6th World Congresses of Structural and Multidisciplinary Optimization
Rio de Janeiro, 30 May - 03 June 2005, Brazil

Multidisciplinary Design Optimisation of Unmanned Aerial Vehicles (UAV) using Multi-Criteria Evolutionary Algorithms

Luis. F. González[1], K. Srinivas[1], Jacques Périaux[2] and Eric J. Whitney[1],
[1]School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Sydney, Australia
{gonzalez, eric,ragh}@aeromech.uysd.edu.au
[2]INRIA Sophia Antipolis, OPALE project associate and Pole Scientifique Dassault Aviation 78 Quai Marcel Dassault, 99214 Saint-Cloud, France.
jperiaux@free.fr

A growing area in aerospace engineering is the use and development of Unmanned Aerial Vehicles (UAVs) for military and civilian applications. There are difficulties in the design of these vehicles because of the varied and non-intuitive nature of the configurations and missions that can be performed. Similar to their manned counterparts, the challenge is to develop trade-off studies of optimal configurations to produce a high performance aircraft that satisfy mission requirements The goal in the present study is to address these issues from a multi-criteria and multidisciplinary design optimisation (MDO) standpoint.

Traditional deterministic optimisation techniques for MDO are effective when applied to specific problems and within a specified range. These techniques are efficient in finding locally optimum solutions if the objective and constraints are differentiable. If a broader application of the optimiser is desired or when the problem is multi-modal, involve approximation, is non-differentiable or involve multiple criteria and multi-physics, robust and alternative numerical tools are required. Emerging techniques such as Evolutionary Algorithms (EAs) have shown to be robust as they require no derivatives or gradients of the objective function, have the capability of finding globally optimum solutions amongst many local optima, are easily executed in parallel and can be adapted to arbitrary solver codes without major modifications.

This paper examines the requirements, initial development, and application of a framework for Multidisciplinary Design and Optimisation (MDO) of UAVs. The framework includes a Graphical User Interface (GUI), a robust EA optimiser, several design modules, mesh generators and post-processing capabilities in an integrated platform. The application of the method is illustrated on a multi-criteria and multidisciplinary design problem. Results indicate the robustness of the method in finding optimal solutions and trade-offs between the disciplinary analyses and producing a set of individuals represented in an optimal Pareto front.

Keywords: Multidisciplinary Design Optimisation (MDO), Evolutionary Design, Parallel Computing.

1. Introduction

Aerospace design and optimisation is a complex process where the design team is presented with a problem that involves numerous criteria and multi-physics. Hence a systematic approach, which is regarded as Multidisciplinary Design Optimisation (MDO) that accounts for the coupling between the disciplines and variables, is desirable. These problems may be multi-modal, non-convex or discontinuous. Wing design is an example of a multi-criteria MDO problem as there is a strong interaction between aerodynamics and structures.
There are different approaches for solving a MDO problem using traditional optimisation techniques [1-4]. These techniques are effective when applied to specific problems and within a specified range and are efficient in finding optimal global solutions if the objective and constraints are differentiable and if the designer guesses where the global solution is located. Robust and alternative numerical tools are required if a broader application of the optimiser is desired or if the complexity of the problem arises because it is multi-modal, involve approximations, is non-differentiable or involve multiple criteria and physics.
One of the emerging techniques for optimisation is Evolutionary Algorithms (EAs) [5, 6]. These are based on Darwinian evolution; whereby populations of individuals, which represent the design variables, evolve over a search space and generate offspring by the use of different mechanisms such as mutation, crossover and selection. An attractive feature of EAs is that they evaluate multiple populations of points and are capable of finding simultaneously a number of solutions in a Pareto set. EAs have been successfully applied to different aircraft, wing, aerofoil and rotor blade design and optimisation problems [7-11]. One drawback of EAs is that they are slow in converging, as they require a large number of function evaluations to find optimal solutions and have poor performance with increasing number of variables. Hence the continuing effort has been on developing robust and fast numerical techniques to overcome these challenges and facilitate the complex task of design and optimisation in aeronautics. In this work we describe the design and implementation of a framework for the design and optimisation of aeronautical systems. This framework uses a robust evolutionary technique, which is scalable to preliminary design studies with higher fidelity models for the solution and is applicable to the design and optimisation of UAV systems.
The rest of the paper is organised as follows, section 2 summarises some requirements for a robust framework for multi-criteria and multidisciplinary design optimisation, section 3 describes the design of the framework, implementation of the framework is presented in section 4, section 5 illustrates the application of the method to real world problems. Finally section 6 provides summary and future directions for the research.

2. Requirements for a Multi-Criteria Multidisciplinary Design Optimisation Framework in Aeronautics.

Complex optimisation problems involve non-linearities, multi-criteria, multidisciplinary considerations. In order to handle these complexities it is desirable to develop a system, which facilitates integration of a series of design and analysis tools, graphical user interfaces (GUI) and post-processing capabilities. This section focuses on the requirements, development and implementation of such a framework using Evolutionary Algorithms in which different multidisciplinary and multi-criteria problems can be analysed. The fundamental idea with the framework is to simplify the task of integration to the design team so that it can focus on the problem itself. The idea on the development of this framework is a generic system that can be easily developed, maintained and extended. The basic requirements for a MDO framework can be subdivided into architectural design and information access, optimisation methods, problem formulation and execution [12,13].

Architectural Design

The 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.
- based on standards.

Information Access

The framework should:

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

Optimisation Methods

The framework should:

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

Problem Formulation and Execution

The 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 is desirable.

3. Design

With these requirements in mind the general scope for the framework was identified. The framework developed in this work addresses these requirements. Figure 1 is a representation of different components to satisfy these requirements. The framework has a GUI, a robust optimisation tool, several analysis modules and capabilities for parallel computing, mesh generators, CAD, Design of Experiments (DOE) and post-processing.

4. Implementation

Integrating these components is a complex task. This work involves the development of the architecture, a GUI, definition and implementation of a robust optimisation tool, a general formulation for MDO and multi-criteria problems, and capabilities for pre and post-processing. The DOE capability has been accounted for, but has been evaluated only on simple mathematical test problems. The following sections describe how the requirements are satisfied within the framework.

4.1 Architectural Design and Information Access

To satisfy the architectural design requirements the platform uses an object-oriented approach in C++. The benefits of using object-oriented software are the ease of implementation and extension of software in a modular fashion by the use of classes and methods. In an industrial and academic environment the need for a user-friendly application is required hence a simple GUI was designed. There were many considerations and options for the GUI development, but knowledge in C++ and the use of object-oriented principles were the main considerations. The Fast Toolkit (FLTk) library [14] was selected for this task. This toolkit provides a friendly and easy to use environment for different implementations. The GUI is simple and modular on its implementation and consists of five main modules as illustrated in figure 2. The main modules are: Design and Analysis Module, Design of Experiments Module, Post-processing Module and Parallel Processing Module. 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.

4.1.1 Design and Analysis Module

As illustrated in figure 2 the Design Module allows the user to conduct a single design and optimisation for different aeronautical applications and mathematical test cases. So far this module contains five sub-modules for aerofoil, multi-element aerofoil, nozzle, wing, aircraft and mathematical test functions design and optimisation. As designed the framework is flexible and provides for ease of implementation of other design modules under development such as those for propeller, cascade aerofoils or rotor blade design.

4.1.2 Development of Aeronautical Design Modules

Before using a sub-module it is necessary to develop a design module interface, this comprises a series of files written in C++ that allow communication between the GUI, analysis codes, the optimiser and the parallel processing capability. When designing the interface a choice has to be made depending upon if the source code for the analysis tool was available or not. In the current implementations minimal modification to the source code was required, ideally it is desirable to operate only through the input/output files of the analysis tool. In the implementations considered, a design template was used in conjunction with one or two additional files which contain the necessary linking subroutines allowing a rather fast implementation of the design modules. So far, there are subroutines for aircraft, nozzle, wing and full aircraft configuration design. Each of these options allows the user to perform a single design analysis or a full optimisation. A general algorithm for the implementation of a new design module is represented in algorithm 1.

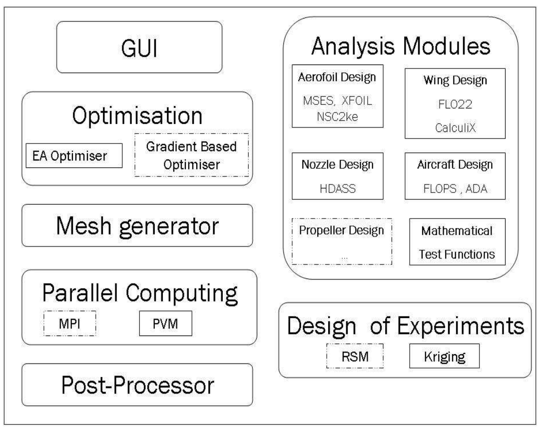

Figure 1. MDO Framework

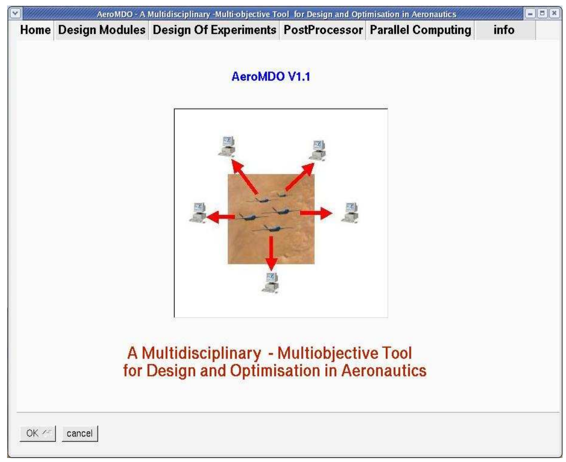

Figure 2. GUI Sample

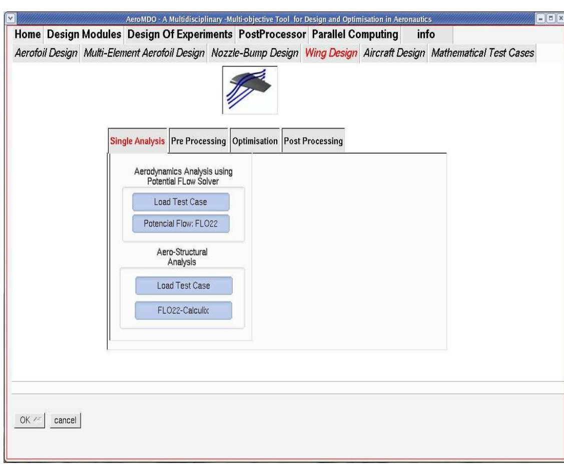

Figure 3. Wing Design and Optimisation Module

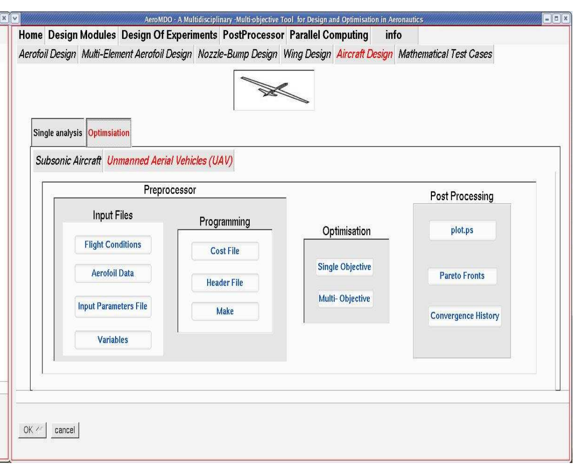

Figure 4. Aircraft Design Module

Wing Design and Optimisation Module: This module allows the user to conduct a single analysis on a wing or an optimisation study. These could be studies in one or several objectives or with multiple disciplines. Figure 3 illustrates this module. Details on the analysis tools used in this module and its application to multi-criteria and multidisciplinary wing design are presented in section 5.

Aircraft Design and Optimisation Module: This module allows the user to analyse and optimise different problems related to aircraft external configuration design. It can be used to design and optimise different subsonic, Unmanned Aerial Vehicles, transport or supersonic aircraft. Single or multi-criteria optimisation studies can be performed. Comparison of different multi-criteria analysis such as Pareto optimality and Nash equilibrium approach is possible. Figure 4 illustrates this module. The user can select from two different analysis codes: An object-oriented Aircraft Design and Analysis Software (ADA) developed by the first author or using the Flight Optimisation System (FLOPS) software developed by Arnie McCullers at NASA Langley. FLOPS [15], a more robust solver, is a workstation-based code has capabilities for conceptual and preliminary design and evaluation of advanced concepts. The sizing and synthesis analysis in FLOPS are multidisciplinary in nature. It has numerous modules and capabilities for takeoff, performance, structural, control, aerodynamic and noise analysis. This code is used in some universities for MDO development as well as aerospace firms and governmental institutions. It allows an integral multidisciplinary analysis for the entire aircraft mission and calculation of performance parameters such as range, endurance, takeoff field length and landing field length.

Aerofoil Design and Optimisation Module: This module allows the user to perform a single analysis or a full aerofoil optimisation routine. Three different CFD codes can be used: A panel method (XFOIL)[16], an Euler + boundary layer (MSES) [17] and a Navier-Stokes analysis (NSC2ke [18])

Multi-element Aerofoil Design and Optimisation Module: Similar to the aerofoil design module this module allows the user to perform a single analysis or a full optimisation, the user can choose from an Euler or Navier-Stokes analysis.

Nozzle-Bump Design and Optimisation Module: The Nozzle -Bump design module allows a single two-dimensional analysis or optimisation using the CUSP solver developed by Srinivas [19].

Mathematical Test Functions Module: This module allows the user to design, and evaluate single, or multi-criteria mathematical test functions which give confidence in the robustness and performance of the optimisation method before deciding on its application to real world problems. The current implementation includes mathematical test function for single or multiple criteria, constrained optimisation, DOE and non-linear goal programming problems.

4.1.3 Design of Experiments Module

In the implementation considered in this research, the optimiser uses an EA for the optimisation, but as discussed in section 1, one of the drawbacks of EAs is that they suffer from slow convergence. By providing a DOE capability into the framework we wish to hybridise the desirable characteristics of EAs and surrogate models such as RSM to obtain an efficient optimisation system. Within this context, the DOE samples a number of design candidates at which the analysis code (CFD) will run), the surrogate model is then constructed for the computationally expensive problem. Different sampling and DOE strategies can be used; Latin hypercube, Response Surface Methods or DACE/Kriging. There is sufficient literature and software developed specifically for DOE, after a careful selection of software packages it was decided to implement the DACE tool box [20] which is robust and allows different options for sampling strategies and DOE. This software was ported to Octave (a mathematical package common in most UNIX installations) and integrated with the framework but if desired, different DOE methods can be implemented.

4.1.4 Parallel Computing Module

Recent work on multi-criteria parallel evolutionary algorithms has allowed significant performance and robustness gains in global and parallel optimisation [21, 22]. This module allows the users to dynamically create, add or delete nodes on the parallel implementation. The framework considers the implementation of 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). [23]

4.1.5 Post Processing

The approach considered for post-processing was to use a combination of visualisation capabilities within each analysis software, and the use of GNUplot (a graphics software common in most UNIX installations). Common to all design modules is visualisation of the evolution progress of the fitness function and Pareto fronts for multi-criteria problems. Post-processing tools on each analysis module include a top view of the wing planforms and a general 3D view of the resulting aircraft configurations. Visualisation tools within each analysis software module include the pressure coefficient distribution on the aerofoil using an Euler + Boundary Layer solver or pressure or Mach contours using a Navier-Stokes solver. Examples of some visualisation capabilities are presented in section 5.

4.2. Optimisation Methods.

The second requirement is the incorporation of robust optimisation tools. In this research we use and extended the Hierarchical Asynchronous Parallel Evolutionary Algorithm (HAPEA) approach developed by Whitney [24, 25] for multidisciplinary and multi-criteria analysis.

4.2.1 Hierarchical Asynchronous Parallel Evolutionary Algorithms (HAPEA)

The foundation of the HAPEA algorithm lie upon traditional evolution strategies and incorporate the concepts of multi-criteria optimisation, hierarchical topology, parallel computing and asynchronous evaluation.

4.2.2 Evolution Strategies (ES)

Evolution strategies are a special instance of EAs [6]. ES were first developed by Ingo Rechenberg and Hans-Paul Schwefel [26]. Their first algorithm worked using only two individuals, a population (μ) with a single individual or parent and one offspring (λ). Each individual was real coded; each problem variable was assigned a floating-point value in the chromosome. The variation operator involved applying a random mutation to each floating-point value in the parental chromosome to arrive at the offspring individual. The selection operator was entirely deterministic, and was simply the result of a competition between parent and offspring to determine which remained. In the standard nomenclature this strategy is denoted the (1+1) ES, the first digit indicating the number of parents, the ' + ' indicating competition between parents and offspring and the final digit indicating the number of offspring. From the beginning ES have been designed almost exclusively with real coding in mind, as opposed to original EA variants where real parameter optimisation comes about by the piecewise interpretation of the binary chromosome associated with each individual. An evolution strategy would therefore seem a logical starting point for evolutionary optimisation using real coded problem variables.

Subsequent developments in ES introduced multi-membered populations for both parents and offspring. The first algorithm of this type was the (μ+1) ES [6,26]. This worked by applying some variation operator to the parent population 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 the worst member of the population. Both the (1+1) ES and the (μ+1) ES used deterministic control of the mutation size (variations applied to design variables) which were normally distributed when applied to real coded problems. A pseudo code of a canonical evolution strategy is illustrated in algorithm 2. A population (μ_o) is initialised and then evaluated. Then for a number of generations (g) and while a stopping condition (maximum number of function evaluation or target fitness value) is not met, offsprings (λ^{g+1}) go recursively through the process of recombination, mutation, evaluation and selection. Another major advantage of ES and of HAPEA is that these algorithms can tackle multi-criteria problems directly (by considering vector fitnesses and not the more traditional weighted aggregation of several criteria).

4.2.2 Multi-Criteria (Multi-objective) EAs (MOEAs)

Most real world problems involve conflicting criteria and there is no unique optimum, but a set of compromised individuals known as Pareto optimal solutions or non-dominated individuals. The Pareto Optimality principle is one where a solution to a multi-criteria problem is considered Pareto optimal if there is no other solutions that satisfy better all the criteria simultaneously. Figure 5 illustrates the concept for a problem with two conflicting criteria. The objective of Pareto Optimality is then to provide a set of Pareto optimal solutions that trade-off the information among the conflicting criteria.

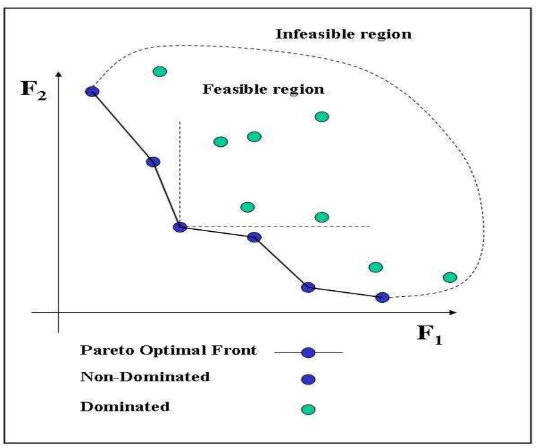

Figure 5. Pareto Optimality

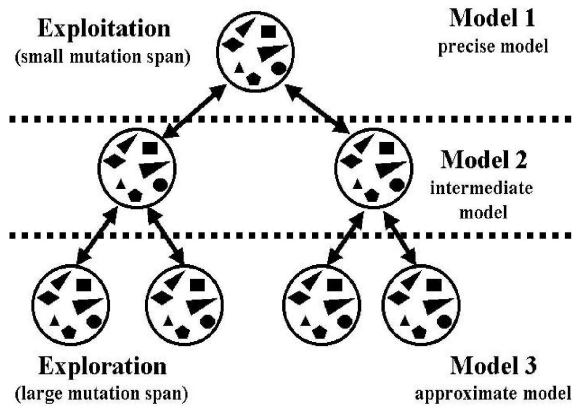

Figure 6. Hierarchical Topology

As EAs evaluate multiple populations of points, they are capable of finding a number of solutions in a Pareto set. The optimisation method-HAPEA works using a Pareto tournament selection operator that selects the non-dominated individuals from the population to generate the Pareto Optimal front28.

There are some problems on applying EAs to multi-criteria problems, Deb [27] for example describe and analyse problem features that might cause a multi-criteria evolutionary algorithm to converge to the true Pareto front. The optimisation method-HAPEA was tested and proves to be robust and efficient to find optimal Pareto fronts or Nash Equilibria for multi-modal, multi-criteria, deceptive, convex, non-convex, discontinuous and non-uniformly represented Pareto optimal fronts [24,25,28] problems.

4.2.3 Hierarchical Topology

The optimisation method is also designed to handle multiple fidelity models for the solution [29]. Figure 6a shows a representation of this formulation. 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. To take full benefit of a hierarchical structure, the top layer uses a very precise model meaning a time-consuming solver. But at the same time, the sub-populations of the bottom layer need not yield a very precise result, as their main goal is to explore the search space. What this means is that the most refined computational grid or the most robust solver that better describes the physics involved is used at the top level. In this hierarchical topology, solutions go up and down the layers and progressively the best solutions keep going up until they are completely refined. The benefit of such approach as compared to a single population EA is that instead of having a time consuming solver on the top level to perform all the evaluations, the load is distributed to the intermediate and bottom levels. At these levels there is an exploration phase where there is no need for great precision, the most time consuming solvers at the top levels should be used only for the most promising solutions as they become more refined. A comparison study of the results obtained using a single population or a hierarchical topology of solvers or resolutions can be found in Whitney et al. [25].

4.2.4 Parallel Computing and Asynchronous Evaluation
EAs are particularly adaptable to parallel computing, candidate individuals can be sent to remote machines, evaluated and incorporated back into the optimisation process [21, 22]. In this paper the optimisation was parallelised on a network of computers at The University of Sydney. The system has ten machines with performances varying between 2.0 and 2.8 GHz. The master computer carries on the optimisation process while the remote machines compute the solver code. The message-passing model used is the Parallel Virtual Machine (PVM). The parallel implementation requires modifications to the canonical ES [26], which ordinarily evaluates entire populations simultaneously.

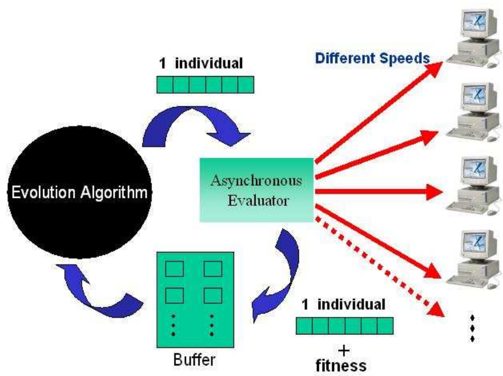

Figure 7. Parallel Computing and Asynchronous Evaluation

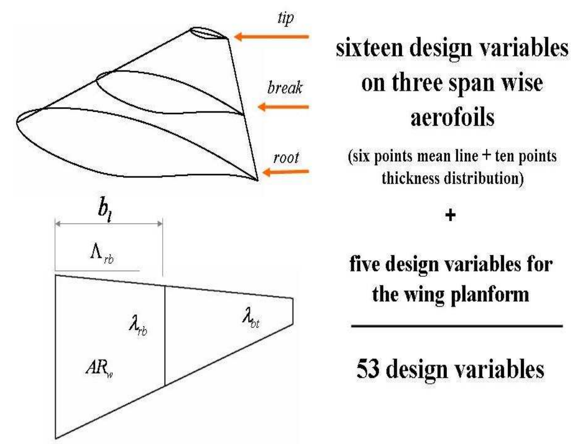

Figure 8. Design variables for multidisciplinary wing design.

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 EAs10. Consequently solutions can be generated and returned out of order. This allows the implementation of an asynchronous fitness evaluation giving the method its name. Figure 6b shows a schematic representation of this approach. The optimisation method-HAPEA has been applied to different design problems including deceptive and multi modal Pareto solutions, viscous two-dimensional nozzle optimisation and multi-criteria constrained aerofoil design problems. In all these cases the algorithm successfully converged to an optimal solution or set of solutions. The benefit of using an asynchronous approach can be found in Whitney et al. [25].

4.2.5 Implementation of Different Legacy codes
Legacy codes in different programming languages C, C++, Fortran 90, and Fortran 77 have been implemented. The optimiser has been successfully coupled with the following aerodynamic and analysis software: FLO22 [30], FLOPS, ADA, XFOIL, MSES, CalculiX [31] and NSC2Ke. One of the benefits of using an Evolutionary optimiser is that EAs require no derivatives of the objective function. The coupling of the algorithm with different analysis codes is by simple function calls and input and output data files.

4.3 Problem Formulation and Execution
A third requirement is on how to incorporate different multi-criteria and MDO formulations. There are many strategies proposed for multi-criteria and MDO and the development of these optimisation methods, architectures and decomposition methodologies has been an active field of research. The framework developed in this research is applicable to an integrated analysis or distributed MDO analysis 1-4. Examples on the application of the method for these formulations are presented in section 5. The framework has also capabilities for parallel computing; different candidate members of the population can be sent to remote parallel heterogeneous computers. Once a solution is computed it is returned to the optimiser and framework.

5. Applications

The framework has been used to evaluate several real world problems including inverse and direct problems for aerofoil, high- lift aircraft system, multidisciplinary and multi-criteria wing and aircraft design and optimisation problems [24,25,28]. In the following we illustrate the application of the method for a multi-criteria multidisciplinary UAV wing design and optimisation problem. This test case considers the UAV to be flying at cruise Mach number 0.69 at an altitude of 10000 ft. The wing area is set to 2.94 m2 and the corresponding CL is fixed at 0.19. For the solution we initially compute the pressure distribution over the wing using a potential flow solver to obtain the wing aerodynamics characteristics that include the spanwise pressure distribution, CL and total drag coefficients CDw. Concentrated loads replace the lift distribution and the spar cap area is calculated to resist the bending moment. The weight is then approximated as the sum of the span-wise cap weight. The strong interaction between the aerodynamic pressure distribution and the structural deflections is ignored.

5.1.1. Design Variables and Constraints.
The wing geometry is represented by three aerofoil sections and five variables for the wing planform. In total fifty-three design variables are used for the optimisation. Figure 7 illustrates the main design variables and table 1 indicates their upper

and lower bounds for the wing planform. The aerofoil geometry is represented by the combination of a mean line and thickness distribution, which is a very common concept in classical aerodynamics [32]. Both lines are represented by Bézier curves with leading and trailing edge points fixed at (0.0,0.0) and (1.0,0.0) respectively, and a variable number of intermediate control points whose x-positions are fixed in advance and whose y-heights form the problem unknowns. In this case we take six free control points on the mean line and ten free control points on the thickness distribution.
Constraints are imposed on minimum thickness ($t/c \geq 0.14$ root aerofoil, 0.12 intermediate aerofoil, and 0.11 tip aerofoil) and position of maximum thickness. ($20\% \leq t/c \leq 55\%$). If any of these constraints is violated both fitness are linearly penalised to ensure an unbiased Pareto set.

Table 1. Upper and lower bounds for multidisciplinary wing design variables.

Description	Lower Bound	Upper Bound
Wing Aspect Ratio [AR]	3.50	15.00
Break to root Taper [λbr]	0.65	0.80
Break to tip Taper [λbt]	0.20	0.45
Wing 1/4 Chord inboard Sweep, deg [Λi]	10.00	25.00
Break Location, [bl]	0.30	0.45

5.1.2. Aerodynamics and Weight Analysis
The aerodynamic characteristics of the wing configurations are evaluated using FLO22, a 3-D full potential wing analysis software. This program uses sheared parabolic coordinates and accounts for wave drag [30]. FLO22 was developed by A. Jameson and D. Caughey for analysing inviscid, isentropic, transonic shocked flow past 3-D swept wing configurations. The algorithm is based on free stream Mach numbers limited by the isentropic assumption and weak shock waves are automatically captured wherever they occur in the flow. Also the finite difference form of the full equation for the velocity potential is solved by a relaxation method, 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 transonic free stream velocities. Details on the formulation and implementation can be found in Reference [30].
The fixed lift requirement can be satisfied by performing an extra two function evaluations by varying the angle of attack at the wing root and assuming a linear variation of the lift coefficient. The lift distribution is summed into concentrated loads. The wing weight is estimated from the wing spar cap area designed to resist the bending moment. The local stress has to be less than the ultimate tensile stress in this case for Aluminium Alloy 2024 -T6 $\leq \sigma_{adm.}$

5.1.3. Fitness Functions
The two fitness functions to be optimised are defined as minimisation of wave drag (CD_{wave}) and minimisation of the sum of the spanwise cap weight (WSC) to resist the bending moment.

$$\min(f_1): \quad f_1 = c_{Dwave} \tag{1}$$

$$\min(f_2): \quad f_1 = \sum W_{SC} \tag{2}$$

5.1.4. Implementation
We use the wing design and optimisation module to solve this problem (section 4.1.2) and considered two approaches for the solution; in the first approach the optimiser is configured as a traditional EA with a single population model and computational mesh of 96 x 12 x 16 for the FLO22 code. The second approach uses a hierarchical topology of resolutions with the following settings:

Top Layer: A population size of 30, and a computational mesh of 96 x 12 x 16.
Middle Layer: A population size of 30 and a computational mesh of 72 x 9 x 12.

5.1.5. Numerical Results

The algorithm was run five times for 2000 function evaluations and took in average six hours to compute. Figure 8 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. By inspection we can see that the use of a hierarchical approach gives an overall lower front as compared to a single model approach. The combination of low fidelity models for a rapid exploration of the design space and higher fidelity models for the most promising solutions has used in the optimisation. Figure 9 illustrates the Pareto front for the hierarchical approach and a representative top view of the wing geometries. Figure 10 shows the corresponding aerofoils at root, break and tip for some of the Pareto configurations and table 2 indicates the final values design variables.

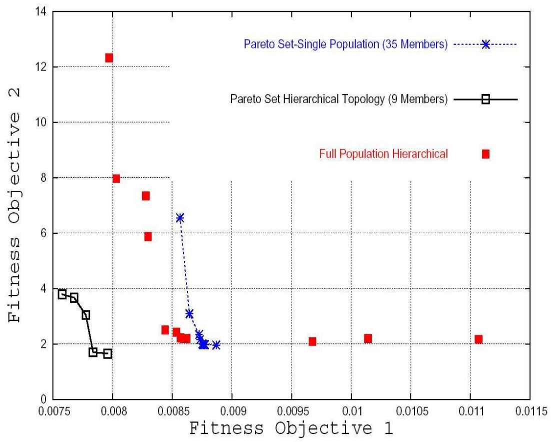

Figure 8. Pareto Fronts after 2000 function evaluations.

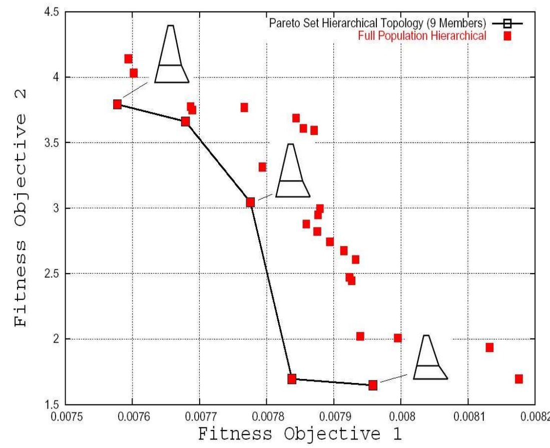

Figure 9. Pareto fronts and wing planforms.

Table 2: Optimum design variables for some members of the Pareto front.

Description	Pareto Member 0	Pareto Member 4	Pareto Member 15
Wing Aspect Ratio [AR]	6.92	6.07	2.56
Wing 1/4 Chord inboard Sweep, deg [Λi]	10.83	10.02	20.30
Wing semispan, ft	2.14	2.00	1.30
Break to root Taper [λbr]	0.74	0.68	0.69
Break to tip Taper [λbt]	0.31	0.24	0.35

This problem demonstrates the use the framework for UAV wing design and optimisation. Results indicate a computational gain on using a hierarchical topology of fidelity models as compared to a single model during the optimisation. Results also show how the algorithm was capable of identifying the trade-off between the multi-physics involved and provide classical aerodynamic shapes as well as alternative configurations from which the design team can choose and proceed into more detailed phases of the design process.

6. Conclusions

This paper presents the requirements, formulation and implementation of a robust design framework with which different aeronautical problems can be analysed. The paper gives a brief description of the different components of the framework. These include several algorithms for design and optimisation, a GUI and different modules for design, optimisation, mesh generation, CAD systems, post-processing and parallel computing. Hence we have within the framework, a complete set of numerical tools for analysing and optimising real world aeronautical and UAV problems. The application of the methodology is illustrated on a wing and UAV multi-criteria MDO problem. The method is capable of identifying the trade-off between the multi-physics involved and provides classical aerodynamic shapes as well as alternative configurations from which the design team can choose. It was observed that there was a computational gain on using a hierarchical topology of fidelity models as compared to a single model during the optimisation. As developed, the evolution algorithm-solver coupling is comparatively easy to setup as it only requires 'payoff' information from the solver used. The benefits of using this framework to provide solutions for single and multi-criteria problems in aircraft/UAV systems are straightforward for the designer. Further research in this design environment using higher fidelity Navier-Stokes turbulent flow analysers with unstructured adapted meshes in the optimisation procedure is needed and applications of the method to more complex aerodynamic configurations are presently under investigation.

7. Acknowledgments

The authors gratefully acknowledge Mourad Sefrioui, Dassault Aviation for fruitful discussions on Hierarchical EAs, and also Steve Armfield and Patrick Morgan, University of Sydney for access to the cluster of computers. We would like to thank Arnie McCullers, NASA Langley RC who kindly provided the FLOPS software, Marc Drela, MIT for providing the XFOIL and MSES software and Antony Jameson, Stanford Univ. and Shigeru Obayashi, Tohuku Sendai University for the use of the FLO22 software for the full potential flow analysis software around a wing.

8. References

1. P. Bartholomew "The Role of MDO within Aerospace Design and Progress towards an MDO Capability, AIAA-98-4705, pp 2157-2165, 7th AIAA/USAF/NASA/ ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA , St. Louis, Mo,1998.

2. R. Braum and P. Gage and I. Kroo and I. Sobieski, "Implementation and Performance Issues in C.O , NASA-AIAA-96-4017, 1996.
3. J. Sobieski, RT Haftka, Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments, AIAA Paper No. 96-0711, 1996.
4. Z. Thomas and A. Green, "Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research" AIAA Paper-1999-3798, Jun, 1999
5. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley,1989.
6. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial Intelligence. Springer-Verlag, 1992.
7. 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, IEEE, October 1998
8. A. Oyama, M.-S. Liou, and S. Obayashi. Transonic Axial-Flow Blade Shape Optimization Using Evolutionary Algorithm and Three-Dimensional Navier-Stokes Solver, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, September 2002.
9. 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, Morgan Kaufmann, July 1999.
10. D. Raymer "Aircraft Design: A Conceptual Approach", American Institute of Aeronautics and Astronautics American Institute of Aeronautics and Astronautics, Third Edition, 1999
11. 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.
12. A. J. Booker, J. E. Dennis, Jr., P. D. Frank, 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.
13. A. O. Salas and J. C. Townsend, Framework Requirements for MDO Application Development, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, AIAA 98-4740, September 2-4, 1998, pp. 11.
14. M. Sweet, C.P Earls and B. Spitzack. FLTK 1.1.6 Programming Manual, Revision 6. www.fltk.org.
15. A. McCullers. *FLOPS User's Guide*, Release 6.02, NASA Langley Research Center, March 2003.
16. M. Drela. *XFOIL 6.94 User Guide*. MIT Aero Astro, 2001.
17. M. Drela. *A User's Guide to MSES V2.3,"* Feb. 1993.
18. B. Mohammadi. Fluid Dynamics Computation with NSC2KE, User-Guide, Release 1.0. Technical Report RT-0164, INRIA, May 1994.
19. K. Srinivas. Calculation of Cascade Flows by a Modified CUSP Scheme, Computational Fluid Dynamics Journal, 1999,2,pp 285-295.
20. S.N. Lophaven, H. B. Nielsen and J. S\ondergaard, Aspects of the Matlab Toolbox DACE, 2002, 44, IMM-TR-2002-13.
21. E. Cantu-Paz, Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Pub, 2000.
22. D. A. Van Veldhuizen, J.B. Zydallis and G. B. Lamont. Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, pp. 144--173, April 2003.
23. 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*. Massachusetts Institute of Technology, 1994
24. E. J. Whitney. *A Modern Evolutionary Technique for Design and Optimisation in Aeronautics*. PhD Thesis, The University of Sydney, 2003.
25. E. J. Whitney, M. Sefrioui, K. Srinivas, J. Périaux: "*Advances in Hierarchical, Parallel Evolutionary Algorithms for Aerodynamic Shape Optimisation*", JSME (Japan Society of Mechanical Engineers) International Journal, Vol. 45, No. 1, 2002.
26. T. Bäck, G. Rudolph, and H. P. Schwefel. Evolutionary programming and evolution strategies: Similarities and differences. Proceedings of the Second Annual Conference on Evolutionary Programming, Evolutionary Programming Society, San Diego, CA, pp.11-22, 1993.
27. K. Deb, Multi-criteria Optimization Using Evolutionary Algorithms, Wiley, 2003.
28. L. 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.
29. M. Sefrioui and J. Périaux. A Hierarchical Genetic Algorithm Using Multiple Models for Optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature, PPSN VI*, pages 879-888, Springer, 2000.
30. A. Jameson, D. Caughey, P. Newman and R. Davis. A Brief Description of the Jameson Caughey NYU Transonic Swept-Wing Computer Program *FLO22*. NASA Technical Memorandum, NASA TM X-73996, Dec. 1976.
31. Dhondt, G., and Wittig, K. CalculiX: A Free Software Three-Dimensional Structural Finite Element Program, http://www.calculix.de.
32. I. H. Abott and A. E. Von Doenhoff. *Theory of Wing Sections*. Dover, 1980.