A Multifidelity Multiobjective Optimization Framework for High-Lift Airfoils

Jean Demange[*], A. Mark Savill[†] and Timoleon Kipouros[‡]

Cranfield University, Bedfordshire, MK430AL, United Kingdom

High-lift devices design is a challenging task as it involves highly complex flow features while being critical for the overall performance of the aircraft. When part of an optimization loop, the computational cost of the Computational Fluid Dynamics becomes increasingly problematic. Methods to reduce the optimization time has been of major interest over the last 50 years. This paper presents a multiobjective multifidelity optimization framework that takes advantage of two approximation levels of the flow equations: a rapid method that provides quick estimates but of relatively low accuracy and a reference method that provides accurate estimations at the cost of a longer run-time. The method uses a sub-optimization, under a trust-region scheme, performed on the low-fidelity model corrected by a surrogate model that is fed by the high-fidelity tool. The size of the trust region is changed according to the accuracy of the corrected model. The multiobjective optimizer is used to set the positions of the flap and slat of a two-dimensional geometry with lift and drag as objectives with an empirical-based method and a Reynolds Averaged Navier-Stokes equations solver. The multifidelity method shows potential for discovering the complete Pareto front, yet it remains less optimal than the Pareto front from the high-fidelity-only optimization.

I.   Introduction

Designing multi-element airfoils is a challenging and time-consuming task, however efficiency of the high-lift systems is primordial[1] both for production cost but also during the exploitation of the aircraft.[2] High-lift devices allow an increase of the lift generated by an aircraft wing. Their design aims at maximizing the maximum lift (highest lift generated by the airfoil before stall) for landing while having the highest lift-over-drag ratio for take-off.[3] The complexity of their design promotes the use of optimizer.[4–6] Furthermore, the aerodynamic flow around such geometries is very complex to model: the spatial discretization, known as meshing, is a complex task, and the physics of the flow intricate.[7,8] A couple of decades ago, the estimation of their performances was mainly performed with wind tunnel, a long and costly process. With the emergence of computational power, two different fidelity levels of Computational Fluid Dynamic (CFD) appeared during the last 50 years:[9] relatively fast methods, but inaccurate in highly non-linear regions of the objective space, based on empirical data and very simplified models, and more recently more complex models, but still simplified, based on the Reynolds Averaged Navier-Stokes[3] (RANS) intended to be more general.

This paper aims at using those different levels of fidelity available in industry to perform optimization with a reduced running time while conserving the accuracy of the RANS simulation. Such method are called multifidelity optimization and has been the subject of many papers over the last decades. Most of the methods are based on the Surrogate Based Optimization (SBO)[10] that requires a sampling prior the optimization. A mathematical model is then fitted on those points and an optimization is performed. New points can also be evaluated and used to update the surrogate model.[11] Evolutionary algorithms are often used with surrogates to find global optimum while limiting the running time.[12] However this does not address

---

[*]Ph.D. researcher, School of Aerospace, Transport and Manufacturing, department of Power & Propulsion. Student Member.
[†]Professor of Computational Aerodynamics Design, School of Aerospace, Transport and Manufacturing, Propulsion Engineering Centre. Senior Member.
[‡]Research Fellow, School of Aerospace, Transport and Manufacturing, Propulsion Engineering Centre. Member.

the use of multiple fidelities available during optimization. The simplest method is similar to the process used by industries: a quick optimization is first performed on the low-fidelity function and promising points are fed to the high-fidelity evaluator. However, good points can be discarded by the inaccurate low-fidelity model.[13–15] Nowadays a lot of work is dedicated towards the trust region methods:[16,17] the optimization on the low-fidelity model, corrected by a surrogate model, is performed under a trust region, where the corrected model is known to be accurate. If a sufficient decrease is achieved by the resolution of the sub-problem, the optimum is evaluated by the high-fidelity model and its value is used to improve the fidelity of the corrected model. In Alexandrov et al.'s[16] work, the corrected model is constructed so that its value and its gradient at the sampling points are exact (first-order accurate). The method was then extended so that the gradients are no-longer required[18–23] to build the model (zero-order accurate). All the previous methods are designed to converge toward the nearest local minimum. Jarett et al.[24] extended the work to use Tabu Search, a step-based meta-heuristic optimizer for global convergence.

Multifidelity applied to multi-element optimization is still uncommon. Alexandrov et al.[25] applied their first-order consistent method to such case. The results were encouraging with a reduction of 5 times of the number of high-fidelity calls. Finally, Demange et al.[26] applied the derivative-free multifidelity method of Jarett et al. to a two-dimensional multi-element airfoil. This paper is an extension of their work to multiobjective optimization. Section II presents the multifidelity multiobjective method. The method is then applied to the multi-element two-dimensional Garteur airfoil[27] in section III. Finally section IV presents conclusion and opens to future work.

II.   Multifidelity Optimization Methodology

The method developed in this paper is an extension of the work carried by the authors in Ref. 26. Two approximations of the aerodynamic physics are used in order to speed up the convergence of the optimizer while retaining accuracy from the most accurate model. The first model is relatively fast (less than a minute) but relies more on empirical corrections; it will be called *low-fidelity* estimator. One can argue that this model includes many empirical correction that are more physics-accurate, yet the accuracy is arguable when going away of the applicability range; this is when the higher fidelity predictor, which have a more consistent response, will come into play. The latter is more accurate but requires more time to run (in the order of 15 minutes for this case, but for industrial applications, time can go up to several hours); it will be referred in the rest of the document as *high-fidelity* estimator.

Let's define the low-fidelity function as $\boldsymbol{f}_{\text{low}} : \mathbb{R}^n \mapsto \mathbb{R}^l$ and the high-fidelity function as $\boldsymbol{f}_{\text{high}} : \mathbb{R}^n \mapsto \mathbb{R}^l$; where n is the design space size and l is the objective space size. In addition to the work presented in Ref. 26, the objective space dimension can be greater than 1: $l \geq 1$. The goal is the minimization (or maximization) of the high-fidelity function as defined in (1), where $\boldsymbol{x}$ is a vector of dimension n and $\mathcal{D}$ is a subset of $\mathbb{R}^n$:

$$\min_{\boldsymbol{x}\in\mathcal{D}} \boldsymbol{f}_{\text{high}}(\boldsymbol{x}). \tag{1}$$

The problem (1) is solved by a derivative-free trust region method presented by Conn et al.[19] Instead of directly solving the high-fidelity problem, an approximation model is used, and at iteration k, the problem (2) is solved under a trust region $\mathcal{T}_k$:

$$\min_{\boldsymbol{x}\in\mathcal{T}_k} \boldsymbol{m}_k(\boldsymbol{x}). \tag{2}$$

The trust region is usually defined as a ball centered on the initial point of the sub-problem but in Ref. 26, a second definition based on the number of improvements of the corrected model is used. This paper only uses the step-based trust region presented in more details in section II.C. The model $\boldsymbol{m}_k : \mathcal{T}_k \mapsto \mathbb{R}^l$ is defined by (3) at iteration k. It is formed by the sum of a surrogate model $\boldsymbol{e}_k$ representing the difference between the low- and the high-fidelity functions (the error in other terms) and of the low-fidelity function.

$$\boldsymbol{m}_k(\boldsymbol{x}) = \boldsymbol{f}_{\text{low}}(\boldsymbol{x}) + \boldsymbol{e}_k(\boldsymbol{x}) \tag{3}$$

The single objective framework is presented in Ref. 26. The full diagram of its extension to multiobjective is shown in figure 1; the next sections follow the diagram and explain each part. Similarly to the previous work, no sampling is required to start the optimization. A starting point is given and evaluated with both low- and high-fidelity functions to provide a first correction. The master script then handles the general progress of the multifidelity optimization.

American Institute of Aeronautics and Astronautics

II.A. Sub-problem optimizer

The sub-problem optimization solves the problem (2). To do so, a surrogate of the error for each objective is constructed as described in section II.E. Thereafter for each function call, all surrogates are evaluated at the current point to obtain $\boldsymbol{e}_k(\boldsymbol{x})$ defined in (2) and the low-fidelity function is evaluated. The corrected function $\boldsymbol{m}_k$ is obtained by adding for each objective the low-fidelity function and the correction model. The use of Euclidean trust region definition[19] does not apply well to multiobjective optimization since solutions are more scattered. Therefore this work focuses on the step-based trust region defined by the number of times the Pareto front is improved, specifically, the number of times a dominating or Pareto-equivalent point is found. Once the required number of Pareto improvements is reached, the sub-problem is considered solved and the candidate Pareto front is fed back to the master script.

II.B. Computation of the ratio of improvement

The framework is extended to handle the high-fidelity function computation in parallel in order to soften the computational time overhead introduced by the increased number of points given by the sub-optimizer. This is used to compute, for each point and for each objective, the ratio of improvement $\boldsymbol{\rho}_k = [\rho_k^{(1)}, \rho_k^{(2)}, \ldots, \rho_k^{(l)}]$ defined by

$$\rho_k^{(i)} = \frac{f_{\text{high}}^{(i)}(\boldsymbol{x}_k) - f_{\text{high}}^{(i)}(\boldsymbol{x}_s)}{m_k^{(i)}(\boldsymbol{x}_k) - m_k^{(i)}(\boldsymbol{x}_s)}, \quad i \in [1, \ldots, l]. \tag{4}$$

Depending on the optimizer used, the candidate Pareto front will have different number of points, sometimes a relatively high number. Because each point evaluation has a high time cost, and because high-fidelity points are used to correct the function via a surrogate model, the number of points to evaluate must be limited. This is achieved by filtering the candidate Pareto front based on the Euclidean distance between each point. All unsuccessful points (configuration that results in a failed high- or low-fidelity function evaluation) are discarded. The remaining ones are retained to be compared with the actual Pareto front and to be used at a later stage to construct the surrogate models. The authors would like to highlight the increased number of high-fidelity points computed at each sub-optimizer iteration compared to its single-objective counterpart. It leads to a faster correction of the design space for a similar running time, increasing the accuracy of the correction.

II.C. Trust region management

In order to handle the increased number of points and the concept of Pareto-equivalence, *memories* are implemented: the one coming from the sub-optimizer (called *current Pareto set*), the one of the overall optimization (called *overall Pareto set*) and the current point that is the point from where the sub-optimizer will start (and also from which the trust region is defined); this point changes only after successful iterations.

Since each point of the current Pareto front is evaluated with the high-fidelity function, the front is updated accordingly: some points that would be Pareto equivalent using $\boldsymbol{f}_{\text{corr}}$ won't have the same status once compared with $\boldsymbol{f}_{\text{high}}$. Figure 3 shows, with star markers, the Pareto front obtained from the sub-optimizer using values of the corrected function $\boldsymbol{f}_{\text{corr}}$. In the sub-iteration presented, no points are filtered out. Each candidate point is then updated with its high-fidelity value depicted in figure 3 with arrows joining $\boldsymbol{f}_{\text{corr}}$ and $\boldsymbol{f}_{\text{high}}$. Red markers represent the Pareto-optimal points using the high-fidelity function; only those points are kept for the determination of the iteration status.

The ratio of improvement's definition was extended according to equations (4) and (6) in order to be used with multiple objectives as follow:

- for each candidate point $\boldsymbol{p}_k = [f_{high}^{(1)}(\boldsymbol{x}_k), \ldots, f_{high}^{(l)}(\boldsymbol{x}_k)]$ in the objective space,

  - if the number of objectives having their ratio of improvement greater than ρ_{bad} is higher or equal to the trigger n_f, the point $\boldsymbol{p}_k$ is marked *moderate*;

  - if the number of objectives having their ratio of improvement strictly greater than ρ_{good} is higher or equal to the trigger n_f, the point $\boldsymbol{p}_k$ is marked *good*;

  - otherwise, the point is marked *bad*.

- If the number of points $\boldsymbol{p}_k$ marked:

American Institute of Aeronautics and Astronautics

- *moderate* is greater than the trigger n_p, the overall iteration is considered *moderate*,
- *good* is greater than the trigger n_p, the overall iteration is considered *good*,
- otherwise the iteration is considered *bad*.

The trust region is increased or shrank according to the status of the current iteration as in (5):

$$\Delta_{k+1} = \begin{cases} \Delta_k - \gamma_s & |\{\boldsymbol{p}_{\mathrm{moderate}}\}| < n_p & \text{(bad prediction)} \\ \Delta_k & |\{\boldsymbol{p}_{\mathrm{moderate}}\}| \geq n_p & \text{(moderate prediction)} \\ \Delta_k + \gamma_e & |\{\boldsymbol{p}_{\mathrm{good}}\}| \geq n_p & \text{(good prediction).} \end{cases} \tag{5}$$

II.D.   New point selection

The current Pareto front is then merged with the overall one and the number of points improving the Pareto front is counted. If at least one point improves the Pareto front, a new point $\boldsymbol{x}_s$ is randomly chosen from the overall Pareto front, otherwise the sub-optimization is restarted from the same point, but this time, with a more accurate corrected function:

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_s & \text{if } |\{\boldsymbol{p}_{\mathrm{good}}\}| \geq n_p \\ & \text{and} \quad \exists \boldsymbol{p} \in \{\boldsymbol{p}_{\mathrm{good}}\}, \forall \boldsymbol{p}_0 \in \{\boldsymbol{p}_{\mathrm{good}} \setminus \boldsymbol{p}\} \, \exists i \in [1,\dots,l], f_{\mathrm{high}}^{(i)}(\boldsymbol{x}_p) < f_{\mathrm{high}}^{(i)}(\boldsymbol{x}_{p_0}) \\ \boldsymbol{x}_k & \text{else,} \end{cases} \tag{6}$$

with $\boldsymbol{x}_p$ and $\boldsymbol{x}_{p_0}$ being the coordinates in the design space of the respective points $\boldsymbol{p}$ and $\boldsymbol{p}_0$.

The random selection is biased towards low-density part of the Pareto front in order to allow its entire exploration. To do so the Euclidean distances d_i between each consecutive point from the Pareto front $\mathcal{P}_{\mathrm{front}} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_f\}$, containing f points, are computed according to (7):

$$\forall i \in [1,\dots,f-1], d_i = \|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\|, \tag{7}$$

and then used as weight for the random selection.

II.E.   Surrogate Model

The method does not require any sampling: it starts from a single point evaluated both with low- and high-fidelity function during initialization, then once at least two sampling points are available in the database, a surrogate model is built. This surrogate is then used each time the low-fidelity method is evaluated during the trust region optimization to provide a correction. Thus the surrogate must be able to be constructed with sparse data points, and should not require time to evaluate new points or to be constructed. Moreover, data used to build the surrogate are coming from the CFD solver near maximum lift, a non-steady phenomena; the convergence achieved is usually lower than for a steady flow, introducing noise in the results. Radial Basis Function (RBF) are well suited when a limited number of training data is available,[28] in addition they proved to be robust when used with a Gaussian kernel function. More details on its implementation can be found in Ref. 26.
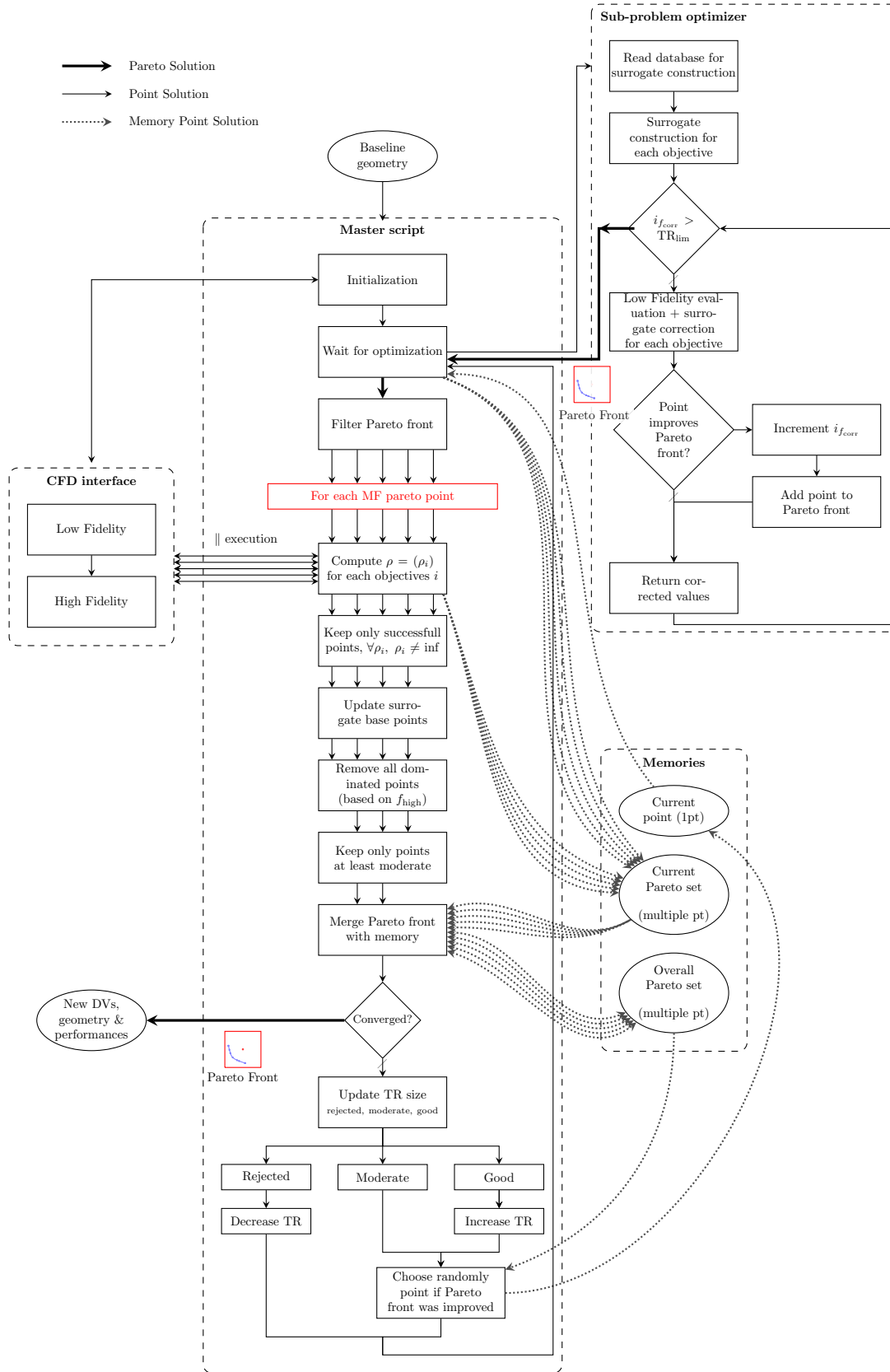
Figure 1. Multiobjective multifidelity optimization framework

American Institute of Aeronautics and Astronautics

III. Multiobjective High-Lift Airfoil Optimization

III.A. Sub-Problem Optimizer

The resolution of the sub-problem (2) requires the use of an optimizer. The aerodynamic optimization of high-lift devices is inherently multiobjective.[4] One way to solve this kind of problem is to use a single objective optimizer solving a weighted sum (or product[29]) of the objectives. The selection of the weights introduces bias in the optimization problem, and therefore the optimum found will be dependent on the choice of the weights.[30] If a point have one of its objective better and one worst than another point, it is not possible to consider one better than the other; all those points form the Pareto front. This set of points represent the optima of the function to optimize considering all objectives.

As in the single objective framework,[26] the Multiobjective Tabu Search (MOTS) implementation[31, 32] from Cambridge University is used to solve the sub-problem. The algorithm uses a Hooke and Jeeves[33] move for the local search. At each iteration, all the directions are tested, then compared and the best one, if allowed, is chosen for the next iteration. To prevent being trapped in local minimum, MOTS uses *memories*:

short term memory records all the visited points. They become *Tabu*, meaning that already evaluated points cannot be revisited. This enables to *climb away* from local minimum.

medium term memory records the best points found during the search. In multiobjective optimization, those points are Pareto-equivalent and form the Pareto front. They are used to restart local search once no new optima are found; this move is called *intensification*

long term memory records the points visited in the design space for exploring under-explored region once no new optima are found; this move is called *diversification*.

MOTS presents a good compromise between global exploration and local search while being robust against noise. It is worth noting that MOTS does not require gradients of the function which are hard to obtain for such high angle of attack study.

III.B. Optimization Set-up

The test case presented in Ref. 26 is also used in this study. The geometry and the design variables (see figure 2) are kept identical: the goal of the optimization is to set the slat and flap positions and deflection angles in order to maximize the two-dimensional performance at take-off. The flow conditions[27] (see table 1) remains also the same. The optimization is performed at fixed angle of attack taken as the angle of attack at maximum lift $\alpha_{cl_{\max}}$.
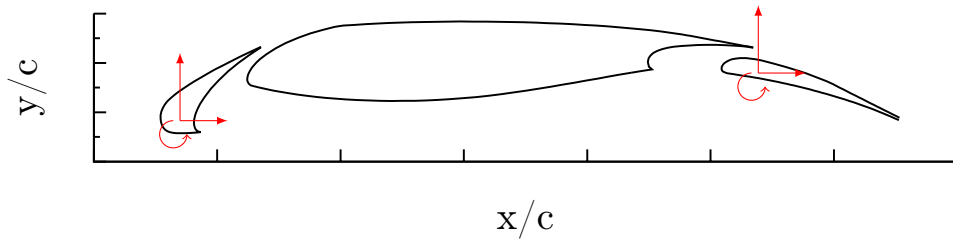

Figure 2. GARTEUR airfoil

Multiple objectives are used, therefore the performances measure of the configuration are no longer restricted to the lift-over-drag ratio. The lift and drag can now be distinguished giving more freedom in the selection of the best positions. In this way, the objectives are set in (8).[34]

$$f^{(1)} = -\frac{c_l}{c_{l_{\mathrm{datum}}}}, \quad f^{(2)} = \frac{c_d}{c_{d_{\mathrm{datum}}}} \tag{8}$$

III.C. Aerodynamic Tools

The same Computational Fluid Dynamics tools are used than in Ref. 26.

American Institute of Aeronautics and Astronautics

Parameters	Notation	Value
Reynolds number	Re	$4.10 \cdot 10^6$
Mach number	M	0.20
Fixed Angle Of Attack	$\alpha_{cl_{\max}}$	24 [deg]
Trigger for bad prediction	ρ_{bad}	0.0
Trigger for good prediction	ρ_{good}	0.0
Trust region shrink value	γ_s	2
Trust region expansion value	γ_e	2
Number of objectives required	n_f	1
Number of points required	n_p	2
Initial trust region size	τ_0	6

Table 1. High-Lift optimization parameters

HIGH-FIDELITY The commercial tool Fluent[35] is used as the most accurate tool for the study. The mesh is generated with ICEM[36] and a mesh regeneration method is preferred due to the large movement allowed for the slat and flap. The same reliable and robust meshing method than in Ref. 26 is used.

Reynolds Averaged Navier Stokes (RANS) simulations are the workhorse of the industry[37] to obtain accurate but relatively fast aerodynamic performances. This model along with a Spalart-Allmaras turbulence model[4] is used. Upwind second-order are used for all equations from iteration 500 to reduce divergence at start. The flow is considered converged when residuals fall below 10^{-5}, or if the number of iterations is greater than 2000[a]. Because convergence is not always achieved, the lift and drag coefficients are averaged over the last 100 iterations.

Validations against wind tunnel data is performed in Ref. 26. A single point evaluation requires between 10 to 20 minutes to run. This will be the expensive function in our study.

LOW-FIDELITY The MSES software from Professor Drela, MIT is used as the low-fidelity estimator. It is a coupled viscid/inviscid software: the inviscid Euler equations are coupled with a multiequation integral formulation, the coupling being performed with a Newton solver. This software handles wakes from each elements and can predict separation in the cove regions. It allows a good prediction of the lift and drag even near maximum lift.[9]

When used in challenging aerodynamic case, such as high-lift conditions, a single point evaluation requires up to 2 minutes. Massive separation, that can easily appear for bad design generated during optimization, produces divergence in the coupling procedure. This phenomena is amplified by the quality of the discretization used for the Euler solver. Despite the attention given to the selection of the mesh points distribution on the profile, divergence occurs for bad designs resulting in a restricted design space.

LOW-FIDELITY To reduce the time needed by the low-fidelity estimator, ESDU methods[38–42] were implemented to predict the maximum lift and its associated drag. Although the time required for this estimation is very small (of the order of 500 ms) leading to large reduction in the sub-problem resolution time, the method applicability is restrictive. Outside those bounds, the method provides either wrong trends, or no trends at all.

III.D. Results

Before drawing conclusions on the final solution set, some remarks regarding the sub-optimization are given in the following section.

III.D.1. Sub-Iteration

Figure 3 presents a result from the third sub-optimization: MOTS returns a Pareto front that contains points that should improve both objectives when considering the corrected function (shown with stars).

---

[a]This may be not enough to have fully converged flow but is sufficient to capture most of the flow physics

American Institute of Aeronautics and Astronautics

After evaluation with the high-fidelity function (with gray points, and arrows pointing from the corrected function to the high-fidelity one), those points reveal their true nature: only 3 points are Pareto equivalent; the others are discarded but still used to improve the corrected function for the following iterations. Out of those 3 points, one does not improve the overall Pareto front, depicted in blue in figure 3, but because the two other points improve the Pareto front, this iteration is considered successful and a new point will randomly be selected for the next iteration.

The optimization is run using either ESDU or MSES. Table 2 shows the iteration status distribution for the overall multifidelity optimizations and figure 4 shows their distribution with respect to the iterations. Considering ESDU optimization, the distribution is well balanced (apart from iteration 40 to 80) which indicates a good selection of the parameters presented in table 1. The high number of *bad* designs from iteration 40 to 80 is the result of the sub-optimizer always sending back a similar f_{corr}-Pareto front. Therefore, at each iteration only a small number of points are computed with CFD leading to a slow improvement of the correction. This is also found on the MSES optimization but no improvement is found during later iterations leading to an early stop. Finally, for the first 30 iterations, MSES and ESDU multifidelity optimizations presents the same behavior. Compared to the status *good* for the ESDU optimization, there are more *restart* status, which occurs when the sub-optimizer and the corrected function does a good job of providing well predicted points, but not good enough when compared with the overall Pareto front. This status appears mostly when ESDU gets unstuck but is also present on the MSES optimization. Despite the increased number of sampled points at each iteration, the number of refused points (marked as *bad*) is still higher than the number of accepted points: this highlights an issue in the accuracy of the corrected function that could be coming from either the correction or the low-fidelity function.
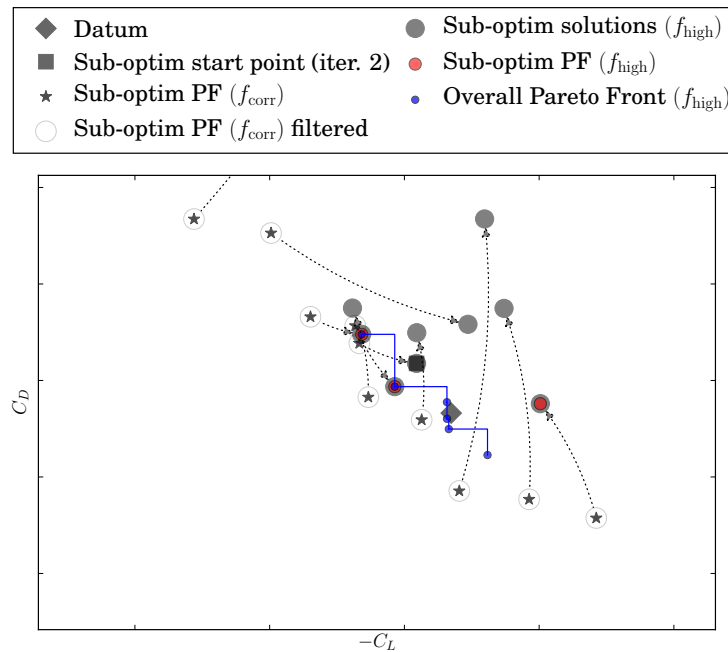

Figure 3. Multifidelity sub-optimization results with ESDU model at iteration 3. Datum is the starting point of the overall optimization. Points coming from the sub-optimizer are marked with a star and use their corrected function values; points kept after filtering are circled. Each point is evaluated with the high-fidelity function and their new position is plotted with gray circles; an arrow links the corrected and high-fidelity functions points. High-fidelity points that are Pareto-equivalent are marked with red circles. Finally, the overall Pareto front at the end of the iteration is shown in blue. *PF* stands for Pareto Front.

III.D.2. Final Pareto-front and Convergence

The validation of the method is done by comparing the Pareto front obtained by the f_{high}-only optimization with those obtained with the multifidelity framework. This is shown in figure 5 after the same number of high-fidelity function calls (CFD calls, as opposed to function evaluations —*eval*— that contains also
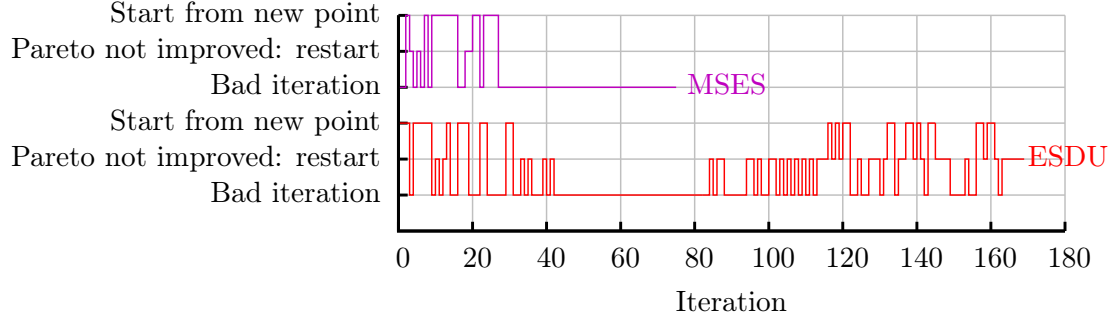
American Institute of Aeronautics and Astronautics

Figure 4. Iteration status distribution per iteration during the ESDU and MSES multifidelity optimizations

	Total	Bad	Restart	Good iteration
ESDU	170	92	47	31
MSES	76	57	4	15

Table 2. Global iterations status distribution during ESDU and MSES multifidelity optimizations

database recalls). The MSES multifidelity optimization though has a lower number of high-fidelity due to the early stall. Although the multifidelity optimizations do not reach the same accuracy than the high-fidelity optimization (similar outcome than for the single objective optimization[26]), the Pareto front is much more spread out giving much quicker the upper and lower part of the solution, parts being Pareto-dominant over the high-fidelity's one.

The convergence of the optimization is analyzed by plotting in figure 6 the Pareto fronts at 6 stages of the optimizations. Results observed in ref. 26 cannot be extended for the convergence of this optimization. The multifidelity optimization's Pareto front is constantly behind its high-fidelity-only counterpart. However, from the first iteration, the multifidelity front covers directly a bigger part of the objective space, and it remains true until the end of the optimization as shown in figure 5. MSES allows a quicker exploration of the low-drag part of the design space but seems to struggle to find good solutions in the high-lift part. Even though a smaller number of high-fidelity calls is used by the MSES optimization, the conclusion can hardly be extended to the elapsed time needed for the optimization. No time comparison is performed between high- and multifidelity since a different number of cores was allowed for the optimization. However, figure 7 shows the multifidelity time distribution between CFD and sub-optimization for each iteration. Depending on the low-fidelity tool used, the most time-consuming part changes between the CFD part for the quickest low-fidelity tool to the sub-optimization part for MSES, the most expensive tool . Therefore the low-fidelity model will have an impact on the overall time needed for the optimization but will be smeared out if the CFD time increases, for finer mesh or three-dimensional applications.

Figure 8 presents for each Pareto point the three optimizations' design variables with a parallel-coordinates plot:[43] each vertical axis represents a variable (first six are design variables and last two are the objective variables), and each colored line represents a point in the 8-dimensions space. The differences on the Pareto front are also seen on the design space. However some similar behavior are found between multifidelity and high-fidelity-only optimizations: although the slat x and y positions are not equal, their relative positioning is similar (lines between x and y slat positions are nearly parallel). Slat and flap deflections are also similar but more spread out for the multifidelity ones which explains the greater extent of the their Pareto fronts. Multifidelity solutions seem to be less advanced that the high-fidelity one: most of design variables are in the right direction but closer to the starting point, comment even more true for MSES that might have trouble to converge for design variables far from the starting point. Interestingly, the high-fidelity-only optimization considers only one positioning on the x-axis of the slat and a reduced number of positioning on its y-axis.
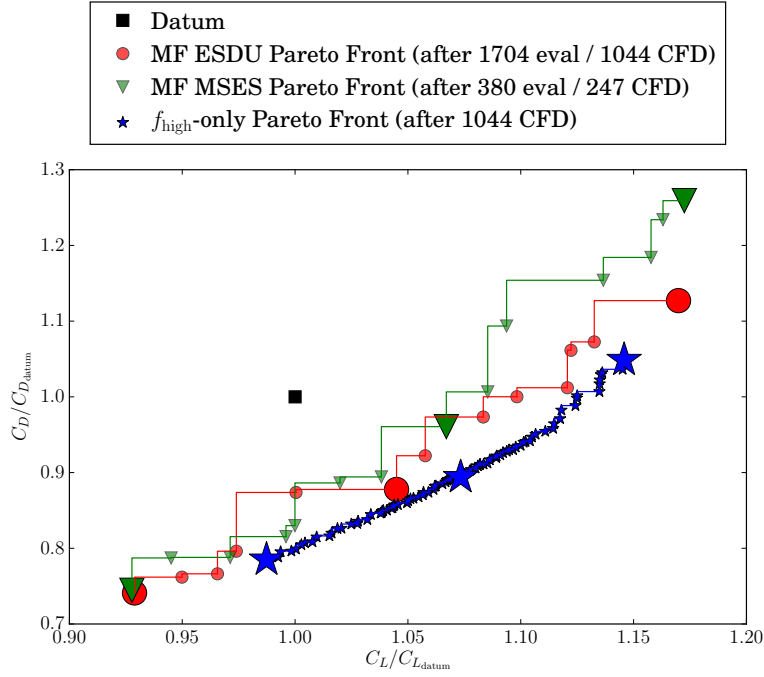
Figure 5. Pareto front after the same number of high-fidelity calls for the f_{high}-only and both multifidelity optimizations (ESDU and MSES). The number of evaluation (*eval*) takes into account both CFD runs and the use of already-run points during the optimization whereas *CFD* only takes into account the expensive calls. The multifidelity points are points computed by the high-fidelity function. The bigger markers are the configurations used for the deeper analysis.

III.D.3. Low-Fidelity Effect

To understand the effect of the low-fidelity on the final multifidelity solutions, figure 8 presents as well the Pareto front and design variables associated with the optimization performed only with the ESDU model as low-fidelity function. Firstly, none of the Pareto-equivalent points are improving the starting point: the low-fidelity does not give reliable information in the high-C_L and low-C_D part of the objective space. Secondly, we can focus on the design space regions found with the low-fidelity-only optimization. The solutions are much more spread out compared to the high-fidelity-only and multifidelity optimizations; yet, if only the region with most of the designs are considered, ESDU gives opposite trends for the slat position and deflection. The flap positions are however well estimated since ESDU tends to push the flap downstream, behind the main element, similarly to the high-fidelity position. Thus, ESDU methods is relatively accurate for positioning the flap, which accounts for most of the lift gain, but fails to position the slat in the required position to prepare the flow for the aggressive flap positioning; this results in a flow that cannot stay attached, and therefore results in a loss of lift. This illustrates the complicated objective space resulting from the highly non-linear behavior of the Navier-Stokes equations. The multifidelity framework is doing a great job at correcting the low-fidelity estimation, especially seen on the slat positioning, but does not use sufficiently the good information, particularly on the x-axis flap positioning.

III.D.4. Analysis of 3 Configurations

To further understand these discrepancies, the flow around the optimized geometry is compared for three positioning shown with bigger markers in figure 5. They are selected to represent: the maximum lift, the minimum drag and the *compromise* solution of the high-fidelity-only and multifidelity optimizations. The two extremes are the approximations of the *anchors points* and defines the bounds of the solution set. Those two points also define the *utopia point*[29] $\boldsymbol{P}_u$, which is the point having as objective space coordinates the corresponding anchor's coordinate:

$$\boldsymbol{P}_u = \{P_u^{(i)}, i \in [1, \dots, l]\} \text{ with } P_u^{(i)} = \min_{\boldsymbol{x} \in \mathcal{D}} f_{\text{high}}^{(i)}(\boldsymbol{x}). \tag{9}$$
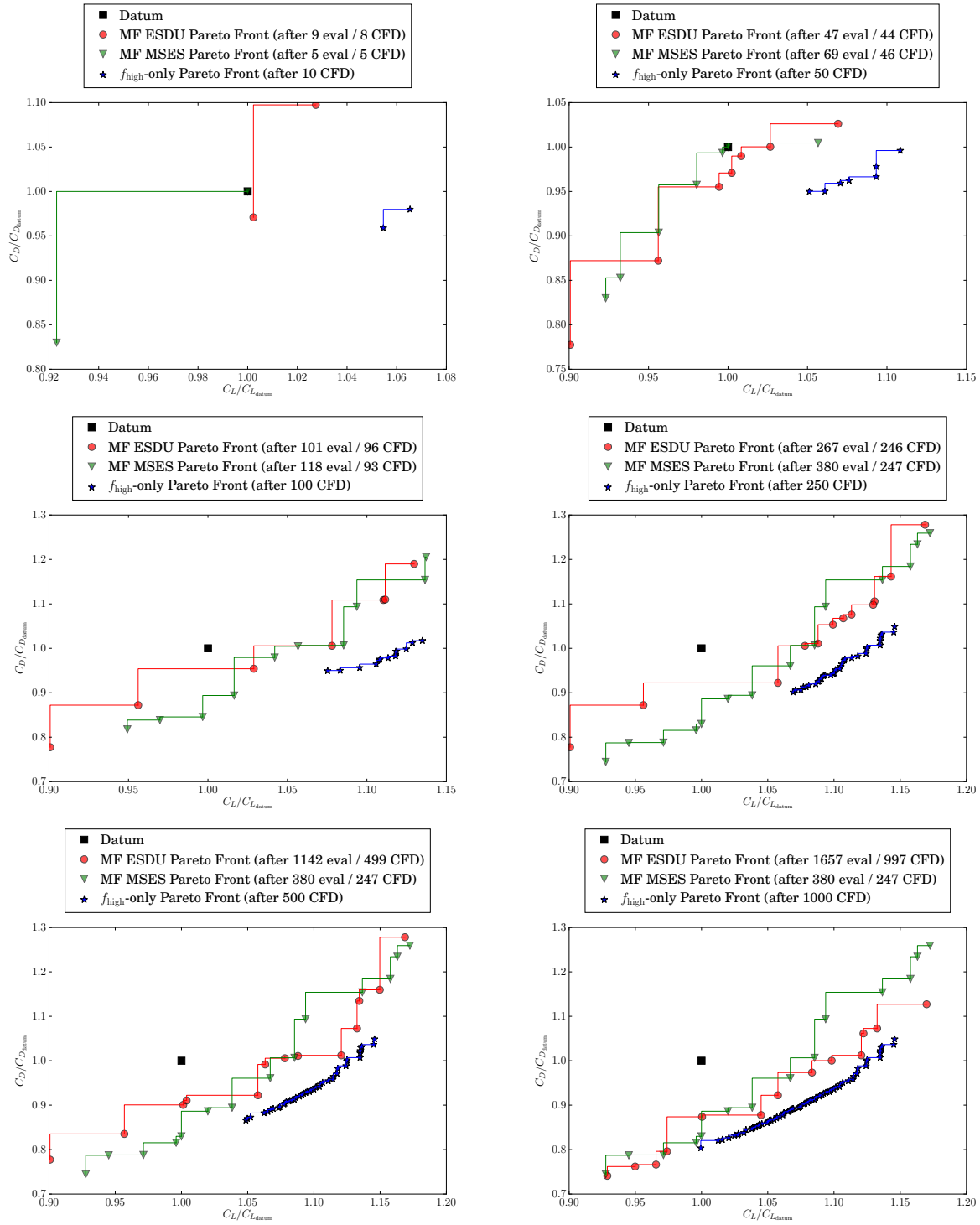
Figure 6. From top left to bottom right: Pareto front after 10, 50, 100, 250, 500 and 1000 CFD calls for both the high- and multifidelity optimizations.
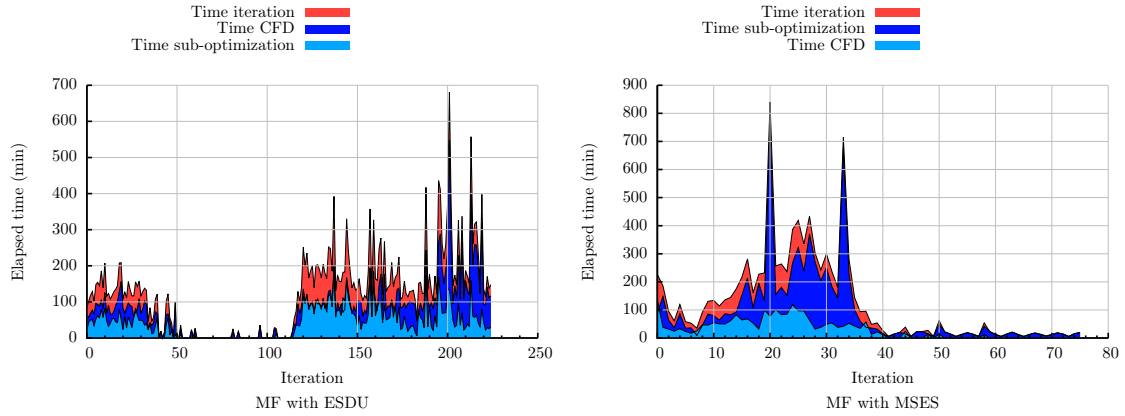
Figure 7. Times distribution of the multifidelity optimization between CFD and sub-optimization. On the left is the multifidelity optimization performed with ESDU and on the right with MSES.
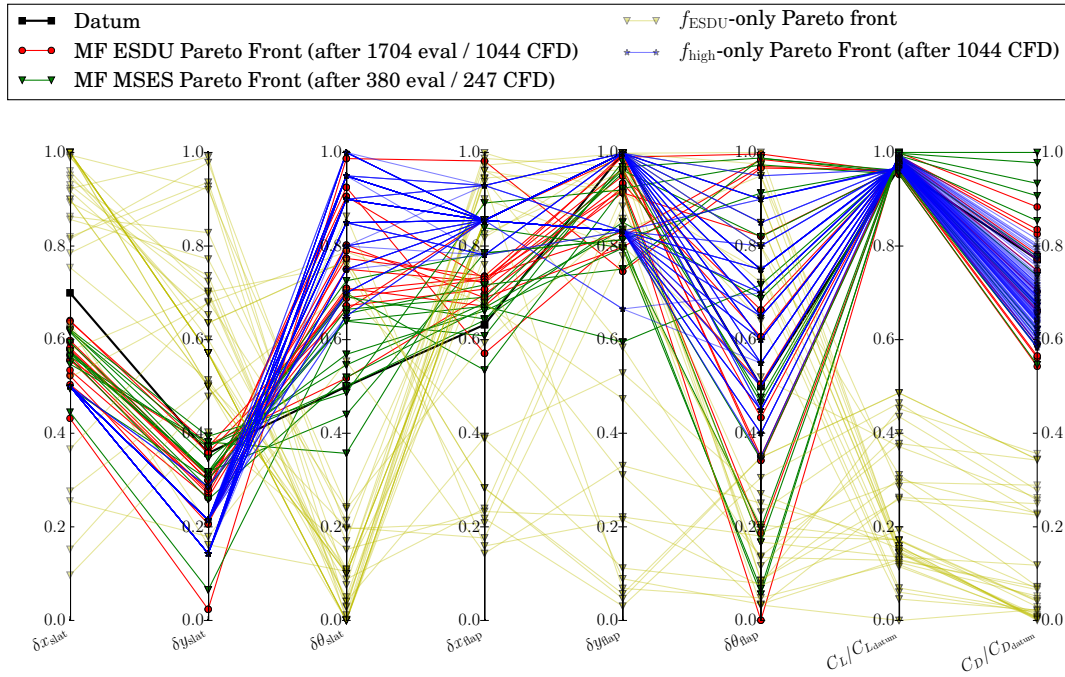

Figure 8. Parallel coordinates plot comparison of the Pareto fronts for: the f_{high}-only (blue), f_{low}-only (yellow), multifidelity ESDU (red) and mulifidelity MSES (green) optimizations including all design variables and the two high-fidelity objectives.

American Institute of Aeronautics and Astronautics

This point is approximated with the Pareto front obtained: $P_u^{(i)} \approx P_{u_\text{approx}}^{(i)} = \min_{\boldsymbol{p} \in \mathcal{P}} p^{(i)}$ with $\mathcal{P}$ the Pareto front. Similarly, the *nadir* $\boldsymbol{P}_n$ point can be approximated with $P_{n_\text{approx}}^{(i)} = \max_{\boldsymbol{p} \in \mathcal{P}} p^{(i)}$.

The *compromise* point $\boldsymbol{P}_c$ is then estimated by using the closest Pareto-front's point in the sense of the Euclidean distance after normalizing each point with the anchors:

$$\boldsymbol{P}_c = \min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p'} - \boldsymbol{P}_u\| \tag{10}$$

with $\mathcal{P}$ being the Pareto front and $\boldsymbol{p'} = [p'^{(1)}, \dots, p'^{(l)}]$:

$$p'^{(i)} = (p^{(i)} - P_{u_\text{approx}}^{(i)})/(P_{n_\text{approx}}^{(i)} - P_{u_\text{approx}}^{(i)}). \tag{11}$$

The approximated anchors and compromise points are shown with a parallel-coordinates plot in figure 9 for the three optimizations; the corresponding geometries are shown in figure 10. Table 3 presents the similar tendencies observed: the slat positioning presents the highest variability since apart from the low-drag configuration, its position differs for all methods. The flap positions however are more consistent: as expected the high-lift solutions have a high flap deflection, the low-drag solutions a low-deflection and the compromise a mid-position. All the high-lift solutions present a smaller overlap (flap more aft). In addition the multifidelity MSES high-lift solution shows a good agreement with the high-fidelity-only ones for all flap variables and slat deflection.
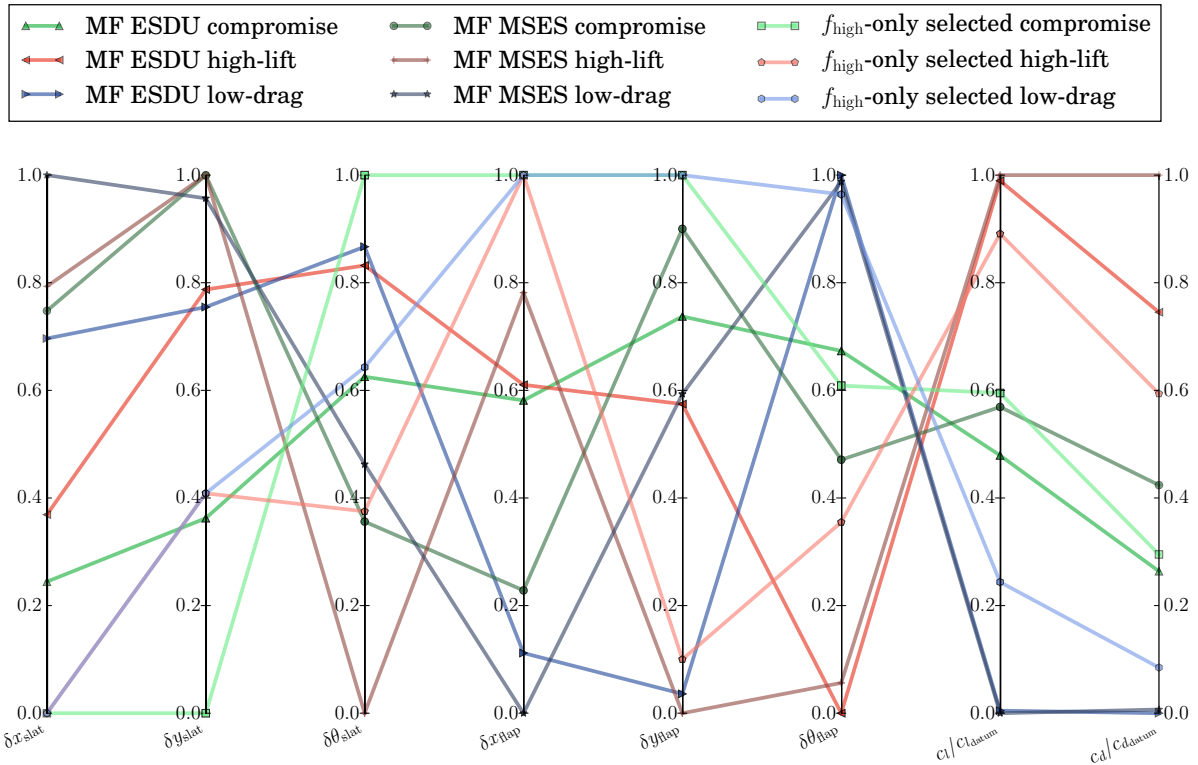

Figure 9. Normalized parallel-coordinates plot of the selected configurations representing the low-drag, compromise and high-lift configurations for the high- and both multifidelity optimizations.

Figure 11 shows the high-fidelity and multifidelity optimizations' lift and drag breakdown for each element, with the values normalized with the baseline configuration. The ESDU multifidelity high-lift solution presents a higher global lift due to the main element creating more lift whereas the high-fidelity-only optimization is driven towards an increased of the flap loading. In return the drag is increased for all elements of the ESDU multifidelity solution as well as the high-fidelity-only, but to a lower extent. This is confirmed by looking at
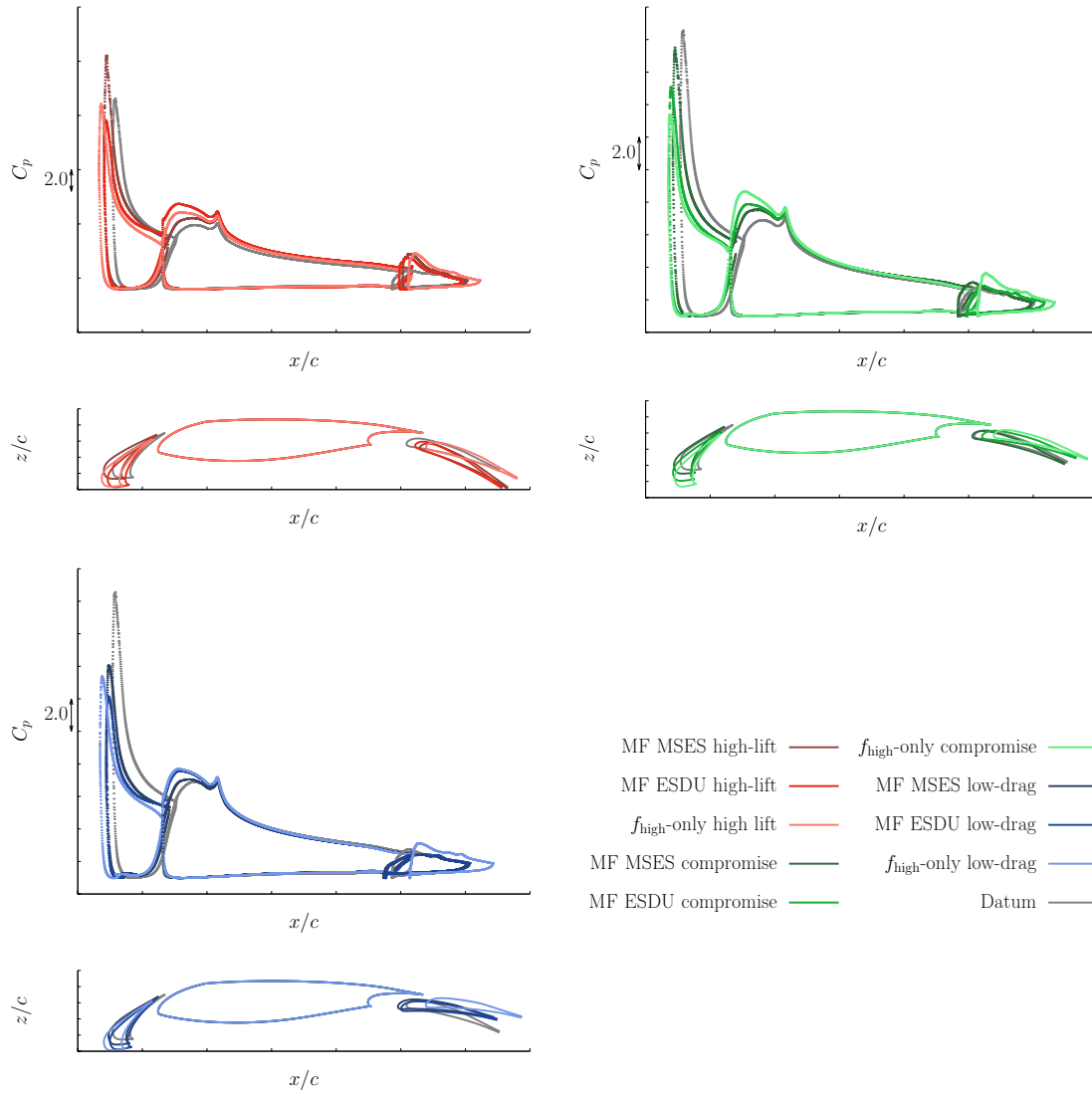
American Institute of Aeronautics and Astronautics

Figure 10. Pressure-coefficient plots (negative C_p on top) are also plotted for (from top left to bottom): high-lift, compromise and low-drag configurations.

	Slat	Flap
High-lift	No tendency	Low overlap and high deflection
Compromise	No tendency	High y_{flap} and mid deflection
Low-drag	Mid $\delta\theta_{slat}$	High $\delta\theta_{flap}$

Table 3. Similar tendencies on selected points

the flow wakes in figure 12: the multifidelity slat's wake is lower compared to the high-fidelity's one, but in return the global wake from the high-fidelity is reduced. This could be explained by a better understanding of the wake interaction by the high-fidelity-only optimization. This interaction is included in the ESDU method via empirical methods, but apparently not sufficiently to capture all its non-linearities. MSES multifidelity high-lift solution however increased drastically the slat and flap loading which allows the same production of lift than the ESDU multifidelity but generates also a lot more drag. Concerning the low drag solutions, both multifidelity and high-fidelity-only reduce the drag from all elements, the multifidelity ones being the most improved. The high-fidelity-only is attracted toward a more loaded flap region compared to the datum and multifidelity solutions. No clear remarks can be deduced from the contour plots in figure 12. Finally, regarding the compromise solution, again the flow layout is different between the two methods: ESDU multifidelity solution loads more the slat at the cost of lower loading on main and flap but benefiting the drag. Still, MSES solution seems closer in the physics sense to the high-fidelity solution compared to the multifidelity ESDU one. Hence increasing the low-fidelity accuracy seems to bring the multifidelity behavior closer to the high-fidelity one.

Concluding the results discussion, both methods' Pareto-front trends are similar, but when focusing on anchors and compromise solutions, many dissimilarities are observed. Likewise, the effect of the low-fidelity method cannot be separated from the final solution but an more accurate low-fidelity seem to lead to a better capture of the physics during the optimization.
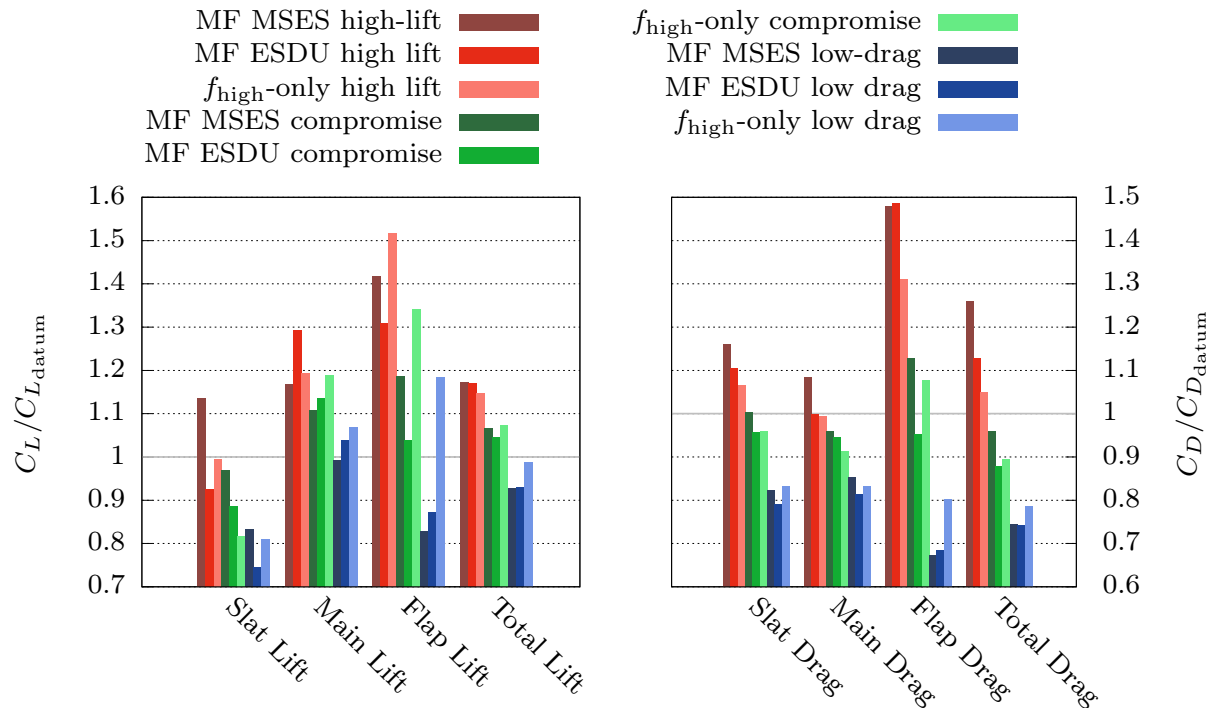

Figure 11. Lift and drag coefficients breakdown for both high- and multifidelity optimizations' high-lift, compromise and low-drag solutions. Values are normalized with the baseline configuration.

IV. Conclusions and Future Work

The multifidelity framework for optimization presented in Ref. 26 is extended to be used in a multiobjective optimization. Two fidelity levels are used: one of moderate accuracy but fast to run and another one, more accurate and more general but requiring more time to run. The method performs the optimization on the low-fidelity model corrected by a Gaussian RBF surrogate model. The optimization is done in a trust region defined by the number of improvements of the corrected function. The method is applied to the Garteur airfoil[26] with two objectives: the lift and the drag coefficients relative to the datum configura-
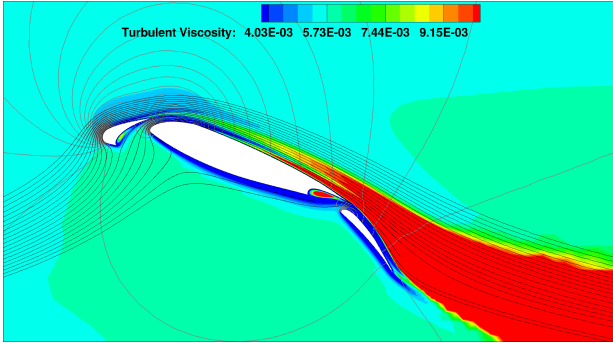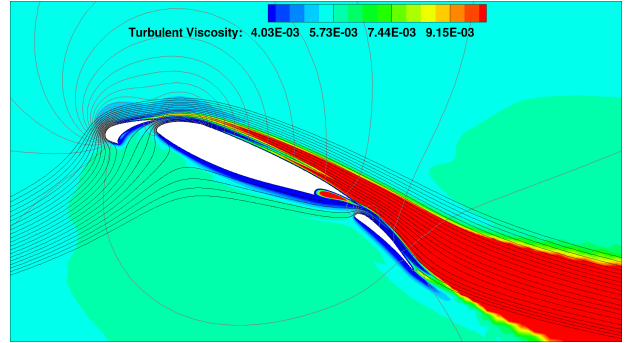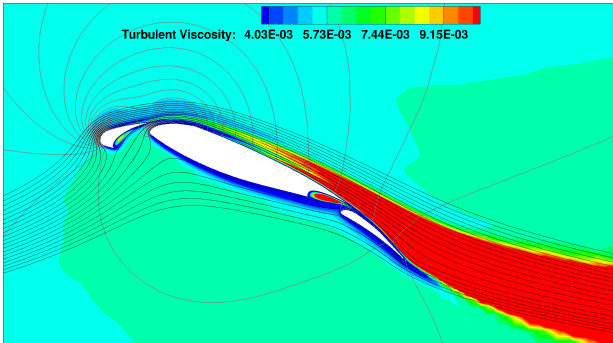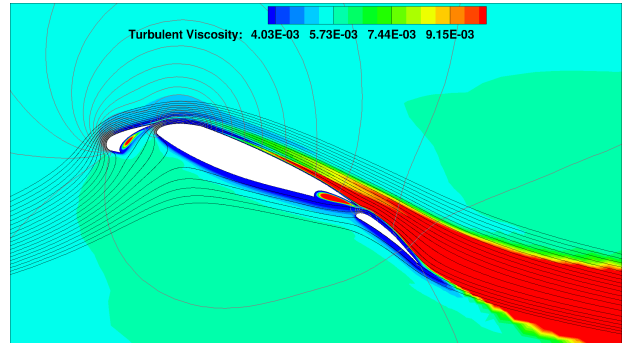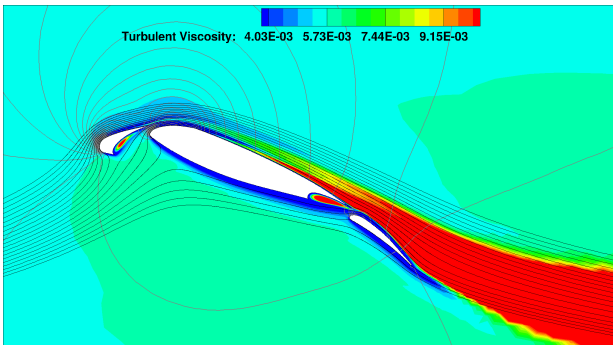
American Institute of Aeronautics and Astronautics
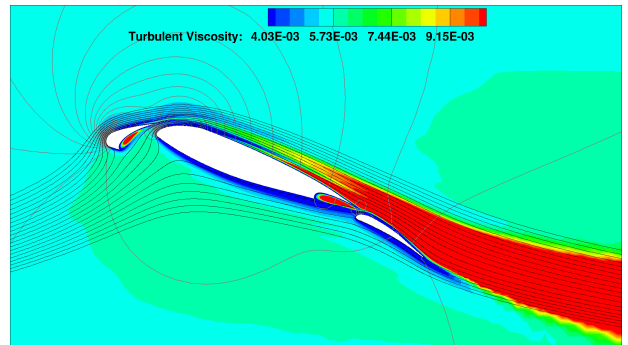
Figure 12. Turbulent viscosity contour plots for each selected points, namely high lift, compromise and low drag. Streamlines are plotted in black and iso-Mach number contour lines are in gray.

American Institute of Aeronautics and Astronautics

tion. The multifidelity multiobjective optimizer presented here yields to a Pareto front more extended than the solution from the high-fidelity-only optimizer. Yet, the Pareto-front from the multifidelity optimization appears to be dominated by the high-fidelity-only's one. A study of the flow for three points on the Pareto front is performed without drawing any conclusion as for the effect of the low-fidelity function on the final solutions. Yet, an increase in the accuracy of the low-fidelity seems to produce optima solutions closer to the high-fidelity-only ones.

Future work will be focusing on the comparison of different sub-optimizers and a more thorough analysis on the effect of the low-fidelity method on the final solution. Finally, the use of industrial three-dimensional test cases will be investigating to take advantage of the higher time ratio between high- and low-fidelity computations.

Acknowledgments

References

[1]Meredith, P., "Viscous phenomena affecting high-lift systems and suggestions for future CFD development," *AGARD, High-Lift System Aerodynamics 8 p(SEE N 94-18415 04-01)*, 1993.

[2]Rudolph, P. K., "High-lift systems on commercial subsonic airliners," 1996.

[3]Flaig, A. and Hilbilg, R., "High-Lift Design for Large Civil Aircraft," *High Lift System Aerodynamics, AGARD Conference Proceedings 515*, September 1991.

[4]Wild, J., "Multi-objective constrained optimisation in aerodynamic design of high-lift systems," *International Journal of Computational Fluid Dynamics*, Vol. 22, No. 3, March 2008, pp. 153–168.

[5]Brezillon, J., Dwight, R., and Wild, J., "Numerical aerodynamic optimisation of 3d high-lift configurations," *26th ICAS Congress*, 2008.

[6]Trapani, G., Savill, A. M., Kipouros, T., Agostinelli, C., and Rampurawala, A., "A Rapid Approach to the Aerodynamic Design of a Flexible High-Lift Wing," *32nd AIAA Applied Aerodynamics Conference*, June 2014.

[7]Greenman, R. M. and Roth, K. R., "High-lift optimization design using neural networks on a multi-element airfoil," *Journal of fluids engineering*, Vol. 121, No. 2, 1999, pp. 434–440.

[8]Smith, A. M. O., "High-Lift Aerodynamics," *Journal of Aircraft*, Vol. 12, No. 6, June 1975, pp. 501–530.

[9]van Dam, C. P., "The aerodynamic design of multi-element high-lift systems for transport airplanes," *Progress in Aerospace Sciences*, Vol. 38, 2002.

[10]Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, 2001, pp. 345–383.

[11]Shan, S. and Wang, G., "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 2, 2010, pp. 219–241.

[12]Lian, Y., Oyama, A., and Liou, M.-S., "Progress in design optimization using evolutionary algorithms for aerodynamic problems," *Progress in Aerospace Sciences*, , No. 46, 2010, pp. 199–223.

[13]Choi, S., Alonso, J. J., Kroo, I. M., and Wintzer, M., "Multifidelity design optimization of low-boom supersonic jets," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 106–118.

[14]Choi, S., Alonso, J. J., and Kroo, I. M., "Two-level multifidelity design optimization studies for supersonic jets," *Journal of Aircraft*, Vol. 46, No. 3, 2009, pp. 776–790.

[15]Dalle, D. and Fidkowski, K., "Multifidelity airfoil shape optimization using adaptive meshing," 2014, cited By 0.

[16]Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models," *Journal of Aircraft*, Vol. 38, No. 6, November-December 2001, pp. 1093–1101.

[17]Alexandrov, N., J. E. Dennis, J., and Lewis, R. M., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," Tech. Rep. 97-50, ICASE, October 1997.

[18]Oeuvray, R. and Bierlaire, M., "BOOSTERS: A derivative-free algorithm based on radial basis functions," *International Journal of Modelling and Simulation*, Vol. 29, No. TRANSP-OR-ARTICLE-2008-013, 2009, pp. 26–36.

[19]Conn, A. R., Scheinberg, K., and Vicente, L. N., "Global Convergence of General Derivative-Free Trust-Region Algorithms to First and Second-Order Critical Points," *Siam Journal on Optimization*, Vol. 20, 2009, pp. 387–415.

[20]Wild, S. M., Regis, R. G., and Shoemaker, C. A., "ORBIT: Optimization by radial basis function interpolation in trust-regions," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 3197–3219.

[21]Wild, S. M. and Shoemaker, C., "Global convergence of radial basis function trust-region algorithms for derivative-free optimization," *SIAM Review*, Vol. 55, No. 2, 2013, pp. 349–371.

[22]March, A. and Willcox, K., "Constrained multifidelity optimization using model calibration," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 1, January 2012, pp. 93–109.

[23]March, A. and Willcox, K., "Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives," *AIAA Journal*, Vol. 50, No. 5, 2012 2012, pp. 1079–1089.

[24]Jarrett, J. P. and Ghisu, T., "An Approach to Multi-fidelity Optimization of Aeroengine Compression Systems," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, USA*, 2012.

[25]Alexandrov, N. M., Nielsen, E. J., Lewis, R. M., and Anderson, W. K., "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization," *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, September 2000.

[26]Demange, J., Savill, A. M., and Kipouros, T., "Multifidelity Optimization for High-Lift Airfoils," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0557.

[27]Thibert, J. J., "The GARTEUR High Lift Research Programme," *High Lift System Aerodynamics, AGARD Conference Proceedings 515*, September 1991.

[28]Wang, L., Beeson, D., Wiggs, G., and Rayasam, M., "A comparison of metamodeling methods using practical industry requirements," *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Newport, RI*, 2006.

[29]Marler, R. T. and Arora, J. S., "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, Vol. 26, No. 6, 2004, pp. 369–395.

[30]Wild, J. and Brezillon, J., "Application Of Multi Objective Constrained Optimization In Aerodynamic High Lift Desing," *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems EUROGEN 2005*, 2005.

[31]Glover, F. and Laguna, M., "Tabu Search," Tech. rep., Kluwer Academic Publishers, Boston, MA, 1997.

[32]Jaeggi, D. M., Parks, G. T., Kipouros, T., and Clarkson, P. J., "The development of a multi-objective Tabu Search algorithm for continuous optimisation problems," *European Journal of Operational Research*, Vol. 185, 2008, pp. 1192–1212.

[33]Hooke, R. and Jeeves, T. A., ""Direct Search" Solution of Numerical and Statistical Problems," *Westinghouse Research Laboratories*, 1960.

[34]Trapani, G., *The design of High-Lift aircraft configurations through muti-objective optimisation*, Ph.D. thesis, Cranfield University, March 2014.

[35]ANSYS, Fluent 14.5, "Users Manual," ANSYS Inc.

[36]ANSYS, ICEM CFD 14.5, "Users Manual," ANSYS Inc.

[37]Rumsey, C. L. and Ying, S. X., "Prediction of high lift: review of present {CFD} capability," *Progress in Aerospace Sciences*, Vol. 38, No. 2, 2002, pp. 145 – 180.

[38]ESDU, "ESDU 97002: Information on the use of Data Items on high-lift devices," 2007.

[39]ESDU, "ESDU 94026: Introduction to the estimation of the lift coefficients at zero angle of attack and at maximum lift for aerofoils with high-lift devices at low speeds," 1995.

[40]ESDU, "ESDU 94027: Increments in aerofoil lift coefficient at zero angle of attack and in maximum lift coefficient due to deployment of various leading-edge high-lift devices at low speeds," 1994.

[41]ESDU, "ESDU 87005: Increment in aerofoil profile drag coefficient due to the deployment of a single-slotted flap," 2008.

[42]ESDU, "ESDU 87024: Low-speed drag coefficient increment at constant lift due to full-span plain flaps or controls," 2006.

[43]Inselberg, A. and Dimsdale, B., "Parallel coordinates," *Human-Machine Interactive Systems*, Springer, 1991, pp. 199–233.