A Design Method For Turbine Blades Using Genetic Algorithms On Parallel Computers

K.C. Giannakoglou [1]

Abstract. The design of 2-D turbine blades using target pressure considerations is described. It is based on the optimization of a set of free-parameters through Genetic Algorithms (*GAs*). Profile shapes are modeled using combinations of circular arcs and Bezier functions. The number of free-parameters may vary, depending on the particular design requirements; they practically consist of the radii and orientations of the circular arcs and the control-points of the Bezier functions, but other parameters may be optimized as well. The exploitation capabilities of the *GA* are improved using adaptive ranges for the free-parameters. Close to the optimum solution, the *GA* switches to an iterative hillclimbing (*HC*) method, which undertakes the final refinement. The method application is demonstrated in a steam-turbine blade design problem.

1 INTRODUCTION

The increasingly demanding turbine operating conditions (higher temperatures and pressures, higher output power levels, etc) require design methods capable to produce new blade shapes that meet certain requirements. Given that the experimental testing of new blades is both costly and time consuming, CFD methods should be used as far as posible. In general, these tools rely on:

(a) the improved understanding of complex flow phenomena occurring in turbine cascades,
(b) the progress in the available CFD analysis methods,
(c) the availability of powerful computing facilities, including parallel processing and
(d) the availability of effective optimization tools.

The proposed method is a contour design method for 2-D blades that is based on a novel geometrical model for the blade geometry. This consists of the combination of two circular arcs for the leading (*LE*) and trailing (*TE*) edges with Bezier curves for the major part of the pressure (*PS*) and suction (*SS*) sides. Free-parameters are identified and controlled by *GAs* and *HC*. Their role is to compute the set of parameters, that give the targeted pressure distribution along the blade surface, under prescribed flow conditions. A primitive variable flow solver for unstructured grids is used for the evaluation of the candidate blade shapes, through solving the flow field in 2-D cascades. Inviscid or viscous flow models and various turbulence models can be used to associate a fitness score with each shape, through post-processing. A distributed-memory parallel computer, in the so-called SIMD mode, helps saving computing time through the concurrent evaluation of multiple candidate solutions.

The literature survey on aerodynamic shape optimization methods will be restricted to some recent works in the field of turbine blade optimization through *GAs*, and is by no means exhaustive. Bezier curves are used in [3] to model the turbine blade contour, whereas a 17-parameter geometrical coding is used in [9] and fourth-order parametric splines in [5]. Due to their smoothing properties, Bezier functions have been preferably used in external aerodynamic shape optimizations as well (see, for instance [2]). Aerodynamic criteria, with or without mechanical and geometry constraints, constitute the objective function. The convergence of *GAs* is often enhanced, by combining them with other optimization methods, using variable population size, or through heuristics.

2 AIRFOIL BLADE PARAMETERIZATION

The first aim is to convey to the reader the proposed parameterization for 2-D turbine blades. The usually rounded form of these blades close to the *LE* and *TE* yields the

[1] National Technical University of Athens, Laboratory of Thermal Turbomachines, P.O. Box 64069, 15710, Athens, Greece

ECCOMAS 98.
Published in 1998 by John Wiley & Sons, Ltd.

modeling through combined circular arcs and Bezier-Bernstein polynomials very attractive. The two circular arcs model the front and rear part of the airfoil. The two Bezier polynomials model the remaining parts along the *PS* and the *SS*, (Fig. 1) and are controlled by $N_P + 2$ and $N_S + 2$ control-points, respectively. The two end-points of each Bezier curve coincide with the end-points of the circular arcs (nodes T1 to T4). In the "main" design scenario, the free-parameters are those listed in Table 1. It should be noted that, the blade stagger angle and the two chord end-points are fixed. Using these points and R_1, R_2, b'_1 and b'_2 values, the centers and radii of the *LE* and *TE* edge circles are defined. The T1 to T4 nodal points are then computed, by drawing the tangents from P1, S1, P5 and S5, respectively (assume that $N_P = N_S = 5$).

R_1	radius of the *LE* circle
b'_1	inlet blade angle
R_2	radius of the *TE* circle
b'_2	exit blade angle
$x_i, i = 2, N_P + 1$	x-coordinate of N_P control-points over the *PS*
$y_i, i = 2, N_P + 1$	y-coordinate of N_P control-points over the *PS*
$x_i, i = 2, N_S + 1$	x-coordinate of N_S control-points over the *SS*
$y_i, i = 2, N_S + 1$	y-coordinate of N_S control-points over the *SS*

Table 1. Set of free parameters

According to this table, the number of free-parameters is equal to

$$N_{FP} = 4 + 2N_P + 2N_S$$

There is reason to couple circular arcs with Bezier curves. According to the Bezier polynomial theory [1], the p^{th} derivative at the each end-point is determined by the point itself and the p adjacent ones. Of course, the first derivative is equal to the slope of the straight line joining the end-point and the adjacent interior control-point. Consequently, drawing the tangent from the 2^{nd} or the $(N_P + 1)^{th}$ (or $(N_S + 1)^{th}$) Bezier points to the leading or trailing edge circles respectively, the location of the 1^{st} and the $(N_P + 2)^{th}$ (or $(N_S + 2)^{th}$) Bezier points can be found. It is important to note that the two curves remain continuous at the junction points.

The tabulated set of free-parameters in neither mandatory nor even the optimum one. Alternative sets can be defined, depending on the design requirements. For instance, so far, blade chord, cascade pitch and stagger angle were fixed to values reflecting experience from previous successful designs. Some or all of them are likely to be considered as extra free-parameters to be optimized as well. On the other hand, N_{FP} can be considerably reduced, by fixing the $N_P + N_S$ x-coordinates of the Bezier control-points; for instance, they could be evenly distributed along the axial chord. In many cases, fixing the x-coordinates of the control-points is harmless, though in some other cases this might be detrimental.

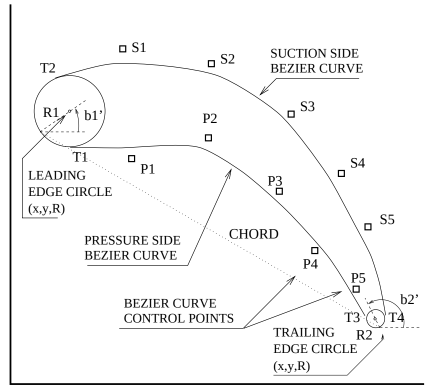

Figure 1. Parameterization of a turbine blade

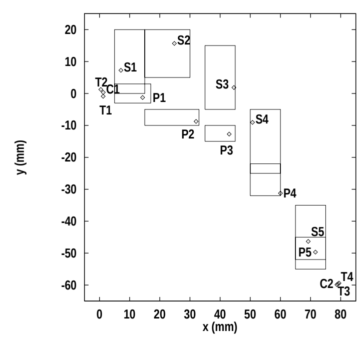

Figure 2. The range of variation (search) of the $N_P + N_S$ Bezier control points

3 OPTIMIZATION THROUGH *GAs*

The optimization problem will be associated with the vector of free-parameters defining the blade shape. Let us denote these parameters by $u_i,\ i = 1, N_{FP}$. The aim is to find a set of u_i values so that the surface pressure distribution of the corresponding blade be as close as possible to the targeted one. Each u_i set represents a search point in the space of potential solutions. *GAs*, viz. random search algorithms based on the model of biological evolution, will be used to find the optimum solution.

In the beginning, a population of N_{pop} individuals is randomly selected over the search space of possible solutions. Each individual is given a fitness score that denotes its suitability. Minimization and maximization problems may be equally handled through a simple transformation of the fitness score. The problem at hand is a minimization problem, where the optimum solution is the one corresponding to the individual with the lowest fitness score among its competitors. The computation of fitness scores requires flow problem solutions through CFD software which are the time-consuming. Depending on their scores, the individuals compete to reproduce offspring; the fittest of them survive while the weaker die out. This is carried out in a repetitive manner, up to the point where no better solutions are likely to appear. At each generation, individuals are subject to genetic mechanisms like selection, mating, crossover and mutation.

The critical problem of modeling each individual, in a manner which is suitable for *GAs*, is now taken up. Each u_i parameter is coded as a binary chromosome

$$C_i\left(b_1,\, b_2,\, b_3,\, \cdots,\, b_{n_i}\right)$$

where $b_m = 0$ or 1. In general, the binary strings for the N_{FP} parameters could be of variable length. The length stands for the number n_i of bits in each chromosome. The lower $u_{i,min}$ and the upper $u_{i,max}$ bounds for each free-parameter are imposed and the binary string C_i is decoded as follows

$$u_i\left(b_1,\, b_2,\, \cdots,\, b_{n_i}\right) \,=\, u_{i,\,min} \,+ \\ + \frac{\parallel C_i\left(b_1,\, b_2,\, \cdots,\, b_{n_i}\right) \parallel}{\parallel C_i\,(1,\, 1,\, \cdots,\, 1) \parallel}\,\left(u_{i,\,max} \,-\, u_{i,\,min}\right)$$

where

$$\parallel C_i\left(b_1,\, b_2,\, \cdots,\, b_{n_i}\right) \parallel = \sum_{k=1}^{n_i} b_k\, 2^{k-1}$$

The concatenation of N_{FP} binary strings gives rise to the full binary string utilized by the *GA*. So, the chromosome length, i.e. the number L of bits used to encode the ensemble of N_{FP} free-parameters is equal to

$$L \,=\, \sum_{i=1}^{N_{FP}} n_i$$

As a supplement, Gray coding [7] representation can be optionally incorporated. Gray coding produces slightly better results and accelerates convergence.

The evolution of the N_{pop} randomly selected individuals is carried through the following genetic operations:

(1) *Parent Selection.* Reproductive trials are allocated to each individual, according to their fitness score. These trials define the number of copies of each individual in the mating pool. The fitter individuals are likely to receive more than one copies. Some of the less fit individuals are likely to stay out of the mating pool. First, all individuals are mapped onto a roulette wheel with N_{pop} slots. The size of the slot corresponding to the i^{th} individual is proportional to the difference between its score and the score of the worst individual in this generation. Using the roulette wheel, the mating pool for the next generation is formed.

(2) *Crossover.* Pairs of individuals are selected from this pool at random and their chromosomes are cut at a point selected at random; the parts after and before the cuts are mutually exhanged. Here, a 1-point crossover per free-parameter is used, which corresponds to a N_{FP}-point crossover for the full chromosome of length L. So, the substring of any free-parameter contain a mixture of genes from the parental substrings. The crossover possibility for each pair in the mating pool is kept very high (90%).

(3) *Mutation.* Mutation is applied to each individual after crossover. It randomly alters bits with a small probability (usually less than 1%).

4 THE EVALUATION OF CANDIDATE SHAPES

The evaluation of candidate solutions is carried through a 2-D flow solver for unstructured grids with triangular elements. The flow solver is based on the finite-volume method. Over each control volume, the governing equations are discretized using second-order upwind schemes for the convective terms and a linear distribution scheme for the diffusion ones [6]. The numerical solution is performed through the Jacobi scheme, involving a number of sub-iterations. For viscous flow calculations, the standard Jones-Launder $k-\varepsilon$ model [4] has been used. Wall functions are applied along the walls by assuming that the local node is in a small but constant distance from a hypothetical solid surface. Thus, slip conditions are locally imposed.

Having obtained the numerical solution of the flow equations in the 2-D cascades formed using the candidate blade profiles, the evaluation of the corresponding chromosome takes place. The fitness function is defined as the area between the targeted and the predicted pressure distributions along the profile. The integration yields a positive real value which constitutes a measure of the chromosome's quality. It is indispensable to perform these computations using a $p = p(s)$ representation, where s is the arc-length along the airfoil contour. By doing so, differences in the pressure distribution occuring close to the rounded *LE* can be distinctly taken into account.

5 METHOD ACCELERATION

In the proposed method, the chromosome length L exceeds the limits which, according to our experience, allow the efficient usage of *GAs*. This is due to the large number of free-parameters (in the "main" design scenario, $F_{FP} = 24$, for $N_P = N_S = 5$), each of which needs to be coded using a five- to seven-bit length substring. Therefore, the length of the coding string is expected to be in the range of $120 - 168$ bits. With these figures in mind and a population size of about $30 - 50$ chromosomes, the computing cost of the *GA* increases considerably. It has been also observed that the number of generations needed to reach an almost optimum solution increases too. In general, the standard *GA* manages to rapidly improve the fitness score of the best solution during the first generations, but drifts slowly (and often endlessly) during the subsequent generations.

In order to overcome this unpleasant behaviour, an adaptive-range scheme incorporated to the standard *GA* and, towards convergence, the hillclimbing optimizer was used instead.

The adaptive-range scheme consists of the re-definition of the range of variation for each free-parameter, according to the *GA* results. The re-definition of $u_{i,min}$, $u_{i,max}$, $i = 1, N_{FP}$ takes place at a certain generation, if no real benefit is expected from a longer calculation. A criterion based on the lack of improvement of the best score over the last k, $k = 5 - 10$ generations, is applied. Let $u_{i,min}$, $u_{i,max}$, be the $i - th$ variable bounds used for the *GA* during the first generations. Assume, also, that the *GA* derived the best chromosome which corresponds to the u_i^* values. Then, the subsequent *GA* operations will be carried out using a new range, given by

$$u_{i,min}^{new} = u_i^* - \frac{1}{2}\frac{u_{i,max} - u_{i,min}}{p},$$
$$u_{i,max}^{new} = u_i^* + \frac{1}{2}\frac{u_{i,max} - u_{i,min}}{p} \qquad (1)$$

where p is a small integer ($p = 2 - 4$). The re-definition of the range may be used more than once, if this contibutes to the deeper convergence of the *GA*.

Upon convergence of the *GA*, with or without intermediate re-definitions of the search space, the *HC* algorithm takes over. This is based on the best chromosome calculated by the *GA* and promotes small changes resulting to score improvement. In its iterative version, the algorithm builds L new chromosomes by successively altering one bit in the string. After evaluating them through the Navier-Stokes solver, the best of them (if different from the previous best) is chosen and the procedure restarts up to the final convergence. A maximum number of iterates is allowed, otherwise *HC* terminates whenever improvement to the current chromosome is not feasible. The *HC* improves further the solution obtained through the *GA*, though in the expense of costly computations. Nevertheless, the *HC* cost can be reduced through (a) parallelization, (b) selective evaluation of a small subset of the L new chromosomes, according to a preceding sensitivity analysis (it is beyond our scope to discuss it here), or (c) combination of both.

Apart from its sequential version, the proposed method is also available in a parallel version for distributed-memory computers. The Intel-Paragon computer of the Supercomputing Center of NTUA was used along with the so-called fine-grain parallelization of the *GA* and the *HC* part of the algorithm. Since the costly part is the evaluation of the candidate chromosomes through iterative CFD direct solvers, each processor is associated with one chromosome and undertakes the relevant computations. The generation of the unstructured grid, the iterative solution of the Navier-Stokes equations up to a specified accuracy and the comparison between predicted and target pressure distributions, are all performed in parallel.

On the contrary, all genetic operators are undertaken by a master processor. Thus, the proposed method is not a parallel *GA* in the strict sense, but the available processors of a parallel computer are used to minimize the duration of a single turbine blade design.

6 RESULTS - THE DESIGN OF A STEAM-TURBINE BLADE

In this Section, the design of a steam turbine blade will be demonstrated. An existing steam-turbine blade shape, with data available in [8], (chord $= 100mm$, stagger angle $= 37.11deg.$, pitch-to-chord ratio $= 0.55117$) will be used as the background profile. With the inlet flow angle and the isentropic exit Mach number fixed to the experimental values, $19.3deg.$ and 1.189 respectively, a shock that emanates from the trailing edge appears, which then reflects on the *SS* of the adjacent blade. The shock reflection is visible on the *SS* pressure distribution at about 75 percent axial chord, as a region of recovered pressure.

The experimental pressure distribution (or more pre-

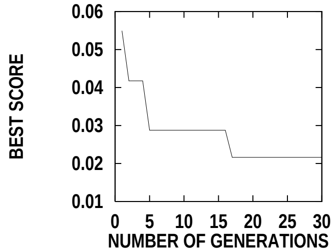

Figure 3. Best fitness score history (*GA1*)

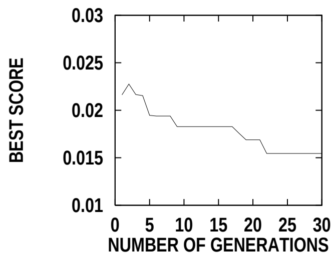

Figure 4. Best fitness score history (*GA2*)

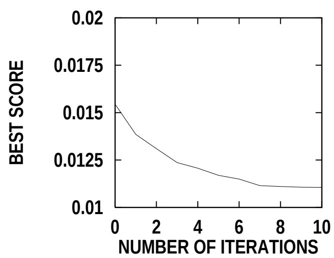

Figure 5. Best fitness score history (*HC*)

cisely that obtained using a direct calculation) was purposely modified to give the targeted pressure distribution. A linear distribution replaced the wavy one over the front part of the *SS*, while the pressure distribution representing the shock reflection over the *SS* rear part remained intact. Using the so-defined target pressure distribution, we proceeded to the design of the turbine blade that was expected to be quite close, though not identical, to the original one; the latter will be referred to as the "reference" blade. The target pressure distribution is repeated in most of the figures of this paper; for practical reasons, $p = p(x)$, instead of $p = p(s)$, plots are shown.

Fig. 2 illustrates indicative ranges of variation for the 10 Bezier control points. The lower and the upper bounds of the free-parameters were assumed to differ sunstantially during the first phase of the *GA*-based optimization (*GA*1). Using the user-specified ranges, *GA*1 resulted to the evolution of the best score shown in 3. A rapid improvement of the best score in the early generations was followed by a slower one. The *GA*1 could probably terminate about the 15th generation, but 30 generations are presented in order to demonstrate the evolution trends. It is worth noting that there is lack of improvement of the best score during the last (more than 10) generations.

At the end of *GA*1, the range of variation for all free-parameters was re-adjusted, according to eq. 1 ($p = 4$) and the second *GA* phase started (*GA*2). Its convergence history is shown in fig. 4; improvements occured much more often and the best score gradually improved. At the end of the 30th *GA*2 generation, the optimization tool switched to *HC*, which took about 10 iterates (fig. 5) to improve the best score. The finally obtained blade is shown in fig. 6. We recall that the "reference" blade shape is not the targeted shape and deviations in fig. 6 should not be interpreted as design errors.

Figs. 7, 8 and 9 illustrate the pressure distributions along the solid walls of the optimum blade shapes at the end of *GA*1, *GA*2 and *HC*. Using only *GA*1, the pressure distribution over the *PS* was "easily" captured, despite the coarse binary analysis (5 bits per variable). The shock wave location and intensity was also correctly reproduced, as may be seen from the satisfactory comparison of predicted and targeted pressure distributions at the rear part of the *SS*. Indeed, the shock reflection patterns on the *SS* are very close to the targeted ones, even if discrepancies do appear in the front part of the *SS*. The improvements due to *GA*2 and *HC* are mainly concerned with the latter. The blade obtained using *GA*1, *GA*2 and *HC* gives a pressure distribution that is in perfect agreement with the targeted pressure distribution.

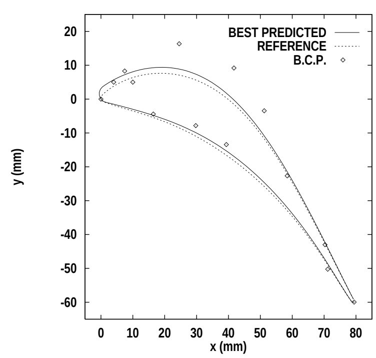

Figure 6. Blade shape corresponding to the (*HC*) best solution

7 CONCLUSION

A design method for 2-D turbine blades was presented. The designer's requirements are cast in the form of a targeted pressure distribution and a number of geometrical parameters are controlled by *GAs*, with or without re-adjustment of their range or *HC*. The undertaken transonic steam-turbine design problem was succesful and the major conclusions are as follows:

(a) The proposed geometrical model, using circular arcs and Bezier polynomials, proved to be very effective for the modeling of turbine blades.

(b) *GAs* proved to be efficient optimizers, especially if combined with acceleration techniques (adaptive ranges of the free-parameters, hillclimbing). The adaptive-range method overcomes convergence stagnation problems and improves considerably the optimum solution. The iterative hillclimbing undertakes the final refinement.

(c) The parallelized version of this method divides the design time approximately by the number of available processors.

REFERENCES

[1] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*, Academic Press, Inc, San Diego, CA, 1988.

[2] M. Galan, G. Winter, G. Montero, D. Greiner, J. Periaux, and B. Mantel. A transonic flow shape optimization using genetic algorithms. ECCOMAS 96, John Wiley I& Sons, 1996.

[3] S. Goel, J.I. Cofer, and H. Singh. Turbine airfoil design optimization. ASME Paper 96-GT-158, 1996.

[4] W.P. Jones and B.E. Launder, 'The prediction of laminarization with a two-equation model of turbulence', *Int. J. Heat Mass Transfer*, **15**, 301–314, (1972).

[5] Li Jun, Feng Zheping, Chang Jianzhong, and Shen Zuda, 'Aerodynamic optimum design of transonic turbine cascades using genetic algorithms', *J. of Thermal Science*, **6(2)**, 111–116, (1997).

[6] D.G. Koubogiannis, L.C. Poussoulidis, D.V. Rovas, and K.C. Giannakoglou, 'Solution of flow problems using unstructured grids on distributed memory platforms', *Comp. Meth. Appl. Mech. Eng.*, (1998).

[7] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin Heidelberg, 2nd edn., 1994.

[8] M. Stastny and P. Safarik. Experimental analysis data on the transonic flow past a plane turbine cascade. ASME Paper 90-GT-313, 1990.

[9] M.A. Trigg, G.R. Tubby, and A.G. Sheard. Automatic genetic optimization approach to 2D blade profile design for steam turbines. ASME Paper 97-GT-392, 1997.

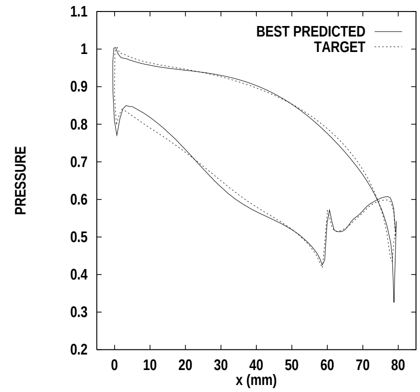

Figure 7. Pressure distribution of the (*GA1*) best solution

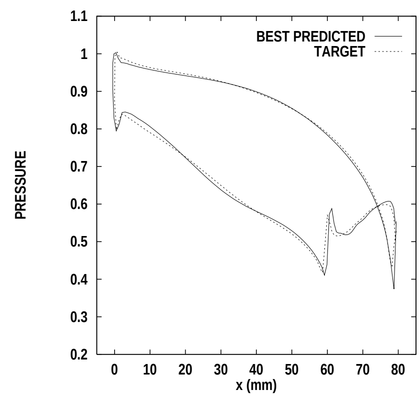

Figure 8. Pressure distribution of the (*GA2*) best solution

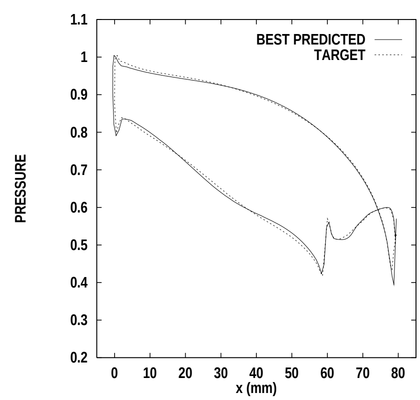

Figure 9. Pressure distribution of the (*HC*) best solution