Single-point optimization of the LS89 turbine cascade using a hybrid algorithm

Arnaud Châtel

Turbomachinery and Propulsion Department, von Karman Institute for Fluid Dynamics, Belgium, arnaud.chatel@vki.ac.be

Supervisor: Tom Verstraete

Associate Professor, Turbomachinery and Propulsion Department, von Karman Institute for Fluid Dynamics, Belgium, tom.verstraete@vki.ac.be

University Supervisor: Grégory Coussement

Professor, Fluids-Machines Unit, University of Mons, Belgium, gregory.coussement@umons.ac.be

Abstract

This paper presents a hybrid optimization algorithm which combines the two main families of optimization methods namely an evolutionary algorithm and a gradient-based method. The better performance of the developed hybrid method, compared to a classical evolutionary algorithm, is first demonstrated on two analitycal functions. Then, the algorithm is used to optimize LS89 turbine cascade, resulting in a new design with about 20 percent lower entropy production. A thorough flow analysis shows that the improvements are largely due to a significant decrease in trailing edge losses.

Keywords: aerodynamic optimization, hybrid algorithm, evolutionary algorithm, gradient-based method, profile losses

1. Introduction

Nowadays, optimization methods are integrated into the design process in various industrial sectors, as they have proven to reduce the design cost and time while at the same time improving product quality [1; 2; 3; 4]. Due to this, they are increasingly used in more elaborate and complex design tasks, which require more efficient and versatile optimization algorithms. Since these optimizations require more computational resources and can be very time-consuming, the improvement of the optimization methods themselves becomes a crucial challenge. This improvement is particularly relevant when the performance of the design is quantified using advanced computational methods. This is the case in aerodynamic design processes where Computational Fluid Dynamics (CFD) are used to predict the flow performance.

Currently, two main families of optimization methods can be distinguished depending on the order of the derivative of the objective function used. Gradient-free methods use only the value of the objective function, while gradient-based methods require additionally the computation of gradient values in order to perform the optimization [5; 6]. Each of these two families has its own advantages and disadvantages. The most prevalent gradient-free methods in the field of shape optimization are evolutionary algorithms which are based on Darwinian evolution [5]. They allow a global exploration of the complete design space using a population of individuals which evolve in the search space. However, because of the so-called ”curse of dimensionality” [7], the convergence performance of these algorithms becomes relatively poor when the number of design variables increases. Even though gradient-free methods can rapidly identify the region of the design space where the global minimum is located, the convergence towards this minimum can be very long [8; 9]. On the contrary, gradient-based methods concentrate

the optimum search in a direction defined by means of the gradients of the objective function. When combined with the adjoint method to compute the gradients efficiently, these methods tend to converge more rapidly and do not suffer to the same extend from the curse of dimensionality [10]. However this family of methods performs a local exploration of the design space and tends to be trapped in local minima.

In this context, hybrid optimization algorithms, which combines both families of methods, benefit from the respective strengths of gradient-free and gradient-based methods. Therefore, the hybrid algorithms have better convergence performance towards the global optimum of the design space. A review of different kinds of hybrid optimization algorithms can be found in [11]. Most of the methods described in the paper are applied on job-shop scheduling problems [12; 13]. In the engineering field, the common hybridization strategy consists in the incorporation of a gradient-based method into the gradient-free method [11]. In this strategy, two different approaches can be distinguished in the literature. First, the gradient-based method can be integrated directly in the loop of the evolutionary algorithm in order to locally improve individuals of each population. This approach is generally called Lamarckian approach or memetic algorithm [14; 9; 15; 8]. In the second approach, the gradient-free method performs a certain number of generations and then gradient-based methods are applied on the most promising individuals of the last population [16].

In this work, we propose an hybrid algorithm which combines an evolutionary algorithm called Differential Evolution [17] and the classical steepest descent method [5]. The hybridization is based on the Lamarckian evolution briefly mentioned above. This hybrid algorithm is tested on the well-known LS89 turbine cascade [18]. A single point optimization of the geometry profile is performed in order to minimize the entropy generated by the flow through the cascade.

This paper is structured as follows: the first section is dedicated to the hybrid optimization algorithm which is extensively presented. Then the proposed method is tested and validated on two analytical functions specially designed to test the performance of optimization algorithms. The third section describes the LS89 application, the geometry parametrization, and the aerodynamic performance parameters used in the optimization process. At the end of this section, the results of the hybrid optimization are compared to the results obtained in a previous study [19] in which the LS89 profile was optimized using a classical gradient-based method. Finally, in the last section, the optimized profile found by the hybrid algorithm is compared to the LS89 profile in order to understand the mechanisms behind the improved performance.

2. Hybrid optimization algorithm

The first part of this section describes the basic principle of the proposed hybrid optimization algorithm. Then, in order to explain the methodology, three different aspects of the hybrid method are discussed namely the position of the hybridization, the selection of the individuals for local improvements and the imposition of constraints.

2.1. Basic principle

The aim of the implemented hybrid optimization algorithm is to combine efficiently a gradient-free and a gradient-based method. This combination should benefit from the respective strengths of both methods, namely the faster convergence of the gradient-based method and the capacity to locate the global optimum of the gradient-free method. The chosen methodology is based on the Lamarckian evolution [8; 15] and consists in incorporating directly the gradient-based method inside the gradient-free algorithm.

In this work, the hybrid algorithm uses the Differential Evolution (DE) algorithm developed by Price and Storn [17] as the gradient-free method and the steepest descent associated with line search method [20; 21] as the gradient-based search. The ideal step length in the descent direction is computed by a line search method based on the Wolfe conditions [21]. The integration of the gradient-based method inside the loop of the evolutionary algorithm is presented in Fig. 1.

The hybrid algorithm starts like the classical DE algorithm with a population of random individuals which are evaluated and reproduced in order to create a new population. The reproduction is composed by two mechanisms called mutation and crossover as described in [17]. Then, the hybridization of the DE algorithm occurs within the generated new population. In this population, the most promising individuals are chosen to be locally improved using

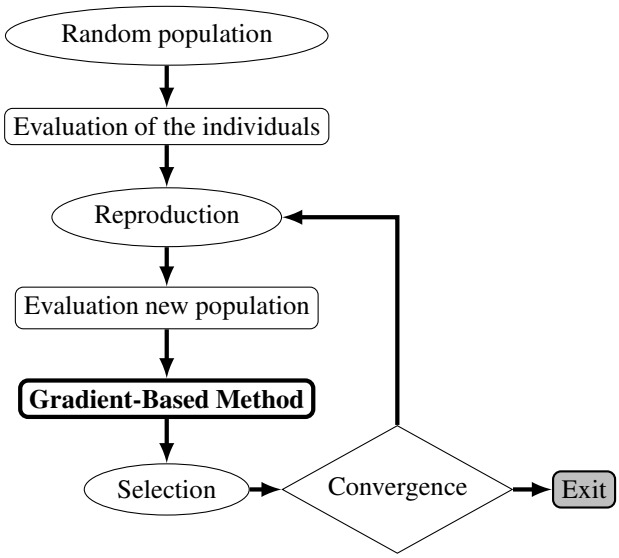

Figure 1: Hybrid Algorithm combining the DE algorithm and the steepest descent method.

the gradient-based method regarding them as starting points for the steepest descent algorithm. Therefore, the gradient values of the objective function with respect to the design parameters are computed only for these promising individuals. For each individual selected for a local search, only one iteration of the steepest descent method is performed because there is no need to bring the individuals exactly to their minimum which could be just local [14; 8]. Continuing the steepest descent method until convergence of each individual can generate a prohibitive computational cost just to locate local minima. Once the selected individuals have been locally improved, they are reintegrated in their original population and the *selection* between this population and the previous one is carried out. Finally, if the stopping criterion are not respected, the evolutionary process restarts.

Hereinafter, three different points of the hybrid algorithm are discussed: the position of the hybridization in the DE algorithm, the selection of the individuals which are locally improved and the imposition of the constraint in the hybrid algorithm.

2.2. Position of the hybridization

The gradient-based methods are always performed before the *selection* of the DE algorithm because it allows to conserve the diversity introduced by the evolutionary algorithm. Indeed, in this situation, the steepest descent algorithm is applied to new created individuals at each generation. In contrast, if the local improvement is performed after the *selection*, an individual with a good potential can be replaced during the *selection* by another on which local searches have already been realized in the previous generations. In that case, the gradient-based method is always applied to the same individuals and, therefore, the algorithm can get stuck in a local minimum during more generations.

2.3. Selection of the individuals for local improvement

The most promising individuals of the new populations are chosen by assigning a probability to be locally improved. In a population, the individual with the lowest value of the objective function has the highest probability value equal to 1. The probability p_i of the other individuals of the population is computed relatively to the objective function of the best individual $f_{obj}|_{BestInd}$ and can be expressed by:

$$p_i = \frac{\frac{1}{f_{obj}|_i}}{\frac{1}{f_{obj}|_{BestInd}}}. \quad (1)$$

Finally, for each new generation, gradient-based methods are applied on all the individuals of the population which have a probability above a certain limit p_{limit} defined beforehand by the user.

2.4. Imposition of constraints

In many optimizations, the aim is to identify a design $\vec{X}$ which minimizes the objective function $f(\vec{X})$ while respecting certain constraints $g(\vec{X})$. Therefore the problem is usually expressed as follows:

$$\text{Minimize:} \quad f(\vec{X}) \quad (2)$$

$$\text{Subject to:} \quad g(\vec{X}) \leq 0. \quad (3)$$

In the DE algorithm, the constraints are imposed by eliminating the individuals which do not respect the Eq. 3 in the *selection* process. Therefore, in the hybrid method, the individuals locally improved have to respect the constraints otherwise they will be directly eliminated in the next *selection* process and the computational cost of the steepest descent method will be lost. For gradient-based methods, however, such simplified handling of the constraints is not possible and more elaborate algorithms are needed. The simplest approach consists in adding a penalty term to the objective function, which penalizes the objective when the constraint is violated [19; 21]. This exterior

penalty method converges in case of active constraints to a design which violates the constraint as a balance between the improvement of the objective and deterioration of the penalty, which is perceived as a serious deficit in the hybrid method as the design will be thrown away in the next round of the DE *selection* process. Therefore, in this work, another method is implemented which corrects the individuals with active constraint. The methodology is called the Projected Gradient method and consists in projecting the concerned individuals on the constrained limit expressed by:

$$g(\vec{X}) = 0 \tag{4}$$

The principle of the projected gradient method is presented in Fig. 2 where the grey area contains all the designs which respect the constraint (expressed by Eq. 3). During the steepest descent, the design $\overrightarrow{X_i}$ is modified in the direction $\vec{S}$ computed using the gradients and, for a certain value of the step length, a design $\overrightarrow{X_{i+1}}$ which does not respect the constraint can be obtained. In this particular case, the design $\overrightarrow{X_{i+1}}$ is projected on the limit of the constraint (expressed by Eq. 4) using the vector $\vec{\delta}$ in order to obtain a new design $\overrightarrow{X_{Cor}}$. The computation of the vector $\vec{\delta}$ starts with the linear approximation of the constraint function around the corrected design $\overrightarrow{X_{Cor}}$:

$$g(\overrightarrow{X_{Cor}}) = g(\overrightarrow{X_{i+1}} + \vec{\delta}) = g(\overrightarrow{X_{i+1}}) + \vec{\delta}.\nabla g(\overrightarrow{X_{i+1}}) = 0 \tag{5}$$

Since the aim of the method is to project the design $\overrightarrow{X_{i+1}}$ on the limit of the constraint where $g(\vec{X}) = 0$, the right hand side of Eq. 5 is equal to 0 and the vector $\vec{\delta}$ can be computed using the final formula 6. In this development, only a linear approximation of the function $g(\vec{X})$ is used, and therefore few iterations of this process could be needed in order to correct properly the design. It can be seen, in the previous development, that the projected gradient method makes use of the gradient values of the constraint (∇g) which have to be computed if a design does not respect the constraint.

$$\vec{\delta} = -\frac{g(\overrightarrow{X_{i+1}})}{\nabla g(\overrightarrow{X_{i+1}})} \tag{6}$$

The projected gradient method is used at two different places in the hybrid optimization algorithm. First, in the obvious case of an individual which does not respect the constraint during the steepest descent method. In that case the design is modified in the direction given by the projected gradient method until the constraint is satisfied. Secondly, the projected gradient method is also applied to correct new promising individuals generated by the evolutionary algorithm which eventually do not respect the constraint. These individuals are first corrected by the projection process and, then, are locally improved by the steepest descent method.

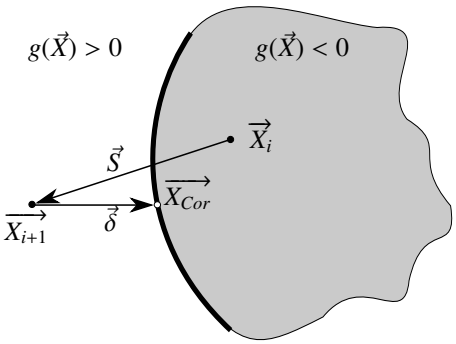

Figure 2: Principle of the projected gradient method

3. Validation on analytical functions

The proposed hybrid algorithm is first validated on two analytical functions specially designed to assess the performance of optimization methods [22]. The functions are called Rastrigin and Rosenbrock and are respectively expressed by Eq. 7 and Eq. 8. Both functions are represented in Fig. 3 and their values at the global minimum are equal to zero.

$$f_{Rastrigin} = 10 + \sum_{i=1}^{N_{Param}} [x_i^2 - 10.cos(2\pi x_i)] \tag{7}$$

$$f_{Rosenbrock} = \sum_{i=1}^{N_{Param}-1} [100.(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \tag{8}$$

As shown in Fig. 3, the Rastrigin function is highly multimodal which makes the localization of the global minimum relatively complicated. On the opposite, the Rosenbrock function is unimodal but the only minimum is located inside a parabolic and flat valley. This function is used as a benchmark problem in optimization because the valley is easy to find, while the convergence towards the global minimum inside the valley can be difficult even for

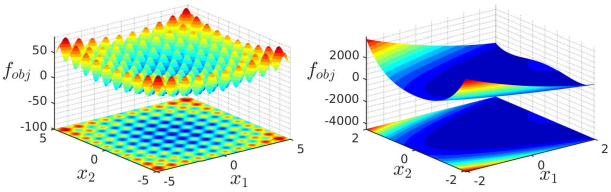

Figure 3: Rastrigin function (left) and Rosenbrock function (right) defined 2 parameters

gradient-based methods. In order to demonstrate the effect of the hybridization on optimizations conducted in larger design spaces, the algorithm is tested on the Rastrigin functions described by 2 and 5 parameters and on the Rosenbrock functions computed by means of 2 and 20 parameters.

The aim of the proposed algorithm is to localize the global minimum of the functions using a limited computational cost. Such a global exploration of the design space is not feasible using gradient-based methods and can only be performed with evolutionary algorithms. Therefore, in this section, the performance of the new hybrid method in terms of number of iterations and number of evaluations of the function is compared to the performance of the classical DE algorithm. Since evolutionary algorithms are non-deterministic methods, the performance of the two different algorithms needs to be statistically analyzed. Due to this, each optimization is performed 40 times and the averaged results are presented in Fig. 4 and in Fig. 5. In these figures, the y-axis represents the ratio between the objective of the best individual of the new populations generated f_{best} and the objective of the best individual taken in the starting population f_{Max}. Generally, the computational cost for the gradients evaluation is of the same order of magnitude than the cost for the objective function evaluation. Indeed, in industrial applications involving CFD analysis, the gradients can be obtained at a similar cost as the CFD using the adjoint method [10]. Hence, in the results, both kinds of evaluations are considered to take one unit of evaluation time.

As shown in Figs. 4 and 5, the hybrid method converges faster than the DE towards the global minimum of the two analytical functions. Moreover, the comparisons represented in the bottom plots of the figures are the most important because they show that the hybridization of the DE algorithm allows to decrease the number of evaluations needed to locate the minimum.

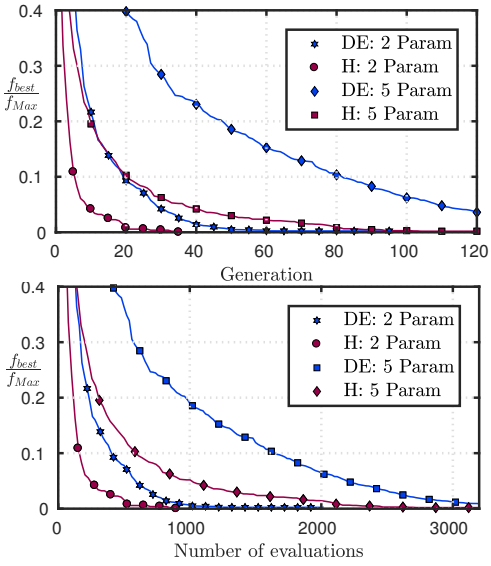

Figure 4: Comparison of the performance of the DE and the hybrid (H) algorithms in terms of number of generations (top) and in terms of number of evaluations (bottom) for the Rastrigin function

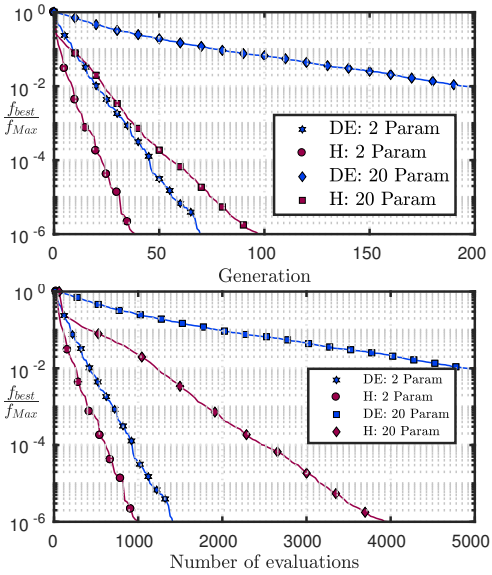

Figure 5: Comparison of the performance of the DE and the hybrid (H) algorithms in terms of number of generations (top) and in terms of number of evaluations (bottom) for the Rosenbrock function

And, in aerodynamic shape optimization, where the objectives are computed by means of CFD, the reduction of this evaluation number is crucial in order to limit the computational cost of the complete optimization. Finally, for both functions, the effect of the curse of dimensionality can be clearly identified in the plots when the number of parameters defining the objective functions increases. Indeed, for the Rastrigin and Rosenbrock functions respectively computed using 5 and 20 parameters, the DE algorithm converges relatively slowly and requires to perform many evaluations. In the case of the hybrid method, the hybridization improves the performance of the evolutionary algorithm and therefore, the proposed algorithm is less affected by the curse of dimensionality.
In conclusion, this section showed that the new hybrid optimization method allows to locate the global minimum of a function at a lower computational cost than classical evolutionary algorithms.

4. Application on a turbomachinery case

In this section, the proposed hybrid algorithm is tested on a typical turbomachinery application: the turbine cascade LS89. The first part of this section describes the different characteristics of the LS89 optimization while the second part presents the results and compares them with those obtained using a classical gradient-based method.

4.1. LS89 Application

In this work, a single point optimization of the LS89 geometry profile is performed in order to minimize the entropy generated by the flow through the cascade. The LS89 turbine blade, represented in Fig. 6, is a highly loaded blade with a flow turning of about 75°. The LS89 cascade has been designed at the Von Karman Institute and extensively tested in various flow operating conditions from the subsonic to the transonic regime [18]. The single point optimization presented in the paper is performed for an operating point characterized by a downstream isentropic Mach number of 0.9. The conditions at the inlet and at the outlet for this operating point are summarized in the Table 1.

4.1.1. Parametrization

In order to enable the geometrical design optimization with regards to its flow performance, the geometry of the cascade is constructed using the parametrization method described in [19; 23]. The turbine is completely defined by 19 design parameters represented in Fig. 6. First the blade camber line is represented with a second order Bézier curve using the control points P_{LE}, P_M and P_{TE}. These control points are obtained using the axial chord c_a, the stagger angle γ and the outlet metal angle β_{out}. Then to define the blade thickness, the control points of the SS and the PS B-Splines are located in the direction normal to the camber line by imposing the distance t_i as shown in Fig. 6. At the LE, the position of the first control points of the SS and the PS is computed in order to obtain a G2 continuity between the SS and PS B-splines, assuring a continuity of the radius of curvature which ensures a smooth flow at the connection. The last control points of the PS and the SS are localized using respectively the angle φ_{PS} and φ_{SS}. Finally, the profile of the blade is closed at the TE by a circular arc. The construction of the cascade is completed by imposing the pitch between two successive blades. During the optimization process, the axial chord and the TE radius remain constant for all the designs. The axial chord is maintained in order to conserve the axial dimension c_a of the cascade and the TE radius r_{TE} is kept fixed in order to respect manufacturing tolerances.

Table 1: Definition of the operating point conditions

Inlet	P_0	147 500 Pa
	T_0	420 $°K$
	α_{in}	0 °
Outlet	P	80 000 Pa

4.1.2. Objective and constraints

The objective of the optimization consists in minimizing the entropy generation J through the cascade while maintaining the flow turning of the original blades in order to achieve at least the same work. This aerodynamic constraint on the outlet flow angle expressed by Eq. 9 is imposed during the optimization process.

$$\frac{\alpha}{\alpha_{LS89}} \geq 1 \tag{9}$$

The exit flow angle α is mass-averaged at the outlet of the computational domain and is equal to $\alpha_{LS89} = 74.89°$ for the original LS89 blade.

The entropy generation (Eq. 10) is computed between the inlet and the outlet of the computational

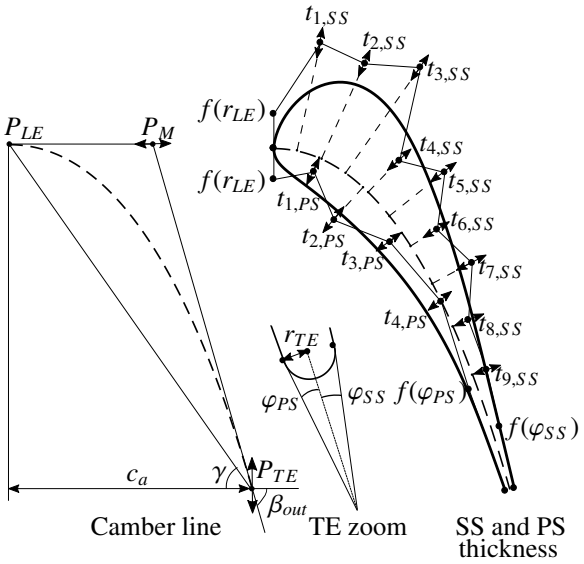

Figure 6: Parametrization of the LS89 turbine cascade

domain using the mass-flow averaged entropy at the respective position.

$$J = S_{out} - S_{in} \tag{10}$$

4.2. Results

In the hybrid method, since the gradients of the objective function and of the constraint can be needed for certain individuals, a complete process chain of tools has been developed at the von Karman Institute, which comprises a geometry generator, a grid generator and a CFD solver. All these tools have been differentiated in order to accurately compute the gradient values of the different functions using a CAD-based adjoint method [2]. The references [19; 23] provide a detailed description of the complete chain of tools and a validation of the gradients computed in the case of the LS89 turbine. In this application, the automatically generated mesh is structured and constructed using one O-grid block around the blade and 6 others H-grid blocks. The mesh generated for the LS89 application is presented in Fig. 7. The numerical simulations are performed by solving the Reynolds-Averaged Navier-Stokes (RANS) equations and using the Spalart-Allmaras equation in order to model the flow turbulence. The validation of the CFD computation is presented in Fig. 8 by comparing the evolution of the isentropic Mach number along the blade obtained numerically with experimental results.

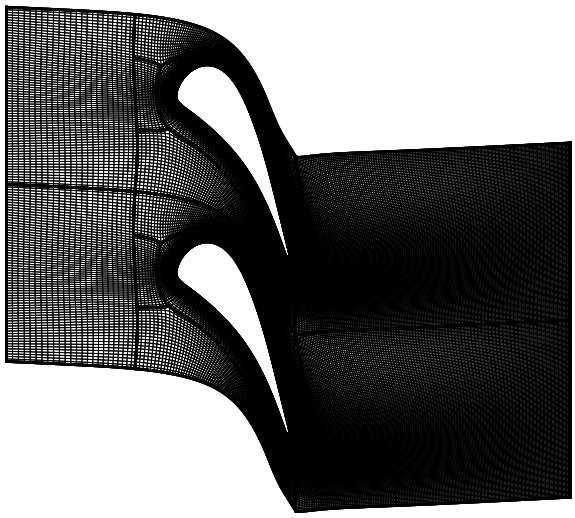

Figure 7: Mesh generated for the LS89 cascade

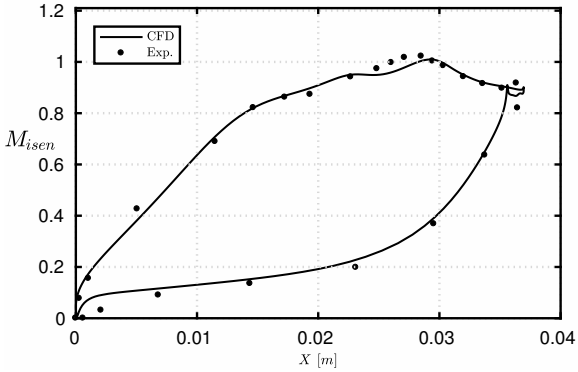

Figure 8: Validation of the CFD solver

The evolution of the population during the hybrid optimization is visualized in Fig. 9. The figure shows for each generation number the composition of the population in terms of objective (top) and constraints (bottom). A very large spread in objective function is present during the first steps of the optimization, but this rapidly reduces with generation number. At the same time, many designs do not satisfy the constraints at the start, but these designs disappear after only 5 iterations through a combination of the selection process of the DE and the gradient-based projection method. Near the end of the optimization process all designs are just satisfying the constraints, meaning that the constraint becomes active and that only an improvement is possible in a direction perpendicular to the constraint gradient. The final population has

still some variation in the objective function, which indicates that some diversity in the population is preserved. This is an essential ingredient in global exploration. Figure 9 shows that the optimization converges towards a population where all the individuals reduce the entropy generation through the cascade by about 20 percent while maintaining the flow turning of the original profile. The evolution of the best individual of each generation is represented in Fig. 10 by a solid line. These are the results of the algorithm after the steepest descent improvement. Also shown in the figure by dots are the designs created by the proceeding evolutionary step which have been improved afterwards by the steepest descent method to the designs represented by the solid line. The figure demonstrates the two main advantages of the hybrid algorithm, namely the acceleration of the convergence by locally improving an individual (arrow A in the figure) and the possibility to correct promising individuals which do not respect the constraint (arrow B in the figure).

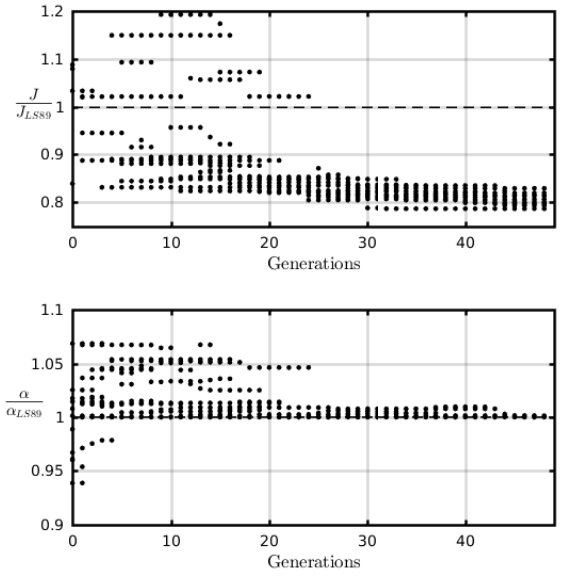

Figure 9: Values of the objective function and the constraint of all the individuals for each generation

The best individual found using the hybrid algorithm reduces the entropy generation through the cascade by 21.3 percent while maintaining exactly the same flow turning. In Ref. [19], based on an identical geometry parametrization, the same single point optimization of the LS89 turbine has been performed

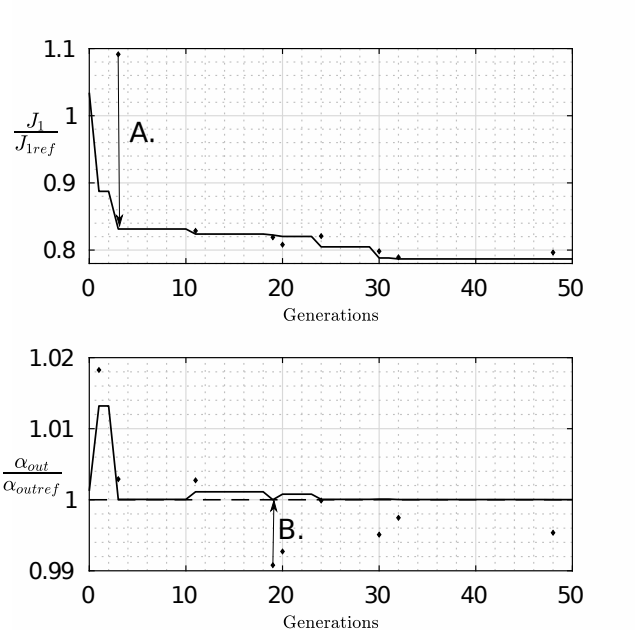

Figure 10: Evolution of the best individual of the populations during the optimization.

using a gradient-based method (GBM). In this optimization, the aerodynamic constraint is imposed by adding a penalty term to the objective function when the inequality Eq. 9 is not satisfied. The results of both optimization methods are compared in Tab. 2. The table shows that the method described in this paper allows to find a much better design improvement of the LS89 turbine cascade (21.3 % compared to 11.6 % improvement respectively). This is due to the fact that the hybrid method performs a global exploration of the design space.

The last row of the Tab. 2 compares the number of evaluations performed during both optimizations where the adjoint computations have the same computational cost than the CFD simulations. It can be seen from this comparison that the GBM needs significantly less evaluations of the objective function to converge. However, the hybrid algorithm allows to perform a global exploration of the design space and therefore, the localization of a better minimum is possible. Moreover, the hybrid method is easily parallelisable by evaluating the different individuals of a population in parallel. Furthermore, between two successive populations, the local improvements of the most promising individuals using the steepest descent method can also be performed in parallel. In terms of computational cost, each CFD and adjoint compu-

tation lasts approximately 20 minutes using two processors. For the overal gradient-based optimization described the Ref [19], two Intel(R) Core(TM) i7-4790K CPU 4.00GHz processors were used for approximately 2 days, while the hybrid method took more or less 10 days using 6 processors of the same type. To conclude, the new algorithm developed allows to find a better minimum than classical optimization methods at a reasonable computational cost.

Table 2: Comparison between GBM results and hybrid results

	LS89	GBM [19] Variation	Hybrid Variation
$J\ [Pa/(kg/m^3)]$	826.7	731 -11.6 %	650.5 -21.3%
$\alpha_{out}[^\circ]$	74.89	74.89 -0 %	74.89 -0%
Evaluations		70	1600

5. Comparison of the profiles

This section describes different characteristics of the new profile obtained at the end of the optimization using the hybrid algorithm. The baseline and the optimized cascades are compared in terms of geometry, isentropic Mach number distribution along the blades, total pressure loss, boundary layer parameters and base pressure.

Comparison of the cascade geometries:. The two geometries of the cascade are compared in Fig. 11. The figure shows that the pitch between two consecutive blades has significantly increased from 0.057 *mm* to 0.076 *mm*. We observe as well an increase in chord length of the blades by lowering the stagger angle (from 54.9° to 59.7°), such that the decrease in solidity remains limited (from 1.11 to 0.96). The leading edge of the new blades is less curved, which leads to a modification of the stagnation point position along the PS. The larger pitch between two consecutive blades and throat area increase the mass-flow through the optimized cascade by about 25% compared to the baseline configuration.

Isentropic Mach number distribution and loading:. The evolution of the isentropic Mach number along the SS and PS of the blades are presented in Fig. 12. Due to the lower curvature on the suction side in the front part of the optimized vane and due to the increase of the mass-flow, the flow is quickly accelerated until X ≈ 0.02m. This acceleration leads to a small supersonic pocket followed by a weak shock on the SS of the blade. In the last part of the SS, the flow is strongly decelerated and reaches a smaller isentropic Mach number than the one obtained downstream of the baseline blades. The comparison of the isentropic Mach numbers clearly illustrates a significantly higher loaded optimized blade, which is in-line with the decreased solidity. The Zweifel coefficient Z_w [24] allows to quantify how efficiently a blade profile is loaded and is expressed as follows:

$$Z_w = \frac{\int_0^1 (P_{PS} - P_{SS}) \frac{dx}{c_a}}{(P_{0in} - P_{out}).c_a}. \tag{11}$$

The Zweifel coefficient of the LS89 turbine cascade is equal to 0.61 [19], while the value of the coefficient computed in the case of the optimized profile is 0.84.

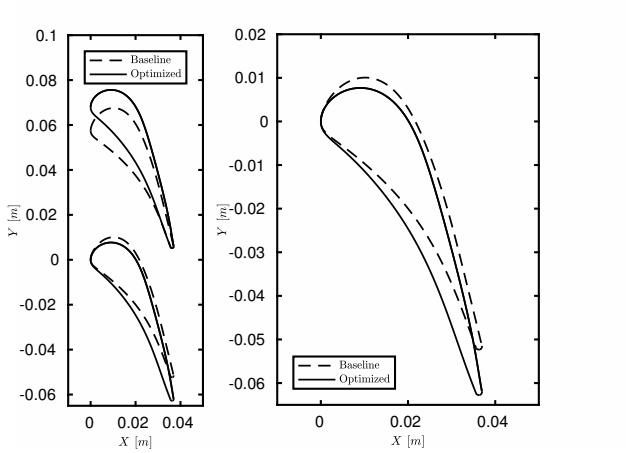

Figure 11: Comparison of the cascade geometries

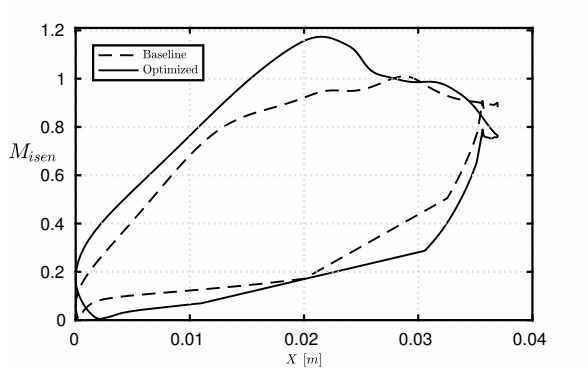

Figure 12: Isentropic Mach Number comparison

The significant difference demonstrates that the new profile is much more efficiently loaded which allows to achieve the same flow turning than the baseline cascade while increasing the pitch.

Total pressure loss:. The evolution of the total pressure loss relatively to the dynamic pressure q in a plane downstream of the blades (in $\frac{X}{C_a} = 1.433$) for the two different cascade geometries is represented on Fig. 13. The figure illustrates that the total pressure losses generated in the wake downstream of the blades is smaller for the optimized geometry. Moreover, the width of the wake compared to the pitch is smaller for the new design. These two factors allow to explain the reduction of the total pressure loss (from 2.665 kPa to 2.07 kPa) between the inlet and the fully mixed-out flow at the outlet which equals to a diminution of about 22.4 percent.

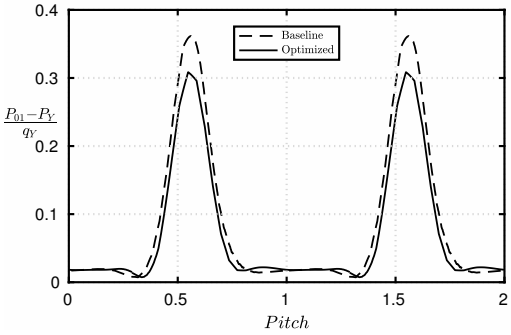

Figure 13: Total pressure loss downstream (at $\frac{X}{C_a} = 1.433$) of the blade

Boundary layer parameters:. The boundary layers have an important impact on the profile losses generated in turbine cascades [25]. Therefore, the boundary layer parameters computed for the baseline and the optimized profiles in two locations along the SS are compared in Tab 3. The first point (1) is placed at 85 percent of the axial chord which represents a location where the strong deceleration starts in the case of the optimized profile. The second point (2) is located closer to the TE at 97 percent of the axial chord. Table 3 shows that, even at the beginning of the deceleration in (1), the boundary layer is thicker in the case of the optimized profile. This is due to the presence of the weak shock at the end of the supersonic bubble on the SS which destabilizes the boundary layer. Moreover, due to the strong deceleration in the last part of SS, the thickness of the boundary layer increases faster along the profile of the new geometry. Comparatively, the boundary layer thickness, displacement thickness and momentum thickness are higher for the optimized cascade which should lead to higher profile losses in that case [25]. However, another term plays a significant role in the loss generation of turbine cascades which is the static pressure at the trailing edge [26; 27; 28; 25].

Table 3: Comparison of the boundary layer parameters along the SS in two different locations (1 and 2 respectively at 85 percent 97 percent of the axial chord)

	$LS89_1$	Opt_1	$LS89_2$	Opt_2
δ $[mm]$	1	1.3	1.3	2
δ* $[mm]$	0.19	0.25	0.25	0.53
θ $[mm]$	0.11	0.14	0.15	0.3
H	1.75	1.74	1.65	1.83

Base pressure:. In turbine cascades, a large part of the blade loss comes from the entropy generated close to the trailing edge. Indeed, at transonic speeds, 70 percent of the losses come from downstream the TE [28]. In this region, the entropy is generated by the mixing of the boundary layers originating of the SS and the PS. As explained in detail in [25; 26; 27; 28], the mixing process leads to a dissipation which is directly linked to the static pressure at the TE of the blades, also called base pressure. The comparison between the base pressure of the LS89 cascade and the one obtained numerically for the optimized geometry is presented in Fig. 14. The figure demonstrates that the pressure distribution at the TE of the optimized profile is higher than the base pressure obtained for the LS89 turbine which eventually leads to lower mixing losses. The difference between the two base pressures is of the order of 12 percent and can be explained by the higher boundary layer thickness just before the TE. Xu and Denton proved in [28] that higher boundary layer thickness upstream the TE leads to larger pressure distribution at the TE. Therefore, the higher base pressure obtained in the case of the optimized profile can be associated to a smaller entropy generation in the TE region. This is illustrated by Fig. 15 which compares the distribution of streamlines around the TE for the baseline and the optimized profiles. The figure shows that the recirculation bubble located in the TE region is smaller in the case of the new geometry which leads to a higher base pressure and, therefore, to a smaller entropy generation

in this region. Therefore, this higher pressure distribution along the TE measured explains the reduction of the entropy generation computed for the new optimized cascade.

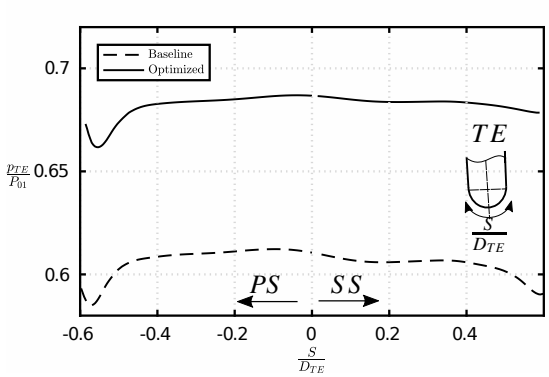

Figure 14: Comparison of the base pressure distributions obtained using RANS simulations

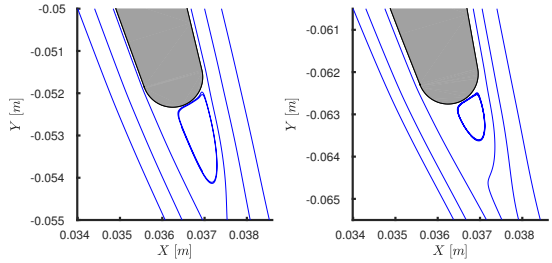

Figure 15: Distribution of the streamlines around the TE for the baseline profile (left) and the optimized profile (right)

6. Conclusion

This paper presents a single point aerodynamic optimization of the LS89 turbine cascade using an hybrid optimization algorithm. The hybrid algorithm combines a classical evolutionary algorithm with a steepest descent method. This algorithm has been developed in order to tackle the most well known disadvantages of both methods and to benefit from their respective strengths. Especially, the hybrid method merges the capacity to perform a global exploration from the evolutionary algorithm with the faster local convergence of the steepest descent method. The hybrid algorithm used in the context of this work is constructed using the Lamarckian evolution approach which consists in incorporating the gradient-based method directly inside the loop of the evolutionary algorithm. Moreover, in this new algorithm, a projected gradient method is combined to the steepest descent method in order to correct the promising individuals which do not respect the aerodynamic constraint.

First, the effectiveness of the new hybrid method is demonstrated on two benchmark problems by comparing its performance with the performance of the classical DE algorithm. Then, the hybrid method is applied to a typical turbomachinery application: the LS89 turbine cascade. The aim of the optimization is to modify the geometry of the cascade in order to minimize the entropy generated inside the cascade while maintaining the flow turning of the original vane. The use of the presented algorithm allows to find a new design which reduces the generated losses by a factor of 21 percent. The optimized geometry is mainly characterized by a larger pitch distance between 2 consecutive blades, a slightly longer chord and the curvature is higher in the front part of the blade. Finally, the analysis of the flow obtained in the optimized cascade in terms of loading, boundary layer parameters and base pressure demonstrates that the reduction of the profile losses are due to a smaller dissipation in the TE region characterized by higher base pressure.

Acknowledgments

This work has been performed in the context of the MOTOR project (http://motor-project.eu/) funded by the European Union Horizon 2020 Framework Programme for Research and Innovation under Grant Agreement No. 678727. Therefore, the authors would like to thanks the European Union for funding the MOTOR project.

References

[1] J. C. Vassberg and A. Jameson, Industrial applications of aerodynamic shape optimization, no. VKI Lecture Series on Introduction to Optimization and Multidisciplinary Design, Brussels, Belgium, 2016.

[2] L. Mueller and T. Verstraete, CAD Integrated Multipoint Adjoint-Based Optimization of a Turbocharger Radial Turbine, Int. J. Turbomach. Propuls. Power.

[3] T. Verstraete, S. Amaral, R. Van den Braembussche and T. Arts, Design and Optimization of the Internal Cooling Channels of a High Pressure Turbine Blade, Part 2 Optimization, J. Turbomach.

[4] S. Shahpar, S. Caloni and L. de Prielle, Automatic Design Optimization of Profiled Endwalls Including Real Geometrical Effects to Minimize Turbine Secondary Flows, Journal of turbomachinery 139(7).

[5] T. Verstraete, Introduction to optimization and multidisciplinary design, no. VKI Lecture Series on Introduction to Optimization and Multidisciplinary Design, Brussels, Belgium, 2016.

[6] J. R. R. A. Martins and J. T. Hwang, Multidisciplinary Design Optimization of Aircraft Configurations - Part 1: A modular coupled adjoint approach, VKI Lecture Series on Introduction to Optimization and Multidisciplinary Design.

[7] S. Chen and J. Montgomery, Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution, Applied Intelligence (2015) 514–526.

[8] T. A. El-Mihoub, A. A. Hopgood, L. Nolle and A. Battersby, Hybrid genetic algorithms: A review, no. Engineering Letters, 13:2.

[9] V. Kelner, F. Capitanescu, O. Léonard and L. Wehenkel, A hybrid optimization technique coupling an evolutionary and a local search algorithm, no. Journal of Computational and Applied Mathematics 215, 2008, pp. 448–456.

[10] A. Jameson, Aerodynamic Design via Control Theory, Journal of Scientific Computing (1988) 233–260.

[11] Y. G. Xu, G. R. Li and Z. P. Wu, A novel hybrid genetic algorithm using local optimizer based on heuristic pattern move, Applied Artificial Intelligence: An International Journal (2010) 601–631.

[12] R. Cheng, M. Gen and Y. Tsujimura, A tutorial survey of job-shop scheduling problems using genetic algorithms: Part II Hybrid genetic strategies, Int J. Computers Industrial Engineering 37(1) 51–55.

[13] T. Yamade and R. Nakano, A genetic algorithm applicable to large-scale job-shop problems, In Parallel problem solving from nature:PPSN II, eds. R. Manner and B. Manderick. North-Holland: Elsevier Science Publisher 281–290.

[14] D. Whitley, S. Gordon and K. Mathias, Lamarckian evolution, the Baldwin effect and function optimization, Davidor Y., Schwefel HP., Mnner R. (eds) Parallel Problem Solving from Nature PPSN III. PPSN 1994. Lecture Notes in Computer Science (1994) 6–15.

[15] N. K. W. E. Hart, J. E. Smith, Recent Advances in Memetic Algorithms, Springer, Berlin, 2005.

[16] F. Muyl, L. Dumas and V. Herbert, Hybrid method for aerodynamic shape optimization in automotive industry, Journal of Computers and Fluids, 33(1) (2008) 97–119.

[17] K. Price and N. Storn, Differential Evolution, Dr. Dobb's Journal (16) (1997) 18–24.

[18] M. L. d. R. T. Arts, A. W. Rutherfort, Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations, Tech. Rep. 174, von Karman Institute for Fluid Dynamics (1990).

[19] I. S. Torreguitart, T. Verstraete and L. Mueller, Optimization of the LS89 axial turbine profile using a CAD and adjoint based aproach, no. 12th European Conference on Turbomachinery Fluid dynamics & Thermodynamics, Stockholm, Sweden, 2017.

[20] T. Verstraete, Multidisciplinary Turbomachinery Component Optimization Considering Performance, Stress and Internal Heat Transfer, Ph.D. thesis, Universiteit Gent (2008).

[21] G. N. Vanderplaats, Numerical Optimization Techniques for Engineering Design.

[22] M. Molga, Test functions for optimization needs.

[23] I. S. Torreguitart, T. Verstraete and L. Mueller, Cad kernel and grid generation algorthmic differentiation for turbomachinery adjoint optimization, no. 7th European Congress on Computational Methods and Applied Sciences and Engineering, Hersonissos, Greece, 2016.

[24] O. Zweifel, The spacing of turbo-machine blading, especially with large angular deflection, Brown Boveri Rev.32 (1945) 436–444.

[25] J. D. Denton, Loss mechanisms in turbomachines, no. ASME 1993 International Gas Turbine and Aeroengine Congress and Exposition, 1993.

[26] C. H. Sieverding, H. Richard and J-M Desse, Turbine blade trailing edge flow characterisics at high subsonic outlet mach number, Journal of turbomachinery 125(2) (2003) 298–309.

[27] C. H. Sieverding, M. Stanislas and J. Snoeck, The base pressure problem in transonic turbine cascade, Journal of Engineering for Power 102 (1980) 711–718.

[28] L. Xu and J. D. Denton, The base pressure and loss of a family of four turbine blades, Journal of turbomachinery 110 (1988) 9–17.

[29] T. Verstraete, CADO: A Computer aided design and optimization tool for turbomachinery applications, no. 2nd Conf. on Engineering Optimization, Lisbon, Portugal, 2010.

[30] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, Berlin, 2006.

[31] T. S. Schobeiri, Turbomachinery flow physics and dynamic performance, Springer, Berlin, 2012.

[32] S. Vagnoli, T. Verstraete, B. Mateos and C.H. Sieverding, Prediction of the unsteady turbine trailing edge wake flow characteristics and comparison with experimental data, Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy.

[33] S. Pierret, Designing turbomachinery blades by means of the function approximation concept based on artificial neural networks, genetic algorithm, and the navier-stokes equations, Ph.D. thesis, Faculté Polytechnique de Mons (1999).

[34] J. Kopriva, G.M. Laskowsky, H. Reza and and al., Assessment of high pressure cooled and uncooled turbine blade wakes via RANS and URANS at engine scale conditions, Paper No. GT2013-94285, ASME turbo expo.