Neurocomputing 142 (2014) 39–47

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Evolving parametric aircraft models for design exploration and optimisation

Jonathan Byrne *, Philip Cardiff, Anthony Brabazon, Michael O'Neill

University College Dublin, Belfield, Dublin 4, Ireland

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form
22 April 2014
Accepted 23 April 2014
Available online 6 May 2014

Keywords:
Parametric design
Optimisation
Evolutionary algorithms
Computational fluid dynamics

ABSTRACT

Traditional CAD tools generate a static solution to a design problem. Parametric systems allow the user to explore many variations on that design theme. Such systems make the computer a generative design tool and are already used extensively as a rapid prototyping technique in architecture and aeronautics. Combining a design generation tool with an analysis software and an evolutionary algorithm provides a methodology for optimising designs. This work combines NASA's parametric aircraft design tool (OpenVSP) with a fluid dynamics solver (OpenFOAM) to create and analyse aircraft. An evolutionary algorithm is then used to generate a range of aircraft that maximise lift and reduce drag while remaining within the framework of the original design. Our approach allows the designer to automatically optimise their chosen design and to generate models with improved aerodynamic efficiency. Different components on three aircraft models are varied to highlight the ease and effectiveness of the parametric model optimisation.

1. Introduction

Parametric systems are changing the conceptual design process in the same way as spreadsheets changed finance. Both operate on the same principle. The user defines the relationships in a system and then changes variables in that system to rapidly explore alternative possibilities. Instead of manually creating a CAD model by dragging and dropping components, the parametric design is specified using variables and functions. Just as changing the value in a cell causes the spreadsheet to recalculate all related values, changing a variable that defines part of a model will adapt all the connected components so as to maintain a coherent design. Although there is a longer lead time to implement the initial model, once it is encoded the user can easily create endless variations on the original.

Evolutionary algorithms (EAs) have shown their ability to optimise the shape and form of designs [1,2]. One of the primary considerations when applying an evolutionary algorithm to a design problem is the representation used. The representation limits the search space by defining all the designs the algorithm could possibly generate. Poor representations generate designs that are invalid (internal faces, unconnected parts), infeasible (wrong scale) or missing the desired functionality. Creating a suitable representation is a difficult task that requires knowledge of both programming and of the specific domain.

Parametric systems provide a novel solution to the representation problem. A well-implemented parametric system will only generate valid designs and incorporates domain knowledge. It also allows a designer with no formal programming experience to define the representation for the evolutionary algorithm. The designer provides the initial model and specifies the range limits so as to generate appropriate variations of their design. Parametric models make evolutionary optimisation directly accessible to the designer and allows them to use their domain knowledge to create a representation that generates feasible designs.

This work combines NASA's parametric aircraft system (OpenVSP) and a computational fluid dynamics solver (OpenFOAM) with an evolutionary algorithm to generate a variety of optimised and novel designs. Section 2 gives an overview of parametric design systems and their application in industry. Section 3 describes the fluid dynamics solver used to generate the fitness values for the model. Section 4 discusses previous aircraft optimisation examples that used evolutionary approaches. Three parametric aircraft models are optimised in this work. The settings consistent for all the experiments are shown in Section 5. Section 6 describes the experiments carried out on the blended wing body model where the airfoil and the wing were varied. Section 7 describes the experiments carried out on the Cessna 182 model where the wing was exclusively varied. Section 8 describes the experiments carried out on the MIG 21 model where the wing and the tail section were optimised simultaneously.

* Corresponding author. Tel.: +353 863257989.
E-mail address: jonathan.byrne@ucd.ie (J. Byrne).

http://dx.doi.org/10.1016/j.neucom.2014.04.004
0925-2312/

Finally Sections 9 and 10 discuss the results of the experiments and the conclusions that can be drawn from them respectively.

2. Parametric design

Parametric design defines the relationships between components in a design. Generating a model consisting of hierarchical and geometric relations allows for exploration of possible variations on the initial design while still limiting the search space. Instead of manually placing and connecting components as is done in traditional CAD, component generating algorithms are linked with user definable variables. Defining the relationship between the components prevents invalid design generation. A change to one component will automatically effect a change on any connected component.

Parametric systems traditionally consist of basic components tailored for a particular design problem. An example of this would be the wing, fuselage and engine components in OpenVSP. Predefined components allow for domain knowledge to be embedded in the software and simplifies the design process. Although the user can explicitly define design components by programming them, normally model creation is done by combining existing components using a graphical interface. Many parametric design systems, such as grasshopper [3], are implemented using a drag and drop interface, shown in Fig. 1. The user can then manipulate the input and evaluate the benefit of the component to the overall design. An important aspect of parametric design is that the user observes the effects caused by manipulating a variable in real time, allowing the user to treat the underlying algorithm as a black box. Showing the effect of changing input to the system means that the user does not require an understanding of the underlying mechanics of the system, but instead gives them an intuitive understanding of how the components in a system are related to each other (Fig. 2).

Parametric design tools have now been introduced into mainstream design software. There is the Grasshopper parametric design tool plug-in for the Rhino modelling system [3], Bentley Systems have implemented a program called Generative Components [4] based on the parametric design paradigm and Dassault Systems have developed CATIA, a CAD system combined with a parametric design tool. Parametric functionality was introduced to AutoCAD 2010 to allow for algorithmic manipulation of a design.

Combining parametric systems with structural analysis allows the user to make informed decisions about the geometric alterations during the conceptual design stage [5]. EIFForm is a parametric design system that optimises lattice structures by using a structural analysis and a simulated annealing algorithm. The results have been used to design a structure in the inner courtyard of Schindler house [6]. Bollinger et al. [7] have developed parametric design systems that incorporate structural considerations and have used it to generate roofing structures for the BMW Welt Museum, Munich and the Rolex learning centre, EPFL, Lausanne. CATIA was combined with GSA structural analysis software [8] to evolve roofing structures for a football stadium [5].

The software used in this work is open vehicle sketch pad (OpenVSP). It was originally developed by NASA and Sterling Software as a rapid geometry modeler for conceptual aircraft [9] and has since developed into a stand-alone aircraft modelling tool. It was released as open-source software in 2012 under the NASA open source agreement. This work combines aerodynamic analysis with OpenVSP to analyse the lift and drag of the models. The next section discusses how the aerodynamic analysis was performed and the solver that was used.

3. Computational fluid dynamics

Computational Fluid Dynamics (CFD) uses numerical methods to solve how liquids and gases interact with surfaces. Although the calculations are computationally intensive, the dramatic increase in the power of standard hardware enables basic CFD analysis to be carried out on standard desktop machines. OpenFOAM (open-source field operations and manipulation) [10] is used as the CFD solver in the experiments. Although primarily used for fluid dynamics simulations, it provides a toolbox of different solving techniques for applications such as combustion, electromagnetism, solid mechanics and heat transfer. It is designed for parallel execution due to the high processor demand of CFD modelling. It is highly extensible and has been adapted for calculating transonic aerodynamics [11], marine cavitation models [12] and orthotropic solid mechanics [13].

The solver used in the experiments is the semi-implicit method for pressure linked equations (SIMPLE) algorithm [14]. It is a steady state numerical solver for efficiently solving the Navier–Stokes equations that describe fluid motion. The algorithm forms the basis of CFD software and has been adapted to calculate the transfer of mass and momentum in a discretised three dimensional environment. The solver iteratively calculates the pressure and the velocity within the system. Post-processing then calculates the lift and drag forces generated by the model and these are used as the fitness value.

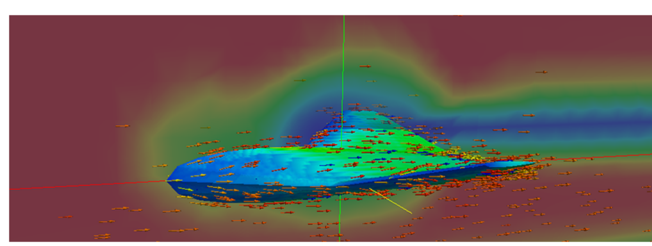


Fig. 2. The relative wind velocity and turbulence caused by the blended wing body model.

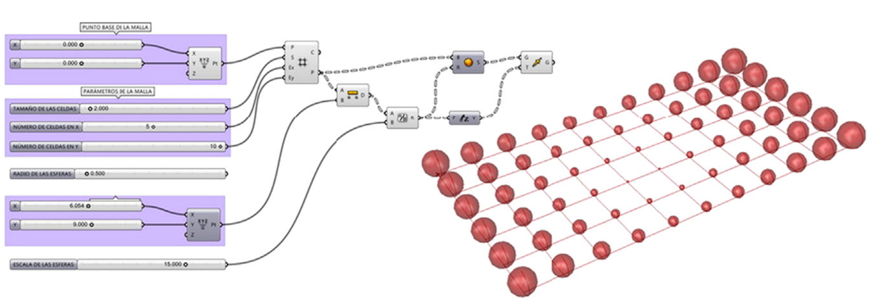


Fig. 1. The GUI for the Grasshopper parametric system. The variables are shown in the purple boxes on the left and are connected to the shape generating functions. The output design is on the right. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

4. Evolutionary aircraft optimisation

> "Since design problems defy comprehensive description and offer an inexhaustible number of solutions the design process cannot have a finite and identifiable end. The designer's job is never really done and it is probably always possible to do better." [15].

Design problems inevitably involve some trade off between desirable attributes [16]. In aircraft design there is a trade off between lift and drag which is known as aerodynamic efficiency. A design must have not only a minimal cross-sectional area to reduce drag but also a large wing to maximise lift. Conflicting objectives mean there is no one perfect solution, instead there is a pareto front of equally viable designs. Multi-objective problems are difficult to optimise but the population based approach of evolutionary algorithms has been shown to be a successful approach [17]. Multi-objective evolutionary algorithms (MOEAs) have been shown to be a useful approach for finding the best compromise when tackling a multi-objective problem [18].

Accordingly there have been several MOEA approaches to evolving aerodynamically efficient aircraft. Due to the computational expense of CFD analysis most approaches focus on 2D optimisation of airfoils [19,2,20]. Different components have been optimised individually, such as the wing [21] or the turbine blade positions [22]. Although some large-scale optimisation examples have been carried out [23,24] the difficulty in defining such a complex representation has limited its application. The next section describes the aircraft model that is the basis for optimisation and the multi-objective algorithm used to optimise the aerodynamic efficiency.

5. Experimental settings

A standard genetic algorithm (GA) was used in all the experiments. The settings used by the GA are shown in Table 1. The source code is freely available to download at [25] under the GNU public license. A context free grammar mapping [26] was used to convert the integer values of the GA representation into values for the parametric model. As the grammar was changed for different optimisation tasks, each grammar is shown in its respective section. Both lift and drag are being used as fitness values to evaluate the designs. The SIMPLE algorithm discussed in Section 3 returns the coefficients of lift (the force perpendicular to the oncoming flow direction) and drag (the force parallel to the flow direction) in Newtons for a particular design. The two values are then used by the NSGA2 algorithm to calculate the fitness value for that design.

In order to evolve designs that incorporated both of these features, the non-sorting genetic algorithm II (NSGA2) multi-objective fitness function was used for selection and replacement [18]. Multi-objective search algorithms do not assume that there is a globally optimal solution but that there is a set of non-dominated solutions. The non-dominated solutions are solutions that are better than the rest of the population for at least a single constraint and at least equivalent for all other constraints. This can be stated mathematically as $\mathbf{f}$ which is the set of fitness functions: $\mathbf{f} = [f_o, \ldots, f_n]$ such that $\forall f \in \mathbf{f}$ where $f^{\text{non-dom}} \leq f^{dom}$ and $\exists f \in \mathbf{f}$ where $f^{\text{non-dom}} < f^{dom}$.

The parent and child populations are combined and the NSGA2 algorithm selects the non-dominated solutions from the Pareto front. It then selects the least dominated solutions incrementally until the population size has been reached. The new population of non-dominated solutions is used as the parent population for the next generation. Elitism is implemented by comparing the adult and child populations and selecting the best of both for the new adult population.

In order to evaluate the performance of the evolutionary algorithm, the results were compared against randomly generated designs from the search space, essentially a brute force approach. This comparison examines if any useful genetic information is being transferred between individuals and whether the parametric representation is amenable to evolutionary search. Due to limited available computing power only two runs were carried out for each experiment. Although this does not constitute a sufficient sample size to support the efficacy of stochastic methods such as an EA, the intention of this work is to examine if the aerodynamic efficiency of a parametric model can be optimised. As such the pareto-efficiency of the individuals in the final population will be used to judge the effectiveness of the algorithm as an active design tool.

Table 1
Experimental settings.

Property	Setting
Population size	50
Generations	50
No. of Runs	2
Mutation operator	Per codon
Mutation rate	1.5%
Crossover operator	Single point
Crossover rate	70%
Selection & Replacement	NSGA2
Random number generator	Mersenne twister

6. Optimisation of blended wing body design

In traditional aircraft the fuselage provides little or no lift to the craft. Originally developed by NASA, the blended wing body (BWB) flattened the fuselage into the shape of an airfoil so that the entire craft generates lift. The BWB model has been used extensively as a test case for multidisciplinary design optimisation (MDO) [27]. MDO uses optimisation techniques to solve design problems that span multiple disciplines. A parametric model of the BWB design was used as a test case due to the simplicity of the model. It consists of a single wing component that is made up of three sections. In total the model contains 1104 facets which means that it is processed quickly in a CFD analysis. The model is shown in Fig. 3.

One of the main advantages of parametric design optimisation is that it is easy to optimise specific features of a design. In order to highlight this two separate experiments were carried out. The first experiment solely optimised the airfoils while maintaining the predefined wing shape, so as to improve the design while remaining visually the same. The second experiment varied the

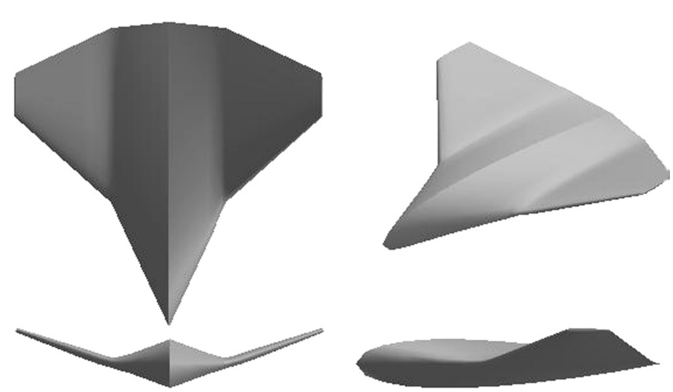


Fig. 3. The blended wing body model.

shape of the wing sections and their airfoil shape simultaneously, allowing the algorithm to alter the entire model and explore many different wing configurations.

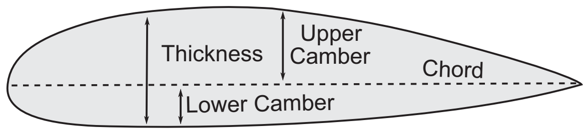


Fig. 4. NACA profile of an airfoil.

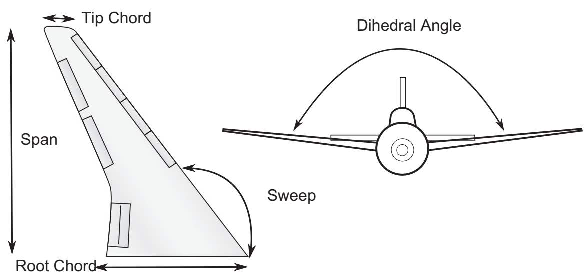


Fig. 5. The features of a wing section.

```
<aircraft> ::= <foil0> <foil1> <foil2>
<airfoil>  ::= {'Camber':<r>, 'Thickness':<r>}
<foil0>    ::= self.plane['foil0'] = <airfoil>
<foil1>    ::= self.plane['foil1'] = <airfoil>
<foil2>    ::= self.plane['foil2'] = <airfoil>
<r>        ::= 0.<digit><digit><digit><digit><digit>
<digit>    ::= 1|2|3|4|5|6|7|8|9|0
```

Fig. 6. The encoding used to describe the camber and the thickness of each airfoil on the wing.

The initial experiment only allows variation of the airfoil sections off the wing. The airfoil is defined by a National Advisory Committee for Aeronautics (NACA) profile system [28]. The NACA profile combines mean lines and thickness distribution to obtain the desired airfoil shapes. The NACA system allows the airfoil to be defined using only three parameters: thickness, camber and camber location. The wing on the BWB consists of three distinct wing sections. Only the camber and the thickness were varied while the camber location remained fixed. Fixing the camber location of the airfoils means that the overall shape and the configuration of the aircraft remain close to the original model (Fig. 4).

The second experiment increases the number of variables in the representation to include the span, sweep, tip chord, root chord and dihedral angle of the wing. These features of the wing are illustrated in Fig. 5. Although changing this many features means that the model will vary greatly from the original design, it examines if the optimiser can be used as an explorative tool. Increasing the amount of variability in the representation will generate more infeasible design but does open up the possibility of finding an improved yet unexpected configuration. A grammar was used as an interface to describe the components of the parametric model. Fig. 6 shows the grammar used for optimising the airfoil components while Fig. 7 shows the grammar for optimising the wing and airfoil components.

6.1. BWB optimisation results

A scatter plot of airfoil optimisation results is shown in Fig. 8(a). The graph shows how well the design maximised lift on the *x*-axis and how well it reduced drag on the *y*-axis. The original model is shown in black. The evolved solutions and the brute force

```
<aircraft> ::= <section0> <section1> <foil0> <foil1> <foil2>
<section>  ::= {'Span':<r>, 'TC':<r>, 'RC':<r>, 'Sweep':<r>,'Dihedral':<r>}
<airfoil>  ::= {'Camber':<r>, 'Thickness':<r>}
<section0> ::= self.plane['section0'] = <section>
<section1> ::= self.plane['section1'] = <section>
<foil0>    ::= self.plane['foil0'] = <airfoil>
<foil1>    ::= self.plane['foil1'] = <airfoil>
<foil2>    ::= self.plane['foil2'] = <airfoil>
<r>        ::= 0.<digit><digit><digit><digit><digit>
<digit>    ::= 1|2|3|4|5|6|7|8|9|0
```

Fig. 7. The encoding used to vary each section and airfoil of the wing.

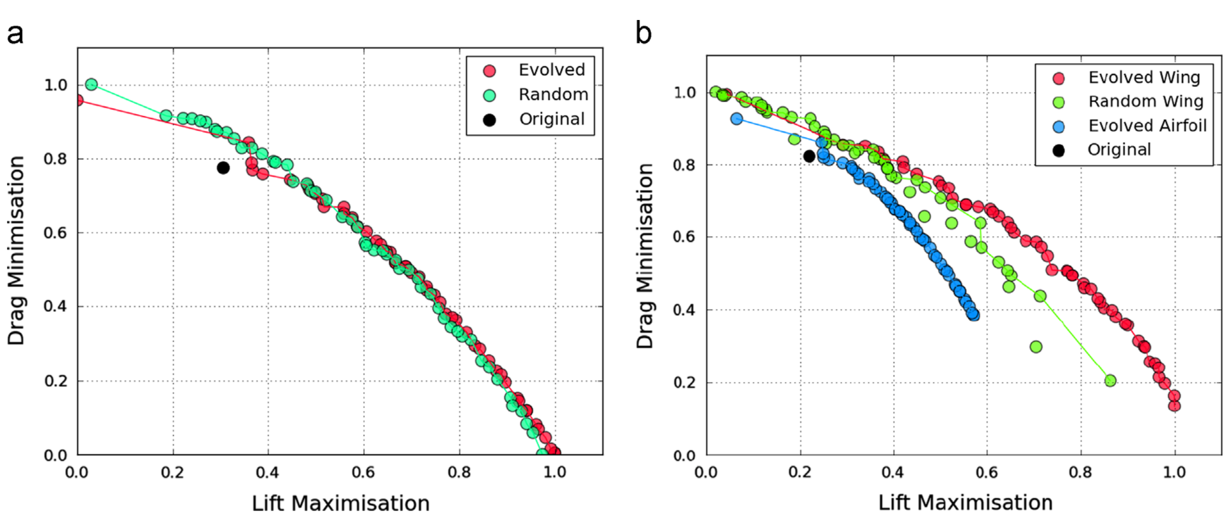


Fig. 8. The pareto front for the final generation of aircraft. The results from the airfoil optimisation are shown in blue in the wing optimisation for comparison: (a) airfoil optimisation and (b) wing optimisation. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

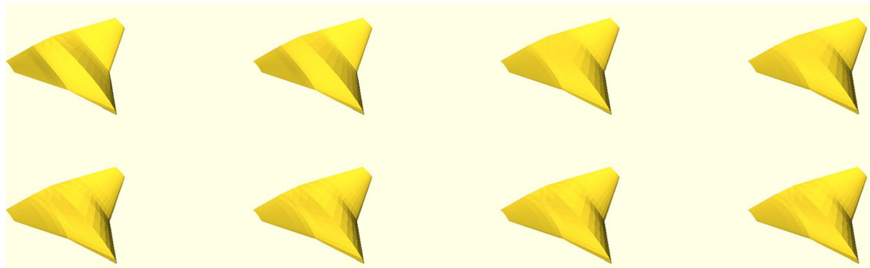


Fig. 9. Airfoil optimisation in the order of increasing lift (and increasing drag) from top left to bottom right. The overall shape of the design remains the same.

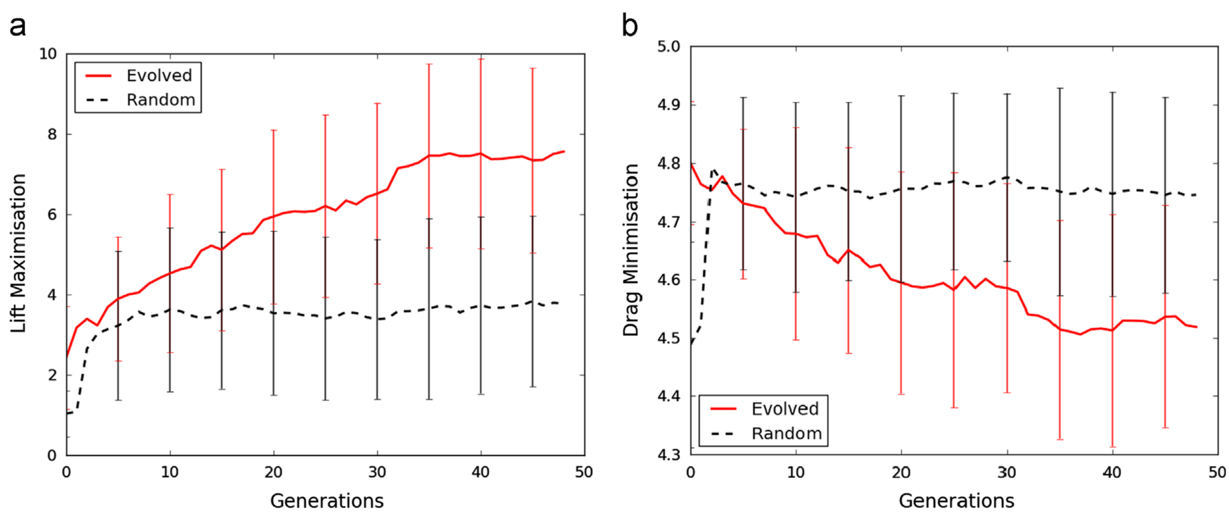


Fig. 10. The change in average lift/drag during the course of the run: (a) average lift maximisation and (b) average drag minimisation.

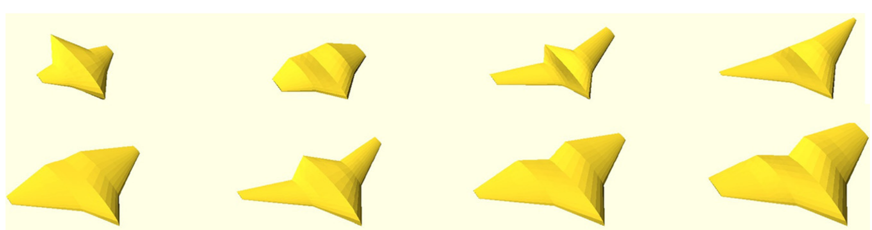


Fig. 11. Wing optimisation in the order of increasing lift (and increasing drag) from the top left to the bottom right. The increased number of variables resulted in different wing configurations.

solutions are shown in red and green respectively with a line connecting individual on the pareto front. Overall the pareto front of the evolved solutions is equivalent to the randomly generated solutions, indicating that no benefit was provided by the genetic information.

That an evolutionary approach did not outperform a brute force approach could be the result of the constrained nature of the representation. Each of the three airfoil sections had two variables. Although each individual was encoded by 30 integers, the range of each variable was limited to viable designs. Such a representation could generate good solutions purely by random variation, indicating that it is too constrained. This conclusion would be supported by the fact that both approaches generated pareto optimal designs that outperformed the original model. A sample of individuals from the pareto front is shown in Fig. 9. Limiting the evolvable representation to the airfoils produced optimised solutions that maintained the same overall design as the BWB aircraft.

A scatter plot of wing and airfoil optimisation is shown in Fig. 8(b). Again the original model is shown in black and the

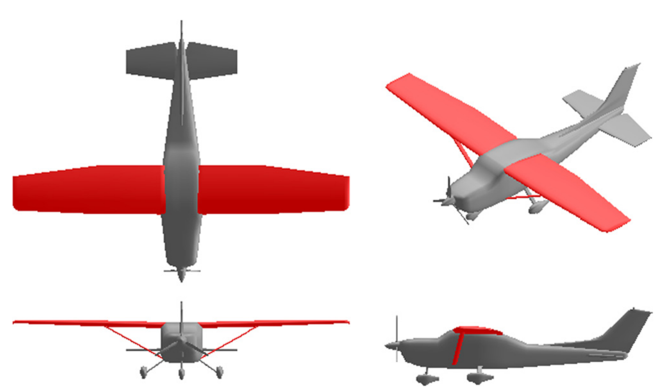


Fig. 12. The Cessna 182 model. The optimised sections are highlighted in red. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

evolved and brute force solutions are shown in red and green respectively. The graph shows how well the design maximised lift on the x-axis and how well it reduced drag on the y-axis.

```
<aircraft> ::= <sect0> <sect1> <foil0> <foil1>
<section>  ::= {'Span':<r>, 'TC':<r>, 'RC':<r>, 'Sweep':<r>}
<airfoil>  ::= {'Camber':<r>, 'Thickness':<r>}
<sect0>    ::= self.plane['section2'] = <section>
<sect1>    ::= self.plane['section3'] = <section>
<foil0>    ::= self.plane['foil2'] = <airfoil>
<foil1>    ::= self.plane['foil3'] = <airfoil>
<r>        ::= 0.<digit><digit><digit><digit><digit>
<digit>    ::= 1|2|3|4|5|6|7|8|9|0
```

Fig. 13. The encoding used for the Cessna 182. The dihedral angle was not altered to maintain the overall theme of the design.

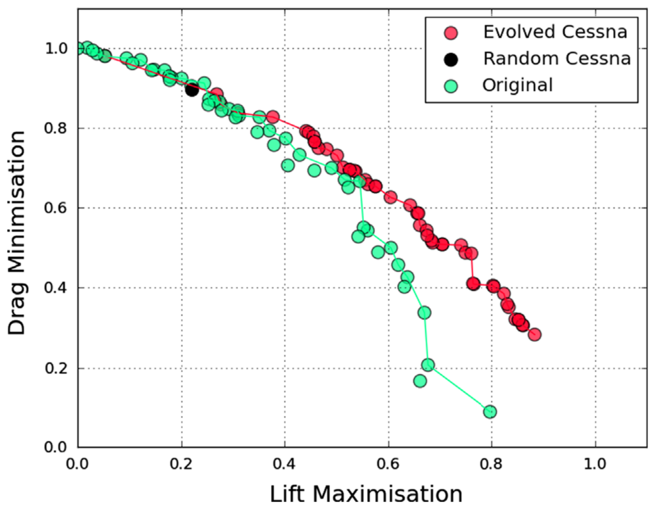


Fig. 14. The respective pareto fronts of the evolved and randomly selected designs. The original Cessna 182 model is shown in black.

The increased variability of the representation greatly increased the range of the Pareto fronts when compared to the airfoil optimisation results, shown in blue.

The evolved pareto front is distinct from the brute force approach. The randomly generated individuals tend to cluster around minimal drag designs as it is easy to find a design with a smaller wing, all the algorithm has to do is reduce the size of the aircraft. It is more difficult to find a design with an aerodynamically viable wing and this is where the evolutionary algorithm excels.

This result is highlighted by examining the average population fitness during the course of a run, as shown in Fig. 10. The NSGA2 selection operator compares child and adult populations and takes the best of both to create a new adult population. This requires two populations to be generated before evaluation can take place and so the graphs start at the second generation. The evolutionary algorithm is already populated with high-fitness designs at this point while the selection pressure quickly builds up the elite population of the brute force approach, thus improving the average fitness. In both drag and lift graphs the brute force approach plateaus after five generations. The evolutionary approach on the other hand continues to improve lift (while sacrificing drag efficiency) for the duration of the run as shown in Fig. 10(a) and (b). Fig. 11 shows a sample of optimised designs from the pareto front.

The relaxing of the evolvable representation resulted in many different wing configurations being generated. The amount of variation shows that such design problems are highly open-ended with no single optimal design configuration. It also suggests that allowing the algorithm to evolve more components of the representation could result in novel yet highly efficient designs.

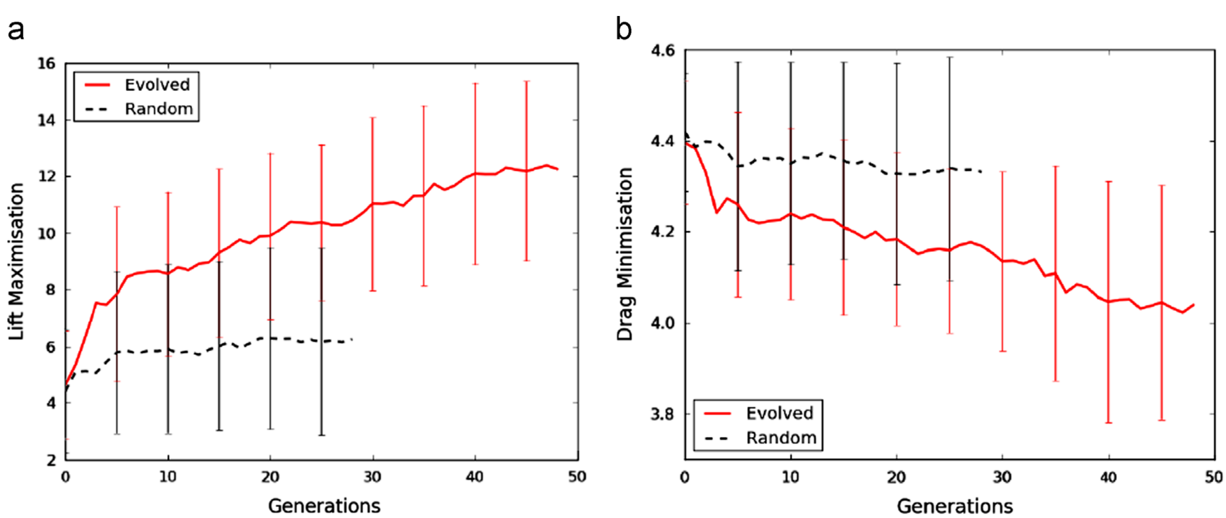


Fig. 15. The change in average lift/drag during the course of the run: (a) average lift maximisation and (b) average drag minimisation.

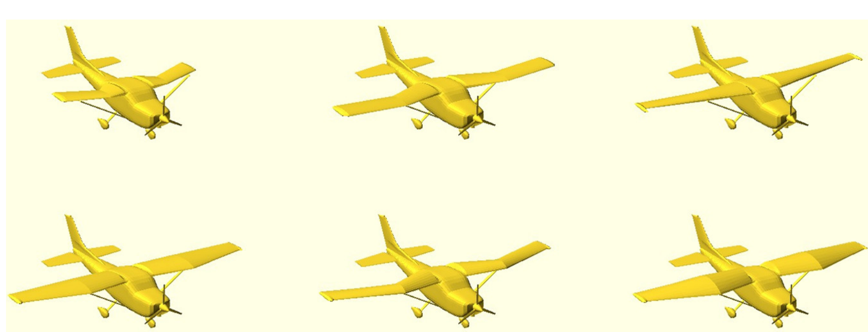


Fig. 16. A sample of the optimised Cessna 182 designs from the pareto front.

7. Optimisation of the Cessna 182 wing

This section demonstrates the selective optimisation possible with the parametric representation. Only the wing structure of a Cessna 182 aircraft is optimised. The section and the airfoil of the wing are varied while the fuselage, propeller, tail section and undercarriage remain fixed. The Cessna 182 is the second most popular Cessna variant in production. The model is more complex than the BWB design as it is composed of 13,476 facets. Although the increased complexity affects the amount of time taken to analyse the model, the parametric model has a similar number of components to the BWB representation. The wing component is defined as two separate sections, each of which has its own distinct airfoil. The grammar in Fig. 13 describes the representation of the Cessna 182 wing.

An additional advantage of a parametric representation is that a component may be analysed in conjunction with the total structure. A single component cannot be analysed in isolation. For example, a wing optimised separately from the aircraft could perform differently when fixed to the aircraft. It may cause eddies or turbulence on other surfaces directly behind it, such as the fuselage or tail section. Optimising a component as part of a total structure generates a more realistic analysis. The optimised area of the Cessna 182 is shown in red in Fig. 12.

7.1. Cessna 182 wing optimisation results

The scatter plot of the Cessna wing optimisation results is shown in Fig. 14. The graph shows how well the design maximised lift on the x-axis and how well it reduced drag on the y-axis. The original model is shown in black. The results for the average objective fitness during the run are shown in Fig. 15. As the amount of variation allowed for the overall design is less, both approaches start with similarly performing aircraft designs. The evolutionary approach again increases lift performance during the course of the run while sacrificing drag minimisation.

The brute force approach generates little improvement in either drag or lift during the course of the run. There is significant overlap of the drag results for both evolutionary and brute force approaches in Fig. 15(b), indicating that the difference is statistically insignificant, although more runs will have to be carried out before this can be conclusively shown (Fig. 16).

8. Optimisation of the MIG 21 wing and tail sections

As an extension of the previous experiment multiple surfaces of the MIG 21 model are optimised simultaneously. The MIG 21 model was chosen as it is composed of 26,600 facets, highlighting the complexity of aircraft models it is possible to optimise. Different components of a design cannot be optimised individually and be expected to perform similarly when combined. The wing and the tail section of the MIG 21, as shown in red in Fig. 17, are varied in this experiment. One additional limitation is placed on the model. As the vertical stabiliser does not provide any lift an optimiser might remove this structure altogether. The variable ranges of the vertical stabiliser are reduced to prevent this happening. The grammar describing the changes to the MIG 21 model is shown in Fig. 18.

8.1. MIG 21 wing and tail section optimisation results

The scatter plot of the MIG 21 optimisation results is shown in Fig. 21. The graph shows how well the design maximised lift on the x-axis and how well it reduced drag on the y-axis. The original model is shown in black (Fig. 19). Once again both brute force and evolutionary approaches generate design that outperform the original design. There is an overlap of the pareto fronts for drag minimisation designs but the evolutionary approach generates aircraft with better lift optimisation.

Fig. 20(a) and (b) shows more clearly what is happening. Similar to the Cessna experiments, both approaches start with

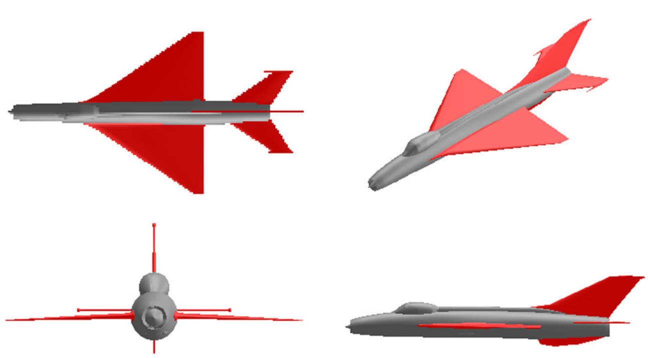


Fig. 17. The MIG 21 model. The optimised sections are highlighted in red. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

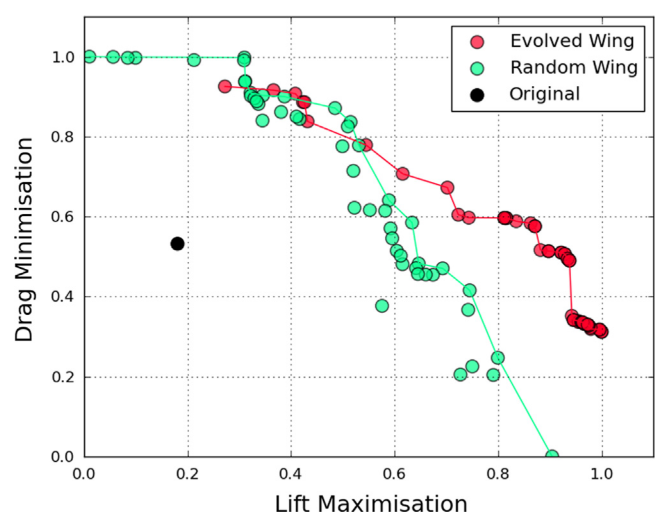


Fig. 19. The respective pareto fronts of the evolved and randomly selected designs. The original MIG 21 model is shown in black.

```
<aircraft> ::= <wing> <vert> <horz>
<wingsect> ::= {’Span’:<r>, ’TC’:<r>, ’RC’:<r>, ’Sweep’:<r>, ’Dihedral’:<r>}
<horzsect> ::= {’Span’:<r>, ’Sweep’:<r>,’Dihedral’:<r>}
<vertsect> ::= {’Span’:<r>, ’Sweep’:<r>}
<wing>     ::= self.plane[’wingsection2’] = <wingsect>
<horz>     ::= self.plane[’horzsection1’] = <horzsect>
<vert>     ::= self.plane[’vertsection1’] = <vertsect>
<r>        ::= 0.<digit><digit><digit><digit><digit>
<digit>    ::= 1|2|3|4|5|6|7|8|9|0
```

Fig. 18. The encoding used for the MIG 21. Three different components of the design were evolved simultaneously.

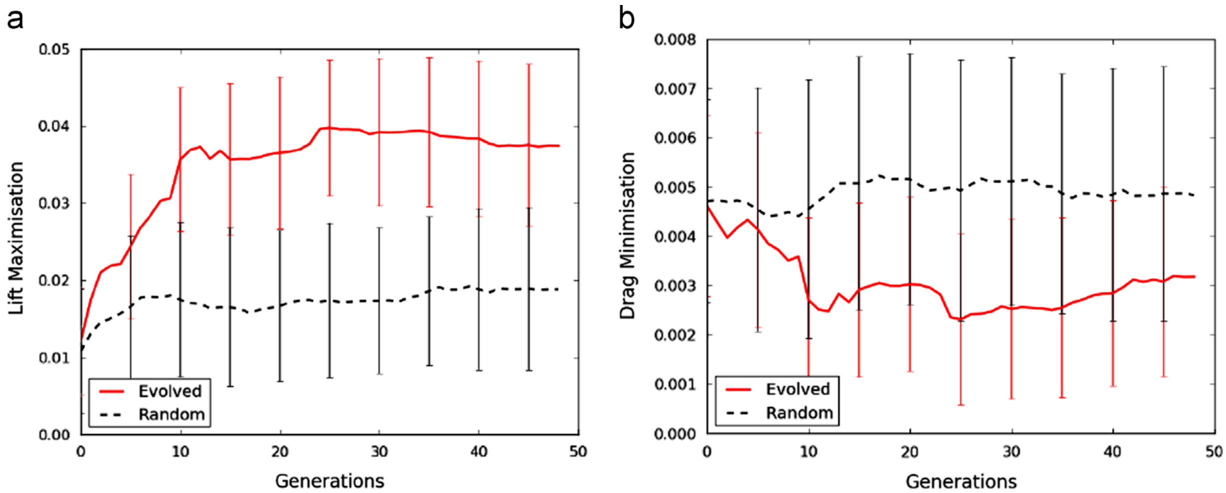


Fig. 20. The change in average lift/drag during the course of the run: (a) average lift maximisation and (b) average drag minimisation.

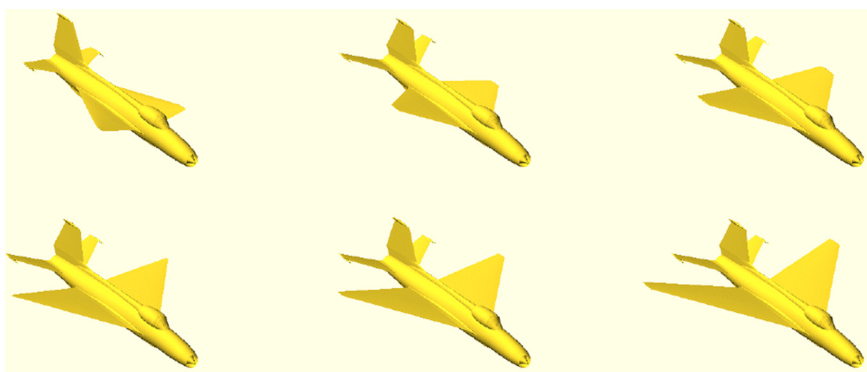


Fig. 21. Selected individuals from the pareto front of the MIG 21 optimisation results in the order of increasing lift (and increasing drag) from top left to bottom right.

comparable performance. This is due to only parts of the design being optimised. Overall the brute force approach only generates some limited improvement before the results plateau. The evolutionary approach, on the other hand, generates a greater lift improvement in its designs while sacrificing drag performance.

9. Discussion

The results from the experiments, with the exception of the BWB airfoil optimisation results, indicate that an evolutionary approach generates more aerodynamically efficient aircraft than a brute force approach. Although more runs will have to be conducted before this can be conclusively shown, it is a promising result. One unexpected result was that a brute force approach still produced designs that surpassed the original design. Normally a random approach generates poor optimisation results but as the parametric representation constrains the amount and type of the variation, even randomly selected designs were still airworthy and found a niche on the pareto front.

10. Conclusions

A parametric system allows the designer, not the programmer, to specify the design to be evolved. Three different aircraft modelled using the OpenVSP design tool were optimised. The experiments showed that the level of design optimisation could be varied. Specific components of the model can be optimised or the model can be used as the basis for generating entirely different aircraft configurations. Although the sample size of the experiment is too small to draw any significant conclusions, initial results indicate that the parametric representation is capable of being optimised by an evolutionary algorithm. Even in experiments where brute force approaches performed comparably to evolutionary approaches, both generated designs that outperformed the original parametric model. This approach could potentially be applied to any existing parametric design to generate optimised solutions, turning the computer into an active design tool in the conceptual design process.

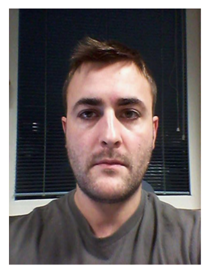
Acknowledgements

We would like to thank Science Foundation Ireland, the Financial Mathematics Computation Cluster and Andrea McMahon for her help during this project. We also wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. This work was funded by the SFI Grants 08/RFP/CMS1115, 08/IN.1/I1868 and 08/SRC/FM1389.

References

[1] Rafal Kicinger, Tomasz Arciszewski, Kenneth De Jong, Evolutionary computation and structural design: a survey of the state-of-the-art, Comput. Struct. 83 (23–24) (2005) 1943–1978, http://dx.doi.org/10.1016/j.compstruc.2005.03.002, ISSN 0045-7949.

[2] Emre Alpman, Airfoil Shape Optimization Using Evolutionary Algorithms, Aerospace Engineering Department, Pennstate University, 2004.
[3] Martyn Day, Grasshopper, Generative Modelling. 〈http://www.grasshopper3d.com/〉, 2010.
[4] Bentley sytems, Generative Components, v8i. 〈http://www.bentley.com/getgc/〉, 2011.
[5] D. Holzer, R. Hough, M. Burry, Parametric design and structural optimisation for early design exploration, Int. J. Archit. Comput. 5 (4) (2007) 625–643.
[6] K. Shea, R. Aish, M. Gourtovaia, Towards integrated performance-driven generative design tools, Autom. Constr. 14 (2) (2005) 253–264, ISSN 0926-5805.
[7] K. Bollinger, M. Grohmann, O. Tessman, Form, force, performance: multi-parametric structural design, Archit. Des. 78 (2) (2008) 20–25.
[8] Oasys Software. GSA, Structural Analysis Version 8.5, 〈http://www.oasys-software.com/gsa-analysis.html〉, 2011.
[9] James R. Gloudemans, Paul C. Davis, Paul A. Gelhausen, A rapid geometry modeler for conceptual aircraft, in: The 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, January, 1996, pp. 15–18.
[10] Henry G. Weller, G. Tabor, Hrvoje Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (1998) 620.
[11] Benjamin Wüthrich, Y.J. Lee, Simulation and validation of compressible flow in nozzle geometries and validation of OpenFOAM for this application (Ph.D. thesis), ETH, Swiss Federal Institute of Technology Zurich, Institute of Fluid Dynamics, 2007.
[12] Rickard E. Bensow, Goran Bark, Simulating cavitating flows with LES in openfoam, in: The Fifth European Conference on Computational Fluid Dynamics, 2010, pp. 14–17.
[13] P. Cardiff, A. Karač, A. Ivanković, A large strain finite volume method for orthotropic bodies with general material orientations, Comput. Methods Appl. Mech. Eng. (2013).
[14] S.V. Patankar, D. B Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (10) (1972) 1787–1806, ISSN 0017-9310. http://dx.doi.org/10.1016/0017-9310(72)90054-3. URL 〈http://www.sciencedirect.com/science/article/pii/0017931072900543〉.
[15] B. Lawson, How Designers Think: The Design Process Demystified, Elsevier/Architectural, 2006, ISBN 9780750660778. URL 〈http://books.google.ie/books?id=lPvqZJNAdG8C〉.
[16] H.A. Simon, The Sciences of the Artificial, MIT Press, 1996.
[17] Eckart Zitzler, Lothar Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.
[18] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, ISSN 1089-778X.
[19] Domenico Quagliarella, Domenic D'ambrosio, Angelo Iollo, Airfoil Design Using Navier-Stokes Equations and Hybrid Evolutionary Optimization Techniques, Technical Report, DTIC Document, 2003.
[20] Boris Naujoks, Lars Willmes, Werner Haase, Thomas Bäck, Martin Schütz, Multi-point airfoil optimization using evolution strategies, in: Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS00) (CD-Rom and Book of Abstracts), 2000, p. 948.
[21] Shigeru Obayashi, Multidisciplinary design optimization of aircraft wing planform based on evolutionary algorithms, in: The 1998 IEEE International Conference on Systems, Man, and Cybernetics, 1998, IEEE, vol. 4, 1998, pp. 3148–3153.
[22] T. Rogalsky, R.W. Derksen, S. Kocabiyik, An aerodynamic design technique for optimizing fan blade spacing, in: Proceedings of the Seventh Annual Conference of the Computational Fluid Dynamics Society of Canada, Citeseer, 1999, pp. 2–29.
[23] George S. Dulikravich, Aerodynamic shape design and optimization-status and trends, J. Aircraft 29 (6) (1992) 1020–1026.
[24] Ian C. Parmee, Andrew H. Watson, Preliminary airframe design using co-evolutionary multiobjective genetic algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, 1999, pp. 1657–1665.
[25] Jonathan Byrne, Plane Generator Source Code, 〈https://github.com/squeakus/planegen/〉, 2013.
[26] James McDermott, Jonathan Byrne, John Mark Swafford, Martin Hemberg, Ciaran McNally, Elizabeth Shotton, Erik Hemberg, Michael Fenton, Michael O'Neill, String-rewriting grammars for evolutionary architectural design, Environ. Plan. B: Plan. Des. 39 (4) (2012) 713–731, URL 〈http://www.envplan.com/abstract.cgi?id=b38037〉.
[27] Yan-Yee Andy Ko, The multidisciplinary design optimization of a distributed propulsion blended-wing-body aircraft (Ph.D. thesis), Virginia Polytechnic Institute and State University, 2003.
[28] Eastman N. Jacobs, Kenneth E. Ward, Robert M. Pinkerton, The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel, Technical Report, DTIC Document, 1933.



Jonathan Byrne is a research scientist working with the Urban Modelling Group in UCD. His research focuses on design optimisation, structural wind modelling and 3D printing.

Dr Philip Cardiff is a post-doctoral research fellow with the mechanical engineering department in UCD. His research focuses on finite volume methodologies for numerical analysis.

Professor Anthony Brabazon is currently the Associate Dean of the Smurfit School of Business and head of the Financial Mathematics Computation Cluster.

Professor Michael O'Neill is currently the director of the Complex and Adaptive Systems Laboratory and head of the Natural Computing Research and Applications Group.