High-Lift Devices Topology Optimisation using Structured-Chromosome Genetic Algorithm

Lorenzo Gentile[1,3], Elisa Morales[2,4], Domenico Quagliarella[2],
Edmondo Minisci[3], Thomas Bartz-Beielstein[1], and Renato Tognaccini[4]

[1]*Institute for Data Science, Engineering, and Analytics*, *TH Köln*, Köln, Germany
{lorenzo.gentile, thomas.bartz-beielstein}@th-koeln.de
[2]*Fluid Mechanics Department*, *Italian Aerospace Research Centre*, Capua, Italy
{e.moralestirado, d.quagliarella}@cira.it
[3]*Aerospace Centre of Excellence*, *University of Strathclyde*, Glasgow, UK
edmondo.minisci@strath.ac.uk
[4]*Department of Industrial Engineering*, *University of Naples "Federico II"*, Naples, Italy
renato.tognaccini@unina.it

Abstract—This paper addresses the problem of including the choice of the High-Lift Devices (HLDs) configuration as a decision variable of an automatic optimisation tool. This task requires the coupling of an estimation routine and an optimisation algorithm. For the former, SU2 flow solver has been used. The Structured-Chromosome Genetic Algorithm (SCGA) optimiser has been employed to search for the optimal HLD. SCGA can overcome the limitations dictated by standard fixed-size continuous optimisation algorithms. Indeed, using hierarchical formulations, it can manage configurational decisions that are conventionally the responsibility of expert designers. The search algorithm bases its strategy on revised genetic operators conceived for handling hierarchical search spaces. The presented research not only shows the practicability of delegating to a specialised optimisation algorithm the complete HLD design but is intended to be a proof of concept for the whole field of multidisciplinary design optimisation. Indeed, the aerospace sector as a whole would benefit by reducing human intervention from the decision process.

Index Terms—Aerospace engineering, High-lift devices, Optimisation, Genetic Algorithm, Mixed-variable, Variable-size

I. INTRODUCTION

Recently, in the field of commercial aircraft design, the interest regarding the optimal design of High-Lift Devices has increased [1], [2]. HLDs are components located on the aircraft's wing that aim to increase the lift force produced by the wing during slow flight phases, mainly take-off and landing. A wide variety of HLD types exists, but, the most common, and studied in this paper, are slats and flaps. The slat is located at the airfoil leading edge and has the effect of delaying stall by increasing the angle of attack at which the maximum lift is attained. Whereas, the flap is a trailing edge device that increases the lift coefficient even at low angles of attack [3], [4].

As discussed in [5], the HLD design is an arduous task because it is as a multidisciplinary (aerodynamics, structures, systems integration disciplines are involved), multi-objective (generally in conflict) [6], and multi-point (designed for take-off, climbing, approach, landing flight phases) [7] design problem. Thus, HLD design is very demanding from a computational point of view. One critical point is the proper convergence of the Computational Fluid Dynamic (CFD) solver because configurations at high angles of attack, close to stall conditions, must be simulated. Generally, these configurations lead to separate flows that make the accurate prediction of the aerodynamic performance very challenging. As a consequence, experimental results to validate the CFD are necessary.

Classically, the design of HLDs is a two-stage procedure, first a promising configuration type (number of airfoil elements) is selected, and second, the position and shape of the elements are obtained through optimisation, often employing heuristic algorithms [5]. Furthermore, because most of them are limited to continuous numeric variables and fixed-length search spaces, optimisation algorithms are assigned to solve a circumscribed part of the original design problem exclusively. It is a tough task to know beforehand which configuration is optimal in terms of performance, weight, and cost. As a result, this approach is likely to lead to sub-optimal solutions. This work aims to demonstrate the feasibility of including the HLDs configuration as a design variable of the optimisation phase. In particular, the optimiser explores the HLD search space changing the number of elements (or configuration) and their position and rotation. In the sake of simplicity, the objective of the optimisation is limited to maximise the lift generated by the HLD at a given angle of attack regardless of its drag, weight, costs and practical construction feasibility. Because of this, the problem falls under the area mixed-variable single objective global optimisation.

Dealing with these kinds of problems implies tackling many challenges. Indeed, even a proper definition of the design problem may be very difficult, as well as finding an efficient mapping and encoding of the design variables to facilitate the optimiser efficiency. This problem is even more critical when variables of a different type, like continuous (numerical) and categorical (nominal), are used to simultaneously encoding

topological and dimensional features. Among the different algorithms, one of the most suited to face this particular kind of problems are Genetic Algorithms (GAs) [8].

A variety of strategies for facing variable-size global optimisation is described in the literature [9], [10]. The *hidden gene* adaptation of GA for the optimisation of interplanetary trajectories is introduced in [10]. Here the maximum number of genes that can describe a candidate is set. Then, each candidate is represented using all the possible genes plus a set of *activation genes* indicating which genes to consider when computing the objective function. However, in the cases wh the activation of genes depends on the activation of other o or on the values they assume, defining an effective probl formulation can be very tough or even impossible.

A more complex, but efficient and flexible adaptation of is proposed in [9]. In this case, a hierarchical multi-level ch mosome structure is adopted in place of the standard *str* one. Unlike the standard GA formulation, both vicinity hierarchy relationships link the genes of each chromoso However, the operators implemented act regardless of the t of variables and can result ineffective or even destructive presented in [11].

In the current paper, a mixed-variable with dynamic varying search space global optimiser based on a genetic al gorithm, the Structured-Chromosome Genetic Algorithm [11]– [14] will be used. This approach consists of an adaptation of GA that allows structured-chromosomes definitions. The al gorithm takes advantage of hierarchical problem formulations and makes use of revised operators that behave considering the type of variables and the history of the optimisation [11], [13], [14]. Moreover, this offers the possibility to define dynamic bounds and dynamic variables dependencies, and this makes it suitable for a wide variety of problem formulations and applications.

The ambition of this work is to show that an appropriately crafted optimisation algorithm can handle configurational decisions that are usually the responsibility of experienced designers. Indeed, this achievement can represent a turning point not only in the field of design optimisation but also in multidisciplinary design optimisation, structural optimisation and many other design disciplines.

The structure of the paper is the following. Section II describes the autonomous aerodynamic computational chain. Section III goes through the details of the algorithm employed, SCGA. A validation of the optimiser on a *"toy"* problem is presented in Section IV to demonstrate the adequacy of SCGA to tackle airfoil optimisation problems where continuous play a dominant role. Section V focuses on the HLD optimisation problem and gives details about the formulation, the constraints handling and the algorithm's settings. Finally, in Section VI the results of the optimisation run are shown. To conclude the paper, Section VII resumes the critical aspects of this research and presents future works.

II. AERODYNAMIC COMPUTATIONAL CHAIN

When facing aerodynamic design optimisation problems, it is essential to have a complete autonomous aerodynamic computational chain. It takes as input the variables given by the optimiser, generates the candidate to be evaluated (in this case, a multi-element airfoil), builds the mesh, and, finally, runs the aerodynamic solver. Once the performance of the candidate has been obtained, the computational chain provides these values to the optimiser. This process is depicted as in Fig. 1 and explained in the following sub-sections.
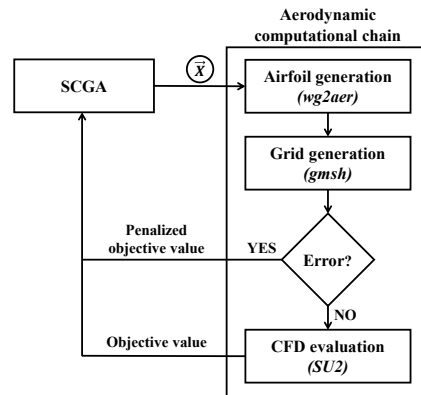

Fig. 1: Aerodynamic computational chain representation.

Moreover, in Section II-D the CFD solver is validated. In Section II-E the design of the different multi-element airfoil used as baseline configurations for the optimisation are generated.

A. Airfoil generation

During the optimisation, the candidate airfoils are generated using *wg2aer*[1]. It is a program that accepts as input a set of design variable values and modifies a specified starting airfoil, accordingly, using a set of modification functions. For obtaining the baseline airfoil, the design variable values must be equal to 0. In this paper, the introduced design variables are the settings of each airfoil element, which are the translation in the 2-dimensional space (ΔX and ΔY) and the rotation ($\Delta \theta$) with respect to the starting airfoil. The reference point for rotation is the leading edge of the corresponding element.

B. Grid generation

The grid is automatically built using a self-developed procedure based on the open-source grid generator *Gmsh* [15]. It generates an unstructured square grid combining triangular and rectangular elements. The far-field is located at 40 airfoil chords. The flat plate theory with the flow parameters is used to generate a proper set-up for the near-wall area with rectangular elements to accurately solve the boundary layer. Also, the presence of the wake is considered by refining that region.

---

[1]Program developed at Centro Italiano Ricerche Aerospaziali (CIRA)

C. CFD Evaluation

The CFD solver used for the HLD design optimisation is the open-source fluid-dynamic solver SU2 [16]. The compressible Reynolds-averaged Navier-Stokes (RANS) equations are solved using SA turbulence model [17]. In addition, JST central scheme with artificial dissipation coupled with an implicit Euler method for the pseudo-time stepping is used for spatial integration.

D. CFD results validation and design point selection

Before performing an optimisation run, the CFD solver must be validated. This step is essential when designing HLDs. During their design, configurations at high angles of attack are studied. These configurations could present highly separated flows, hence, predicting the airfoil performance is a challenging task for the solver. Therefore, a comparison with experimental results is required.

A series of wind-tunnel experiments were conducted at NASA Langley Research Center for the three-elements airfoil McDonnell Douglas (MDA) 30P-30N [18], [19]. Thus, the 30P-30N airfoil (Fig. 2) was selected as the starting configuration for this design optimisation problem. These experimental data are used to validate the numerical results.
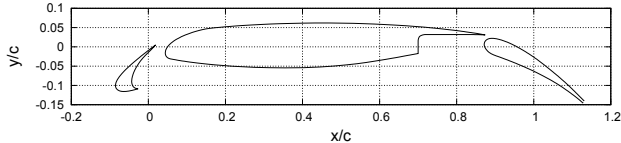
Fig. 2: 30P-30N airfoil.

The considered working conditions are: free-stream Mach number $M_\infty = 0.2$ and Reynolds number $Re = 5 \times 10^6$ (based on the airfoil chord, c) at several angles of attack, namely, $\alpha = [0°, 4°, 8.12°, 16.21°, 21.29°, 22.5°, 24°, 25°, 26°]$. A mesh with ~ 200000 cells is used.

To demonstrate the ability of the CFD solver to assess stall, a comparison of experimental and numerical lift coefficient (C_l) versus angle of attack (α) curves is shown in Fig. 3a. It is observed that the computed lift is slightly higher than the experimental data. In addition, the computed stall angle ($\alpha \simeq 24°$) is higher than the experimental ($\alpha \simeq 21°$).

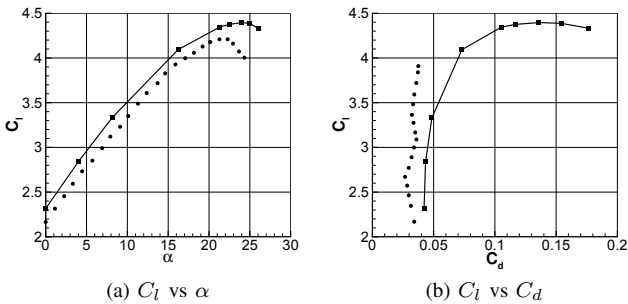(a) C_l vs α　　　　　　(b) C_l vs C_d

Fig. 3: SU2 (—■—) and experimental (•) data comparison.

The drag polar (Fig. 3b) shows that numerical simulations predict an higher drag coefficient (C_d). This trend is expected

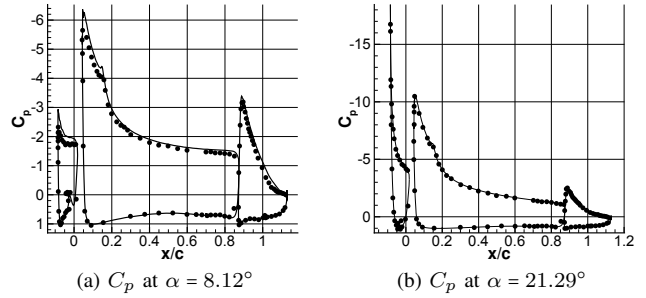(a) C_p at $\alpha = 8.12°$　　(b) C_p at $\alpha = 21.29°$

Fig. 4: Comparison of the pressure coefficient at the body surface. SU2 results (—) and experimental data (•).

since a fully turbulent approach was adopted, while the experiments were conducted in free transition. Moreover, computed pressure distributions (C_p) at the body surface for $\alpha = 8.12°$ and $\alpha = 21.29°$ are compared with the experiments in Fig. 4. A satisfactory match between both is observed.

Summing up, it can be concluded that the CFD solver is capable of predicting the airfoil performance, although there are some variations between numerical and experimental results.

E. Flap typologies generation

Several multi-element airfoil configurations must be considered as candidates for the HLD optimisation. Therefore, the 30P-30N airfoil was set as the reference baseline to design four different flap configurations, in particular, three different types of double slotted flap and one triple slotted flap.

(a) Double slotted flap type 1 (DS_1) configuration (—).

(b) Double slotted flap type 2 (DS_2) configuration (—).

(c) Double slotted flap type 3 (DS_3) configuration (—).

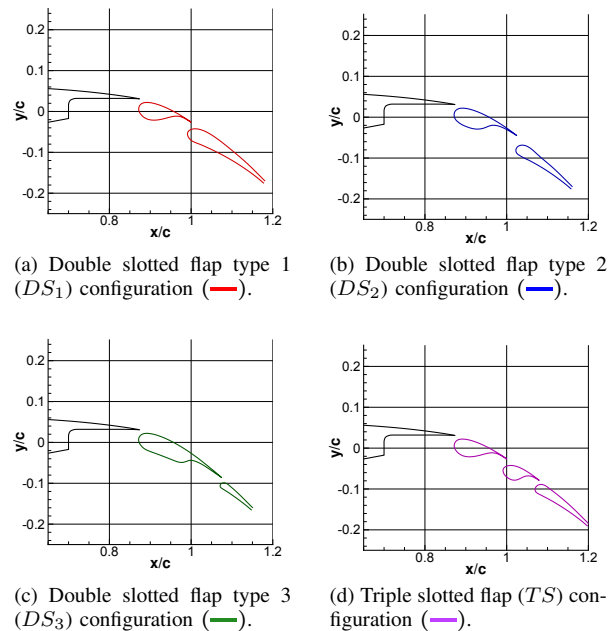(d) Triple slotted flap (TS) configuration (—).

Fig. 5: Illustration of all the configurations considered in the optimisation.

For splitting the single flap of the baseline airfoil in several elements, the multi-element airfoil manipulation program AIRSET [20] was used. Once the new elements were generated, XFOIL [21] was employed for smoothing the splitting surfaces.

Thus, during the HLD optimisation, five different HLD candidates will be studied: a single flap configuration (*SF*), which is the 30P-30N airfoil, three types of double-slotted flap (DS_1, DS_2, and DS_3), and a triple slotted flap (*TS*) (see Fig. 5). Note that, the slat and the main body parts are the same for all the configurations.

III. SCGA

SCGA [14] is a heuristic algorithm optimisation that aims at coping with mixed-variable optimisation problems with dynamically varying search space. This algorithm aims at dealing with a class of problems as broad as possible. Notably, it can effectively manage variable-size problem formulations containing continuous, integers and nominal categorical variables.

SCGA is a general-purpose optimiser, able to deal with problems from different fields and which may present different specific characteristics. For this reason, it consists of a revised version of the popular GA [22]. In this work, the flexibility of the SCGA is leveraged to incorporate the decision about the HLDs topology in the optimisation loop. Contrary to standard GAs where the chromosomes are encoded as strings, so the variables are independent, SCGA encodes the chromosome as trees allowing the definition of dependencies between variables. This creates a hierarchy between variables. For instance, the value of one variable can determine the presence or absence of other variables, or even their permissible values. This fact is particularly useful in the cases where the optimiser is controlling a configurational decision where each possibility requires the specification of different additional parameters. In the presented HLD optimisation, the variable selecting the type of flap impacts on the number of variables needed to specify the position and rotation of all the elements of the flap. In light of these considerations, it is clear that standard genetic operators cannot be adopted. SCGA, makes use of genetic operators revised to handle hierarchical problem definitions. The customised crossover and mutation operators not only permit meaningful transformations that respect the interdependence between variables, but permit to take advantage of insights the user can have about the problem [11], [13], [14]. The crossover operation consists of swapping genes between two different chromosomes (parents) to produce two new candidates (children). This operation aims at merging the information contained in the parents into the children. In this way, hopefully, the children inherit the characteristics that originated the performance of their parents and, combining them, they can reach even better performances. The crossover implementation in SCGA takes advantage of the hierarchy operating not only on the selected variables but also on all the dependent variables. This operation is necessary for creating meaningful solutions and helps to preserve the

overall information encapsulated by the selected variables and the associated ones [11]. The mutation operator aims at avoiding premature convergence and increase the diversity in the population. The variables that undergo mutation are perturbed from their current state. The strength of the perturbation determines the quantity of randomness introduced and it should be varied during the optimisation considering a variety of factors, i.e., the diversity in the population and the type of variable [23], [24]. However, this is strongly problem dependent and difficult to foresee without an in-depth knowledge of the problem. The mutation operator employed in SCGA implements a self-adaptive step size that aims to adjust the strength of the perturbations autonomously [14]. The operation differs significantly depending on the type of the variable undergoing the mutation. In case of continuous or integer variables, the operator perturbs the value to change with a small perturbation. A normal distribution generates the perturbation for the continuous variables, while the difference of two geometrical distributions is used for the integer ones [25]. In the case of nominal categorical variables, since all the possibilities are equally likely, the mutation operator simply substitutes the current value with another one. Particularly, this is re-sampled (uniform randomly) from the set of possible values deprived of the current assumed (for details, please refer to [25]).

IV. SCGA VALIDATION

The algorithm has been tested on a preliminary *'toy'* problem to validate SCGA capabilities when solving airfoil optimisation problems. Indeed, although the potential of SCGA was shown in [11], [13], [14], it has never been applied to problems with characteristics similar to those of HLD design. In particular, the purpose of this validation is to check the capacity of finely tuning continuous variables. This prerequisite is inescapable for the problem object of this study, where the continuous variables are predominant.

Another goal of this preliminary study is to test the robustness of the optimiser to different settings. Due to computational budget limitations, only one run of SCGA for the mixed-variable optimisation is possible. Consequently, an accurate algorithm parameter tuning is out of the reach of this study. Nevertheless, some assessment work on a similar problem is necessary to check the real performance of the algorithm.

A. Problem description

The *'toy'* problem is the deterministic aerodynamic design optimisation problem of an airfoil in incompressible flow conditions and subject to geometric and aerodynamic constraints. The objective is the improvement of airfoil performance, minimise drag in this case, by modifying its shape. The baseline airfoil is the NACA 2412 and the working conditions are Mach = 0 and Reynolds = 5×10^5 at a fixed lift coefficient $C_l = 0.5$. To obtain a fixed C_l, the angle of attack is adjusted.

The airfoil parametrisation is made by a linear combination of an initial geometry $(x_0(s), y_0(s))$, and modification functions $y_i(s)$. The airfoil shape is given by Eq. (1). The shape

is controlled by the design parameters w_i and the scale factor k used to scale the airfoil to the required thickness.

$$x(s) = x_0(s), \qquad y(s) = k\left(y_0(s) + \sum_{t=1}^{n} w_i y_i\right) \qquad (1)$$

In this optimisation problem the number of design parameters is $n = 20$ in the range $w_i \in [-5, 5]$. Moreover, the maximum thickness (t) of the airfoil is fixed to 12% of the chord (c).

The quantity of interest J to be optimised is the drag coefficient, C_d. The airfoil optimisation is subjected to some geometrical and aerodynamic constraints. Specifically, the constraints are: trailing edge angle (TEA) greater than or equal to $13°$, leading edge radius (LER) greater than or equal to 0.7% of the chord, and the boundary layer transition point on the lower surface (XTR_{LOW}) of the airfoil cannot take place at x/c greater than 0.95. Furthermore, an $Error$ flag, if equal to 1, will indicate the no convergence of the solver. Bearing this in mind, the objective function is given by

$$J = \begin{cases} C_d & \text{if } Error = 0 \text{ \& fulfilled constraints} \\ p\ C_d & \text{if } Error = 0 \text{ \& no fulfilled constraints} \\ p & \text{if } Error = 1, \end{cases} \qquad (2)$$

with $p = 1000$. Thus, the deterministic problem reads:

$$\min J . \qquad (3)$$

The drag coefficient is computed using XFOIL aerodynamic analysis code [21]. It couples, in a fully interactive way, a high-order panel method with an integral boundary layer code.

B. Experiment

Five different settings of SCGA have been tested on the *'toy'* problem. The parameters varied are the *population size* and the *tournament size*. Increasing the former helps to keep diversity in the population and preventing from premature convergence. The latter is the size of the pool of candidates tested during a selection operation. This size heavily impacts on the diversity in the mating pool for the *crossover* operator [22]. The smaller the *tournament size* is, the higher the chances for a low-fitness candidate to bequeath its information to candidates of the next population. Therefore, the higher the *tournament size*, the higher the pressure to select only the high-fitness candidates, with the possible drawback of letting the population collapse toward a local optimum. The values tested for the *population size* are $[5, 10, 20]$ and for the *tournament size* are $[2, 4]$. However, the combination with *population size* equal to 5 and *tournament size* equal to 4 cannot be run because of the implementation of the tournament selection in SCGA. The number of maximum function evaluations has been set to 7000 to match the ones used in the reference run. The remaining parameters of SCGA have been left to the default values [12].

The reference run is made using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [26] available in the Adaptive Genetic algorithm (ADG) optimisation library[2]. This

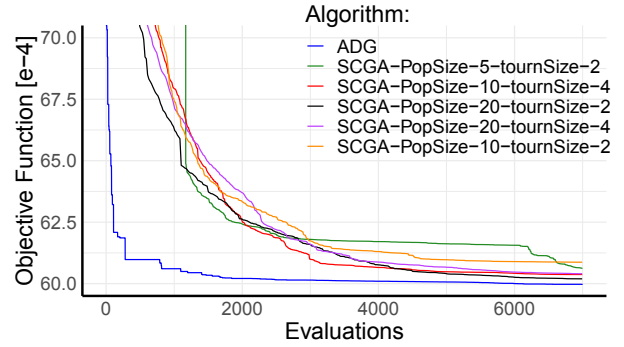[2]Optimiser developed at Centro Italiano Ricerche Aerospaziali (CIRA).

Fig. 6: Best found solution history. All the instances are compared to the ADG run.

library has been already tested for a bunch of aerodynamic design optimisation problems facing multi-element airfoils [5], [6]. It is a stochastic optimisation algorithm based on self-adaptation of the covariance matrix of a multivariate normal distribution. It is mainly used for design optimisation problems up to a few hundreds of design variables. The parameters for this optimisation algorithm are the maximum number of allowed evaluations, equal to 7000, the population size ($\lambda = 4$), and the initial standard deviation ($\sigma = 0.1$). For each setting of SCGA, 50 independent runs have been performed to have statistically significant results. All the tested instances perform adequately well as shown in Fig. 6 and are resumed in Table I. Particularly, the values of the best found solution of each instance of SCGA are compared in terms of relative difference between the minimum, the mean and the maximum value found in the 50 independent runs with respect to the reference value. Considering the differences between the two classes of optimisers and that the initial population of the ADG was built mutating the solution representing the nominal airfoil, the output of the validation can be considered as positive.

TABLE I: Results of the optimisation runs.

Population Size	Tournament Size	Min diff [%]	Mean diff [%]	Max diff [%]
5	2	0,17	1,08	7,44
10	2	0,21	0,66	1,50
10	4	0,33	1,07	3,00
20	2	0,14	0,72	1,68
20	4	0,12	1,84	20,9

V. PROBLEM FORMULATION

One of the many challenges of mixed-discrete variable-size optimisation problems is to implement the appropriate mapping between the design variables of the problem. This mapping is described in Section V-A. After finding a solution for a proper connection between the variables, the High-Lift Device optimisation design problem can be set up (Section V-B).

A. Optimisation design variables

The HLD optimisation degrees of freedom are the type of flap and the settings of each flap component (position in the 2-dimensional space and rotation). The settings of the slat

are also considered. The optimiser receives this information encoded as a set of design variables. The SCGA allows the definition of hierarchical problem formulations. In this problem, the top of the hierarchy is composed by four independent variables: one for indicating the flap typology (SF, DS_1, DS_2, DS_3 and TS) and three variables ($\Delta\theta_S, \Delta X_S, \Delta Y_S$) indicating the settings of the slat. Specifically, they indicate the deviance of a proposed position from the nominal one. The second level of the hierarchy is composed of all the variables determining the flap's settings. They are all dependent on the flap typology variable, and its value determines their presence. They are three in the case of the single flap, six in the case of the double-slotted flap (whatever the type), and nine in the case of the triple slotted flap. A graphical representation of the variables hierarchy is shown in Fig. 7. The description of each variable, its number for identifying in the hierarchy, its type, the possible value that it can assume (as possibility or bounds), and the dependency of any variables to it are given

(a) From double slotted type 1 (—) to triple slotted.

(b) Translation of one element. Initial position (—).

Fig. 8: Examples of the mutation operator adopting the *naive* (- - -) and the proposed (—) formulations.
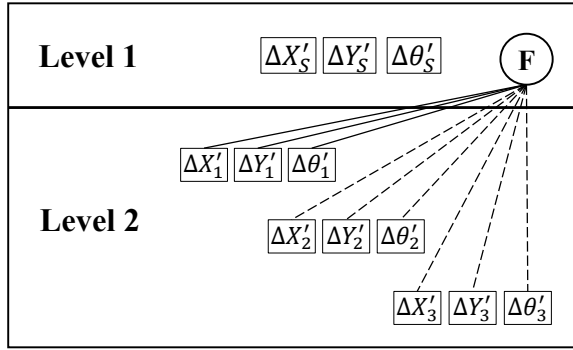

Fig. 7: Variables hierarchy. Level 1 is constituted by the independent variables (referring to Table II, variables 1-4). Level 2 is made by all the remaining variables dependent by variable 1. Solid lines indicate variables that are present in all the candidates. Whereas, dashed lines indicate variables that can also not be present.

Concerning the encoding of the variables indicating the settings of the flap elements, the most straightforward and

naive problem formulation would be to use the same approach used for encoding the settings of the slat. However, this would imply significant difficulties from the optimisation perspective. For example, let us suppose that one candidate undergoes the mutation operator and only the variable defining the type of the flap is mutated. A desirable outcome would be an HLD with different elements but similar overall topology. However, the example of this transformation is given in Fig. 8a. Here, two possible outcomes of the mutation operating on the flap type of a solution adopting a double slotted DS_2 flap are shown. As one can see, by using *naive* problem formulation, the mutation of one single variable would result in an unwanted considerable change in the HLD overall topology. The resulting topology would even be unfeasible. Another example is depicted in Fig. 8b. In this case, the variable determining the x-coordinate of the first flap element is mutated. If the *naive* problem formulation is used, the second component of the resulting flap completely loses its alignment with the first component. Therefore, the *naive* problem formulation may lead very often to undesirable outcomes because a small variable perturbation can induce considerable topology transformations.

In light of these considerations, another problem formulation has been proposed. Its backbone is to make the flap elements be logically connected to the preceding elements. The absolute difference between the proposed solutions and the nominal solutions is described as follows:

$$\boldsymbol{\Delta X} = \begin{cases} \Delta X_S = \Delta X_S' \\ \Delta Y_S = \Delta Y_S' \\ \Delta\theta_S = \Delta\theta_S' \\ \begin{cases} \Delta X_i = \Delta X_i' \\ \Delta Y_i = \Delta Y_i' \qquad\qquad \text{if } i = 1 \\ \Delta\theta_i = \Delta\theta_i' \end{cases} \\ \begin{cases} \Delta X_i = \Delta X_i' + \Delta X_{i-1}' \\ \qquad + RX_{i-1} - X_i^{LE} \\ \Delta Y_i = \Delta Y_i' + \Delta Y_{i-1}' \qquad \text{if } i \neq 1. \\ \qquad + RY_{i-1} - Y_i^{LE} \\ \Delta\theta_i = \Delta\theta_{i-1}' + \Delta\theta_i' \end{cases} \end{cases} \quad (4)$$

TABLE II: Problem formulation.

Description	N	Variable type	Possibilities		Dep
Flap type	1	Nominal discrete	$[SF, DS_1, DS_2, DS_3, TS]$		$[5-13]$
Description	N	Variable type	Lower Bound	Upper Bound	Dep
$\Delta\theta_S'$	2	Continuous	−15	15	–
$\Delta X_S'$	3	Continuous	−0.1	0.1	–
$\Delta Y_S'$	4	Continuous	−0.025	0.025	–
$\Delta\theta_1'$	5	Continuous	−15	15	–
$\Delta X_1'$	6	Continuous	−0.2	0.2	–
$\Delta Y_1'$	7	Continuous	−0.05	0.05	–
$\Delta\theta_2'$	8	Continuous	−15	15	–
$\Delta X_2'$	9	Continuous	−0.1	0.1	–
$\Delta Y_2'$	10	Continuous	−0.025	0.025	–
$\Delta\theta_3'$	11	Continuous	−15	15	–
$\Delta X_3'$	12	Continuous	−0.1	0.1	–
$\Delta Y_3'$	13	Continuous	−0.025	0.025	–

Where RX_{i-1} and RY_{i-1} are the translation due to the rotation of the previous element:

$$\begin{bmatrix} RX_{i-1} \\ RY_{i-1} \end{bmatrix} = \begin{bmatrix} X_{i-1}^{LE} \\ Y_{i-1}^{LE} \end{bmatrix} + \begin{bmatrix} \cos(\Delta\theta_{i-1}') & -\sin(\Delta\theta_{i-1}') \\ \sin(\Delta\theta_{i-1}') & \cos(\Delta\theta_{i-1}') \end{bmatrix} \begin{bmatrix} d_X \\ d_Y \end{bmatrix} \quad (5)$$

and d_X and d_Y are the distance between the leading edges of the i-th and $i-1$-th elements along the two axes:

$$\begin{aligned} d_X &= X_{i-1}^{LE} - X_i^{LE} \\ d_Y &= Y_{i-1}^{LE} - Y_i^{LE}. \end{aligned} \quad (6)$$

From Fig. 8 the impact of this new problem formulation on the mutation operations can be appreciated. In the example shown in Fig. 8a, if the proposed formulation is used, the resulting airfoil presents a fairly similar overall settings to the original one. Therefore, the changes are only due to the different type of flap adopted. In the example shown in Fig. 8b, the relative alignment between the flap's elements is preserved. The change in the topology is accompanied by a translation along the x-axis of the HLD as a whole.

B. High-Lift Device optimisation problem

A deterministic mixed variable-size aerodynamic design optimisation problem of a multi-element airfoil is solved. The baseline configurations are those shown in Section II-E and the working conditions are Mach = 0.2 and Reynolds = 5×10^6 at a fixed angle of attack $\alpha = 21.29°$. The goal of the optimisation is the improvement of the multi-element airfoil performance, maximise lift coefficient, by selecting the most suitable flap typology and the settings (position and rotation) for the flap elements, as for the slat.

The quantity of interest to be optimised is the lift coefficient, C_l. However, the failure of the grid generation or the non-convergence of the CFD solver have to be treated. This latter is considered to happen when the difference between the up-to-date C_l and the mean lift coefficient in the last 1000 iterations of the CFD solver ($C_{l_{AVG}}$) is lower or equal to 0.005 ($|C_l - C_{l_{AVG}}| \leq 0.005$). CFD analysis of configurations close to stall or even post-stall is notoriously a complex task, and the results present a high margin of uncertainty even in the case of fully converged solutions. Consequently, it is not wise and appropriate to use CFD results that have not reached full convergence because they would introduce an unacceptable margin of uncertainty and imprecision. The aerodynamic computational chain returns an indication of the occurrence of one of these errors as an error flag. This flag is 1 in case of error, 0 in case of a successful evaluation. Only successful candidates can return a correct prediction of C_l. Therefore, the objective function is reformulated by introducing a step penalty as follows:

$$Q(\mathbf{\Delta X}) = \begin{cases} C_l & \text{if } Error = 0 \\ p & \text{if } Error = 1, \end{cases} \quad (7)$$

with $p = -1000$ (to be noted that it is a maximisation problem and values ~ 1 are expected). Hence, the optimisation problem reads:

$$\max Q(\mathbf{\Delta X}) \quad (8)$$

Finally, the used optimiser algorithm is SCGA. A list of the algorithm parameters is given in Table III. In Table III *size* is the population size, *mutRate* is the mutation rate, and *probability* is the probability of a design variable to be selected by the operators. The remaining parameters of SCGA have been left as default [12]. The optimisation stopped when the available computational budget was exhausted. In particular, this happened after 86 iterations.

TABLE III: SCGA parameters

size	tournamentSize	maxEvaluations	elitism	mutRate	probability
19	3	1550	1	0.05	[3,1,...,1]/16

VI. RESULTS

In this section, the most notable results are analysed. Firstly, the convergence history of the best solution found is commented. From Fig. 9, it can be seen that, starting from a randomly generated population, only 355 evaluations were needed to find an HLD topology which performs better than the 30P-30N airfoil, and 505 to improve DS_1 baseline airfoil. The optimum airfoil is a double-slotted of type 1 (DS_1). Specifically, the airfoil type DS_1 prevailed as the best configuration after a short initial phase of exploration. The evolution of the presence of the different flap types in the populations is represented in Fig. 10. Here, it is shown that all the configurations are fairly equally represented in the initial population. Later, the good performance of candidates having the flap type DS_1 makes that a significant part of the candidates assumes this airfoil type. However, the other configurations never disappear, thus the effect of the configurational variable is investigated during all the optimisation. Moreover, it must be specified that 41% of the evaluations returned $Error = 1$. The majority of these cases were due to the no convergence of the CFD analysis. This behaviour is typical when studying near stall conditions. Although, the failure rate was high, the optimiser was able to find a configuration improving the airfoil performance at the studied working conditions. This evidences the potential of SCGA.

In addition, Fig. 11 shows the history of each variable of the best element in each population. If variables $\Delta X_3'$, $\Delta X_3'$
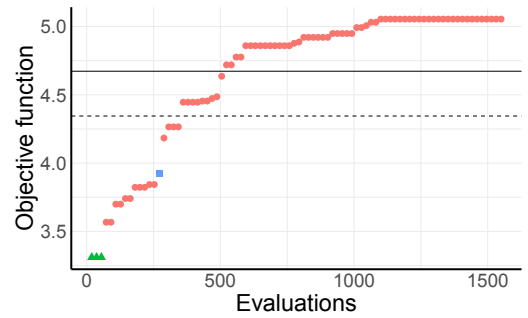

Fig. 9: Best found objective function history. Airfoil types: DS_1 (●), TS (■), and DS_3 (▲). Lift coefficient of the baseline airfoil of type DS_1 (—) and C_l of the 30P-30N airfoil (- - -).
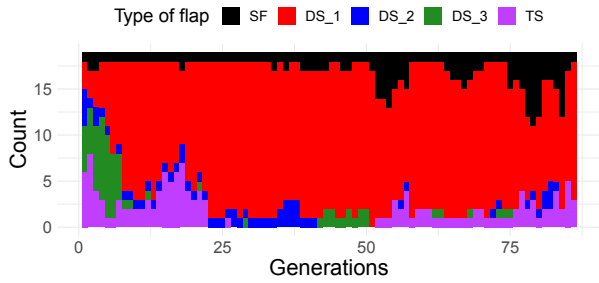
Fig. 10: Presence of candidates with different airfoil typology in every generation.

or $\Delta\theta_3^{'}$ are missing means that the best solution is not a triple slotted configuration.
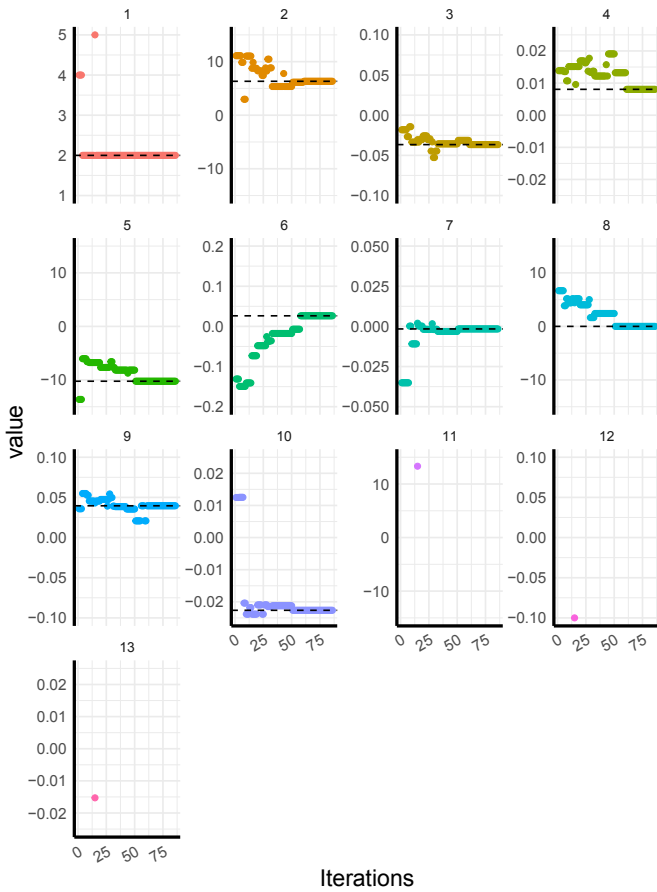

Fig. 11: Best found history. Variables enumeration follow the one given in Table II.

Moreover, it is appreciated that part of the optimum values were rapidly identified, others, i.e. $\Delta X_1^{'}$ and $\Delta\theta_2^{'}$, changed significantly during the whole optimisation.

The optimal multi-element airfoil is depicted in Fig. 12 and is compared with the baseline double-slotted airfoil, of type one (DS_1), and the starting airfoil, the 30P-30N airfoil. The optimised airfoil presents an 8.2% increase in C_l com-
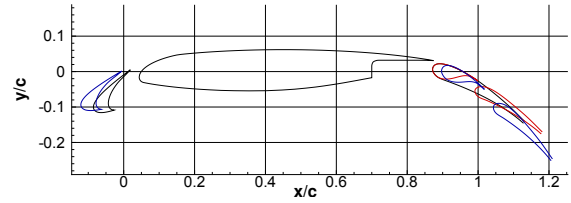

Fig. 12: Airfoil comparison between the 30P-30N (—), the baseline airfoil of the double slotted flap of type one (DS_1) (—), and the optimal airfoil (—).

pared to the first, and a 16.3% to the second, considering the working conditions of the optimisation design problem. Moreover, the aerodynamic performance of the optimal airfoil must be contrasted to the baseline airfoils aforementioned. Thus, the polars of the three airfoils are given in Fig. 13a and Fig. 13b. The comparison among the three configurations shows a clear improvement, in terms of lift coefficient, for the optimal configuration. Also, the optimised airfoil reaches the stall condition at a lower angle of attack. Notably, the optimal airfoil shows an abrupt stall type. This behaviour could compromise the aircraft performances in landing conditions. The landing design condition should, therefore, be explicitly introduced in the optimisation problem to overcome this problem. Indeed, the present work is mainly about the proposed optimisation methodology, while this additional condition will be considered in further developments using a multi-objective or a multi-point optimisation approach.
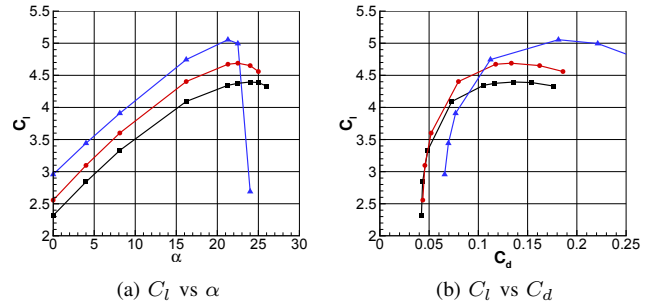

(a) C_l vs α (b) C_l vs C_d

Fig. 13: Comparison of the aerodynamic performance between the 30P-30N airfoil (—■—), the baseline DS_1 airfoil (—●—), and the optimal configuration (—▲—).

Finally, in Fig. 13b, it is appreciated that the drag coefficient is penalised. However, this was an expected result because this performance indicator has not been considered in objective function.

VII. Conclusions and Future Work

This paper presents a novel method to improve classical design strategies by including configurational decisions in an automated optimisation tool. The integration of these decisions, usually taken a-priori based on previous knowledge, could represent a step forward for the multidisciplinary design optimisation field. The potential of the method has been shown in an High-Lift Device design problem, an application that

presents many difficulties itself. Among the others, the considerable computational cost and the demanding constraints of the design problem represented a challenge. Furthermore, it was quite challenging to obtain an adequately converged Computational Fluid Dynamic solution for the majority of population elements due to the near-stall conditions at which the airfoil had to operate. An automatic estimation routine consisting of an aerodynamic computational chain based on the SU2 solver has been adopted to cope with these problems. This procedure has been coupled with the Structured-Chromosome Genetic Algorithm for optimising the design of an HLD equipped with slat and flap devices based on the McDonnell Douglas 30P-30N airfoil. In particular, the goal of the optimisation was the identification of the typology, for the multi-element airfoil, which maximises the lift coefficient. The results of the optimisation show that, given an appropriated problem formulation, the optimiser was able to improve the reference performance by 16.3% in a very limited number of function evaluations. Nevertheless, the optimiser well performed, although a significant amount of failed evaluations.

As future outlook, many improvements can be made. As an optimisation standpoint, different more sophisticated constraint handling techniques can be adopted. Additionally, better tuning of CFD analysis stopping and convergence criteria may lead to an increasing number of successful evaluations. In terms of problem formulation, also the decision about which devices (flap and slat) to employ might be delegated to the optimiser. Moreover, the variables describing the settings of each element might be gathered as in [13] to take advantage of the characteristics of the SCGA's operators. As a further step, the shape of each element might be part of the design problem. This would lead to a significant increase in the number of variables and, consequently, of the problem complexity. Furthermore, the optimisation could consider a higher number of performance indicators as the weight or the structural complexity. Besides, a control on the drag coefficient can be added to satisfy the constraints during take-off and climbing phases. Finally, for a more rigorous assessment of the advantages offered by the presented approach, comparisons with traditional design strategies and optimisers need to be done.

ACKNOWLEDGEMENT

REFERENCES

[1] R. Rudnik and H. Geyr, "The european high lift project eurolift ii-objectives, approach, and structure," in *25th AIAA applied aerodynamics conference*, p. 4296, 2007.

[2] C. Van Dam, "The aerodynamic design of multi-element high-lift systems for transport airplanes," *Progress in Aerospace Sciences*, vol. 38, no. 2, pp. 101–144, 2002.

[3] A. M. O. Smith, "High-lift aerodynamics," *Journal of Aircraft*, vol. 12, no. 6, pp. 501–530, 1975.

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

[5] P. Iannelli, F. Moens, M. Minervino, R. Ponza, and E. Benini, "Comparison of optimization strategies for high-lift design," *Journal of Aircraft*, vol. 54, no. 2, pp. 642–658, 2017.

[6] P. Iannelli and D. Quagliarella, "Multi-objective/multi-point shape and setting high-lift system optimization by means of genetic algorithm and 2D Navier-Stokes equations," in *EUROGEN 2011 Conference proceedings, Capua, Italy*, 2011.

[7] F. Moens and C. Wervaecke, "Multi-point optimization of shapes and settings of high-lift system by means of evolutionary algorithm and Navier-Stokes equations," *Engineering Computations*, 2013.

[8] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin, "How to deal with mixed-variable optimization problems: An overview of algorithms and formulations," in *World Congress of Structural and Multidisciplinary Optimisation*, pp. 64–82, Springer, 2017.

[9] H. M. Nyew, O. Abdelkhalik, and N. Onder, "Structured-chromosome evolutionary algorithms for variable-size autonomous interplanetary trajectory planning optimization," *Journal of Aerospace Information Systems*, vol. 12, no. 3, pp. 314–328, 2015.

[10] S. A. Darani and O. Abdelkhalik, "Space trajectory optimization using hidden genes genetic algorithms," *Journal of Spacecraft and Rockets*, vol. 55, no. 3, pp. 764–774, 2017.

[11] C. Greco, L. Gentile, G. Filippi, E. Minisci, M. Vasile, and T. Bartz-Beielstein, "Autonomous generation of observation schedules for tracking satellites with structured-chromosome GA optimisation," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 497–505, IEEE, 2019.

[12] L. Gentile, "LorenzoGentile/SCGA: SCGA second release. Update." https://doi.org/10.5281/zenodo.3627555, Jan. 2020.

[13] L. Gentile, C. Greco, E. Minisci, T. Bartz-Beielstein, and M. Vasile, "An optimization approach for designing optimal tracking campaigns for low-resources deep-space missions," in *70th International Astronautical Congress*, 2019.

[14] L. Gentile, C. Greco, E. Minisci, T. Bartz-Beielstein, and M. Vasile, "Structured-chromosome GA optimisation for satellite tracking," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1955–1963, 2019.

[15] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[16] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An open-source suite for multiphysics simulation and design," *AIAA Journal*, vol. 54, no. 3, pp. 828–846, 2016.

[17] P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th aerospace sciences meeting and exhibit*, p. 439, 1992.

[18] V. Chin, D. Peters, F. Spaid, and R. Mcghee, "Flowfield measurements about a multi-element airfoil at high Reynolds numbers," in *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, p. 3137, 1993.

[19] S. M. Klausmeyer and J. C. Lin, "Comparative results from a CFD challenge over a 2D three-element high-lift airfoil," 1997.

[20] M. Drela, "A user's guide to MSES 3.05," *Massachusetts Institute of Technology (MIT), Cambridge*, 2007.

[21] M. Drela and H. Youngren, *XFOIL 6.94 User Guide*. MIT Aero & Astro, Dec. 2001.

[22] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[23] G. Papa, "Parameter-less algorithm for evolutionary-based optimization," *Computational Optimization and Applications*, vol. 56, no. 1, pp. 209–229, 2013.

[24] I. Korejo and S. Yang, "Comparative study of adaptive mutation operators for genetic algorithms," in *in MIC 2009: The VIII Metaheuristics International Conference, 2009. 196 Journal of Emerging Technology and Advanced Engineering Website: www. ijetae. com (ISSN 2250-2459, Volume 2, Issue 2*, Citeseer, 2012.

[25] R. Li, M. T. Emmerich, J. Eggermont, T. Bäck, M. Schütz, J. Dijkstra, and J. H. Reiber, "Mixed integer evolution strategies for parameter optimization," *Evolutionary computation*, vol. 21, no. 1, pp. 29–64, 2013.

[26] N. Hansen, "The CMA Evolution Strategy: A comparing review," in *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms* (J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, eds.), pp. 75–102, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.