Global Optimization Methods for the Aerodynamic Shape Design of Transonic Cascades

T. Mengistu and W. Ghaly

Department of Mechanical and Industrial Engineering
Concordia University, Montreal, Quebec, Canada

Email: mengistu@me.concordia.ca, ghaly@alcor.concordia.ca

ABSTRACT

Two global optimization algorithms, namely Genetic Algorithm (GA) and Simulated Annealing (SA), have been applied to the aerodynamic shape optimization of transonic cascades; the objective being the redesign of an existing turbomachine airfoil to improve its performance by minimizing the total pressure loss while satisfying a number of constraints. This is accomplished by modifying the blade camber line; keeping the same blade thickness distribution, mass flow rate and the same flow turning. The objective is calculated based on an Euler solver and the blade camber line is represented with non-uniform rational B-splines (NURBS). The SA and GA methods were first assessed for known test functions and their performance in optimizing the blade shape for minimum loss is then demonstrated on a transonic turbine cascade where it is shown to produce a significant reduction in total pressure loss by eliminating the passage shock.

1. INTRODUCTION

Aerodynamic optimization methods are becoming very attractive in today's competitive environment as they can reduce the design cycle time by automating the design process. Until recently, designers were relying mostly on manual optimization. As Computational Fluid Dynamics (CFD) matured over the last decade and as computing technology has greatly improved and has become more affordable, simulation-based optimization is becoming affordable and more popular than ever. This is due to the fact that optimization techniques give direct control on performance parameters, even if the computational cost is at least one order of magnitude larger than the cost of an analysis calculation.

Aerodynamic shape optimization allows the designer to explore the design space to achieve a given objective. One possible design objective is to minimize flow losses, which can be measured by the total pressure loss (or entropy generation). Minimization of flow losses can be achieved by properly reshaping the blade profile. Automated aerodynamic design is accomplished by coupling a CFD (Computational Fluid Dynamics) flow simulation code with numerical optimization methods. As the aerodynamic shape optimization problem is a complex one with possibly many local minima, gradient-based methods can be trapped in a local optimum, unless the initial guess is close to the global minimum. For this reason, heuristic/evolutionary global algorithms such as Genetic Algorithm (GA) and Simulated Annealing (SA) are used to ensure reaching the global minimum. These algorithms have been recently applied in turbomachinery design problems; examples of such algorithms are given in [1-3].

Dennis *et al.* [1] used a combination of genetic and SQP algorithms to optimize a two-dimensional turbine cascade. They enforced the equality constraints on the mass flow rate, blade exit angle and other quantities. The optimization scheme required from 220 to 675 calls to the flow analysis code. Wang *et al.* [2] implemented the Simulated Annealing method on multiple processors for Aerodyamic Shape Optimization. Their major focus was in the reduction of the computation associated with the stochastic global optimization algorithm, simulated annealing. Oyama *et al.* [3] worked on 3D blade shape optimization by including the mass flow rate and pressure ratio as constraints in the objective function.

In this work, the objective is to minimize the total pressure loss for a two-dimensional cascade assuming inviscid transonic flow and given that the original and the optimized blades have the same thickness distribution, same mass flow rate, and accomplish the same overall flow turning (or work); they also have the same inlet and exit flow conditions. The objective function is constrained. The shape optimization is carried out for different transonic cascades to demonstrate the ability of the optimizer to minimize the total pressure loss by eliminating the passage shock. The present work uses stochastic and global optimization algorithms, Simulated Annealing (SA) and Genetic Algorithm (GA) as optimization schemes.

2. THE DESIGN PROBLEM

2.1 Problem definition

A common aerodynamic shape design problem is the redesign of an existing turbomachine airfoil to improve its performance. Aerodynamic shape redesign for an existing airfoil requires that a number of specific constraints be enforced; these constraints are as follows. The original and the new blades should have the same axial chord so that it may fit into the existing turbomachine. The inlet and exit flow angles should be the same in the redesigned blade as in the original one otherwise the velocity triangles will not match with the neighboring blade rows. The mass flow rate through the new blade row must be the same as the original blade row otherwise the turbomachine will perform at an off-design mass flow rate which can lead to serious drop in efficiency and create unsteady flow problems. The spacing between the blades will be kept fixed as the total number of blades in the turbomachine is to be maintained. The thickness distribution for the redesigned airfoil is chosen to be the same as the original airfoil. This is to avoid an unacceptably thin blade and to fix the section area so that it will be able to sustain the expected loads without performing a detailed elasticity analysis of the blade geometry. The objective is then defined as the minimization of the total pressure loss for a given specified flow boundary conditions and structural requirements.

2.2 NURBS representation

The geometric representation of the profiles is an important part in any aerodynamic shape optimization procedure. In this work, the blade profile is defined using the mean camber line and the tangential thickness distribution as shown in Fig. 1. Since the thickness distribution is prescribed, the camber line has to be modified to achieve the objective. The shape optimization procedure is conducted by iteratively determining the optimum airfoil shape that satisfies all the constraints and meets the objective function. This implies that a large number of airfoils will be created and examined using an appropriate fluid flow analyzer and optimization algorithm. In order to minimize the computations task, the airfoil shape is parameterized with a relatively small number of parameters that will then serve as design variables. This is achieved using Non-Uniform Rational B-Splines (NURBS). In addition NURBS have the advantage that local shape modification can be easily done without affecting other parts of the airfoil.

The NURBS curve is given by a sum over all control points, n, of a rational B-spline $N_{i,p}(u)$, times the control point coordinates, $\vec{P}_i$, times a weight, w_i, so that the coordinates of the blade profile are determined once the control points and the corresponding weights are specified. The NURBS curve is defined as [4]:

$$\vec{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \vec{P}_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j} \tag{1}$$

Where $\vec{P}_i$ are the x- and y-coordinates of control point i, w_i is the corresponding weight, $N_{i,p}$ is the p^{th} degree B-spline basis function, and $\vec{C}_i(u)$ are the x- and y-coordinates of control point i on the curve, which corresponds to u_i, the i^{th} element of the knot vector $\vec{u}$. The latter is determined using the chord length method [4]. The basis functions $N_{i,p}$ vanish everywhere except in the vicinity of point i, where the size of this vicinity depends on the order p. The weight w_i provides control on the curve attraction towards control point i. The NURBS are defined on the non-uniform parameters called knots, so that some of the control points affect a larger region of the curve while others affect a smaller region depending on the knot vector distribution.

The key feature of a NURBS curve is that its shape is determined/controlled by the set of control points and the corresponding weights.

Moreover, placing and moving either one or more of the control points, the knots or the weights can accomplish either a local or a global change of the target shape. A NURBS curve also represents conics exactly, e.g. circles, ellipses, hyperbolas, cylinders, cones. This implies that NURBS functions can represent a much wider family of curves compared with what B-splines or Bézier curves can represent, while simultaneously ensuring the profiles smoothness. They are generalizations of B-splines and Bézier curves and surfaces and are becoming industry standard in surface representation.

3. Optimization Algorithms

The aerodynamic shape optimization problem reduces to solving a constrained optimization problem. Two global optimization techniques were used: Simulated Annealing (SA) and Genetic Algorithm (GA) methods are the two algorithms that are studied and the results are compared.

3.1 Genetic Algorithm

Genetic algorithms are general-purpose search algorithms based upon the principles of evolution observed in nature. Genetic algorithms combine selection, crossover, mutation, and elitism operators with the goal of finding the best solution to a problem. Genetic algorithms search for this optimal solution until a specified termination criterion is met [5,6].

The variables for the GA algorithm (the genes) can be either binary coded or real coded. In this work, a real coded genetic algorithm is implemented where the design parameters are represented using a floating-point representation where an individual is characterized by a vector of real numbers. A floating-point representation is more natural and closer to the design space than a binary representation, which also requires a longer string to represent the individual.

The GA algorithm that was developed, implemented and reported in this work, involves four basic operations namely, selection, crossover, mutation and elitism.

Selection is a genetic operator that chooses a chromosome from the current generation's population for inclusion in the next generation's population. Before making it into the next generation's population, selected chromosomes may undergo crossover and/or mutation (depending upon a probability of crossover and mutation) in which case the offspring chromosome(s) are actually the ones that make it into the next generation's population.

There are different types of selection schemes. The one implemented in this work is the Roulette (rank biased) selection [6]. It is a selection operator in which the chance of a chromosome getting selected is proportional to its fitness or rank (which is based on the objective function). This is where the concept of survival of the fittest comes into play.

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-defined crossover probability.

Two kinds of crossover operations are included in the real-coded GA developed in this work namely, arithmetic and heuristic crossover operators.

Arithmetic crossover operator combines linearly two parent chromosome vectors to produce two new offspring while heuristic crossover operator uses the fitness values of the two parent chromosomes to determine the search direction and creates the new offspring.

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at a solution better than was previously possible. Mutation is an important part of the genetic search as it helps to prevent the population from stagnating at any local optimum. Mutation occurs during evolution according to a user-defined mutation probability. This probability should usually be set fairly low (the default value is set to 0.01). If its value is set too high, the search will turn into a primitive random search.

Uniform type mutation is used for the algorithms in this work; it replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene.

For a generational GA, elitism makes few like two copies of the best performer in the old pool and places them in the new pool, thus ensuring that the fit chromosome survives. It is simply the guarantee that the fit solution found to date would remain within the population.

Three stopping criteria are imposed: when the specified fitness threshold is reached, or the specified maximum number of generations is reached, or the elapsed evolution time exceeds a specified maximum.

3.2 Simulated Annealing

As its name implies, the Simulated Annealing exploits analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a general system. If a physical system is melted and then cooled slowly, the entire system can be made to produce the most stable (crystalline) arrangement, and not get trapped in a local minimum.

The SA algorithm was first proposed by Metropolis *et al.* [9] as a means to find the equilibrium configuration of a collection of atoms at a given temperature. Kirkpatrick *et al.* [7] were the first to use the connection between this algorithm and mathematical minimization as the basis of an optimization technique for combinatorial (as well as other) problems.

SA's major advantage over other methods is its ability to avoid being trapped in local minima. The algorithm employs a random search, which not only accepts changes that decrease the objective function f, but also some changes that would increase it. The latter are accepted with a probability

$$p = \exp\left(-\frac{\delta f}{T}\right) \tag{2}$$

Where δf is the increase in f and T is a control parameter, which by analogy with the original application is known as the system "temperature" irrespective of the objective function involved. Briefly SA works in the following way. Given a function to optimize, and some initial values for the variables, simulated annealing starts at a high, artificial, temperature. While cooling the temperature slowly, it repeatedly chooses a subset of the variables and changes them randomly in a certain neighborhood of the current point. If the objective function has a lower function value at the new iterate, the new values are chosen to be the initial values for the next iteration. If the objective function has a higher function value at the new iterate, then the new values are chosen to be the initial values for the next iteration with a certain probability, depending on the change in the value of the objective function and the temperature. The higher the temperature and the lower the change, the more probable the new values are chosen to be the initial variables for the next iteration. Throughout this process, the temperature is decreased gradually, until eventually the values do not change anymore. Then, the function is presumably at its global minimum. The global minimum is obtained by choosing an appropriate "cooling schedule", which includes the temperature and its cooling rate.

A cooling schedule describes the temperature parameter T, and gives rules for lowering it as the search progresses. Unfortunately, there is no systematic way of determining the best annealing schedule for a given optimization problem. The one implemented in this work is that suggested by Corana *et al.* [8] in which T is decreased geometrically after a predetermined number of function evaluations. An initial temperature T_{in} is given based on Kirkpatrick's [7] suggestion, namely that a suitable initial temperature is one that results in an acceptance rate of about 80% for all moves and, after every m steps, T is multiplied by ε, a temperature reduction factor, which is a factor between zero and one.

4. OPTIMIZATION SCHEME VALIDATION

In order to test the SA and GA implementations, some standard optimization problems were chosen from the literature [14]. The problems size ranges from 2 to 40 design variables. The tests include the Rosenbrock banana function and the Rastrigin function. The test functions vary in difficulty, in number of local minima, and in number of design variables X. They have a global extremum that is hidden among many local extrema. Therefore they are appropriate for testing different implementations of global search optimizers such as GA and SA. The search range that was chosen for each function includes several local minima. These problems are given below.

4.1 The Rosenbrock function

The Rosenbrock test function is given by:

$$f(X)=\sum_{i=1}^{n-1}(100(x_i^{\,2}-x_{i+1})^2+(x_i-1)^2) \quad (3)$$

It is a classic optimization problem, also known as the banana function. The global minimum is inside a long, narrow, parabolic shaped flat valley. The convergence to the global optimum is difficult and hence this problem has been repeatedly used to assess the performance of optimization algorithms.

The function has a global minimum value of zero at the point x_i=1 for all i, i.e. f(1)=0 is the global optimum. Different ranges of the design variables were defined as ±2.048, ±10.048 and ±60.048. In the following results, the range ±10.048 was used.

Table 1, which compares the current SA results with those of Ref. [14], shows that the present implementation of SA is more efficient in terms of the number of function evaluations needed to reach the global minimum. Figure 2 shows the convergence history for the problem of minimizing the Rosenbrock function with forty design variables, n=40. Both GA and SA have shown a large improvement in the optimal solution relative to the initial one, but SA outperforms GA in getting very close to the global solution. SA was able to find the global optimum with an absolute error of 10^{-20} with 2,424,000 function evaluations.

In all cases, the SA was able to find the global optimum point very accurately without any significant change in the computation time whereas the GA algorithm had difficulty in reducing the absolute error below 20, except for the case where n=2 (i.e., two variables) the GA did find the global minimum very accurately.

4.2 The Rastrigin function

The Rastrigin function is given by the following expression:

$$f(X)=10n+\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)) \quad (4)$$

It is a widely used multivariable multimodal (i.e., with several local extrema) test function. The function global minimum value is 0 and it occurs at x_i=0, i.e. the global minimum is f(0)=0. In the optimization search, all the x_i's are defined in the range ±5.12 and ±10.12.

Figure 3 shows a comparison of the SA and GA algorithms applied to the Rastrigin function, with forty design variables, n=40. Both SA and GA brought a large improvement in the optimal value. SA improved the optimal value by 99.6 % relative to the initial value while GA improved it by 100 %. The absolute error is less than 10^{-14} for GA and 4.97 for SA.

The range of the design variables was successively increased form ±5.12 to ±10.12, which increased the number of local optima from 11 to 21 as each integer corresponds to a local minimum of the Rastrigin function. In all cases the GA was able to find the global minimum accurate to 10^{-14} without any significant change in the computation time. The values reported in Table 2 correspond to an x_i range of ±10.12.

Fogel and Beyer [15] reported results for the Rastrigin function with 30-variables; the best function value that they reported was larger than 10 after 200,000 function evaluations. In the present work, starting from the same initial range of design variables, a function value of 10^{-6} was achieved in 88,100 function evaluations, and a value less than 10^{-14} was achieved with 278,800 function evaluations.

Deb *et al.* [14] reported results for the same function but with 20-variables and starting the optimization from an initial solution away from the global optimum, they were only able to reduce the function to a value between 10 and 20.

4.3 Concluding remarks

Further tests were carried out with GA and SA and, in almost all cases tested, it was found out that SA and/or GA were able to track the global optimum in a reasonable computational effort however the algorithm parameters have to be well tuned.

The solution accuracy and convergence rate for both SA and GA algorithm depends on the respective parameters. For SA, the solution is most sensitive to the cooling schedule parameters i.e., the temperature (T) and its reduction factor (rt). A temperature of $5\text{x}10^6$ and 10^6 are used for the Rosenbrock and Rastrigin test functions respectively, and a temperature reduction factor of 0.85 is used for both cases.

The value of the parameters used in testing the GA on the Rosenbrock and Rastrigin functions are as follows. The cross over probability is 0.65 and 0.85, the mutation probability is 0.01 and 0.05 and the elitism is 2, respectively.

The test functions that vary in difficulty, in number of local minima, and in number of design variables X, have been tested with the GA and SA routines developed by the authors. The GA algorithm had difficulty with the Rosenbrock function, which has a long flat valley near the global minimum; while the SA algorithm had difficulty with the Rastrigin function, which has numerous local minima. This suggests that any optimizer would not necessarily work for all cases.

5. AERODYNAMIC OPTIMIZATION PROBLEM

Blade shape optimization is a crucial step in the design cycle of a turbomachine. This research is focused on adjusting the blade profile so as to minimize the total pressure loss subject to some geometric as well as flow constraints.

Blade design using optimization techniques is becoming desirable and practical; one important issue is how to choose the design variables and reduce their number to a minimum while maintaining the freedom and quality of the blade shape representation. The x-coordinates of the NURBS control points, used in the blade geometric parameterization, were fixed while the y-coordinates and weights were taken as the design variables. This parameterization ensures good continuity of the blade profile, it fixes the chord length, and the parameters defining the geometry have intuitive meaning that facilitates imposing constraints on their variation so as to restrict the design space. The cost function is the total pressure loss subject to several aerodynamic and mechanical requirements, which include fixed mass flow rate, fixed inlet and exit flow angles, fixed exit static pressure, fixed axial chord and spacing and fixed thickness distribution.

In order to redesign a given blade, the following approach is used. Initially we have a transonic blade which has a shock formed on the blade suction side. The original blade profile, which is described by its camber line and tangential thickness distribution, is taken as the initial design. The camber line is represented by NURBS using from 7 to 11 control points whereas the thickness distribution remains fixed for all design candidates so that the resulting profile does not end up with an unacceptably thin blade.

The optimization algorithm generates a new blade profile by changing the control points for the camber line. The unstructured grid is generated for this new blade geometry, Euler equations CFD solver is then used to simulate the flow and compute the total pressure loss, which is the objective function, and the mass flow rate and exit flow angle, which are the constraints. These steps are repeated until the total pressure loss is reduced to a minimum, which implies that the shock is either eliminated or at least weakened, and the mass flow rate and exit flow angle are fixed to within a tolerance.

The objective function is penalized with delta mass flow rate and delta exit flow angle (to enforce a given mass flow rate and flow turning) and the other constraints are directly enforced through the prescribed inflow boundary conditions imposed in the CFD code, which are the total pressure, total temperature and inlet flow angle as well as the static pressure at the outflow boundary. The mathematical form of the objective function, E, could be expressed as

$$E = \Delta P_t + C_1 |\Delta m| + C_2 |\Delta \beta_2| \tag{5}$$

Where ΔP_t is the total pressure loss, $\Delta\beta_2$ is the difference between the computed and the target exit flow angles in degree, and Δm is the difference between computed and target mass flow rates. The weights C_1 and C_2 are user specified penalty coefficients; they assume the following values.

For GA, C_1 and C_2 are zero below a certain threshold, otherwise they take the following values:

$C_1 = 1000$ when $|\Delta m| > 0.01$
$C_2 = 10$ when $|\Delta\beta_2| > 1^\circ$

For SA, $C_1 = 0.5$ and $C_2 = 0.002$; they are chosen so as to have equal penalizing effect on the objective function.

5.1 Flow simulation method

The two-dimensional inviscid transonic flow in a linear cascade is simulated using a cell-vertex finite volume space discretization method on an unstructured triangular mesh. The steady state solution is reached by pseudo-time marching the Euler equations using an explicit five-stage Runge-Kutta scheme. Local time stepping and implicit residual smoothing were used for convergence acceleration. The non-linear blend of second and fourth order artificial viscosity was found to be successful in capturing shocks and eliminating pressure-velocity decoupling with minimal numerical diffusion. The method of characteristics was used to impose inflow and outflow boundary conditions. More details on the discretization method can be found in Ahmadi and Ghaly [13].

The flow at the inlet and exit planes, which are placed one chord upstream and downstream of the cascade, is always subsonic. At the exit plane, the exit static to inlet total pressure is specified. The boundary conditions at inlet that can be imposed in this CFD code are total pressure, total temperature and inlet flow angle, or reduced mass flow rate, total temperature and inlet flow angle. The first set of inlet boundary conditions is used in the constrained optimization where the objective function is penalized by delta mass flow rate, while the second set is used in the optimization with no constraints, since all geometry and flow requirements are explicitly specified.

6. RESULTS AND DISCUSSION

The optimization algorithms, presented in the previous sections, were used to design several transonic cascades. Since the algorithms behavior for these different cascades was similar, only one of these cases in hereby presented, that of an impulse turbine cascade, see Ref. [10], which is originally transonic; and it was redesigned for minimum total pressure loss.

The optimization parameters for the GA and SA algorithms were studied and were properly tuned for this case.

6.1 A transonic turbine cascade

This is an impulse cascade with sharp leading and trailing edges. The spacing to chord ratio is 0.526 and the thickness distribution assumes a parabolic profile with a maximum thickness to chord ratio of 21.45% occurring at mid-chord. The inlet flow angle is 40.63° and the ratio of exit static to inlet total pressure is 0.833.

Figure 4 gives the optimization history for the GA and SA. For the GA, the objective function is reduced from 0.0043 to 0.0025 in six generations, each consisting of 30 individuals, hence involving 180 CFD calls in all. For about the same objective function reduction (from 0.0043 to 0.0026), the SA algorithm used about 118 function evaluations to eliminate the shock.

A single flow field analysis took around eight minutes (when starting from a converged solution obtained for a given candidate of the previous generation) with a CFL number of 4 on a mesh with 2800 points.

Figures 5 and 6 give the Mach number profiles for the original and the GA-optimized blade, where the flow over the original cascade exhibits a shock around mid-chord; this shock has been eliminated in the optimized profile. Figures 7 and 8 give the Mach number profiles for the original and SA-optimized blade.

Note that the inlet and exit Mach numbers are about the same in the original and optimized cascades. In both schemes, it was possible to eliminate the shock by modifying the blade camber line. Figure 9 shows that the changes in shape between the original and the redesigned blades are relatively small, which would be difficult to achieve by manually changing the blade shape. Note also that both designs have a reversed curvature along the blades suction side so that the flow can compress reversibly. Other researchers [10,11], who were using inverse design methods, have also observed similar behavior.

7. CONCLUSIONS

A robust aerodynamic shape optimization method that uses two global optimization algorithms namely, GA and SA, has been developed and demonstrated. The objective function is evaluated using a 2D Euler solver. The blade geometry was approximated using NURBS, which provide great flexibility and accuracy. The GA has better power in exploring the design region and best suited for large scale problem where the number of design variables and the size of the design space are relatively large, while the SA method seems to be more accurate and quick for relatively low number of design variables. The design optimization algorithm was applied to redesigning a two-dimensional cascade to be retrofitted into a turbomachine, with the objective of minimizing the losses (by eliminating any existing passage shocks) for the same mass flow rate, same thickness distribution and the same overall flow turning (work load). The optimized blades were shock-free transonic blades and the optimization process took a relatively short number of function evaluations, each involving a CFD calculation.

REFERENCES

1. Dennis, B.H., Dulikravich, G.S., and Han, Z.-X., 1999, “Constrained Shape Optimization of Airfoil Cascades Using a Navier-Stokes Solver and a Genetic/SQP Algorithm,” ASME Paper No. 99-GT-441.

2. Wang, X., and Damodaran, M., 2000, “ Aerodynamic Shape Optimization Using Computational Fluid Dynamics and Parallel Simulated Annealing Algorithms”, AIAA-2000-4847.

3. Oyama, A., Liou, M.-S., and Obayashi, S., 2002, “Transonic Axial-Flow Blade Shape Optimization Using Evolutionary Algorithm and three Dimensional Navier-Stokes Solver,” AIAA 2002-5642.

4. Piegl, L., and Tiller, W., 1995, “The NURBS Book,” Springer, 1995.

5. Goldberg D. E., 1985, “Genetic Algorithms in Search, Optimization and Machine learning,” Addison-Wesley, 1985.

6. Gen, M., Cheng, R., 1997, “Genetic Algorithm and Engineering Design,” Wiley, 1997.

7. Kirkpatrick, S., Gelatt Jr., C.D., and Vecchi, M.P., 1983, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671-680, 1983.

8. Corana, A., Marchesi, M., Martini, C., and Ridella, S., 1987, “Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm” ACM Transaction on Mathematical Software, Vol. 13, No. 3, pp. 262-280, 1987.

9. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., 1953, "Equation of State Calculations by Fast Computing Machines", J. Chem. Phys., Vol. 21, No. 6, pp. 1087-1092, 1953.

10. Ahamdi, M., and Ghaly, W., 1998, “Aerodynamic Design of Turbomachinery Cascades using a Finite Volume Method on Unstructured Meshes", Journal of Inverse Problems in Engineering,” Vol. 6, pp. 281-298, 1998.

11. Dang, T., 1995, “Inverse Methods for Turbomachine Blades using Shock-Capturing techniques”, AIAA Paper No. 95-2465.

12. Fottner, L., 1990, “Test Cases for Computation of Internal Flows in Aero Engine Components,” AGARD-AR-275, Propulsion and Energetics Panel, 1990.

13. Ahamdi, M., and Ghaly, W., 1998, “ A Finite Volume Method for Inviscid Transonic Cascade Flow with Solution Adaptation on Unstructured Meshes” Vol. 44, No. 3, CASI Journal, pp175-181.

14. Kalyanmoy Deb, Ashish Anand and Dhiraj joshi, 2002, “ A Computationally Efficient Evolutionary Algorithm for Real-Parametre Optimization” KanGAL Report Number 2002003.

15. Fogel, D. B. and Beyer, H.-G. 1995, “A note on the Empirical Evaluation of Intermediate Recombination. Evolutionary Computation Journal, pp. 491-495.

Table 1. The SA result for the Rosenbrock function

	ROSENBROCK PROBLEM # OF FUNCTION EVALUTAIONS (Absolute error <10^{-20})	
	PRESENT WORK	LITERATURE [14]
20 DV	1,174,000	1,396,496
30 DV	1,764,000	3,719,887
40 DV	2,424,000	---

Table 2. The computation result of SA and GA for the Rastrigin function

	RASTRIGIN PROBLEM 20,30,40 Design Variables (DV)			
	SA		GA	
Cases	Optimal Value	# of function	Optimal Value	# of function
20 DV	2.98	5,601	10^{-14}	592,000
30 DV	2.98	8,401	10^{-14}	278,900
40 DV	4.97	16,801	10^{-14}	318,700

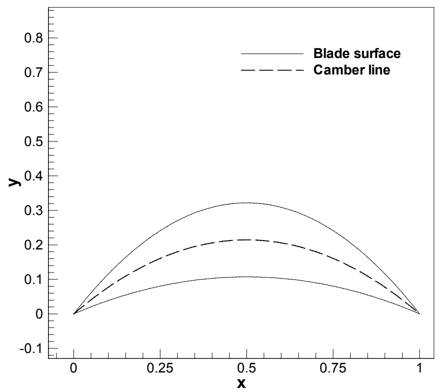

Figure 1. Blade geometry description

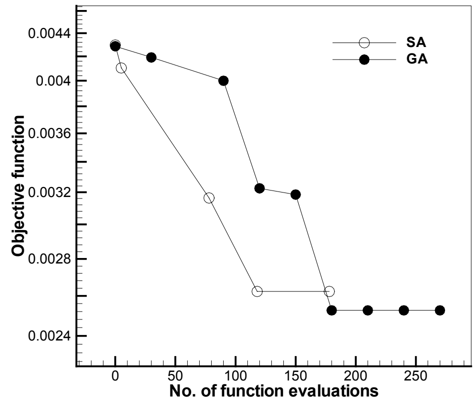

Figure 4. Convergence history of Impulse Turbine Cascade

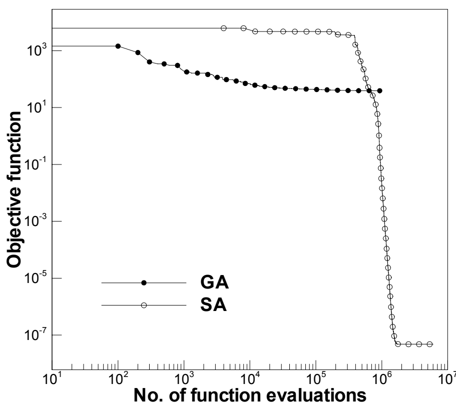

Figure 2. Convergence history for the Rosenbrock test function with 40 design variables

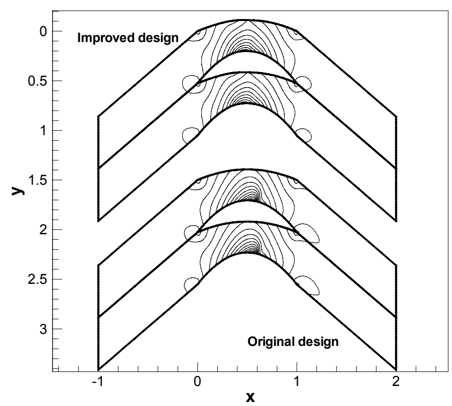

Figure 5. Isentropic Mach contours for Impulse Turbine Cascade using GA.

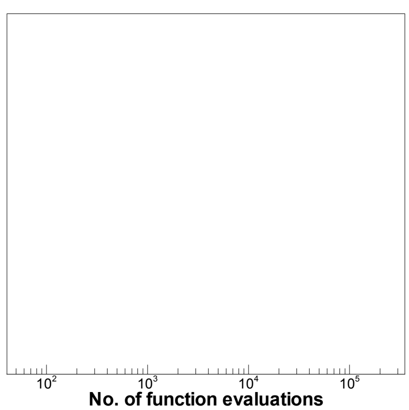

Figure 3. Convergence history for the Rastrigin test function with 40 design variables

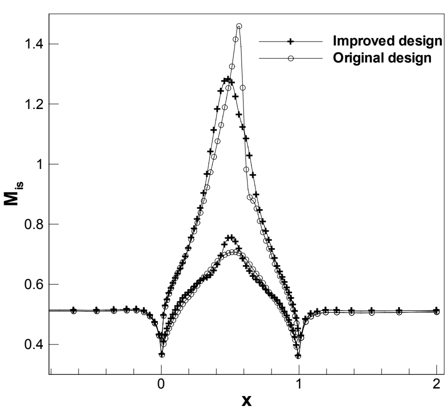

Figure 6 Isentropic Mach number along the blade surfaces for the impulse turbine using GA

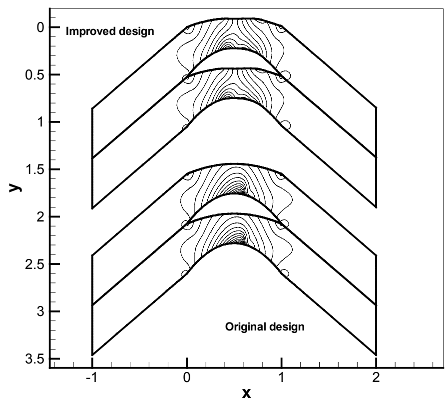

Figure 7. Isentropic Mach contours for Impulse Turbine Cascade using SA

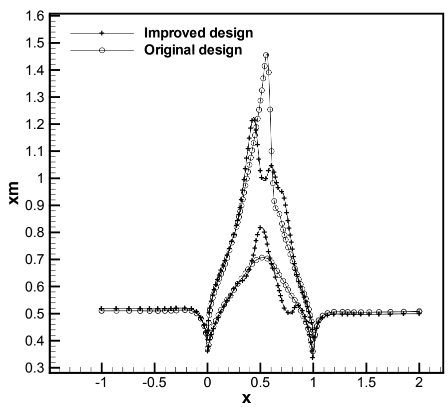

Figure 8 Isentropic Mach number along the blade for impulse turbine using SA

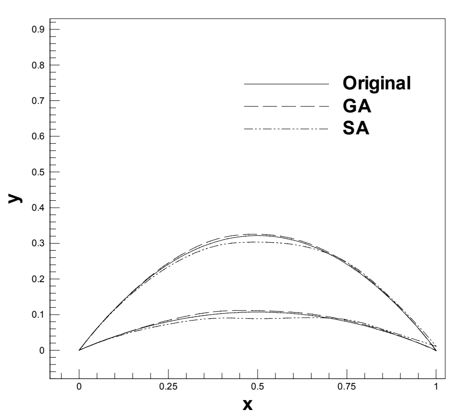

Figure 9 Original and redesigned blade profiles