DOI: 10.13009/EUCASS2022-6041
9TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Robustness of optimisation algorithms for transonic aerodynamic design

Francisco Sánchez-Moreno[1], David MacManus[1], Fernando Tejero[1], Jesús Matesanz-García[1], Christopher Sheaf[2]*

[1] Centre of Propulsion and Thermal Power Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire, United Kingdom, MK43 0AL

[2] Rolls-Royce plc., P.O. box 31, Derby, United Kingdom, DE24 8BJ

[*]Corresponding author: fran.sanchez-moreno@cranfield.ac.uk

Abstract

In design optimisation problems, it is essential to ensure the convergence to the optimal design space with the lowest variability possible. In this respect, the optimisation algorithm plays a key role as it drives the exploration of the design space. This paper presents a statistical assessment of two genetic algorithms (NSGA-II and IBEA) and a particle swarm optimiser (OMOPSO) for the transonic aerodynamic design of compact nacelles for future aero-engines. OMOPSO is the most suitable optimisation algorithm due to the lowest variability to find an optimised design with a reasonable convergence rate of the optimisation.

1. Introduction

Aerodynamic shape optimisation is a complex problem which is usually dominated by different operating point and multi-disciplinary constraints [1]. Depending on the problem, a global optimal solution may not be found and trade-offs between the different optimisation objective functions are typically required. Depending on the nature of the problem, different optimisation methods can be more suitable and they are generally divided into: (a) gradient-based methods (GBM) and (b) gradient-free methods (GFM) [2]. GBM are typically more efficient for very high dimensional optimisation problems but they are also more prone to get trapped in local optima [3]. Aerodynamic shape optimisation problems are typically governed by non-linear flow physics which makes the problem non-convex. In this respect, GFM may be a good method to increase the likelihood of finding the global optima when the computational overhead associated to the dimensionality of the problem is acceptable [4].

Within GFM, genetic algorithms (GA) and particle swarm optimizers (PSO) have extensively been used for non-linear aerodynamic optimisations [5–8]. Both methods are classified as evolutionary algorithms (EA) as they use mechanisms inspired by nature that emulate the behaviour of living organisms [9]. While GA is based on the Darwinian theory of survival of the fittest the PSO method comes from the swarming motion of collective animal behaviour [2]. GA and PSO are similar in the sense that both methods follow a population-based searching approach which require the sharing of information among the population members through a set of deterministic and probabilistic rules. However, there are differences that make each algorithm more adequate for a specific optimisation problem. Comparative assessments of GA and PSO algorithms have been reported in the literature [10–13]. The emphasis of these studies was on the relative performance between the optimal individual obtained with each algorithm and on the convergence rate. These studies agree on the computational efficiency superiority of PSO methods over GA. This computational effort saving is problem dependant and is reduced as the optimisation problem is more constrained [13]. Within the aerodynamic design area, Lyu et al. [14] applied a GA and a PSO algorithms to the non-linear aerodynamic optimisation of the NASA common research model wing at cruise condition. The method considered a computational fluid dynamics (CFD) model based on the Reynolds-Average Navier-Stokes (RANS) equations and 8 design variables to parameterise the geometry. The PSO algorithm found a slightly better design with a reduction of 0.8 aircraft drag counts relative to the GA optimal design for an increase in the convergence rate of PSO compared to GA of about two.

DOI: 10.13009/EUCASS2022-6041

Additionally, Mukesh et al. [15] compared a GA and a PSO method to maximise the lift coefficient (C_L) of the NACA 2411 airfoil at cruise condition. The optimisation problem considered 12 design variables and used inviscid computational flow solutions. While the GA reported an increase in C_L from 0.842 to 0.968, the PSO-based optimisation achieved a C_L = 1.035.

For challenging design spaces, the optimisation process may fall in premature convergence towards a local optima. This is undesirable and a waste of resources as the optimisation would yield a sub-optimal design. Randomness is introduced in the optimisation process to mitigate premature convergence and maintain genetic diversity in the searching of the design space. In an optimisation driven by an EA the randomness is present both in the initialization of the optimisation through a Design of Experiments (DoE) in the first generation and also in the mutation and crossover operators used to generate an offspring population between generations. To focus on the performance of an algorithm, the second source of randomness is statistically assessed by fixing the samples distribution in the first generation. There is a lack of information in the open literature about the uncertainty carried in the optimisation due to the searching performance of the algorithm for transonic non-linear aerodynamic problems. This work presents a statistical comparison between different GA and PSO algorithms for: (a) the identified optimal design spaces, (b) the variability in the optimal design space reached by each optimiser through repeated optimisations, and (c) the convergence rate. The transonic aerodynamic optimisation of a compact nacelle for the next generation of ultra-high bypass ratio (UHBR) aero-engines [16,17,18] is used as a test case for the analysis.

2. Methodology

2.1 Geometry definition

The nacelle geometry definition is parameterised through the Class Shape Transformation (CST) model developed by Kulfman [19]. This method has extensively been used for aerodynamic shape parameterisation for airfoils and wings [20–22]. A CST curve is built through the product of a class function $C(\psi)$ and a shape function $S(\psi)$ plus a vertical offset between the end points (Eq. 1), where ψ and ξ are the non-dimensional abscissa and ordinate and c is the curve length. The class function is selected based on the underlying geometry to represent. For the parameterisation of a nacelle a rounded leading edge and sharp trailing edge profile is selected (Eq. 2). The shape function contains the Bernstein polynomial coefficients bp_i of n+1 polynomials and are used to adjust the class function to the aimed geometry (Eq. 3).

$$\xi(\psi) = C(\psi)S(\psi) + \psi\Delta\xi_{TE}\ ;\ \psi = \frac{x}{c}\ ,\xi = \frac{y}{c} \tag{1}$$

$$C_{1.0}^{0.5}(\psi) = \psi^{0.5}[1-\psi]^{1.0} \tag{2}$$

$$S(\psi) = \sum_{i=0}^{n} bp_i \binom{n}{i} \psi^i (1-\psi)^{n-1} \tag{3}$$

Christie et al [23] extended this method to the intuitive CST (iCST) approach and applied it for the intake and fan cowl parameterisation [24]. This allows a more straightforward relation between the design parameters and the Bernstein polynomial coefficients which makes the geometry parameterisation problem more intuitive and automatic. Seven design variables are used to describe the fan cowl: r_{hi}, r_{te}, l_{nac}, r_{if}, r_{max}, f_{max}, β_{nac} (Figure 1). Within an optimisation process, the nacelle length and end points are typically fixed. This work is focused on a compact nacelle configuration given by l_{nac}/r_{hi} = 3.1 and r_{te}/r_{hi} = 0.95 with a fan size representative of an UHBR aero-engine. The remaining four parameters are varied during the optimisation: r_{if}, r_{max}, f_{max} and β_{nac} (Figure 1). A detailed description on the nacelle parameterisation was provided by Tejero et al. [25].

As the focus is on the optimisation of the nacelle fan cowl, a representative intake that has been designed to alleviate adverse flow features in the investigated design space has been used [24]. Additionally, a surrogate conical exhaust with a single stream efflux has been considered to produce representative flow features and momentum force term in the exit stream-tube of the exhaust [25,26] (Figure 1). This reduces the computational costs associated with the grid generation and the CFD calculation within the optimisation approach.

DOI: 10.13009/EUCASS2022-6041

2.2 Computational method

The optimisation method used Computational Fluid Dynamics (CFD) for the flow field data generation. A viscous numerical model based on the Reynolds-Average Navier-Stokes (RANS) equations is used to compute the nacelle flow field [27]. Turbulent closure is achieved through the k-ω SST model [28]. Sutherland's law [29] is used to calculate the ideal gas dynamic viscosity. A second order upwind spatial discretization scheme is used with the Green-Gauss node-based gradient interpolation method. Numerical convergence is achieved in the calculation when two conditions are reached: (a) a reduction of five order of magnitude in every residual and (b) an oscillation in the fan cowl force lower than 0.05% in the last 500 iterations.

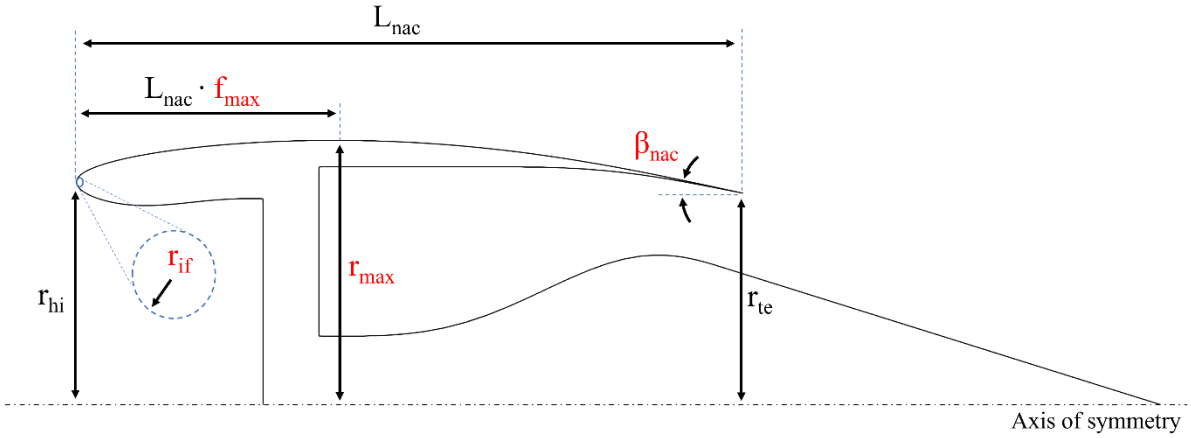


Figure 1: 2D axisymmetric nacelle geometry parameterisation. Parameters marked in red are floated in the optimisation process.

The 2D axisymmetric nacelle is located in a semi-circular domain. A domain independence study conducted by Tejero et al. [16] determined a radius of 80 times the nacelle maximum radius r_{max}. A structured mesh strategy with a C-grid refinement for the boundary layer is applied. The first layer size is adjusted for a y^+ lower than one to solve the boundary layer. A grid independence study based on Roache's method [30] determined a GCI of 0.3% in nacelle cruise drag for a mesh with an overall cell count of 70k elements. The farfield is modelled with a pressure-farfield boundary condition where the freestream conditions are set. The fan face is modelled with a pressure-outlet boundary condition with a target mass flow. A pressure-inlet boundary condition is used in the simplified exhaust. In order to minimise the impact of the exhaust flow on the nacelle drag performance, the exhaust total pressure and temperature match the freestream conditions. Non-slip wall boundary conditions are used in the intake and fan cowl to model the shock-boundary layer interaction phenomena and to account for shear stress term of the nacelle drag. The method was validated by Robinson et al. [31] with available experimental data for 2D axisymmetric nacelles that covered a range of transonic Mach numbers from 0.8 to 0.89 and mass flow capture ratio (MFCR) from 0.45 to 0.7. At the cruise condition defined by M = 0.85 and MFCR = 0.7 the difference in nacelle drag between experimental measurements and the CFD calculations was within 3.5%. In terms of drag rise Mach number, a difference of 0.005 was quantified between experiments and CFD.

2.3 Drag calculation method

The nacelle drag is the key metric to minimise in the nacelle optimisation process. A nearfield method based on industrial standard [32] has been used to assess the nacelle drag. In the thrust-drag bookkeeping method the forces are considered positive in the downstream direction. The forces external to the streamtube define the drag domain and are expressed by ϕ while the internal forces correspond to the thrust domain and are represented by θ (Figure 2). The forces across the upstream and downstream boundaries (F_G) are computed through the integration of the pressure and momentum terms in the boundary of interest. The forces exerted on the walls are calculated by integrating the pressure and viscous terms on the corresponding boundaries. Based on this accounting method, the nacelle drag is composed of not only the viscous and pressure terms on the fan cowl (ϕ_{nac}) but also the pressure term in the drag direction of the pre-entry (ϕ_{pre}) and post-exit (ϕ_{post}) flow streamtube (Eq. 4) (Figure 2). The main challenge of the method lies with the calculation of ϕ_{pre} and ϕ_{nac} as it requires the identification of the stagnation point which changes with the operating condition and the nacelle geometry. To address this problem, Christie et al. [33] developed the modified nearfield method. This approach calculates ϕ_{pre} and ϕ_{nac} as a combined term based on the balance of forces in the

DOI: 10.13009/EUCASS2022-6041

pre-entry control volume. ϕ_{post} is directly computed from the integration of pressure in the downstream direction of the post-exit streamtube. A description of the method was provided by Christie et al. [33].

$$D_{nac} = \phi_{pre} + \phi_{nac} + \phi_{post} \tag{4}$$

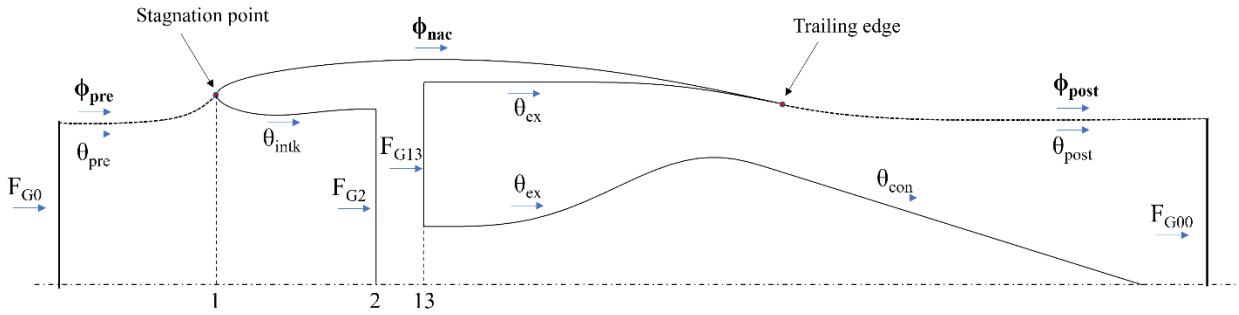


Figure 2: Thrust and drag bookkeeping method for the nacelle drag evaluation.

2.4 Optimisation approach

Compact nacelle designs increase the surface curvature gradients and make the designs more sensitive to important off-design operating conditions [26]. Therefore, several operating conditions must be considered during the nacelle optimisation process to achieve aerodynamic robust designs. This justifies the need for a multi-point optimisation method. In this work, in addition to the cruise condition given by M = 0.85 and MFCR = 0.7 (Eq. 5) the sensitivities to an increase in the flight Mach number from 0.85 to 0.87 (Eq. 6) and a decrease in the aero-engine MFCR from 0.7 to 0.65 (Eq. 7) are also considered. The minimisation of the nacelle drag at these conditions defines the three objective functions of the optimisation approach.

$$C_{D-cruise} = \frac{D_{nac}|_{M=0.85,MFCR=0.7}}{\frac{1}{2}\rho_\infty V_\infty^2 A_{hi}} \tag{5}$$

$$C_{D-\Delta M} = \frac{D_{nac}|_{M=0.87,MFCR=0.7}}{\frac{1}{2}\rho_\infty V_\infty^2 A_{hi}} \tag{6}$$

$$C_{D-EoC} = \frac{D_{nac}|_{M=0.85,MFCR=0.65}}{\frac{1}{2}\rho_\infty V_\infty^2 A_{hi}} \tag{7}$$

The optimisation provides a trade-off between the different objective functions in the format of a Pareto front. A Pareto front is a n-dimensional representation of the non-dominated designs in the space of the n objective functions. The concept of non-domination between two designs A and B stands while no design is better than the other for every objective function. From the Pareto front a traded design can be down selected based on a weighted criterion of the different objective functions. In the present work, the design with lowest cruise drag from all of those designs with a ratio $C_{D-\Delta M}/C_{D-cruise}$ and $C_{D-EoC}/C_{D-cruise}$ lower than prescribed factors (K_1, K_2) depending on the application (Eq. 8).

$$\min C_{D-cruise} \ \ subjected\ to: \begin{bmatrix} C_{D-\Delta M} < K_1\ C_{D-cruise} \\ C_{D-EoC} < K_2\ C_{D-cruise} \end{bmatrix} \tag{8}$$

A CFD-in-the-loop optimisation approach is applied in this work (Figure 3). As such, every design assessed throughout the optimisation process is evaluated with CFD at every operating condition. The optimisation starts with a DoE based on the Latin Hypercube Sampling (LHS) technique to provide an initial seed for the optimisation. The population level of the design space in this first generation of the optimisation can have a substantial impact on the scatter in the Pareto and on the convergence rate of the optimisation. Based on the work of Robinson et al. [34] for nacelle optimisations, a ratio of number of samples in the DoE to the number of design variables (N_{DoE}/N_{DOFs}) of 100 has been used. Subsequent generations are assessed until the optimisation reaches convergence. Based on previous experience [34], a sample size N_S to number of degrees of freedom (N_S/N_{DOFs}) of 12.5 has been used for each of these generations. The

DOI: 10.13009/EUCASS2022-6041

optimisation convergence is evaluated through the Hypervolume index (HV). This metric quantifies the n-dimensional space that is covered by a solution set after a generation relative to an arbitrary point used as a global reference [34]. The convergence criterion is specified as a percentage change in the HV lower than 1% in the last 3 generations which is a common practice for this optimisation problem [16]. Throughout the optimisation, geometric constraints are applied to avoid inflexion on the geometry of the nacelle. Flow constraints on the peak Mach number are also applied to ensure aerodynamic robustness of the designs within the optimal Pareto front. The optimisation is led by an evolutionary algorithm which proposes a new set of possible optimal designs after every generation.

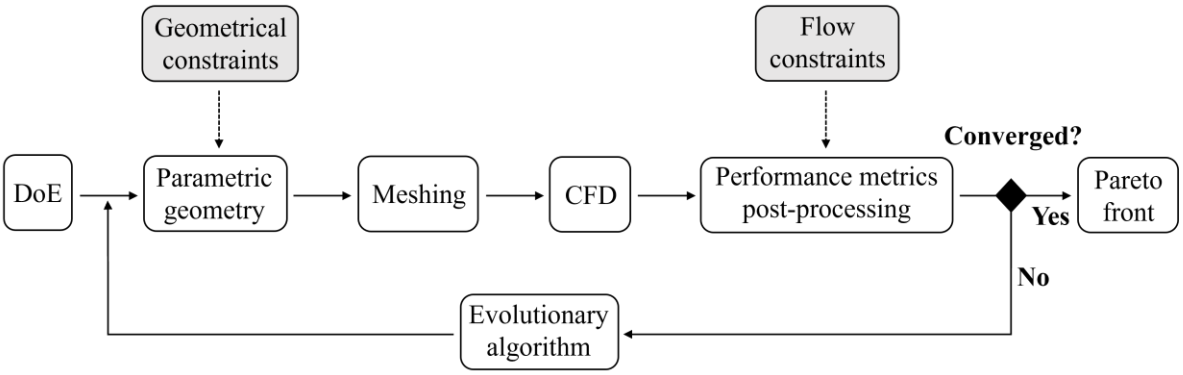


Figure 3: CFD-in-the-loop optimisation approach.

In the present study two GA and one PSO algorithms are compared in terms of the robustness of the convergence to the optimal design space. An assessment of the convergence rate for this particular application is also carried out. Based on the best performing algorithms previously found for aerodynamic nacelle optimisations by Cheval [35], the non-dominated sorting genetic algorithm II (NSGA-II) [36] and the indicator-based evolutionary algorithm (IBEA) [37] are selected as GA and the Reyes-Coello multi-objective particle swarm optimiser (OMOPSO) is the PSO optimiser considered [38]. The set-up of each optimiser used in this work has been verified against a set of optimisation benchmark problems. Zitzler-Deb-Thiele's test function 1 and test function 2 were used for this purpose [39].

2.4.1 NSGA-II

NSGA-II [36] is perhaps the most popular GA for multi-objective optimisations due to an improvement on the spread of the solution and on the convergence near the optimal Pareto front relative to other GAs. This optimiser technique is based on elite-preservation and diversity-preservation. For this purpose, the individuals of a population are sorted based on their elitism rank and on the crowding distance between each other. While the elitism rank keeps the best performing individuals the crowding distance sorting method helps to spread the population on the design space to mitigate local optima. Individuals are selected based on a tournament selection to generate an offspring population. Out of the parent population, two individuals are randomly chosen and compared based on the elitism rank. The individual with highest rank is used to reproduce. In case of both individuals having the same rank, the crowding distance factor is used to select Parent 1. This is repeated to find Parent 2. Subsequently, crossover and mutation techniques are applied to generate an offspring individual. The crossover operator randomly chooses certain genes from each parent and combines them (Figure 4). The mutation operator applies a random change to a gene that is randomly selected out of all of genes (Figure 4). This is repeated iteratively in each generation until having an offspring population of the same size as the parent population. In this work a Gaussian mutation operator defined by a standard deviation size of 0.005 and a mutation rate of 0.2 are used [16,34]. A blended (BLX) crossover technique is used to enlarge the range of variability of a certain gene further from the interval defined by both parents (M) which helps to maintain a genetic diversity in the offspring (Figure 5) [40]. For this work, $\alpha = 0.5$ is considered for the BLX crossover which allows the selection of parameters to generate the offspring up to 0.5M on either side of each parent parameter (Figure 5).

2.4.2 IBEA

In contrast to previous population-based GA, IBEA [37] adapts the search by using the HV index in the selection process. Based on an initial population, the fitness value $F(x_i)$ of each individual is calculated. Among many options, one possibility is to sum up the HV values of an individual with respect to the rest of the population (Eq. 9) where P represent the Pareto data set. As for NSGA-II, a tournament selection is carried out for the parent population members based on a ranking of the corresponding fitness values ($F(x_i)$). An offspring is generated from the mating selection through a crossover operator that randomly combine certain genes from each parent and through mutation which randomly picks a gene and modifies it (Figure 4). This is repeated iteratively until the convergence of the optimisation is reached. In the present study, uniform mutation and binary crossover operators are used.

DOI: 10.13009/EUCASS2022-6041

$$F(x_i) = \sum_{x_j \epsilon P \setminus \{x_i\}} HV\left(\{x_j\}, \{x_i\}\right) \tag{9}$$

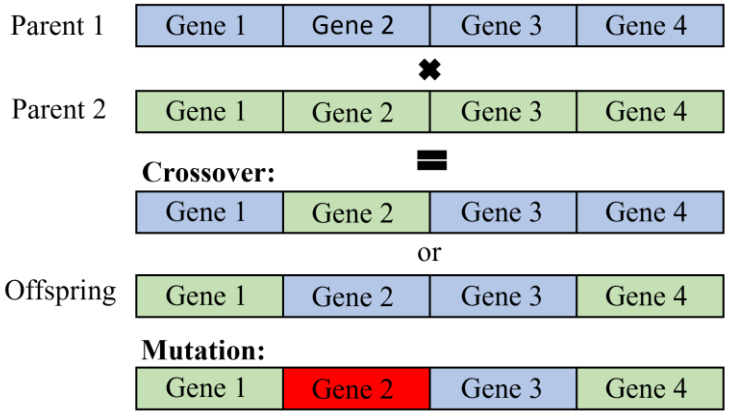


Figure 4: Randomness in crossover and mutation operators.

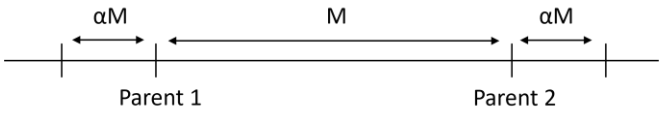


Figure 5: Blended BLX-α crossover operator.

2.4.3 OMOPSO

A PSO method is based on the ability of semi-autonomous particles to communicate with each other and to update their status. In this method the particles are able to sense their surroundings in the design space. This particular searching approach is modelled by the individual position (Eq. 10) and velocity (Eq. 11) vectors which are updated in each generation. The position vector shows the particle position in the design space. The velocity vector indicates the direction and intensity of the movement of that particle in the landscape. Three components constitute the velocity vector. The first term $w\ \overrightarrow{V_i^t}$ is the inertial one which essentially maintain the current direction of the movement of the particle. The second term of the velocity vector $c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right)$ is the individual or cognitive component which considers the personal best position of the particle ($\overrightarrow{P_i^t}$). The last term $c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right)$ is the social term that takes into account the global best position achieved within the whole population ($\overrightarrow{G^t}$). w, c_1 and c_2 are weighting factors to tune the balance between the exploration and exploitation in the searching process. While high values of w and c_2 = 0 maximise the exploration, low values of w and c_1 = 0 maximise exploitation. A high exploration ability is interesting for the initial generations of the optimisation. Exploitation is preferred when the optimisation is close to reach convergence. r_1 and r_2 are random values bounded from 0 to 1 that are applied to the learning terms.

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}} \tag{10}$$

$$\overrightarrow{V_i^{t+1}} = w\ \overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right) \tag{11}$$

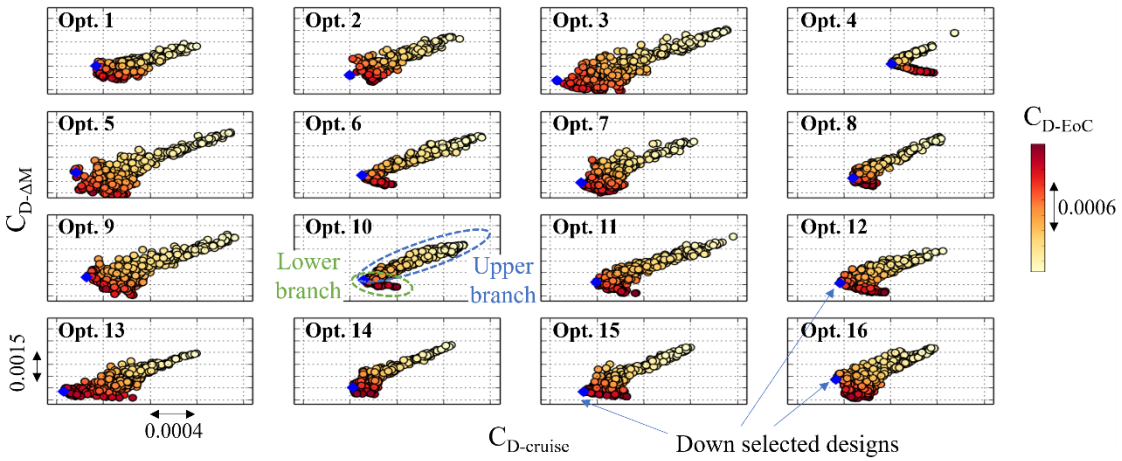
The OMOPSO [38] algorithm applies bounded random values to the weighting coefficients of the particle velocity vector. In particular, w is randomly set from 0.1 to 0.5 and c_1 and c_2 vary from 1.5 to 2. For a multi-objective optimisation the leader selection becomes challenging. For this purpose, OMOPSO applies the Pareto dominance criteria and a crowding distance factor [36] to the set of possible leaders. The algorithm divides the swarm in three equal groups applying three different mutation approaches to the velocity vector update of each group: no-mutation, uniform mutation, non-uniform mutation. A mutation probability factor of 0.1 is used in the present work [41].

DOI: 10.13009/EUCASS2022-6041

3. Results and discussion

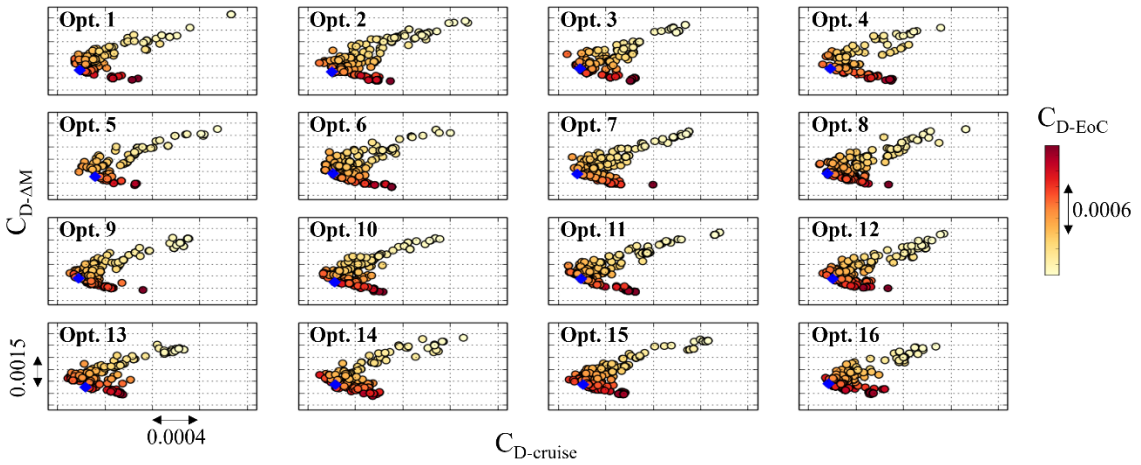
The configurations previously shown for NSGA-II, IBEA and OMOPSO (Section 2.4) have been applied for RANS CFD-based optimisations for a compact nacelle configuration. Each algorithm has been used to drive 16 different optimisations. This enables a statistical comparison of the optimal design space reached by every optimiser and also an analysis of the variability of each algorithm to converge to a certain solution. To purely focus on the impact of the optimiser, every optimisation starts from the same DoE. This removes the effect of randomness in the population of the design space in the first generation through the LHS sampling technique.

The Pareto data set obtained from the optimisation constitute a three-dimensional surface of the non-dominated designs (Section 2.4). In this work it is presented as a two-dimensional projection on the plane defined by $C_{D\text{-cruise}}$ - $C_{D\text{-}\Delta M}$ (Figure 6). Every data point is coloured by $C_{D\text{-EoC}}$. In a typical Pareto front for this optimisation problem, as $C_{D\text{-}\Delta M}$ decreases there is a penalty in $C_{D\text{-EoC}}$ which highlights the compromise between these two performance metrics (Figure 6a). For cruise drag the Pareto is divided into two branches. While in the upper branch a $C_{D\text{-cruise}}$ reduction implies a penalty in $C_{D\text{-EoC}}$, a reduction in $C_{D\text{-cruise}}$ in the lower branch increases $C_{D\text{-}\Delta M}$ (Figure 6a). However, this layout of the Pareto front changes substantially for some of the optimisations driven by NSGA-II. For example, optimisations 2, 3 and 4 result in different Pareto families with different sensitivities in the trade-off between the objective functions (Figure 6a). IBEA and OMOPSO show less variability in the Pareto front with convergence to the same expected Pareto front family composed of two branches (Figure 6b, Figure 6c). A statistical comparison has been carried out using the designs with minimum drag value for each objective function which provides a sample size of 16 for each operating condition. This is quantified relative to the NSGA-II results using the mean value ratio ($\mu/\mu_{NSGA\text{-II}}$), standard deviation ratio ($\sigma/\sigma_{NSGA\text{-II}}$) and amplitude range ratio ($\Delta/\Delta_{NSGA\text{-II}}$) where Δ is defined by the difference between the maximum and the minimum value (Table 1). Using the standard deviation ratio $\sigma/\sigma_{NSGA\text{-II}}$ as a measure of the variability of each optimiser to converge to a solution, IBEA and OMOPSO reduce the variability at cruise to 15% and 24% of $\sigma_{NSGA\text{-II}}$ (Table 1). Improvements of the same order of magnitude are obtained for the increased Mach number and end-of-cruise conditions for which OMOPSO outperforms IBEA (Table 1). Additionally, a comparison of the mean value ratio $\mu/\mu_{NSGA\text{-II}}$ and the amplitude range $\Delta/\Delta_{NSGA\text{-II}}$ indicate a high similarity based on drag performances in the optimal design space reached by IBEA and OMOPSO. However, a comparison with NSGA-II shows notable differences in the optimal design spaces. Although there is a slight difference between OMOPSO/IBEA and NSGA-II in $\mu/\mu_{NSGA\text{-II}}$ of about ±1% which equates to a negligible change in ΔC_D, the amplitude range $\Delta/\Delta_{NSGA\text{-II}}$ increases for NSGA-II between 80% and 85% depending on the condition. This indicates a potential low reliability on NSGA-II for this optimisation problem. An assessment of the population level of the Pareto shows an averaged reduction of 64% and 34% for OMOPSO and IBEA relative to NSGA-II. However, the variability in the population size reached by IBEA is higher than OMOPSO by a factor of 2.5. Moreover, OMOPSO shows a good sample resolution in the Pareto front which is the region of interest when down selecting a design (Figure 6c).

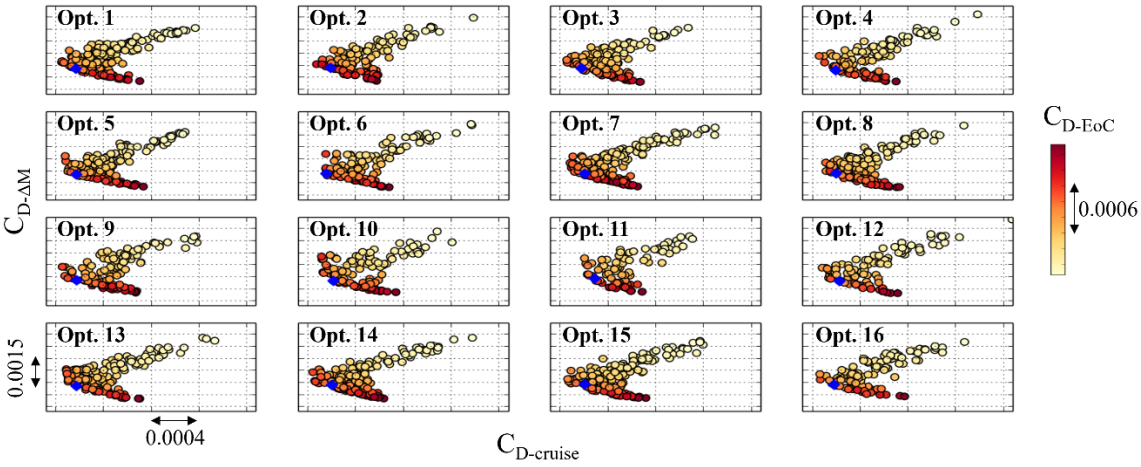


a) NSGAII

DOI: 10.13009/EUCASS2022-6041



b) IBEA



c) OMOPSO

Figure 6: Pareto fronts for the different optimiser. The blue dot is the down selected design (Eq. 8).

Table 1: Statistical comparison between IBEA and OMOPSO relative to NSGA-II for the designs with minimum drag at each operating condition.

		IBEA	OMOPSO
Min $C_{D-cruise}$ designs	$\mu/\mu_{NSGA\text{-}II}$	0.99	0.99
	$\sigma/\sigma_{NSGA\text{-}II}$	0.15	0.24
	$\Delta/\Delta_{NSGA\text{-}II}$	0.14	0.22
Min $C_{D-\Delta M}$ designs	$\mu/\mu_{NSGA\text{-}II}$	1.01	1.01
	$\sigma/\sigma_{NSGA\text{-}II}$	0.22	0.14
	$\Delta/\Delta_{NSGA\text{-}II}$	0.20	0.14
Min C_{D-EoC} designs	$\mu/\mu_{NSGA\text{-}II}$	1.01	1.01
	$\sigma/\sigma_{NSGA\text{-}II}$	0.22	0.19
	$\Delta/\Delta_{NSGA\text{-}II}$	0.21	0.16
Population	$\mu/\mu_{NSGA\text{-}II}$	0.66	0.33
	$\sigma/\sigma_{NSGA\text{-}II}$	0.85	0.34
	$\Delta/\Delta_{NSGA\text{-}II}$	0.73	0.28

A similar comparison of the optimal design space can be made based on the design variables (f_{max}, r_{max}, r_{if}, β_{nac}, Figure 1) of the designs contained in the Pareto data set. This data is presented in the format of a box plot for each parameter of each optimisation which compares how the data is distributed in the design space. The box plot extends from the minimum to the maximum value (Figure 7). Within this range, the median (or second quartile) divides the dataset into

DOI: 10.13009/EUCASS2022-6041

two halves. Additional split between the median and the ends are shown by the first quartile (Q1) and the third quartile (Q3) (Figure 7). The range between these first and third quartile is called interquartile range (IQR). This is a key parameter to identify outliers. A datapoint is an outlier if it is either below (Q1-1.5IQR) or above (Q3+1.5Q3).

A comparison of the optimal design space for each algorithm shows a high similarity for every design variable between IBEA and OMOPSO which is in line with the conclusion drawn based on the drag performance (Figure 8, Figure 6). Conversely, NSGA-II converges to a different optimal design space specially for f_{max} and β_{nac} with differences up to $0.03L_{nac}$ and 1.5° respectively (Figure 8). As was observed in the Pareto plots, the optimisations with NSGA-II have a high variability in the optimal design space whereas OMOPSO and IBEA are very consistent in the solution reached (Figure 8). This has been quantified based on a down selected design for each optimisation. The down selection of a design from a Pareto dataset is the final outcome from a multi-point optimisation problem. In this work, from all the possible optimal designs in the Pareto that meet $C_{D-\Delta M}/C_{D-cruise} < K_1$ and $C_{D-EoC}/C_{D-cruise} < K_2$, the design with the minimum cruise drag is selected (Section 2.4, Eq. 8). This provides a population of 16 for each algorithm to statistically compare the variability based on drag performances and design variables using the standard deviation ratio $\sigma/\sigma_{NSGA\text{-}II}$. For the drag at cruise and increased Mach number condition, IBEA and OMOPSO reduce the variability relative to NSGA-II to below $0.26\sigma_{NSGA\text{-}II}$ and $0.17\sigma_{NSGA\text{-}II}$ respectively (Table 2). At end-of-cruise condition, the reduction in variability is around $0.52\sigma_{NSGA\text{-}II}$ for IBEA and $0.63\sigma_{NSGA\text{-}II}$ for OMOPSO (Table 2). Substantial improvement for IBEA and OMOPSO are observed in terms of geometry parameters as well. While the reduction in $\sigma/\sigma_{NSGA\text{-}II}$ shown by IBEA is between 0.41 for f_{max} and 0.22 for r_{if}, OMOPSO further reduces the variability to 0.30 for f_{max} and 0.13 for r_{if} (Table 3).

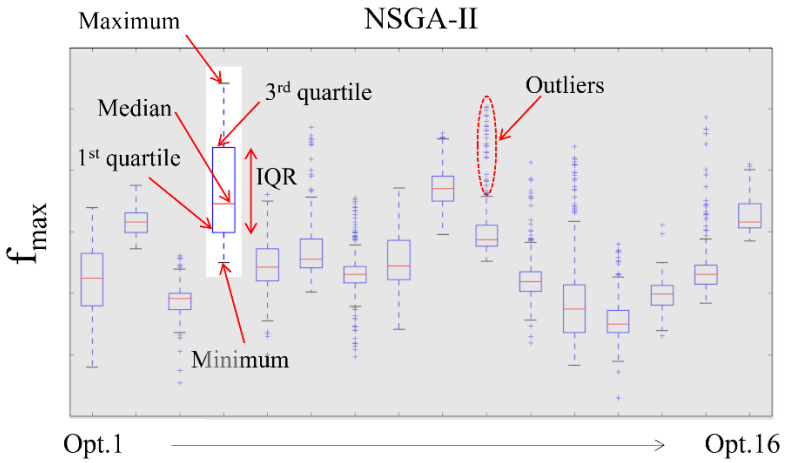


Figure 7: Definition of the box plot used to represent the optimal design space for an optimisation using the designs contained in the Pareto data set. Example for f_{max} and NSGA-II.

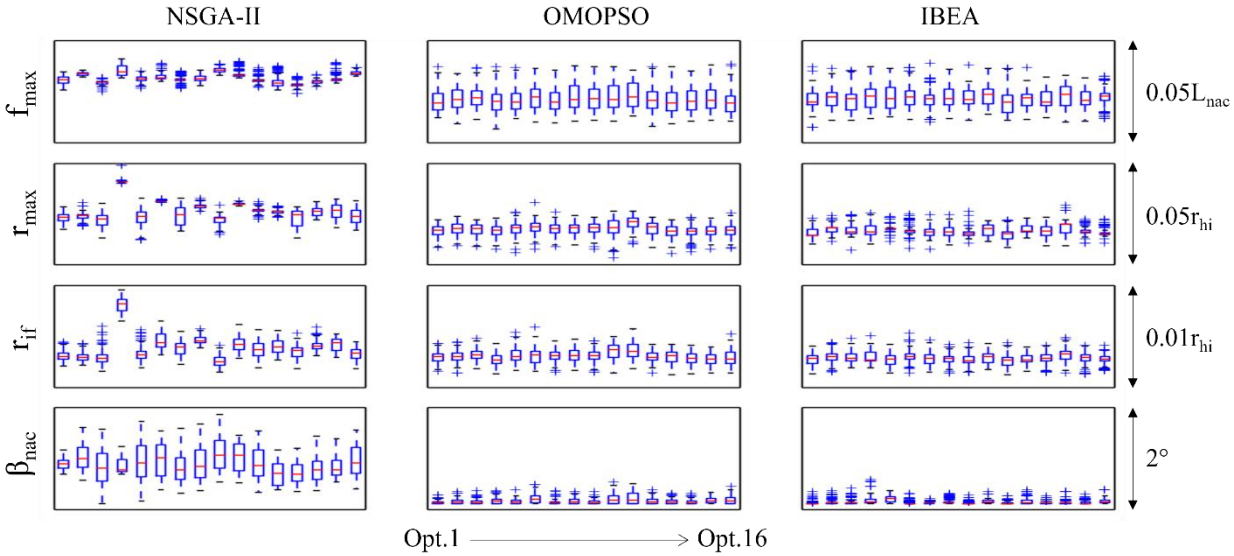


Figure 8: Optimal design space for every optimisation showed by box plots of the designs in the Pareto front.

DOI: 10.13009/EUCASS2022-6041

Table 2: Statistical comparison between IBEA and OMOPSO relative to NSGA-II for the variability of the drag performances of the down selected designs.

	$\sigma/\sigma_{\text{NSGA-II}}$		
	$C_{D-cruise}$	$C_{D-\Delta M}$	C_{D-EoC}
IBEA	0.21	0.26	0.52
OMOPSO	0.17	0.17	0.63

Table 3: Statistical comparison between IBEA and OMOPSO relative to NSGA-II for the variability of the design variables of the down selected designs.

	$\sigma/\sigma_{\text{NSGA-II}}$			
	f_{max}	r_{max}/r_{hi}	r_{if}/r_{hi}	β_{nac}
IBEA	0.41	0.23	0.22	0.25
OMOPSO	0.30	0.20	0.13	0.16

The convergence rate for each optimiser has been assessed based on the HV index (Section 2.4) and the associated number of generations of constant number of samples required to achieve convergence. Considering the criteria of a change in the HV index lower than 1% in the last 3 generations, NSGA-II turns out to be the fastest algorithm. The optimisations led by NSGA-II converge in average 4 and 6 generations earlier than OMOPSO and IBEA (Figure 9). It can be inferred that OMOPSO requires 2 generations less than IBEA to meet the convergence criteria (Figure 9) which provides a 6% reduction in the computation effort.

Overall, NSGA-II is discarded for this non-linear aerodynamic optimisation problem due to its high variability in the convergence to a solution compared to OMOPSO and IBEA. OMOPSO is selected as the most suitable algorithm for this application due to: (a) the overall lower variability in the convergence to the optimal design space and a down selected design and (b) a reduction of 6% in the computational overhead compared to IBEA.

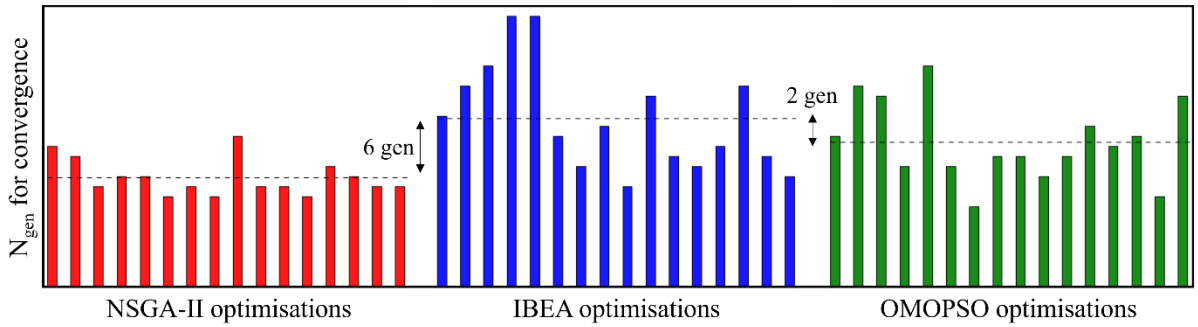


Figure 9: Comparative quantification of the convergence rate for each optimiser to meet the convergence criteria.

4. Conclusions

This paper presents a comparative assessment between two genetic algorithms (NSGA-II and IBEA) and a particle swarm optimizer (OMOPSO) for the aerodynamic shape optimisation of compact nacelles for ultra-high bypass ratio aero-engines. This problem is used due to the non-linearity that govern the aerodynamics of the nacelle. The focus of the study is on the variability of each algorithm to reach a solution and the comparison of the optimal design space obtained with the different optimisers. To conduct a statistical assessment, 16 separate optimisations have been carried out with each optimisation algorithm. A high variability in the optimal design space with different Pareto front families has been found for NSGA-II. In this respect, both IBEA and OMOPSO outperform NSGA-II. In terms of down selected designs, IBEA and OMOPSO reduce the variability relative to NSGA-II by a factor of up to about 8 and 9 respectively. Overall, OMOPSO shows the lowest variability to find the optimal design space and a down selected design with a reduction in the computation effort of 6% relative to IBEA to meet the optimisation convergence criteria. It is concluded that OMOPSO is the most suitable optimisation algorithm for this application.

DOI: 10.13009/EUCASS2022-6041

5. Acknowledgement

This project has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No 820997.

References

[1] Antoine, N. E. and I. M. Kroo. 2005. Framework for aircraft conceptual design and environmental performance studies. *AIAA J*. 43:2100–2109.
[2] Skinner, S. N. and H. Zare-Behtash. 2018. State-of-the-art in aerodynamic shape optimisation methods. *Appl. Soft. Comput*. 62: 933–962.
[3] Li, J. and M. Zhang. 2021. Data-based approach for wing shape design optimization. *Aerosp. Sci. Technol*. 112:1–22.
[4] Kim, J. H., B. Ovgor, K. H. Cha, J. H. Kim, S. Lee and K. Y. Kim. 2014. Optimization of the aerodynamic and aeroacustic performance of an axial-flow fan. *AIAA J*. 52: 2032–2044.
[5] Nejati, V. and K. Matsuuchi. 2003. Aerodynamics design and genetic algorithms for optimization of airship bodies. *JSME Int. J., Ser. B*. 46:610–617.
[6] Antunes, A. P. and J. L. F. Azevedo. 2014. Studies in aerodynamic optimization based on genetic algorithms. *J. Aircr*. 51:1002–1012.
[7] Haghighat, S., J. R. R. A. Martins and H. H. T. Liu. 2012. Aeroservoelastic design optimization of a flexible wing. *J. Aircr*. 49:432–443.
[8] Xia, C., T. Jiang and W. Chen. 2017. Particle swarm optimization of aerodynamic shapes with nonuniform shape parameter-based radial basis function. *J. Aerosp. Eng*. 30:1–12.
[9] Slowik, A. and H. Kwasnicka. 2020. Evolutionary algorithms and their applications to engineering problems. *Neural. Comput. Appl*. 32:12363–12379.
[10] Godinez, A. C., L. E. M. Espinosa and E. M. Montes. 2010. An experimental comparison of multiobjective algorithms: NSGA-II and OMOPSO. In: *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*. 1–6.
[11] Eberhart, R. C. and Y. Shi. 1998. Comparison between genetic algorithms and particle swarm optimization. In: *International Conference on Evolutionary Programming*. 611–616.
[12] Hassan, R., B. Cohanim, O. de Weck and G. A. Venter. 2005. A comparison of particle swarm optimization and the genetic algorithm. In: *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. 1138-1150.
[13] Wihartiko, F. D., H. Wijayanti and F. Virgantari. 2018. Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem. In: *IOP Conference Series: Materials Science and Engineering*. 1-6.
[14] Lyu, Z., Z. Xu and J. R. R. A. Martins. 2014. Benchmarking optimization algorithms for wing aerodynamic design optimization. In: *Proceedings of the 8th International Conference on Computational Fluid Dynamics*. 1-18.
[15] Mukesh, R., K. Lingadurai and S. Karthick. 2012. Aerodynamic optimization using proficient optimization algorithms. In: *2012 International Conference on Computing, Communication and Applications*. 1-5.
[16] Tejero, F., M. Robinson, D. G. MacManus and C. Sheaf. 2019. Multi-objective optimisation of short nacelles for high bypass ratio engines. *Aerosp. Sci. Technol*. 91:410–421.
[17] Tejero, F., D. G. MacManus, J. Matesanz-García, A. Swarthout and C. T. Sheaf. 2022. Towards the design and optimisation of future compact aero-engines: intake/fancowl trade-off investigation. In: *AERO2022: 56th 3AF International Conference*.
[18] Tejero, F., D. MacManus, J. Hueso-Rebassa, F. Sánchez-Moreno, I. Goulos and C. Sheaf. 2022. Aerodynamic optimisation of civil aero-engine nacelles by dimensionality reduction and multi-fidelity techniques. In: *AERO2022: 56th 3AF International Conference*.
[19] Kulfan, B. M. 2008. Universal parametric geometry representation method. *J. Aircr*. 45:142-158.
[20] Kulfan, B. M. 2010. Recent extensions and applications of the CST universal parametric geometry representation method. *Aeronaut. J*. 114:157-176.

DOI: 10.13009/EUCASS2022-6041

[21] Olson, E. D. 2015. Three-dimensional piecewise-continuous class-shape transformation of wings. In: *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* 1-16.
[22] Lane, K. A. and D. D. Marshall. 2010. Inverse airfoil design utilizing CST parameterization. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. 1-14.
[23] Christie, R., A. Heidebrecht and D. MacManus. 2017. An automated approach to nacelle parameterization using intuitive class shape transformation curves. *J. Eng. Gas Turbines Power*. 139:1-9.
[24] Christie, R., M. Robinson, F. Tejero and D. G. MacManus. 2019. The use of hybrid intuitive class shape transformation curves in aerodynamic design. *Aerosp. Sci. Technol*. 95:1-13.
[25] Tejero, F., R. Christie, D. MacManus and C. Sheaf. 2021. Non-axisymmetric aero-engine nacelle design by surrogate-based methods. Aerosp. Sci. Technol. 117:1-14.
[26] Schreiner, B. D. J., F. Tejero, D. G. MacManus and C. Sheaf. 2020. Robust aerodynamic design of nacelles for future civil aero-engines. In: *ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition*. 1-11.
[27] Ansys Inc. ANSYS FLUENT User's Guide. 275 Technology Drive.
[28] Menter, F. T. 1994. Two-equations eddy-viscosity turbulence models for engineering applications. *AIAA J*. 32:1598-1605.
[29] Sutherland, W. 1893. LII. The viscosity of gases and molecular force. *Lond. Edinb. Dublin Philos. Mag. J. Sci*. 36:507-531.
[30] Roache, P. J. A. 1994. A method for uniform reporting of grid refinement studies. *J. Fluids Eng.* 116:405-413.
[31] Robinson, M., D. G. MacManus and C. Sheaf. 2019. Aspects of aero-engine nacelle drag. *Proc. Inst. Mech. Eng. G.* 233:1667-1682.
[32] AGARD. 1979. Guide to in-flight thrust measurement of turbojets and fan engines. AGARD Report AG-237.
[33] Christie, R., S. Ramirez and D. G. MacManus. 2014. Aero-engine installation modelling and the impact on overall flight performance. In: Advance Aero Concepts, Design and Operations Conference.
[34] Robinson, M. H., D. G. MacManus, A. Heidebrecht and N. Grech. 2017. An optimisation method for nacelle design. In: *55th AIAA Aerospace Sciences Meeting*. 1-17.
[35] Cheval, A. 2019. Multi-objective optimisation of ultrafan nacelles. MSc Thesis. Cranfield University.
[36] Deb, K., A. Pratap, S. Agarwal and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6:182-197.
[37] Zitzler, E. and S. Künzli. 2004. Indicator-based selection in multiobjective search. In: *Lecture Notes in Computer Science*. 832-842.
[38] Sierra, M. R. and C. A. Coello Coello. 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. In: *Lecture Notes in Computer Science*. 505-519.
[39] Zitzler, E., K. Deb and L. Thiele. 2000. Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput*. 8:173-195.
[40] Eshelman, L. J. and J. D. Schaffer. 1993. Real-coded genetic algorithms and interval-schemata. In: *Foundations of Genetic Algorithms: 2*. 187-202.
[41] Hadka, D. 2015. Platypus Documentation. Python Software Documentation.