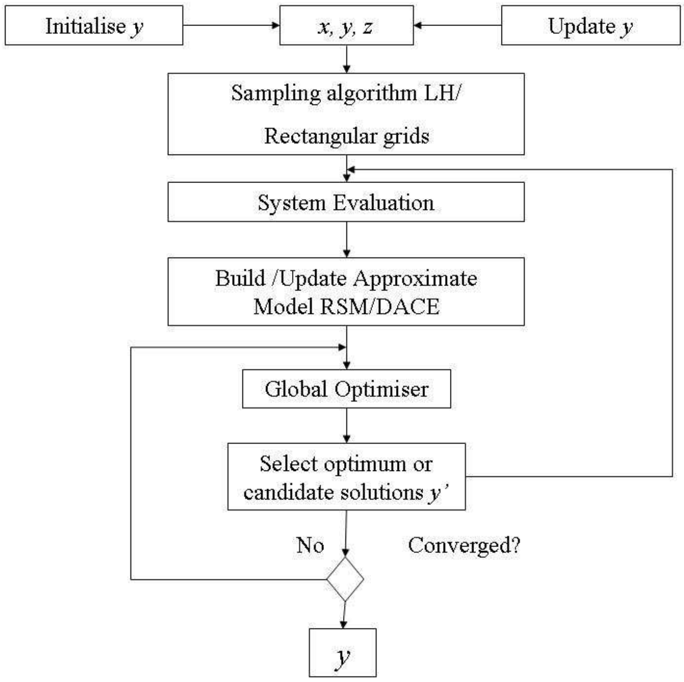

Figure 2.10: MDO using surrogate/approximate models.

2.5.7.1 Sampling Methods

Among the sampling methods we have the orthogonal/rectangular grid, Latin Hypercube (LH) and Improved Latin Hypercube Sampling (ILHS). Theory on these methods is quite extensive and can be found in the literature [14, 91, 150, 175]. As will be indicated in Chapter 3, the LH and orthogonal grid sampling are implemented in this thesis.

2.5.7.2 Model Definition

Among the model approaches we can find the Response Surface Model (RSM), Neural Networks and DACE (Kriging). The most widely used approximation technique is the polynomial model (RSM); this is because of the simplicity and ease of implementation. Theory on RSM can be found in Myers and Montgomery [112]. The main drawback of this technique is that it is not suitable for representing multi-modalities and non-linearities.

Design and Analysis of Computer Experiments (DACE)/ Kriging [150] is a more recent approach developed in the field of spatial statistics and geostatistics and, although more complex than polynomial approach, has gained wide popularity on its implementation to aircraft design problems [58, 91, 150]. Theory and applications of Kriging/DACE modelling are extensive and will be omitted here ,as they are not directly connected with the present work, but can be found in Srivastava *et al.* [165]. As will be described in Chapter 3, the multi-objective multidisciplinary framework implements the DACE technique.

2.5.8 MDO Applications Using Traditional Deterministic Methods

The applications of MDO for aeronautical problems are rapidly increasing [20, 54]. Some requirements, benefits and drawbacks of MDO are described in several applications to aeronautical problems. Wakayama and Kroo [173] applied MDO for subsonic wing platform design; their results highlighted the importance of considering a flexible architecture coupled with appropriate detailed analysis for structural and aerodynamic analysis to achieve realistic designs.

Bartholomew [13] explains the role of MDO within aerospace design and progress toward an MDO capability, based on lessons learned from research activities within the European community. The study stresses the importance of the incorporation of MDO in the design process at an industrial level. It also points out the requirement for an MDO framework to provide a flexible user interface definition and to monitor the MDO progress, and stresses the fact that the benefits of MDO provide means to avoid fragmentation with current recursive or intuitive methods that are time-consuming and limit the efficiency in the design.

The benefits of MDO, parallel computing and variable fidelity models for aircraft design have been studied by Giunta *et al.* [57, 58]. In these studies Giunta compared the use of RSM and DACE techniques and obtained optimum realistic aircraft design configurations. Other applications of MDO in engineering are related to other complex systems, such as ground vehicle design [16, 85, 104]. These studies, although not directly related to aircraft or aircraft component design, provide an indication of the complexities of the MDO process and the benefit of parallel computing when large space systems are analysed.

2.5.9 Limitations of Current Optimisation Techniques for MDO

As discussed in the previous sections, most of the optimisation methods used for MDO use traditional deterministic techniques. While the use of these methods is largely successful and efficient at finding optimal global solutions, problems do arise. Gradient-based methods usually work best with unimodal functions, but their effectiveness decreases with the presence of local optima or ridges in the fitness landscape. Also, the presence of numerical noise inhibits the application of many gradient-based optimisation techniques. The numerical noise causes an inaccurate calculation of gradients, which in turn slows or prevents convergence during optimisation [57, 58]. In aircraft design, for example, the problem of numerical noise is of special concern when an accurate solution is sought through a high-fidelity analysis but the computation of gradients is complex and a single aerodynamic or structural analysis might take several CPU hours on a supercomputer.

An emerging optimisation technique for MDO is Evolutionary Algorithms (EAs); these techniques are robust in finding optimal solutions for single- and multi-objective problems, but have found limited applications to MDO due to the computational burden associated with them. The challenge is then to study, develop, apply and improve the speed and robustness of these methods so that confident applications and use within MDO is possible.

It is important to highlight that in this work the use of EAs will be restricted to conceptual and preliminary MDO studies where the number of variables is still relatively small, less than a hundred, and where the use of EAs is still of potential benefit. On a larger scale, the use of EAs can be extended for an increased number of variables and coupled with other techniques such as Design of Experiments (DOE).

2.6 Evolutionary Algorithms (EAs)

As indicated in the previous sections, optimisation and MDO in aeronautics are complex tasks. Therefore, it is desirable to use robust optimisation tools that can explore the design envelope and capture the global optimal solution. EAs are one of such techniques. In this thesis the Hierarchical Asynchronous Parallel Evolutionary Algorithm approach developed by Whitney [179] is extended for multidisciplinary and multi-objective analysis are introduced. Before describing the particulars and extensions of this technique, the following subsections will provide the reader with the foundations behind the theory, developments and limitations of EAs.

2.6.1 Fundamentals of Evolutionary Algorithms

Evolutionary algorithms are based on the Darwinian theory of evolution [37] in which populations and individuals evolve in a search space and adapt to the environment by the use of different mechanisms such as mutation, crossover and selection. These features have produced a series of different evolutionary algorithms that resemble biological evolution and natural selection. Independent of each other, different researchers in the 1960s started applications to stochastic search inspired by the theory of evolution. Broadly speaking, Evolution Algorithms cover: Evolutionary Programming (EP), Evolutionary Strategies (ES) and Genetic Algorithms (GAs).

Evolutionary Programming (EP): In 1992, L. J. Folgel [50, 51] studied and developed an artificial intelligence method by evolving mathematical formulation to automatically predict a series of binary time series and defined EP. In 1980 he extended his research into application

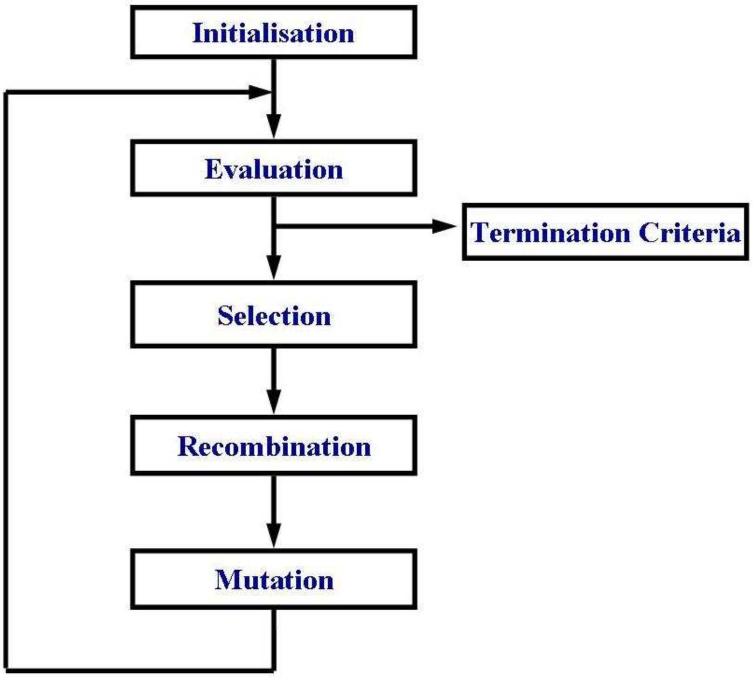

Figure 2.11: General flowchart of Evolutionary Algorithms.

to different tasks including machine learning and optimisation [50, 51]. In EP each parent generates an offspring by Gaussian mutation and better individuals, both among parents and their offspring, are selected as parents for the next generation.

Genetic Algorithms: GAs have their origins in and are accredited to the work by Holland in 1975 [71], although Goldberg [59] extended and generalised the concept. The following features distinguish GA from other evolution-based computational methods:

- The use of a bit or, more recently. a real-value string representation.
- The representation is at the level of genotype instead of phenotype.
- Crossover as the primary method for reproduction.
- Use of proportional selection.

A basic genetic algorithm, regarded as SGA, proposed by Holland [71] is illustrated in Figure 2.11 and Algorithm 1.

Algorithm 1 Simple Genetic Algorithm.

```
Initialise Population,
Evaluate population,
while stopping condition not met,
    Select solutions for next population,
    perform crossover and mutation,
    Evaluate population,
loop
```

Evolutionary Strategies (ES): Evolution Strategies were developed by two students at the Technical University of Berlin, Ingo Rechenberg and Hans-Paul Schwefel [11], in 1964. Since then different researchers have extended, modified the basic reproduction mechanisms and applied them to different problems. The common features in ES are:

- ES model organic evolution at the level of individuals' phenotypes.
- Use of real-coding design parameters.
- The means for evolution in ES are mutation and deterministic selection.
- Strategy parameters and adaptation are keys for success of ES.

Evolution strategies are of particular interest to this thesis and will be explained in more detail in Section 2.6.3.

2.6.2 Mechanics of EAs

EAs share common elements: representation of individuals, fitness function, and an iterative selection based on fitness, recombination, mutation, elitism and the dilemma between the exploration and exploitation of the search space.

2.6.2.1 Representation of Individuals

There are many types of representations, the most common being binary and floating point. The binary representation uses a bit string to represent an individual. With this representation, the real design variables are transformed into binary numbers that are concatenated to form a chromosome. This chromosome encodes the total number of design variables of the problem.

In floating-point representation, a vector of real numbers characterises an individual. In aerofoil design, for example, the design variables are the control points for a Bézier or Spline curve that generates the aerodynamic shape. Figure 2.12 illustrates the concept.

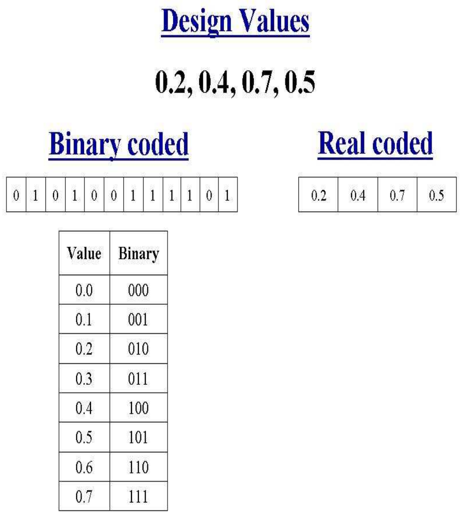

Figure 2.12: Real-coded and binary representation.

It has been reported by different researchers that real-coded EAs have outperformed binary-coded EAs in many applications [107]. The explanation for this is that in a binary representation the variables are concatenated to represent an individual and this results in a big string length which is difficult to handle. Another problem is that the binary representation of real design parameters presents a difficulty with what is called “hamming cliffs” which is the discrepancy between the representation space and the problem space [126]. As a consequence, it is difficult for a binary-coded EA to exploit the search in the vicinity of the current population [107]. On the other hand, a real-code representation is conceptually closer to the real design space and the length of the string vector is equal to the number of design variables.

2.6.2.2 Fitness Function

Similar to the concept of survival of the fittest in nature, EAs use a fitness function to evaluate the performance to determine the quality of the vector string and to define whether the individual is a suitable candidate for the next generation. The fitness function is a critical aspect of EAs; a general rule is that it should reflect as closely as possible the desired real aspect of the solution. Examples of fitness functions in aeronautics can be, for example, drag minimisation (c_d), aerodynamic performance $(Mach \times L/D)$ or gross weight (Wg).

2.6.2.3 Evaluation

Evaluation consists of the means by which each individual in the population is evaluated. These could be, for example, an analytical function or a complex CFD or FEA analysis. For a real-world problem in aeronautics the means for evaluation can be a panel code or a CFD code that evaluates the flow around the aerofoil and provides an estimate of lift and drag coefficients, or an aero-structural analysis that computes aerodynamic performance and structural weight that can be used to compute the fitness function.

2.6.2.4 Selection

The selection process is where individuals compete and are selected to produce offspring for the next generation; design candidates are selected by comparing their fitness values. Several parent selection techniques have been proposed, but the application of them is usually problem-dependent [59, 107]. A method that is normally used is fitness proportional selection. In this case the selection probability of the individuals is calculated by dividing their fitness by the sum of all the other fitnesses of the individuals.

Parents can also be selected by *roulette wheel* selection [59] or Stochastic Universal Sampling [107]. Figure 2.13 shows a schematic representation of roulette wheel selection. An individual is selected by spinning the wheel, which is divided according to the selection probability.

Individual	Fitness	Selection probability
Individual 1	4	0.4
Individual 2	3	0.3
Individual 3	1	0.2
Individual 4	1	0.1

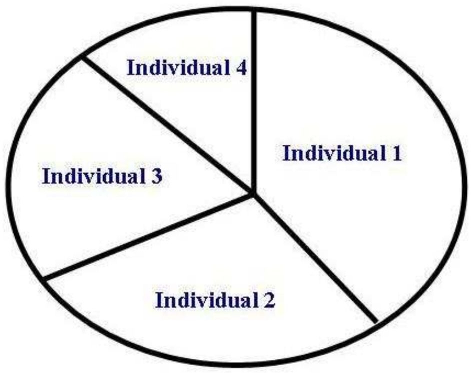

Figure 2.13: Roulette wheel selection.

Tournament selection operates by choosing some individuals randomly from a population and selecting the best from this group to survive in the next generation. Its simplest form is binary selection, whereby two random pairs of individuals are selected from the population and the pair with higher fitness is copied to the mating pool or population. Another method for selection is *ranking*, whereby individuals are ranked by their fitness values. The best individual receives rank 1, the second receives rank 2 and so on. A selection probability is reassigned in accordance with the ranking order.

An appropriate level of selection pressure is critical for the success of the evolution. If too much pressure is applied there could be loss of diversity, and premature convergence occurs. This is because the population is not infinite, and some individuals who are comparatively highly fit but not optimal rapidly dominate the population. The basic idea is then to control the number of reproductive opportunities that each individual has in order to prevent highly fit individuals taking over the population. On the other hand, if a low selection pressure is imposed, the search is ineffective and will take excessive time for convergence.

2.6.2.5 Recombination

Recombination, also known as crossover, is the process in which two or more parent individuals (or chromosomes) are combined to produce an offspring chromosome (individual). Recombination is necessary in those cases when the offspring is to have multiple parents, since mutation by itself provides no mixing of the chromosomes. Figure 2.14 illustrates this concept for a single recombination/crossover on a binary code representation.

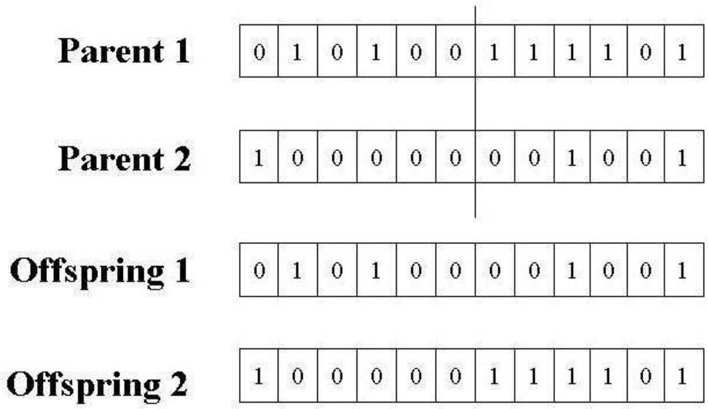

Figure 2.14: Recombination/crossover.

2.6.2.6 Mutation and Adaptation

The importance of mutation is to keep diversity in the population and to expand the search to areas that cannot be represented by the current population. Different mutation operators have been proposed. A common method is uniform mutation, whereby a random number with probability p_c is added to each component of the individuals [59, 107]. In Gaussian mutation, a number from a Gaussian distribution with zero mean is added to each component of the individual vector. Another approach developed by Hansen and Ostermier [68, 66, 69, 125] uses Covariance Matrix Adaption (CMA) and a mutative strategy parameter control (MSC) that is applied to the adaptation of all parameters of a N dimensional normal mutation distribution and provides a second order estimation of the problem topology. Some of their results indicate that in problems that are expected to be significantly non-separable, an EA with CMA is preferable to any EA with only global or individual step size adaptation. Distance-dependent mutation (DDM) is a technique developed to avoid premature convergence and is a technique developed to maintain diversity in small populations [155]. Instead of a fixed mutation rate, each individual has its mutation rate computed after each mating, and mutation depends on the distance between the two parents.

2.6.2.7 Elitism

Another aspect of EAs is an elitist strategy. This is because, as the process of evolution in EAs depends on stochastic operators, there is no guarantee that there would be a monotonic improvement in the fitness function value. With an elitist strategy the best individuals are copied to the next generation without applying any evolutionary operators.

2.6.2.8 Exploration – Exploitation Dilemma

In EAs, a critical aspect is the balance between the exploration for areas of the search space and exploitation of the learned knowledge to progress in the evolution. As these are conflicting objectives, EA researchers have developed different alternatives to balance these trade-offs. Therefore, when developing or selecting an EA it is important to test the algorithm so that it has a good balance between these two criteria and that it has capabilities for benchmarking different mathematical test functions to test its robustness and performance. An area that has shown promising results is using the concepts of sub-populations that explore and exploit different regions of the search space or that get refined as the evolutions progresses [155].

2.6.3 Evolution Strategies

Evolution Strategies are similar to genetic algorithms in that both attempt to find the optimal or near-optimal solution to a problem within a search space (all possible solutions to the problem) without exhaustively testing all solutions. Evolution strategies are based on what is called the principle of “strong causality”, which states that similar causes have similar effects. What this means is that a small change to one encoding of the problem only changes its optimality a little.

The original ES implementation considered an algorithm that worked using only two individuals -one parent and one offspring. In the implementation each individual is real coded and each problem variable is assigned to a floating point value in the chromosome. The variation operator considers a random mutation to each floating point value in the parental chromosome to arrive at the offspring individual. The selection operator is deterministic; in order to determine which individual remains, there is a competition between the parent and offspring. The nomenclature for this strategy is denoted the (1+1) ES, where the first digit indicates the number of parents, the ' + ' indicates the competition between parent and offspring and the final digit indicates the number of offsprings.

Subsequent developments in ESs introduced multi-membered populations, and the first algorithm of this type was the $(\mu + 1)$ ES [11]. This worked by applying a variation operator (recombination, mutation) to μ parents to produce a single offspring. The offspring is selected by determining whether it is better than the worst member of μ, and, if so, it replaces it.

Both the $(1 + 1)$ and $(\mu + 1)$ ES use deterministic control of the mutation size (variations applied to design variables) which is normally distributed when applied to real-coded problems. Following the notation in Whitney [179], the process of evolution strategies can be summarised as illustrated in Algorithm 2. In this algorithm, the letter μ means the total number of parents, λ stands for the number of offspring and g marks the number of generations.

Algorithm 2 Canonical Evolution Strategy

```
Initialise: init(μ^0)
Evaluate:   f(μ^0)
g = 0
while stopping condition not met,
    Recombine: λ_R^{g+1} = reco(μ^g)
    Mutate:    λ_M^{g+1} = mut(λ_R^{g+1})
    Evaluate:  λ^{g+1} = f(λ_M^{g+1})
    Select:    μ^{g+1} = sel(μ ∪ λ) (plus strategy) or,
               μ^{g+1} = sel(λ) (comma strategy)
    g = g + 1
loop
```

EP, ES and GA have distinguishing features according to their evolution modelling and representation, but this classification has been blurred as the features of one method have been incorporated into other methods. For instance, some GA applications and developments have abandoned the bit string for a floating-point representation, ES uses some form of crossover operators for reproduction and EP has been extended and is not only limited to evolution of finite state machines.

As there is no longer clear separation between these methods any more, these systems are regarded as "Evolutionary Computation" or "Evolution Algorithms". In general, an EA (that is including ES, GA and EP), can be defined as indicated in Algorithm 3:

Algorithm 3 Simple Evolutionary Algorithm. Ref: Back [10]

$EA = \ (I, \Phi, \Omega, \Psi, s, \tau, \mu, \lambda)$
Where:
$I = \ A_x \times \ A_s$,is the space of individuals
.....where A_x and A_s, are arbitrary sets
$\Phi: \quad I \rightarrow$
R; denotes a fitness function assigning real numbers to individuals
$\Omega = \left\{w_{\theta_1}, ..., w_{\theta_z} \mid w_{\theta_i} : I^{\lambda} \mapsto I^{\lambda}\right\} \cup \ \left\{w_{\theta_0} : I^{\mu} \mapsto I^{\lambda}\right\}$ is a set of ...
... probabilistic operators w_{θ_i}each controlled by specific...
... parameters summarised in the set $\theta_i \subseteq R$.
λ is a natural number denoting number of offspring individuals.
$s_{\theta_s} : \left(I^{\lambda} \cup \ I^{\lambda+\mu} I^{\lambda} \mapsto I^{\lambda}\right)$ denotes the selection operator where $\lambda, \mu \ \in N$.
$\tau : \quad I^{\mu} \ \mapsto \{true, \ false\}$ is the termination criteria for the EA.
This formalisation covers $(\lambda + \mu)$and (λ, μ) strategies.

2.6.4 Application of EAs to Constrained Problems

Engineering problems usually involve a number of constraints due to technical, manufacturing, human resources requirements and limitations. It is necessary to incorporate those constraints into the design and optimisation process to obtain a realistic design. Evolution Algorithms are unconstrained optimisation procedures, therefore some handling techniques have been introduced to incorporate constraints into the fitness functions.

Different approaches have been developed in order to satisfy design constraints [87]. The use of the penalty function is the most common approach and is based on adding penalties to the objective function [38]. When applying a penalty to an infeasible individual it is important to determine if it is to be penalised for simply being infeasible or penalised also by some amount related to its in-feasibility and the number of constraints violated. As reported by different researchers [59, 148], penalties which are functions of the distance from feasibility perform better than those which are only a function of the number of violated constraints. Joines and Houck [79] describe *static penalties* and *dynamic penalties*. In the first, the user defines several levels of violation and a penalty coefficient is chosen for each so that the penalty coefficient increases as a higher level of violation is reached. The drawback of this approach is that it requires a high number of parameters. In the latter, the dynamic function increases as the optimisation progresses through generations.

Other methods include *annealing penalties* [108, 109] that are similar to simulated annealing in which the penalty coefficients are changed once the algorithm is trapped in a local optimum. The main problem with this approach is that it is sensitive to the values of its parameters and it is difficult to choose an appropriate cooling scheme. *Adapting penalties* proposed by Bean and Hadj Aloune [64] work by modifying the penalty based on a feedback from the last *k* generations. The inconvenience with this approach is the selection of the number of generations to wait before it is applied.

The *Death penalty* is the easiest way to handle constraints and works by rejecting infeasible individuals; the main drawback is the loss of information that can be contained in the individual that is discarded. It can also be lengthy, especially in cases where it is difficult to approach the feasible region. Extensive theory on constrained optimisation methods can be found in Schoenauer and Xanthakis [154], Tsang [169] and Mackworth [93].

2.6.5 Multi-objective Optimisation and EAs

As EAs work via a population-based approach, they are well suited for multi-objective problems. In general, an EA designed for multi-objective optimisation is regarded as MOEA [28, 42]. Different MOEAs have been proposed and include the use of preferences, lexicographic, Pareto and Nash approaches [28, 29, 35, 42, 100, 157]. This thesis will focus on Nash- and Pareto-based approaches, as these have been shown to be promising and robust for evolutionary techniques.

2.6.5.1 Nash EAs

A well-known concept from game theory with promising potential for fast multi-objective or multidisciplinary optimisation is the Nash equilibrium approach [114, 115, 157]. As described by Sefrioui [157], the Nash equilibrium is determined by n players competing symmetrically for n criteria, where each player optimises a unique set of optimisation variables, and all other variables are determined by the other players. When no player can improve on its own criteria, the system has reached a state of equilibrium called the *Nash Equilibrium*.

For example, for player i the vector of problem variables is $X = (x_1,\ \bar{x}_2,\ x_3,\ \bar{x_4}, \bar{x}_5, ...,\ \bar{x_n})$ where all variables $x_i$are free and the reminder are 'locked' by the other players. Player i is interested only in one objective, namely $f_i = f_i(X)$ where $F(X) = (f_1(X), f_2(X), ..., f_n(X))$ is the entire multi-objective problem.

The original Nash implementation uses one EA for each player, as illustrated in Figure 2.15, whereby information is exchanged between the EAs after a generation k has occurred. The equilibrium point or Nash equilibrium point is obtained when no player can improve its set of design variables.

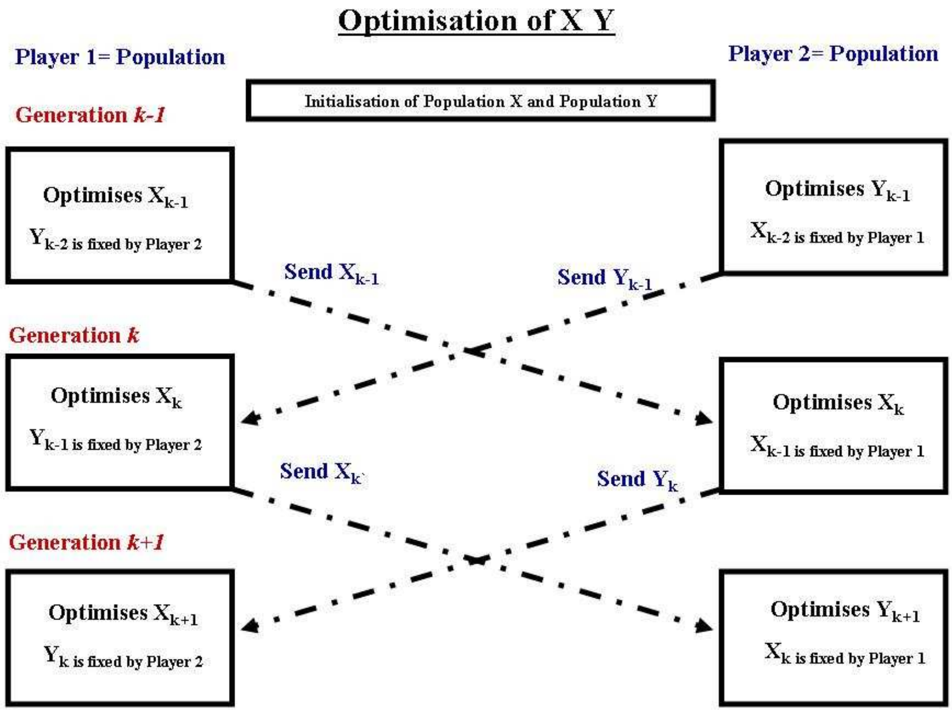

Figure 2.15: Nash Evolutionary Algorithms

2.6.5.2 Pareto-based EAs

As described in Section 2.4, most real-world problems involve conflicting objectives and there is no unique optimum, but a set of compromised individuals. Various methods that use the Pareto optimality approach have been proposed.

The Multiple Objective GA (MOGA) proposed by Fonseca and Flemming [53] describes the use of a scheme where the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Non-dominated individuals are therefore assigned the same rank, while dominated ones are penalised. Non-dominated Sorting Genetic Algorithms (NSGA) use a procedure defined by Goldberg [59]. In NSGA the non-dominated individuals get the same probability of reproduction; to perform this, the algorithm has to find first the Pareto individuals, give them rank one and remove them from the population. The process is repeated with the remaining population members, giving them rank two and so on until the population has been completely ranked. As will be illustrated in Section 2.7.3, this approach has some drawbacks.

Niched Pareto GA [72] uses a tournament selection scheme based on Pareto dominance, in which good performance depends on a sharing factor and tournament selection size. Goldberg

[59] proposed fitness-sharing to maintain diversity in the population and sample uniform Pareto optimal solutions, and uses scaling mechanisms based on the idea that individuals in a particular niche share the available resources.

Another MOEA approach is the Pareto tournament selection approach used in Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA) and described in Section 2.7.

Some problems arise when developing and applying EAs to multi-objective problems. Deb [42], for example, describes and analyses problem features that might prevent a multi-objective genetic algorithm from converging to the true Pareto front, and defines difficult test problems that serve as a guideline to evaluate the multi-objective features of an algorithm. By carrying out an analysis of these functions an algorithm can be tested for multi-modal problems, deceptive problems, problems having convex or non-convex fronts and non-uniformly represented Pareto optimal fronts. The algorithms developed and used in this research will be tested in Chapter 4 for some mathematical test cases and will prove to be robust and efficient to find optimal Pareto optimal fronts for difficult problems.

2.6.6 Distributed and Parallel Evolution Algorithms (pEAs)

One of the main drawbacks of EAs is the CPU time, as they require many function evaluations. For this reason it was realised that it was necessary to incorporate some sort of parallel computing techniques. EAs are particularly adaptable to parallel computing, as candidate individuals or populations are sent to remote machines for evaluation.

The most common approach for parallelisation is global parallel EAs. This consists of a master-slave implementation whereby the master controls the process and sends individuals to solver nodes where their fitness is evaluated by processors (slaves). The master collects the results and applies evolutionary operators to produce the next generation.

A different approach is to divide the global population into several sub-populations [135]. The main idea is the use of small series of interconnected sub-populations instead of a single large population; these sub-populations evolve independently on each node for a time or period called an epoch. After each epoch a period of migration and information is exchanged between nodes and successive new periods of isolation occur. With this approach sub-populations can explore different regions of the search space; by doing this the robustness improves and it is easier to escape from local minima.

Another common approach is to preserve the global population while parallelising the EA operators that are restricted to neighbouring individuals. This is considered an extension to the second approach and is categorised as a cellular EA. Details of these methods can be found in Veldhuizen *et al*. [171] and Cantu-Paz *et al.* [26].

A variant to these approaches is the introduction of asynchronous evaluations proposed by Whitney [179], which will be discussed in the next section. A good summary of research in parallel EAs (pEAs) can be found in Cantu-Paz [24], Dorigo and Maniezzo [45] and Veldhuizen *et al.* [171]. Guidelines for designing them can be found in Cantu-Paz [25, 26].

pEAs have been applied to different engineering problems. Marco *et al.* [98] applied them for multi-objective optimisation in computational fluid dynamics and aerodynamic shape design; in their optimisation process the parallelisation strategy is based on the computational effort needed to solve the Euler flows for each individual. They employed a two-level parallelisation strategy: the first level is parallelisation of the flow solver which combines domain partition techniques with a Message Passing Interface [MPI]. The second level is the parallelisation of GAs and it exploits the methods in MPI to handle process groups. Two groups are defined; each group evaluates the criteria for each individual and each group contains the same number of processes [45, 98].

Other applications of parallel EAs can be found in Jones *et al.* [80] and Sefrioui *et al.* [80, 159]. This recent work on pMOEAs has allowed significant performance and robustness gains in global and parallel optimisation. It is clear that a way forward for generalised design and optimisation in aeronautics is the satisfactory coupling of CFD and evolutionary methods.

2.6.7 Hybrid Algorithms

As discussed in the preceding sections, there are benefits and drawbacks to the use of optimisation techniques and for this reason, several researchers have developed a series of hybrid algorithms. These algorithms are a combination of stochastic-stochastic, stochastic-deterministic or stochastic-approximation techniques [27, 89, 81]. Some of these techniques have been shown to be robust and efficient [22, 27]. Nonetheless there are several issues, such as when to switch from an evolutionary to a deterministic optimiser or, in the case of approximation techniques such as DACE/RSM, the accuracy of these approximations is of special concern.

2.6.8 The Development of Evolutionary Algorithms for Design and Optimisation in Aeronautics

The potential benefits of EAs for optimisation problems in engineering have been recognised for some time. The following sub-sections describe the application of EAs to some aeronautical and engineering areas.

2.6.8.1 Single and Multi-element Aerofoil Design

The most common application of EAs to aerodynamic shape optimisation is to aerofoil design, which includes works by Marco *et al.* [98, 97] and Whitney *et al.* [181, 180, 184], Périaux *et al.* [134] and Obayashi [122]. Optimisation of cascade aerofoils using EAs can be found in Obayashi *et al.* [118, 124, 123]. Some studies on the use of parallel genetic algorithms for aerofoil design and optimisation are presented in Jones *et al.* [80], Sefrioui *et al.* [160], De Falco *et al.* [39] and Makinen *et al.* [95].

These applications give a general overview of the benefits of EAs for aerofoil shape optimisation and give insight into some of the complexities that arise when applying and coupling EAs for multi-objective problems and parallel computations.

There are only a few studies that apply EAs to multi-element high-lift aerofoil design. In [140, 141], Quagliarella, for example, used a viscous solver for high-lift aerofoil design and produced optimal results for single- and multi-objective problems. On a co-authored paper [62], the author of this thesis used a Euler solver for the reconstruction of the target pressure distribution over a three-element aerofoil set. This study illustrated that an optimal combination of design variables for the slat and flap can be obtained, but it is necessary to use full Navier-Stokes equations and a good topology representation that does not vary with the mesh representation to fully account for the changes in the evolution process and the complexities of this type of flow.

2.6.8.2 Intake/Nozzle Design

There are a few applications of EAs for nozzle design and optimisation. Sefrioui and Périaux, [157] for example, used a Nash Genetic algorithm for two objectives and were able to find a compromise, Nash equilibrium solution. In a different study Sefrioui and Périaux used a Hierarchical Genetic Algorithm to find optimal nozzle geometries [156]. Whitney *et al.* [181] compared different EA approaches for inverse nozzle design. Optimisation results for 2-D and 3-D air intakes were studied by Knight [84]; in that study Knight was able to find optimal geometries for single and multiple flight points.

2.6.8.3 Wing Design

Different studies explore the potential benefits of EAs for wing design and optimisation. Obayashi, for example, applied EAs for several wing platform design problems [119, 120, 127, 128]. In these studies, different niching and elitist models are applied to a Multi Objective Genetic Algorithm (MOGA) to find optimal Pareto fronts for transonic and supersonic wing design. The transonic case considered three (3) objectives: minimise aerodynamic drag, minimise aircraft weight and maximise fuel weight stored in the wing. The constraints imposed were lift greater than given aircraft weight and structural strength greater than aerodynamic loads.

Other applications of EAs for wing design include the work by Oyama *et al.* [127, 128], Takahashi *et al.* [167] and Anderson *et al.* [9]. In a co-authored paper [62, 63], the author of this thesis studied and illustrated the application of a hierarchical topology of EAs for multi-objective multidisciplinary wing design.

Important results of these studies indicate the importance of variable fidelity models, the broad applicability and the ability of EAs to find optimal Pareto solutions for three-dimensional applications and problems with more than two objectives.

2.6.8.4 Aircraft Design

On aircraft design Crispin [32] applied GAs for aircraft conceptual and preliminary design and found it useful to obtain reasonable and feasible designs. Crossley *et al.* [33] applied GA for helicopter conceptual design and were able to show the effectiveness of GAs and obtain optimal feasible configurations. One of the results of his work was that the use of parametric variations conducted by GAs can significantly reduce the amount of time and money in the early stages of aircraft design.

Perez [149] conducted some work on GAs for aircraft conceptual design and found a 5% weight savings when compared to a conventional design. Ali and Behdinan [7] applied GAs to determine an optimal combination of design variables for a medium-size transport aircraft. They studied different selection and crossover strategies and indicated that with a GA approach it was possible to generate feasible and efficient conceptual designs. In these studies, the authors also highlighted the effectiveness and importance of EAs in saving money in the initial stages of the design process.

Parmee and Watson [132] proposed a preliminary airframe design using co-evolutionary multi-objective genetic algorithms. Their algorithm was able to find local objective optimal solutions after a few generations and identify paths to trace the trade-off surface to some extent. This research also mentions that on-line sensitivity analysis has a role to play as the number of objectives increases and suggests that quicker, less detailed runs can easily be achieved using smaller population sizes.

Other applications of EAs for aircraft design include the work by Cvetkovick *et al.* [34, 36] and Raymer [142, 143]. Not only traditional transport or commercial aircraft have been studied and optimised using EAs. A few studies on the benefits of EAs in exploring large variations in the design space have been conducted for novel, non-notional configurations such as Unmanned Aerial Vehicles (UAVs/UCAVs) and Micro Aircraft Vehicles [117, 61, 60, 182].

2.6.8.5 Other Engineering Areas

EAs have also been applied to other engineering areas including structural optimisation [49, 130, 129], robot trajectory optimisation [136], radar cross-section minimisation [133] and topology optimisation [90].

Barlow *et al.* [12] used evolutionary algorithms to evolve autonomous controllers for Unmanned Aerial Vehicles, and found optimal solutions for multi-objective problems. Pires *et al.* [136] used a genetic algorithm to generate a robot structure and an optimal trajectory to avoid colliding with obstacles. They found that through using a GA optimiser and the direct kinematic theory, the singularities which will cause problems for traditional optimisation methods are not a problem.

The design and optimisation of small wind turbines using multi-objective evolutionary algorithms have been also investigated by Hampsey [65], Belessis *et al.* [15] and Woods [183]. In their respective work they found optimal aerodynamic shapes for rotor blades using panel methods and differential evolution.

A good review of the application of EAs to different engineering areas can also be found in [139].

All these works provide an indication of the broad applicability and robustness of Evolutionary Algorithms to find optimal solutions and how industry is increasingly applying them to solve complex problems in engineering.

2.6.9 Comparative Studies of EAs and Other Methods for MDO

A considerable amount of research has been devoted to comparing EAs with traditional deterministic techniques [122, 138, 145, 153]. A comprehensive study on the use of adjoint or Genetic algorithms for multi-objective viscous aerofoil optimisation was performed by Pulliam *et al.* [138]. Other studies include the works by Courty *et al.* [30]. An indication of the performance of EAs as compared to traditional methods can be found in Obayashi and Tsukahara [122] and Sasaki *et al.* [153].

These studies provide an indication of the benefits and broad applicability and performance of EAs, but in general, it can be said that the application of a method depends both on the particular problem to solve and the different specific EA parameters of the evolutionary method.

Little research has been conducted on comparing the application of EAs and other methods for MDO. Raymer and Crossley [143, 142] and Raymer [143] applied and compared different MDO methods, Monte Carlo, Random Walk, Simulated Annealing, GAs and orthogonal steepest descent search to enhance aircraft conceptual design and MDO.

In his research Raymer applied these methods to four aircraft concepts: a fighter, a commercial airliner, an asymmetrical light twin and a tactical UAV showing a broad applicability of GA. One of the conclusions of his work indicates that aircraft conceptual design can be improved by proper application of optimisation methods for MDO. He found that the proper selection of a technique can reduce the weight and cost of an aircraft concept by minor changes in the design variables. His results also indicate that the orthogonal steepest descent method provides a slightly better result with the same number of function evaluations, but, as the number of variables is increased, the evolutionary methods seem to be superior. Raymer limited his research to single-objective problems, used only one type of fidelity model for the aircraft analysis and limited his research to seven design variables and the inclusion of any propulsion variables other than engine size via *T/W*.

2.6.10 Advantages and Limitations of Traditional EAs for Aeronautical Problems

Even though there are definite advantages to using EAs [59, 134], they have not seen widespread use in industry or for multidisciplinary design optimisation applications. The main reason for this is the computational expense involved, as they require more function evaluations than gradient-based methods. Therefore, the continuing challenge has been to improve their performance and develop new numerical techniques. It is clear that a possible and realistic avenue is the application of robust and efficient evolutionary methods for MDO. One of the viable alternatives are the Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA) described in the next section. These algorithms have proven to be robust and will be extended in this research for multi-objective and multidisciplinary design optimisation problems in aeronautics.

2.7 Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA)

This thesis implements and extends a robust evolutionary technique: the Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA) approach developed by Whitney [179, 181]. The foundations of this technique lie in traditional evolution strategies (Algorithm 2 and Section 2.6.3) and incorporate the concepts of Covariance Matrix Adaptation (CMA) [67, 68], distance-dependent mutation (DDM) [158], multi-objective optimisation (Sections 2.4 and 2.6.5), a hierarchical topology of EAs, asynchronous evaluation, Pareto tournament selection and parallel computing.

2.7.1 Hierarchical Genetic (Evolutionary) Algorithms (HEAs)

Hierarchical Genetic Algorithms (HGAs), or, in general, Hierarchical EAs (HEAs), are a particular approach developed by Sefrioui and Périaux [156]. This approach uses a hierarchical topology for the layout of the sub-populations; Figure 2.16 illustrates this concept. The bottom layer can be entirely devoted to exploration, the intermediate layer is a compromise between exploitation and exploration and the top layer concentrates on refining solutions.

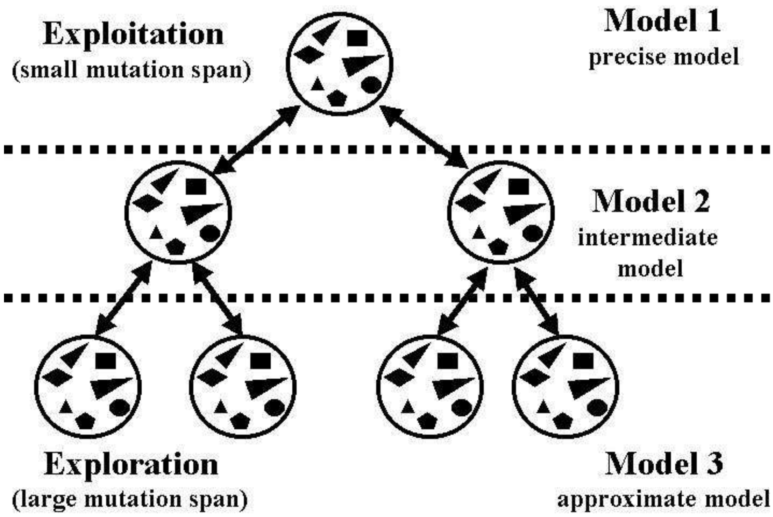

Figure 2.16: Hierarchical topology.

The main feature is the interaction between the given layers. The best solutions progress from the bottom layer to the top layer where they are refined. This circulation of solutions up and down the tree becomes even more interesting if we keep in mind that each node can be handled by a different EA where specific parameters can be tuned.

In other words, the nodes of each layer can have a different purpose, defined by their associated EA:

1. The top layer concentrates on refining solutions. That can be achieved by tuning the EA in a way that takes very small steps between successive crossover and mutation operations.

2. The intermediate layer is a compromise between exploitation and exploration.

3. The bottom layer can be entirely devoted to exploration. That means that the EA can make large leaps in the search space.

All the bottom layer nodes can use a less accurate, fast model to compute the fitness function of the individuals of the sub-population. Even though these solutions may be evaluated rather roughly, the hierarchical topology allows their information content to be used. As these solutions are sent up to the intermediate layer during the migration phase, they are re-evaluated using a more precise model to give a more accurate representation of the actual quality of the solution. However, model two is also a compromise model, as it is deliberately not too precise for the sake of speed. The process is repeated again by sending the solutions up to the top layer during the migration process. These solutions are reevaluated with model one, the most precise model that gives a genuinely accurate value for the fitness function.

In a practical implementation, this model can take the form of :

- **Bottom Level:** a coarse grid or simple algebraic equations and historical trends for the different physics involved.
- **Intermediate Level**: an intermediate grid or more complex models for the physics involved: potential flow solvers or panel methods for aerodynamics, simple FEA modules for structures and blade element theory or helicoidal vortex model for propulsion.
- **Top Level:** a refined grid or more precise models: Navier-Stokes equations for aerodynamics and refined FEA for structural analysis.

Whitney *et al.* [181] tested the performance of a traditional EA, Hierarchical EA and a Hierarchical EA with multiple models, based on the computational expense needed. It was found that when compared to a traditional EA implementation with a single population evolutionary algorithm, the hierarchical approach can speed up an optimisation process by a factor of three.

2.7.2 Asynchronous Evaluation

As discussed in Section 2.6.6, it is common to implement parallelisation strategies with EAs, but the problem is that researchers refer to the use of parallel computing but do not provide good detail on the parallelisation strategy employed. A variant of the classical parallel EAs implementation was proposed by Whitney [181, 180]. In this case the remote solvers do not need to run at the same speed (synchronise) or even on the same local network. Solver nodes can be added or deleted dynamically during the execution. This parallel implementation requires modifications to the canonical Evolution strategy which ordinarily evaluates entire populations simultaneously. The distinctive method of an asynchronous approach is that it generates only one candidate solution at a time and only re-incorporates one individual at a time, rather than an entire population at every generation as is usual with traditional EAs. Consequently ,solutions can be generated and returned *out of order*. This allows the implementation of an asynchronous fitness evaluation, giving the method its name. This is an extension of the work by Wakunda and Zell [174]. With an asynchronous approach there is no waiting time (or bottleneck) for individuals to return. As soon as a solution is available, it is incorporated back into the process.

2.7.3 Pareto Tournament Selection

Another feature of the HAPEA approach is Pareto tournament selection. Whitney [179] proposed an extension of the standard tournament operator which is popular in many approaches

[59]. Simply, to determine whether a new individual x is to be accepted into the main population, it is compared with the selection buffer by assembling a small subset of the buffer, called the tournament $Q = [q, q_2, ...q_n]$. Then the tournament Q is assembled by selecting individuals from the buffer, exclusively at random, until it is full. Next, it is necessary to ensure that the new individual is not dominated by any in the tournament. If this is the case, then it is immediately accepted, and is inserted according to the replacement rules. The only parameter that needs to be determined in advance is the tournament size, a parameter that would exist in a single objective optimisation anyway. Selection of this parameter requires a small amount of problem-specific knowledge, and should vary between between $Q = \frac{1}{2}B$ (strong selective pressure) and $Q = \frac{1}{6}B$ (weak selective pressure).

A schematic representation of the asynchronous approach and Pareto tournament selection approach is illustrated in Figures 2.17 and 2.18. A pseudo-code of the HAPEA approach is illustrated in Algorithm 4.

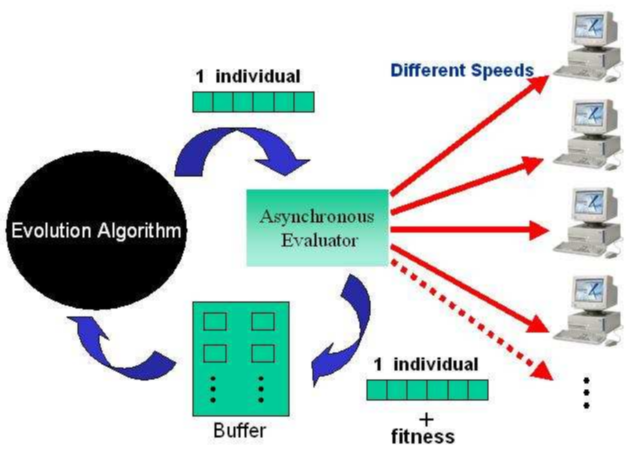

Figure 2.17: Asynchronous evaluation.

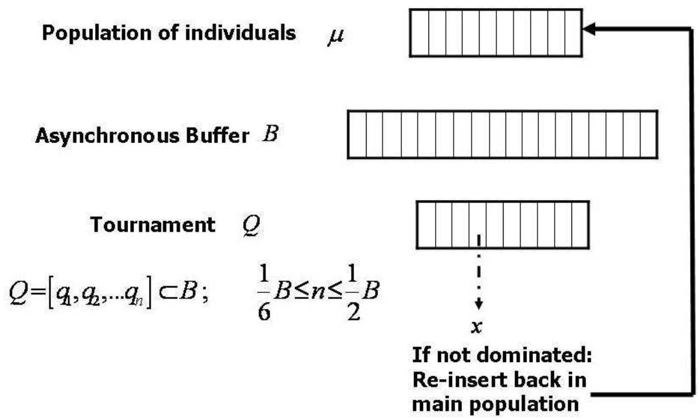

Figure 2.18: Pareto tournament.

Algorithm 4 Basic HAPEA.

Define design variables x, z: $Define(x,\ z)$
Define number of subpopulations (nodes) i, hierarchical levels $Define(i)$
for all levels initialise subpopulations: $init(\mu_0^0, \mu_1^0 ... \mu_n^0) : a_k$
  Layer 1: Uses Type 1 analysis: $init(\mu_0^0) : a_1$
  Layer 2: Uses Type 2 analysis: $init(\mu_1^0, \mu_2^0) : a_2$
  Layer 3: Uses Type 3 analysis: $init(\mu_3^0, \mu_4^0, \mu_5^0, \mu_6^0) : a_3$
loop
$g = 0$
while stopping condition not met,
   Recombine: $\lambda_R^{g+1} = reco(\mu^g)$
   Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$
   Evaluate candidate using specific integrated analysis...
   ...type: $\lambda^{g+1} = f(\lambda_M^{g+1}) : a_k$
   Get output analysis a_i, parameters p, check constraints...
   ...and add penalty: $\lambda^{g+1} = f(\lambda_M^{g+1}) + penalty$
   Select: $\mu^{g+1} = sel(\mu \cup \lambda)$ (plus strategy) or,
     $\mu^{g+1} = sel(\lambda)$ (comma strategy)
   if Multi-objective:
     Calculate Pareto fronts: $Pareto\ \ Front = Pareto(\lambda_M^{g+1})$
  $g = g + 1$
  if epoch completed:
    Start migration: $(\mu_i^{g_k}) = mig(\mu_i^{g_k} \to \mu_{i\pm1}^{g_k}) : a_k$
    Layer 1: Receive best solutions from layer 2 and reevaluate using type 1 ....
    ...integrated analysis: $(\mu_1^{g_k}, \mu_2^{g_k} \to \mu_0^{g_k}), \mu_0^{g_k} = f(\mu_0^{g_k}) : a_1$
    Layer 2: Receive random solutions from layer 1 and best from layer 3 ...
    ...and reevaluate them using type 2 integrated..
    ...analysis: $\mu_0^{g_k} \to \mu_{1,2}^{g_k}, \mu_{3,4,5,6}^{g_k} \to \mu_{1,2}^{g_k}, \mu_{1,2}^{g_k} = f(\mu_{1,2}^{g_k})$: a_2
    Layer 3: Receive random solutions from layer 2 and reevaluate them...
    ...using type 3 integrated analysis: $\mu_{1,2}^{g_k} \to \mu_{3,4,5,6}^{g_k} \to \mu_{3,4,5,6}^{g_k} = f(\mu_{1,2}^{g_k})$:
a_3
loop

2.7.4 Extensions to HAPEAs

The HAPEA algorithms have been shown to be robust and efficient for aeronautical problems, but a careful evaluation of and developments were implemented in this research work to account for difficulties of several multi-criteria and MDO problems. These extensions will be described in detail in the next chapter and include different methods and algorithms for multidisciplinary and multi-objective optimisation formulations, the introduction of a Nash equilibrium approach, a comprehensive test bench of numerical methods to test mathematical functions and several methods and algorithms for analysis, optimisation and post-processing.

2.8 Summary

This chapter described the basic definitions adopted throughout this thesis. Formal definitions of the concepts of optimisation, multi-objective and MDO were outlined. A short review of different approaches and industrial needs for MDO was presented. Details of evolutionary algorithms and their specific applications to aeronautical design problems were discussed. The chapter then provided specific details on a particular EA used in this research.

Concluding this chapter, it is noted that there are different methods, architectures and applications of optimisation and multidisciplinary design optimisation methods for aeronautical problems. However, still further research and proposals for alternative methods are still required to address the industrial and academic challenges and needs of this industry. EAs can be an alternative option to satisfy some of these needs, as they are easily coupled, particularly adaptable, easily parallelised, require no gradient of the objective function(s), have been used for multi-objective optimisation and successfully applied to different aeronautical design problems. Nonetheless, EAs have seen little application at an industrial level due to the computational expense involved in this process and the fact that they require a larger number of function evaluations, compared to traditional deterministic techniques.

Therefore, the following chapters will concentrate on the development and applications of robust methods and a framework that addresses some of the complexities of MDO and the limitations of current optimisation approaches in order to facilitate the analysis, design and optimisation of multidisciplinary and multi-objective problems in aeronautics.

Chapter 3

Design and Implementation

> "*Nature uses only the longest threads to weave her patterns, so that each small piece of her fabric reveals the organization of the entire tapestry.*" Richard P. Feynman.

3.1 Introduction

The objective of this chapter is to describe and define the requirements, formulate and implement robust evolutionary methods and provide a framework for multi-objective, multi-disciplinary design and optimisation in aeronautics. As illustrated in the previous chapter, the foundation and application of optimisation and MDO methods for aeronautical design problems have been an active field of research. During the execution of this literature review limited information was found on the robust application of evolutionary techniques for MDO problems. This is due to the fact that these problems are complex, time-consuming and may involve non-linearities, multi-objective and multidisciplinary considerations. In order to handle these complexities it is desirable to develop a system and methods which facilitate integration of several components for analysis, optimisation, parallel computing and post-processing. The fundamental idea is to develop robust algorithms and methods that simplify the task of integration so the user can focus on the problem itself. The methods and algorithms are integrated in a framework. This framework is developed in a generic way so that it can be easily implemented, maintained and extended.

The chapter is organised in the following manner: Section 3.2 describes the steps followed for the creation and validation of the methods and framework. Section 3.3 indicates and summarises the basic requirements of problem formulation, problem execution, architectural design and information access. Section 3.4 focuses on the general design of the framework, Section 3.5 details how the different algorithms are created and implemented, Section 3.6 provides an initial qualitative assessment of the methods and algorithms. Finally, Section 5.5 provides a summary of the chapter.

3.2 Development

Figure 3.1 illustrates the approach considered for the development of the methods and framework.

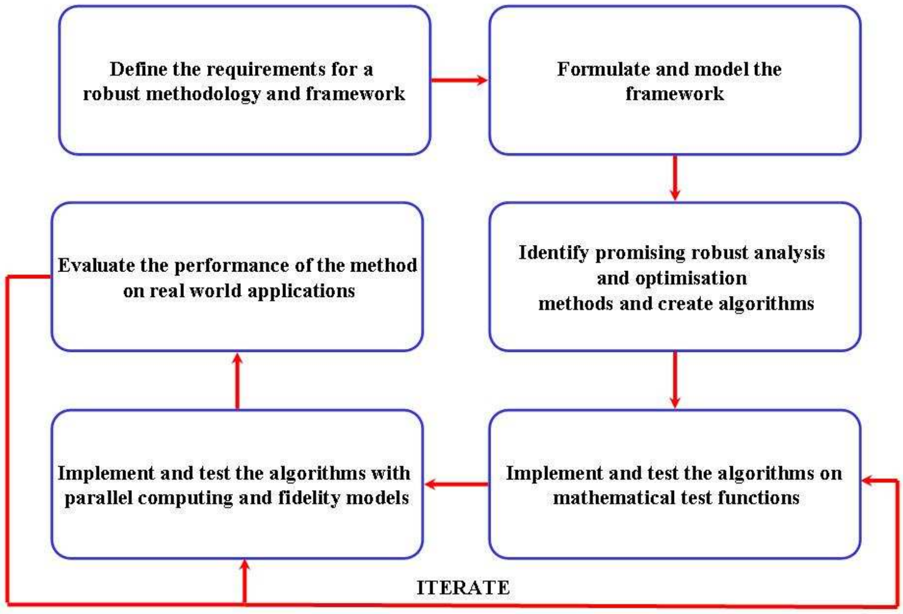

Figure 3.1: Methodology.

A first step for the development of the methods and framework was to define the main requirements. These include defining capabilities for different problem formulations, selection of robust design and optimisation methods, requiring a robust software architecture, providing a user-friendly interface and identifying pre- and post-processing tools [151, 152].

A second step was to formulate and model the methods in such a way that it is generic and applicable to different types of complex problems; highly non-linear, multi-modal, involving approximations, being non-differentiable or involving multiple objectives and physics, with convex, non-convex or discontinuous Pareto optimal fronts.

Following this, promising robust analysis tools were identified and optimisation algorithms were created. For the selection of analysis methods it was important to consider the type of fidelity of the solver and its computational expense. A significant part of this work was devoted to identifying, compiling and evaluating a series of analysis tools for aerodynamic, structural and aircraft design. The compilation and validation of these codes was crucial to enable the development of algorithms for coupling with the optimisation routines.

The appropriate selection of a robust optimisation method was also of great concern. The Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA) technique [179] was tested, extended and implemented in this work. This technique was designed to be robust and efficient. However, due to the the complexity of the problems to be solved, an evaluation and extension of the method was required.

Therefore, the next step was to create different algorithms to test, extend and apply these techniques to multi-objective and multidisciplinary design optimisation problems. The performance of the algorithms with different parameter settings was investigated, giving the designer insight and an indication of the parameter settings of the algorithms for real-world problems that may exhibit similar behavioural characteristics. The relevance of this step cannot be overstated, as many algorithms fail to be robust on different test problems or fail to find the real answer.

The next step was to implement and test the algorithms with parallel computing strategies and hierarchical topologies with different solvers. This provided an indication of the performance of the algorithm with increasing numbers of computers. It also enhanced the understanding of a good balance and combination of fidelity solvers or grid resolutions of the analysis tools.

Finally, the methods are tested for real-world test cases. Most of the applications in this thesis are related to aerodynamic and aero-structural optimisation of Unmanned Aerial Vehicles systems. But they can also be applied to manned vehicles as well. The development and use of these vehicles for military and civilian applications is rapidly increasing. Similar to their manned counterparts, the challenge is to develop optimal configurations that produce a high-performance aircraft which satisfies the mission requirements. Therefore these vehicles provide a good application environment for testing the methods as outlined.

The iteration process is present during the development of the methods as the results obtained from the mathematical test cases and practical implementations serve to feed back into, and refine the algorithms and methods.

3.3 Definition of Requirements

Several requirements need to be satisfied for methods and a framework to be robust on their application to MDO and multi-objective problems in aeronautics. The basic requirements are problem formulation and optimisation tools, problem execution, architectural design and information access [17, 21, 151, 152, 178].

3.3.1 Architectural Design

The methods and framework should:

- be developed using object-oriented principles.
- provide an easy-to-use and intuitive GUI.
- be easily extensible by developing new interfaces required to integrate new processes and numerical methods into the system.
- not impose unreasonable overhead on the optimisation process.
- handle large problem sizes.
- be based on standards.

3.3.2 Optimisation Methods

The methods and framework should allow:

- ease of integration of robust optimisation methods.
- coupling of different disciplinary analyses with different optimisation methods and provide schemes for sub-optimisations within each design module.
- incorporate legacy codes, which can be written in different programming languages and proprietary software where no source code is available.

3.3.3 Problem Formulation and Execution

The methods and framework should:

- allow the user to configure and reconfigure different multi-criteria and MDO formulations easily without low-level programming.
- allow the execution and movement of data in an automated fashion.
- be able to execute multiple processes in parallel and through heterogeneous computers.
- execute different optimisation runs (desirable).

3.3.4 Information Access and Post-processing

The methods and framework should:

- provide facilities for database management.
- provide capabilities to visualise intermediate and final results from the analysis or optimisation.
- allow capabilities for monitoring and viewing the status of an execution and its system status.
- provide a mechanism for fault tolerance.

3.4 Design and Formulation

With these requirements in mind, the general scope for the methods and the framework was identified. Figure 3.2 shows a representation of different components to satisfy these requirements.

The framework addresses and integrates all of these requirements to some extent. It will have five major constituents: A Graphical User Interface (GUI), a general problem formulations capability embedded within each analysis module, a robust optimisation tool, some analysis modules, and capabilities for parallel computing, Design of Experiments (DOE) and post-processing. Each of these will be described in more detail in the subsequent sections, briefly:

GUI: The GUI is the main interface for interaction of the user and the framework.

General MO-MDO Problem Formulation and Execution using EAs: This provides ease of implementation of different MDO architectures and multi-objective methods.

Analysis Modules: This contains the different methods, and design module interfaces for different applications in aeronautics. So far the system has implementations for aerofoil, nozzle, wing and full aircraft configuration design.

Optimisation Module: The optimisation module manages the execution and definition of different optimisation codes. So far the implementation includes the Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA) described in the previous chapter, with some extensions, as will be described in Sections 3.5.2 and 3.5.2.3.

Parallel Computing: This module provides capabilities for dynamically adding computers and conduct studies with an increasing number of computers.

Design of Experiments Module: One of the drawbacks of EAs is that they suffer from slow convergence. By providing a DOE capability into the framework we hybridise the desirable characteristics of EAs and surrogate models, such as RSM or DACE, and obtain an efficient optimisation system.

Post-processing: Post-processing capabilities to visualise the progress of the optimisation and final results.

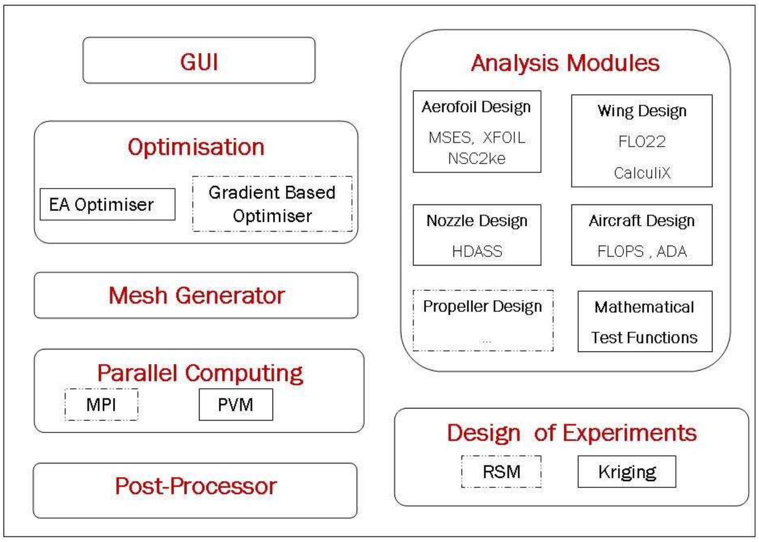

Figure 3.2: MDO Framework.

3.5 Implementation

Integrating these components is a complex task. This research considers the development of the methods, architecture, definition of a robust optimisation tool, implementation of a general formulation for the MDO problem, generating a simple but intuitive GUI, integration and application studies of different analysis modules for design and optimisation in aeronautics. The DOE capability has been accounted for, but only applications to mathematical test problems and sampling techniques have been tested.

3.5.1 Robust Optimisation Tools – Hierarchical Topology of Asynchronous Evolutionary Algorithms for MDO

Following Section 3.3, the first consideration is the incorporation of robust optimisation tools. In this research the Hierarchical Asynchronous Parallel Evolutionary Algorithms are implemented and extended. As will be detailed in Sections 3.5.2 and 3.5.3, several algorithms were created to extend and integrate this method for mathematical test functions, multi-objective and multidisciplinary design optimisation MDO formulations.

3.5.2 General MO–MDO Problem Formulation and Execution Using EAs

A second consideration is how to incorporate different MDO and multi-objective formulations and legacy codes. As indicated in Chapter 2, there are many strategies proposed for multi-objective formulations and MDO and the development of these optimisation methods, architectures and decomposition methodologies has been an active field of research. The selection of the appropriate optimisation architecture is of great importance for an efficient solution of MDO problems, as one MDO architecture may find a feasible solution without high computational expense, while another might be too slow or even fail [23]. The following sub-sections formulate, extend and develop algorithms that integrate the HAPEAs for different MDO formulations and the Nash equilibrium approach.

3.5.2.1 Integrated Multidisciplinary Feasible (MDF) Formulation Using Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA)

In an integrated MDF analysis, the set of design variables is evaluated by solving the systems of equations, guaranteeing interdisciplinary constraints and returning the objective functions to be manipulated by the optimiser. When integrated with a hierarchical evolutionary algorithm, this analysis takes the form as illustrated in Figure 3.3 and Algorithm 5.

The system uses a hierarchical approach with three levels: on the bottom level a coarse type analysis to direct the exploration, at the top level a more precise model that better describes the physics involved and at an intermediate level, a compromised balance between top and bottom layers is used.

Initially the system will specify the design variables, then it will generate a random sub-population of individuals at each layer, then define the number of subpopulations (nodes) and number of hierarchical levels which, for simplicity, are equal to the number of analyses. Once these initial populations are generated, the algorithm will go through an isolation phase in which evolution occurs. During this evolution phase individuals go through the process of recombination, mutation and evaluation by the type of integrated analysis at the level to which they belong. The optimiser will take output analysis to guarantee satisfaction of constraints and compute the overall fitness function. If the problem is multi-objective, the algorithm will find the non-dominated individuals and will calculate the Pareto fronts.

On a hierarchical topology with three levels, when the epoch is finished or the migration criteria are satisfied, the migration phase occurs: Layer 1 receives best solutions from Layer 2 and re-evaluates them using analysis type one, Layer 2 receives random solutions from Layer 1 and

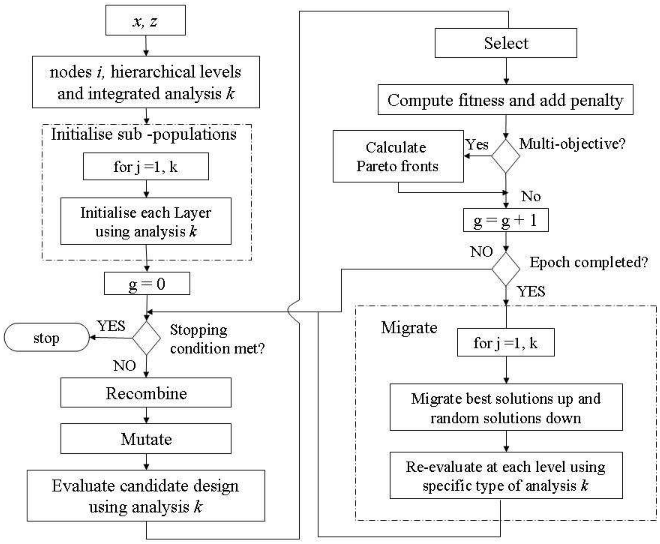

Figure 3.3: Integrated multi-objective and Multidisciplinary Feasible (MDF) MDO formulation flow diagram using HAPEAs.

receives best solutions from Layer 3 and re-evaluates them using analysis type two, Layer 3 receives random solutions from Layer 2 and re-evaluates them using analysis type three. Practical implementations of this approach are presented in Chapters 5, 6, 7 and 8.

Algorithm 5 Integrated multi-objective and Multidisciplinary Feasible (MDF) MDO algorithm using HAPEAs.

```
Define design variables x,z: Define(x, z)
Define number of sub-populations (nodes) i, hierarchical levels...
...and integrated analysis k: Define(i,k)
for all levels initialise subpopulations:init(μ_0^0, μ_1^0...μ_n^0) : a_k
   Layer 1: Uses Type 1 integrated analysis: init(μ_0^0) : a_1
   Layer 2: Uses Type 2 integrated analysis: init(μ_1^0, μ_2^0) : a_2
   Layer 3: Uses Type 3 integrated analysis: init(μ_3^0, μ_4^0, μ_5^0, μ_6^0) : a_3
loop
g = 0
while stopping condition not met,
    Recombine: λ_R^{g+1} = reco(μ^g)
    Mutate:    λ_M^{g+1} = mut(λ_R^{g+1})
    Evaluate candidate using specific integrated analysis type:
    ...λ^{g+1} = f(λ_M^{g+1}) : a_k
    Get output analysis a_i, parameters p, check constraints and...
    ...add penalty: λ^{g+1} = f(λ_M^{g+1}) + penalty
    Select:    μ^{g+1} = sel(μ ∪ λ) (plus strategy) or,
               μ^{g+1} = sel(λ) (comma strategy)
    if Multi-objective:
         Calculate Pareto fronts:Pareto Front = Pareto(λ_M^{g+1})
    g = g + 1
    if epoch completed:
       Start migration:(μ_i^{g_k}) = mig(μ_i^{g_k} → μ_{i±1}^{g_k}) : a_i
       Layer 1: Receive best solutions from layer 2 reevaluate using type 1 ....
       ...integrated analysis: (μ_1^{g_k}, μ_2^{g_k} → μ_0^{g_k}),μ_0^{g_k} = f(μ_0^{g_k}) : a_1
       Layer 2: Receive random solutions from layer 1 and best from layer 3 ...
          ...reevaluate them using type 2 integrated...
       ...analysis: μ_0^{g_k} → μ_{1,2}^{g_k}, μ_{3,4,5,6}^{g_k} → μ_{1,2}^{g_k}, μ_{1,2}^{g_k} = f(μ_{1,2}^{g_k}) : a_2
       Layer 3: Receive random solutions from layer 2 reevaluates them using...
       ...type 3 integrated analysis: μ_{1,2}^{g_k} → μ_{3,4,5,6}^{g_k} → μ_{3,4,5,6}^{g_k} = f(μ_{1,2}^{g_k}) : a_3
loop
```

3.5.2.2 Distributed Individual Discipline Feasible (IDF) Formulation Using Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA)

In an IDF formulation, each discipline takes as its inputs an individual, composed by a set of design variables and produces a set of analysis outputs. The system-level design variables set includes local and shared variables between the disciplines. One assumption is that the discipline-specific analyses are independently solvable; given the system variables each discipline analysis is capable of separating the local and shared variables and computing the solution output. The output of the analysis includes satisfaction of local constraints and data that is passed to the other discipline as parameters.

If we consider an aero-structural wing design example, the system variables for the geometry of the wing are supplied by the optimiser, the disciplinary analysis for aerodynamics computes the flow and produces an output solution. The input parameters from structures to aerodynamics include wing geometry data, while input parameters from aerodynamics to structures include aerodynamics loads. Similarly for structures, the system variables for the geometry of the wing are supplied by the optimiser and the structural analysis produces an output which includes structural response.

When integrated with a hierarchical evolution algorithm, this analysis takes the form illustrated in Figure 3.4 and Algorithm 6.

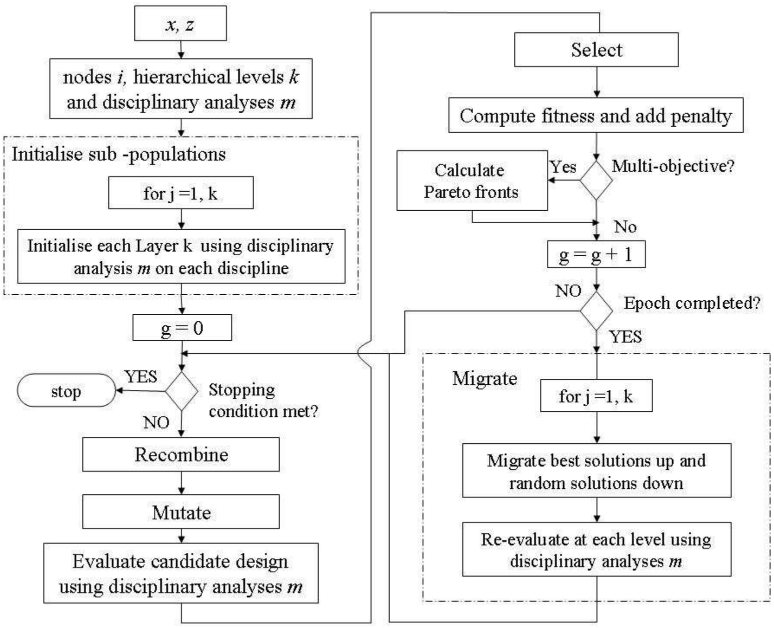

Figure 3.4: Distributed multi-objective Individual Discipline Feasible (IDF) MDO formulation flow diagram using HAPEAs.

Algorithm 6 Distributed multi-objective and Multidisciplinary Feasible (IDF) MDO algorithm using HAPEAs.

Define design variables x, z: $Define(x, z)$

Define number of sub-populations (nodes) i, hierarchical levels
...and disciplinary analysis$k$: $Define(i, k,$ w)
for all levels initialise sub-populations: $init(\mu_0^0, \mu_1^0 ... \mu_n^0) : a_k$
  Layer 1: Uses Type 1 analysis for each discipline: $init(\mu_0^0) : a_1$
  Layer 2: Uses Type 2 analysis for each discipline: $init(\mu_1^0, \mu_2^0) : a_2$
  Layer 3: Uses Type 3 analysis for each discipline: $init(\mu_3^0, \mu_4^0, \mu_5^0, \mu_6^0) : a_3$
loop
$g = 0$
while stopping condition not met,
  Recombine: $\lambda_R^{g+1} = reco(\mu^g)$
  Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$

  Evaluate and optimise using specific analysis type:
  ... $\lambda^{g+1} = f(\lambda_M^{g+1}), a_k$
  Get output analysis a_i, parameters p, check constraints and
  .... add penalty: $\lambda^{g+1} = f(\lambda_M^{g+1}) + penalty$
  Select: $\mu^{g+1} = sel(\mu \cup \lambda)$ (plus strategy) or,
  $\mu^{g+1} = sel(\lambda)$ (comma strategy)
  if Multi-objective:
    Calculate Pareto fronts: $Pareto\ Front = Pareto(\lambda_M^{g+1})$
  $g = g + 1$
  if epoch completed:
    Start migration: $(\mu_i^{g_k}) = mig(\mu_i^{g_k} \rightarrow \mu_{i\pm1}^{g_k} : a_k$

    Layer 1: Receive best solutions from layer 2 reevaluate using type 1...

    ...analysis for each discipline:...
    $...(\mu_0^{g_k} \rightarrow \mu_2^{g_k}, \mu_3^{g_k},), (\mu_2^{g_k}, \mu_3^{g_k} \rightarrow \mu_0^{g_k}), \mu_0^{g_k} = f(\mu_0^{g_k}) : a_1$

    Layer 2: Receive random solutions from layer 1 and best from layer 3 ...

    ...reevaluate them using type 2 analysis for each discipline:...
    $\mu_0^{g_k} \rightarrow \mu_{1,2}^{g_k}, \mu_{3,4,5,6}^{g_k} \rightarrow \mu_{1,2}^{g_k}, \mu_{1,2}^{g_k} = f(\mu_{1,2}^{g_k}) : a_2$
    Layer 3: Receive random solutions from layer 2 reevaluates them using...

    ...type 3 analysis for each discipline: ...
    $...\mu_{1,2}^{g_k} \rightarrow \mu_{3,4,5,6}^{g_k} \rightarrow \mu_{3,4,5,6}^{g_k} = f(\mu_{1,2}^{g_k}) : a_3$
loop

In this case the system uses a hierarchical approach with three levels. Initially the system will specify the design variables, then it will generate a random sub-population of individuals at each layer that are evaluated with an analysis type unique to each discipline and to this level. The process resumes in an isolation and migration phase in the same manner as described in the previous section, but performing the corresponding type of disciplinary analysis at each level.

Subspace Optimisation:

An alternative option to the previous approach is the inclusion of a subspace optimisation using evolutionary algorithms 2.5.4.3. When integrated with a hierarchical evolution algorithm, this analysis takes the form illustrated in Figure 3.5 and Algorithm 7.

The major difference is that in this case individuals are optimised within each node by a deterministic or an EA optimiser. The approach considered was to define the lower and upper bounds in these sub-optimisations to be ± 10% of the value of the current design variable. Once individuals are optimised in these subspaces, they are returned to the system-level optimiser where the interdisciplinary constraints are checked, individuals are manipulated to be refined and reproduced and the overall fitness function is evaluated. Practical implementations of distributed approaches will be illustrated in Chapter 8.

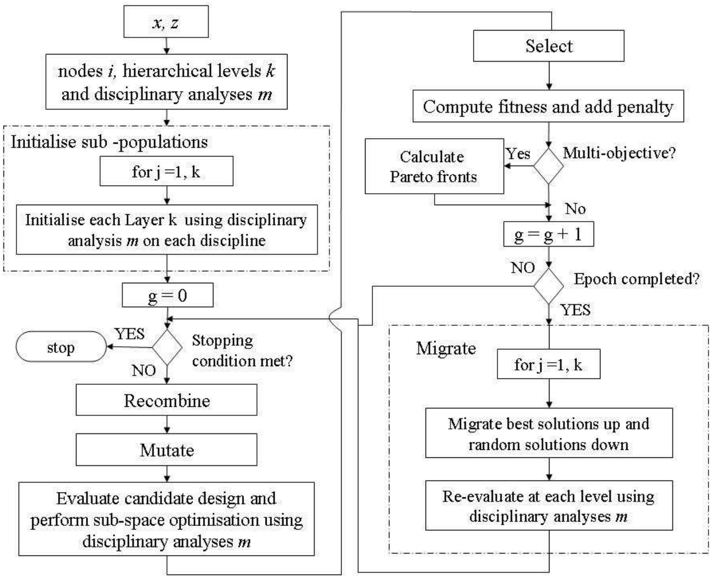

Figure 3.5: Distributed multi-objective and Multidisciplinary Feasible (IDF) MDO formulation flow diagram with subspace EA optimisation.

3.5.2.3 Nash Implementation

In this case, a modified version of the approach developed by Sefrioui and Périaux [157] is implemented. When coupled with HAPEAs, a Nash equilibrium approach takes the form represented in Figure 3.6 and Algorithm 8. Let us consider $s = XY$ to be a string which represents the potential solution for a two-objective problem, where X corresponds to the first objective and Y to the second one. The idea is to assign the optimisation task of X to a player; $Player$ 1, and the optimisation task of Y to a $Player$ 2. Then, following the Nash Equilibrium principle, each player optimises s for its own criteria. $Player$ 1 optimises for the first objective by modifying the design variables X, while design variables Y are fixed by $Player$ 2. In the same manner $Player$ 2 optimises for the second objective by modifying the design variables Y, while X is fixed by $Player$ 2. In a hierarchical implementation, the optimisation task of each player is performed by a hierarchical topology of EAs. The initialisation and generation of individuals proceeds in a similar fashion as in the previous algorithms, through recombination, mutation, evaluation and selection. Practical implementations of the Nash approach will be illustrated in Chapter 8.

Algorithm 7 Distributed multi-objective and Multidisciplinary Feasible (IDF) MDO algorithm with subspace EA optimisation.

Define design variables x, z: $Define(x, z)$

Define number of subpopulations (nodes) i, hierarchical level and disciplinary...
...analysis k and corresponding subspace optimiser w: $Define(i, k, w)$
for all levels initialise subpopulations: $init(\mu_0^0, \mu_1^0 ... \mu_n^0) : a_k$
  Layer 1: Uses Type 1 analysis for each discipline: $init(\mu_0^0) : a_1$
  Layer 2: Uses Type 2 analysis for each discipline: $init(\mu_1^0, \mu_2^0) : a_2$
  Layer 3: Uses Type 3 analysis for each discipline: $init(\mu_3^0, \mu_4^0, \mu_5^0, \mu_6^0) : a_3$
loop
$g = 0$
while stopping condition not met,
  Recombine: $\lambda_R^{g+1} = reco(\mu^g)$
  Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$

  Evaluate and optimise using specific analysis and subspace ....
  ...optimiser type: $\lambda^{g+1} = f(\lambda_M^{g+1}), opt(\lambda_M^{g+1}), a_k, Optimiser_w$

  Get output analysis a_i, parameters p, check constraints and...
  ... add penalty: $\lambda^{g+1} = f(\lambda_M^{g+1}) + penalty$

  Select: $\mu^{g+1} = sel(\mu \cup \lambda)$ (plus strategy) or,
  $\mu^{g+1} = sel(\lambda)$ (comma strategy)

  if Multi-objective
    Calculate Pareto fronts: $Pareto\ Front = Pareto(\lambda_M^{g+1})$
  $g = g + 1$
  if epoch completed:
    Start migration: $(\mu_i^{gk}) = mig(\mu_i^{gk} \rightarrow \mu_{i\pm1}^{gk} : a_k$

    Layer 1: Receive best solutions from layer 2 reevaluate using type 1 ....

    ...analysis for each discipline:...
    $...(\mu_0^{gk} \rightarrow \mu_2^{gk}, \mu_3^{gk},), (\mu_2^{gk}, \mu_3^{gk} \rightarrow \mu_0^{gk}), \mu_0^{gk} = f(\mu_0^{gk}) : a_1$

    Layer 2: Receive random solutions from layer 1 and best from layer 3 ...

    ...reevaluate them using type 2 analysis for each ...

    ...discipline: $\mu_0^{gk} \rightarrow \mu_{1,2}^{gk}, \mu_{3,4,5,6}^{gk} \rightarrow \mu_{1,2}^{gk}, \mu_{1,2}^{gk} = f(\mu_{1,2}^{gk}) : a_2$

    Layer 3: Receive random solutions from layer 2 reevaluates them using...

    ...type 3 analysis for each discipline:...
    ... $\mu_{1,2}^{gk} \rightarrow \mu_{3,4,5,6}^{gk} \rightarrow \mu_{3,4,5,6}^{gk} = f(\mu_{1,2}^{gk}) : a_3$
loop

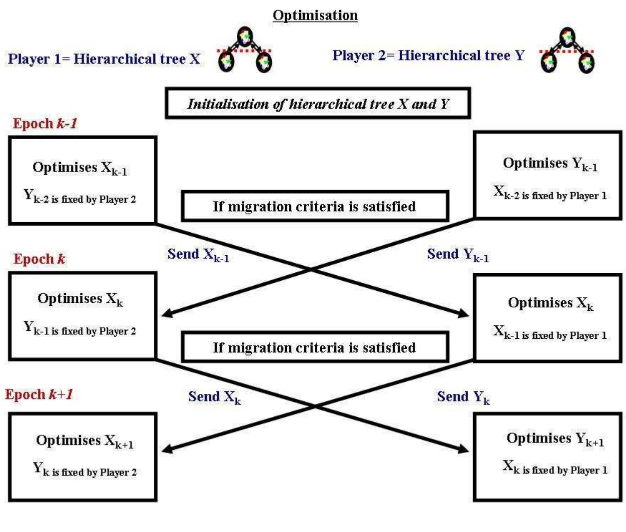

Figure 3.6: Nash HAPEA.

Algorithm 8 Nash algorithm.

Define design variables $x,\ z$: $Define(x,\ z,)$
Define number of objectives: n : $F(X) = (f_1(X), f_2(X), ..., f_n(X))$
Define a Player for each objective: $Player_n \to f_i = f_i(X)$

Define design variables for Player n:...

... $Player_n(X) : X \ = (x_1,\ \bar{x}_2,\ x_3,\ \bar{x_4},\ \bar{x_5}, ...,\ \bar{x_n})$....

...all variables $x_i$are free and the remainder are 'locked' by....
....the other players.

Define number of hierarchical tress h equal to number of players: $h = n$

Define number of subpopulations (nodes) i, hierarchical...

...levels and integrated analysis k on each hierarchical...

...tree:$H_i(Player) : Define(i, k)$

for all levels on each tree initialise subpopulations:$init(\mu_1^0, \mu_2^0...\mu_n^0) : a_k$
Layer 1: Uses Type 1 integrated analysis: $init(\mu_1^0) : a_1$
Layer 2: Uses Type 2 integrated analysis: $init(\mu_2^0, \mu_3^0) : a_2$

Layer 3: Uses Type 3 integrated analysis: $init(\mu_4^0, \mu_5^0, \mu_6^0, \mu_7^0) : a_3$

loop

$g = 0$
while stopping condition not met,

Recombine: $\lambda_R^{g+1} = reco(\mu^g)$
Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$

Evaluate candidate using specific integrated analysis..
...type: $\lambda^{g+1} = f(\lambda_M^{g+1}) : a_k$
Check constraints and add penalty: $\lambda^{g+1} = f(\lambda_M^{g+1}) + penalty$
Select: $\mu^{g+1} = sel(\mu \cup \lambda)$ (plus strategy) or,
$\mu^{g+1} = sel(\lambda)$ (comma strategy)
if Multi-objective:
Calculate Pareto fronts:$Pareto\ \ Front = Pareto(\lambda_M^{g+1})$

$g = g + 1$

if epoch completed:
Start migration: $(\mu_i^{g_k}) = mig(\mu_i^{g_k} \to \mu_{i\pm1}^{g_k} : a_k$

Layer 1: Receive best solutions from layer 2 reevaluate using type 1 ....
...integrated analysis: $(\mu_2^{g_k}, \mu_3^{g_k} \to \mu_0^{g_k}), \mu_0^{g_k} = f(\mu_0^{g_k}) : a_1$

Layer 2: Receive random solutions from layer 1 and best from layer 3 ...

...reevaluate them using type 2 integrated...
...analysis: $\mu_0^{g_k} \to \mu_{1,2}^{g_k}, \mu_{3,4,5,6}^{g_k} \to \mu_{1,2}^{g_k}, \mu_{1,2}^{g_k} = f(\mu_{1,2}^{g_k}) : a_2$
Layer 3: Receive random solutions from layer 2 reevaluates them using...

...type 3 integrated analysis: $\mu_{1,2}^{g_k} \to \mu_{3,4,5,6}^{g_k} \to \mu_{3,4,5,6}^{g_k} = f(\mu_{1,2}^{g_k}) : a_3$

Start Nash migration: $(\mu_i^{g_k}) = mig(\mu_i^{g_k} \to \mu_{i\pm1}^{g_k} : a_k$

Migrate/Update best variables on top node of hierarchical tree h, ...

...to the top node of the other hierarchical trees

Check improvement in design variables for each player:

if $X^{g+1} = X^g$, equilibrium point has been reached
break

loop

3.5.2.4 Implementation of Different Legacy Codes and Hierarchical Topology of Fidelity Models

The requirement of implementing different legacy codes is satisfied within the current methods and framework. So far the implementation has been coupled with legacy codes in different programming languages *C*, *C++*, *Fortran 90*, *Fortran 77*. It has been successfully coupled with representative analysis codes for design in aeronautics: *FLO22* [76], *FLOPS* [105], *ADS*, *VLMpc* [88, 99], *MSES* [46], *XFOIL* [47] and *CalculiX* [44].

As discussed in the previous chapter, one of the advantages of EAs is that they require no derivatives of the objective function. This is of particular importance in the case of proprietary analysis tools where no source code is available or the computation of gradients is prohibitive. The coupling of the algorithms with different analysis codes is performed by simple function calls and input and output data files; there is no need to compute derivatives.

The integration of a hierarchical topology of fidelity models with an MDO approach is illustrated in Figure 3.7. The only difference with the traditional hierarchical topology described previously is that each member of a population on a hierarchical level is analysed asynchronously by a multidisciplinary analysis.

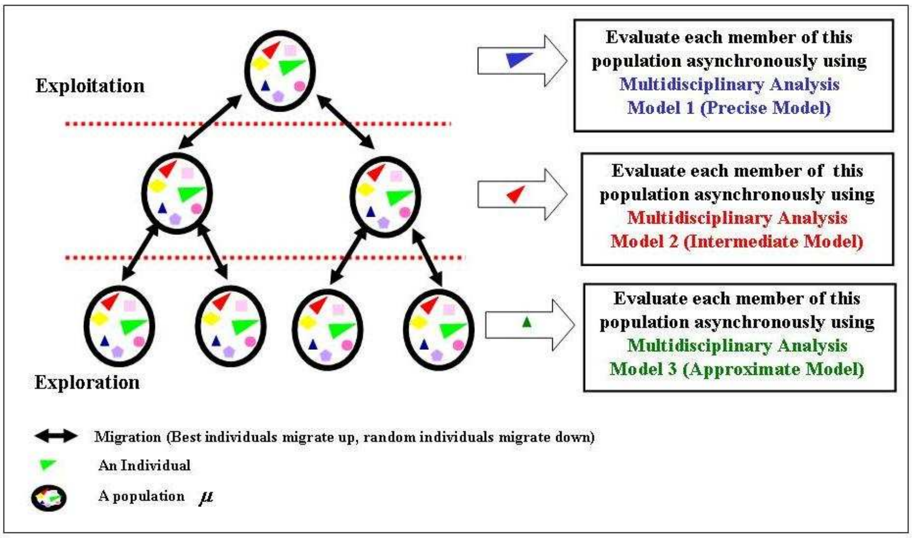

Figure 3.7: Hierarchical topology of EAs for MDO.

Practical implementations of multiple fidelity models will be presented in Chapters 5, 6, 7 and 8.

3.5.3 Architectural Design, Information Access and Post-processing Implementation

To satisfy the architectural design, information access and post-processing requirements defined in Section 3.3, the evolutionary methods and framework use an object-oriented platform in *C++*. The benefits of using an object-oriented software is the ease of implementation and extension of software in a modular fashion by the use of classes and methods. The core of this platform is a GUI, which provides a user-friendly application for its use in an industrial or academic environment.

There are many platforms and options for GUI development, but knowledge of *C++* and object-oriented principles were the main consideration, for this reason the Fast Library Toolkit (FLTK) and the Fast Light User Interface Designer (FLUID) [166] were selected.

This toolkit is a free source code that provides a friendly and easy-to-use environment for the implementation. The designed GUI is named AeroMDO and consists of five main modules, as illustrated in Figure 3.8. These are: Design and Analysis, Design of Experiments (DOE), Parallel Computing and Post-processing. The GUI facilitates development, extension and modifications of modules in a rather simple manner. The user has to create only a few subroutines within the corresponding module.

The GUI is configured in such a way that the analysis codes and optimisation interfaces are embedded within each design module. The user also has the option of performing a single analysis or optimisation using different fidelities of analysis codes or resolutions.

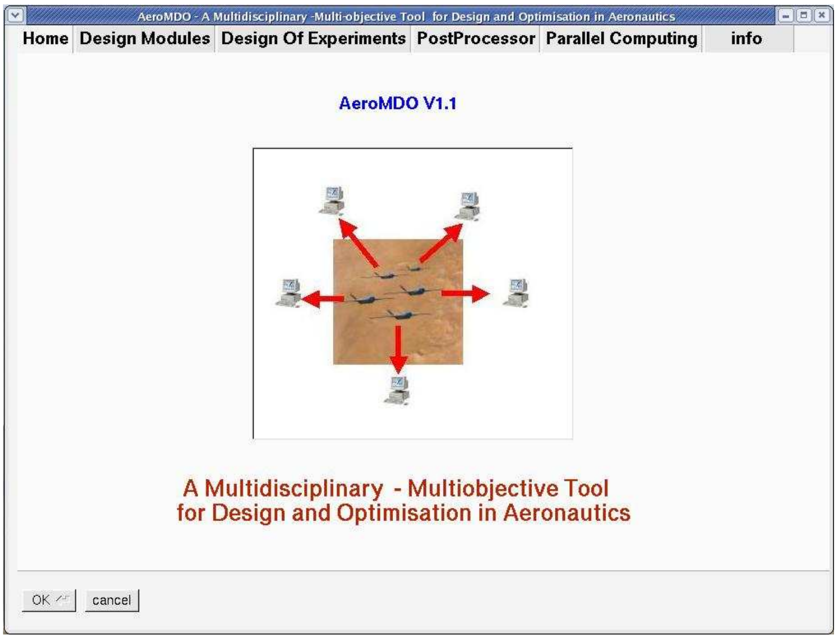

Figure 3.8: Graphical User Interface (GUI).

3.5.3.1 Design and Optimisation Module and Methods

The design module, represented in Figure 3.9, integrates and applies different evolutionary methods for design and allows the user to conduct a single design or optimisation of aerofoils, multi-element aerofoils, nozzle, wing, aircraft design or mathematical test cases. Each of these will be described in detail in the following sub-sections. As developed, the framework is flexible and provides for ease of implementation of other design modules. Current extensions under development by postgraduate students include adaptive wing design, cascade aerofoil design and rotor blade design.

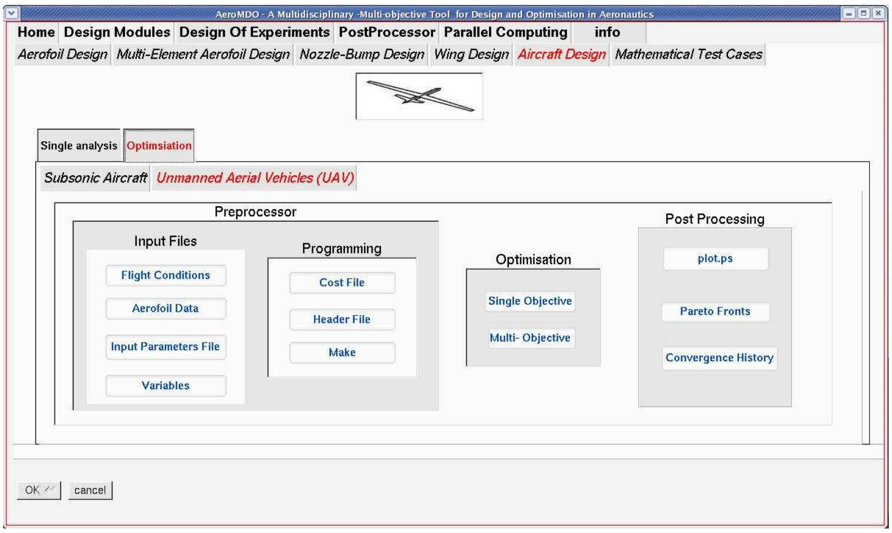

Figure 3.9: Design modules sample.

Analysis tools with variable fidelities were implemented within each design module to allow a robust selection of tools for aeronautical design. A comprehensive set of analysis tools for aerofoil, aircraft and wing design has been assembled, compiled, validated and integrated.

3.5.3.2 Development of Aeronautical Design Modules and Methods

Before using different analysis codes on the optimisation, it is necessary to develop a design module interface. These are a series of files written in *C++* which allow communication between the analysis codes, parallel processing architecture, GUI and optimiser.

When designing the interfaces, a choice had to be made depending on whether the source code for the analysis tool was available or not. In the current implementations, minor modification to the source code was required. Ideally it is desirable to operate only through input/output files. In all these implementations, a design template was used in conjunction with one or two additional files which contain the necessary linking subroutines. This allows a relatively fast implementation, but should such a template not be required, the user can define his/her own template. So far there are subroutines for aircraft, nozzle, wing and full aircraft configuration, which allow the user to perform a single design analysis or a full optimisation.

A general pseudo-code for a new method and its implementation is represented in Algorithm 9. A copy of all the methods, interfaces, modules and algorithms designed during this research is contained in Appendix B.

Algorithm 9 Generic algorithm for aerofoil/wing/aircraft/nozzle/ design.

for;; //Infinite loop
Receive information from optimiser $\mathcal{O}$: $O \rightarrow X, g_i, D$
... X(design variables + evolution path)...
... g_i constraints, D aerodynamic, structural data (flow conditions, modes)
Generate Geometry: $G = f(X)$, G =(aerofoil wing nozzle, aircraft)
    for i =0, n //n Number of objectives
        Evaluate: $J(G, g_i, D)$ // Analyse candidate geometry
        Convergence: Check for convergence
        Calculate the fitness vector: $f_i = J(G, g_i, D)$ $f_i = f_i(X)$
        Return computed individual to optimiser:
        ...$X + f_i \rightarrow O$, X(design variables + evolution path+ Fitness vector)
    end loop
loop

3.5.3.3 Aerofoil Design and Optimisation Module and Methods

This module integrates several robust methods and algorithms and allows the user to perform a single aerofoil analysis test or a full optimisation routine. Three different analysis codes can be used: a panel method, an Euler+ Boundary Layer and a Navier-Stokes solver. Figure 3.10 illustrates this module. Details of the analysis tools used within this module and applications are presented in Chapters 5 and 6.

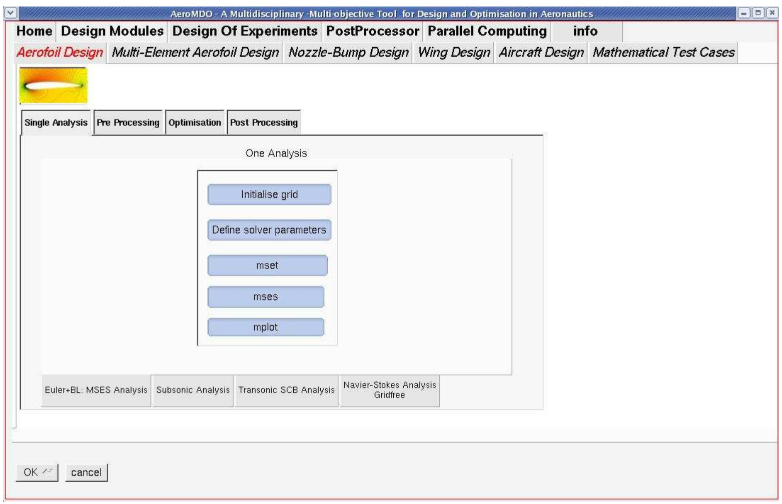

Figure 3.10: Aerofoil design module.

3.5.3.4 Multi-element Aerofoil Design and Optimisation Module and Methods

Similar to the aerofoil design module, this module integrates several robust methods and algorithms and allows the user to perform a single multi-element aerofoil analysis or full optimisation. The user can choose from an Euler or Navier-Stokes evaluation. Figure 3.11 illustrates this module. Details of the analysis tools used within this module and applications are presented in Chapter 6.

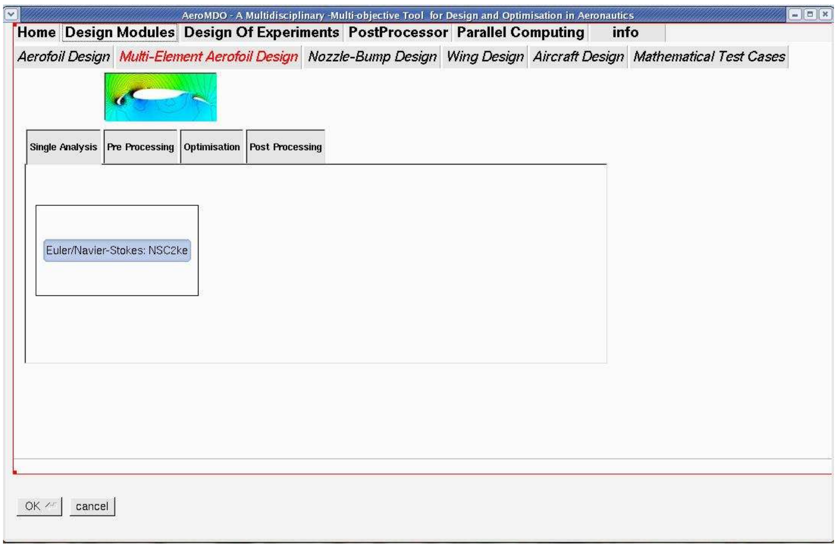

Figure 3.11: Multi-element aerofoil design module.

3.5.3.5 Nozzle and Bump Design and Optimisation Module and Methods

The Nozzle and Bump design module allows a single two-dimensional analysis or optimisation using the CUSP scheme developed by Srinivas [164]. Different design and optimisation problems were studied in the course of this research, for simplicity purposes; they will be omitted in this thesis but can be found in Whitney *et al.* [180].

3.5.3.6 Wing Design and Optimisation Module and Methods

This module integrates several robust methods and algorithms and allows the user to conduct a series of analyses for wing design and optimisation; these include single, multi-objective or multidisciplinary design studies. The module also allows for aero-structural wing design and analysis studies. Figure 3.12 illustrates this module. Details on the analysis tools used within this module and applications for multi-objective and multidisciplinary design are presented in Chapters 7.

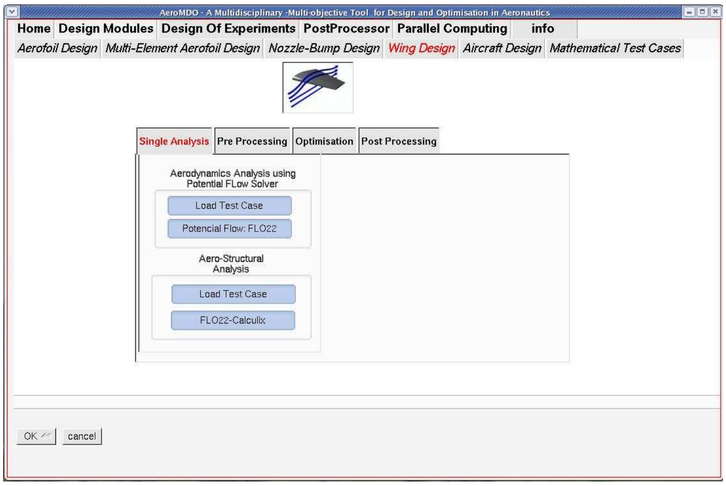

Figure 3.12: Wing design and optimisation module.

3.5.3.7 Aircraft Design and Optimisation Module and Methods

This module, illustrated in Figure 3.13, integrates several robust methods and algorithms and allows the user to analyse different problems in aircraft design using two different analysis codes, one developed by the author of this thesis: the Aircraft Design and Analysis Software (ADA), or using the FLOPS software [105]. Different types of aircraft can be designed and optimised including subsonic, Unmanned Aerial Vehicles, transport or supersonic aircraft. Single- or multi-objective optimisation studies can be performed. Also, comparison of multi-objective approaches such as Pareto optimality or the Nash equilibrium approach can be investigated. Details of the analysis tools used within this module and applications are presented in Chapter 8.

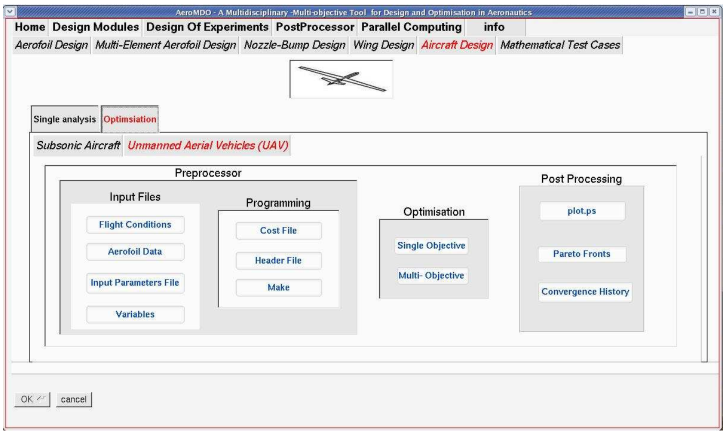

Figure 3.13: Aircraft design module.

3.5.3.8 Mathematical Test Functions Module and Methods

As indicated in Chapter 2, Section 2.6.5, it is important to test the robustness and performance of an optimisation method before deciding on its application to real-world problems. This module integrates several robust methods and algorithms and allows the user to design, and evaluate single-, or multi-objective mathematical test functions. The current implementation includes a mathematical test function for single or multiple objectives, constrained optimisation, goal programming and DOE problems. It is designed so that it can be easily extended to analyse other mathematical test functions. Figure 3.14 illustrates this module.

A general pseudo-code for the implementation of mathematical test functions for multi-objective problems is described in Algorithm 10. Detailed description and results on these types of functions are illustrated in Chapter 4.

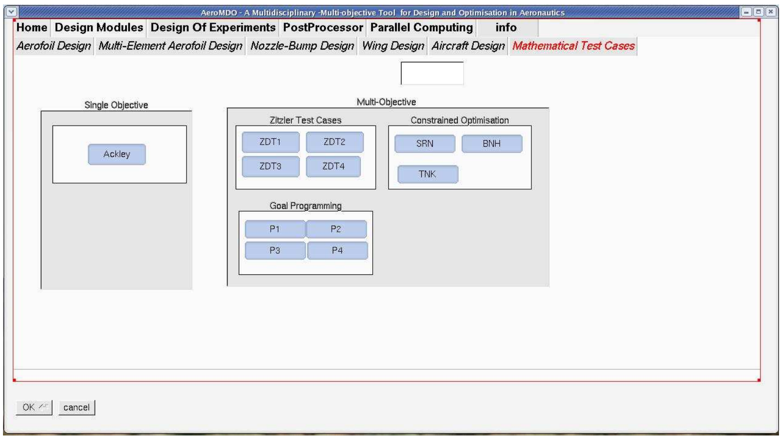

Figure 3.14: Mathematical test functions module.

Algorithm 10 Mathematical test function algorithm.

Define EA parameter:s population size, buffer length. maximum number of...
...function evaluations, type of recombination, number of runs.
Define function to evaluate: f_i :Fonseca(FON), Zitzler (ZDT3)
Define number of parameters: p
Evaluate: $\lambda^{g+1} = f(\lambda_M^{g+1})$
for i=0, n: Number of runs:
  Initialise: $init(\mu^0)$
  $g = 0$
  while stopping condition not met,
    Recombine: $\lambda_R^{g+1} = reco(\mu^g)$
    Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$
    Evaluate: $\lambda^{g+1} = f(\lambda_M^{g+1})$
      Compute *g.h* as in Deb: $g.h$
      Compute and return fitness vector: $f(x) = f_1, f_2, ...f_n$
    Select: $\mu^{g+1} = sel(\mu \cup \lambda)$ (plus strategy) or,
      $\mu^{g+1} = sel(\lambda)$ (comma strategy)
    $g = g + 1$
loop

3.5.3.9 Parallel Computing Module and Methods

As discussed in Chapter 2, recent work on multi-objective parallel evolutionary algorithms has allowed significant performance and robustness gains in global and parallel optimisation. The main problem is that researchers usually do not provide details of their implementations. In the following we classify the current implementation:

Classification of the Parallel Implementation

The algorithm used in this research is similar to the hierarchical hybrid pMOEA model described in Cantu-Paz [25]. It uses a master-slave pMOEA but incorporates the concept of isolation and migration through a hierarchical topology binary tree structure, in which each level executes different MOEAs/ parameters (heterogeneous). The distribution of objective function evaluations over the slave processors is where each slave performs *k* objective function evaluations. The method proposed by Whitney [179] has been implemented.

The parallel environment used is a cluster of PCs, wherein the master carries on the optimisation process while remote nodes compute the solver code. The message-passing model used is the Parallel Virtual Machine (PVM) [55]. In order to accelerate the convergence of the method, the implementation uses a multi-level optimisation approach involving various mesh densities and solvers with an asynchronous computing.

Under the current implementation only the PVM can be executed; future work will focus on extending the capabilities for MPI or other methods. With the parallel processing implementation, different design points on the Design of Experiments module can be executed in parallel, reducing the computational time of the sampling process. The current implementation has been tested in two different clusters of computers:

- BORG: A cluster of 32 PCs of performance varying between 1.0 and 3.8 GHz at the University of Sydney. This is a cluster of heterogeneous CPUs, RAMs, caches, memory access times, storage capabilities and communication attributes at the University of Sydney. This cluster can be configured with up to 18 machines and consists of Pentium 4 machines with 1.6 and 2.4 GHz processors using the Intel® 850 chipset, connected via a 100 *mbit/s* fast Ethernet switch.

- Barossa: Barossa is a distributed memory computer, constructed from 152 dual processor Linux computers, connected together by a fast network. Its processors are 3GHz Intel Pentium 4 Xeons. This cluster is administered by the Australian Centre for Advanced Computing and Communications (AC3).

This module allows the users to dynamically create, add or delete nodes on the parallel implementation. Figure 3.15 illustrates the implementation of the Parallel Computing module. Studies on parallel computing are described in Chapter 5.

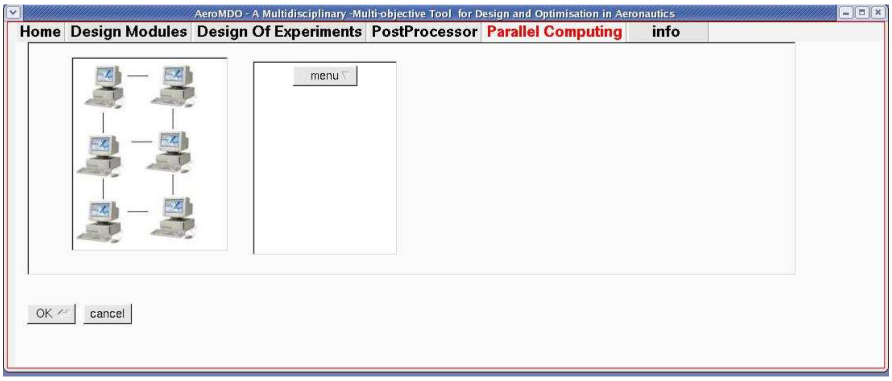

Figure 3.15: Parallel computing module.

3.5.3.10 Design of Experiments Module and Methods

As indicated in Chapter 2, one of the drawbacks of EAs is that they suffer from slow convergence. By providing a DOE capability, the desirable characteristics of EAs and surrogate models such as RSM/DACE and obtain an efficient optimisation system are combined. Within this context, the DOE samples a number of design candidates on which the analysis code will run, then the surrogate model is constructed for the computationally expensive problem.

In the current implementation the user can define and choose from different strategies such as rectangular grids, Latin Hypercube (LH) for sampling and DACE/Kriging for DOE. There are several software codes developed for DOE; after a careful evaluation it was decided to implement the DACE tool box [91], which is robust and allows different options for sampling and DACE strategies. This software was translated into Octave (a mathematical package common in most UNIX installations) and implemented in the framework. But, if desired, or if the nature of the problem requires a different DOE technique, it can be easily implemented in modular fashion. Figure 3.16 illustrates the DOE module within the GUI.

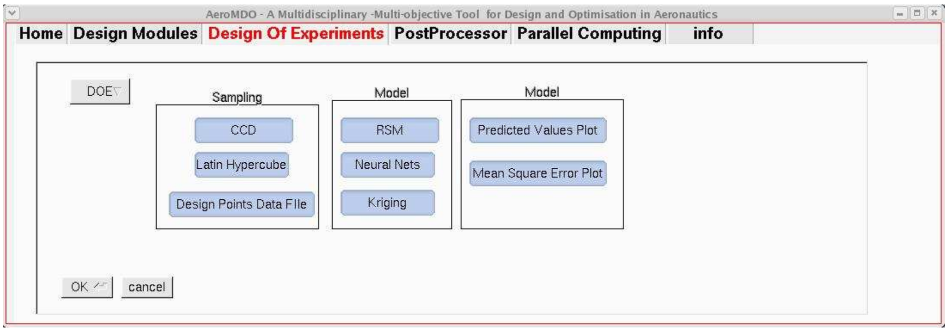

Figure 3.16: Design of Experiments (DOE) module.

3.5.3.11 Post-processing

Post-processing capabilities are embedded within each module; this is due to the fact that different applications require different visualisation techniques. The approach considered was to develop simple applications for GNUplot (a graphics package common in most UNIX installations) and use the full benefits of the visualisation capabilities within each analysis software.

Common to all design modules is visualisation of the progress of the evolution process (fitness function versus number of function evaluations), visualisation of Pareto fronts and the Nash equilibrium point for multi-objective problems and top, three-dimensional and side views of the optimal geometries. Visualisation tools within each analysis software include the pressure coefficient distribution on the aerofoil using an Euler+BL solver or pressure and Mach contours using Navier-Stokes solver. It is also possible to generate a DXF file which can be read and modified in CAD software such as QCad or Autocad to produce drawings which can be used to make a wind tunnel model or used with other analysis tools. For the aircraft design and optimisation implentation it is also possible to generate a CAD model and evaluate the performance characteristics of the extreme Pareto aircraft configurations using the commercial flight simulation software X-PLANE [106].

3.6 Qualitative Evaluation of the Methods and Framework

Before proceeding into the application of the methods to mathematical test functions and real-world problems it is important to provide a qualitative assessment of the fundamental characteristics of the proposed methods and framework. As designed, the methods and framework address the requirements of problem formulation and optimisation tools, problem execution, architectural design and information access. The methods also satisfy the necessary requirements on problem *formulation* and *algorithm* described in Section *2.5.5*:

3.6.1 Formulation

- **Consistency:** The framework and methods are developed and implemented consistently; similar principles are used for all modules and problem formulations.

- **Well Posedness**: Well posedness is guaranteed within each design and analysis formulation. This will be further illustrated in the next chapters.

- **Equivalence of Formulations:** In the current implementation, different multidisciplinary analysis formulations are evaluated directly with zero-th order methods. There are no derivatives or transformation on the mathematical formulation of the problem that introduce noise or require fine-tuning of the solution.

- **Optimality Conditions:** EAs are global optimisation techniques where in optimality conditions are imposed from the beginning to guarantee feasible designs.

- **Sensibility to Solutions:** In contrast to formulations that use deterministic optimisers, stochastic techniques are based on random searches. For mathematical test functions or inverse problems, the target optimum solution or known Pareto front can be verified. For direct optimisation problems, being a stochastic process, there is no guarantee that the same solution will be found in different runs, but a monotonic improvement of the objective function is expected during the evolution process.

- **Ease of Implementation:** Different from traditional approaches where the ease of implementation is determined usually only by the way the system of equations are coupled and solved, the implementation involves other types of complexities as the proper definition of EA-specific parameters, such as population size, buffer size and migration criteria. The choice of these parameters will direct the convergence properties and computational expense involved. A discussion of these choices will be presented in the next chapters once the test cases are evaluated for different parameter settings.

- **Robustness:** As mentioned, with the zero-th order there is no relaxation of the problem. There is no requirement for fine-tuning, the algorithm is robust from the start. The method has been shown to be robust, successfully coupled and validated for mathematical test functions and real-world problems in aeronautics with increasing complexity.

3.6.2 Algorithm

- **Local convergence rates, global convergence properties and iterative cost:** EAs are global optimisation techniques; the algorithms developed in this thesis have been shown to be robust for finding optimal solutions for both mathematical test cases and real-world problems. There is no intrinsic iteration on the evolutionary algorithm; the iteration cost is relative to the internal iterative cost of each disciplinary and multidisciplinary analysis. The cost and performance of the computation will be indicated for each particular application in the subsequent chapters. Conclusions will be drawn for each case.

- **Performance with increasing number of variables:** The application of EAs is limited when the number of design variables is large, as it is in the case of MDO problems. It is important to highlight that in this work the use of EAs is restricted to conceptual and preliminary MDO studies where the number of variables is still relatively small, less than a hundred, and where the use of EAs is still of potential benefit. On a larger scale, when coupled with other techniques such as Design of Experiments (DOE), the use of EAs can be extended for an increased number of variables.

3.7 Summary

This chapter developed and implemented several evolutionary methods algorithms and a framework for multi-objective and MDO, in which different mathematical test functions and aeronautical problems can be analysed. The chapter provided a detailed description of the different elements. These details included selection and extension of a robust optimisation tool, several algorithms for problem formulation, the definition and development of a GUI and implementation of different modules for design, optimisation, post-processing and parallel computing.

Therefore, we have a complete set of numerical tools for designing, analysing and optimising mathematical test functions, and also real-world problems in aeronautics. In the following chapters the methods are evaluated in terms of their application to mathematical test functions, and real-world problems with different complexities.

Chapter 4

Evaluation and Performance on Mathematical Test Functions

"I have been impressed with the urgency of doing. Knowing is not enough; we must apply. Being willing is not enough; we must do." Leonardo da Vinci.

4.1 Introduction

The objective of this chapter is to ensure the feasibility of the proposed evolutionary methods and framework for complex mathematical test functions before attempting their application to complex multi-objective and multidisciplinary design optimisation problems in aeronautics. One of the most important considerations when developing software and software integration is to provide capabilities for benchmarking test functions which will guide the user in providing information on different parameter settings and the performance of the method and optimisation algorithms. The framework developed in Chapter 3 includes capabilities for benchmarking mathematical test functions. Figure 4.1 illustrates the module developed for this task.

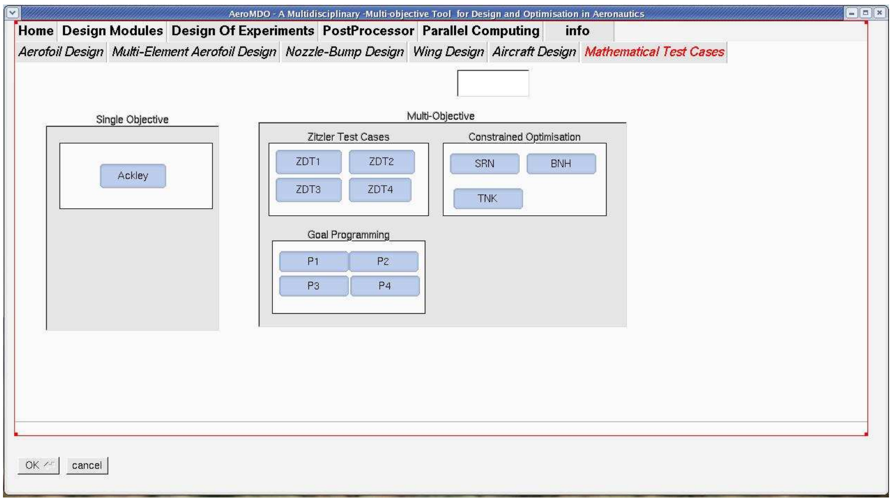

Figure 4.1: Mathematical test function module.

4.2 Multi-objective Test Functions

As discussed in Chapter 2, there are different methods and approaches to solve multi-objective and non-linear goal programming problems. These include the weighted goal and the lexicographic approach.

These methods have several drawbacks, as the user has to define in advance a series of weight factors, which define the relative significant importance of each objective; this is highly subjective to the user and has difficulties for problems that have non-convex decision spaces [28, 42].

With the current implementation of evolutionary methods, a Pareto set of solutions can be found. The advantage of the Pareto approach is that each Pareto solution corresponds to a specific set of weight factors. The Pareto approach can obtain multiple solutions in a single run (simultaneously) without a-priori definition by the user.

There are different studies on mathematical test functions for evolutionary algorithms and multi-objective EAs; these include works by Fonseca and Fleming [52], Zitzler [185], Veldhuizen and Lamont [170] and Deb [40, 42, 43]. These references provide comprehensive sets of mathematical test functions that highlight the difficulties of multi-objective evolutionary algorithms and methods converging to the Pareto optimal front.

There are different problem features that might cause a multi-objective genetic algorithm to converge to the true Pareto front. Following the notation given in Deb [41], three functions f_1, g_1 and h are considered:

- f_1 test the ability to handle difficulties along the Pareto optimal region,

- g_1 test the ability to handle difficulties lateral to the Pareto optimal region and continuous optima and
- h test the ability to handle difficulties arising from different shapes of the Pareto optimal region.

A preliminary study on the use of the optimisation algorithms used in this research was conducted by Whitney [179]. This study highlighted the potential of the algorithms, but a comprehensive test suite was required and was developed to evaluate the performance of the algorithm and methods for complex multi-objective and MDO problems.

This test suite includes complex multi-objective test functions, a series of constrained optimisation functions, and some non-linear inverse/goal programming problems. By conducting an analysis of these functions, an algorithm can be tested for multi-modal multi-objective problems, deceptive multi-objective problems, multi-objective problems having convex, non-convex and continuous optima fronts and non-uniformly represented Pareto optimal fronts.

Table 4.1 summarises the test suite. A method and an algorithm were developed for each test case.

4.2.1 Parameter Setting Investigation

For each test case several runs were performed to find a good setting for the evolutionary parameters. An algorithm was devised for this task. This algorithm creates different combinations of parameters for the evolutionary optimisation parameters (population size, tournament size, intermediate/discrete recombination, buffer size) and produces a graph which shows the optimal Pareto front and the computed Pareto front with different combinations of parameter settings.

One of such graphs for the ZDT2 problem is illustrated in Figure 4.2. In this graph the nomenclature is PopSize-D-PPP-BBB-T. The first digit represents a population size multiplier, the second set of digits represents the population size, the third set of digits represents the buffer size and the last digit represents the tournament size. For example, PopSize-1-100-250-2 means that a population size of 100, a buffer size of 250 and a tournament ratio of 2 are used. By inspection it can be determined which combination of parameters produces a uniform and lower Pareto front. In the following test cases, only results with the best parameter settings are reported.

Function	Definition	Expression
ZDT1	Definition	$Objective\ 1: \quad f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $Objective\ 2: \quad f_2 = 1 - g.h$ $Bounds: \quad 0 \leq x_i \leq 1$
ZDT2	Convex	$Objective\ 1: \quad f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2$ $Objective\ 2: \quad f_2 = 1 - g.h$ $Bounds: \ 0 \leq x_i \leq 1$
FON	Non-convex	$Objective\ 1: \quad Minimise\ f_1(x) = 1 - exp\left(-\Sigma_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right)$ $Objective\ 1: \quad Minimise\ f_2(x) = 1 - exp\left(-\Sigma_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$ $Bounds: \ -4 \leq x_i \leq 4 \ \ i = 1, 2, 3..., n$
ZDT3	Discontinuous	$Objective\ 1: \quad f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right) sin(10\pi f_1)$ $Objective\ 2: \quad f_2 = g.h$ $Bounds: \ 0 \leq x_i \leq 1$
TNK	Discontinuous – Constrained	$Minimise \quad Objective\ 1: \quad f_1(x) = x_1$ $Objective\ 2: \quad f_2(x) = x_2$ $subject\ to \quad C(x) \equiv x_1^2 + x_2^2 - 1 - 0.1cos\left(16arctan\frac{x_1}{x_2}\right) \geq 0$ $C_2(x) \equiv (x_1 - 0.5)^2 + (x_2 - 0.5)\, x_1^2 \leq 0.5$ $Bounds: \ 0 \leq x_1 \leq \pi, \quad 0 \leq x_2 \leq \pi$
SRN	Discontinuous–Constrained	$Objective\ 1: \quad f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2$ $Objective\ 2: \quad f_2(x) = 9x_1 + (x_2 - 1)^2$ $C_1(x) \equiv x_1^2 + x_2^2 \leq 225,$ $C_2(x) \equiv x_1 - 3x_2 + 10 \leq 0$ $Bounds: -20 \leq x_1 \leq 20, -20 \leq x_2 \leq 20$
BNH	Convex – Constrained	$Objective\ 1: \quad f_1(x) = 4x_1^2 + 4x_2^2$ $Objective\ 2: \quad f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$ $C_1(x) \equiv (x_1 - 5)^2 + x_2^2 \leq 25,$ $C_2(x) \equiv (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$ $Bounds: \quad 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3$

Table 4.1: Test function suite for multi-objective problems.

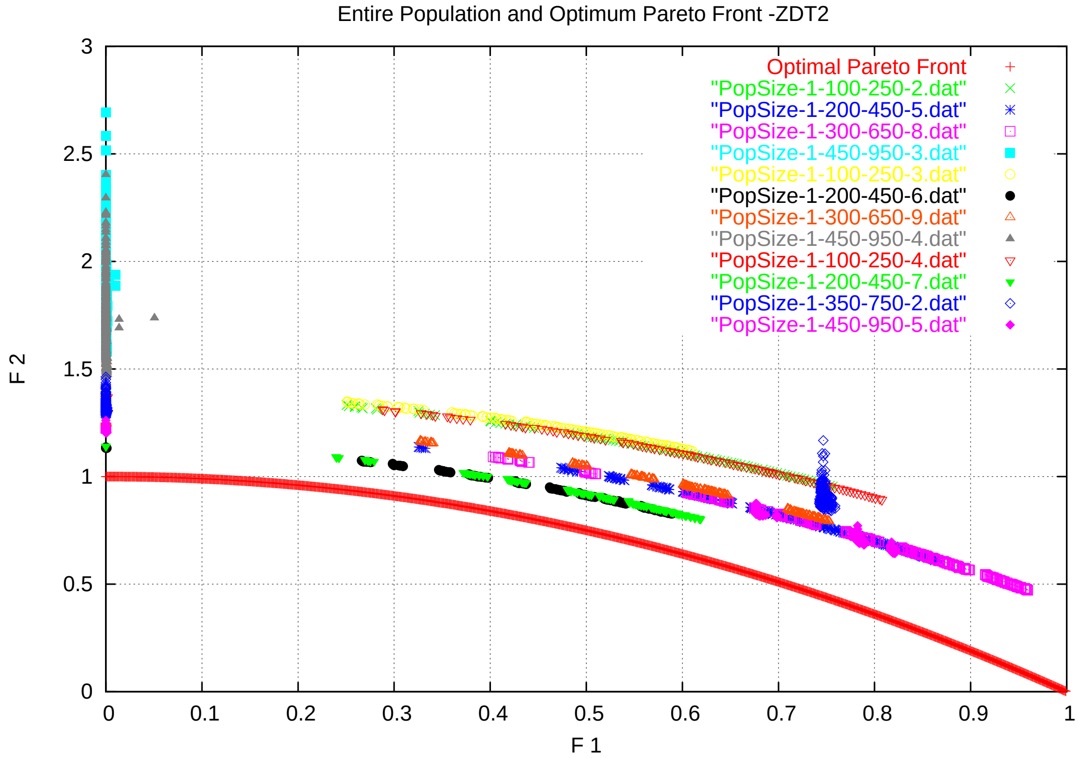

Figure 4.2: Parameter-setting investigation example.

4.2.2 Convex and Non-convex Pareto Optimal Front

The following problems illustrate the capabilities of each algorithm to capture convex and non-convex Pareto Optimal fronts

ZDT1

This is an $n = 30$ variable problem with a convex Pareto optimal front. The optimal region corresponds to the region $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 1, 2, 3..., 30$. Figure 4.3 shows how this algorithm finds the optimal Pareto set. The optimal parameter settings for this problem are a population size of 150, a buffer size of 250, a tournament size of 2 and discrete recombination.

ZDT2

This is a difficult problem with 30 variables that has a non-convex Pareto front. The Pareto optimal front corresponds to $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 1, 2, 3..., 30$. Figure 4.4 shows the Pareto optimal front and the best solution found using a parameter settings of population

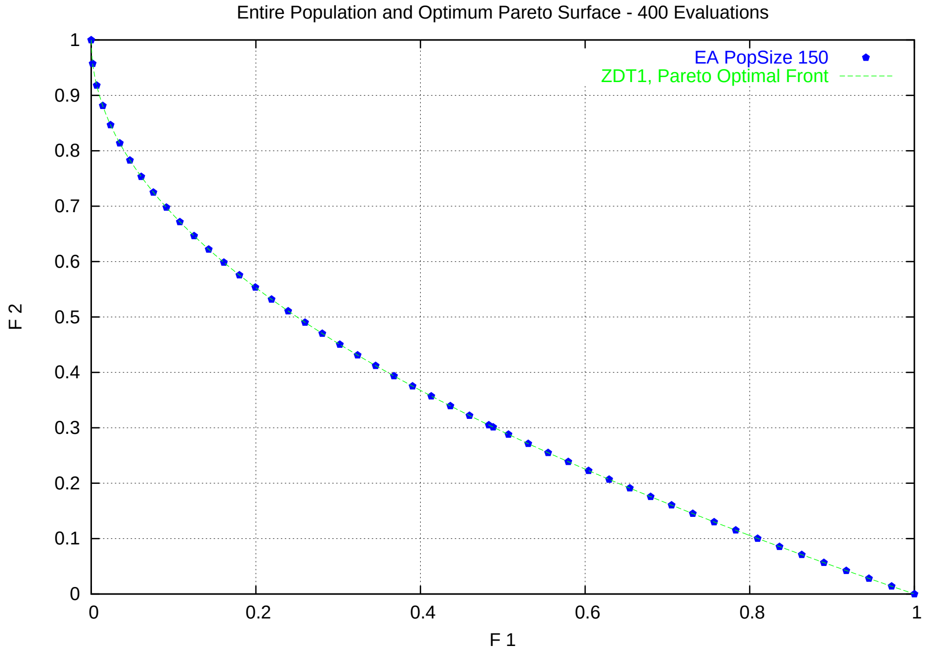

Figure 4.3: Pareto optimal front for convex Pareto optimal front problem ZDT1.

size 200, buffer size of 350, tournament size of 2 and discrete recombination.. The results in this case are not particularly promising, but compare well with the results found by traditional multi-objective algorithms such as NSGAII and SPEA [28, 42].

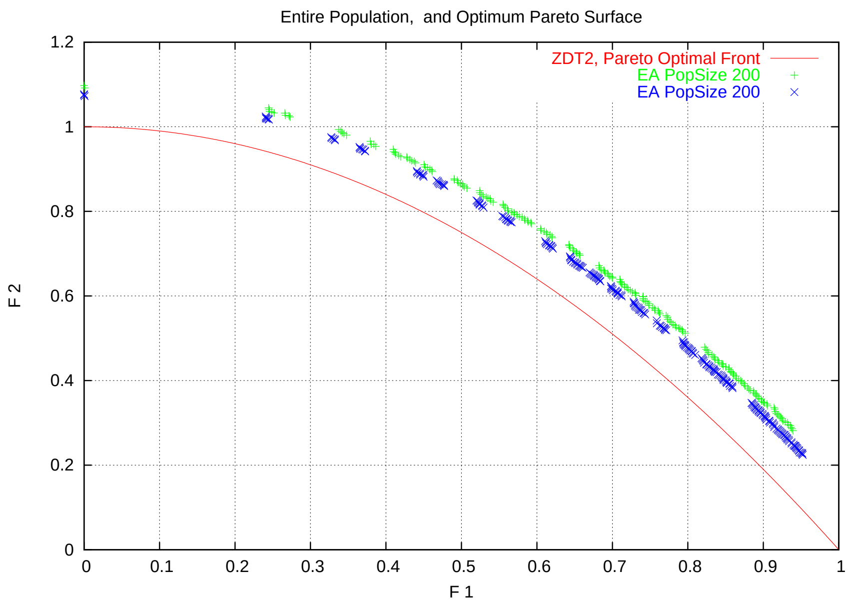

Figure 4.4: Pareto optimal front for non-convex Pareto optimal front problem ZDT2.

FON

Fonseca and Fleming [52] developed a two-objective optimisation problem with n variables. As described in Deb [42], the optimal solution to this problem is $x_i \, \epsilon \, [-1/\sqrt{n}, +1/\sqrt{n},] \leq 1$ and $x_i = 0$ for $i = 1, 2, 3..., n$. Figure 4.5 shows the objective space for $n = 10$. A special characteristic of this problem is that the Pareto optimal set is non-convex, hence a weighted sum approach will fail or have difficulties finding the solution [42]. A population size of 50, a buffer size of 100, a tournament size of 2 and intermediate recombination produced good results.

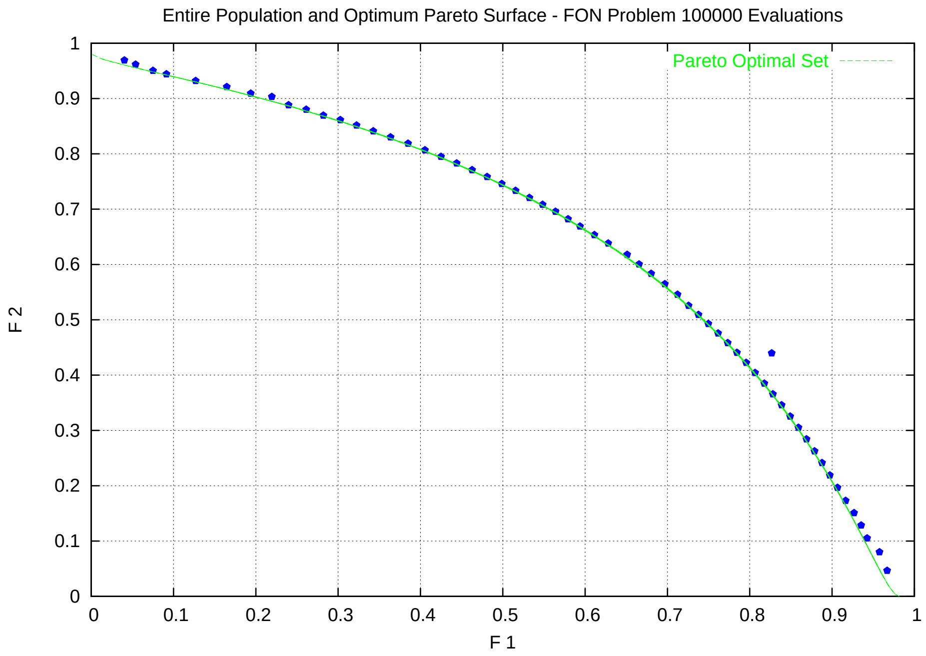

Figure 4.5: Pareto optimal front for non-convex Pareto optimal front problem FON.

4.2.3 Discontinuous Pareto Optimal Front

Amongst the discontinuous Pareto optimal front test functions, it is important to test the capabilities for convergence of this algorithm to a known Pareto front. The ZDT3, is a $n = 30$ variable unconstrained problem which has five discontinuous Pareto optimal regions, and would not be able to be solved using a traditional optimiser. Figure 4.6 shows the results of the computation and the Pareto optimal region which corresponds to $x_i^* = 0$. This algorithm is able to find all five discontinuous regions with a good spread of the solutions. The parameter settings used were a population size of 150, buffer length of 350, a tournament size of 2 and discrete recombination.

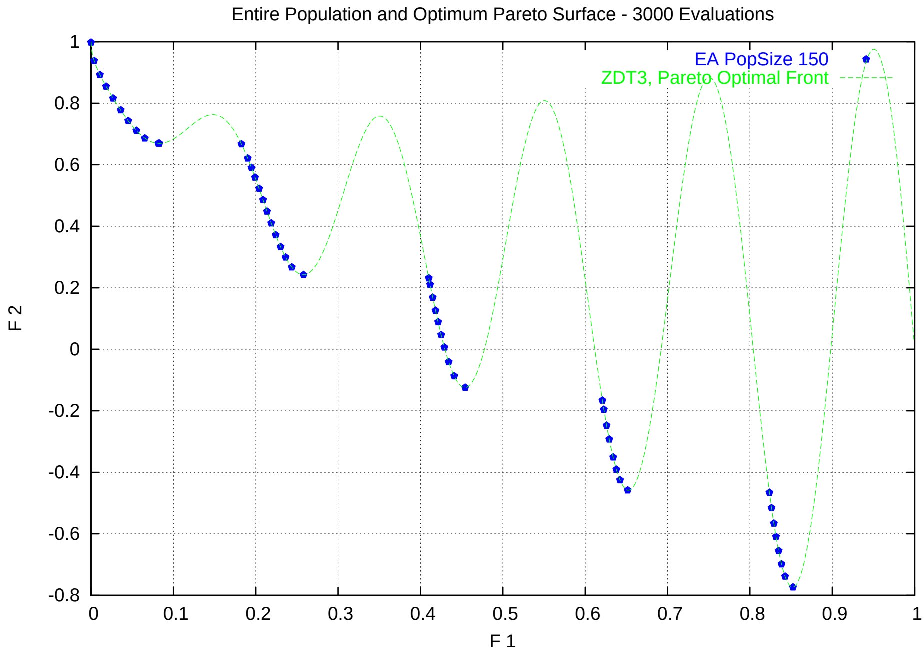

Figure 4.6: Pareto optimal front for discontinuous Pareto optimal front problem ZDT3.

4.2.4 Constrained Optimisation Test Cases

Real-world engineering problems are subjected to several manufacturing, technical and human constraints. This section studies the performance of the method for mathematical test functions involving one or several constraints.

TNK

The first constrained optimisation problem considers a discontinuous constrained optimisation problem. It would be difficult if not impossible for a conventional optimiser that uses gradient-based techniques to find the solution to this problem. The feasible decision variable design space is shown in Figure 4.7. Being $f_1 = x_1$ and $f_2 = x_2$, the feasible objective space is the same as the feasible decision variable space. Figure 4.7 shows the feasible search space and Pareto optimal front with an optimal combination of parameters: population size of 100, a buffer length of 400, a tournament size of 2 and discrete recombination.

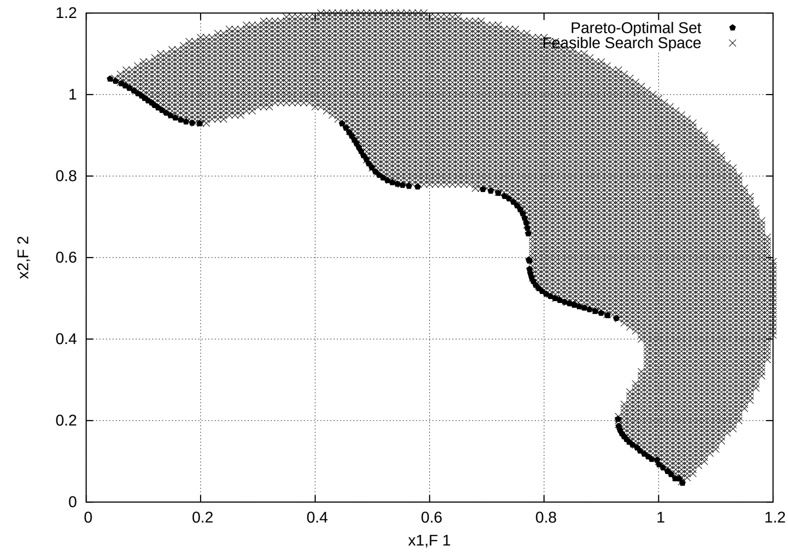

Figure 4.7: Feasible search space and Pareto optimal front for discontinuous constrained Pareto optimal front for problem TNK .

SRN

This is a two-variable constrained optimisation problem. The feasible decision variable space is shown in Figure 4.8. Figure 4.9 shows the feasible search space and Pareto optimal front for this problem. The optimal parameter settings used were a population size of 150, a buffer length of 350, a tournament size of 2 and discrete recombination.

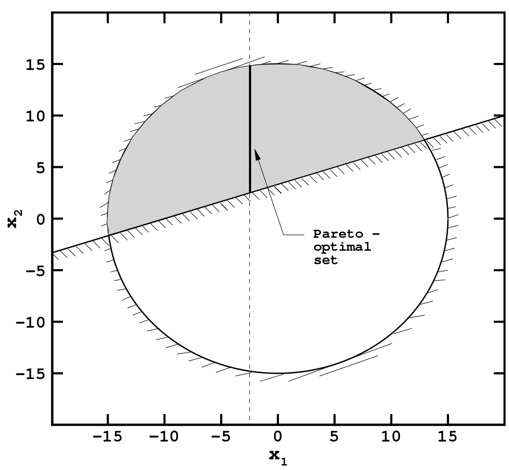

Figure 4.8: Decision variables space for constrained problem SRN.

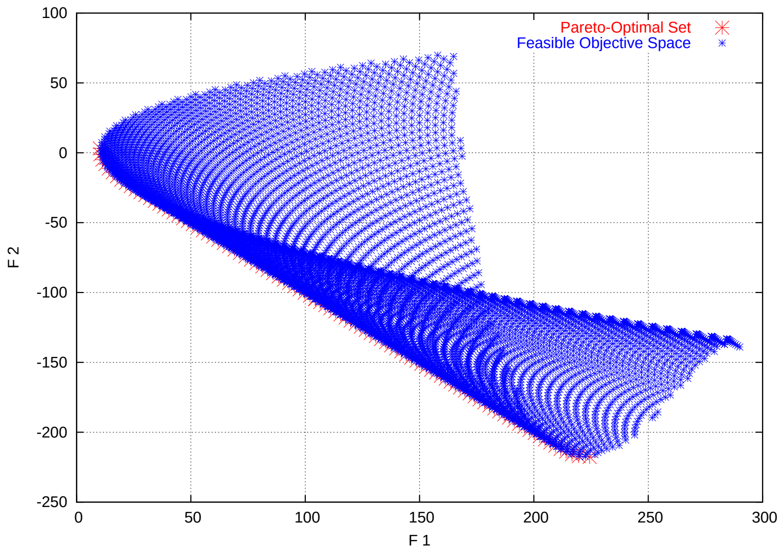

Figure 4.9: Feasible objective search space and Pareto optimal front for constrained problem SRN.

BNH

The BNH is a two-variable constrained optimisation problem. Figure 4.10 shows the feasible search space and Pareto optimal front with a population size of 100, buffer length of 400, tournament size of 2 and discrete recombination.

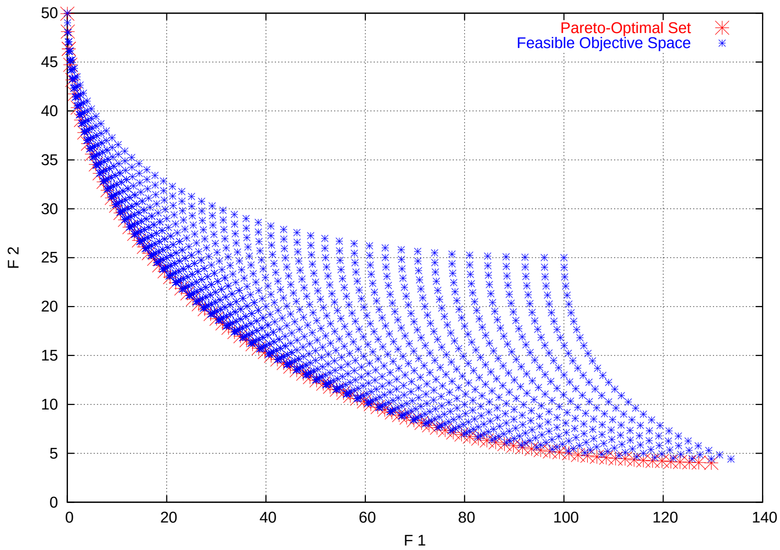

Figure 4.10: Feasible objective search space and Pareto optimal front for constrained problem BNH.

4.2.5 Non-linear Goal Programming/Inverse Design

Practising engineers often know the desired target or goal for their problems, but do not always have tools for finding a feasible solution. As indicated in Chapter 2, EAs have shown some benefits for inverse/non-linear goal-programming problems.

Following Deb [42], in goal-programming the aim is to find a set of solutions that attain a predefined target for one or more objective functions. But if no such set exists, the aim is to find solutions which minimise deviation from the desired targets.

In the approach proposed by Deb [42] the goals are converted into objective functions by minimising the deviations from the desired target. In this case the relationships defined in Table 4.2 are used.

Type	**Goal**	**Objective function**
$\leq$	$f_j(x) \leq t_j$	$Minimise\ \langle f_j(x) - t_j \rangle$
$\geq$	$f_j(x) \leq t_j$	$Minimise\ \langle t_j - f_j(x) \rangle$
$=$	$f_j(x) \leq t_j$	$Minimise\ \lvert f_j(x) - t_j \rvert$
$Range$	$f_j(x) \leq t_j$	$Minimise\ max\ \left(\left\langle t_j^l - f_j(x) \right\rangle, \left\langle f_j(x) - t_j^u \right\rangle \right)$

Table 4.2: Non-linear goal-programming (NLGP) relationship operators.

The bracket operator returns the value of the operand if the operand is positive, otherwise returns zero. The advantage of this formulation is that:

- there is no need for any additional constraint for each goal,
- since EAs do not require objective functions to be differentiable, the above functions can be used directly.

In the following subsection the application of the method to three representative goal-programming test functions will be considered. Table 4.4 summarises the test suite. Real-world non-linear goal-programming/inverse aeronautical engineering problems will be discussed in subsequent chapters.

Function	Definition	Expression
Problem 1 (P1)	NLGP	$Minimise \quad \langle f_1(x_1,\ x_2) - 2 \rangle,$ $Minimise \quad \langle f_2(x_1,\ x_2) - 2 \rangle$ $Subject\ to\ F \equiv (0.1 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 10)$ $f_1 = 10x_1\ and\ f_2 = \left(10 + \frac{(x_2-5)^2}{10x_1}\right)$
Problem 2 (P2)	NLGP	$Minimise \quad \langle 0.9 - f_1(x_1,\ x_2) \rangle,$ $Minimise \quad \lvert f_2(x_1,\ x_2) - 0.55 \rvert$ $Subject\ to\ F \equiv (0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1)$ $f_1 = x_1\ and\ f_2 = \left(1 - \sqrt{x_1(1-x_1)}\right)(1 + 10x_2^2)$
Problem 3 (P3)	NLGP	$Minimise \quad f_1 = x_1 \in [0.25, 0.75],$ $Minimise \quad f_2 = \left(1 - \sqrt{x_1(1-x_1)}\right)(1 + 10x_2^2) \leq 0.4$ $Subject\ to\ F \equiv (0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1)$

Table 4.4: Non-linear goal-programming test suite.

Figure 4.11 shows the objective search space (before considering the targets) for problem P1. The feasible objective space lies above the hyperbola. The figure also shows the optimal Pareto front obtained by this algorithm. Here the population size is 100, buffer length of 400, tournament size of 2 and discrete recombination. The figure also shows how this algorithm was able to find different solutions in the design space in one single run. A traditional method using user-predefined weighting factors would require multiple separate runs.

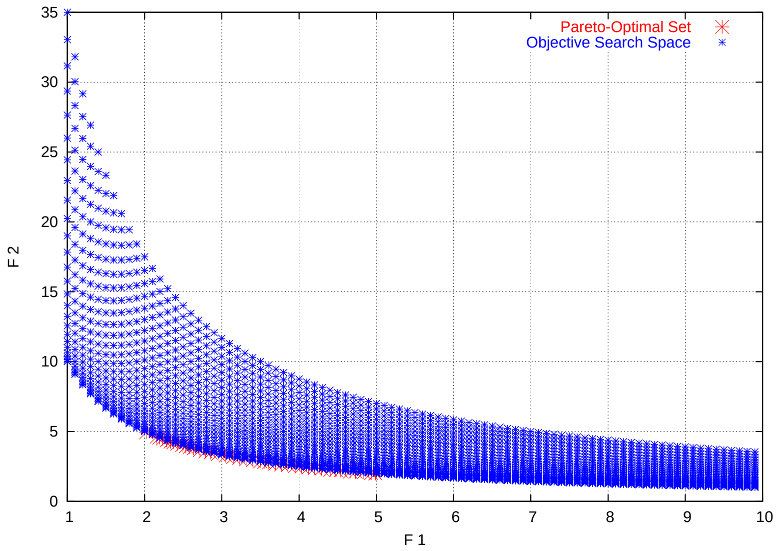

Figure 4.11: Objective search space and optimal Pareto front for NLGP problem 1.

The feasible design space for P2 is represented in Figure 4.12 by the region above the semicircle.

The figure also shows the Pareto optimal front and how this algorithm was able to find many solutions in the desired range in one single run with a population size of 50, buffer length of 100, tournament size of 2 and intermediate recombination.

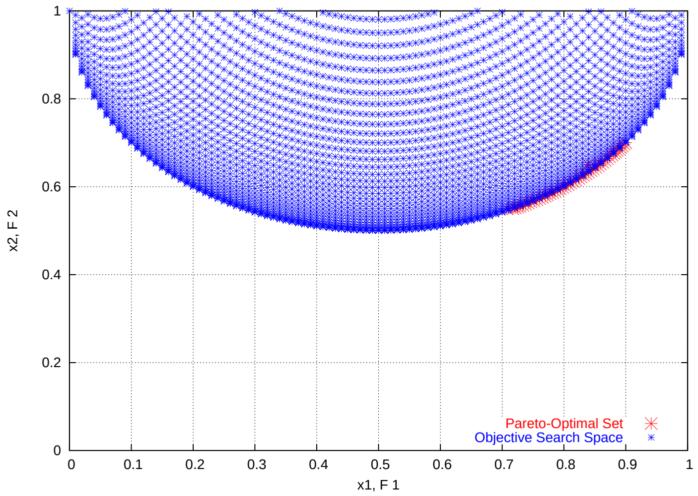

Figure 4.12: Objective search space and optimal Pareto front for NLGP problem 2.

By modifying P2, P3 is obtained; the feasible decision space is shown in Figure 4.13 by the region above the semicircle. As indicated in the figure there is only one solution; $x_1 = x_2 = 0.5$, independently of the weight factors to use. This is due to the fact that the solution is the shortest deviation from the objective space. As illustrated, the population converges to the single solution indicated in the figure. In this problem the robustness of this algorithm is tested to demonstrate how, with a multi-objective approach, it is possible to find a single solution as the Pareto optimal solution. A real-world example similar to this case will be illustrated in Chapter 6.

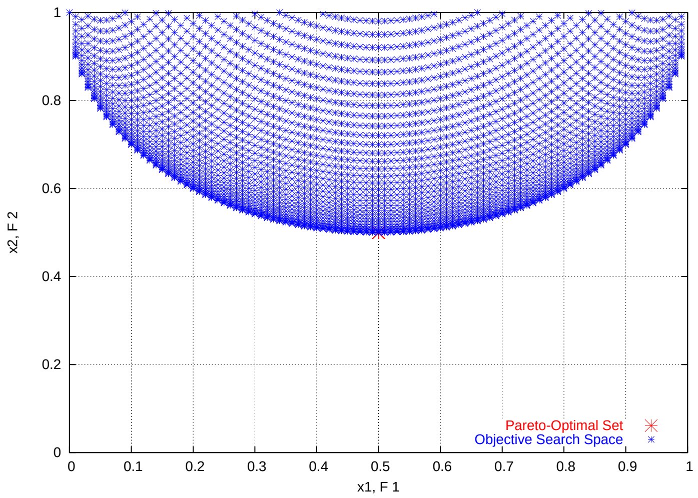

Figure 4.13: Objective search space and optimal Pareto front for NLGP problem 3.

4.3 Summary

This chapter presented an indication of the potential capabilities of the methods and algorithms for multi-objective problems and mathematical test functions. A series of mathematical test problems were selected and then investigated. The objective of these test problems was to gain an understanding of a good parameter setting, the strength and weaknesses of the methods and algorithms. The determination of parameter settings is highly dependent on the problem.

Results indicate that the modular implementation allows new test cases to be implemented in a rather simple manner, the method is robust, well-suited and capable of finding optimal solutions for unconstrained, constrained and non-linear goal-programming problems with convex, non-convex or discontinuous Pareto optimal fronts. These results provide confidence in the application of these evolutionary methods to real-world problems in aeronautics.

Chapter 5

Parallel Evolutionary Computation and Hierarchical Topology

> "*We are at the very beginning of time for the human race. It is not unreasonable that we grapple with problems. But there are tens of thousands of years in the future. Our responsibility is to do what we can, learn what we can, improve the solutions, and pass them on.*" Richard P. Feynman.

5.1 Introduction

The objective of this chapter is to illustrate and demonstrate the coupling of the proposed evolutionary optimisation methods with parallel computations and different fidelity solvers for problems in aeronautics.

As described in Chapter 2, EAs are easy to parallelise; in general the objective of a parallel multi-objective evolutionary algorithms is to increase the effectiveness at finding a solution that is better than or as good as the serial evolutionary algorithm in less time and by exploring more of the solution search space.

The chapter starts with Section 5.2, by outlining some important aspects concerned with the coupling of EAs with analysis tools and parallel computing strategies. In Section 5.3, a general mathematical formulation for shape optimisation problems is illustrated. Section 5.4 considers problems related to inverse aerofoil design, an evaluation study on the use of different fidelity solvers during an optimisation process and a study on the performance of the algorithms with an increasing number of computers. Finally, some conclusions are furnished in Section 5.5. Table 5.1 summarises the test cases devised and evaluated in this chapter.

Type	**Single-/ Multi-objective**	**Description**	**Solver**
Inverse	Single-objective	Single-objective Aerofoil Pressure Coefficient Distribution Reconstruction	Euler+Boundary Layer
Inverse	Multi-objective	Aerofoil Operating at Two Design Points	Euler+Boundary Layer
Inverse	Multi-objective	Two Aerofoils at Two Different Design Points	Euler+Boundary Layer
Inverse	Multi-objective	Multi-objective Single Aerofoil Section Design: First Attempt	Euler and Panel Methods
Inverse	Multi-objective	Multi-objective Single Aerofoil Section Design: Second Attempt	Euler+Boundary Layer
Inverse	Multi-objective	Multi-objective Single Aerofoil Section Design: Increasing Number of Computers	Euler

Table 5.1: Test cases for parallel computing and hierarchical topology of solvers.

5.2 Components of CFD/FEA and pEA Optimisation

A satisfactory coupling of pEAs and CFD/FEA starts by selection of the following elements:

- **Optimisation Method:** In general a pMOEA needs to be robust and efficient, as there is a high computational expense on a CFD/FEA analysis. This research uses the optimisation algorithms described and implemented in Chapter 3.
- **Flow/Structural Analysis Method**: A decision has to be made in order to compromise on the use of a higher fidelity solver, such as a Navier-Stokes solver/Finite Element Analysis which are computationally expensive but accurate, the use of lower fidelity models, such as Euler or panel or analytical methods, which are fast but could be unstable, or a combination of both. Another alternative is using a single model but defining different grid densities.
- **Geometry Representation:** The selection of an appropriate geometric representation, that accounts for the complexities of the design space.
- **Parallel Computing Strategy.** A proper definition of parallelisation strategy: master-slave or island model for example. This depends on the available computer resources and ease of implementation. The parallel computing capabilities of the algorithm and the parallel computing module (Figure 5.1) are used.

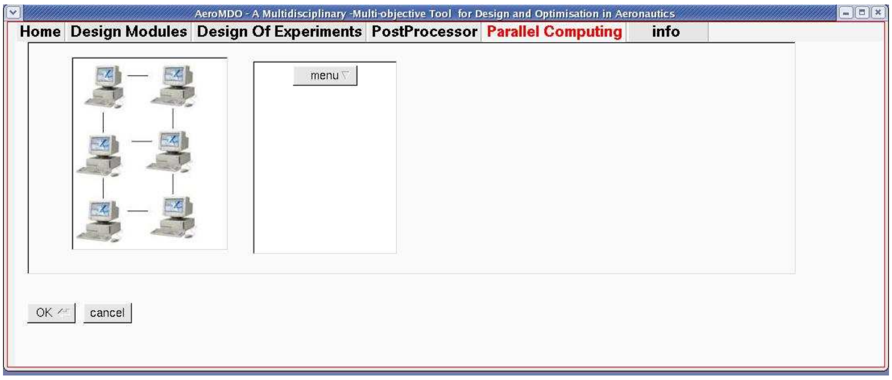

Figure 5.1: Parallel computing module.

5.3 Basic Formulation of an Optimisation Problem

When considering the solution to a real-world engineering problem, the scientist/engineer usually tries to create a well-posed mathematical formulation which is representative of the problem at hand. The problem is that there are several complexities involved in the process, therefore some assumptions have to be made.

This can be illustrated this with a simple example: the shape of the design and optimisation of an aircraft wing or a wing section as illustrated in Figure 5.2; during the entire aircraft mission the wing is subjected to numerous flight conditions which are characterised by different Mach numbers, Reynolds numbers, angles of attack and other flight parameters. If we want to obtain the optimum shape for this wing, the design has to have good characteristics with regard to the aircraft payload, total mass, aerodynamic and structural performance. This is a complex task, therefore a sound solution to the problem is to make some assumptions and identify the main flight conditions, design parameters, constraints and then construct an optimisation problem in order to improve the performance.

- **Design Parameters and Shape Parameterisation:** The first consideration is the design variables and shape parameterisation of the blade contour by Bézier or Spline curves, for example.
- **Geometric Constraints:** Secondly, we need to specify a series of constraints, such as maximum and minimum thickness location, pitching moment and lower and upper bounds for the Bézier/Spline control points.
- **Aerodynamic/Structural Analysis:** Then the type of analysis is considered. The aerodynamic characteristics of the blade, for example, can be obtained by solution of the

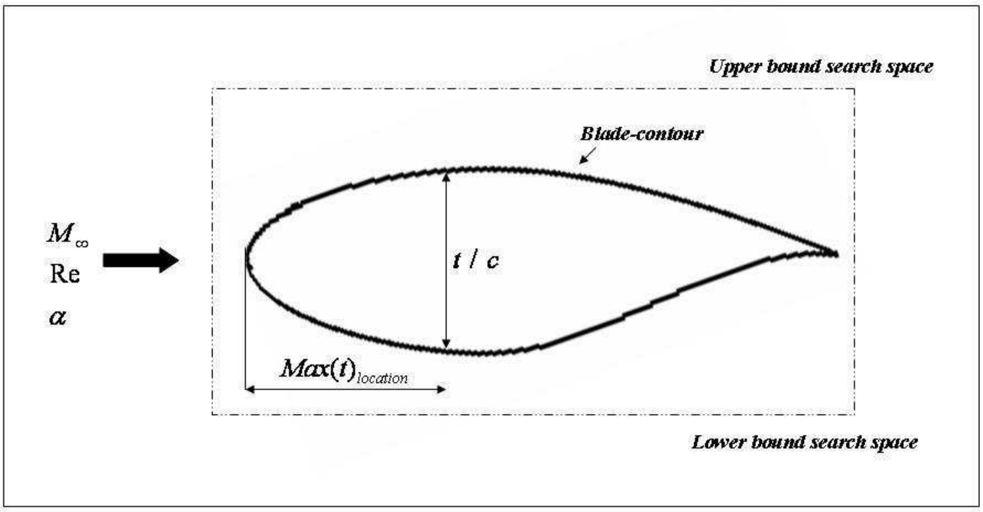

Figure 5.2: Two-dimensional section of an aerofoil wing.

Navier-Stokes, Euler equations, potential flow solver or using panel methods. The structural properties can be computed with a FEA or an analytical method.

In general, a shape optimisation problem can be considered as a constrained minimisation problem of a cost function that involves the evaluation of a series of partial differential equations on a parameter-dependent geometrical domain. The basic problem consists in finding the optimal shape in a search space of feasible shapes while satisfying non-linear design constraints. This optimum shape then has to minimise an objective or multiple objectives that depend on the selected shape and that conform to the solution of the partial differential equations corresponding to the disciplinary or multidisciplinary analysis. In general, the problem can be defined as:

$$Optimisation\ Problem\ P = \left\{ \begin{array}{c} Find\ a\ Shape\ \Omega^*\ such\ that \\ j\ (\ \Omega^*) = min\ j\ (\ \Omega) = min\ J\ (\ \Omega,\ W_\Omega) \\ under\ the\ constraint\ S\ (\ \Omega) \leq \varepsilon_o, \\ and\ with\ the\ state\ W_\Omega\ solution\ of\ the\ equation \\ f\ (\ \Omega,\ W_\Omega) = 0 \end{array} \right\}$$

where:

- Ω denotes the unknown shape and is a subset of $\Re^3$: Ω is defined in general by a finite set of parameters $z \in \Re^3$ called the design variables,
- W_Ω are the state variables that characterise the state of the system under study. In general, W_Ω is the unique solution of a set of partial differential equations.

- $j(\Omega) = j(\Omega, W)$ is the objective or cost function which must be minimised.
- $S(\Omega)$ defines the set of design constraints imposed on the shape.

The application of this formulation to aerofoil design problems is illustrated in the following sections and Chapter 6. Extension and applications of this concept to wing and aircraft design and optimisation are presented in Chapters 7 and 8.

5.4 Test Cases on Parallel Evolutionary Algorithms and Hierarchical Topology of EAs

5.4.1 Introduction

The purpose of this section is to evaluate the methodologies and algorithms with parallel computing and hierarchical topologies of solvers. Different test cases are considered which highlight the robustness of the method to find optimal solutions. These studies provide guidelines on the complexities involved when combining different flow-solvers. Initially the shape parametrisation will be discussed, next the different flow-solvers used in these studies are described, then a series of test cases is analysed. In these problems we use the methods and algorithms in the aerofoil design module (Figure 5.3).

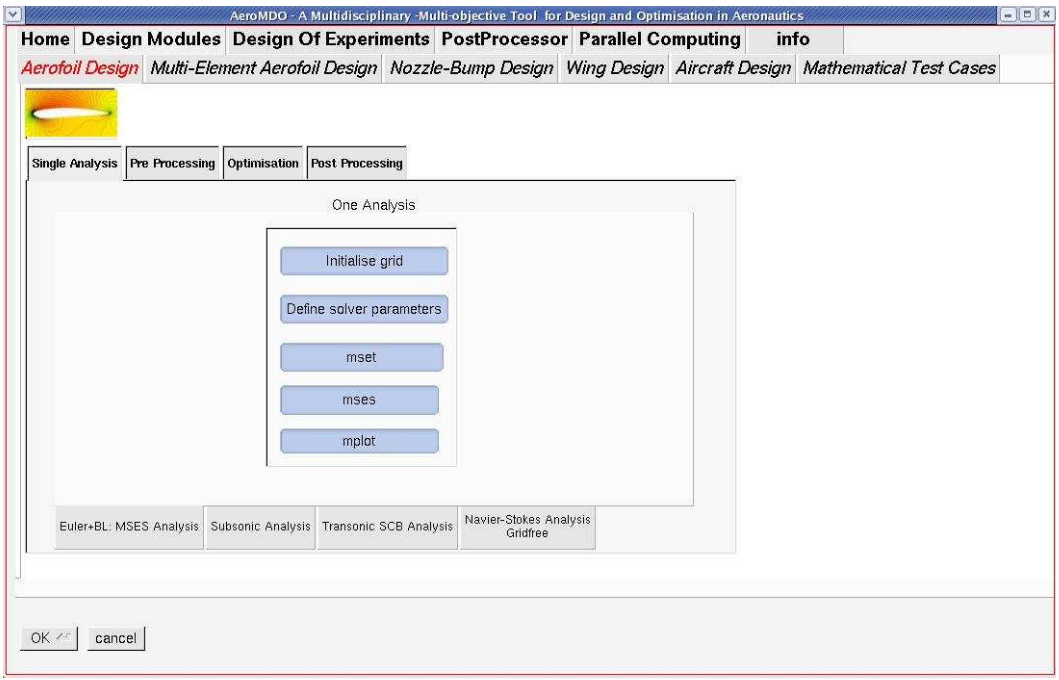

Figure 5.3: Aerofoil design module.

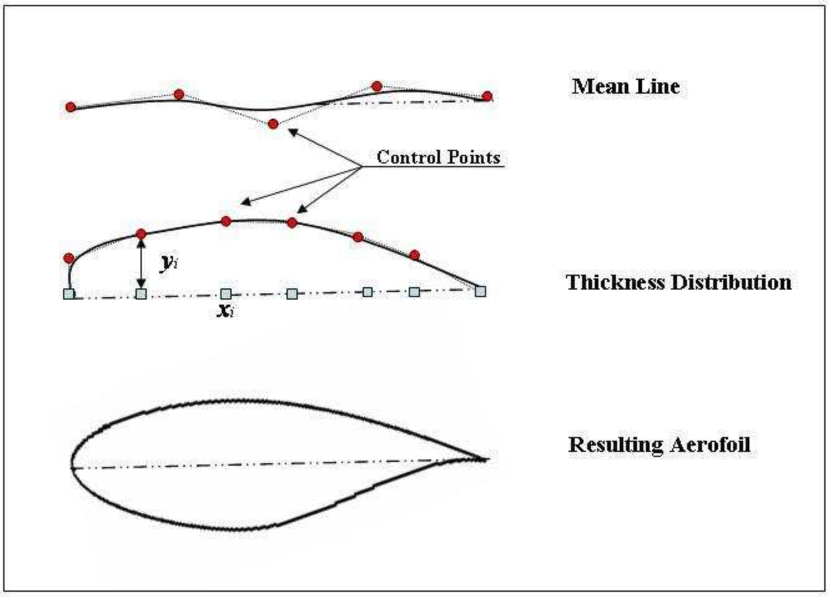

Figure 5.4: Bézier curve representation for aerofoil design. Mean line and thickness distribution.

5.4.2 Shape Parameterisation

One of the main difficulties in EA is an adequate representation of the problem. In this case the mapping of $X(z)$ is introduced by constructing a grid as a function of control parameters z. This will focus on a two-dimensional description; the same principles apply to three-dimensional problems. The aerofoil is represented by two Bézier curves; one for the mean line and one for the thickness distribution, which is a very common concept in classical aerodynamics [1]. Both lines are represented by Bézier curves with leading and trailing edge points fixed at $(0.0, 0.0)$ and $(1.0, 1.0)$ respectively, and a variable number of intermediate control points whose $x - ordinates$ are fixed in advance and whose $y - abscissas$ form the problem unknowns. This concept is illustrated in Figure 5.4

5.4.3 Flow Solvers

Three different fidelity flow-solvers are considered in these studies: A Navier-Stokes solver –*NSC2Ke* [110], a panel method solver –*XFOIL* [47] and an Euler + Boundary Layer solver –*MSES* [46]. These codes were validated standard on test cases in order to gain an understanding of the convergence properties and confidence in its implementation and coupling with the framework, the GUI and optimisation algorithm.

5.4.3.1 Navier-Stokes/Euler Solver (NSC2Ke)

The NSC2Ke is a Navier-Stokes solver developed by B. Mohammadi. *NSC2Ke* is a Finite-Volume Galerkin program computing 2D and axisymmetric flows on unstructured meshes that has capabilities for viscous or Euler flow. The maximum magnitude of the residual considered in all these problems is set to 10^{-4}. Additional details on *NSC2Ke* can be found in Mohammadi [110].

5.4.3.2 Euler + Boundary Layer Solver (MSES).

The solver is based on a structured quadrilateral streamline mesh which is coupled to an integral boundary layer based on a multi-layer velocity profile representation. Details on *MSES* can be found in Drela [46].

5.4.3.3 Panel Method Solver (XFOIL)

The *XFOIL* [47] solver comprises a higher order panel method with a coupled integral boundary layer. In these studies a free transition point for the boundary layer is allowed. Details on *XFOIL* can be found in Drela [47].

5.4.4 Initial Test Cases – Inverse Design

Practising engineers in aerodynamics design often know an appropriate pressure distribution for their problems, but do not always have tools for finding a feasible solution. This problem has been studied extensively by Dulikravich [48], Jameson [75, 78], Kim and Rho [81], Klein and Sobieczky [83] and Obayashi and Takanashi [121] amongst many others.

In this section, some results on the application of the evolutionary methods are presented to find either correct or compromised solutions to multi-objective inverse viscous aerodynamic problems.

Three examples of inverse viscous design, including subsonic and mixed subsonic/transonic design points solved with Euler+boundary layer flow analysis are presented. In the following problems, the same design variables, constraints and fitness functions are used, and these are:

5.4.4.1 Design Variables

The aerofoil geometry in all cases is represented by the combination of a mean line and thickness distribution. We use four free control points on the mean line and five free control points on the thickness distribution.

5.4.4.2 Fitness Function

The objective function is defined as minimisation of the area difference between computed pressure distributions and the one on the target aerofoils.

$$f_1 = \min\ (A_{candidate} - A_{taget}) \tag{5.1}$$

where A indicates the area under the pressure coefficient distribution graph.

5.4.4.3 Implementation

Details on parameter settings for the optimisation algorithm are:

Top Layer: A population size of 40, intermediate recombination used between two parents, and a mesh of 215 $\times$ 36.

Middle Layer: A population size of 20, discrete recombination used between two parents, and a mesh of 165 $\times$ 27.

Bottom Layer: A population size of 20, discrete recombination used between two parents, and a maximum of of 151$\times$ 24.

5.4.4.4 Test Case I: Single-objective Aerofoil Pressure Reconstruction

This test case considers a single-objective problem: the reconstruction of the pressure distribution over a NACA0012 operating at $M = 0.2$, $Re\ = 2.7\ \times 10^6$, $\ Angle\, of\ Attack\ = 1.25\, deg$.

Computational Results

Figure 5.5 shows the convergence rate on the fitness function for this problem. This case was run for 2500 function evaluations of the head node, and took approximately six hours on the BORGS cluster with ten machines. As illustrated in Figures 5.6 and 5.7, there is a good agreement between the target and final geometry.

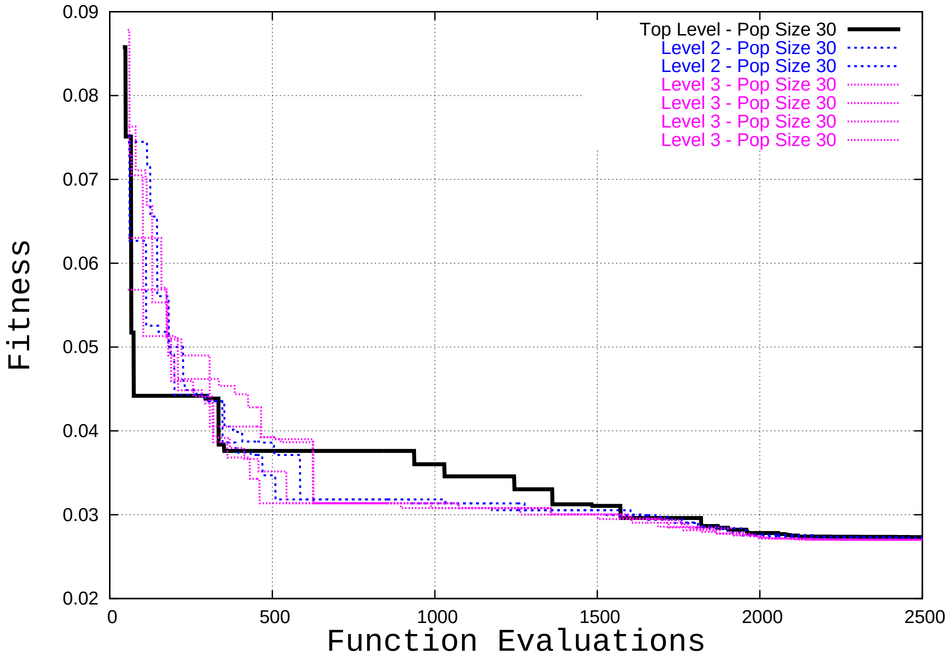

Figure 5.5: Evolution progress for single-objective aerofoil pressure coefficient distribution reconstruction problem.

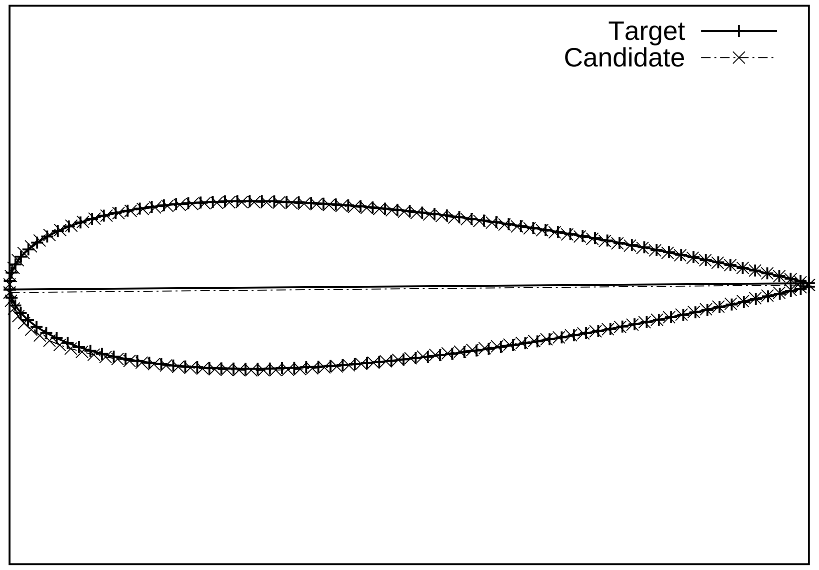

Figure 5.6: Target and computed geometries for single-objective aerofoil pressure coefficient distribution reconstruction problem.

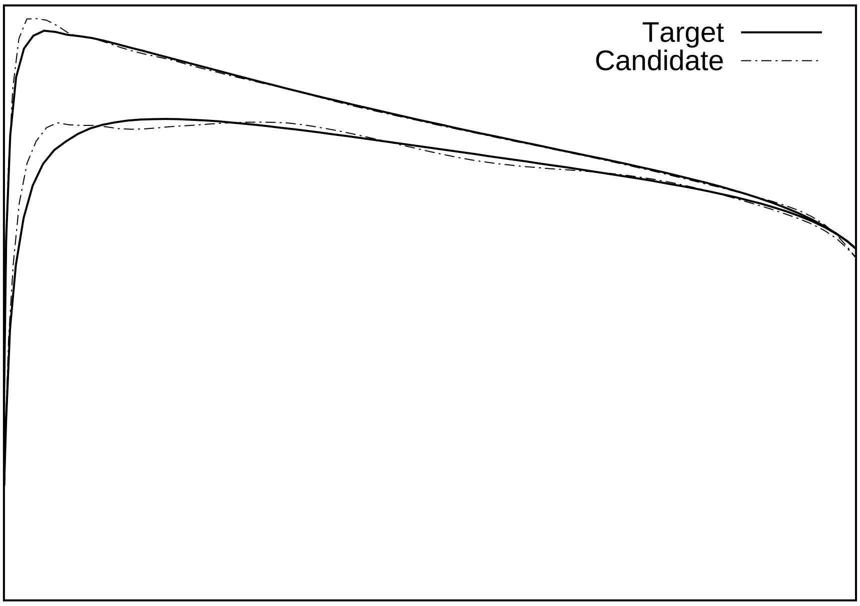

Figure 5.7: Target and computed pressure distribution for single-objective aerofoil pressure coefficient distribution reconstruction problem.

5.4.4.5 Test Case II: Aerofoil Operating at Two Design Points

This case considers the reconstruction of pressure distribution over a NACA0012 aerofoil operating at two different design points. Design point one corresponds to flow conditions $M = 0.2$, $Re = 2.7 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$ and design point two corresponds to flow conditions $M = 0.75$, $Re = 9.0 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$.

The target pressure distributions are illustrated in figure 5.8. For illustration purposes Figures 5.9 and 5.10 show the pressure contours under these flow conditions. In this case it is clear that the final Pareto front should collapse to a single point which matches the NACA0012 geometry.

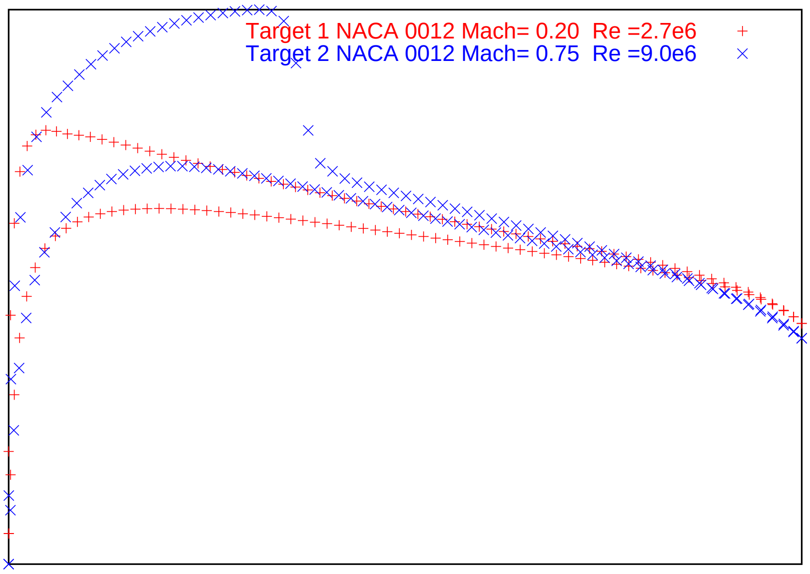

Figure 5.8: Target pressure distributions over NACA 0012 at two flow conditions: $M = 0.2$, $Re = 2.7 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$ and $M = 0.75$, $Re = 9.0 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$.

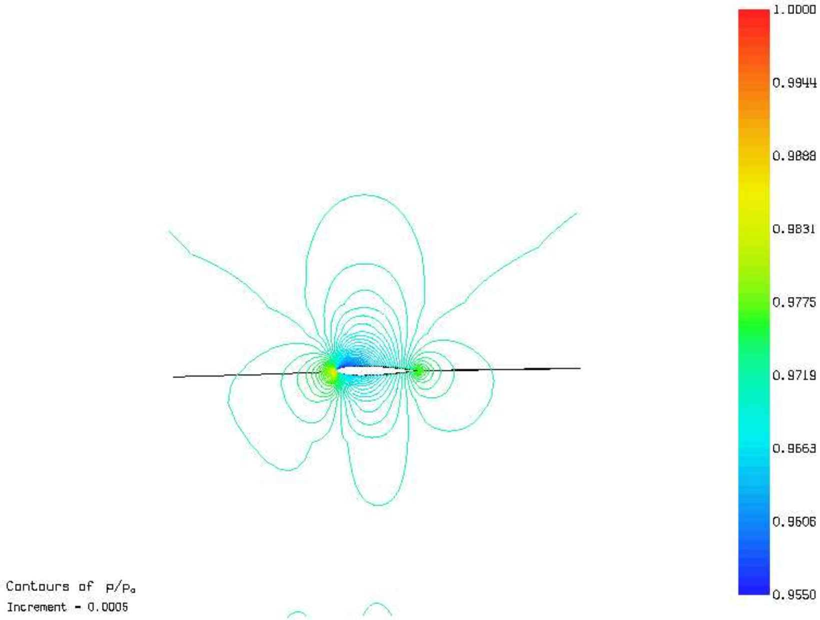

Figure 5.9: Target pressure contours distributions over NACA 0012 at flow conditions: $M = 0.2$, $Re = 2.7 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$.

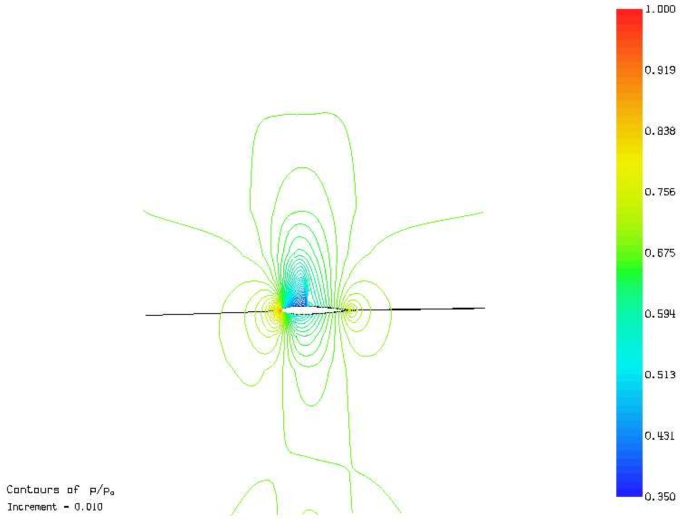

Figure 5.10: Target pressure contours distributions over NACA 0012 at flow conditions: $M = 0.75$, $Re = 9.0 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg.$

Computational Results

This problem was run for 2500 function evaluations of the head node, and took approximately five hours on the BORGS cluster with six machines. The Pareto optimal front for this case is shown in Figure 5.11. As expected, the Pareto front collapses to a single point. As illustrated in Figures 5.12 and 5.13, a good match is found between the target and computed geometries and pressure distributions.

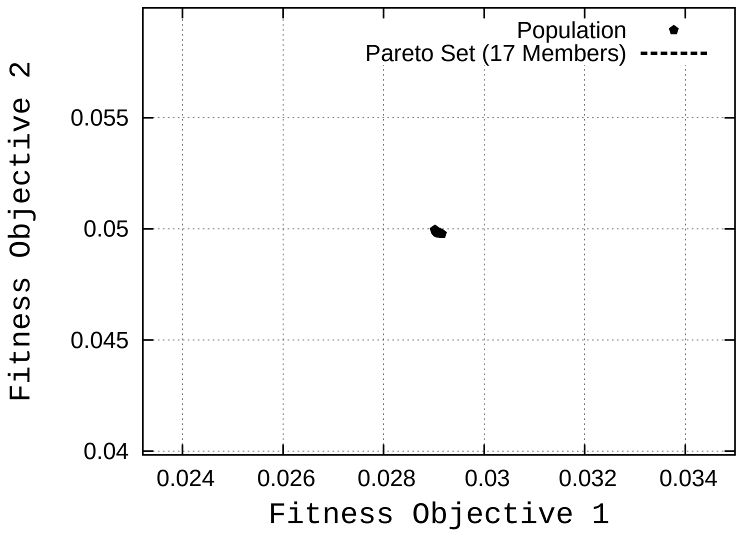

Figure 5.11: Pareto front for aerofoil operating at two flow conditions.

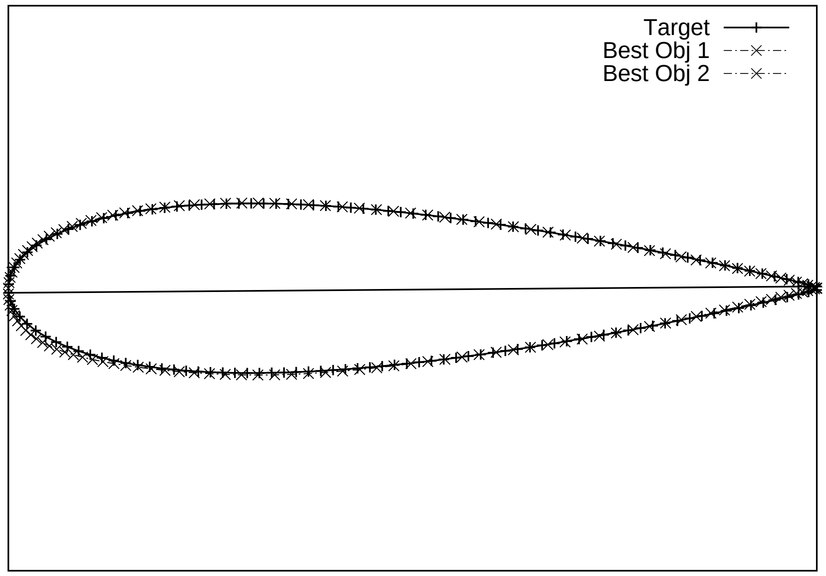

Figure 5.12: Target and computed geometries for aerofoil operating at two flow conditions.

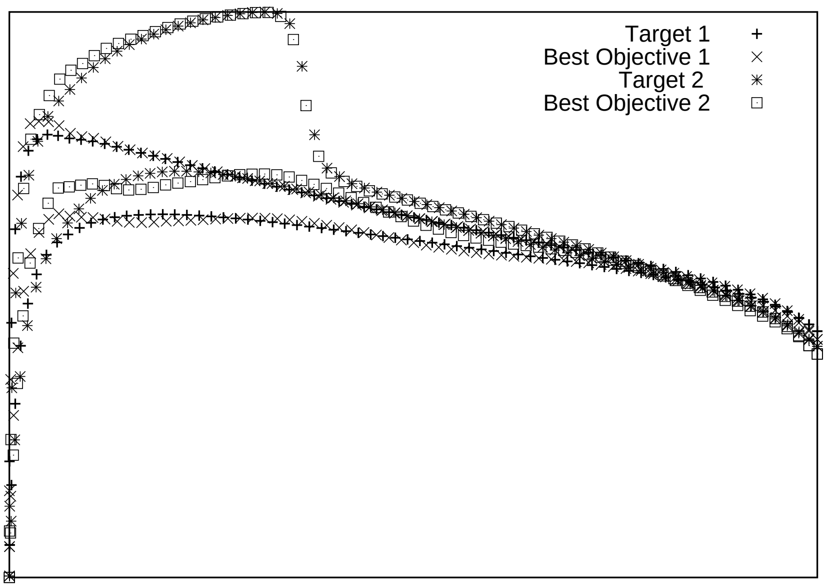

Figure 5.13: Target and computed pressure distribution or aerofoil operating at two flow conditions.

5.4.4.6 Test Case III: Two Aerofoils at Two Different Design Points

This case considers minimisation of the area difference between computed/optimised pressure distributions over two different aerofoils (NACA0012 and RAE2822) at two different design points. The flow conditions for the two points analysed are $M = 0.2$, $Re = 2.7 \times 10^6$, $Angle\ of\ Attack = 1.25\ deg$ (NACA0012) and $M = 0.75$, $Re = 9.0 \times 10^6$, $M = 0.75$, $Angle\ of\ Attack = 1.0\ deg$ (RAE 2822).

5.4.4.7 Computational Results

This problem was run for 3000 function evaluations of the head node, and took approximately ten hours on the BORGS cluster with twelve machines. Figure 5.14 shows the Pareto front obtained for this test case. Figures 5.15, 5.16 and 5.17 show a comparison of the target geometries and surface pressure distributions respectively. As illustrated, there is a good match on the computed and target surface pressure distribution.

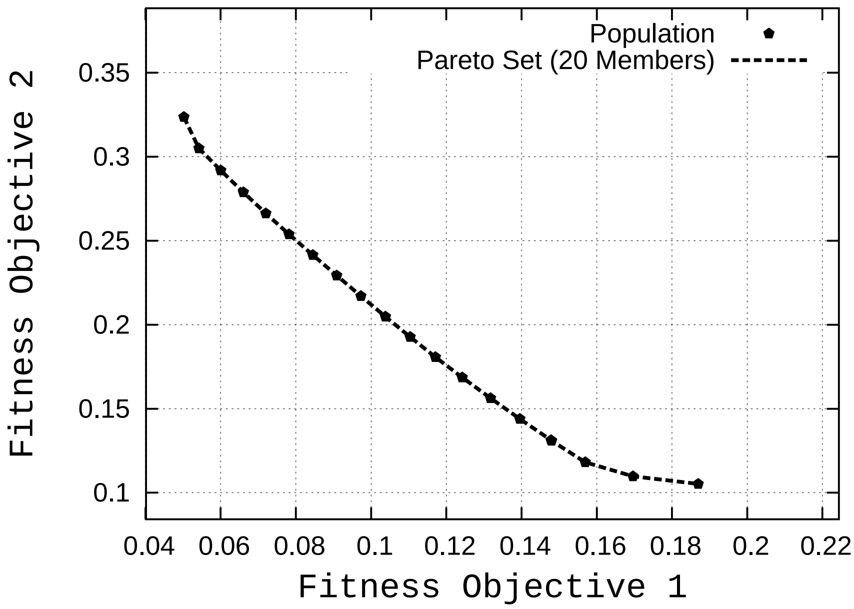

Figure 5.14: Pareto front for multi-point aerofoil design.

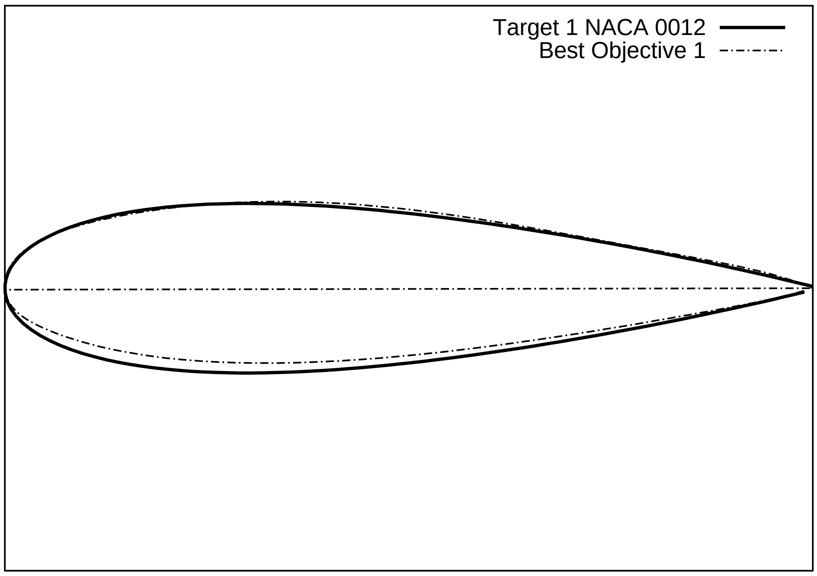

Figure 5.15: Target and computed geometries, multi-point aerofoil design, objective one.

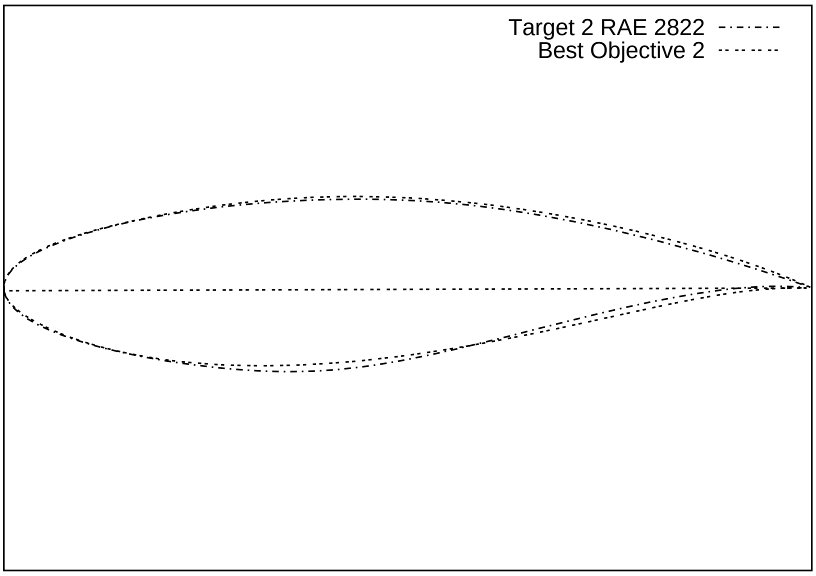

Figure 5.16: Target and computed geometries, multi-point aerofoil design, objective two.

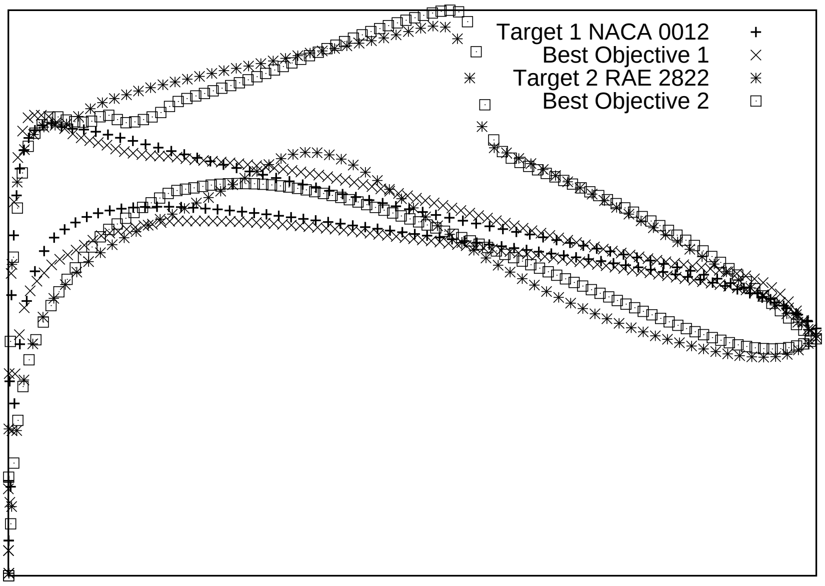

Figure 5.17: Target and computed pressure distribution for multi-point aerofoil design.

For illustration purposes, Figure 5.18 illustrates some additional visualisation capabilities within the framework; the DXF CAD model for the best geometry for objective two; the RAE2822 aerofoil. This model can then be used for further CFD or wind tunnel analysis.

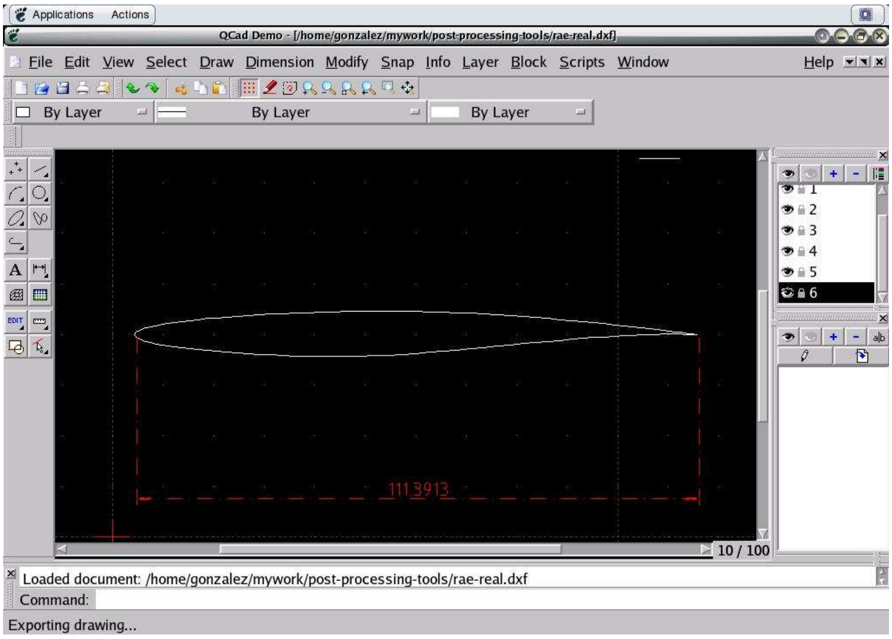

Figure 5.18: DXF model for Pareto optimal aerofoil geometry.

Concluding these test cases, it is shown that, using the evolutionary approach and methods developed, it is possible to capture complex Pareto fronts describing the trade-off between the objectives for inverse problems. The main advantage of these methods is that they can find globally optimum Pareto fronts and that they are integrable with a pre-existing flow-solver without modification or differentiation.

5.4.5 Hierarchical Topology of Solvers

5.4.5.1 Problem Definition

The purpose of this problem is to evaluate the performance of the evolutionary methods with different types of hierarchal topologies of analysis-solvers or resolutions of the computational grid. The problem considers an inverse case with two objectives: minimisation of the difference between computed surface pressure distribution of two pre-defined target aerofoils, as illustrated in Figure 5.19.

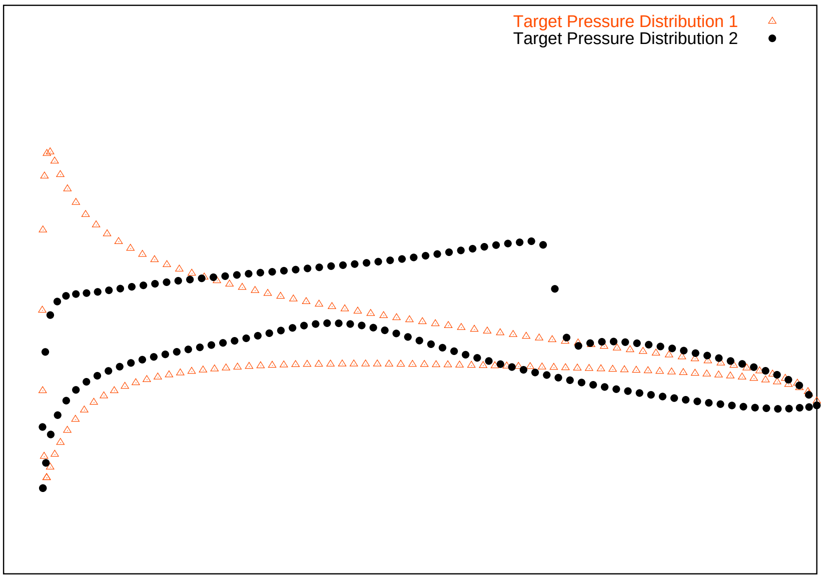

Figure 5.19: Target pressure coefficient distributions.

The two objectives and flow conditions are:

- *Flight Condition 1:* Surface pressure distribution over a typical high-lift aerofoil at subsonic conditions. NACA0012, $M = 0.2$, $Angle\ of\ Attack = 5\ deg$
- *Flight Condition 2:* Surface pressure distribution over a typical transonic aerofoil RAE

2822, $M = 0.75$, $Angle\ of\ Attack = 1\ deg$

$$
\begin{aligned}
f_1 &= \min\left(\frac{1}{N}\Sigma\left(\Sigma_{i=1}^{N}\left(Cp_{candidate} - Cp_{NACA0012}\right)\right) \rightarrow M = 0.2\right. \\
f_2 &= \min\left(\frac{1}{N}\Sigma\left(\Sigma_{i=1}^{N}\left(Cp_{candidate} - Cp_{RAE2822}\right)\right) \rightarrow M = 0.75\right.
\end{aligned}
\qquad (5.2)
$$

This test case is challenging, as the aerofoil geometry and target pressure distribution over these two aerofoils are very different, so possible candidate geometries can vary on a wide search space.

5.4.5.2 Design Variables

The aerofoil geometry is represented by the combination of a mean line and thickness distribution. In this case four free control points on the mean line are taken and five free control points on the thickness distribution. Two different implementations are considered.

5.4.5.3 First Attempt Implementation

The first implementation consists of a hierarchical topology using two different solvers with *NSC2ke* at the top node and *XFOIL* at the intermediate and bottom levels. The following settings are used for the computations:

Top Layer: A population size of 20, intermediate recombination used between two parents, and a maximum of 2500 mesh vertices on the *NSC2Ke* solver.

Middle Layer: A population size of 20, discrete recombination used between two parents, and 129 panels on the *XFOIL* solver.

Bottom Layer: A population size of 20, discrete recombination used between two parents, and 99 panels on the *XFOIL* code.

5.4.5.4 Computational Results

This case was run for 750 function evaluations of the head node, and took approximately eight hours on the BORGS cluster with eighteen machines. The progress on the evaluation of the fitness function evolution is shown in Figure 5.20.

As illustrated, while the panel method solver (*XFOIL*) in levels two and three shows a good convergence rate, the Euler solver finds it difficult to go below the 0.1 residual. This is mainly because while the topological representation of an individual on the panel method does not change, the topology of the grid and number of points representation changes within the Euler evaluation.

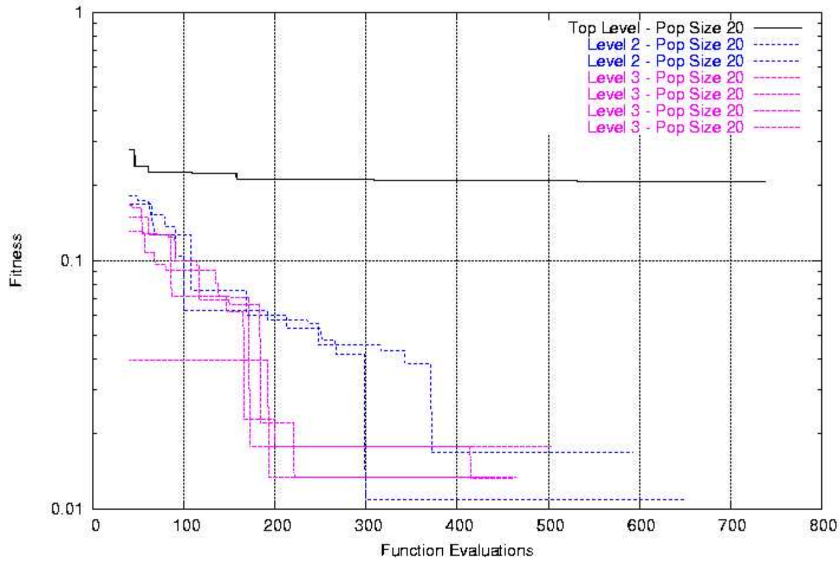

Figure 5.20: Evolution progress for multi-point aerofoil design, first attempt implementation.

These difficulties in the convergence are also due to the re-meshing process along intermediate evaluations of the EA. This re-meshing process is working with a different number of nodes for each candidate individual and generates numerical noise; that is, the finite space where the search for the solution takes place varies within the Euler solver from one candidate solution to another.

This can be further investigated by some numerical experiments starting very close to the solution, imposing a fixed number of nodes for all the meshes, changing the Evolutionary algorithm imposing the same topology. By doing this the local mesh topology close to the body is preserved. To avoid this numerical noise it is necessary then to provide this additional information to all the possible meshes during the optimisation process. This is similar to the structured mesh, which has always the same (topology) close to the body.

Concluding this case, the designer or design team has to do more than use an existing mesh generator in an optimisation process; they have to assess the limitations of the solver to adapt it to the optimisation problem (shape or position or both).

5.4.5.5 Second Attempt Implementation

A second implementation considers a single solver, an Euler + Boundary layer (MSES). The following settings are used:

Top Layer: A population size of 40, intermediate recombination used between two parents, and a mesh of 173 × 36.

Middle Layer: A population size of 20, discrete recombination used between two parents, and a mesh of 163 × 36.

Bottom Layer: A population size of 20, discrete recombination used between two parents, and a maximum of 153 × 36.

5.4.5.6 Computational Results

This case was run for 250 function evaluations of the head node, and took approximately four hours on the BORGS cluster with eighteen machines. The progress of the evolution is shown in Figure 5.21. Figure 5.22 shows a well-distributed Pareto front. Figure 5.23 shows a representative sample of the aerofoil geometries on the Pareto front. A comparison of the pressure distribution for the target geometries is shown in figures 5.24 for NACA0012 and 5.25 for RAE 2822, respectively. A comparison between the target and final geometries for objective one and two is presented in figures 5.26 and 5.27. In this case it can be seen that the problem of local and numerical noise of the first implementation are not present, the re-meshing process occurs but the number of nodes and topology of the mesh is preserved during the optimisation process.

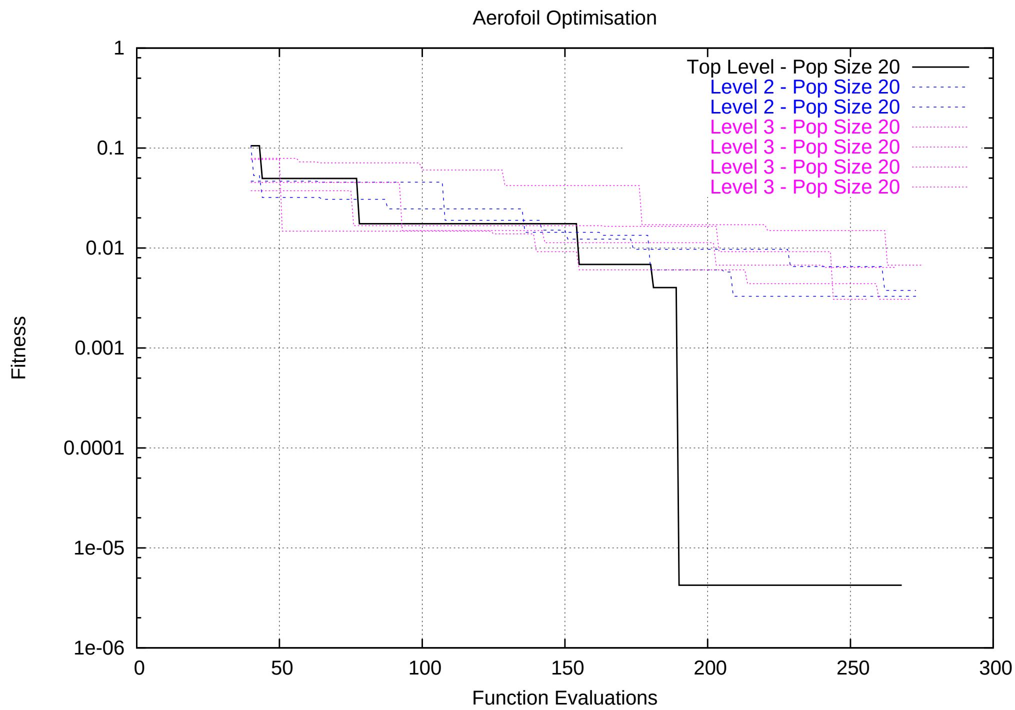

Figure 5.21: Evolution progress for multi-point aerofoil design, second attempt implementation.

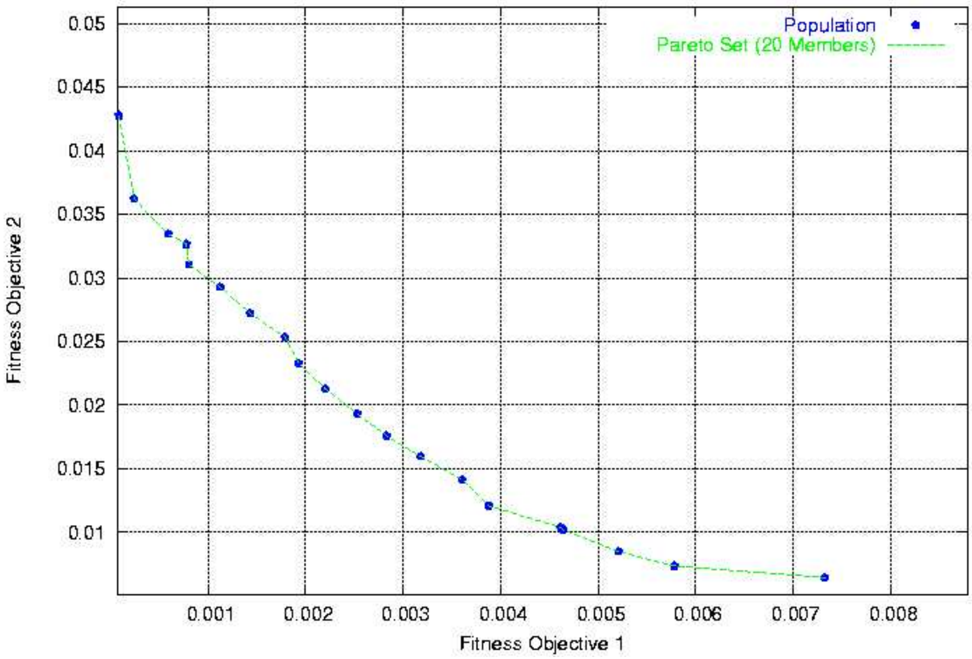

Figure 5.22: Pareto front for multi-point aerofoil design, second attempt implementation.

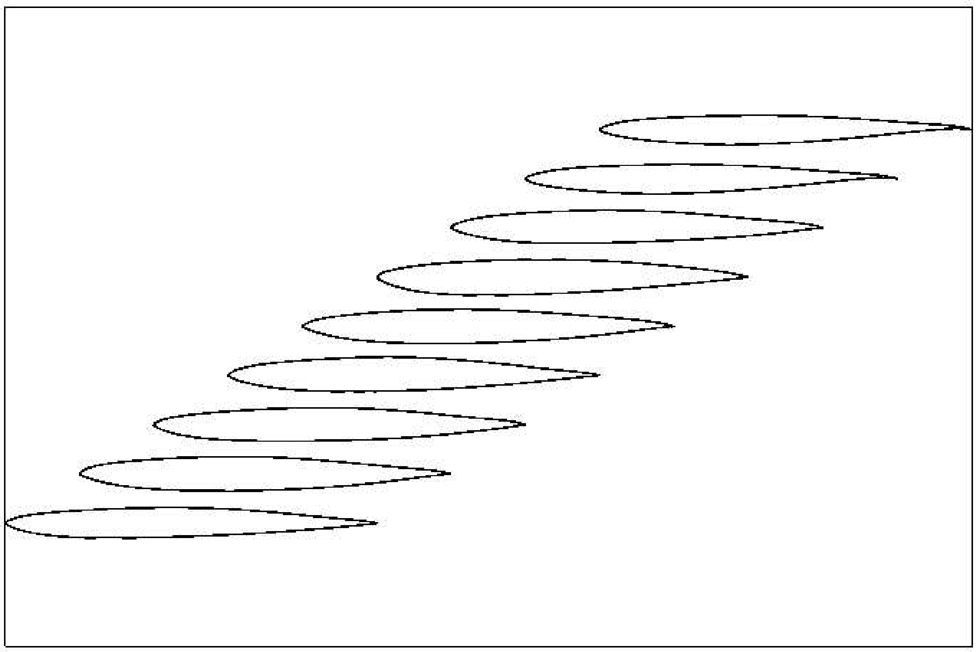

Figure 5.23: Ensemble of aerofoils in Pareto front for multi-point aerofoil design, second attempt implementation.

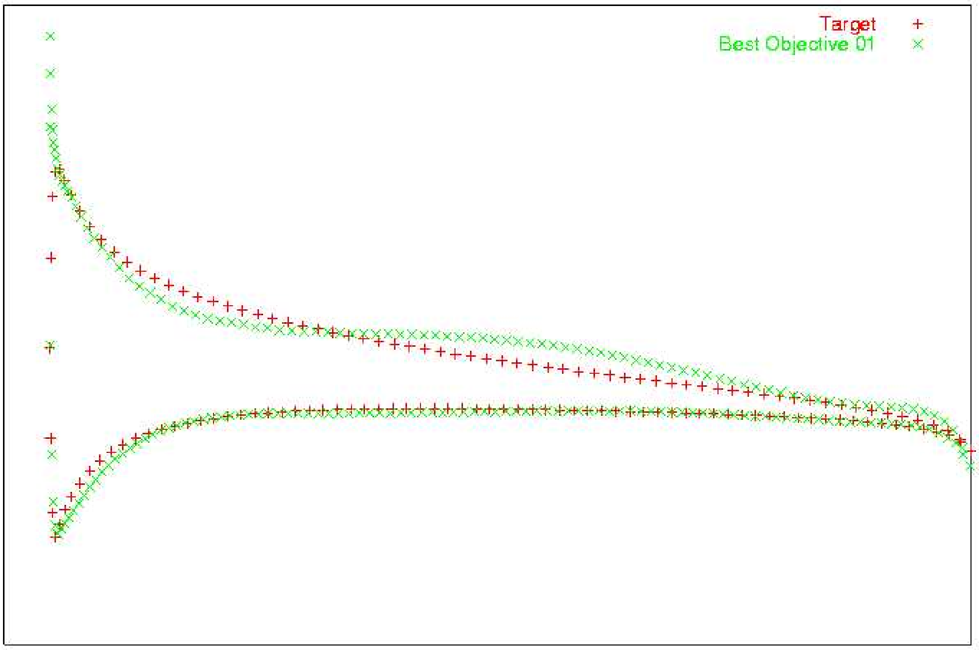

Figure 5.24: Pressure coefficient distribution for objective one.

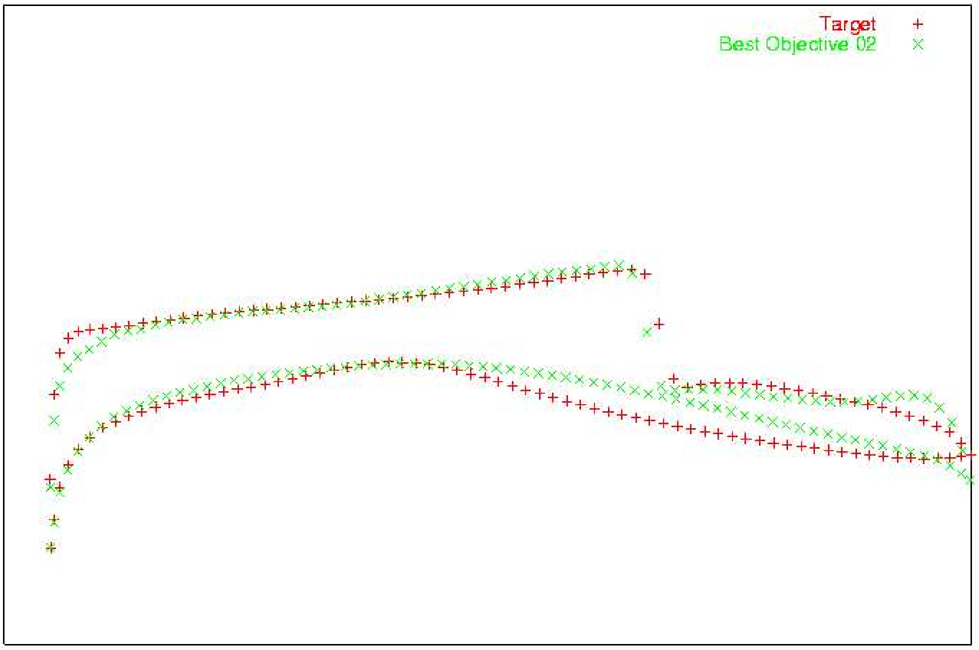

Figure 5.25: Pressure coefficient distribution for objective two.

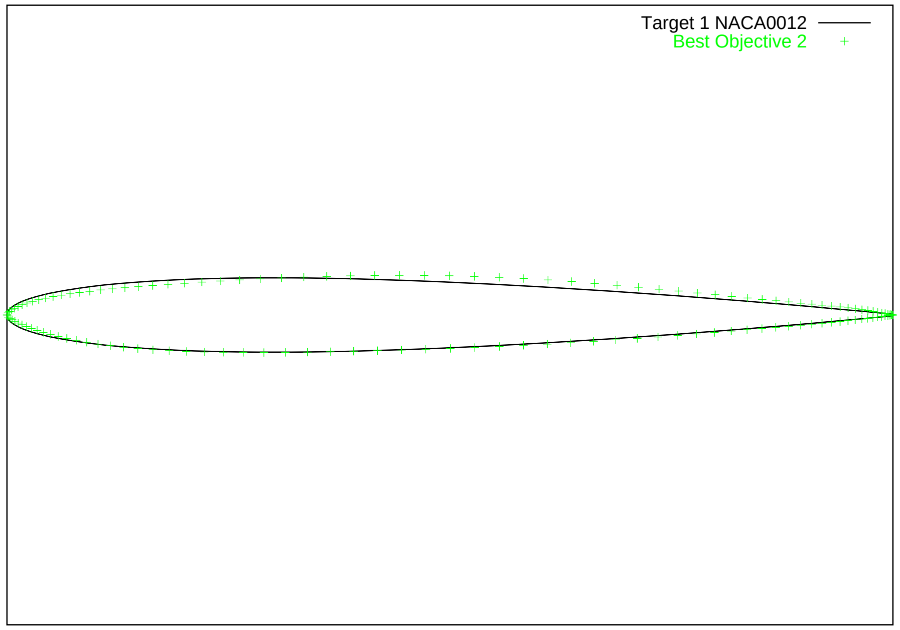

Figure 5.26: Target and optimum geometry for objective one.

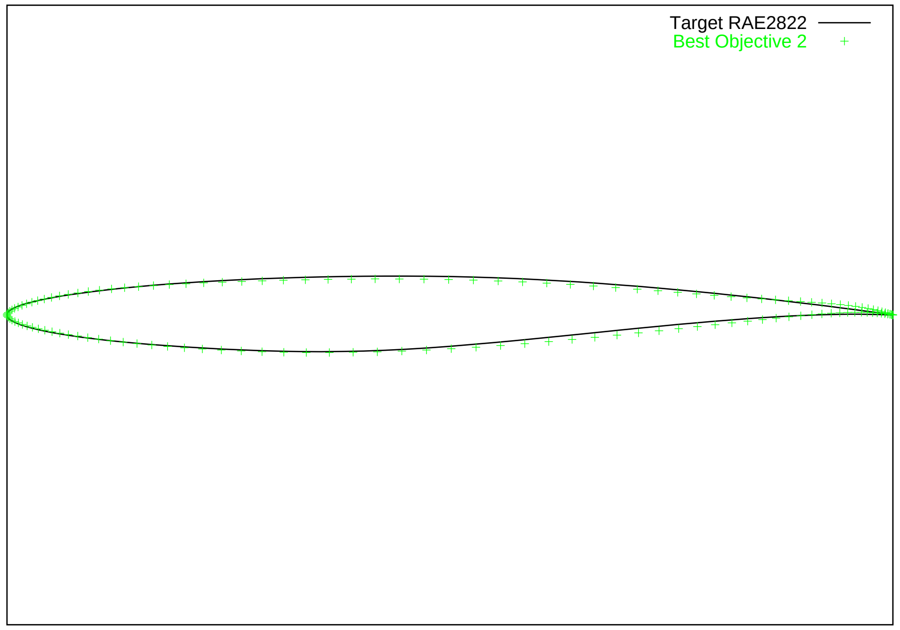

Figure 5.27: Target and optimum geometry for objective two.

5.4.6 Performance with Increasing Number of Computers

This subsection considers the performance of the algorithm with an increasing number of computers. The test case corresponds to that described in the previous sub-section (5.4.5.5). The parallel environment used is the BORGS cluster of PCs. The message-passing model used is the Parallel Virtual Machine (PVM) [55]. Figure 5.28 shows the speed-up of the computation as the number of computes increases and comparison with a linear speed-up for reference purposes.

These test case, clearly show that the methods benefit from the use of parallel computing strategies.

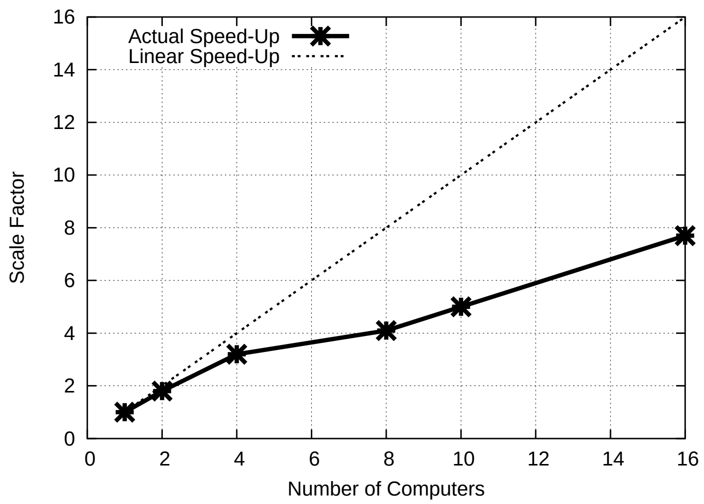

Figure 5.28: Speed-up of the computation with increasing number of computers and comparison with linear speed-up.

5.5 Summary

In this chapter the capabilities and feasibility of the method and its coupling with parallel computing were demonstrated and applied to realistic practical, real-world examples in aeronautics. The importance of appropriate selection of key aspects related to parallelisation strategies, analysis tools, geometry representation and sound engineering judgment was also described. Different studies on inverse aerofoil design, hierarchical topology and paralle computing were conducted.

The objective of conducting studies in parallelisation was to gain an understanding of the complexities, strengths and weaknesses of the proposed methods for problems that might involve higher computational expense. The evolutionary methods seemed to perform well with an increasing number of computers.

The benefits of using parallel computing and evolution algorithms to provide solutions for single- and multi-criteria problems in aeronautics was demonstrated. In this case, the study was limited to a maximum of 16 computers; in order to reach further conclusions it is necessary to extend the studies with higher numbers of computers, in the order of a hundred.

In these chapter it was also observed that issues such as topology representation of an individual and its possible change during the optimisation, the proper selection of a CFD solver and its limitations during the optimisation process, are of great importance. As illustrated, the coupling of a CFD analysis tool with a robust optimiser and the proper selection of a hierarchical topology and parallelisation strategy make the use of EAs more affordable.

Also, it is important to consider that the application of sound engineering judgment in conjunction with evolutionary techniques and parallel computing architectures can lead to optimal design solutions and significant computational savings, when applied to real-world problems. All solutions found herein are 'from scratch' and no initial estimate of the initial geometry was required.

Chapter 6

Aerodynamic Shape Optimisation Applications

"*It is change, continuing change, inevitable change, that is the dominant factor in society today. No sensible decision can be made any longer without taking into account not only the world as it is, but the world as it will be.*" Isaac Asimov.

6.1 Introduction

The objective of this chapter is to ensure the feasibility of the proposed methods for aerodynamic shape optimisation. As indicated in Chapter 2 (Section 2.3.1), aerodynamic shape optimisation is of paramount importance, as even a small drag reduction or improvement in aerodynamic performance can provide savings in fuel and operational costs. As noted in Chapter 4, the evolutionary methods and algorithms are well suited for deceptive, multi-objective, constrained and inverse problems. Also as indicated in Chapter 5, the methods perform well with parallel computing, therefore the methods seem to be robust for application to more complex problems.

In general, the problems considered in this chapter fall into the category of aerodynamic shape optimisation. Section 6.2 considers problems related to single aerofoil design. Section 6.3 illustrates the application to multi-element aerofoil design and Section 6.4 presents a summary of the chapter. Table 6.1 summarises the test cases devised and evaluated in this chapter.

Type	Single-/ Multi-objective	Description	Solver
Direct	Multi-objective	Two Objectives Section Optimisation: A Constraints Definition Study	Euler+Boundary Layer
Inverse	Multi-objective	Multi-Element High-lift Aircraft System Design and Optimisation	Euler

Table 6.1: Aerodynamic shape optimisation test cases.

6.2 Two Objectives UAV Aerofoil Section Optimisation: A Constraints Definition Study

6.2.1 Problem Definition

In this case, the detailed design of a single-element aerofoil for a small UAV application similar to the RQ-7A Shadow 200 Tactical UAV is considered. The aerofoil design module and its evolutionary methods are used for this task. The operating conditions and data are based on reference [111]. The aircraft's maximum gross weight is approximately 320 *Lbs*, it has a wingspan of approximately 12.8 ft, a mean chord of approximately 2 *ft*, length of 11 *ft*, and a planform shape with little to no-sweep. It is assumed that the aircraft is operating at 3000 *m*, between a slow cruise 33.3 *m/s* and fast cruise 46.6 *m/s* approximately. This results in the airframe, flight parameters and operating conditions indicated in Table 6.2. These conditions assume an aircraft at mid weight-cruise during an extended cruise phase at intermediate altitude.

Aerofoil Section	*NACA4415*
Wing Span (aprox), *ft*	12.8
Wing chord (aprox), *ft*	2.0
Length, *ft*	11.2
Cruising altitude, *m*	3000

Flight Condition	Flight Condition One Slow Cruise	Flight Condition Two Fast Cruise
Velocity, *m/s*	33.3	46.6
Mach	0.1025	0.141
Reynolds	$1.085\ x\ 10^6$	$1.490\ x\ 10^6$
C_L	1.18	0.6140

Table 6.2: UAV data and operating conditions.

For the optimisation, an existing aerofoil geometry operating at the two suggested design points, is initially assumed and then an aerofoil that preserves the original thickness while reducing the drag coefficient is designed.

The baseline aerofoil geometry is the *NACA4415*. This aerofoil is *15%* thick. Figures 6.1 and 6.2 show the pressure coefficient (C_p) distribution and some aerodynamic data for the two flight conditions considered. The combined polars for the *NACA4415* aerofoil are shown in Figure 6.3. It is noted that both cruise points operate inside the invariant drag region of the aerofoil; the low speed cruise condition giving approximately $C_d = 0.016$ and the high speed giving approximately $C_d = 0.012$.

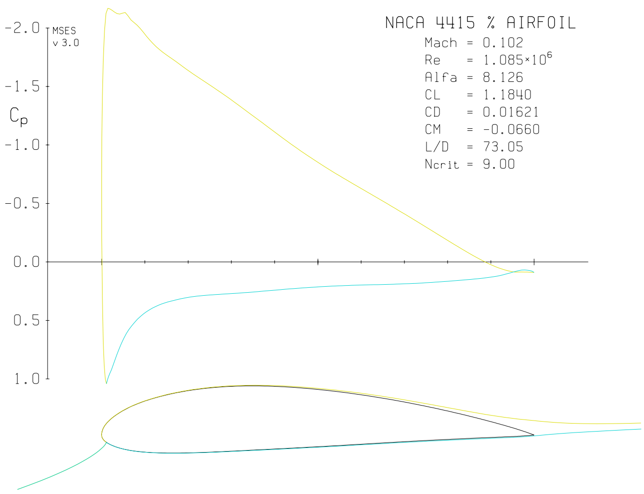

Figure 6.1: Pressure coefficient distribution and aerodynamic data for NACA 4415–flight condition one [Slow-Cruise].

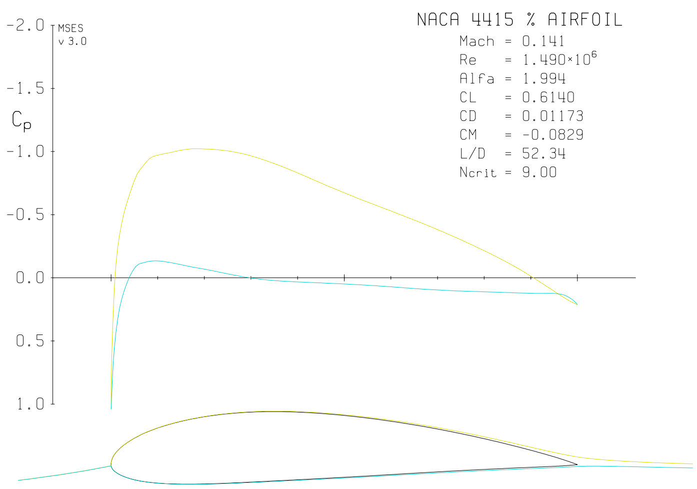

Figure 6.2: Pressure coefficient distribution and aerodynamic data for NACA 4415–flight condition two [Fast-Cruise].

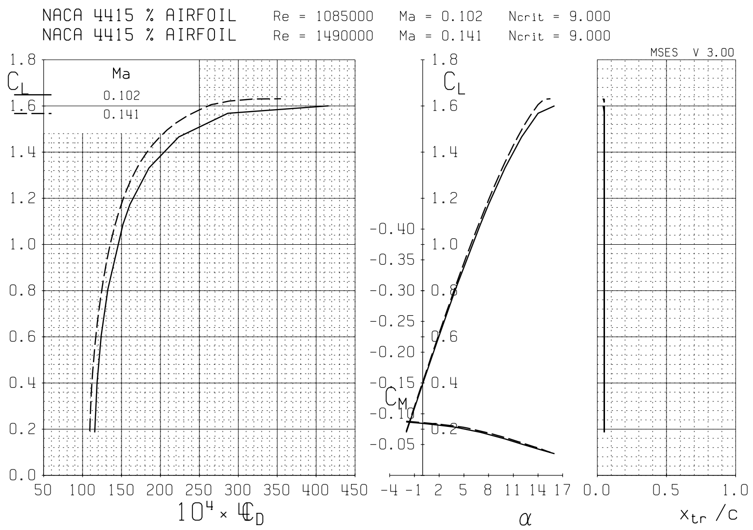

Figure 6.3: Polar computation for NACA 4415 aerofoil.

6.2.2 Design and Optimisation Rationale

In designing a replacement aerofoil for this UAV platform, the following design factors are considered:

- Maintain approximately the same C_L so as not to impinge upon the assisted launch and landing length.
- Maintain at least the current thickness, so as not to increase the weight of the wing.
- Lower the drag at both cruise points, in a multi-objective fashion.
- Study the implication of constraining the pitching moment coefficient during the evolutionary optimisation.

6.2.3 Shape Parameterisation

In the following examples the aerofoil is represented by two Bézier curves, one for the mean line and one for the thickness distribution. In this case six free control points on the mean line and ten free control points on the thickness distribution are considered.

6.2.4 Fitness Functions

The fitness functions for this problem are defined as minimisation of drag (C_d) at the two flight conditions:

$$f_1 = \min\ C_D \rightarrow Re = 1.085\ \times 10^6\ C_L = 1.184\ [Objective\ One] \tag{6.1}$$

$$f_2 = \min\ C_D \rightarrow Re = 1.490\ \times 10^6\ C_L = 0.6140\ \ [Objective\ Two] \tag{6.2}$$

6.2.5 Design Constraints

There are three types of constraints: maximum thickness, maximum thickness location and pitching moment (C_m). The thickness of each aerofoil must exceed *15%* . If a constraint on pitching moment is applied, this must not be more severe than -0.0660 ($C_m \geq -0.0660$). When all constraints are considered, they are added up and applied by equally penalising both fitness values via a linear penalty method.

6.2.6 Solver

The aerodynamic characteristics of the candidate aerofoils are evaluated using the MSES software. This solver is based on a structured quadrilateral streamline mesh which is coupled to an integral boundary layer based on a multi-layer velocity profile representation. Details on *MSES* can be found in Drela [46].

6.2.7 Implementation

Two test cases are run for comparison.

Test Case I – [C_mUnconstrained]: The first case considers only two constraints: minimum thickness and position of maximum thickness.

Test Case II – [C_mConstrained]: The second case considers the three constraints: minimum thickness, position of maximum thickness and pitching moment coefficient.
The method uses the following parameter settings for the evolutionary optimisation algorithm:

Top Layer: A population size of 20, intermediate recombination used between two parents, and a mesh of 215 × 36 on the MSES solver.

Middle Layer: A population size of 20, discrete recombination used between two parents, and a mesh of 165 × 27 on the MSES solver.

6.2.8 Test Case I – [C_mUnconstrained] Results

This problem was run for 2000 function evaluations on the top level. The Pareto front obtained for this case is shown in Figure 6.4. Figure 6.5 illustrates an ensemble plot of all these aerofoils. From this front, three aerofoils are selected: objective one optimal, objective two optimal and compromise aerofoil from the middle of the set. These are shown against the *NACA 4415* aerofoil in figure 6.6. It is evident that the evolved aerofoils have much less camber than the original aerofoil, however, the thickness for all three aerofoils has been maintained at *15%*. From these results a compromise aerofoil is taken from the middle of the Pareto front for further evaluation. Figures 6.7 and 6.8 illustrate the pressure coefficient (Cp) distribution and some aerodynamic data for the two operating conditions considered. The combined polars for this aerofoil are shown in Figure 6.9. It is noted that the evolved aerofoil (compromise aerofoil) has a lower C_m than the original *NACA4415* aerofoil. Figure 6.10 shows the comparative drag polars for $Re = 1.085\ \ x\ 10^6$and Figure 6.11 those for $Re = 1.490\ x\ 10^6$.

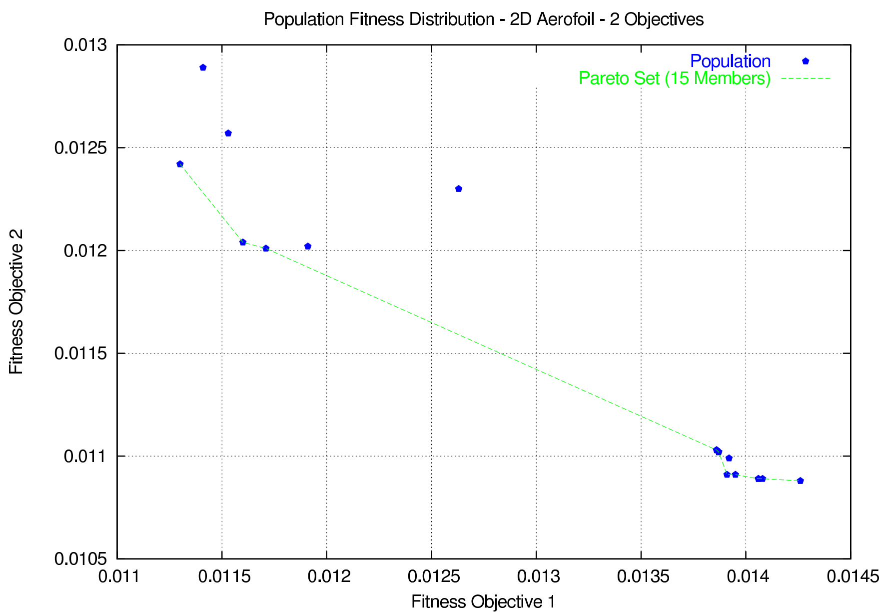

Figure 6.4: Pareto front for the first implementation [C_mUnconstrained].

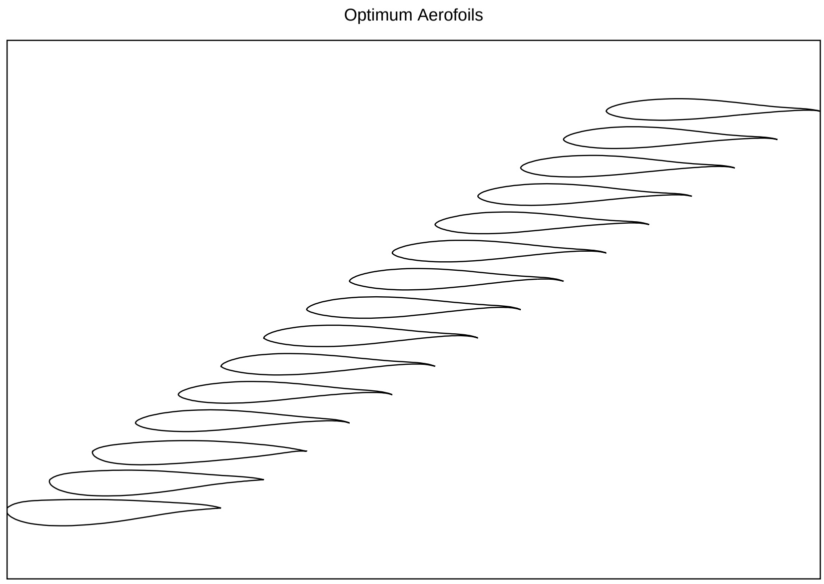

Figure 6.5: Ensemble of aerofoils in Pareto front for the first implementation [C_mUnconstrained].

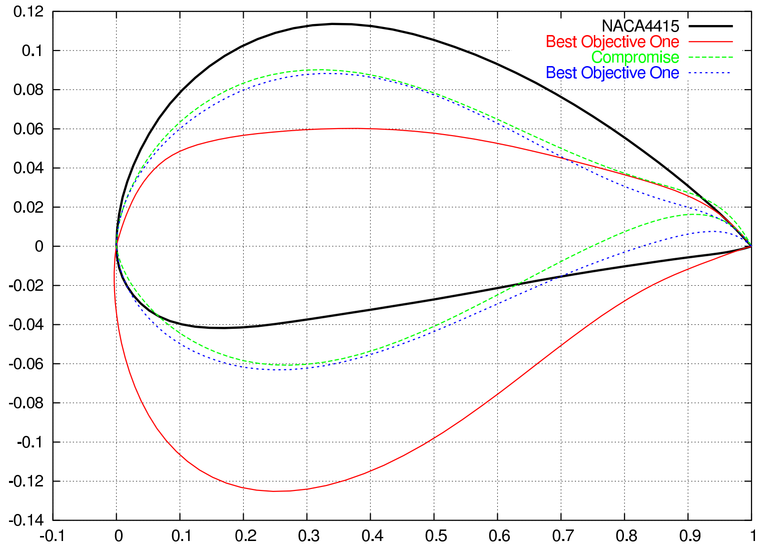

Figure 6.6: Comparison of selected geometries [C_mUnconstrained].

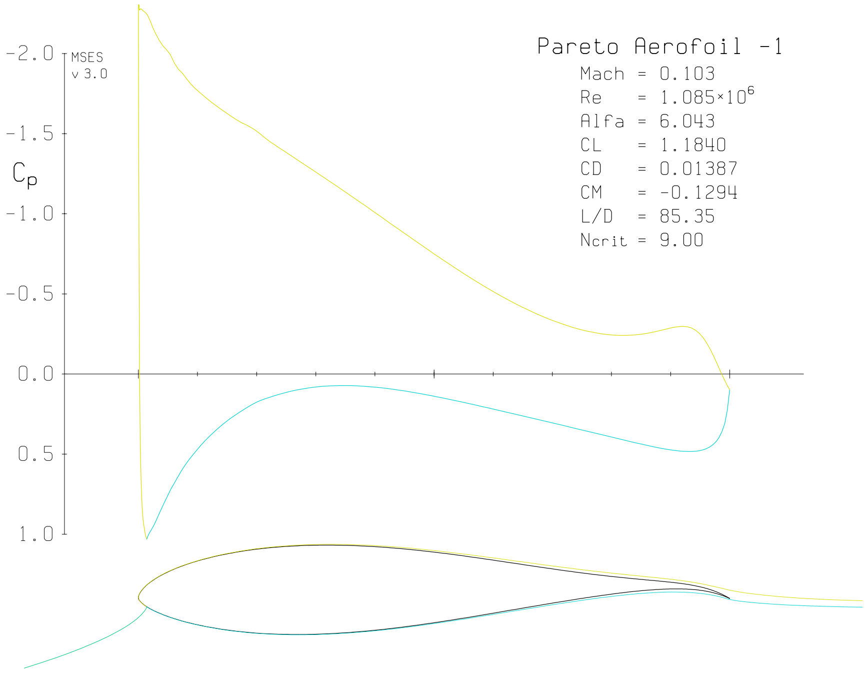

Figure 6.7: Pressure coefficient distribution and aerodynamic data for Pareto 01for flight condition one [Slow-Cruise].

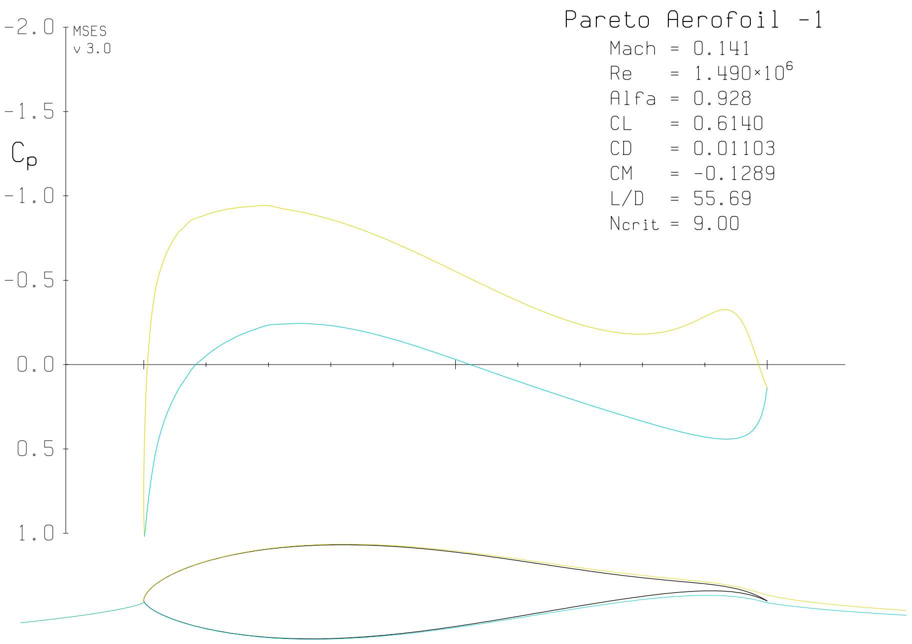

Figure 6.8: Pressure coefficient distribution and aerodynamic data for Pareto 01for flight condition two [Fast-Cruise].

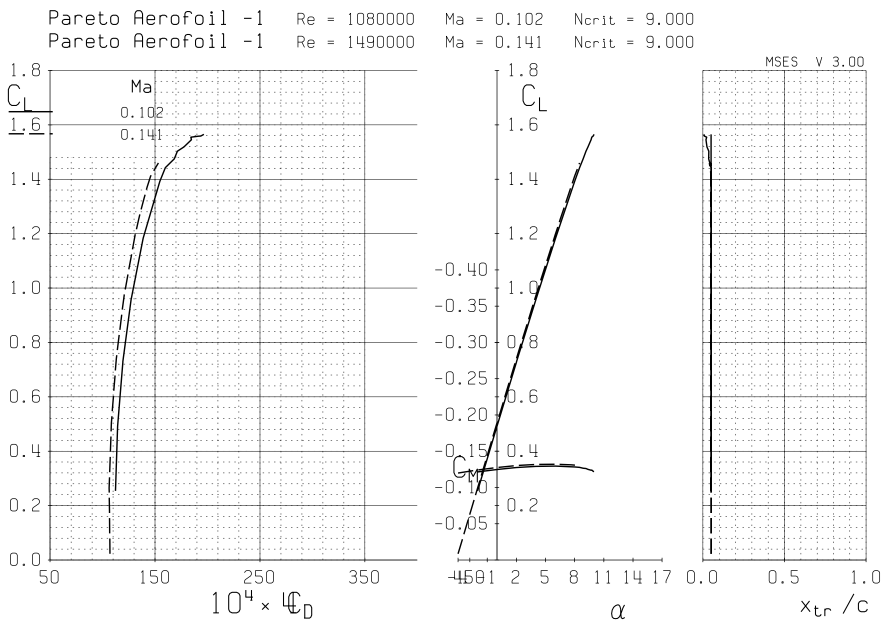

Figure 6.9: Polar computation for Pareto 01 aerofoil [C_mUnconstrained].

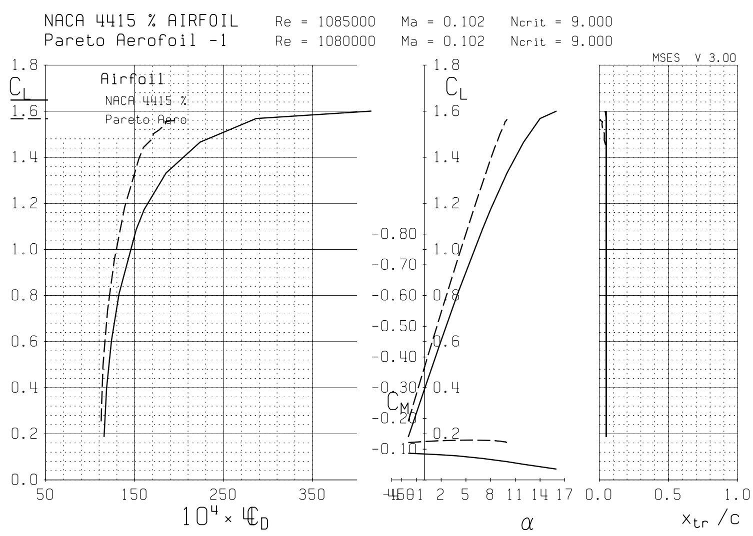

Figure 6.10: Comparative Polars of Pareto 01 and NACA 4415 $Re = 1.085x10^6$ [C_mUnconstrained].

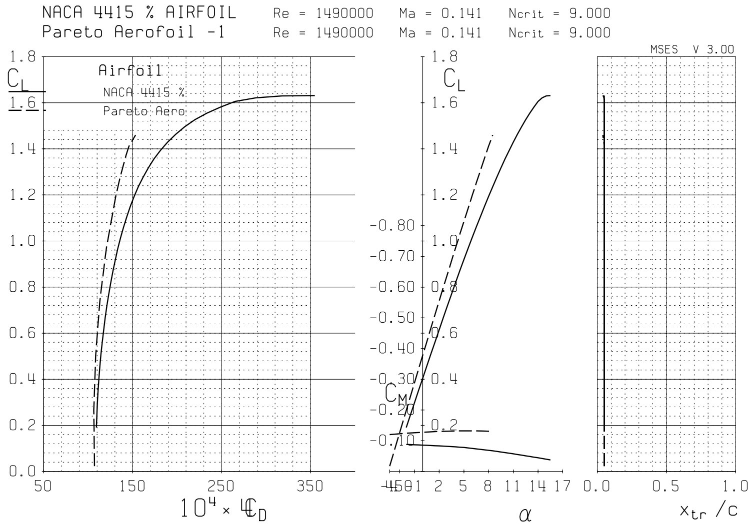

Figure 6.11: Comparative Polars of Pareto 01 and NACA 4415 $Re = 1.490 \; x10^6$ [C_mUnconstrained].

6.2.9 Test Case II – [C_mConstrained] Results

This case was run for the same number of function evaluations as was the previous test case. Figure 6.12 shows the ensemble of aerofoils in the Pareto front. Similar to the unconstrained results, an aerofoil from the middle of the Pareto front is considered for further evaluation. These geometries are shown against the NACA 4415 aerofoil in figure 6.13. Figures 6.14 and 6.15 show the Cp distribution and some aerodynamic data for the two flight conditions. Figure 6.16 shows the combined drag polars at the two operating conditions. Figure 6.17 shows the comparative drag polars for $Re = 1.085 \; x \, 10^6$ and Figure 6.18 that for $Re = 1.490 \, x \, 10^6$. Table 6.3 summarises the drag reduction at the two flight conditions for the two test cases considered.

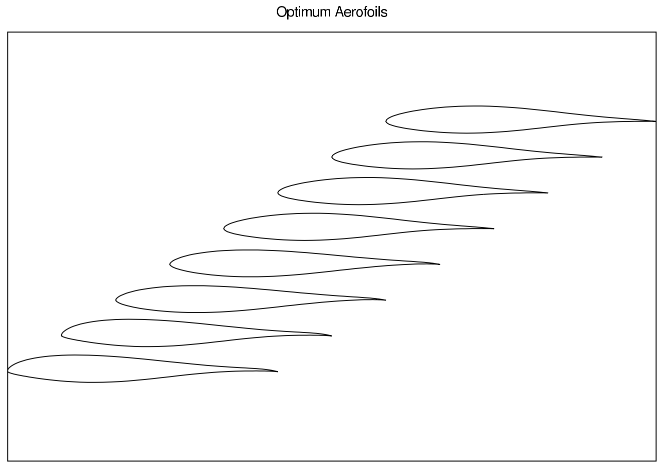

Figure 6.12: Ensemble of aerofoils in Pareto front for second implementation [C_mConstrained].

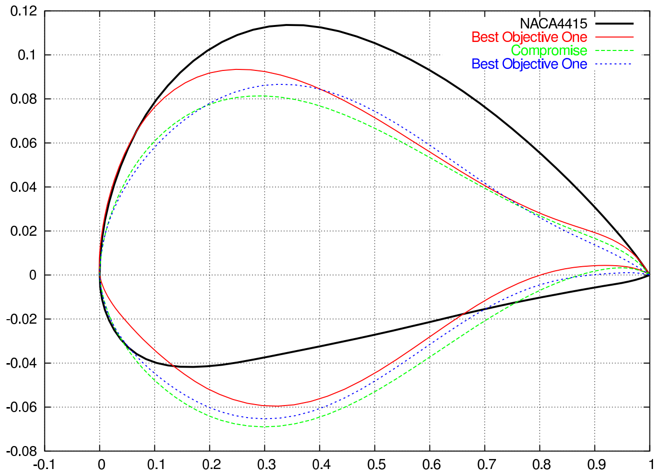

Figure 6.13: Comparison of selected geometries [C_mConstrained].

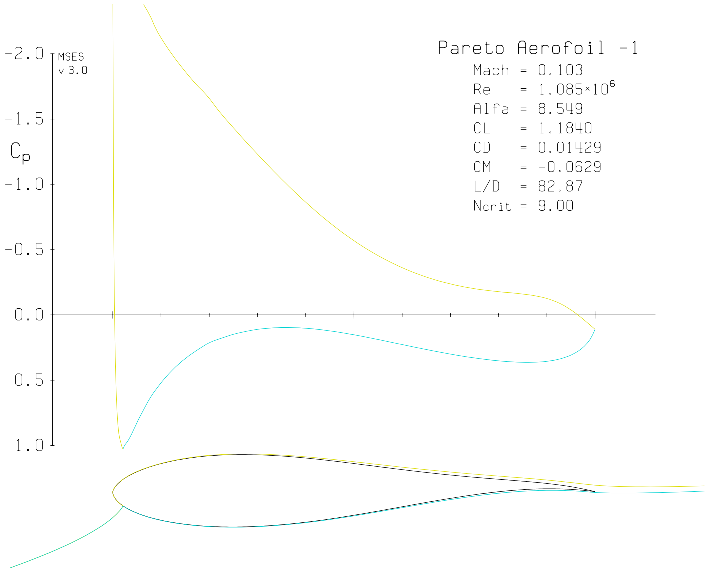

Figure 6.14: Pressure coefficient distribution and aerodynamic data for Pareto 01 for flight condition one [Slow-Cruise] [C_mConstrained].

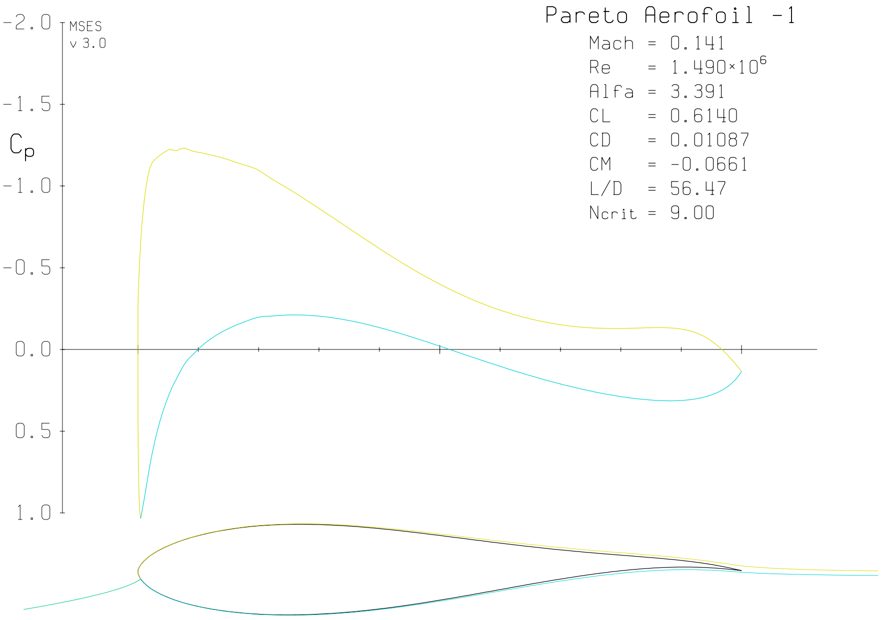

Figure 6.15: Pressure coefficient distribution and aerodynamic data for Pareto 01 for flight condition two [Fast-Cruise] [C_mConstrained].

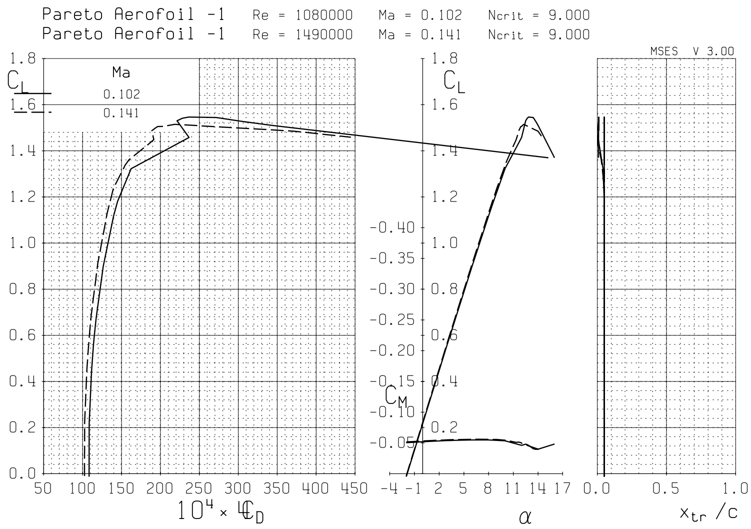

Figure 6.16: Polar computation for Pareto 01 aerofoil [C_mConstrained].

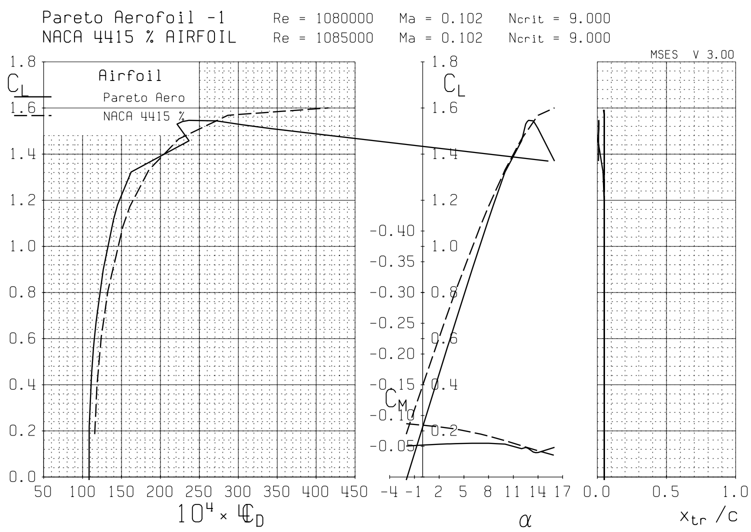

Figure 6.17: Comparative Polars of Pareto 01 and NACA 4415 $Re = 1.085x10^6$ [C_mConstrained].

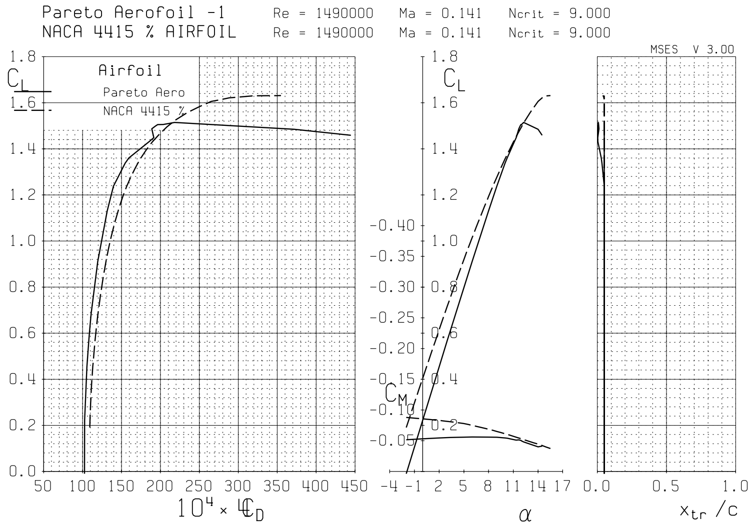

Figure 6.18: Comparative Polars of Pareto 01 and NACA 4415 $Re = 1.490\ x10^6$ [C_mConstrained].

Flight Condition	***NACA4415***	**Compromise Aerofoil** - C_m ***Unconstrained*** [% Gain]	**Compromise Aerofoil** - C_m ***Constrained.*** [% Gain]
Objective One: Low Speed cruise $Re = 1.085x10^6$ $C_L = 1.18$	0.1621	0.01387 [-14.4%]	0.1451 [-10.49%]
Objective Two: Fast speed cruise $Re = 1.49x10^6$ $C_L = 0.0614$	0.01173	0.01102 [-6.0%]	0.01090 [-7.08%]

Table 6.3: Performance gains of optimised aerofoils.

Concluding this case, it is apparent that the evolved aerofoils offer significantly lower drag at both cruise conditions, but with some marked differences on their overall performance and pitching moment coefficient. While both evolved designs produced a rather constant C_m for increasing angle of attack, the requirement of constraining the pitching moment during the evolution process is necessary to avoid obtaining an aerofoil with lower drag for some flight conditions, but with undesirable pitching moment characteristics.

The results obtained show the capabilities of the method to find optimal solutions and classical aerodynamic shapes for flow drag. The importance of sound engineering judgement before, during and after the optimisation cannot be over-emphasised; a proper definition of constraints before performing the evolutionary optimisation and the final results needs to be evaluated to obtain feasible designs.

6.3 Multi-element High-lift Aircraft System Design and Optimisation

The study of high-lift systems is of special interest in aeronautics. High-lift systems consist of a combination of leading and trailing edge devices. The use of leading edge devices (slats) increases the maximum lift of an aerofoil by delaying its stall angle and the trailing edge devices (flaps) are usually designed to produce a lift increment while maintaining a desired high-lift/drag ratio. As expected, the lift coefficient of such a high lift system is a combination of the lift coefficient produced by the interaction of the elements.

The relative position of each aerofoil determines the quality of flow around each element. The accurate computations around these systems is challenging, because the computation of steady viscous flow, maximum lift and pitching moment is complex, especially when there is flow separation. This is a combinatorial problem where the variation in the fitness function is convex and highly non-linear, as the separation point on different configurations can change from one position to another because of the wake/boundary interaction. In this section we study the coupling of the evolutionary algorithm with a Euler solver for aerodynamics shape design and use the multi-element aerofoil design module and its methods for this task (Figure 6.19).

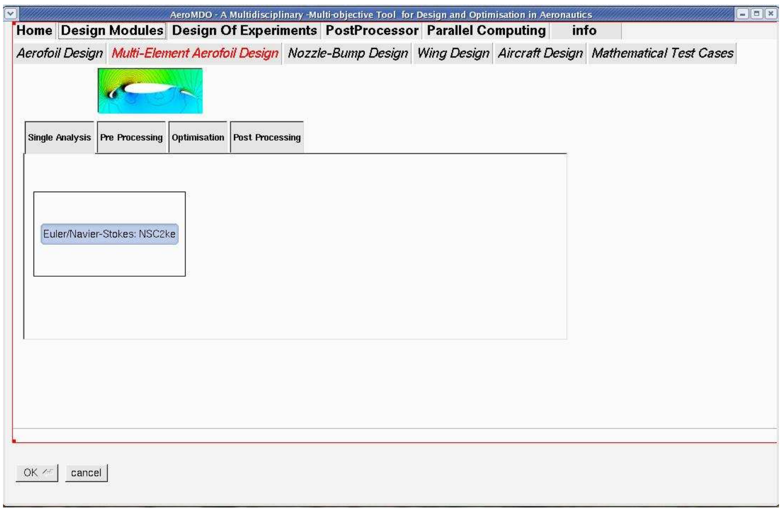

Figure 6.19: Multi-element aerofoil design module.

The multi-element high-lift aerofoil configuration used in this test case is defined as the Dassault: RA16 2D. This test case was selected for the following reasons:

1. The multiple-element aerofoil is a practical configuration of current interest for the aircraft industry.
2. The viscous flow around the aerofoil is complicated by features such as flow separation and the interaction of the wake with the boundary layer.
3. As discussed in Chapter 2, there were only a few previous applications of EAs and MOEAs for multi-element aerofoil optimisation [141].

This configuration is depicted in figure 6.20.

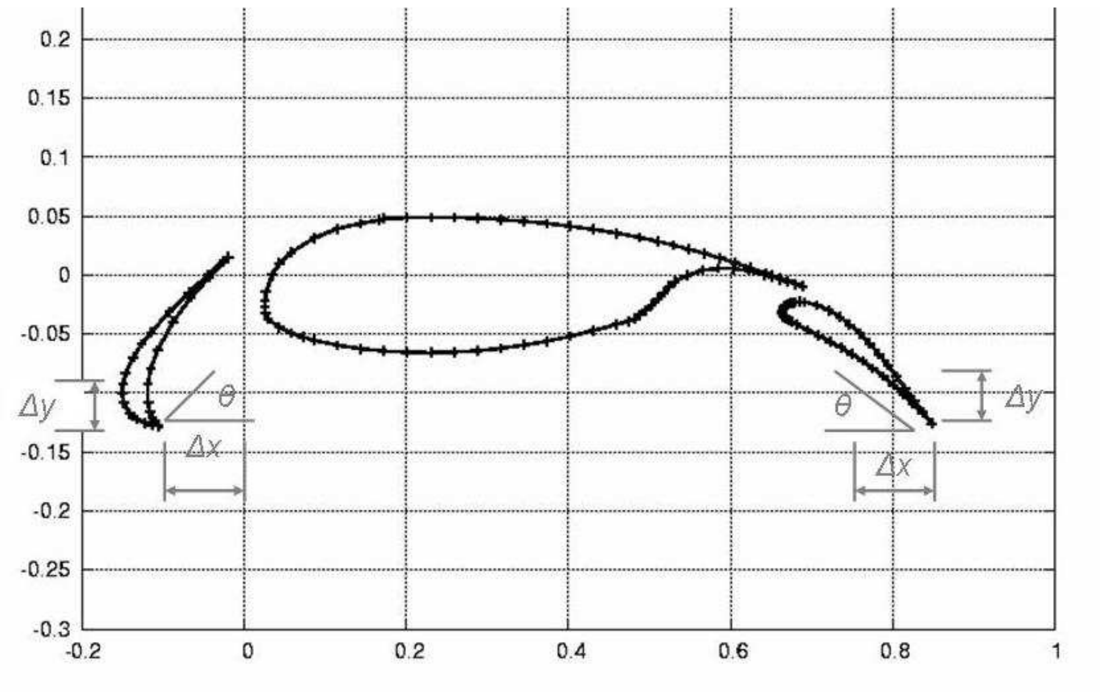

Figure 6.20: Design variables for multi-element aerofoil optimisation.

6.3.1 Design Variables and Fitness Function

The design variables are represented in Figure 6.20. These are the position (Δx, Δy) and rotation (θ) of the slat and flap. The upper and lower bounds of positions and rotations are $\Delta x, \Delta y \pm 0.05$ and rotations of $\theta \pm 30°$ respectively. The algorithm checks and rejects if there is an intersection of the elements in the candidate configuration. The fitness function is the RMS error of the surface pressure coefficients on all three elements. The problem is solved when the fitness goes below a prescribed value.

$$f_1 = \min\left(\frac{1}{N}\Sigma_{elements}\left(\Sigma_{i=1}^{N}\left(Cp_{candidate} - Cp_{target}\right)\right)\right) \tag{6.3}$$

6.3.2 Flow-solver and Mesh Generation/Adaptation

The NSC2ke software developed by Mohammadi [110] is utilised. NSC2KE is a Finite-Volume Galerkin program computing 2D and axisymmetric flows on unstructured meshes that has capabilities for viscous or Euler flow but was restricted in this research to Euler solutions and the magnitude of the residual is set to 10^{-4}. The mesh is generated using the unstructured mesh generator BAMG [70]. For this type of problem it is also important to consider mesh adaptation. The purpose of mesh adaptation is to refine the mesh where the most interesting aspects of the flow occur and appropriately capture the flow phenomena near the aerofoil boundaries. A pseudo-code of the adaptation routine was implemented for this reason a copy can be found in Appendix B.

6.3.3 Two-dimensional One Objective Aircraft High-lift System Design and Optimisation

The present research considers a target pressure distribution reconstruction on a multi-element aerofoil configuration. The flow conditions for this problem are $M = 0.2,\ \alpha = 2\ deg.$

6.3.3.1 Implementation

This method uses a hierarchical topology of evolutionary algorithms and mesh densities with the following settings:

Top Layer: A population size of 40, intermediate recombination used between two parents, and a maximum of 2500 mesh vertices on BAMG.

Middle Layer: A population size of 20, discrete recombination used between two parents, and a maximum of 2000 mesh vertices on BAMG.

Bottom Layer: A population size of 20, discrete recombination used between two parents, and a maximum of 1500 mesh vertices on BAMG.

6.3.3.2 Computational Results

This case was run for 10000 function evaluations of the head node, and took approximately eight hours on a cluster of 18 machines with performances varying between 2.0 and 2.4 GHz. The progress of the evolution is shown in Figure 6.21. Figure 6.22 shows the unstructured grid around one of the candidate geometries during the evolution process. A comparison of the target and final pressure distribution is shown in Figure 6.23 and a comparison between the target and final geometry is presented in Figure 6.24. Figure 6.25 shows the Mach number contours on the target and best geometry and Figure 6.26 that for the pressure contours. There is a good agreement on different pressure and Mach number contour levels.

Concluding this example, it has been shown how parallel computing and coupling of the evolutionary methods with a robust flow-solver using a hierarchical topology provide good solutions for multi-element aerofoil problems.

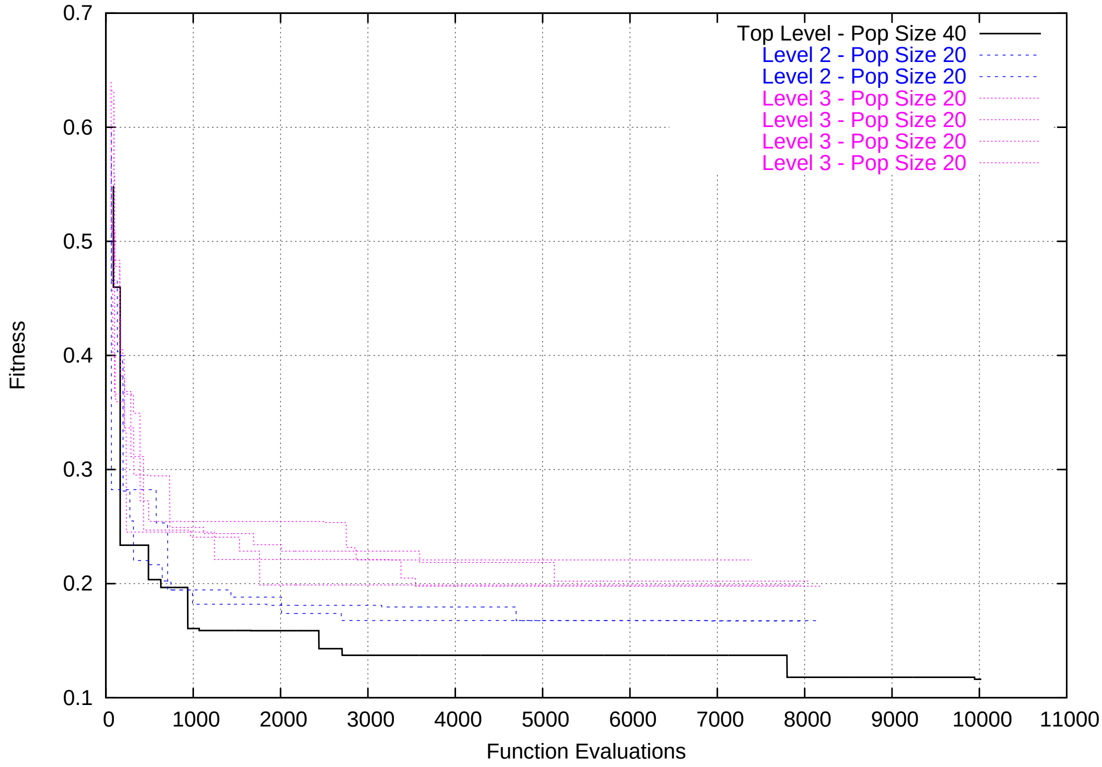

Figure 6.21: Evolution progress for multi-element aerofoil inverse design problem.

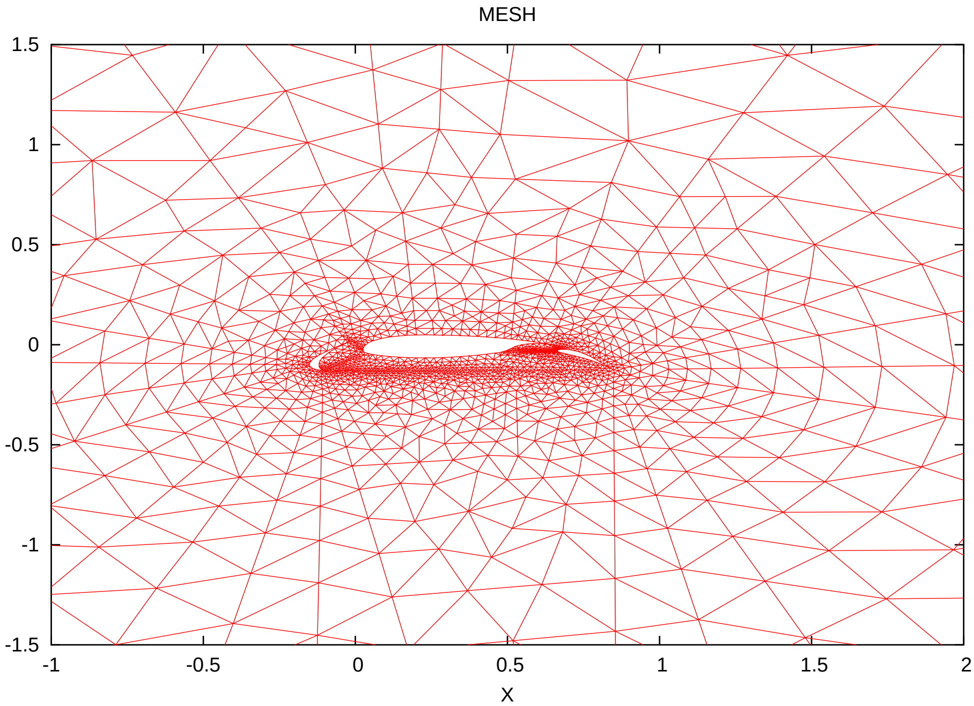

Figure 6.22: Grid around a candidate multi-element aerofoil configuration.

Figure 6.23: Comparison of pressure coefficient distribution over target and final geometry.

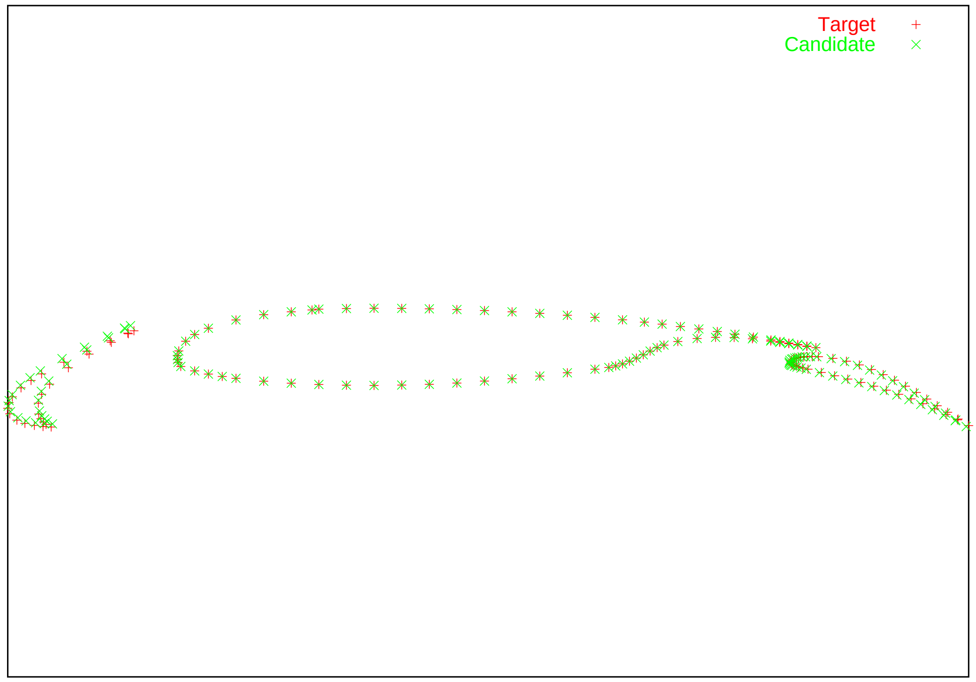

Figure 6.24: Comparison of target and final geometry.

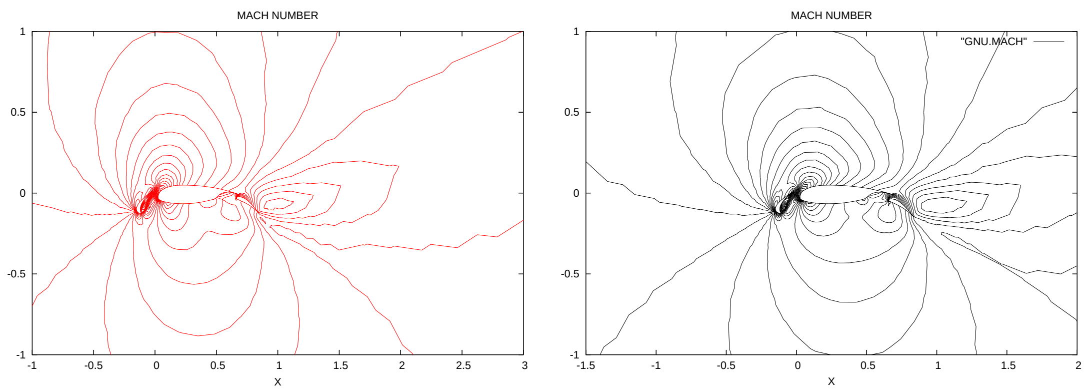

Figure 6.25: Mach contours around target and final geometry.

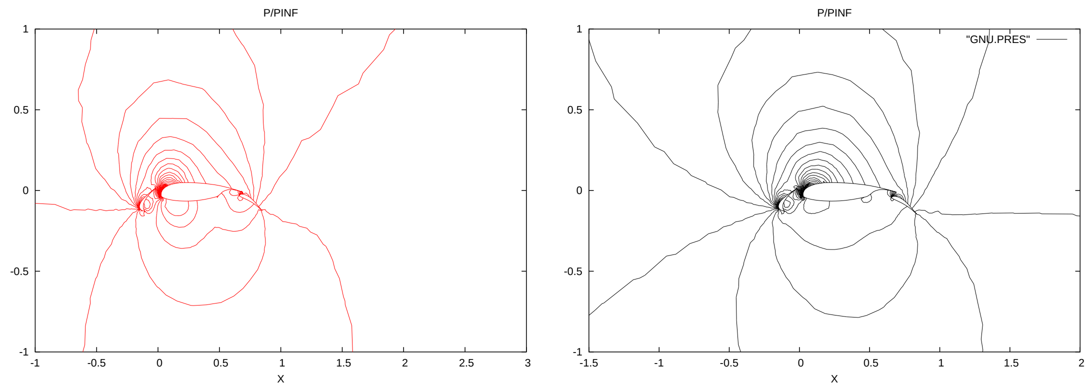

Figure 6.26: Pressure contours around target and final geometry.

6.3.4 Two-dimensional Two Objectives Aircraft High-lift System Design and Optimisation

In this problem the previous test case is extended to two objectives. The problem consists of two objectives aimed at minimising the difference between computed surface pressure distributions of a predefined three-element high-lift aircraft system (deployed slat-main-flap aerofoils during landing or take off) operating at two different flow conditions, as illustrated in Figures 6.27 and 6.28. Similar to the previous section, the fitness functions are the RMS error of the surface pressure coefficients on all three elements. The problem is solved when the positive value of the fitness goes below a prescribed value.

$$f_1 = \min \left(\frac{1}{N}\Sigma_{elements}\left(\Sigma_{i=1}^{N}\left(Cp_{candidate} - Cp_{target-one}\right)\right)\right) \tag{6.4}$$

$$f_2 = \min \left(\frac{1}{N}\Sigma_{elements}\left(\Sigma_{i=1}^{N}\left(Cp_{candidate} - Cp_{target-two}\right)\right)\right) \tag{6.5}$$

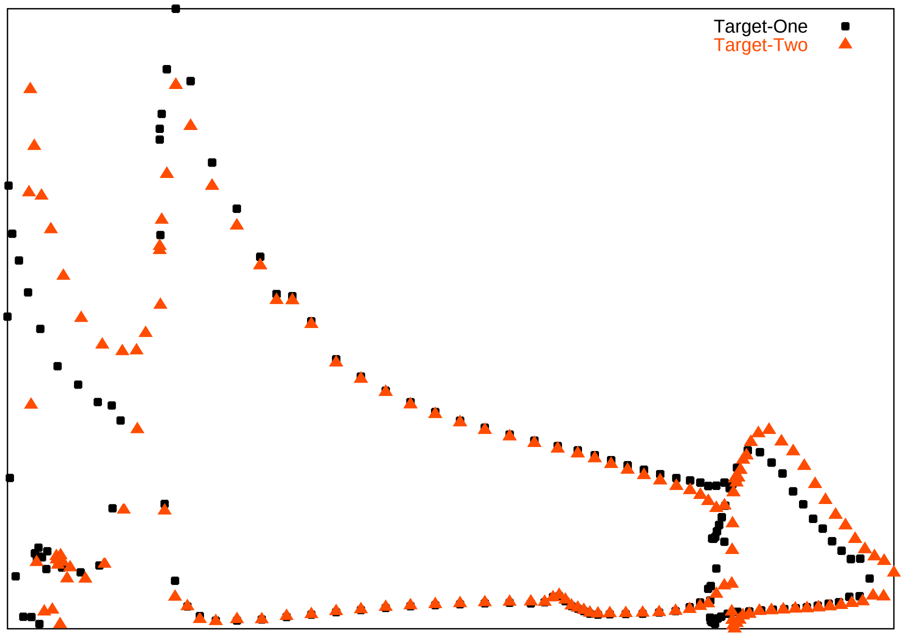

Figure 6.27: Target pressure coefficient distributions.

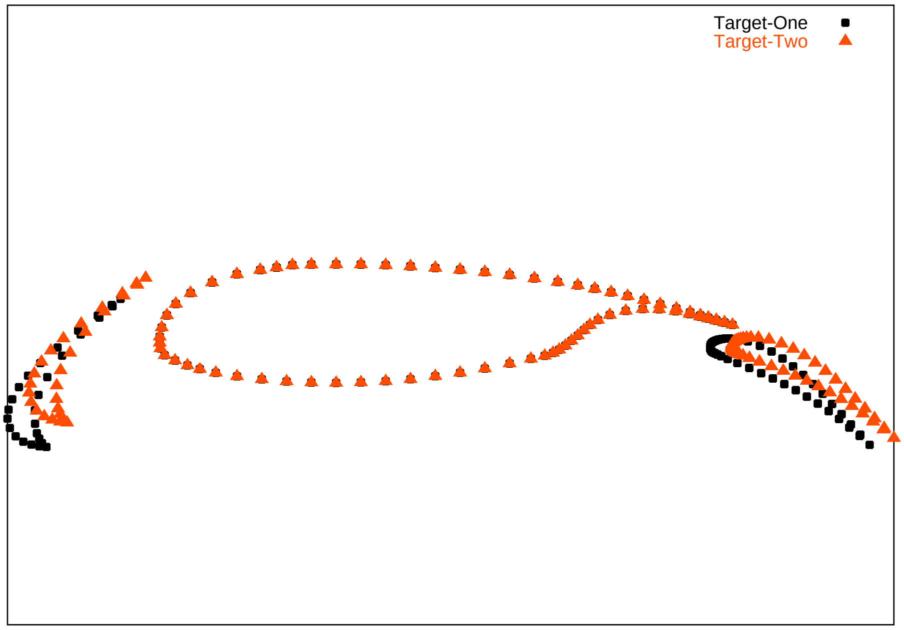

Figure 6.28: Target aircraft high-lift system configuration.

6.3.4.1 Implementation.

This method uses a hierarchical topology of evolutionary algorithms and mesh densities with the following settings:

Top Layer: A population size of 40, intermediate recombination used between two parents, and a maximum of 12500 mesh vertices on BAMG.

Middle Layer: A population size of 40, discrete recombination used between two parents, and a maximum of 11000 mesh vertices on BAMG.

6.3.4.2 Computational Results

This case was run for 500 function evaluations of the head node, and took approximately six hours on a cluster of five machines with performances varying between 2.4 and 2.8 GHz. Figure 6.29 shows a well-distributed Pareto front. A comparison of the pressure distribution for the target and best fit found for objective one and objective two are shown in Figures 6.30 and 6.31, respectively. Figures 6.32 and 6.34 compare the mesh generated around the targets and best geometries. Figures 6.33 and 6.35 show the Mach number contours on the target and best geometries and Figure 6.36 those for the pressure contours. A good agreement can be seen on different pressure and Mach number contours. This case illustrates the benefits of parallel computing and the capabilities of the methods and framework for high-lift aircraft system multi-objective problems. Without any problem-specific knowledge, the correct pressure distribution over a high-lift aircraft system configuration operating at two different flow conditions has been captured.

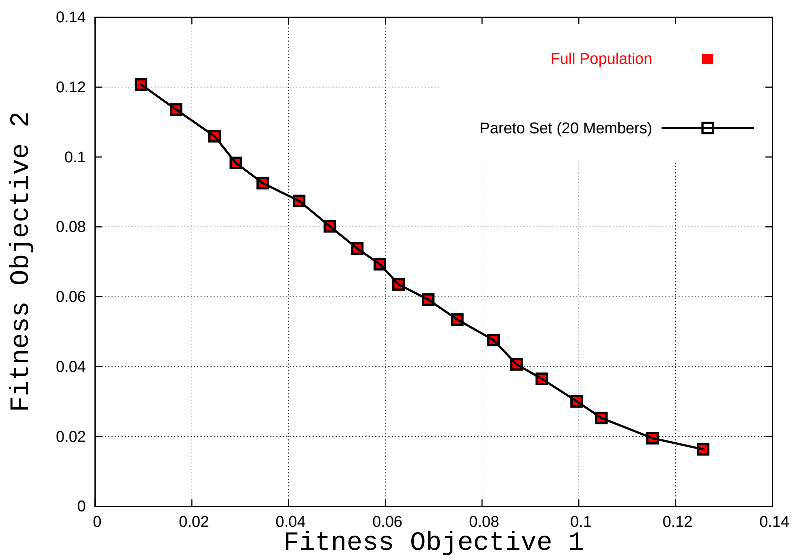

Figure 6.29: Pareto front for high-lift system design.

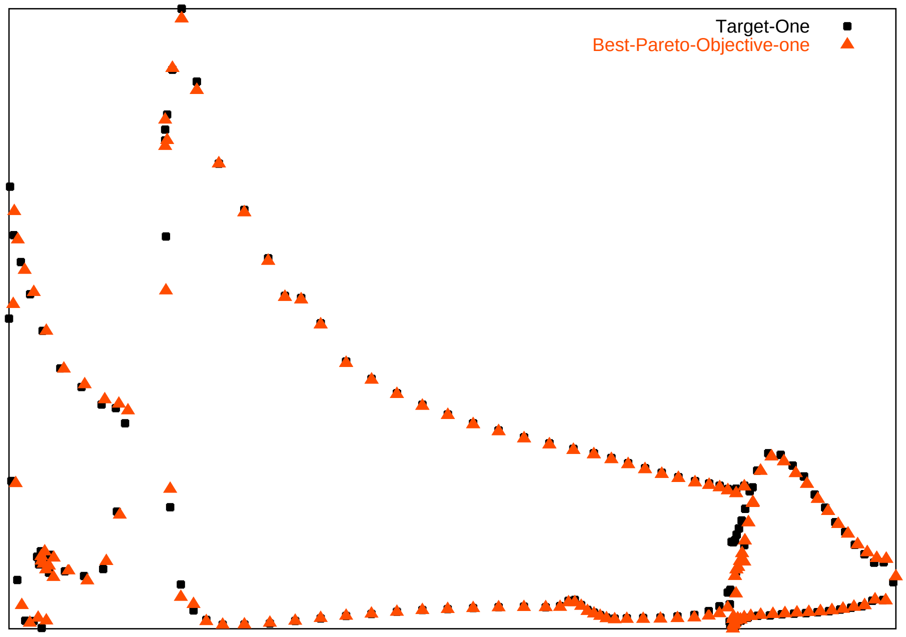

Figure 6.30: Comparison of pressure distribution of Target one and best for objective one.

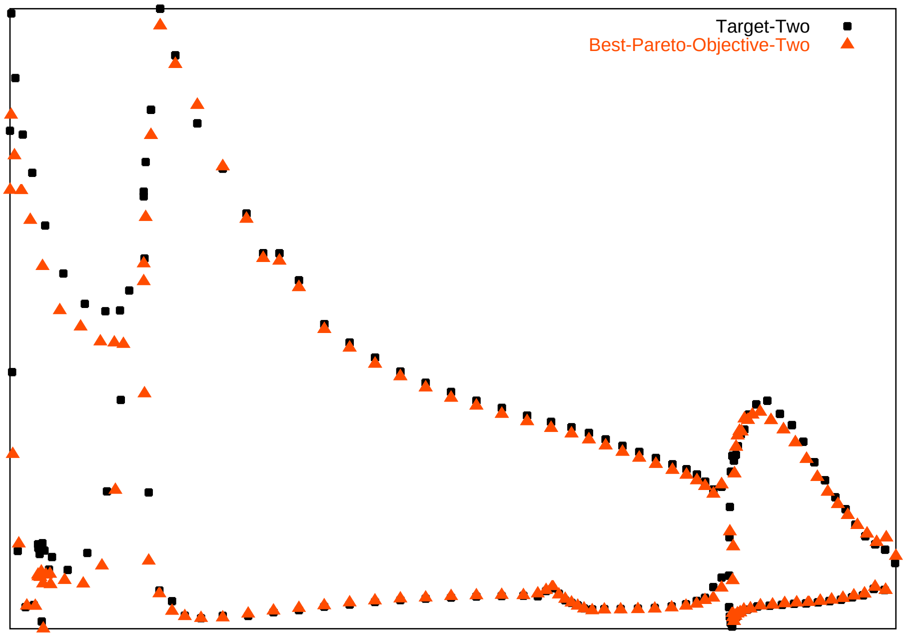

Figure 6.31: Comparison of pressure distributions of Target two and best for objective two.

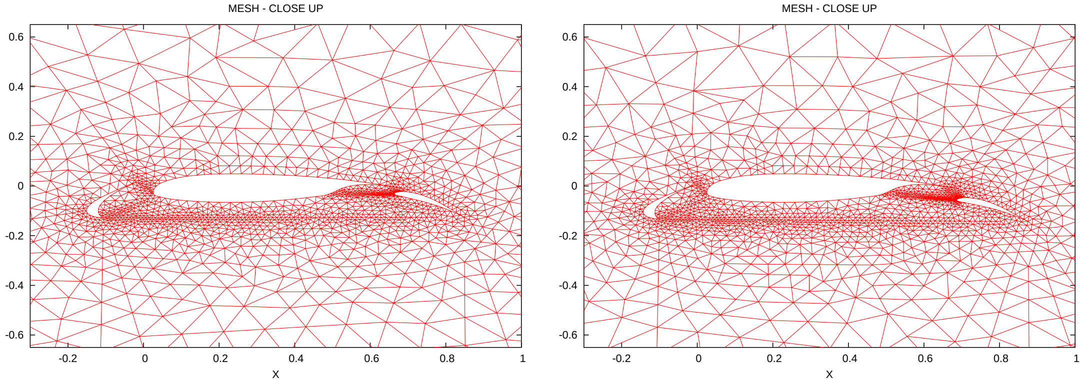

Figure 6.32: Mesh around target one and best for objective one.

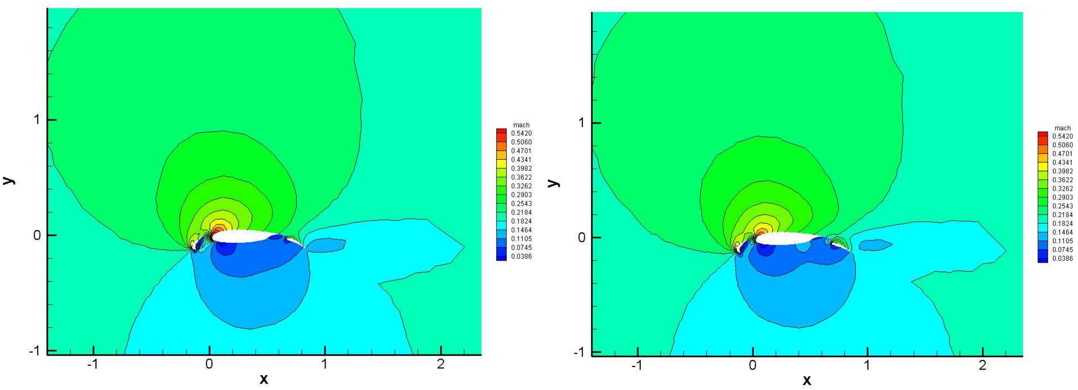

Figure 6.33: Mach contours around target one and best for objective one.

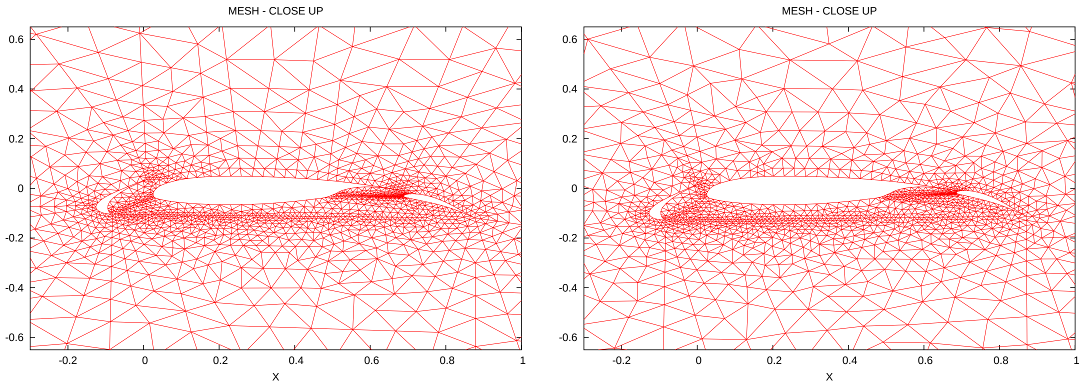

Figure 6.34: Mesh around target two and best for objective two.

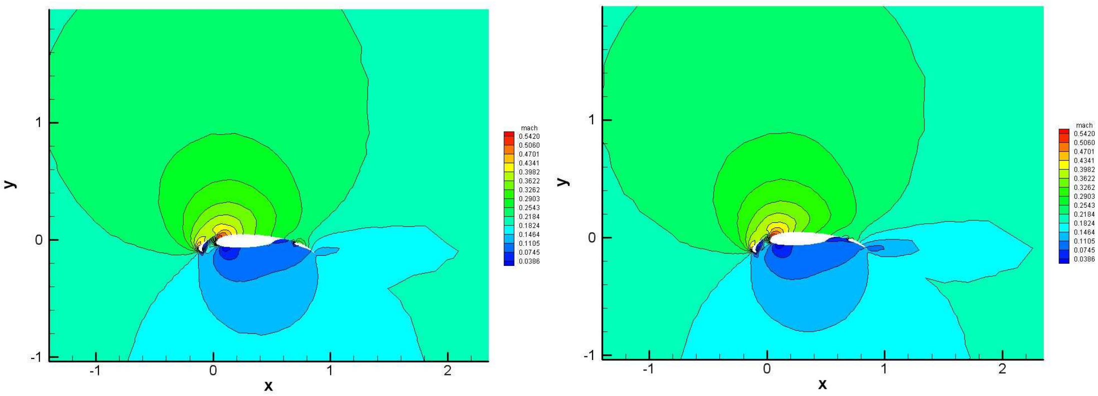

Figure 6.35: Mach contours around target two and best for objective two.

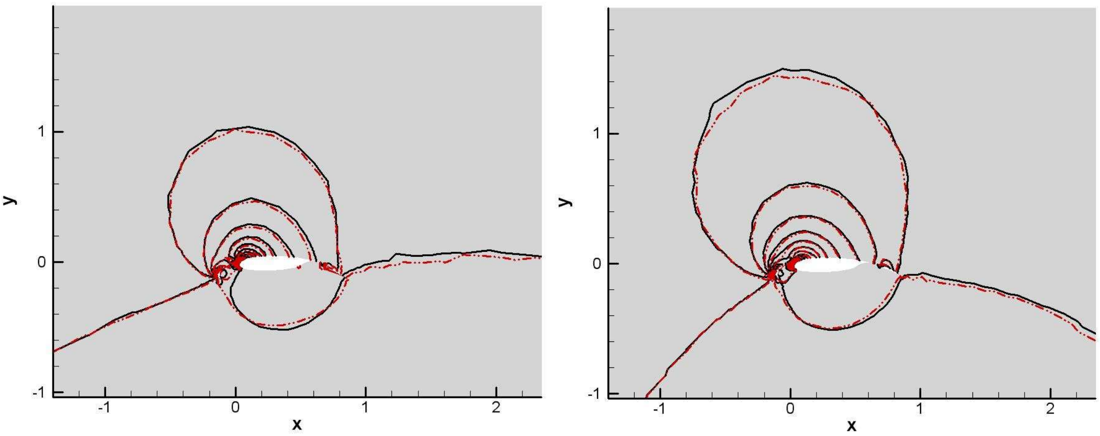

Figure 6.36: Pressure contours comparison around target one and target two.

6.4 Summary

This chapter demonstrated the feasibility and practicality of the methods and framework for aerodynamic shape optimisation. The methods were applied to realistic, practical, real-world examples in aeronautics. Studies show how the methods can be developed to find optimal solutions for inverse problems with one or two objectives. Results on multi-element aerofoil show how that method is capable of finding the correct pressure and flow-field solution. As developed, the coupling of the optimisation tool, the GUI and the solver was easy to set up and requires only a few hours for the simplest cases. The benefits of using parallel computing and evolution algorithms to provide solutions for single- and multi-objective inverse aerofoil design problems is clear. Further study into mesh adaptation and unstructured grid for complex geometries is required.

Chapter 7

Multi–Objective and Multidisciplinary Wing Design and Optimisation Applications

"Well, I must endure the presence of two or three caterpillars if I wish to become acquainted with the butterflies. It seems that they are very beautiful." Antoine de Saint-Exupéry

7.1 Introduction

In this chapter the robustness and practicality of some evolutionary methods for multi-objective and multidisciplinary wing design optimisation problems are demonstrated.

The previous two chapters focused on the application to aerodynamic shape optimisation problems and showed some of the key considerations when coupling an algorithm with analysis software and parallel computing. This chapter takes and expands on these concepts and demonstrates the application of the methods to other complex problems.

The performance and advantages of the algorithms are compared to that of a classical EA which would normally use only a single complex model and involve larger computational expense. In the examples considered, the flight regime can be transonic. Therefore, special consideration of transonic effects was evaluated to select a robust and efficient solver to properly account for these effects.

Table 7.1 summarises the test cases devised and evaluated in this chapter. The problems are solved using the wing design module within the framework (Figure 7.1).

Type	Single/ Multi- objective	Description	Solver
Direct	Multi- objective	Multi–objective Wing Design	Potential flow solver
Direct	Multi- objective	Multi–objective and Multidisciplinary UAV Swept Forward Wing Design	Potential flow solver
Direct	Multi- objective	Multi–objective and Multidisciplinary UAV Wing Design	Potential flow solver
DOE/FEA/CFD	Multi- objective	Aero-Structural Analysis on UAV Wing	Potential flow solver /FEM

Table 7.1: Multidisciplinary and multi-objective wing design test cases.

7.2 Aerodynamics and Structural Analysis

The aerodynamic characteristics of the wing configurations are evaluated using a three dimensional full potential wing analysis code (*FLO22*) which uses sheared parabolic coordinates and accounts for wave drag [76]. The potential flow solver *FLO22* was developed by Jameson *et al.* [76] for analysing inviscid, isentropic, transonic flow past 3-D swept wing configuration. The free-stream Mach number is restricted by the isentropic assumption and weak shock waves are automatically located wherever they occur in the flow. Also, the finite difference form of the full equation for the velocity potential is solved by the methods of relaxation, after the flow exterior to the aerofoil is mapped to the upper half plane. The mapping procedure allows exact satisfaction of the boundary conditions and use of supersonic free-stream velocities. Details on the formulation and implementation of this solver can be found in Jameson *et al.* [76].

The choice of using a potential flow-solver as opposed to a Navier-Stokes solver comes from the fact that the purpose of this thesis is to illustrate the workings of the methods. A full Navier-Stokes solution, which might take hours, is prohibitive due to computational resources, whereas the potential flow solution takes only a few seconds.

The lift can be satisfied by performing an extra two function evaluation by varying the angle of attack at the wing root and assuming a linear variation of the lift coefficient.

$$\alpha_{root} = \left(\frac{C_{L_{target}} - C_{L_{\alpha=\alpha_1}}}{C_{L_{\alpha=\alpha_2}} - C_{L_{\alpha=\alpha_1}}} \right)_1 (\alpha_2 - \alpha_1) + \alpha_1 \tag{7.1}$$

where α_1 and α_2 are set to 3 and 6 degrees respectively. The wing weight can be estimated from the wing spar cap area designed to resist the bending moment.

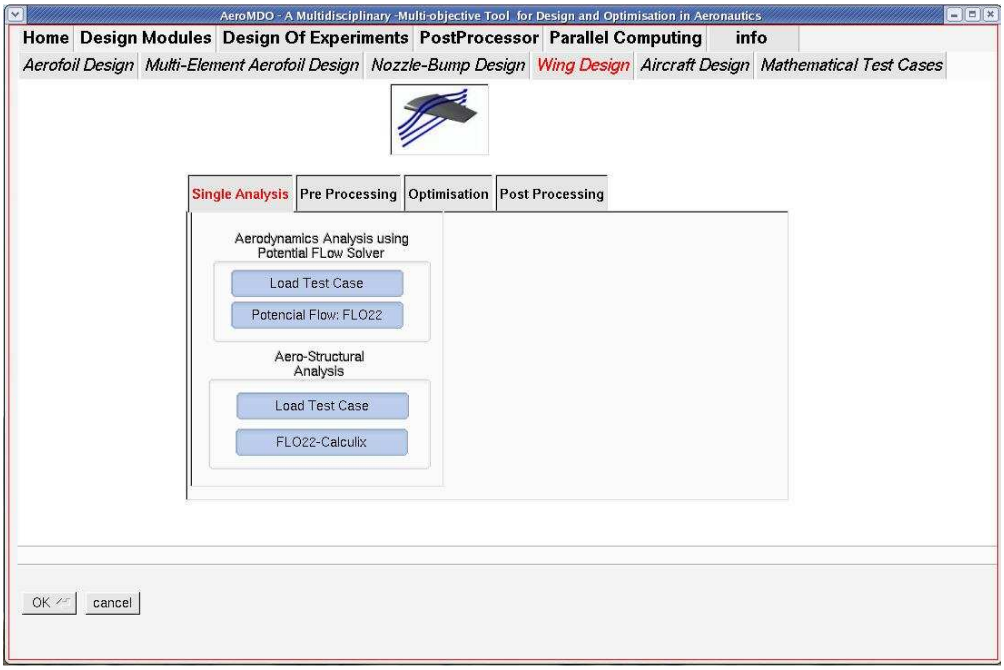

Figure 7.1: Wing design and optimisation module.

An FEM or an analytical formulation can be used for the structural analysis. The lift distribution is replaced by concentrated loads and the span-wise bending moment is calculated by:

$$\frac{d^2M}{dy^2} = -L_y \tag{7.2}$$

The local stress has to be less than the admissible compressive stress, $\sigma < \sigma_{adm}$ and the bending stress at each station is calculated by:

$$\sigma = \frac{M}{I}\frac{t_{max}}{2} \tag{7.3}$$

From here the minimum spar cap area can be calculated as:

$$A_{sc} = Sf \times \frac{M}{\sigma_{allow} * t_{max}} \tag{7.4}$$

For Aluminum Alloy 2024-T6 $E = 7.52\ GPa$, $\varepsilon_{allow} = 0.003$, $\rho_{Al} = 2740\ Kgs/m^3$ and $t_{max} = (t/c)_{max}$ and assuming level flight conditions the safety factor is $Sf = 3.5$.

The spar cap weight at each span station is given then by:

$$w_{sc} = \rho_{Al} * l_i * A_{sc} \tag{7.5}$$

where l_i is the length of the spar cap between two consecutive stations. The total spar cap weight can then be calculated as:

$$W_{sc} = \sum_{i=0}^{N} w_{sc} \tag{7.6}$$

N number of span-wise stations.

7.3 Multi-objective Wing Design

7.3.1 Formulation of the Multi-objective Problem

To illustrate these concepts, a multi-objective application method for the optimisation of a swept-forward wing design for an Unmanned Aerial Vehicle (UAV) was developed. These vehicles can benefit from the use of swept-forward wings which provide some benefits when compared to swept-back wings. The properties of swept-forward wings at low speeds have been known for some time. Swept-forward wings have an uneven span-wise distribution of lift and an excessive root bending moment. The largest loads occur at the root, while an aft-swept wing has a more gradual loading with a maximum lift around mid-span.

In this optimisation problem, a dual-point design procedure is described to find the Pareto front of swept-forward wings for minimum drag at two design points: $M = 0.69,\ C_L = 0.5$ and $M_\infty = 0.69,\ C_L = 0.4$. The cruise altitude is 10000 ft and the wing area is 2.94 m^2.

7.3.2 Design Variables and Constraints.

The wing geometry is represented by three aerofoil sections and six planform variables. Each aerofoil section is represented by the combination of two Bézier curves. In this case, for each aerofoil six free control points are taken on the mean line and ten free control points on the thickness distribution. The wing planform design variables are shown in Figure 7.2, and their upper and lower bounds are presented in Table 7.9. In total, fifty-four design variables are used for the optimisation. Constraints are imposed on minimum thickness: ($t/c\ \geq 14$ % root aerofoil, ≥ 12 % intermediate aerofoil and $\geq 11\%$ tip aerofoil) and position of maximum thickness $\left(20\% \leq x_{t/c} \leq 55\%\right)$. If any of these constraints is violated, both fitnesses are linearly penalised to ensure an unbiased Pareto front.

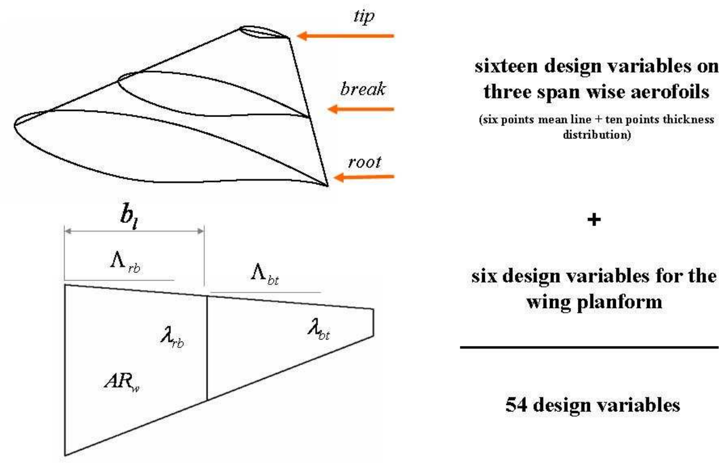

Figure 7.2: Design variables for multi-objective wing design.

Description	Variable	Lower bound	Upper bound
Wing Aspect Ratio	AR	3.0	5.0
Break to root Taper	λ_{wrb}	0.7	0.9
Break to tip Taper	λ_{wbt}	0.2	0.35
Wing 1/4 Chord inboard Sweep, *deg*	Λ_{winb}	10	38
Wing 1/4 Chord outboard Sweep, *deg*	Λ_{woutb}	-28	-1
Break Location, %	b_l	0.05	0.25

Table 7.2: Design variables for a UAV multi-objective wing design.

7.3.3 Fitness Functions

The two fitness functions to be optimised are defined as:

$$\begin{aligned} f_1 &= min(Cd_w) + Penalty \rightarrow M_\infty = 0.69,\ C_L = 0.4 \\ f_2 &= min(Cd_w) + Penalty \rightarrow M_\infty = 0.69,\ C_L = 0.5 \end{aligned} \tag{7.7}$$

7.3.4 Implementation

The solution to this problem has been implemented by developing a method that uses a hierarchical topology of evolutionary algorithms and mesh densities with the following settings:

Top Layer: A population size of 30, intermediate recombination used between two parents, and a mesh of $96 \times 12 \times 16$ on the FLO22 solver.

Middle Layer: A population size of 30, discrete recombination used between two parents, and a mesh of $72 \times 9 \times 12$ on the FLO22 solver.

Bottom Layer: A population size of 30, discrete recombination used between two parents, and a maximum of $48 \times 6 \times 8$ on the FLO22 solver.

7.3.5 Computational Results

The algorithm was allowed to run for 3000 function evaluations and took approximately five hours to run, using six machines on the Barossa cluster (Chapter 3). The final population, including the Pareto optimal front, is shown in Figure 7.3. Objective one and objective two are Cd_w, $flight\ condition_1$, and Cd_w, $flight\ condition_2$, respectively. It can be seen how the method gives a uniformly distributed front. The top-left configuration corresponds to Pareto member one (PM 1), which is best suited for objective one. The bottom-right configuration corresponds to Pareto member 8 (PM 8), which is best suited for objective two. Figures 7.4 and 7.5 show a top and side view of the planforms. Figure 7.6 shows the normalised aerofoils sections at the root, break and tip wing-span stations for a compromise design (PM 4) taken from the middle of the Pareto front. As illustrated in these figures, the algorithm was capable of identifying the trade-off between the two objectives and providing classical aerodynamic shapes as well as alternative configurations from which the designer can choose.

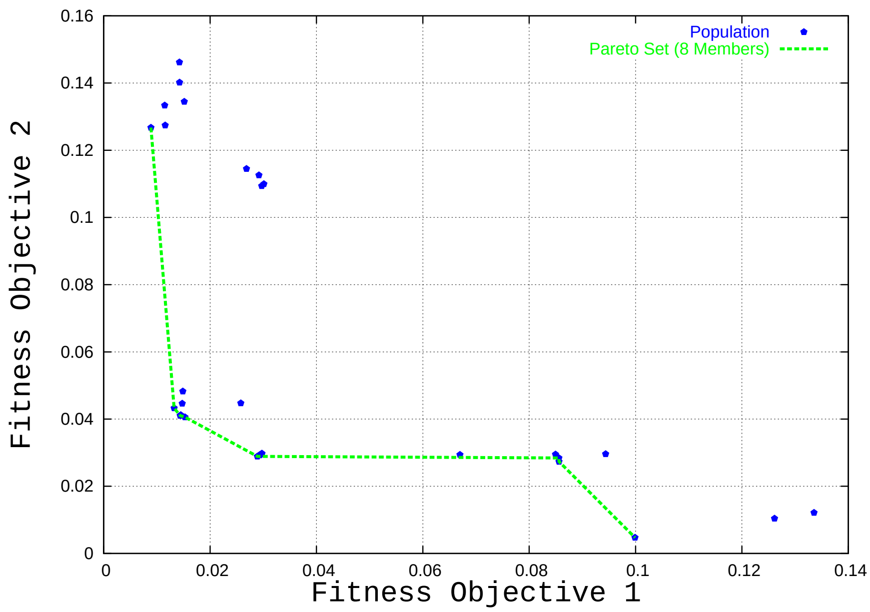

Figure 7.3: Pareto front for UAV wing design.

Figure 7.4: Ensemble of UAV wing planforms in the Pareto front (Top View).

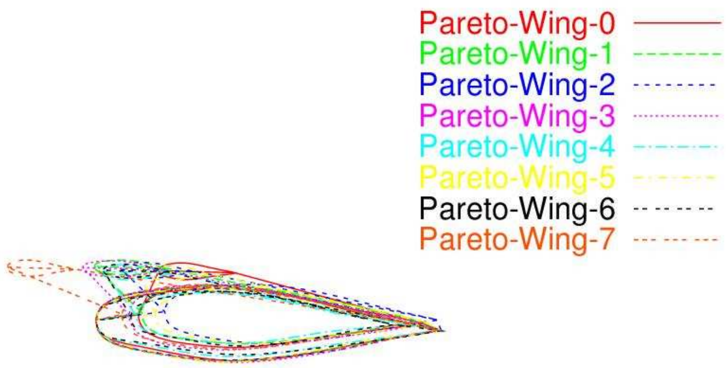

Figure 7.5: Ensemble of UAV wing planforms on the Pareto front (Side View).

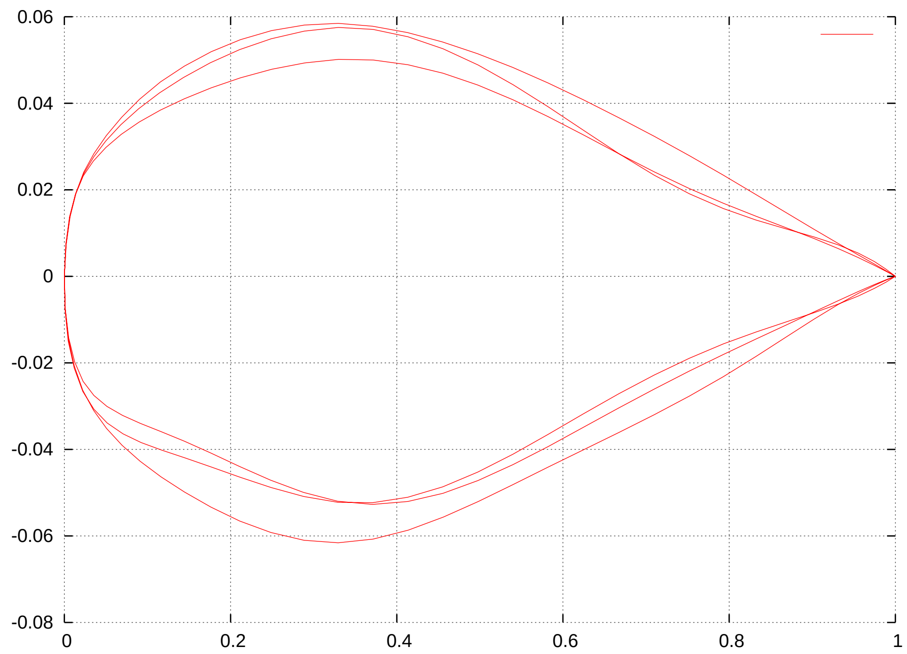

Figure 7.6: UAV wing aerofoil sections for Pareto Member four.

7.4 Multi-objective and Multidisciplinary UAV Swept Wing Design

7.4.1 Formulation of the MDO Problem

This test case, which is slightly different from the previous test case, considers a multidisciplinary, multi-objective optimisation for a swept-forward wing design for an Unmanned Aerial Vehicle (UAV). The two objectives are minimisation of wave drag and wing weight. The cruise Mach number and altitude are 0.69 and 10000 *ft* respectively. The wing area is set to 2.94 m^2 and the corresponding C_L is fixed at 0.19. For the solution, the pressure distribution over the wing is initially computed using a potential flow-solver to obtain the wing aerodynamics characteristics, which include the span-wise pressure distribution, C_L and total drag coefficients C_{D_w}. As indicated in Section 7.2, the lift distribution is replaced by concentrated loads and the spar cap area is calculated to resist the bending moment. The weight is approximated as the sum of the span-wise cap weight. The interaction between the aerodynamic pressure distribution and the structural deflections is ignored.

7.4.2 Design Variables and Constraints

The wing geometry is represented by three aerofoil sections and nine planform variables. Each aerofoil section is represented by the combination of a mean line and thickness distribution. In this case, for each aerofoil six free control points are taken on the mean line and ten free control points on the thickness distribution. The wing planform design variables and its upper and lower bounds are shown in Figure 7.7 and Table 7.3. In total, fifty-nine design variables are used for the optimisation.

Constraints are imposed on minimum thickness: ($t/c \geq 14\,\%$ root aerofoil, $\geq 12\,\%$ intermediate aerofoil and $\geq 11\%$ tip aerofoil) and position of maximum thickness $\left(20\% \leq x_{t/c} \leq 55\%\right)$. If any of these constraints is violated, both fitnesses are linearly penalised to ensure an unbiased Pareto front.

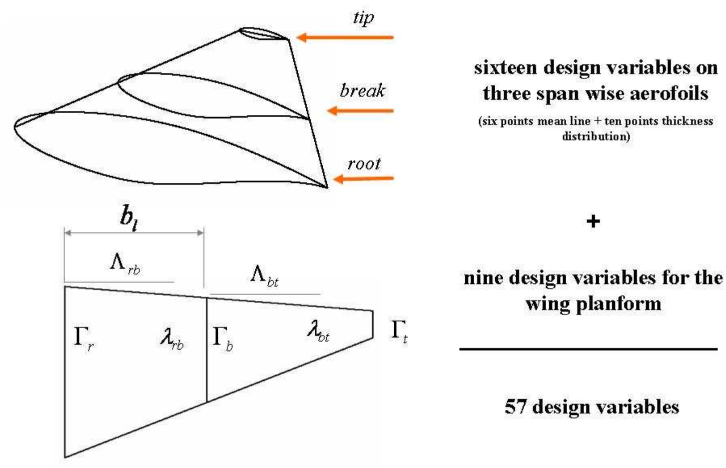

Figure 7.7: Design variables for multidisciplinary wing design.

using a p	**Variable**	**Lower bound**	**Upper bound**
Wing Aspect Ratio	AR	3.50	7.00
Break to root Taper	λ_{rb}	0.65	0.80
Break to tip Taper	λ_{bt}	0.20	0.45
Wing 1/4 Chord inboard Sweep, *deg*	Λ_{inb}	10.00	20.00
Wing 1/4 Chord outboard Sweep, *deg*	Λ_{outb}	-20.00	0.00
Twist at Root, *deg*	ϵ_r	0.00	3.00
Twist at Break, *deg*	ϵ_b	-1.00	0.00
Twist at Tip, *deg*	ϵ_t	-1.00	0.00
Break Location, *%*	b_l	0.20	0.35

Table 7.3: Design variables for a UAV multidisciplinary swept wing design.

7.4.3 Fitness Functions

The two fitness functions to be optimised are minimisation of wave drag and minimisation of spar cap weight:

$$\begin{aligned} f_1 &= min(Cd_w) + Penalty \\ f_2 &= min\,(W_{sc}) + Penalty \end{aligned} \tag{7.8}$$

7.4.4 Implementation

The solution to this problem has been implemented using two methods. The first method uses a traditional EA with a single population model and computational grid of $96 \times 12 \times 16$. The second method uses a hierarchical topology of evolutionary algorithms and mesh densities with the following settings:

Top Layer: A population size of 30, intermediate recombination used between two parents, and a mesh of $96 \times 12 \times 16$ on the FLO22 solver.

Middle Layer: A population size of 30, discrete recombination used between two parents, and a mesh of $72 \times 9 \times 12$ on the FLO22 solver.

Bottom Layer: A population size of 30, discrete recombination used between two parents, and a maximum of $48 \times 6 \times 8$ on the FLO22 solver.

7.4.5 Computational Results

The algorithm was run five times for 2000 function evaluations and took on average six hours to compute, using four machines on the Barossa cluster (Chapter 3). Figure 7.8 shows convergence history for objective one and Figure 7.9 shows the Pareto fronts obtained by using the two approaches. It can be seen how the optimisation technique gives a uniformly distributed front in both cases. In this figure the red line corresponds to the Pareto front computed using a single population approach, while the black line corresponds to the Pareto front obtained using the hierarchical topology. The use of a hierarchical approach gives an overall lower Pareto front, compared to a single model approach. This gives an indication of the benefit of a hierarchical topology for multi-criteria problems.

A compromise design, Pareto member ten (PM10), taken from the middle of the Pareto front is taken for further evaluation. Figure 7.10 shows the root, break and tip aerofoils, Figures 7.11 and 7.12 show the wing geometry and pressure coefficient distribution respectively and Table 7.4 shows the design variables for this Pareto member. Results indicate an aerodynamic shape with a good compromise between the aerodynamic and structural performances.

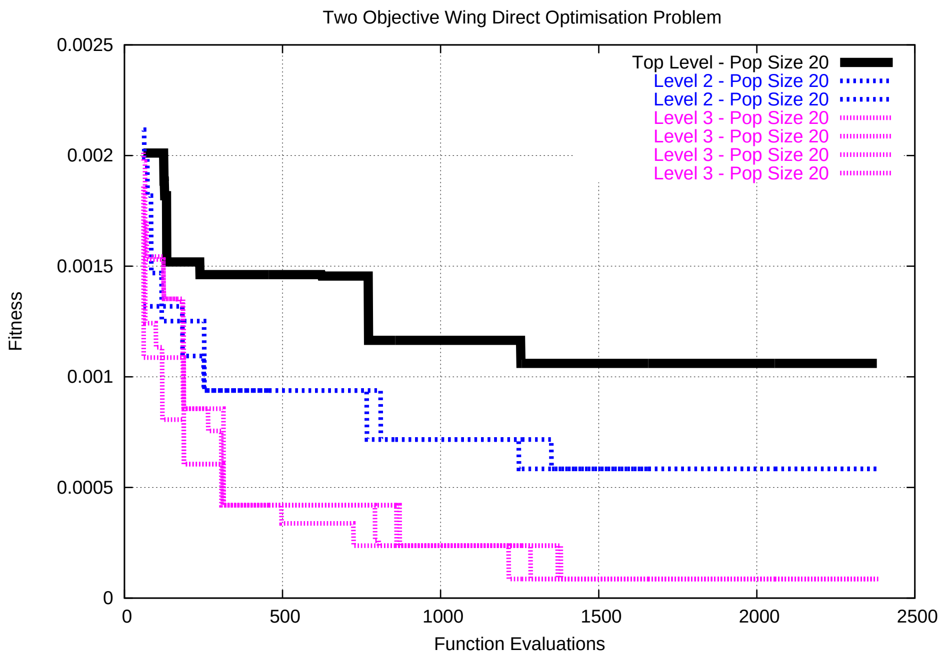

Figure 7.8: Convergence history for objective one.

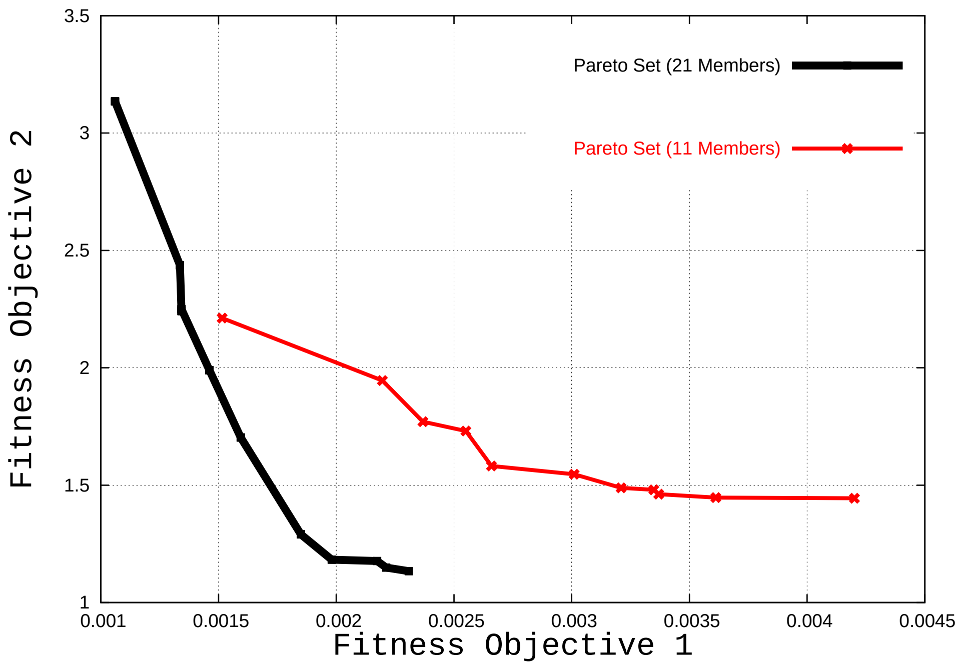

Figure 7.9: Pareto fronts after 2000 function evaluations.

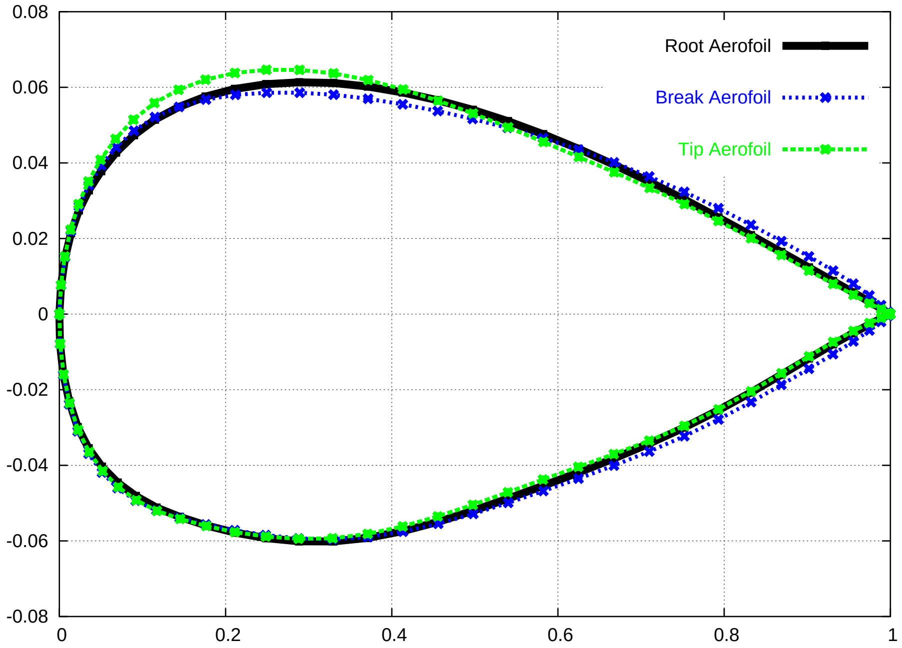

Figure 7.10: Aerofoil sections (root, break and tip) for Pareto Member ten.

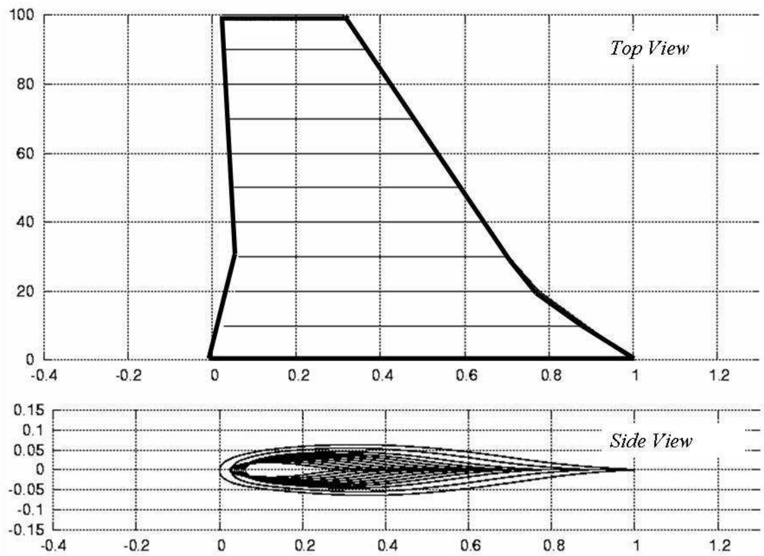

Figure 7.11: Wing top and side view for Pareto Member ten.

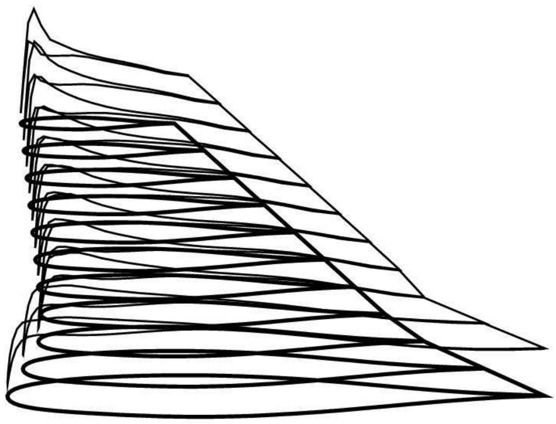

Figure 7.12: Wing span pressure coefficient distribution for Pareto Member ten.

Description	**Variable**	**Pareto Member 10**
Wing Aspect Ratio	AR	3.5
Wing 1/4 Chord inboard Sweep, *deg*	Λ_{inb}	10.2
Wing 1/4 Chord outboard Sweep, *deg*	Λ_{outb}	-1.9
Lift to Drag Ratio	L/D	146.62
Lift coefficient	C_L	0.1970
Drag Coefficient	C_D	0.0013 + *Cdv*

Table 7.4: Optimal design variables for Pareto Member ten.

7.5 Multi-objective and Multidisciplinary UAV Wing Design

7.5.1 Formulation of the MDO Problem

This test case considers the same UAV as in the previous test case and further illustrates the benefits of using a hierarchical approach. In this case the research looks for a normal swept *back* wing and compares two different implementation methods of the optimisation. The wing planform design variables are the same as those shown in Figure 7.7. The upper and lower bounds for this problem are presented in Table 7.5. The objective functions and implementation are the same as in the previous test case.

Description	Variable	Lower bound	Upper bound
Wing Aspect Ratio	AR	3.50	15.00
Break to root Taper	λ_{rb}	0.65	0.80
Break to tip Taper	λ_{bt}	0.20	0.45
Wing 1/4 Chord Sweep, *deg*	Λ_{inb}	10.00	25.00
Break Location, *%*	b_l	0.20	0.35

Table 7.5: Design variables for a UAV multidisciplinary wing design.

7.5.2 Computational Results

The algorithm was run five times for 2000 function evaluations and took on average seven hours to compute, using four machines on the Barossa cluster (Chapter 3). Figure 7.13 shows the Pareto fronts obtained using the two approaches. It can be seen how the optimisation technique gives a uniformly distributed front in both cases.

A trend similar to the previous test case can be seen. The blue line corresponds to the Pareto front computed using a single population approach; the black line corresponds to the Pareto front obtained using the hierarchical topology. It can be seen that the use of a hierarchical approach gives an overall lower Pareto front, compared to a single-model approach; this confirms the benefit of a hierarchical topology for multi-criteria problems. The combination of low fidelity models for a rapid exploration of the design space and higher fidelity models for the most promising solutions has been exploited during the optimisation. Figure 7.14 illustrates the Pareto front for the hierarchical approach and a representative top view of the wing geometries. Table 7.6 indicates the final values of the design variables.

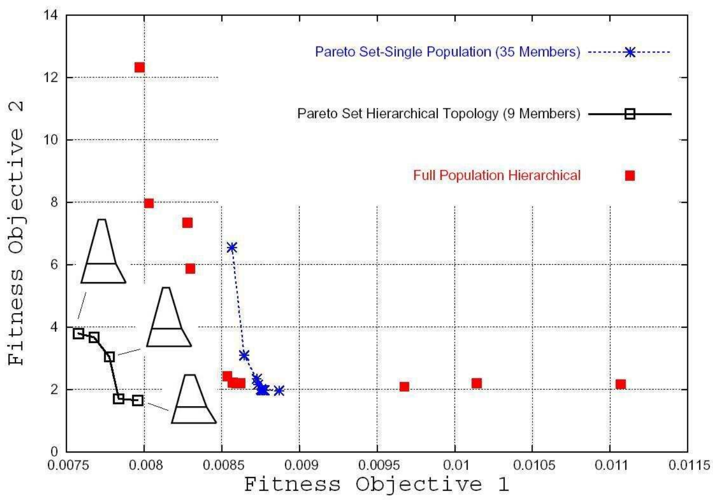

Figure 7.13: Pareto front comparison after 2000 function evaluations.

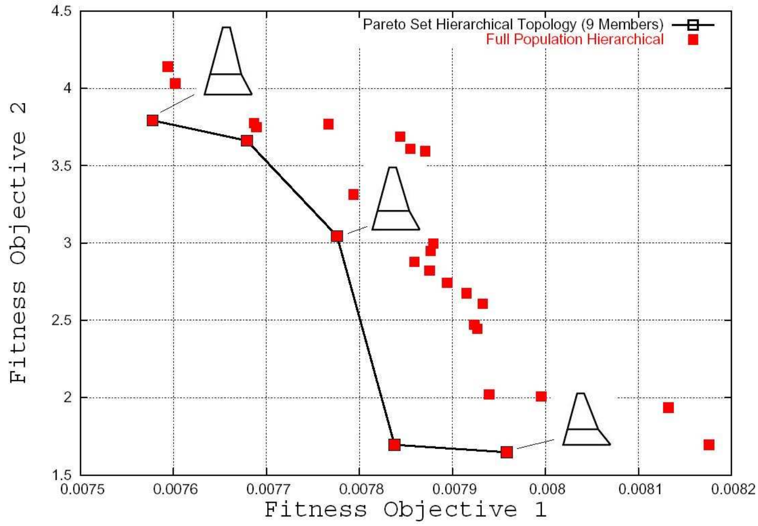

Figure 7.14: Pareto fronts and wing planforms.

Description	**Variable**	**Pareto Member 0**	**Pareto Member 4**	**Pareto Member 15**
Wing Aspect Ratio	AR	6.92	6.07	2.56
Wing 1/4 Chord Sweep, *deg*	Λ_{inb}	10.83	10.02	20.30
Wing semi-span, *ft*	b	2.14	2.00	1.30
Break to root Taper	λ_{rb}	0.74	0.68	0.69
Break to tip Taper	λ_{bt}	0.31	0.24	0.35

Table 7.6: Optimal design variables for some members of the Pareto front.

Concluding these cases, results show a computational gain by using a hierarchical topology of fidelity models, compared to a single model during the optimisation. The methods were capable of identifying the trade-off between the multi-physics involved, and provided classical aerodynamic shapes as well as alternative configurations from which the designer can choose.

7.6 Aero-structural Wing Design Optimisation

The objective of this study is to develop and implement the use of a robust method that couples some FEA, CFD and Design of Experiment (DOE) analysis tools. An aero-structural module that integrates several algorithms was developed for this task. It allows a single aero-structural analysis or optimisation. In the present study, medium-fidelity analysis tools are employed.

7.6.1 Wing Geometry Representation

In the current implementation, the wing geometry is represented by three aerofoil sections and some variables that control the planform shape ($AR, \Gamma,\ \lambda,\ \Lambda$). Each aerofoil section is represented by the combination of a mean line and thickness distribution. In this case, for each aerofoil six free control points are taken on the mean line and ten free control points on the thickness distribution. The wing planform design variables are shown in Figure 7.15. In total fifty-nine design variables can be used to represent the external shape of the wing.

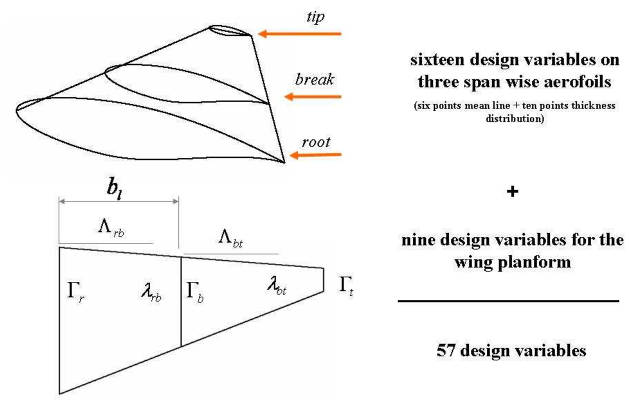

Figure 7.15: Design variables for aerodynamic analysis.

7.6.2 Aerodynamic Analysis

The aerodynamic characteristics of a candidate wing configuration are evaluated using the three -dimensional full-potential wing analysis code (*FLO22*) described in Section 7.2. Although ideally it is desirable to work with a Navier-Stokes solver, the choice of a potential solver comes from the fact that the purpose of this work is to illustrate the working of the method. A full Navier-Stokes solution might take a few hours and is prohibitive due to computational resources. A potential flow solution takes only a couple of minutes for the analysis.

7.6.3 Structural Analysis

The structural analysis is conducted using the CalculiX FEA software. As described by the authors of this program [44]: CalculiX is “A Free Software Three-Dimensional Structural Finite Element Program designed to solve field problems. The method used is the finite element method. The solver is able to do linear and non-linear calculations”. This software was modified to allow the incorporation of subroutines that allow communication between the CFD solver, framework and optimiser.

7.6.3.1 Structural Model and Constraints

For the structural analysis, a simplified finite element model consisting of a varying number of ribs and two spars is used. The model consists of shell elements and, for simplicity, the spars and ribs caps are not modelled. One of such finite element models is illustrated in Figure 7.16. As expected the number of nodes and elements varies depending on the wing geometry. The number of design variables in the structural analysis is related to the nodes and elements, and depends also on the number of internal spars and ribs. In the examples considered in this thesis, the ribs and spars are modelled as single panels with constant thickness.

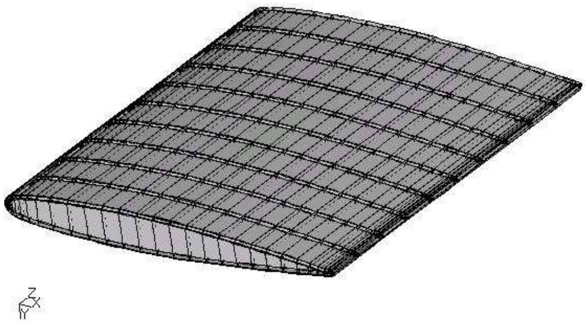

Figure 7.16: Structural finite element model.

The wing top and bottom panels are represented by quadrilateral elements. A schematic representation of one of these panels, with the forces, orientation and dimensions, is illustrated in Figure 7.17.

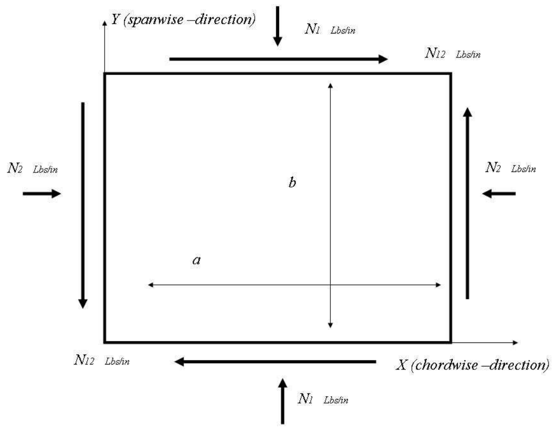

Figure 7.17: Wing cover panel.

7.6.4 Aero-structural Analysis

The aero-structural analysis starts by computing the pressure distribution over the wing. This pressure distribution is interpolated and converted to concentrated nodal forces applied at the area associated with each node of the structural model, as indicated in Equation 7.9:

$$F = W_l \times Cp_{i,\ j} \times S_{i,\ j} \tag{7.9}$$

where i is the span-wise location and j is the chord-wise location of the pressure coefficient C_p. W_l is the wing loading and $S_{i,j}$ is the area associated with each node. If the problem requires, the actual load F can be multiplied by a factor G equal to the g value of flight condition manoeuvre.

7.6.5 General Design and Optimisation Rationale

The general approach for optimisation uses an evolutionary optimiser, the DOE capability and the aero-structural analysis. The procedure is described in Algorithm 11. A general coupling algorithm and program was developed for this method. This code is described in Appendix B and is included in the CD placed in the back cover of this thesis.

Algorithm 11 Aero-structural wing design algorithm.

```
Define design variables, upper and lower bounds for the wing,
Define objective (fitness) functions to minimise/maximise (e.g. Drag, weight).
Create a series of design points at which the analysis will be performed.
..(This can be done using sampling techniques such as rectangular grids ..
...or Latin Hypercube Sampling of the DOE module.(drundace.cpp)
For each sampling point;

 Generate mesh for aerodynamic analysis, (mses2flo22.cpp)

 Compute Pressure/Force value with CFD on wet surface and obtain...
 ...aerodynamic characteristics for the wing (C_L, C_D, C_M). (flo22.f)
 Generate finite element structural model (number of spars, nodes,
 ....elements).(mses2cgx.cpp)
 Interpolate and transfer pressure/forces from fluid to ...
 ....structural model.(assign pressure.cpp)

 Compute displacement, stress and strain using a FEA software. (ccx.f)

 Check constraints violation (max stress, max displacement).(constraints.cpp)

 Transfer the displacements, velocity and acceleration from structure to ...
 ...fluid model after interpolation.
 Compute the objective functions (e.g. drag Cd_w and wing weight W) and ..
 ..generate a database with relevant information  for each design point
 ...(design variables, fitness function values, aerodynamic and structural data).
Generate a RSM or a DACE model using the sampling points information.
Start the optimisation by defining and evaluating a random initial
...population of wing geometries. (The evaluation is conducted using
...the approximate RSM or DACE model)
While stopping condition not met, evolve design variables ...
(recombination, mutation, adaptation), generate and evaluate ...
...new candidate wing geometries.W
```

NOTE: The aerodynamic model is influenced by the deformation for the wing. In the problems considered in this work, this interaction is not accounted for. It will be assumed that the wing is built to a shape that offsets the deformation due to aerodynamic loads.

7.6.6 Aero-structural Analysis of a UAV Wing

To illustrate the use of this algorithm and module the test case considered in Chapter 6 is extended for the design, analysis and optimisation of a wing for a small UAV application similar to the RQ-7A Shadow 200 Tactical UAV.

The objective is to conduct a DOE study and generate sampling points by varying the aerofoil and wing planform shape for this UAV and then compare the aerodynamic and structural properties of each of these points.

As described in that section, this UAV weight is approximately 320 *Lbs*, it has a wingspan of approximately 12.8 *ft*, a mean aerodynamic chord of approximately 2 *ft* and a planform shape with little to no-sweep.

The aircraft is assumed to be operating between a slow cruise 33.3 *m/s* and fast cruise 46.6 *m/s* approximately. This results in the airframe, flight parameters and operating conditions illustrated in Table 7.7. These conditions assume an aircraft at mid-weight-cruise during an extended cruise phase at intermediate altitude.

Aerofoil Section	*NACA4415*
Wing Span (approx), *ft*	12.8
Wing chord (approx), *ft*	2.0
Length, *ft*	11.2
Cruising altitude, *m*	3000

Flight Condition	**Flight Condition One Slow Cruise**	**Flight Condition Two Fast Cruise**
Velocity, *m/s*	33.3	46.6
Mach Number	0.1025	0.141
Angle of Attack	8.13	1.99

Table 7.7: UAV data and operating conditions.

It is also assumed that this UAV uses a single aerofoil, the NACA4415 aerofoil, as the only wing section aerofoil along the wing span. Figure 7.18 shows a 3D view of the original wing. Figure 7.19 show the wing span pressure coefficient (C_p) distribution. The global aerodynamic characteristics for this configuration are listed in Table 7.8. It is also assumed that the wing is manufactured from carbon fibre ($E = \ 14.2\ x\ 10^6\ psi$,$v = 0.34,\ \ G = 0.66\ x\ 10^6\ psi$)

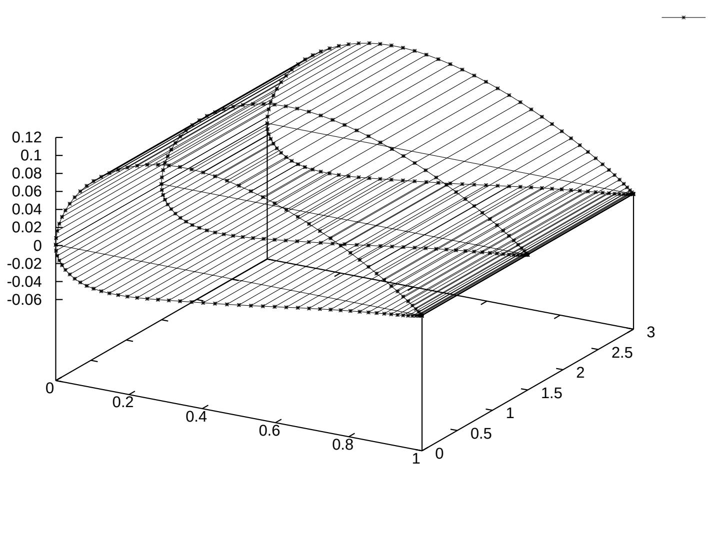

Figure 7.18: 3D view of original UAV Wing.

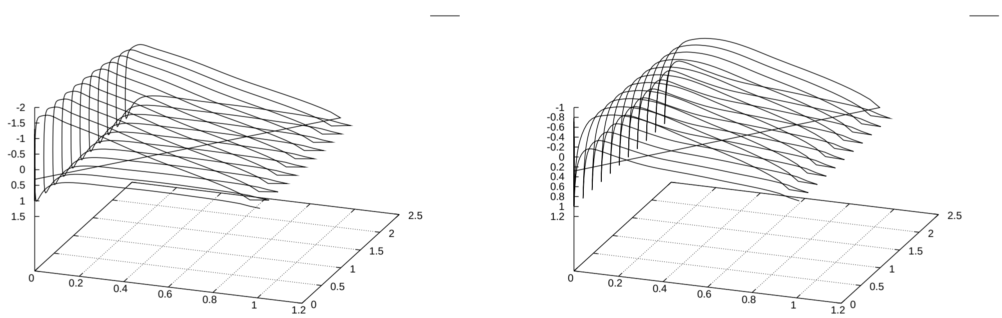

Figure 7.19: Pressure distribution on original UAV wing with NACA 4415 aerofoil.

Description	Flight Condition One Slow Cruise	Flight Condition Two Fast Cruise
C_D	0.0484226	0.0110074
$C_{L:}$	1.01729	0.526114
C_M	-0.357046	-0.23344
L/D	21.01	47.79

Table 7.8: Aerodynamic characteristics of a UAV wing with NACA 4415 aerofoil.

7.6.6.1 Design Variables

For illustration purposes, the design variables are restricted to wing aspect ratio and wing area. Their upper and lower bounds are provided in Table 7.9. A variation of the aerofoil shape is also considered; for illustration purposes the aerofoils can take the form of the LRN-1015, the NACA4415 or the CLARCK Y aerofoil. The choice of these aerofoils is arbitrary, but is based on the fact that these aerofoils have been used traditionally for different UAV planforms; reported data indicates that the LRN-1015 is used in the Global Hawk and Sensor Craft UAVs, the NACA4415 is used in RQ-7A Shadow 200 Tactical UAV and the CLARCK Y aerofoils is typical of small UAV planforms [111].

Description	Variable	Lower bound	Upper bound
Wing aspect ratio	AR	5.0	7.0
Wing Area	SW	4.0	8.0

Table 7.9: Design variables for a UAV wing design.

7.6.6.2 Fitness Functions

The two fitness functions to be computed in this case are defined as:

$$\begin{aligned} f_1 &= min(L/D_{flight\ Condition\ 1}) \\ f_2 &= min\,(L/D_{flight\ Condition\ 2}) \end{aligned} \tag{7.10}$$

7.6.6.3 Design Constraints

In this case, maintaining geometrical constraints such as spar thickness or maximum thickness location is not a concern. The only constraints considered are the maximum stress and displacement allowed. The local stress has to be less than the admissible shear stress, in this case f$\sigma < \sigma_{adm} = 0.66\ x\ 10^6 psi$.

7.6.6.4 Implementation

The problem was implemented using the aero-structural module and the algorithm described above. The design points using rectangular grids DOE are indicated in Table 7.10.

Description	**Aerofoil-Type**	AR	SW
Sampling Point 0 (SP0)	LRN-1015	5.0	4.0
Sampling Point 1 (SP1)	LRN-1015	5.0	6.0
Sampling Point 2 (SP2)	LRN-1015	5.0	8.0
Sampling Point 3 (SP3)	NACA4415	6.0	4.0
Sampling Point 4 (SP4)	NACA4415	6.0	6.0
Sampling Point 5 (SP5)	NACA4415	6.0	8.0
Sampling Point 6 (SP6)	NACA4415	7.0	4.0
Sampling Point 7 (SP7)	CLARCK Y	7.0	6.0
Sampling Point 8 (SP8)	CLARCK Y	7.0	8.0

Table 7.10: Rectangular grid design points.

7.6.6.5 Analysis and Results

Figure 7.20 illustrates a 3D view of some of the wing planform geometries. Figure 7.21 illustrates the pressure coefficient (Cp) distribution for some configurations. Table 7.11 summarises the sampling points and the fitness functions for the two flight conditions. For illustration purposes one of these points (sampling point 4) is taken for further discussion. Figure 7.22 illustrates the deformed geometry. Figure 7.23 shows the structural mesh used for the computation. Figures 7.24 and 7.25 show the displacement and stress distribution for the two flight conditions.

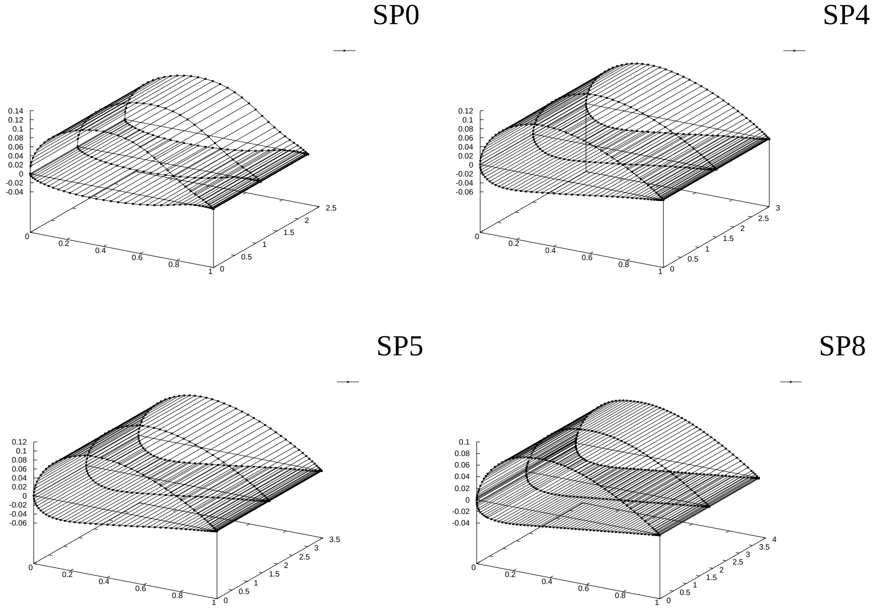

Figure 7.20: 3D view of some of the wing sampling points .

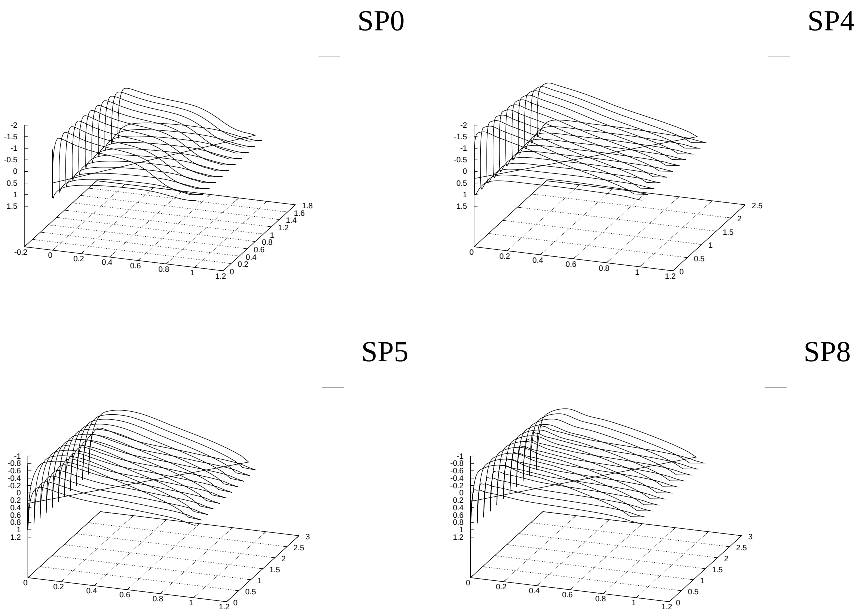

Figure 7.21: Wing-span pressure coefficient distribution for some of the sampling points.

Description	Aerofoil-Type	AR	SW	L/D_{fc1}	L/D_{fc2}
Sampling Point 0 (SP0)	LRN-1015	5.0	4.0	15.42	29.98
Sampling Point 1 (SP1)	LRN-1015	5.0	6.0	17.43	34.22
Sampling Point 2 (SP2)	LRN-1015	5.0	8.0	19.10	37.77
Sampling Point 3 (SP3)	LRN-1015	6.0	4.0	17.71	39.35
Sampling Point 4 (SP4)	NACA4415	6.0	6.0	20.14	45.54
Sampling Point 5 (SP5)	NACA4415	6.0	8.0	22.15	50.80
Sampling Point 6 (SP6)	NACA4415	7.0	8.0	41.55	41.55
Sampling Point 7 (SP7)	CLARCK Y	7.0	4.0	52.24	52.24
Sampling Point 8 (SP8)	CLARCK Y	7.0	6.0	58.10	58.10
Reference Wing	NACA4415	6.4	6.4	21.01	47.80

Table 7.11: Fitness values and aerodynamic data for some wing sampling points.

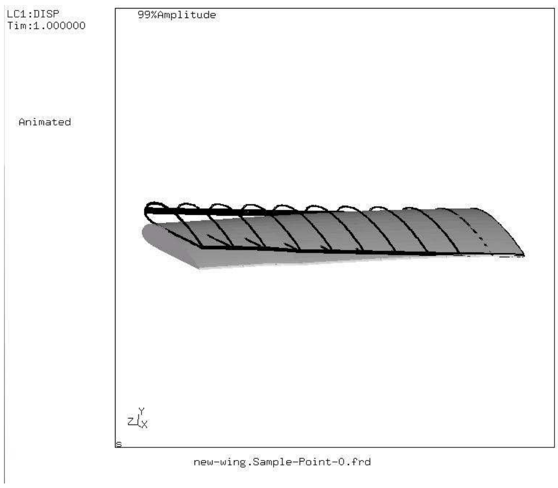

Figure 7.22: Deformed geometry for wing sampling point four (SP4).

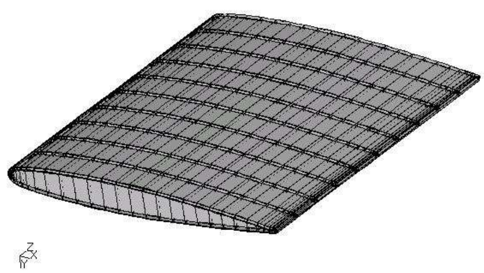

Figure 7.23: Structural finite element model for wing sampling point four (SP4).

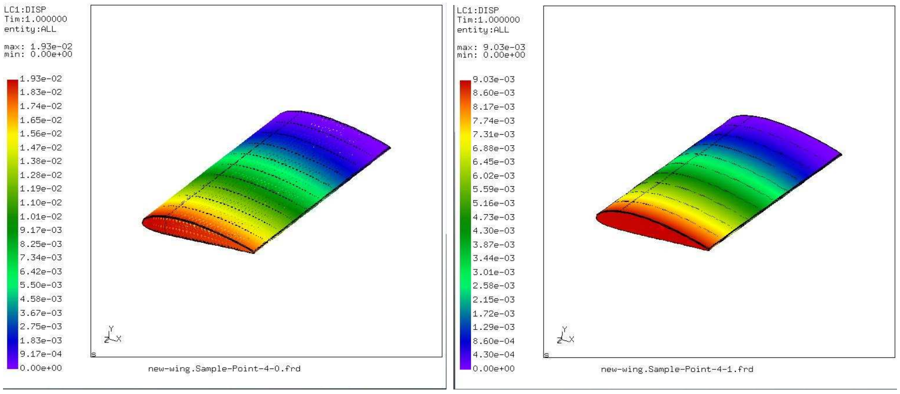

Figure 7.24: Displacement results after structural analysis for wing sampling point four (SP4).

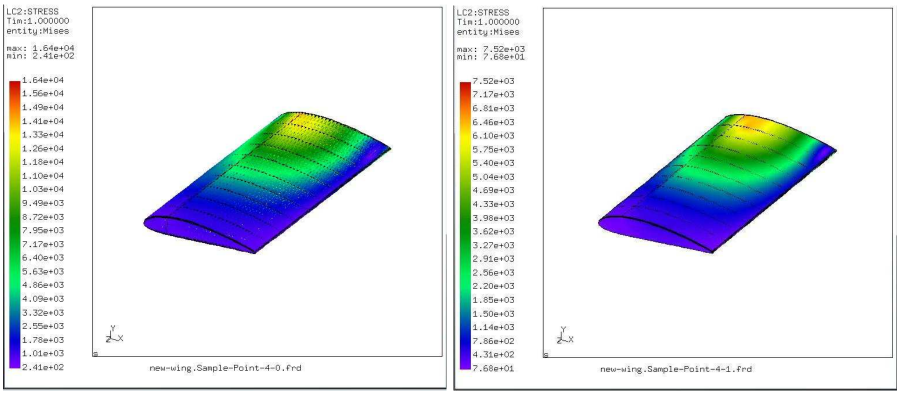

Figure 7.25: Von-Mises results after structural analysis for wing sampling point four (SP4).

Concluding this case, it is demonstrated how this method and algorithm couples a CFD, an FEA and a DOE capability for multi-objective and multidisciplinary design optimisation problems. As designed, the algorithm can be used as part of an evolutionary optimisation process. Such optimisation was not attempted in this research due to time limitations, but, as designed, provides the bulk of work for the optimisation process. The algorithm is capable of computing different aerofoil and planform geometries and is able to map the pressure distribution into the structural model appropriately.

7.7 Summary

In this chapter, several algorithms and methods were applied to practical, real-world examples in aeronautics. Wing design studies ensured the capabilities and feasibility of the proposed evolutionary methods to find robust solutions for multi-objective and multidisciplinary design problems. The design results confirm a computational gain on using a hierarchical topology of fidelity models, compared to a single model during the optimisation. Results also show how the algorithms developed were capable of identifying the trade-off between the multi-physics involved and providing classical aerodynamic shapes, as well as alternative configurations from which the designer can choose and proceed into more detailed phases of the design process. The aero-structural analysis is a robust method that maps the pressure forces with the structural model, and which can be used to evaluate different wing planform shapes, with different aerofoil sections, wing span, aspect and taper ratios.

Chapter 8

Multi-objective and Multidisciplinary Aircraft Design and Optimisation Applications

> *"It is an error to imagine that evolution signifies a constant tendency to increased perfection. That process undoubtedly involves a constant remodeling of the organism in adaptation to new conditions; but it depends on the nature of those conditions whether the directions of the modifications effected shall be upward or downward."* Thomas H. Huxley.

8.1 Introduction

This chapter considers the application of several evolutionary methods and algorithms for multi-objective and multidisciplinary design optimisation problems, and specifically for Unmanned Aerial Vehicles design.

Initially in this chapter, the different analysis tools used in these problems are described. Then four test cases related to aircraft conceptual design with different complexities are evaluated, namely a one-objective subsonic transport aircraft design, a two-objective subsonic aircraft design, a multidisciplinary Unmanned Aerial Vehicle (UAV) design and optimisation and a two-objective air superiority Unmanned Combat Air Vehicle (UCAV) that compares a Pareto optimality and Nash equilibrium approach.

Table 8.1 summarises the test cases evaluated in this chapter. The problems are solved using the methods included in the aircraft design module (Figure 8.1).

Type	**Single-/ Multi-objective**	**Description**	**Solver**
wing	Single-Objective	One-objective UAV – Cargo Transport Design	FLOPS
Direct	Multi-objective	Two-objective UAV – Cargo Transport Design	FLOPS
Direct	Multi-objective	High altitude – Long Endurance (HALE) UAV Design and Optimisation	FLOPS/ VLMpc
Direct	Multi-objective	Two-objective Air Superiority Unmanned Combat Air Vehicle (UCAV), a Pareto Optimality – Game Theory comparison.	FLOPS

Table 8.1: Multi-objective and multidisciplinary UAV design test cases.

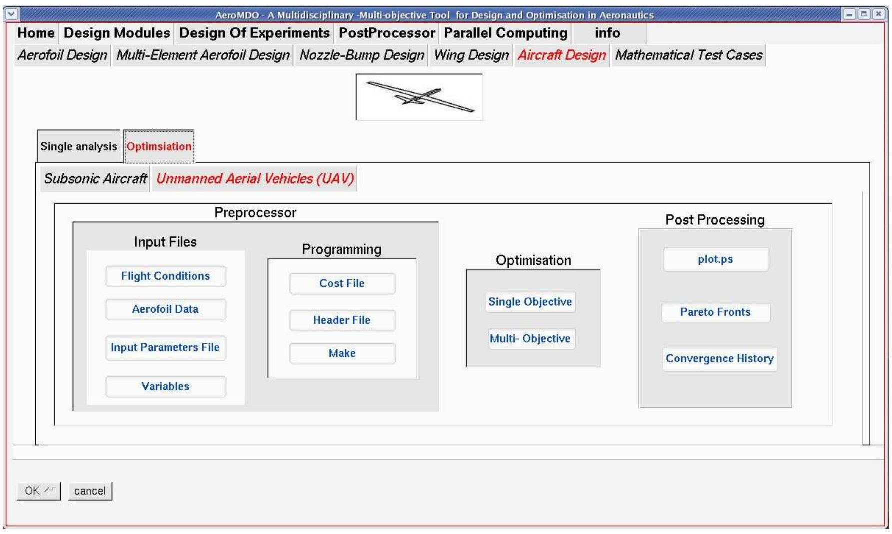

Figure 8.1: Aircraft design module.

8.2 Analysis Tools for Aircraft Design

The solver used to evaluate the aircraft configurations is *FLOPS* (FLight OPtimisation System) developed by McCullers [105]. *FLOPS* is a workstation based code with capabilities for conceptual and preliminary design and evaluation of aircraft concepts. The sizing and synthesis analysis in *FLOPS* are multidisciplinary in nature. It has numerous modules for noise, detailed take-off, performance, structures, control, aerodynamics and other capabilities; it is used in some universities for MDO development as well as aerospace firms and government. It allows an integral analysis for the entire mission and the calculation of aircraft performance parameters such as range, endurance take-off field length and landing field length. The *FLOPS* code also has capabilities for optimisation, but in this case it was used only for analysis and adapted to the evolutionary methods.

8.2.1 Aerodynamic Analysis

The aerodynamic characteristics of the aircraft configurations can be computed internally by *FLOPS* using the Empirical Drag Estimation Technique (*EDET*) or included from an external file. The aerodynamics module in *FLOPS* was enhanced to allow the application of a physics based approach in order to compute the drag polar of the aircraft for the entire mission. In this work, a compromise decision was made in order to demonstrate the working of the evolutionary methods. The approach used a low-medium fidelity vortex-lattice model analysis as opposed to accurate Navier-Stokes computations which are prohibitive in their application in this work due to computational resources limitations.

The introduction of externally generated aerodynamic data allows a better and more accurate computation of the aerodynamic characteristics and is especially important for non-conventional designs such as UAVs. The choice and settings of these aerodynamic codes have a direct influence on the evolutionary optimisation and will be explained in more detail in the following subsection.

The drag calculations are divided into two major groups: vortex-induced drag and viscous drag. In all the applications considered, the flight regime is subsonic during the entire mission, therefore, wave drag calculations are not modelled in the calculations. The interference viscous drag has been accounted for.

The calculation of lift-dependent drag for subsonic configurations is performed using a vortex lattice method (VLM) on the entire aircraft configuration at different Mach numbers and lift coefficients [99]. It includes determination of down-wash on the tail and contribution to pitching moment and static margin. Figure 8.2 shows the vortex location in one of the candidate configurations; the calculation includes modelling of camber/twist and flap deflection.

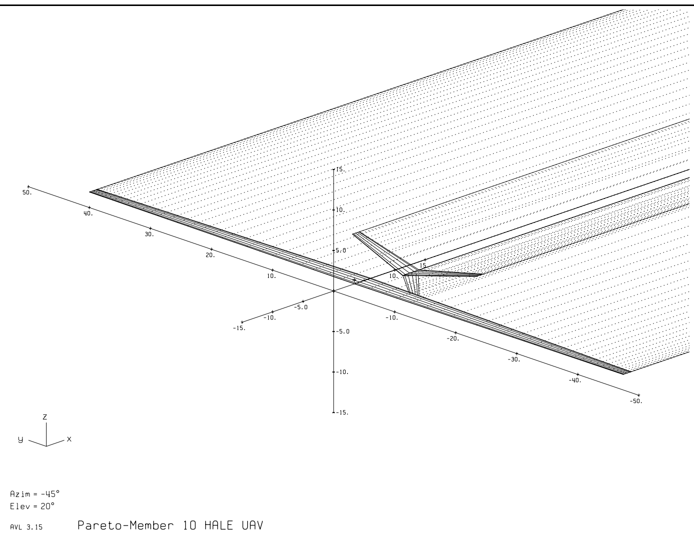

Figure 8.2: Example of trailing filaments in one of the candidate configurations.

The viscous drag that is independent of lift is calculated on each span-wise section using the panel code *XFOIL* [47] or the Euler + boundary layer code *MSES* [46] and integrated over the wing span. The drag on the fuselage and other components is calculated based on the wetted areas and skin friction drag for each component.

8.2.2 Weight Analysis

The aircraft empty weight can be calculated internally by *FLOPS* and consists of the contribution of each aircraft component. However, some known or assumed data, such as systems or engine weight, must be given. Thus the contribution of aircraft weight is related only to the design variables.

8.2.3 Validation of Analysis Tools

The *FLOPS* code was initially validated for known transport and UAV vehicles. A mission analysis check was performed on endurance and range calculations, using known values for specific fuel consumption and horse power for the engine for different aircraft. Once the analysis method was validated, good confidence was obtained for its application and coupling with the optimisation algorithms and optimisers.

8.3 Optimisation Rationale

The general optimisation rationale for externally generated aerodynamic data can be summarised as follows:

1. Define the upper and lower bounds for the aerofoil and aircraft design variables.
2. Compute the flow around the aerofoil sections and obtain a Cd_0 estimate for the wing.
3. Create a vortex lattice model of the candidate geometry while satisfying trim conditions.
4. Compute the drag polar of lift-dependent drag.
5. Compute friction drag on other components, based on wetted areas.
6. Incorporate the drag polar into *FLOPS*.
7. Analyse the configuration for the two objective functions.
8. Send the objective values to the optimiser.
9. Evolve and modify design variables using HAPEA optimiser until stopping criteria is met.

8.4 One Objective UAV – Cargo Transport Design

8.4.1 Problem Definition

The objective in this case was to compare the results using the evolutionary methods and two traditional optimisation techniques, namely, the conjugate gradient based (Polak-Ribiere) algorithm and the Broyden-Fletcher-Goldfarb-Shano (BFGS) algorithm. The test case is a UAV with similar characteristics to manned cargo transport aircraft [74]. The aircraft has two wing-mounted engines. The constraints for this problem are:

- Range > 2500 *nm*,
- maximum allowable take-off field length (FLTO) of 6000 *ft*,
- maximum allowable landing approach velocity of 125 *kts*,
- maximum operating Mach number (VMMO) of 0.8 and
- structural ultimate load factor (ULF) of 4.22.

If any of these requirements are violated, the configuration is immediately rejected prior to analysis. The mission profile is represented in Figure 8.3.

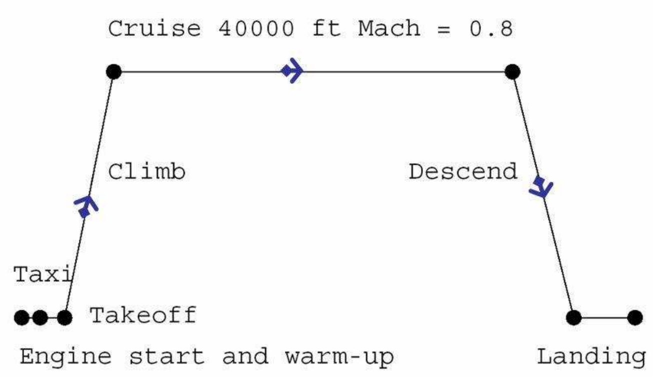

Figure 8.3: Mission profile for UAV cargo transport design.

8.4.2 Design Variables

The design variables for optimisation and its upper and lower bounds are represented in Table 8.2. This choice of design variables is based on cargo aircraft with similar payload capacity, and they are wide enough to provide a wide search space for the optimal solutions.

Description	**Variable**	**Lower bound**	**Upper Bound**
Wing aspect ratio	AR_w	7.0	13.1
Thrust per engine, *Lbf*	$Thrust$	30500	50000
Ref. wing area, *sq ft*	SW	1927	2872
Wing 1/4 chord sweep, *deg*	Λ_w	25	40
Wing thickness-chord ratio	t/c	0.091	0.235
Wing dihedral, *deg*	Γ_w	0.7	–
Wing taper ratio	λ_w	0.27	

Table 8.2: Design variables for a UAV cargo transport design.

8.4.3 Fitness Functions

This case is a single-objective minimisation problem. The fitness function devised for this problem is toward the minimum fuel weight required to complete the mission.

$$f = min\,(W_f) \tag{8.1}$$

8.4.4 Implementation

Two implementations are considered, one using conventional gradient-based optimisation tools and one using the HAPEA evolutionary algorithm. The traditional approaches were started from different points. The evolutionary approach was implemented using different parameter settings.

8.4.5 Computational Results

The evolutionary algorithm was allowed to run for 1500 function evaluations. Figure 8.4 is an example of convergence for this problem. Small population sizes produced good results.

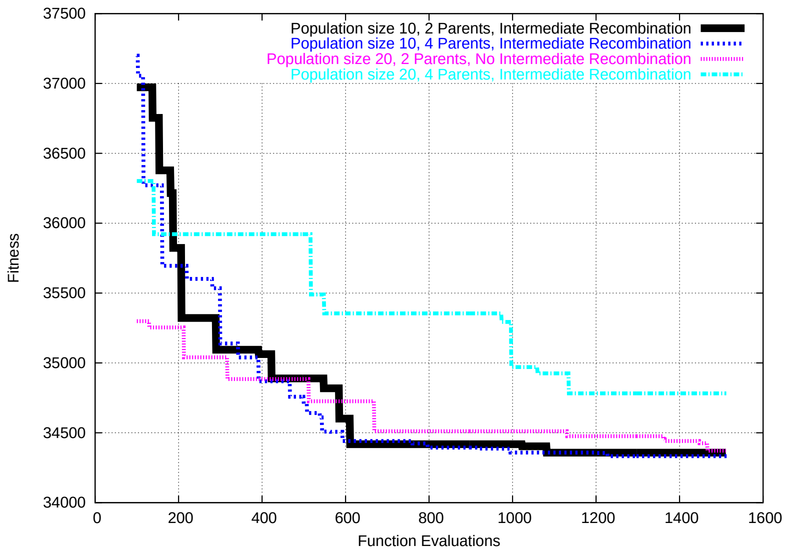

Figure 8.4: Optimisation progress using different HAPEA parameters for UAV cargo transport design.

Table 8.3 shows the design variables and results for the best configuration found, compared to the best results obtained by the conjugate gradient and the BFGS algorithms. This is an important result which illustrates some of the benefits of using the evolutionary method; the EA produces a 3.5% reduction in gross weight when compared to the conjugate gradient, and a 2.4% gross weight reduction when compared to the BFGS.

This study, although performed at a conceptual design level, with a low number of design parameters and low-to-medium fidelity analysis tools, shows the potential benefits and merits of these evolutionary methods for aircraft design and optimisation problems. In the following test cases, more complicated multi-objective and multidisciplinary design optimisation problems with higher fidelity analysis tools will be considered.

Description	Variable	HAPEA Best	(BFGS)	Conjugate Gradient
Wing aspect ratio	AR_w	13.1	13.0	12.8
Thrust per engine, *Lbf*	$Thrust$	34770	38852	39021
Ref. wing area, *sq ft*	SW	1929	2142	2218
Wing 1/4 chord sweep, *deg*	Λ_w	27.0	28.4	27.32
Wing thickness-chord ratio	t/c	0.091	0.112	0.096
Wing taper ratio	λ_w	0.267	0.267	0.267
Fuel weight, *Lbs*	FW	34337	37342	36092
Range, *nm*	R	2500	2500	2500
Gross weight, Lbs	GW	***216702***	***222154***	***224618***

Table 8.3: Comparison of optimal design variables and fitness functions after optimisation (HAPEA, BFGS and Conjugate Gradient).

8.5 Two Objectives UAV – Cargo Transport Design

8.5.1 Problem Definition

Aerodynamic performance, cost minimisation and range might be improved if a multi-objective multi-point optimisation can be developed that considers numerous separate design points. The goal in this case is to address the previous test case, but from a multi-objective perspective. The objectives are minimisation of gross weight and maximisation of the cruise efficiency $\left(M_\infty \times \frac{L}{D}\right)$. The design variables, requirements, constraints and mission profile are the same as for the previous case. The fitness functions to be optimised are:

$$\begin{aligned} f_1 &= min\ (W_g) \\ f_2 &= min\ \left[\frac{1}{\left(M \times \frac{L}{D}\right)_{cruise}}\right] \end{aligned} \tag{8.2}$$

8.5.2 Implementation

The solution to this problem has been implemented using a single population of size 20, buffer size of 60 and tournament size of 2.

8.5.3 Computational Results

The algorithm was allowed to run for 5000 function evaluations. The final population (including the Pareto optimal set) is shown in Figure 8.5; a well distributed Pareto optimal set can be seen.

For comparison purposes, three aircraft configurations are taken from the Pareto front: configuration PM0 – objective one optimal, configuration PM10 – compromise and configuration PM19 – objective two optimal. A summary of the objective and design variables values is presented in Table 8.4. From an analytical point of view, it is clear that even without a priori knowledge of the configurations, the evolutionary method has produced at the two extreme points classical aircraft variables settings for minimum gross weight (small wing area, aspect and taper ratio and high wing t/c) and high performance values (larger wing area, aspect ratio and taper ratio, and smaller wing t/c). The results obtained provide the designer with a starting point, from which higher fidelity models and complexities related to preliminary design can be introduced.

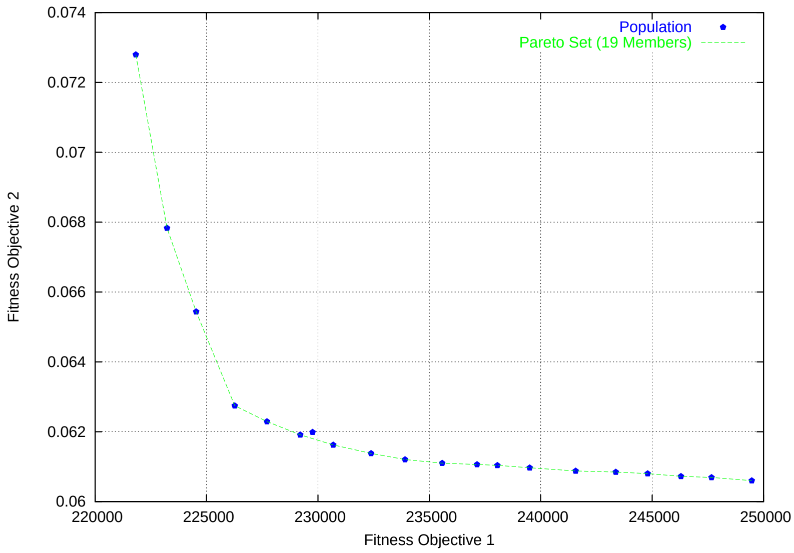

Figure 8.5: Pareto optimal region for UAV cargo transport design.

Description	Variable	PM 0	PM10	PM19
Wing aspect ratio	AR_w	9.44	13.07	13.07
Thrust per engine, *Lbf*	$Thrust$	48925	48925	48925
Ref. wing area, *sq ft*	SW	1938	2307	2362
Wing 1/4 chord sweep, *deg*	Λ_w	25	28	28.25
Wing thickness-to-chord ratio	t/c	0.096	0.091	0.091
Wing taper ratio	λ_w	0.15	0.20	0.41
Gross weight, Lbs	$W,\ lbf$	***221826***	***232722***	***246657***
Cruise efficiency	$\left(M_\infty \times \frac{L}{D}\right)$	***13.74***	***16.36***	***16.47***

Table 8.4: Optimal design variables and objective functions results for a UAV cargo transport design.

8.6 High-altitude – Long-endurance (HALE) UAV Design and Optimisation

This study considers a high-altitude long-endurance (HALE) UAV for scientific research that carries a payload between 600 and 720 *lbs* and has endurance above twenty-four hours. This aircraft is similar to the Altair or Predator B UAVs [74]. The design characteristics consist of a single fuselage and *Y* tail design. The power plant is a single-shaft turboprop with 900 *shp* and *sfc* of 0.558 *lbs/shp/hr.* The aircraft construction is mainly of composites. The mission profile is represented in Figure 8.6. Requirements, which are based on similar aircraft of this type [74, 111] include:

- landing at 55% of maximum take-off weight,
- cruise altitude between 40000 and 52000 *ft*,
- structural ultimate load factor (ULF) of 4.22,
- maximum allowable take-off field length (FLTO) of 1000 *ft* and
- maximum allowable landing approach velocity of 80 *kts*.

If any of these constraints are violated, the configuration is immediately rejected prior to analysis.

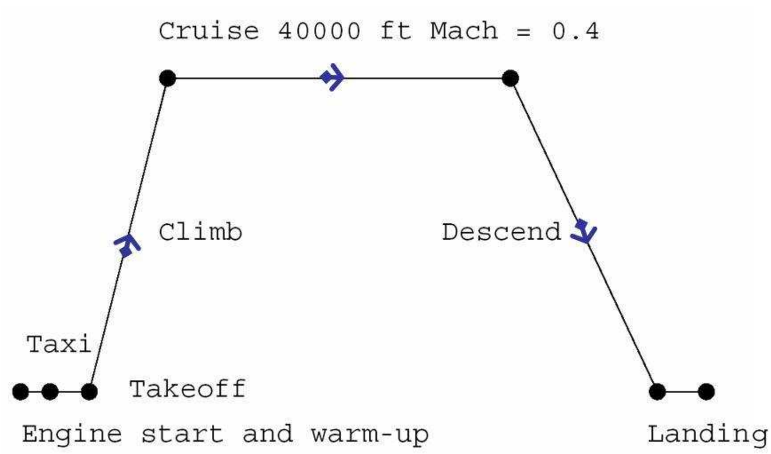

Figure 8.6: Mission profile for a HALE UAV design.

The aircraft performance, cost minimisation and range might be improved if a multi-criteria multi point optimisation that considers numerous separate design points is developed. The challenge in this case is to address the problem considering two objectives: fuel weight minimisation and maximisation of endurance. The objective is to obtain a family or Pareto front of aircraft configurations that will represent the trade-off between the two objectives.

$$\begin{aligned} f_1 &= min(W_f) \\ f_2 &= min\ \left[\frac{1}{E}\right] \end{aligned} \tag{8.3}$$

The main design variables for the aircraft and its upper and lower bounds are represented in Figure 8.7 and table 8.5. The aerofoil geometry is represented by six points on the mean line and ten points on the thickness distribution. In total, twenty-eight design variables are used during the optimisation.

This choice of design variables and their upper and lower bounds is based on similar aircraft and provides a good representation of the external geometry and wide search space, while allowing the creation of a representative vortex lattice model of the aircraft.

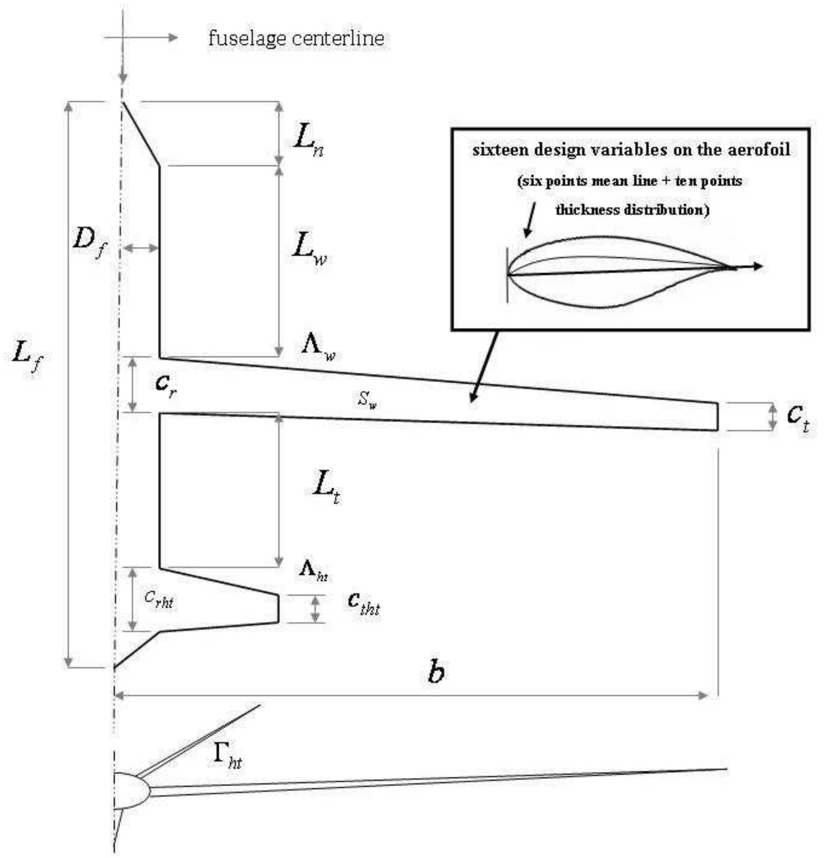

Figure 8.7: Design variables for a HALE UAV design.

Description	Variable	Lower bounds	Upper bounds
Wing area, *sq ft*	SW	280.0	330.0
Wing aspect ratio	AR_w	18.0	25.0
Wing 1/4 chord sweep, *deg*	Λ	0.0	8.0
Wing taper ratio	λ_w	0.28	0.80
Horizontal tail area, *sq ft*	SW_{ht}	65.0	85.0
Horizontal tail aspect ratio	AR_{ht}	3.0	10.0
Horizontal tail taper ratio	λ_{ht}	0.20	0.55
Horizontal tail sweep,*deg*	Λ_{ht}	12.0	15.0
Horizontal tail dihedral, *deg*	Γ_{ht}	35.0	35.0
Vertical tail area, *sq ft*	SW_{vt}	11.0	29.0
Aspect ratio vertical tail	AR_{vt}	1.0	3.2
Vertical tail taper ratio	λ_{vt}	0.28	0.62
Vertical tail sweep, *deg*	Λ_{vt}	12.0	34.0
Fuselage diameter, *ft*	D_f	2.6	5.0

Table 8.5: Design variables for a HALE UAV design.

8.6.1 Hierarchical Topology Implementation

A method was developed with a hierarchical topology of solver resolutions for the aerodynamics analysis. A hierarchical topology of grid densities on the two CFD solvers is implemented.

- *Top Layer:* Population size 20, intermediate recombination, a 141 $\times$ 74 $\times$ 36 grid on the aerofoil and a 20 $\times$ 6 grid on the vortex model.
- *Middle Layer:* Population size 20, intermediate recombination, a 109 $\times$ 57 $\times$ 27 grid on the aerofoil and a 17 $\times$ 6 grid on the vortex model.
- *Bottom Layer:* Population size 20, intermediate recombination, a 99 $\times$ 52$\times$25 grid on the aerofoil and a 15 $\times$ 6 grid on the vortex model.

8.6.2 Computational Results

The optimisation was allowed to run for 1200 function evaluations. Convergence for objective one on each topology layer is shown in Figure 8.8. The final population (including the Pareto optimal set) is shown in Figure 8.9. A well-distributed Pareto optimal set can be seen. For comparison purposes, three aircraft configurations are taken from the Pareto front: configuration PM 0 – objective one optimal, configuration PM 2 – compromise and configuration PM 5 – objective two optimal. A top view of these configurations is shown in Figure 8.10 and a summary of the objective and design variables is provided in Table 8.6.

From an analytical point of view it is clear that even without a priori knowledge of the aircraft configurations, the evolutionary method has identified the trade-off between the two objectives. The design variables for PM0 constitute optimal design variables for minimum fuel weight; the design variables for PM19 constitute those for maximum endurance and the design variables for PM10 are a combination of design variables for a compromise individual. All these configurations satisfy the endurance landing-field length and cruise trim constraints.

In a practical sense, these results provide the designer or team of designers with a starting point; they can select this compromise design or a set of compromise designs and proceed with further analysis, such as a full Navier-Stokes analysis, wind-tunnel testing or a flight simulation model wherein additional complexities can be analysed.

Compromise individual PM10 is taken for further evaluation. Figure 8.11 illustrates the geometry for the aerodynamic analysis and Figure 8.12 shows one of the sample Trefftz Planes and some aerodynamic characteristics during the computation of the drag polar for this configuration.

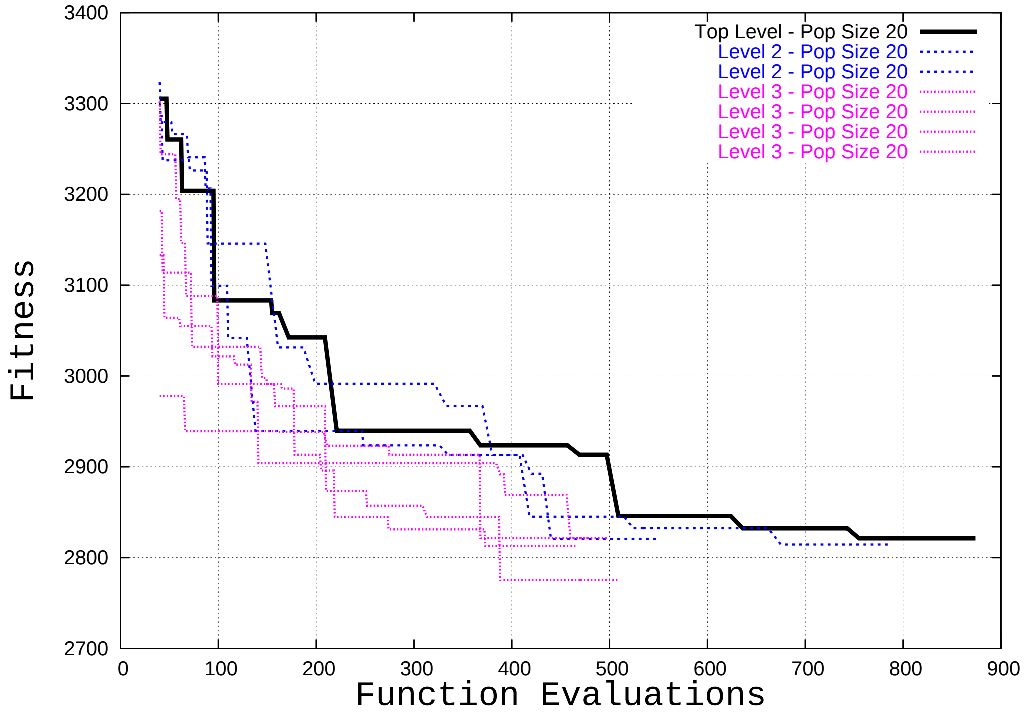

Figure 8.8: Convergence objective one for a HALE UAV design.

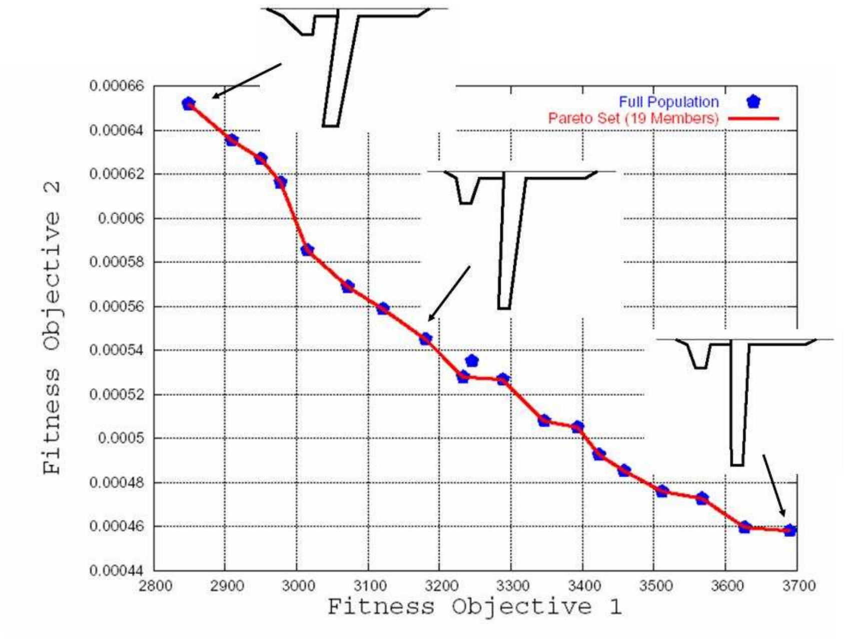

Figure 8.9: Pareto optimal front for a HALE UAV design.

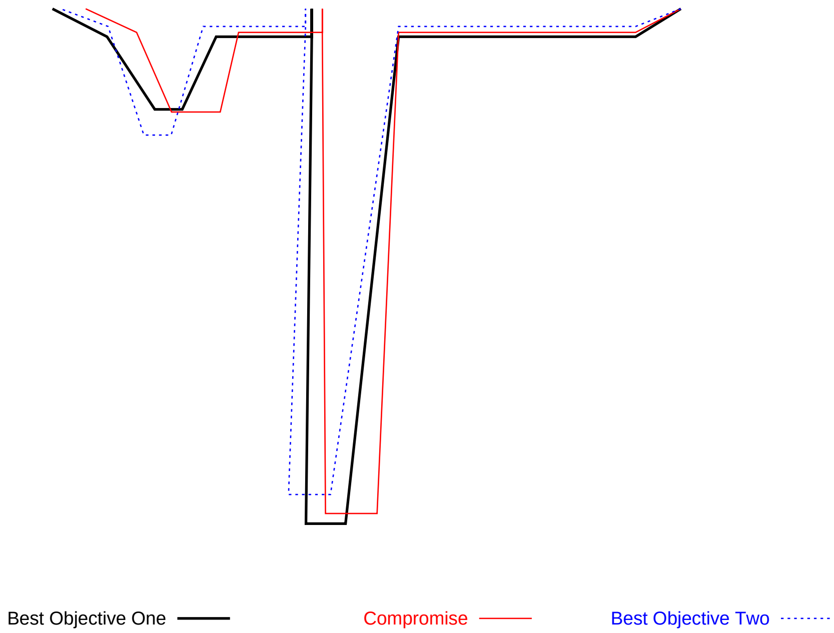

Figure 8.10: Top view of best solutions and compromise individual for a HALE UAV design.

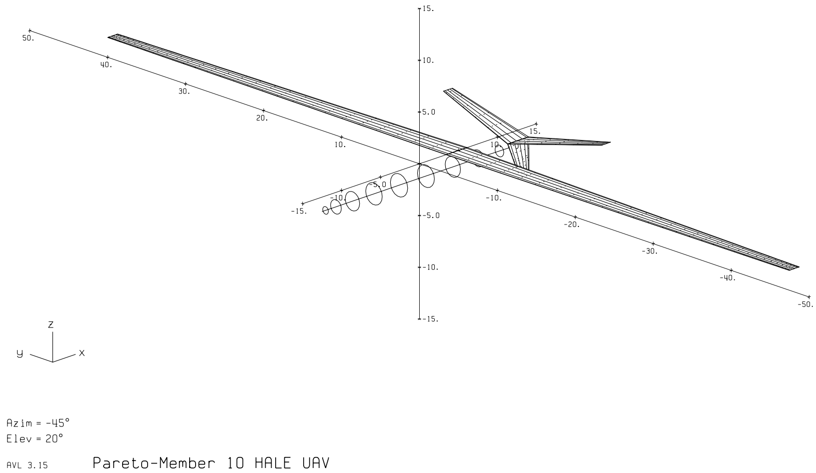

Figure 8.11: Pareto Member ten HALE UAV geometry for aerodynamic analysis.

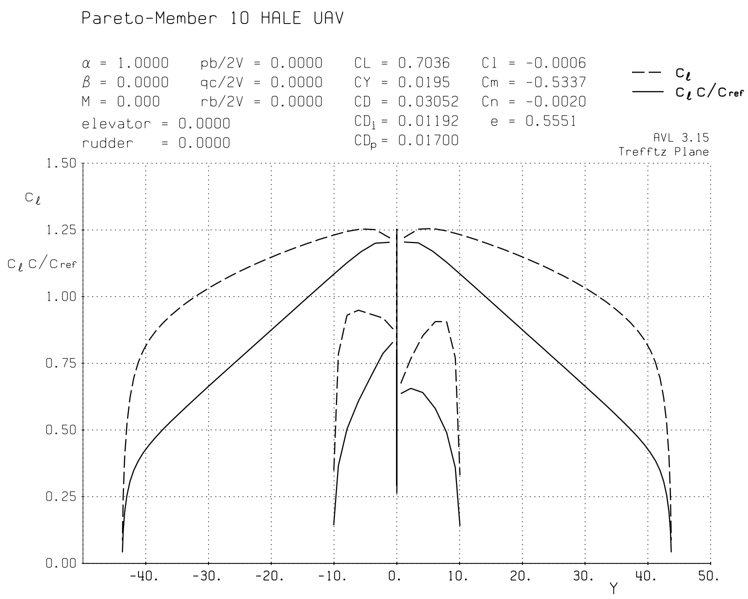

Figure 8.12: Trefftz plane plot for a HALE UAV design – Pareto Member ten.

Description	**Variable**	**PM 0**	**PM10**	**PM19**
Wing area, *sq ft*	SW	301	322	321
Wing aspect ratio	AR_w	18.93	23.77	24.30
Wing 1/4 chord sweep, *deg*	Λ	7.07	4.87	2.78
Wing taper ratio	λ_w	0.71	0.48	0.46
Wing thickness thickness-to-chord ratio	t/c	0.066	0.071	0.081
Wing span, *ft*	b	75,53	87.46	88.36
Horizontal tail area, *sq ft*	SW_{ht}	80.24	75.97	79.33
Horizontal tail aspect ratio	AR_{ht}	3.44	5.43	5.62
Horizontal tail taper	λ_{ht}	0.34	0.43	0.41
Horizontal tail sweep, *deg*	Λ_{ht}	3.49	10.75	5.79
Vertical tail area, *sq ft*	SW_{vt}	19.21	21.03	22.31
Aspect ratio vertical tail	AR_{vt}	1.67	2.32	2.65
Vertical tail taper ratio	λ_{vt}	0.49	0.47	0.45
Vertical tail sweep, *deg*	Λ_{vt}	17.58	23.10	31.30
Fuselage diameter, *ft*	D_f	4.84	4.18	3.02
Fuel weight, Lbs	FW	***2978***	***3150***	***3485***
Endurance, nm	E	***1533***	***1846***	***2184***

Table 8.6: Optimal design variables and objective function results for a HALE UAV design.

8.6.3 Flight Simulation Analysis

It was also decided to evaluate the performance characteristics of this configuration (PM10), using the flight simulation program X-PLANE [106]. The starting point for generating the models is the output file from the optimisation similar to that included in Appendix B. This file contains detailed information about the geometric and aerodynamic characteristics of the airplane. The CAD model for this configuration is shown in Figure 8.13. Figure 8.14 shows some of the flight simulation models in X-PLANE. A video of the flight simulation with the performance of the aeroplane is included in the CD placed in the back-cover of this thesis.

Results of the performance of the aircraft during the simulation are also in agreement with the results obtained during the optimisation process.

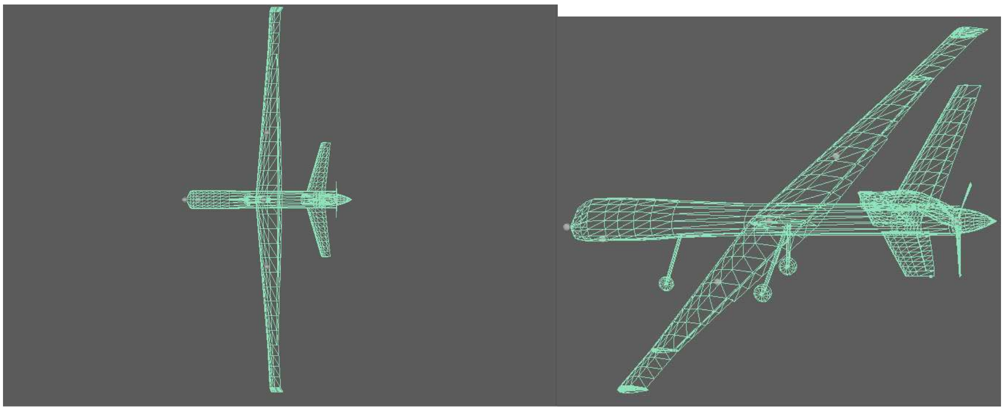

Figure 8.13: Top and side view of Pareto Member ten.

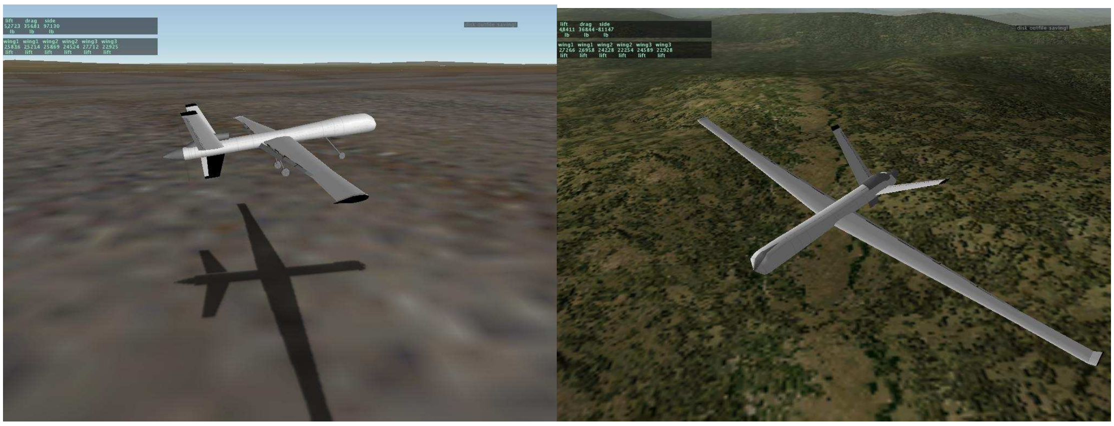

Figure 8.14: X-Plane flight simulation model for Pareto Member ten.

8.7 Two Objectives Air Superiority Unmanned Combat Aerial Vehicle (UCAV), a Pareto Optimality – Game Theory Comparison

8.7.1 Problem Definition

Unmanned Combat Aerial Vehicles are also an important technology under development. The goals in a military application are to develop potential UCAV, capable of performing several missions, including SEAD/ Strike/ Electronic Attack, and the capability of growth for other missions. The benefits of weight and volume reduction may be achieved by UCAVs compared to manned aircraft carrying the same weapons or sensor payloads. Also the manoeuvrability of a UCAV could be greater than that of a manned aircraft with higher g turns.

The goal here is then to address that issue from the conceptual design analysis and develop a method and algorithm for this task. This case is a multi-objective problem, wherein the objectives are maximisation of performance $(Mach\ \times L/D)$ and minimisation of gross weight (W_g).

Two multi-objective approaches are considered for evaluation: Pareto Optimality and Nash equilibrium.

The design requirements and constraints are specified in Table 8.7 and the mission profile is represented in Figure 8.15.

Description	**Requirement**	**Value**
Range, *nm*	R	1000
Cruise Mach Number	$Mach$	1.6
Cruise altitude, *ft*	Alt	40000
Ultimate Load Factor	ULF	12
Upper limit take-off field length, *ft*	$MTOL$	7000

Table 8.7: Design requirements for a UCAV design.

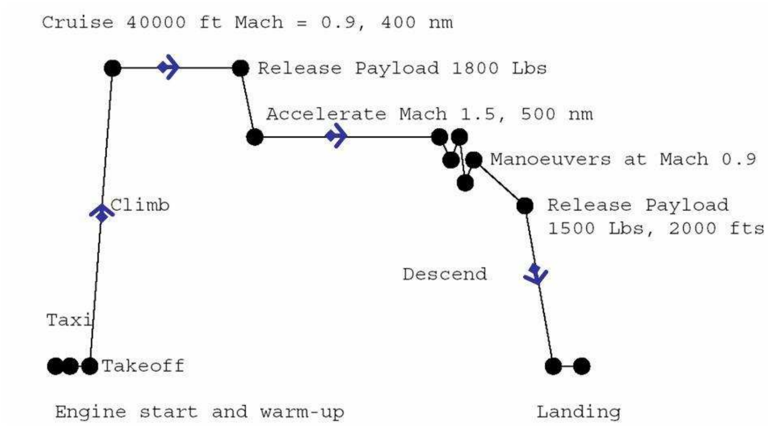

Figure 8.15: Mission profile for a UCAV design.

8.7.2 Design Variables

The design variables for optimisation and its upper and lower bounds are represented in Table 8.8. The upper and lower bounds are based on combat aircraft of similar characteristics and are wide enough to provide a large search space for the optimal solutions [74].

Description	Variable	Lower Bounds	Upper Bounds
Wing aspect ratio	AR_w	3.1	5.3
Thrust per Engine, *Lbf*	$Thrust$	32000	37000
Ref. wing area, *sq ft*	SW	600	1400
Wing 1/4 chord sweep, *deg*	Λ_w	22	47
Wing thickness-chord ratio	t/c	0.02	0.09
Wing dihedral, *deg*	Γ_w	0.15	0.55

Table 8.8: Design variables for a UCAV design.

8.7.3 Fitness Functions

This case is a multi-objective problem, wherein the fitness functions to be optimised are defined as:

$$\begin{array}{c} f_1 = min(W_g) \\ f_2 = min \left[\frac{1}{Mach * L/D} \right] \end{array} \tag{8.4}$$

8.7.4 Design Constraints

The performance constraints, which are based on similar aircraft, are:

- 6 G's at Mach 0.6 at 10000 feet,
- 5 G's at Mach 0.9 at 30000 feet,
- accelerate from Mach 0.9 to 1.5 at 20000 feet in 30 seconds,
- maintain turn rate of 15 *deg/sec* at Mach 0.9 at 20000 feet and
- Excess energy = 50 *ft/sec* at Mach 0.9 at 30000 feet at 4 G's.

If any of these constraints are violated, the configuration is immediately rejected prior to analysis.

8.7.5 Implementation

The Pareto Optimality implementation consists of a single population of size 20 and buffer size 60. For the Nash approach we split the variables between two players; Player One maximises cruise performance $Mach \times L/D$ using $(V1) = (AR_w, tca \ and \ \Lambda_w)$ as design variables while Player Two minimises gross weight, W_g using $(V2) = (SW, \ Thrust \ and \ \lambda_w)$. Only one hierarchical level is used for each player; information is exchanged between the players and EAs after the epoch has terminated; in this case it is equal to five times the population size evaluations. The multidisciplinary analysis for this case is computed solely by FLOPS; that is, the aerodynamic data and other analysis are computed internally by FLOPS.

8.7.6 Computational Results

The algorithm was allowed to run for 600 function evaluations but converged after 300. Figure 8.16 shows an example of convergence history for Player One, and Figure 8.17 that for Player Two. The final population (including the Pareto optimal front) and comparison with the Nash equilibrium result are shown in Figure 8.18. For evaluation purposes three points on the Pareto front and the Nash solution are considered. Figure 8.19 shows a top view of the wing planforms for Pareto members 1, 3 and 7 and the Nash equilibrium solution. Table 8.9 compares the design variables and value for objective functions.

The use of the Nash equilibrium approach seems to enable the splitting of the optimisation problem and provide feasible results, but it can be seen that the point obtained by the Nash equilibrium is a suboptimal solution in the Pareto sense.

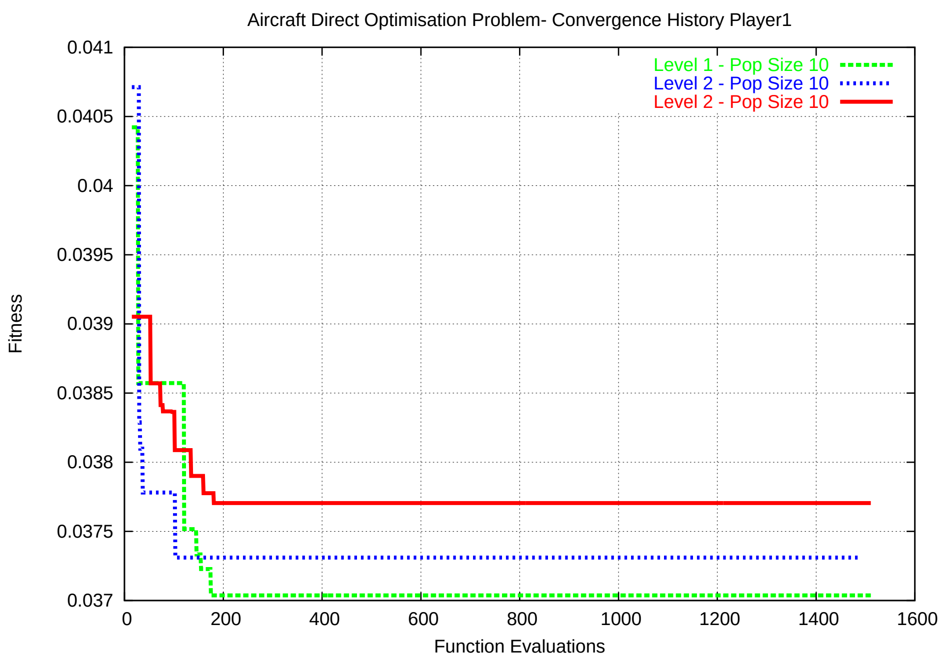

Figure 8.16: Optimisation progress for Player One.

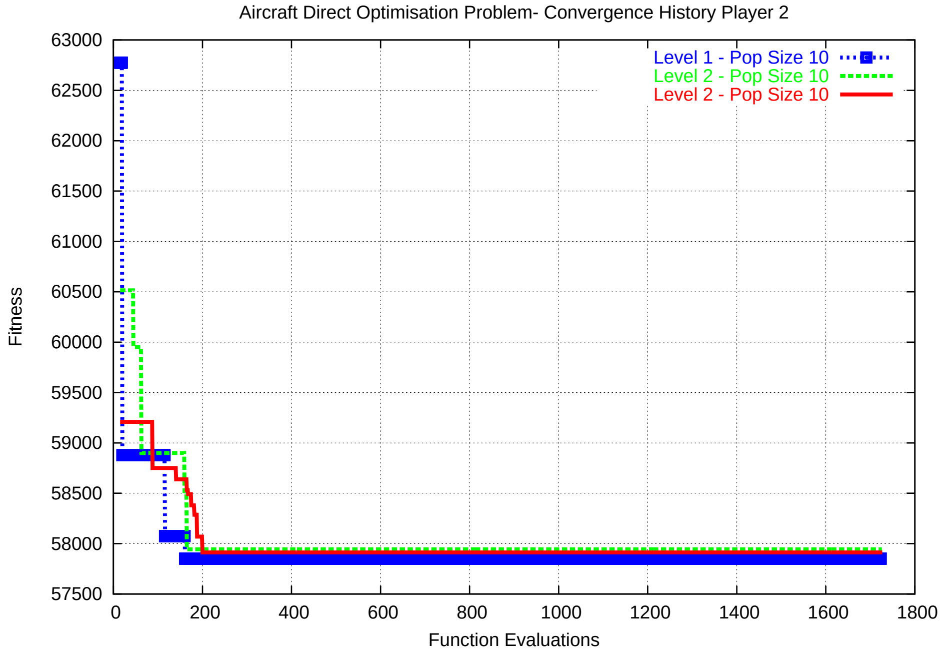

Figure 8.17: Optimisation progress for Player Two.

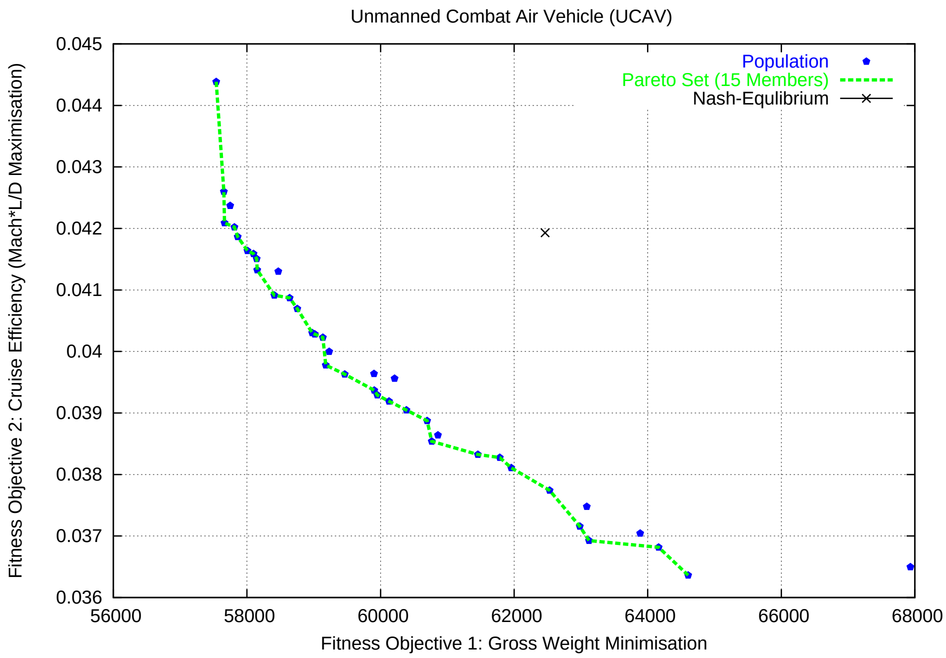

Figure 8.18: Pareto front and Nash equilibrium for a UCAV design.

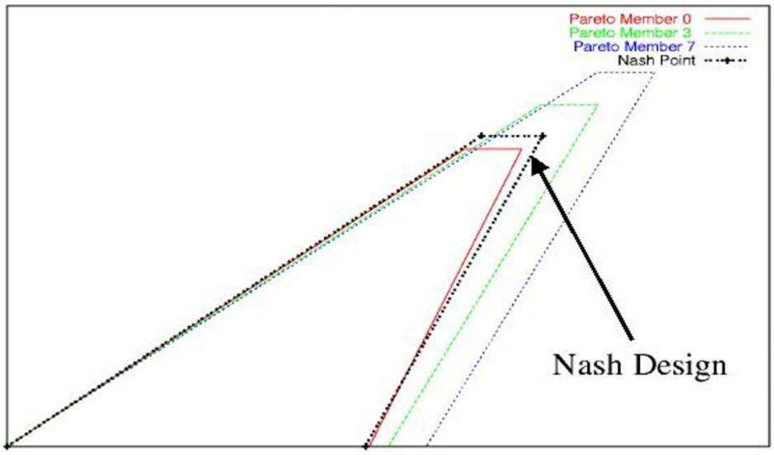

Figure 8.19: Top view of some Pareto front wing geometries and Nash equilibrium for a UCAV design.

Description	**Variable**	***PM 0***	**PM 3**	**PM 7**	**Nash**
Wing aspect ratio	AR_w	4.76	5.23	5.27	5.14
Ref. Wing area, *sq ft*	SW	629.7	743.8	919	618
Wing thickness	t/c	0.046	0.050	0.041	0.021
Wing taper ratio	λ_w	0.15	0.16	0.17	0.17
Wing 1/4 chord sweep, *deg*	Λ_w	28	25	27	28
Thrust per engine, *Lbf*	$Thrust$	320650	32219	32259	33356
Gross weight, Lbs	WE	***57540***	***59179***	***64606***	***62463***
Cruise performance	$M_{cruise} \cdot (\frac{L}{D})_{cruise}$	***22.5***	***25.1***	***27.5***	***23.9***

Table 8.9: Comparison of optimal design variables for three members of the Pareto front and Nash equilibrium.

8.8 Summary

In this chapter four different configuration problems have been analysed. A method was created for each case. The test cases demonstrate that the proposed evolutionary methods are well suited for multidisciplinary aircraft design and optimisation problems.

When compared to traditional optimisation methods, the evolutionary algorithm approaches developed are capable of exploring the solution space better and providing feasible alternative solutions for single- and multi-objective concepts. In one of the test cases considered, the evolutionary approach produces a 3.5% reduction in gross weight when compared to the conjugate gradient, and a 2.4% gross weight reduction when compared to the BFGS. The evolutionary method and framework can reduce the weight and cost of an aircraft concept by minor changes in the design variables.

The use of the Pareto optimality and Nash approach for multi-objective aircraft conceptual design problems has also been investigated. It seems to enable the splitting of the optimisation problem and provide feasible results.

The benefits of using the methodology, framework, GUI, parallel computing and evolution algorithms to provide solutions for single and multi-criteria aircraft multi-disciplinary design optimisation problems were demonstrated.

Chapter 9

Conclusions and Future Work

"The ability to perceive or think differently is more important than the knowledge gained." David Boh.

9.1 Overview

This chapter briefly summarises the main contributions of this thesis and provides suggestions for future research directions.

9.2 Summary of Contributions

The primary objective of this thesis was to design, develop and test evolutionary methods and a framework which would enable the analysis and optimisation of multi-objective and multidisciplinary optimisation problems in aeronautics. Clearly the methods and framework meet these goals.

The fundamental reason for their development was to overcome some of the difficulties with traditional approaches to multidisciplinary design and optimisation (MDO) problems in aeronautics which rely on traditional deterministic gradient based optimisation techniques. The initial research into the design of the methods focused on identifying the main requirements. The methods were then successfully applied to mathematical test cases of different complexities and then to real-world problems representative of the aircraft design process. These problems included aerodynamic shape optimisation of aerofoils and wings as well as conceptual and preliminary studies on MDO and aircraft design. The specific contributions are:

9.2.1 Implementation

A series of requirements for robust evolutionary methods were identified, and several algorithms were created and tested. The methods allow single or multidisciplinary design optimisation studies using analysis tools of different fidelities and computational expense. The evolutionary methods described have the potential to reduce computing time at early stages of the design process.

9.2.2 Mathematical Test Functions

A substantial contribution of this thesis was to generate a series of methods and algorithms for testing mathematical test functions. These functions ranged from simple functions to complex multi-modal deceptive, constrained, goal-programming problems with convex, non-convex or discontinuous Pareto optimal fronts. The performance of the algorithms with different parameter settings was investigated, giving the designer insight and an indication of the parameter settings of the algorithms for real-world problems which might exhibit similar behavioural characteristics. The study provided and extended the understanding and capability of the evolutionary methods and algorithms.

9.2.3 Aeronautical Design and Optimisation

The feasibility of the methods for aeronautical design problems was demonstrated by investigating several real-world single-, multi-objective and multidisciplinary design optimisation problems in aeronautics. These studies highlighted the practicality and benefits of the methods developed and implemented. Particular instances are:

9.2.3.1 Aerofoil Design and Optimisation Methods

A set of design and optimisation methods were developed for aerofoil design. The study of these methods demonstrated their practical use in finding optimal solutions for inverse and direct problems. It was also shown how the methods benefited from a combination of fidelity-solvers and parallel computing strategies. Non-intuitive as well as “aerodynamic expert” type designs were obtained without a priori knowledge of the solution. When compared to existing aerofoils, the methods have produced shapes with significant improvements in drag reduction.

9.2.3.2 Multi-element Aerofoil Design and Optimisation Methods

Studies in developing and applying evolutionary methods for multi-element aerofoil design and optimisation highlighted the complexities on the topology representation and grid definition when coupling these methods with a robust Euler or Navier-Stokes flow-solver. The use and coupling of mesh adaptation during the optimisation process was also investigated. Results indicate that even though the computational burden of mesh adaptation is high during the optimisation process, it is important to consider it, as appropriate features of the flow field such as recirculation could not be identified otherwise.

9.2.3.3 Multidisciplinary and Multi-objective Wing Design and Optimisation Methods

Practical methods for multi-objective wing design problems were developed; the use of the Pareto approach produced classical aerodynamic shapes as well as compromised solutions. The benefits of using a hierarchical topology for multi-objective wing design was demonstrated. The aero-structural analysis module, its methods and algorithms provide a robust means to evaluate candidate wing configurations. A robust algorithm maps the aerodynamic load into the structural model.

9.2.3.4 Multidisciplinary Aircraft Design and Optimisation

In this work, several studies on Multidisciplinary Aircraft Design and Optimisation were performed. When compared to traditional optimisation methods such as BFGS or conjugate gradient, the methods are capable of exploring the solution space better and providing feasible alternative solutions for single- and multi-objective concepts.

A comparison of the merits and drawbacks of the Nash and Pareto Optimality approaches for aircraft conceptual design was undertaken. Results so far indicate the benefit of using a Nash approach if a rapid exploration of the design space is desired, but if a robust solution is sought, extra computational expense is required to obtain a global Pareto front.

A variable fidelity model implementation using Evolutionary algorithms for aircraft design was proposed and evaluated. The coupling of the methods with a medium-to-low fidelity solver to generate the drag polar for the entire mission of each design candidate during the optimisation process was illustrated. The benefit of this approach is different from a traditional approach wherein a fixed point, usually a cruise design point, is considered. In this approach the aircraft is evaluated for its entire mission, allowing a wider diversity in the search space.

One important feature of the application of the methods for multidisciplinary aircraft design and optimisation is its broad applicability. Optimal results were found for different types of problems and types of aircraft: namely, a subsonic aircraft, UCAV and UAV.

9.2.3.5 Modules in Progress

As designed, the method allows ease of implementation of new design and optimisation modules. Ongoing work on different modules is underway, current developments include a multi-objective wing design for a high-altitude long-endurance UAV. Several modules are being developed in cooperation with undergraduate and postgraduate students; these include modules for re-entry vehicle design optimisation using Direct Simulation Monte-Carlo Methods, trajectory optimisation, propeller design, aerodynamic shape optimisation using a grid-free flow-solver [8] and drag reduction on adaptive aerofoils using Shock Control Bump (SCB) theory [92, 163].

9.2.4 Multi-objective Problems

Several multi-objective problems were investigated and Pareto optimal fronts were obtained for convex, non-convex, discontinuous and constrained mathematical test functions. Optimal Pareto fronts were also obtained for the real-world problems considered. The advantage of the Pareto approach is that each Pareto solution corresponds to a specific set of weight factors. Hence the approach can obtain multiple solutions in a single run (simultaneously) without a priori definition by the user.

The Nash equilibrium approach was implemented and studied with the evolutionary methods. The use of this approach allows a single rapid exploration of the design space, compared to a more time-consuming but robust Pareto optimality approach. The benefits of a Nash equilibrium approach are in exploring the most relevant design variables to a particular fitness function. Without a-priori knowledge the algorithm can identify the most relevant variables for a particular fitness function, hence relieving the designer of a pre-defined biased selection.

9.2.5 Framework Architecture and GUI

The framework comprises a GUI, an optimisation software, a mathematical test-suite for validation purposes and several methods and modules for aircraft component and configuration design. The framework was devised as a tool that can be easily revised or modified to incorporate additional optimisation modules and as a platform for future and ongoing projects at the School. The framework also allows implementation of several post-processing or pre-processing tools.

It also enables a single-point evaluation design or optimisation of notional aircraft configurations or components; the graphical interface allows ease of implementation of analysis codes, optimisation routines and pre- or post-processing tools. The framework provides a user-friendly architecture wherein different analysis models; optimisation routines, parallel computing and pre- and post-processing can be incorporated. A good part of this work was devoted to creating simple but comprehensive graphic tools that summarise the design candidates at the end of the optimisation process.

9.2.6 Analysis Tools

The importance of a good understanding of the benefits and drawbacks of some analysis tools was also studied in this research. The selection and coupling of an appropriate design tool with an optimiser is of paramount importance for understanding the quality and type of solutions obtained after the optimisation process. Validation studies of the different analysis tools were also an important contribution to this work.

9.3 Future Work

This thesis has presented comprehensive theory and numerical results supporting the use of evolutionary algorithms for multi-objective and multidisciplinary design optimisation problems in aeronautics. Although comprehensive, further studies are required on architectural issues, GUI development, algorithms, mathematical test functions, design of experiments, and multidisciplinary design optimisation.

9.3.1 Framework and GUI

Further work on the development of the framework is expected, adding capabilities in different modules and full benefits of object-oriented principles. The GUI is simple and can be refined to accommodate additional database management capabilities, data-mining techniques such as SOP, and save and retrieve data from previous studies.

9.3.2 Higher Fidelity Modelling and Parallel Computing

Most of the optimisation studies presented in this thesis have focused on using low-to-medium-fidelity analysis tools. Additional studies and applications using higher fidelity models which account for the complexity in the system are required. These studies require higher computational expense. In order for these studies to be of practical use, a "massive parallelisation" in the order of 100 processors is required.

9.3.3 Design of Experiments Theory

The basic setup for a Design of Experiments using DACE/Kriging was incorporated within the current framework for simple mathematical functions. Implementation and applications of other approximate methods such as Neural Networks, Response Surface Methods, or a hybrid combination with Evolutionary algorithms are an area of interest. The design of experiment theory offers many alternatives to reduce computational expense, some DOE methods such as minimum bias experimental design and space-filling and DACE/Kriging modelling, polynomial response surface, can be easily incorporated into the framework.

9.3.4 Test Suite and Algorithms

Although a comprehensive test suite of mathematical test cases was investigated, further research in other complex functions that involve discrete, continuous variables is required. Work on different optimisation algorithms is also desirable. One of the growing areas of research is the coupling of evolutionary techniques with adjoint methods. Other multi-objective approaches such as min-max or use of preferences could also be investigated.

9.3.5 Pre- and Post-processing

An area for further improvement is that of pre- and post-processing visualisation tools. Candidate intermediate or optimal solutions can be exported to CAD or CAM programs for further evaluation.

9.3.6 Design Modules and Analysis Tools

Different developments in design modules for aeronautical design problems are envisaged.

With regard to aerofoil design, further studies can be into developing applications on adaptive aerofoils with drag reduction, using Shock Control Bump theory. Cascade aerofoil design is another area of application, in addition to aerodynamic shape optimisation using grid free flow solvers.

Design modules for UAV applications can also be those for flight-path, control or navigation optimisation. A module can also be developed to apply these evolutionary methods for small propeller design and optimisation for small UAVs, using a combination of fidelity-analysis tools using blade element theory and helicoidal vortex model theories.

It is also desirable to develop additional design modules and interfaces for advanced or revolutionary aircraft configurations, such as morphing-wing UAVs or blended wing body aircraft. These include developments in weight estimation techniques and structural FEA that can be coupled with the FLOPS software in a similar way to that for the aerodynamic module described in this thesis.

Further work is required on developing applications for complex geometries and on studying the implications of grid adaptation and unstructured mesh topologies. Another possible application could be on applying the evolutionary methods to re-entry vehicle optimisation using Direct Simulation Monte Carlo methods.

Appendix A

Publications Resulting from this Research

The following is a list of journal papers, book chapters, conference papers and seminars arising from this thesis work or in which parts of this thesis work were presented. A pdf version of these is included in the CD contained in the back cover of this thesis.

Journal Papers:

- L. F. González, E. Whitney, K. Srinivas and J. Périaux. Multidisciplinary Aircraft Design and Optimisation Using a Robust Evolutionary Technique with Variable Fidelity Models. (submitted for publication to AIAA *Journal of Aircraft)*.
- L. F. González, E. Whitney, K. Srinivas and J. Périaux. A Robust Evolutionary Technique for Coupled and Multidisciplinary Design and Optimisation Problems in Aeronautics. (submitted for publication to *CFD Journal*).

Book Chapters:

- L. F. González, E. J. Whitney, J. Périaux and K. Srinivas. Evolutionary Optimization Tools for Multi-objective Design in Aerospace Engineering: from Theory to MDO applications. *Evolutionary Algorithms and Intelligent Tools in Engineering Optimization*.W. Annicchiarico, J. Periaux, M. Cerrolaza and G. Winter editors. WIT Press, Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK.
- E.J. Whitney, L. Gonzalez, K. Srinivas and J. Périaux. Multi-Objective Evolution Design for UAV Aerodynamic Applications. *French-Australian Advanced Workshop on Multidisciplinary Methods and Numerical Tools for UAV Design Applications,* UAV-MMNT03,

Sydney, Australia. (to appear).

Lecture Notes:

- L. F. González, E. J. Whitney, J. Périaux and K. Srinivas. Practical Aerodynamic Design for UAVs using Multi-criteria Evolutionary Algorithms. Von Karman Institute (VKI) Lecture Series, Methods & Tools For Multi-Criteria/Multidisciplinary Design, November 15-19, 2004.

Conference Papers:

- L. F. González, K. Srinivas, J. Périaux and E. J. Whitney. Multidisciplinary Design Optimisation of Unmanned Aerial Vehicles (UAV) Using Multi-Criteria Evolutionary Algorithms. In Proceedings of the 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO6), Rio de Janeiro, Brazil, May 30 – June 3, 2005.
- L. F. González, E. J. Whitney, K. Srinivas, K. C. Wong and J. Périaux. A Multidisciplinary Framework for Design and Optimisation in Aeronautics. In Proceedings of the Eleventh Australian International Aerospace Congress (AIAC-11), March 13 – 17, 2005, Melbourne, Australia
- L. F. González, E. Whitney and K. Srinivas and J. Périaux. Multidisciplinary Aircraft Design and Optimisation Using a Robust Evolutionary Technique with Variable Fidelity Models. AIAA Paper 2004-4625, In CD Proceedings 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Aug. 30 – Sep. 1, 2004, Albany, NY.
- L. F González, E. J. Whitney, J. Périaux, M. Sefrioui and K. Srinivas. A Robust Evolutionary Technique for Inverse Aerodynamic Design Design and Control of Aerospace Systems Using Tools from Nature. In Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering, Volume II, ECCOMAS 2004, Jyvaskyla, Finland, July 24-28, 2004 editor: P. Neittaanmaki and T. Rossi and S. Korotov and E. Onate and J. Periaux and D. Knorzer, University of Jyvaskyla, Jyvaskyla, 2004 pages: CD ISBN 951-39-1869-6.
- L. F. González, E. J. Whitney, K. Srinivas, K. C. Wong and J. Périaux. Multidisciplinary Aircraft Conceptual Design Optimisation Using a Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA). In I.C. Parmee, editor, Proceedings of the Sixth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2004), volume 6, Bristol, UK, April 2004. Springer-Verlag.

- L. F. González, E. J. Whitney, K. Srinivas, K. C. Wong, M. Sefrioui and J. Périaux. Robust Evolutionary Methods for Multi-objective and Multidisciplinary Design Optimisation of Manned and Unmanned Aircraft Systems. In Proceedings of Symposium on Applied Aerodynamics and Design of Aerospace Vehicles, (SAROD 2003), 15–16 December 2003, Bangalore, India.
- L. F. González, E. Whitney and K. Srinivas and J. Périaux. Optimum Multidisciplinary and Multi-Objective Wing Design in CFD Using Evolutionary Techniques. In Proceedings of the Third International Conference on Computational Fluid Dynamics ICCFD3, July 2004, Westin Harbour Castle, Toronto, Canada.
- E. J. Whitney, L. González, K. Srinivas, J. Périaux. Adaptive Evolution Design Without Problem Specific Knowledge. In Proceedings of EUROGEN 2003, Barcelona, Spain.
- E. J. Whitney, L. Gonzalez, J. Périaux and K. Srinivas. Playing Games with Evolution: Theory and Aeronautical Optimisation Applications. In Proceedings ofICIAM 2003–5th International Congress on Industrial and Applied Mathematics, Sydney, Australia, July 2003.
- E. J. Whitney, L. González, K. Srinivas, J. Périaux. Multi-Criteria Aerodynamic Shape Design Problems in CFD using a Modern Evolutionary Algorithm on Distributed Computers. In Proceedings of the Second International Conference on Computational Fluid Dynamics, Sydney, Australia.
- J. Périaux, M. Sefrioui, E. Whitney, L. González. K. Srinivas, and J. Wang. Evolutionary Algorithms, Game Theory and Hierarchical Models in CFD. In Proceedings of the Second International Conference on Computational Fluid Dynamics, Sydney, Australia.

Seminars, Symposia, Workshops:

- (2004) K. Srinivas, L.F. González, Developments in Evolutionary Algorithms and Multidisciplinary Design Optimisation. A University of Sydney Perspective. US Air-Force Research Lab (AFRL), Dayton Ohio, August 26, 2004.
- (2004/2003/2002) L.F. González, E. J. Whitney and K. Srinivas, Evolutionary Algorithms for Multidisciplinary Design Optimisation in Aeronautics. University of Sydney Research Showcase.
- (2002) L.F. González, Evolution Algorithms and their Applications to Aeronautical Design Problems, Lunch time seminar School of Aerospace, Mechanical and Mechatronic Engineering, J07 University of Sydney, NSW, 2006 Australia.

Appendix B

Software Developed in this Thesis

B.1 Indexed CD

Different programs were developed and integrated for the algorithms and methods described. In total, these programs account for many thousands of lines of code and several pages. A copy of these programs is included in an indexed CD placed in the back cover of this thesis. It contains an index and html files with several documents, subroutines and programs described in this thesis; the GUI, the different design module and interfaces. If you would like assistance on navigating the contents of this CD please email me at ***gonzalez@aeromech.usyd.edu.au***. Some of the most relevant subroutines and programs are:

B.2 Programs for GUI

-*gui.fl*

The Graphical User Interface

B.3 Programs and Methods for Multi-objective Test Functions Module

A series of algorithms and methods developed and integrated for testing convergence to the Pareto optimal front.

Constrained-Multi-Objective
-*testbenchBNH.cpp*
-*testbenchSRN.cpp*

-*testbenchTNK.cpp*

Convex-Non-Convex-Pareto-Optimal-Fronts
-*testbenchZDT1.cpp*
-*testbenchZDT2.cpp*

Discontinuous-Pareto-Optimal-Front
-*testbenchZDT3.cpp*

Non-Linear-Goal-Programming
testbenchP1.cpp
testbenchP2.cpp
testbenchP3.cpp

B.4 Programs and Methods for Single-element Aerofoil Design and Optimisation Module

-*heaairfoil.cpp*
This file controls the aerofoil optimisation process. It reads upper and lower bounds for the design variables, reads the constraints, flight conditions, and generates candidate aerofoil geometries and sends them for evaluation. It also links with the evolutionary optimiser and parallel computing files that control the evolution process.
-*paracostairfoil.cpp*
This program sends the information for evaluation and computes the objective functions which are returned to the main file.
-*msesaerofoil.cpp*
This program contains the methods, algorithms and classes for aerofoil evaluation.
-*airfoilasync2.cpp*
This program links the main program with parallel computations via PVM.
-*blade.xxx*
Input file for aerofoil analysis using MSES.

B.5 Programs and Methods for Wing Design and Optimisation Module

-*heawing.cpp*
Similar to the single-element aerofoil file, this is the main file for wing design and optimisation.

-flo22wing.cpp
This program contains the methods, algorithms and classes for a wing evaluation.
-paracostwing.cpp
This program sends the information for evaluation of a wing and computes the objective functions, which are returned to the main file.
-mses2flo22.cpp
This file reads input files with wing planform variables, aerofoils, flight conditions and boundaries, and generates a mesh around the wing automatically for an aerodynamic analysis.
-wingasync.cpp
This program links the main program with parallel computations, via PVM.
-flo22.thiswing-xxx
This file is the input file generated automatically by *mses2flo22.cpp* for its analysis, using FLO22.

Specific programs for aero-structural analysis
-mses2cgx.cpp
This file reads input files with wing planform variables, aerofoils, flight conditions and boundaries, and generates a mesh around the wing automatically for an structural analysis.
-new-wing.Sample-Point-xxx
This is the input file for the wing structural analysis.
-drundace.cpp
This file contains the algorithm that runs a DOE using the aero-structural wing analysis program.
-assignpressuretoFEAmesh.cpp
This file contains the algorithms that map the pressure distribution from the aerodynamics into the structural model for analysis.
-wing-aero-str-analysis.cpp
This file contains the algorithms to performs a single aero-structural wing analysis.

B.6 Programs and Methods for Aircraft Design and Optimisation Module

heamodel.cpp
Similar to the single element aerofoil file, this is the main file for aircraft design and optimisation
-paracostmodel.cpp
This program sends the information for evaluation of an aircraft and computes the objective functions, which are returned to the main file.
-FLOPSmodel.cpp

This program contains the methods, algorithms and classes for the evaluation of an aircraft or UAV.

-*modelasync.cpp*

This program links the main program with parallel computations, via PVM.

xflp.thisconfiguation-xxx

This file is the input file generated automatically by FLOPSmodel.cpp for its analysis, using FLOPS.

outputconfiguration-xxx

This is an output file after an aircraft or UAV analysis, using FLOPS.

B.7 Programs and Methods for Aircraft High-lift System Design and Optimisation Module

-*heaairfoil.cpp*

Similar to the single-element aerofoil file, this is the main file for high-lift aircraft system design and optimisation.

-*moveandrotate.cpp*

This file is used to generate rotation and displacement of the high-lift aircraft system elements (slat and flap).

-*mses2bamg-amdba.cpp*

This file reads input files with aerofoil coordinates and boundaries, and generates an unstructured mesh around the aerofoil automatically.

-*nsc2keaerofoil.cpp*

This file is a series of classes and methods for analysing multi-element aerofoils.

-*paracostairfoil.cpp*

This program sends the information for evaluation and computes the objective functions which are returned to the main file.

DATA.xxx

Input file with flight conditions and parameters for the Navier-Stokes solver.

-*adap.sh*

This program performs a mesh adaptation around a candidate geometry, using BAMG as the mesh generator and NSC2ke as the Navier-Stokes solver. Figures B.1, B.2 and B.3 illustrate the concept for a three-element aerofoil. The mesh gets refined where the most interesting aspects of the flow occur.

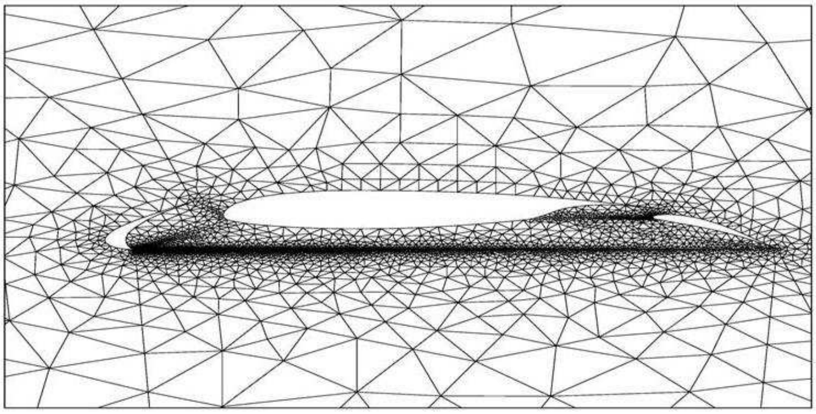

Figure B.1: Mesh 0 of the mesh adaptation process.

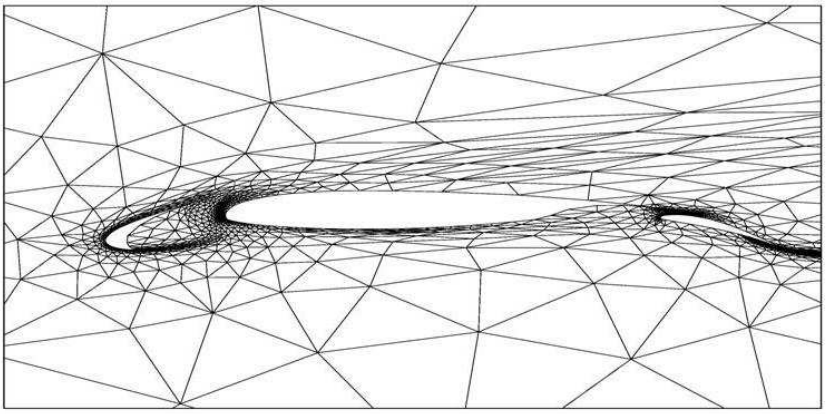

Figure B.2: Mesh 2 of the mesh adaptation process.

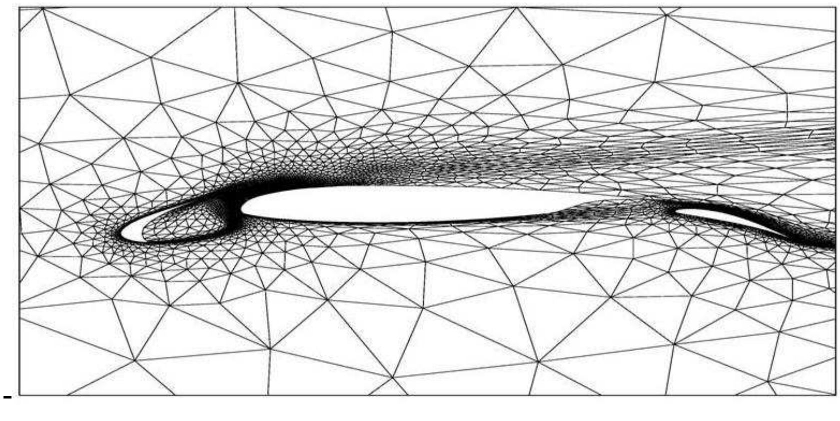

Figure B.3: Mesh 4 of the mesh adaptation process.

Appendix C

Analysis Tools for Aircraft Conceptual and Preliminary Design

One important consideration before conducting optimisation is the selection of appropriate analysis tools. There are many software tools for aerofoil, wing and aircraft design that were evaluated and assessed during the course of this thesis. In the following, a brief description of the availability and capabilities of some commercial and free-source codes evaluated during this research is presented. The descriptions are, in some cases, taken from the Internet web page of their software, user manual or by experience of their use in the School of Aerospace, Mechanical and Mechatronic Engineering (AMME).

C.1 Aerofoil Design

Three different-fidelity flow-solvers were considered and studied: *NSC2Ke* [110], *XFOIL* [47] and *MSES* [47].

Navier-Stokes /Euler Solver (NSC2KE): The highest fidelity solver considered is the Navier-Stokes solver, *NSC2ke* software developed by Mohammadi. *NSC2Ke* is a Finite-Volume Galerkin program computing 2D and axisymmetric flows on unstructured meshes that has capabilities for viscous or Euler flow. Details on *NSC2Ke* can be found in [110],

Panel Method Solver (XFOIL)

The second solver considered is the *XFOIL* software written by Drela. It comprises a higher order panel method with coupled integral boundary layer. Details on *XFOIL* can be found in [47]

Euler+ Boundary Layer Solver (MSES).

The solver is based on a structured quadrilateral streamline mesh which is coupled to an integral boundary layer based on a multi-layer velocity profile representation. Details on MSES can be found in Drela [46].

A copy of the source code was obtained for all this programs and proved to be very useful for this research, as was illustrated in different chapters of this thesis.

C.2 Wing Design

A brief overview of the different analysis codes considered for wing design includes:

Potential Flow Solver (*FLO22*): The potential flow-solver *FLO22* was developed by Jameson (NYU) and Caughey (Cornell) for analysing inviscid, isentropic, transonic flow past 3-D swept-wing configuration. Some details of the algorithm are that the free-stream Mach number is restricted by the isentropic assumption and that weak shock waves are automatically located whereever they occur in the flow. Also, the finite difference form of the full equation for the velocity potential is solved by the methods of relaxation, after the flow exterior to the aerofoil is mapped to the upper half plane. The mapping procedure allows exact satisfaction of the boundary conditions and use of supersonic free-stream velocities. Details on the formulation and implementation can be found in Jameson [76]. Although limited to some applications FLO22 was shown to be robust and fast on its coupling with Evolutionary techniques.

C.3 Aircraft Design

A brief overview of the different analysis codes considered for aircraft design includes:

AAA (Advanced Aircraft Design Analysis) is based on the design textbooks series of Jan Roskam and contains full vehicle analysis and two-discipline capabilities. Level One is for minimal inputs and Level Two for more detailed inputs and analysis. A student version of this code is available and used for different courses at AMME, but no source code is provided, hence it is limited for the purposes of this research.

FLOPS (Flight Optimisation) code was developed by NASA Langley and is widely used at universities, governmental and aerospace companies. This code is robust and provides capabilities for conceptual and preliminary design studies. A copy of the source code was obtained and proved to be very useful for this research as illustrated in different chapters of this thesis.

RDS, developed by Raymer [144], includes sophisticated analysis tools for conceptual design, but no source code was available, hence it is limited for the purposes of this research.

*ADA Aircraft Design and analysis softwar*e: ADA is an in-house solver developed by the author that uses an object-oriented approach, and which is based on the formulation presented in

Raymer [144]. This code is simple and intuitive, but less robust, compared with the FLOPS code. In this research it has been also used for analysis, but can be extended by incorporating more robust methods for aerodynamics, structural and performance analysis.

Bibliography

[1] I. H. Abbott and A. E. Von Doenhoff. *Theory of Wing Sections*. Dover, 1980.

[2] D. Abramson, A. Lewis, T. Peachey, and C. Fletcher. An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics. In *Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)*, Denver, Colorado, 2001.

[3] R. K. Agarwal. Computational Fluid Dynamics for Whole Body Aircraft. *Annual Review of Fluid Mechanics*, 31:125–169, 1999.

[4] N. Alexandrov and S. Kodyalam. Initial Results of an MDO Methods Evaluation Study. In *Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, Missouri, September 1998.

[5] N. M. Alexandrov and R. M. Lewis. Analytical and Computational Properties of Distributed Approaches to MDO. In *AIAA Paper 2000-4718*. AIAA, September 2000.

[6] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Optimization with Variable-fidelity Models Applied to Wing Design. In *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2000. AIAA.

[7] N. Ali and K. Behdinan. Conceptual Aircraft Design-A Genetic Search and Optimisation Approach. In *Proceedings of the 23rd International Congress of Aeronautical Sciences Sciences, ICAS 2002*, Toronto, Canada, 2002.

[8] K. Anandhanarayanan, M. Nagarathinam, and S. M. Deshpande. A Gridfree Navier-Stokes Solver. In *Presented at 5th AeSI CFD Symposium*, Bangalore, India, August 2002.

[9] M. B. Anderson and G. A. Gerbert. Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design. AIAA Paper 96-4023, AIAA, Washington, D. C., 1996.

[10] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York, 1997.

[11] T. Back and M. Schutz. Evolution Strategies for Mixed-integer Optimization of Optical Multilayer Systems. In *Evolutionary Programming: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 33–51, 1995.

[12] G. J. Barlow, C. K. Oh, and E. Grant. Incremental Evolution of Autonomous Controllers for Unmanned Aerial Vehicles using Multi-objective Genetic Programming. In Maarten Keijzer, editor, *Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference*, Seattle, Washington, USA, July 2004.

[13] P. Bartholomew. The Role of MDO within Aerospace Design and Progress towards an MDO Capability. In *Proceedings of Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, St. Louis, Missouri, September 1998.

[14] B. Beachkofski and R. V. Grandhi. Improved Distributed Hypercube Sampling. In *Proceedings 43rd AIAA/ASME/ASCE/AHS/ASC SDM Conference*, Denver, Colorado, April 2002.

[15] M. A. Belessis, D. G. Stamos, and S. G. Voutsinas. Investigation of the Capabilities of a Genetic Optimization Algorithm in Designing Wind Turbine Rotors. In *European Union Wind Conference and Exhibition*, Goteborg, Sweden, May 20-24 1996.

[16] J. Bennett, P. Fenyes, W. Haering, and M. Neal. Issues in Industrial Multidisciplinary Optimization. In *Proceedings of Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, St. Louis, Missouri, September 1998.

[17] J. A. Bennett, M. E. Botkin, C. Koromilas, R. V. Lust, M. O. Neal, J. T. Wang, and R.I. Zwiers. A Multidisciplinary Framework for Preliminary Vehicle Analysis and Design. In *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, 1997.

[18] J. J. Bertin. *Aerodynamics for Engineers*. Prentice Hall, 2002.

[19] E. Besnard, A. Schmitz, E. Boscher, N. Garcia, and T. Cebeci. Two-dimensional Aircraft High-lift System Design and Optimization. Technical Report 98-404, AIAA Paper, 1998.

[20] M. Blair, S. LeClair, J. Zweber, and A. Chemaly. Multidisciplinary Design for Uninhabited Air Vehicles. In *6th Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises (WET-ICE '97)*. AIAA, 1997.

[21] A. J. Booker, J. E. Dennis, P. D. Frank, Jr., D. B. Serafini, V. Torczon, and M. W. Trosset. A Rigorous Framework for Optimization of Expensive Functions by Surrogates. *Structural Optimization*, 17(1):1–13, 1999.

[22] A. H. W. Bos. Aircraft Conceptual Design by Genetic/Gradient-Guided Optimization. *Engineering Applications of Artificial Intelligence*, 11,3(3):377–382, 1998.

[23] R. Braum, P. Gage, I. Kroo, and I. Sobieski. Implementation and Performance Issues in C.O. In *Proceedings of Sixth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, Washington, September 4–6 1996. AIAA Paper No. 96-4017.

[24] E. Cantú-Paz. A Summary of Research on Parallel Genetic Algorithms. Technical Report 95007, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1995.

[25] E. Cantú-Paz. Designing Efficient Master-Slave Parallel Genetic Algorithms. Technical Report 97004, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.

[26] E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Pub, 2000.

[27] H. S. Chung and J. J. Alonso. Multiobjective Optimization Using Gradient Enhanced Genetic Algorithms. In *Proceedings of the 10th AIAA/ISSMO Multidisciplany Analysis and Optimization Conference*, Albany, New York, August 30–September 1 2004. AIAA Paper 2004-4325.

[28] C. Coello-Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multiobjective Problems*. Kluwer Academic Publishers, New York, 2002.

[29] C. A. Coello-Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.

[30] F. Courty, A. Dervieux, and J-C. Gilbert. RD1-1999-10752 AEROSHAPE, Adaptation of Optimization Methods to Optimal Design in Aerodynamics. Technical report, INRIA, 1999.

[31] E. J. Cramer, J. E. Dennis, P.D. Frank, R. M. Lewis, and G. R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, 4:754–776, 1994.

[32] Y. Crispin. Aircraft Conceptual Optimization Using Simulated Evolution, 1994. AIAA Paper 94-0092.

[33] A. W. Crossley and H. Laananen. Design of Helicopters via Genetic Algorithm. *Journal of Aircraft*, 3(6), November–December 1996.

[34] D. Cvetković, I. Parmee, and E. Webb. Multi-objective Optimisation and Preliminary Airframe Design. In *The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation*, pages 255–267. Springer-Verlag, New York, New York, 1998.

[35] D. Cvetković and I. C. Parmee. Use of Preferences for GA-based Multi-objective Optimisation. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1504–1509, Orlando, Florida, USA, 1999. Morgan Kaufmann Publishers.

[36] D. Cvetković and I.C. Parmee. Designer's Preferences and Multi-objective Preliminary Design Processes. In I.C. Parmee, editor, *Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000)*, pages 249–260, PEDC, University of Plymouth, UK, 2000. Springer.

[37] C. Darwin. *On the Origin of Species: being the second part of his big species book written from 1856 to 1858 / edited from manuscript*. C. Stauffer, London, 1975.

[38] D. Dasgupta and Z. Michalewicz. *Evolutionary Algorithms in Engineering Applications*. Springer-Verlag, 1997.

[39] I. De Falco, A. D. DelBalio, R.and Cioppa, and E. Tarantino. A Parallel Genetic Algorithm for Transonic Airfoil Optimisation. *Evolutionary Computation*, 1, 1995.

[40] K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dept. of Computer Science/LS11, University of Dortmund, Germany, 1998.

[41] K. Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In *Proceedings of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99)*, 1999.

[42] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2003.

[43] K. Deb and T. Meyarivan. Constrained Test Problems for Multi-objective Evolutionary Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 284–298. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[44] G. Dhondt. CalculiX CrunchiX USER'S MANUAL version 1.1. Technical report, 2003.

[45] M. Dorigo and V. Maniezzo. Parallel Genetic Algorithms: Introduction and Overview of Current Research. In *Parallel Genetic Algorithms: Theory and Applications*, pages 5–42. Stender, 1992.

[46] M. Drela. A User's Guide to MSES 2.95. Technical report, MIT Computational Aerospace Sciences Laboratory, 1996.

[47] M. Drela. XFOIL 6.95 User Guide. Technical report, MIT Computational Aerospace Sciences Laboratory, 2001.

[48] G. S. Dulikravich, B. H. Dennis, T. J. Martin, and I. N. Egorov. Multi-disciplinary Analysis and Design Optimization. In *Invited Lecture, Mini-Symposium on Inverse Problems–State of Art and Future Trends, XXIV Brazilian Congress on Applied and Computational Mathematics*, Belo Horizonte, Brazil, September 10–13 2001.

[49] S. Dunn and S. Peucker. Genetic Algorithm Optimisation of Mathematical Models Using Distributed Computing. In *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Cairns, Australia, 2002.

[50] D. Fogel. An Analysis of Evolutionary Programming. In D. B. Fogel and W. Atmar, editors, *Proceedings of the Second Annual Conference on Evolutionary Programming (EP'93)*, La Jolla CA, February 1993. Evolutionary Programming Society, San Diego California.

[51] L. J. Fogel, P. J. Angeline, and D. B. Fogel. An Evolutionary Programming Approach to Self-Adaptation on Finite State Machines. In *Evolutionary Programming*, pages 355–365, 1995.

[52] C. M. Fonseca and P. J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[53] C. M. Fonseca and Fleming. P. J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[54] K. Gantois and A. J. Morris. The Multi-disciplinary Design of a Large-scale Civil Aircraft Wing Taking Account of Manufacturing Costs. *Structural and Multidisciplinary Optimization*, 28(1):31–46, August 2004.

[55] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine: A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994.

[56] J. P. Giesing and J. F. Barthelem. A Summary of Industry MDO Applications and Needs. In *Proceedings of Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*. AIAA, St. Louis, Missouri, September 1998.

[57] A. A. Giunta, V. Balabanov, S. Burgee, B. Grossman, R. T. Haftka, W. H. Mason, and L. T. Watson. Variable-complexity Multidisciplinary Design Optimization Using Parallel

Computers. In S. N. Alturi, G. Yagawa, and T. A. Cruse, editors, *Computational Mechanics '95–Theory and Applications, Proc. of ICES '95, International Conference on Computational Engineering Science*, pages 489–494, Mauna Lani, Big Island, Hawaii, July 1995. Springer.

[58] A. A. Giunta, V. Balabanov, D. Haim, B. Grossman, W. H. Mason, R. T. Haftka, and L. T. Watson. Multidisciplinary Optimization of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modeling. *The Aeronautical Journal*, pages 347–356, October 1997.

[59] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[60] L. F. González, E. J. Whitney, K. Srinivas, and J. Périaux. Multidisciplinary Aircraft Design and Optimisation Using a Robust Evolutionary Technique with Variable Fidelity Models. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA Paper 2004-4625*, Albany, New York, August 30–September 1 2004. AIAA.

[61] L. F. González, E. J. Whitney, K. Srinivas, and J. Périaux. Practical Aerodynamic Design for UAVs using Multicriteria Evolutionary Algorithms. In *von Karman Institute for Fluid Dynamics–Lecture Series, Methods and Tools For Multi-Criteria/Multidisciplinary Design*. Belgium, November 2004.

[62] L. F. González, E. J. Whitney, K. Srinivas, and J. Périaux. A Robust Evolutionary Technique for Inverse Aerodynamic Design. In P. Neittaanmaki, T. Rossi, S. Korotov, E. Onate, J. Periaux, and D. Knorzer, editors, *Design and Control of Aerospace Systems Using Tools from Nature. Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering*, volume 2, Jyvaskyla, Finland, 2004. Wiley.

[63] L.F. González, E.J. Whitney, K. Srinivas, and J. Périaux. Optimum Multidisciplinary and Multi-objective Wing Design in CFD Using Evolutionary Techniques. In *Proceedings of the Third International Conference on Computational Fluid Dynamics ICCFD3*, Westin Harbour Castle-Toronto, Canada, July 2004.

[64] A. B. Hadj-Alouane and J. C. Bean. A Genetic Algorithm for the Multiple-Choice Integer Program. *Operations Research*, 45:92–101, 1997.

[65] M. Hampsey. *Multiobjective Evolutionary Optimisation of Small Wind Turbine Blades*. PhD thesis, University of Newcastle, Newcastle, Australia, August 2002.

[66] N. Hansen. Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies. In *Proceedings of the Sixth International Conference Parallel Problem Solving from Nature (PPSN-VI)*, pages 355–364. Springer, 2000.

[67] N. Hansen and A. Ostermeier. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.

[68] N. Hansen and A. Ostermeier. Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_i, \lambda)$-ES. In *Proceedings of fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pages 650–654, Aachen, 1997. Verlag Mainz.

[69] N. Hansen and A. Ostermeier. Completely Derandomised Self-Adaption in Evolution Strategies. In *Evolutionary Computation*, volume 9(2), pages 159–195. MIT Press, 2001.

[70] F. Hecht. Bamg: Bidimensional Anisotropic Mesh Generator: Draft Version v0.58. Technical report, INRIA, 1998.

[71] J. Holland. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[72] J. Horn, N. Nafpliotis, and D. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, Piscataway, New Jersey, 1993.

[73] M. Hutchison, E. Unger, W. Mason, B. Grossman, and R. Haftka. Variable-Complexity Aerdynamic Optimization of a High-Speed Civil Transport Wing. *AIAA Journal of Aircraft*, 31(1):110–116, January–February 1994.

[74] P. Jackson. *Jane's All the World's Aircraft 2002-03*. Jane's Information Group Limited, 2002–2003.

[75] A. Jameson. A Perspective on Computational Algorithms for Aerodynamic Analysis and Design. *Progress in Aerospace Sciences*, 37(2):197–243(47), February 2001.

[76] A. Jameson, D. Caughey, P. Newman, and R. Davis. A Brief Description of the Jameson Caughey NYU Transonic Swept-Wing Computer Program FLO22. Technical Report NASA TM X-73996, NASA/American Institute of Aeronotics and Astronautics, 1976.

[77] A. Jameson, S. Shankaran, L. Martinelli, and B. Haimes. Aerodynamic Shape Optimization of Complete Aircraft Configuration. In *Proceedings of the 42th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2004.

[78] A. Jameson and J. C. Vassberg. Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact. In *Proceedings of the 39th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2001. AIAA.

[79] J. A. Joines and C. R. Houck. On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GA's. In *International Conference on Evolutionary Computation*, pages 579–584, 1994.

[80] B. Jones, W. Crossley, and A. Lyrintzis. Aerodynamic and Aeroacoustic Optimization of Airfoils via a Parallel Genetic Algorithm. In *Proceedings of Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, St. Louis, Missouri, September 1998.

[81] H. J. Kim and O. H. Rho. Dual-Point Design of Transonic Airfoils Using the Hybrid Inverse Optimization Method. *Journal of Aircraft*, 34(5):612–618, September–October 1997.

[82] S. Kim, J. J. Alonso, and A. Jameson. Design Optimization of High-lift Configurations Using a Viscous Continuous Adjoint Method. In *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2002.

[83] M. Klein and H Sobieczky. Sensitivity of Aerodynamic Optimization to Parameterized Target Functions. In G.S. Dulikravich M. Tanaka, editor, *Inverse Problems in Engineering Mechanics, Proceedings of the Int. Symp. on Inverse Problems in Engineering Mechanics (ISIP2001)*, Nagano, Japan, 2001.

[84] D. Knight. Applications of Genetic Algorithms to High Speed Air Intake Design. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 43–50, Athens, Greece, 2001. International Center for Numerical Methods in Engineering (CIMNE).

[85] K. Kodiyalam, R. J. Yang, L. Gu, and C-H. Tho. Large-scale, Multidisciplinary Optimization of a Vehicle System in a Scalable, High Performance Computing Environment. *Structural and Multidisciplinary Optimization*, 26:256–263, 2004.

[86] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski. Multidisciplinary Optimization Methods for Aircraft Preliminary Design. In *Proceedings of Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, Florida, September 7–9 1994.

[87] V. Kumar. Algorithms for Constraint-Satisfaction Problems: A Survey. *AI Magazine*, 13(1):32–44, 1992.

[88] J. E. Lamar and H. E. Herbert. *Production Version of the Extended NASA-Langley Vortex Lattice FORTRAN Computer Code – Vol. I – User's Guide*, April 1982.

[89] G. B. Lamont, S. M. Brown, and G.H. Gates Jr. Evolutionary Algorithms Combined with Deterministic Search. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII*, pages 517–526, Berlin, 1998. Springer.

[90] Y. Li, X. Xin, N. Kikuchi, and K. Saitou. Optimal Shape and Location of Piezoelectric Materials for Topology Optimization of Flextensional Actuators. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 1085–1090, 2001.

[91] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. Aspects of the Matlab Toolbox DACE. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002.

[92] T. Lutz, A. Sommerer, and S. Wagner. Parallel Numerical Optimisation of Adaptive Transonic Airfoils. In *Symposium Transsonicum IV*, Göttingen, Germany, September 2–6 2002.

[93] A. K. Mackworth. Evolutionary Optimization of Constrained Problems. *Artificial Intelligence*, 8(11):99–118, 1977.

[94] R. Mäkinen, P. Neittaanmäki, P. Périaux, and J. Toivanen. A Genetic Algorithm for Multiobjective Design Optimization in Aerodynamics and Electromagnetics. In K. D. Papailiou et al., editor, *Computational Fluid Dynamics '98, Proceedings of the ECCOMAS 98 Conference*, volume 2, pages 418–422, Athens, Greece, 1998. Wiley.

[95] R. Mäkinen, J. Periaux, and J. Toivanen. Shape Design Optimization in 2D Aerodynamics Using Genetic Algorithms on Parallel Computers. In N. Satofuka A. Ecer, J. Periaux and S. Taylor, editors, *Proceedings of the Parallel CFD'95 Conference*, Parallel Computational Fluid Dynamics: Implementations and Results Using Parallel Computers, pages 395–402. Elsevier, 1995.

[96] J. B. Malone, J. C. Narramore, and L. N. Sankar. Airfoil Design Method Using the Navier-Stokes Equations. *AIAA Journal of Aircraft*, 28(3):216–224, March 1991.

[97] N. Marco, J. A. Désidéri, and S. Lanteri. Multi-objective Optimization in CFD by Genetic Algorithms. Technical Report 3686, Institut National de Recherche en Informatique et aen Automatique (INRIA), April 1999.

[98] N. Marco, S. Lanteri, J. A. Désidéri, and J. Périaux. A Parallel Genetic Algorithm for Multi-objective Optimisation in Computational Fluid Dynamics, 1998.

[99] R. J. Margason and J. E. Lamar. Vortex-lattice FORTRAN Program for Estimating Subsonic Aerodynamic Characteristics on Complex Planforms. Technical report, NASA TN D-6142, February 1971.

[100] R. T. Marler and J. S. Arora. Survey of Multi-objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2002.

[101] J. R. R. A. Martins and J. J. Alonso. High-fidelity Aerostructural Design Optimization of a Supersonic Business Jet. *Journal of Aircraft*, 41,3:523–530(8), 2004.

[102] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. Complete Configuration Aerostructural Optimization Using a Coupled Sensitivity Analysis Method. In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 2002.

[103] W. H. Mason, D.L. Knill, A. A. Giunta, B. Grossman, R. T. Haftka, and L. T. Watson. Getting the Full Benefits of CFD in Conceptual Design. In *Proceedings of the AIAA 16th Applied Aerodynamics Conference*, Albuquerque, New Mexico, 1998.

[104] C. McAllister, T. Simpson, K. Hacker, and K. Lewis. Application of Multidisciplinary Design Optimization to Racecar Design Analysis. In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 2002.

[105] A. McCullers. FLOPS User's Guide, Release 6.02, NASA Langley Research Center, March 2003.

[106] A. Meyer and H. Van Kampen. X-Plane On-Line Instruction Manual, Eighth Edition. Technical report, 2002.

[107] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Artifical Intelligence. Springer-Verlag, 1992.

[108] Z. Michalewicz. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In John R. McDonnell, Robert G. Reynolds, and David B. Fogel, editors, *Proc. of the 4th Annual Conf. on Evolutionary Programming*, pages 135–155, Cambridge, Massachussets, 1995. MIT Press.

[109] Z. Michalewicz and N. Attia. Evolutionary Optimization of Constrained Problems. In *Third Annual Conference on Evolutionary Programming*, pages 98–108, 1994.

[110] B. Mohammadi. Fluid Dynamics Computation with NSC2KE: A User-Guide: Release 1.0. Rt-0164, INRIA, 1994.

[111] K. Munson. *Unmanned Aerial Vehicles and Targets*. Jane's Information Group Limited, 2004–2005.

[112] R. H. Myers and D. C. Montgomery. *Response surface methodology: process and product optimization using designed experiments*. Wiley series in probability and statistics. Wiley, New York, 2nd edition, 2002.

[113] S. Nadarajah and A. Jameson. Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optmization. In *Proceedings of the AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, California, June 2001. AIAA.

[114] J. F. Nash. Equilibrium points in N-person games. In *Proceedings of the National Academy of Science*, number 36, pages 46–49, 1950.

[115] J. F. Nash. Noncooperative games. In *Annals of Mathematics*, number 54, page 289. 1951.

[116] M. Nemec and D. Zingg. Towards Efficient Aerodynamic Shape Optimization Based on the Navier-Stokes Equations, June 2001. AIAA Paper 2001–2532.

[117] T. T. H. Ng and G. S. B. Leng. Application of Genetic Algorithms to Conceptual Design of a Micro-air Vehicle. *Engineering Applications of Artificial Intelligence*, 15(5):439–445, September 2002.

[118] S. Obayashi. Aerodynamic Optimization with Evolutionary Algorithms. In *von Karman Institute for Fluid Dynamics–Lecture Series*. Belgium, April 1997.

[119] S. Obayashi. Multidisciplinary Design Optimization of Aircraft Wing Planform Based on Evolutionary Algorithms. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, La Jolla, California, October 1998. IEEE.

[120] S. Obayashi, K. Nakahashi, A. Oyama, and N. Yoshino. Design Optimization of Supersonic Wings Using Evolutionary Algorithms. In *Proceedings of the Fourth ECCOMAS Computational Fluid Dynamics Conference*, Athens, Greece, September 1998.

[121] S. Obayashi and S. Takanashi. Genetic Optimization of Target Pressure Distributions for Inverse Design Methods. *AIAA Journal*, 34(5):881–886, May 1996.

[122] S. Obayashi and T. Tsukahara. Comparison of Optimization Algorithms for Aerodynamic Shape Design. *AIAA Journal*, 35(8):1413–1415, August 1997.

[123] S. Obayashi, T. Tsukahara, and T. Nakamura. Multiobjective Genetic Algorithm Applied to Aerodynmic Design of Cascade Airfoils. *IEEE Transactions on Industrial Electronics*, 47(1):211–216, February 2000.

[124] S. Obayashi, T. Tsukuhara, and T. Nakamura. Cascade Airfoil Design by Multiobjective Genetic Algorithms. In *Proceedings of the Second IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALE-SIA'97)*, Glasgow, Scotland, September 1997.

[125] A. Ostermeier, A. Gawelczyk, and N. Hansen. A Derandomized Approach to Self Adaptation of Evolution Strategies. *Evolutionary Computation*, 2(4):369–380, 1994.

[126] A. Oyama, M. Liou, and S. Obayashi. Transonic Axial-flow Blade Shape Optimization Using Evolutionary Algorithm and Three-Dimensional Navier-Stokes Solver. In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 2002.

[127] A. Oyama, S. Obayashi, K. Nakahashi, and T. Nakamura. Euler/Navier-Stokes Optimization of Supersonic Wing Design Based on Evolutionary Algorithm. *AIAA Journal*, 37(10):1327–1329, October 1999.

[128] A. Oyama, S. Obayashi, and T. Nakamura. Real-Coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization. In *Lecture notes in Computer Science*, volume 1917, pages 712–721. Springer-Verlag, Berlin Heidelberg New York, 2000.

[129] M. Papadrakakis and N. D. Lagaros. *Advances in Structural Optimization*. NTUA, 2000.

[130] M. Papadrakakis, N. D. Lagaros, and Y. Tsompanakis. Structural Optimization Using Evolution Strategies and Neural Networks. *Computer Methods in Applied Mechanics in Engineering*, 156:309–333, 1998.

[131] V. Pareto. *Cours d'Economie Politique*. Rouge, Lausanne, Switzerland, 1896.

[132] I. Parmee and A. H. Watson. Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1657–1665, Orlando, Florida, USA, July 1999. Morgan Kaufmann.

[133] J. Périaux, M. Sefrioui, and B. Mantel. RCS Multi-objective Optimization of Scattered Waves by Active Control Elements using GAs. In *Proceedings of the Fourth International Conference on Control, Automation, Robotics and Vision (ICARCV'96)*, Singapore, 1996.

[134] J. Périaux, M. Sefrioui, E. J. Whitney, L. González, K. Srinivas, and J. Wang. Multi-Criteria Aerodynamic Shape Design Problems in CFD Using a Modern Evolutionary Algorithm on Distributed Computers. In S. Armfield, P. Morgan, and K. Srinivas, editors, *Proceedings of the Second International Conference on Computational Fluid Dynamics (ICCFD2)*, Sydney, Australia, July 2002. Springer.

[135] C. B. Pettey, M. R. Leuze, and J. J. Grefenstette. A Parallel Genetic Algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms, Buenos Aires, Argentina*, 1998.

[136] E. Pires, J. Machado, and P. Oliveira. An Evolutionary Approach to Robot Structure and Trajectory Optimization. In *Proceedings of the 10th International Conference on Advanced Robotics*, pages 333–338, Budapest, Hungary, August 2001.

[137] J. Pittman. Supersonic Airfoil Optimization. *AIAA Journal of Aircraft*, 24(12):873–879, December 1987.

[138] T. H. Pulliam, M. Nemec, T. Hoist, and D. W. Zingg. Comparison of Evolutionary (Genetic) Algorithm and Adjoint Methods for Multi-Objective Viscous Airfoil Optimizations. In *Proceedings of the 41th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2003. AIAA.

[139] D. Quagliarella, J. Periaux, C. Poloni, and G. Winter. *Genetic Algorithms in Engineering and Computer Science*. Wiley, 1997.

[140] D. Quagliarella and A. Vicini. Designing High-Lift Airfoils Using Genetic Algorithms. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Proceedings of EUROGEN'99*, Jyväskylä, Finland, 1999. University of Jyváskylä.

[141] D. Quagliarella and A. Vicini. Viscous Single and Multicomponent Airfoil Design with Genetic Algorithms. Technical Report CIRA-RT-AEP-00-005, Centro Italiano Ricerche Aerospaziali (CIRA), Italy, 2000.

[142] D. Raymer. *Enhancing Aircraft Conceptual Design using Multidisciplinary Optimization*. PhD thesis, KTH, Department of Aeronautics, FLYG 2002-2, 2002.

[143] D. Raymer. Use of Genetic and Evolutionary Algorithms for Aircraft Conceptual Design Optimization. In *Proceedings of the 41th AIAA Aerospace Sciences Meeting*. AIAA, Reno, Nevada, 2003.

[144] D. Raymer. *Aircraf Design: A Conceptual Approach, American Institute of Aeronautics and Astronautics*. American Institute of Aeronautics and Astronautics, Third Edition, 1999.

[145] D. Raymer and W. Crossley. A Comparative Study of Genetic Algorithm and Orthogonal Steepest Descent for Aircraft Multidisciplinary Optimization. In *AIAA Aerospace Sciences Meeting*. AIAA, Reno, Nevada, January 2002. AIAA Paper 2002-0514.

[146] G. Renaud and G. Shi. Evaluation and Implementation of Multidisciplinary Design Optimization (MDO) Strategies. In *Proceedings of the 23rd ICAS Congress*, Toronto, Canada, September 2002.

[147] J. Reuther, A. Jameson, J. Alonso, M. Rimlinger, and D. Saunders. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, part I. *AIAA Journal of Aircraft*, 36(1):51–74, January–February 1999.

[148] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions–Incorporating Problem Specific Knowledge into Genetic Algorithms. In Schaffer J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, George Mason University, 1989. Morgan Kaufmann.

[149] P. Ruben, J. Chung, and K. Behdinan. Aircraft Conceptual Design Using Genetic Algorithms. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. AIAA, Long Beach, California, September 2000.

[150] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–435, 1989.

[151] A. O. Salas and J. C. Townsend. Framework Requirements for MDO Application Development. In *Proceedings of Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, St. Louis, Missouri, September 1998.

[152] M. R. Sankar, I. Amitay, P.M. Mujumdar, and K. Sudhakar. MDO Framework Development-A Case Study With An Elementary Model of Airborne Early Warning System Optimization. In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 2002.

[153] D. Sasaki, S. Obayashi, and H. J. Kim. Evolutionary Algorithm vs. Adjoint Method Applied to SST Shape Optimization. In *The Annual Conference of CFD Society of Canada, CFD2001*, Waterloo, Canada, May 27–29 2001.

[154] M. Schoenauer and S. Xanthakis. Constrained GA Optimization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 573–580, San Mateo, California, July 1993. Morgan Kaufmann.

[155] M. Sefrioui. *Algorithmes Evolutionnaires pour le calcul scientifique. Application à l'electromagnetisme et à la mécanique des fluides numériques*. PhD thesis, University Pierre et Marie Curie, Paris, April 1998.

[156] M. Sefrioui and J. Périaux. A Hierarchical Genetic Algorithm Using Multiple Models for Optimization. In *Proceedings of the Sixth International Conference Parallel Problem Solving from Nature (PPSN-VI)*, pages 879–888. Springer, 2000.

[157] M. Sefrioui and J. Periaux. Nash Genetic Algorithms: Examples and Applications. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 509–516, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 July 2000. IEEE Press.

[158] M. Sefrioui, J. Périaux, and J.-G. Ganascia. Fast Convergence Thanks to Diversity. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, San Diego, California, February 29 1996. IEEE Computer Society Press, MIT Press.

[159] M. Sefrioui, J. Périaux, and B. Mantel. Parallel Genetic Algorithms: A New Shared Information Technology Applied to Domain Decomposition Based Flow Computations on Parallel Computers. In *Proceedings of the Second ECCOMAS Conference on Numerical Methods in Engineering*, Paris, 1996.

[160] M. Sefrioui, K. Srinivas, and J. Périaux. Aerodynamic Shape Optimization Using a Hierarchical Genetic Algorithm. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Barcelona, Spain, September 2000.

[161] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary Aerospace Design Optimization Survey of Recent Developments. Technical Report 96-0711, NASA/American Institute of Aeronautics and Astronautics, 1996.

[162] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments. *Structural Optimization*, 14:1–23, 1997.

[163] A. Sommerer, T. Lutz, and S. Wagner. Numerical Optimization of Adaptive Transonic Airfoils With Variable Camber. In *Proceedings of the 22nd ICAS Congress*, Harrogate, United Kingdom, August 27–September 1 2000. ICAS Paper ICA2.11.

[164] K. Srinivas. Calculation of Cascade Flows by a Modified CUSP Scheme. *Computational Fluid Dynamics Journal*, 2:285–295, 1999.

[165] A. Srivastava, K. Hacker, K. Lewis, and T. Simpson. Development of a Kriging Based Surrogate Approximation Method for Large Scale Systems. *International Journal for Product and Process Improvement, submitted*, 1999.

[166] M. Sweet, C. P. Earls, and B. Spitzak. FLTK 1.1.6 Programming Manual Revision 6. Technical report, 2004.

[167] S. Takahashi, S. Obayashi, and K. Nakahashi. Transonic Shock-free Wing Design with Multiobjective Genetic Algorithms. In *Proceedings of the International Conference on Fluid Engineering*, Tokyo, Japan, July 1997. JSME.

[168] Z. Thomas and A. Green. Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research. In *Proceedings of the 30th AIAA Fluid Dynamics Conference*, Norfolk, VA, June 1999. AIAA Paper 1999-3798.

[169] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, London, 1995.

[170] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective Evolutionary Algorithm Test Suites. In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant, and Gary B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, San Antonio, Texas, 1999. ACM.

[171] D. A. van Veldhuizen, J. B. Zydallis, and G. B. Lamont. Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms. *IEEE Trans. Evolutionary Computation*, 7(2):144–173, 2003.

[172] J. C. Vassberg, P. G. Buning, and C. L. Rumsey. Aerodynamic Shape Optimization of Complete Aircraft Configuration, AIAA Paper-2002-0840. In *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2002.

[173] S. Wakayama and I. Kroo. Subsonic Wing Planform Design Using Multidisciplinary Optimization. *AIAA Journal of Aircraft*, 12(4):746–753, July–August 1995.

[174] J. Wakunda and A. Zell. Median Selection for Parallel Steady-State Evolution Strategies. In *Proceedings of the Sixth International Conference Parallel Problem Solving from Nature (PPSN-VI)*, pages 405–414. Springer, 2000.

[175] G. Wang. Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points. *Transactions of the ASME, Journal of Mechanical Design*, 125:1–11, June 2003.

[176] X. Wang and M. Damodaran. Aerodynamic Shape Optimization Using Computational Fluid Dynamics and Parallel Simulated Annealing Algorithms. *AIAA Journal*, 39(8):1500–1508, August 2001.

[177] J. Werner. *Optimization: theory, and applications*. Braunschweig Vieweg, 1984.

[178] R.P. Weston, J. C. Townsend, T. M. Eidson, and R. L. Gates. A Distributed Computing Environment for Multidisciplinary Design. In *Proceedings of Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, Florida, September 7–9 1994.

[179] E. J. Whitney. *A Modern Evolutionary Technique for Design and Optimisation in Aeronautics*. PhD thesis, The University of Sydney, 2003.

[180] E. J. Whitney, L. González, K. Srinivas, and J. Périaux. Multi-criteria Aerodynamic Shape Design Problems in CFD Using a Modern Evolutionary Algorithm on Distributed

Computers. In Armfield S., P. Morgan, and K. Srinivas, editors, *Proceedings of the Second International Conference on Computational Fluid Dynamics (ICCFD2)*, pages 597–602, Sydney, Australia, July 2002. Springer.

[181] E. J. Whitney, M. Sefrioui, K. Srinivas, and J. Périaux. Advances in Hierarchical, Parallel Evolutionary Algorithms for Aerodynamic Shape Optimisation. *JSME International Journal*, 45(1):23–28, February 2002.

[182] E.J. Whitney, L.F. González, K. Srinivas, and J. Périaux. Adaptive Evolution Design Without Problem Specific Knowledge. In *Proceedings of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'03)*, 2003.

[183] D. H. Wood. Dual Purpose Design Of Small Wind Turbine Blades. *Wind Engineering*, 28(5):511–527, 2004.

[184] K. Yamamoto and O. Inoue. Applications of Genetic Algorithm to Aerodynamic Shape Optimization. In *Proceedings of the 12th AIAA CFD Conference–Collection of Technical Papers Part I*, pages 43–51. AIAA, 1995.

[185] E. Zitzler. Evolutionary Algorithms for Multiobjective Optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 19–26, Athens, Greece, 2001. International Center for Numerical Methods in Engineering (CIMNE).