Proceedings of
ASME TURBO EXPO 2003
June 16-19, 2003, Atlanta, Georgia, USA

Draft GT-2003-38408

OPTIMIZATION OF AN INTERNAL COOLING SYSTEM FOR TURBINE NOZZLES BY USE OF GENETIC ALGORITHMS

K. Funazaki and C.F.F. Favaretto
Department of Mechanical Engineering
Iwate University
Morioka, Japan

ABSTRACT

This paper shows an attempt to optimize an internal cooling system of turbine nozzles. The optimizing technique adopted here is developed based on Genetic Algorithms. The cooling system to be optimized is an impingement cooling combined with pin-fin cooling. The optimization is performed for several design parameters such as the impingement and discharging hole diameters, pin diameter and pin height. The computational grid is automatically generated and boundary conditions prescribed. A commercial *CFD* code is used to evaluate the target function, which is defined as the ratio between the averaged heat transfer coefficient multiplied by the wetted area and the pressure loss.

INTRODUCTION

Thanks to the state-of-the art of cooling technologies such as impingement cooling or film cooling, turbine inlet temperature (*TIT*) of modern gas turbines has exceeded more than 1773K. The next target of turbine cooling designers is to drastically reduce cooling air consumption because thermal efficiency of the gas turbines are now saturating due to the large amount of cooling air mainly used in the turbine section. To meet this goal, one of the present authors investigated heat transfer characteristics of a cooling device that simply combines impingement cooling with pin-fin cooling in order to disturb the impingement jets on target plate but also to enhance the internal surface area (Funazaki et al. [1][2]). Very recently, Funazaki and Hachiya [3] have carried out detailed numerical simulations on this integrated cooling device to clarify effects of several dominating geometrical parameters of the device upon its heat transfer characteristics and pressure loss. Their approach has simply changed pin-height, pin pitches and off-set of the pin location. Although some information useful for effective turbine cooling is obtained from this study, it is also clear that more systematic methods should be pursued for maximizing attainable average heat transfer inside the device or optimizing heat exchange performance in consideration of the pressure loss. The present study has adopted Generic Algorithms (*GA*) as an optimizing tool.

The use of *GA* for multi-parameter optimization has become a popular technique in many engineering fields, such as aerodynamic design of blade profiles or wings [4-6] and, more recently, gas turbine related applications [7-11]. The reason for adopting such technique is found by the fact that it is robust, simple to implement and innovative. Conventional optimization methods such as “hill climbing” algorithms search in one direction of the domain only and are strongly limited to well behaved target functions. The *GA*, on the other hand, can handle complex non-linear target functions and provides a multi-directional search, avoiding premature convergence at local peaks, which may not represent the global maximum of the search domain.

The simplicity in translating the *GA* method into a computational code is also one of its great advantages. A *GA* code consists of basic mathematical operations which can be parallelized in a very straightforward manner. Among many authors, Goldberg [12] described the basics about *GA*, including sample program lists.

The aim of the present paper is to show the development of an optimization tool based on *GA*. The authors believe that this study is one of the first attempts to use *GA* for the optimization of internal cooling system for turbine nozzles.

NOMENCLATURE

Abbreviations

AMG		algebraic multi-grid

Symbols

A	[m^2]	area
c_p	[kJ/kg·K]	specific heat at constant pressure
H	[mm]	height

h	$[W/m^2 \cdot K]$	heat transfer coefficient
k	$[W/m \cdot K]$	thermal conductivity
k	$[m^2/s^2]$	turbulent kinetic energy
P	[Pa]	static pressure
R	[mm]	radius
T	[K]	temperature
Tu		turbulence intensity
Δ		variation
η		efficiency
ρ	$[kg/m^3]$	density
μ	$[kg/s \cdot m]$	molecular viscosity

Subscripts

D	discharge (hole)
I	impingement (hole)
P	pin
w	wetted (area)

Superscripts

—	area average

PROBLEM DESCRIPTION

The model for the current study (Fig. 1) consists of one half of the actual cooling system, assuming that symmetry condition is valid for the side boundaries. The flow is injected from the impingement hole, encounters the target plate and pin surface, partially convects through the symmetric boundaries and partially exits the domain at the discharge hole.

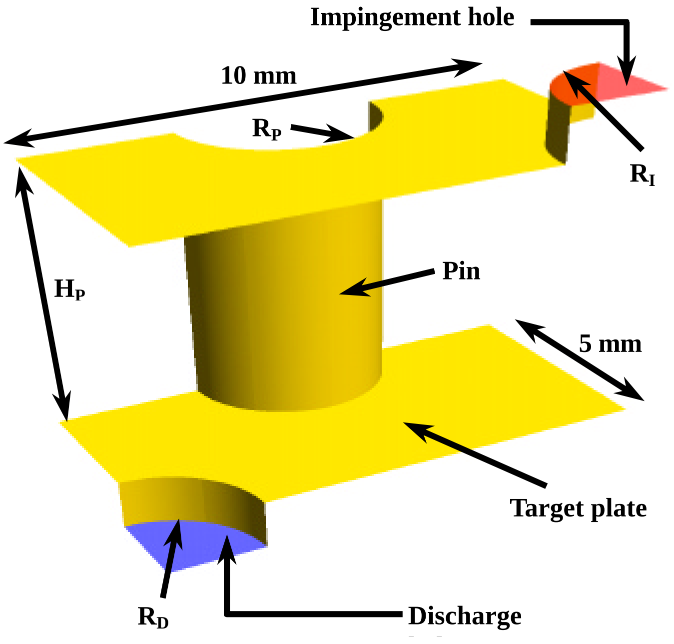


Figure 1 - Internal cooling system diagram.

The design parameters for this particular optimization problem are the pin height (H_P), the pin radius (R_P), and impingement and discharge hole radii (R_I and R_D respectively). Table 1 presents the constraints adopted for H_P, R_P, R_I and R_D. For the present study, R_I and R_D were considered as one design variable.

Table 1 - Constraints adopted for the design parameters (all dimensions in millimeters).

Impingement, discharge hole radius	10 $R_I = R_D$ 20
Pin radius	10 R_P 20
Pin height	10 H_P 80

The optimization problem has one target function, defined as efficiency (Eq. 1). The objective is to maximize the efficiency within the constraints prescribed for the design variables.

$$\eta = \frac{\bar{h} A_w}{\Delta P} \tag{1}$$

where: η is the efficiency, $\bar{h}$ the pin area averaged heat transfer coefficient, A_w the wetted area (pin surface and target plate) and ΔP the pressure difference between the impingement and discharge holes.

OPTIMIZATION TOOL

Genetic Algorithms

The *GA* is a powerful optimization tool based on the theory of evolution, which means that the "best fit" individuals in one generation survive. The "fitness" in *GA* is the function to be optimized (target function) and the parameter set or problem variables are denominated a "chromosome". In the present study, the real value of each design parameter is encoded as a string of binary digits. For instance, string 111000 refers to $R_I = R_D = 18.80$ mm, string 000001 to $R_P = 10.15$ mm and string 001100 to $H_P = 23.33$ mm. The strings for each one of the parameters are blended into a large string, forming the chromosome 111000000001001100. The *GA* works with a number of chromosomes for each iteration or generation, providing a search in multiple directions of the domain simultaneously. Preliminary tests with *GA* code here described showed that a constant population size of 30 chromosomes was suitable for the optimization task.

The starting point for an optimization using *GA* is a process called *initialization*. The initial population of chromosomes can be generated automatically by invoking a pseudo-random number subroutine, usually available in computer language compilers. The *GA* converges to the same result independent of the starting population. According to Trigg et al. [6], a given start with specified values for the chromosomes does not seem to be an important feature since initial convergence is rapid.

After the initial population has been created, the *GA* needs to evaluate the fitness for each one of the chromosomes belonging to the initial generation. This means judging how well each chromosome is performing according to their *phenotype* (design parameters). The tool used to evaluate the fitness is transparent to the *GA* code and does not need to embedded in the program. This is a remarkable feature which allows *GA* to be applied in almost any research field. For the present optimization task, the commercial *CFD* code *CFX4.4* (*AEA Technology, Ltd.*) was used for calculating the heat transfer coefficient on the pin surface and pressure drop between the impingement and the discharge holes. These values were then used in Eq. 1 to evaluate the efficiency η.

With the fitness values for the chromosomes of the initial generation calculated, the *GA* code must select the candidates for mating. Since in the present work the population size was kept constant, the number of chromosomes to be elected for reproduction is half of the population size. This process is called *selection* and it is directly related to the performance of a *GA* code. Several methods for accomplishing this task are described in the literature [4,5,12,13]. The most common method is called *roulette wheel*, in which all chromosomes of a population share a certain sector of a wheel, proportional to their fitness. The wheel is then spun and the chromosome selected. This method, however, causes premature convergence of the results. The best fit chromosome tends to dominate the others and cause their early extinction. The *GA* will follow one direction, leaving other possible maximum locations behind. Considering these limitations, the *tournament selection* method was chosen instead. This method randomly picks two or more chromosomes (depending on the tournament size) from the population. The chromosomes then compete against each other to decide which one will be on the mating pool. The chromosome with the highest fitness will be the winner in the case of a maximization problem. This process is repeated to elect the other chromosome to join the mating pool. Care must be taken in order to avoid the selection of the same chromosome twice.

After selecting the eligible chromosomes for reproduction, the *recombination* process is performed. The chromosomes mate (parents), generating two new chromosomes (children) in order to keep the population size constant. The *genes* or the string bits are exchanged between the children at a probability of 0.6, commonly found in the literature [12]. Like the selection process, there are many ways to perform recombination [5,13]. The one-point crossover was chosen for the present paper. In this recombination method a bit location is randomly chosen. The information from the bit location just after the selected bit to the end of the string is exchanged between child1 and child2. For instance, suppose the selected bit location is 12, child1 = 100100011110001100 and child2 = 100011111010101101. After crossover, child 1 would become 100100011110101101 and child2 100011111010001100.

The tournament selection method was found to have one major drawback. It does not guarantee by itself that the best individual in the population will be picked for the tournament. The preliminary part of the tournament is totally random and does not take into account the fitness values. An additional operator was found necessary to be implemented. The *elitism* model was implemented in order to assure that the best individual is maintained from one generation to the other. The best individual is selected, cloned and added to the other children.

In order to keep the search in many directions of the optimization domain, *mutation* was also implemented. The children, except the cloned ones, are selected at random. Mutation occurs in one or more bits of the selected children by replacing 0 for 1 or 1 for 0, with a probability of 0.0333 [12].

After the four main operations (selection, recombination, elitism and mutation) have been completed the children replace the current population of chromosomes. The process is repeated until the convergence criteria has been satisfied.

Table 2 shows a summary of all *GA* parameters adopted in the present paper.

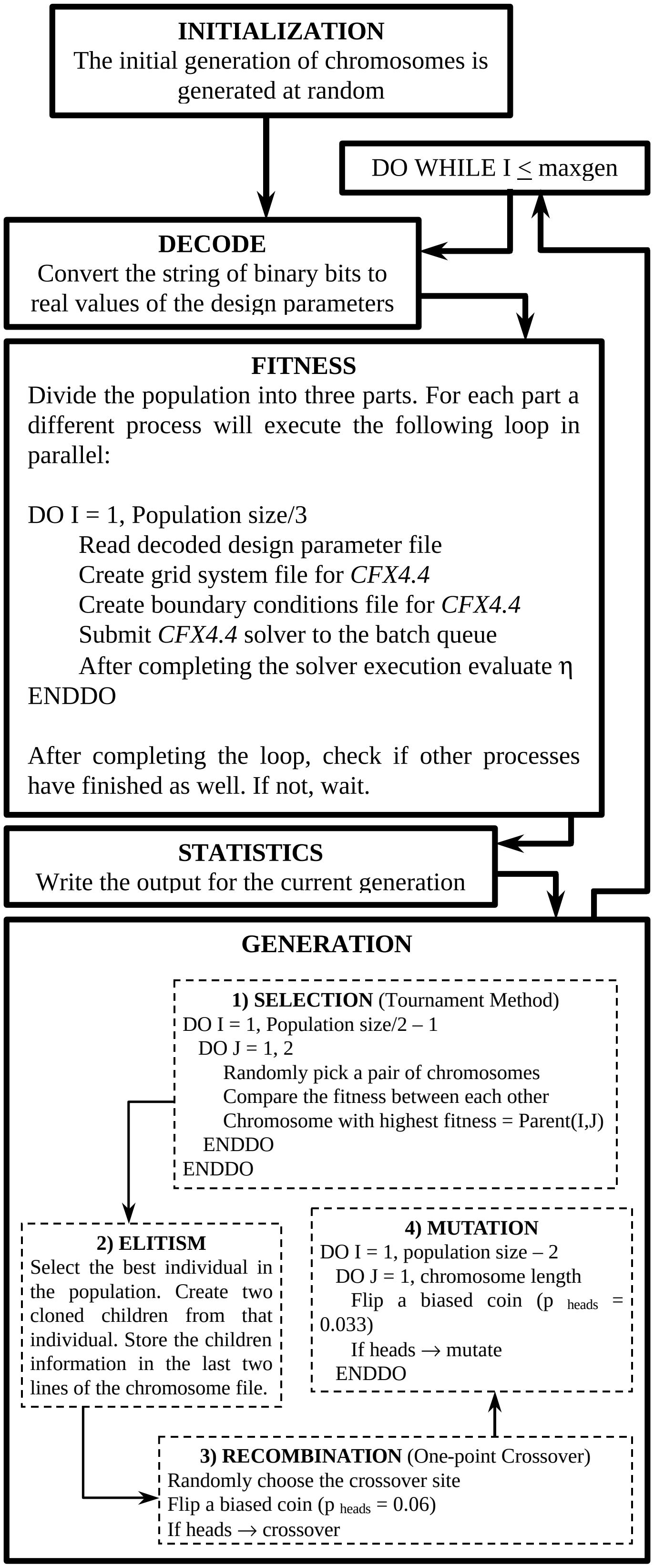
Table 2 - Genetic algorithm parameters.

Chromosome length	18
Type of coding	binary
Population size	30 (constant)
Selection method	tournament
Tournament size	2
Recombination method	one-point crossover
Crossover probability	0.6
Mutation probability	0.0333

Optimizer Implementation

The *GA* code here described was developed in a hybrid *FORTRAN/UNIX* shell script language. This technique was used so that the *CFX4.4* code could be combined with the optimizer. Figure 2 presents a flowchart describing the mechanism of the code.

One of the critical problems when using a *GA* code is the *CPU* time. The fitness has to be evaluated for all chromosomes belonging to every generation. Thus, for a population of 30 chromosomes, after 30 generations, the flow solver would have been called 900 times. If each one of the solver executions are performed sequentially, the *CPU* time would be 900 multiplied by the time required for each run. In order to provide faster results a simple parallel processing strategy is proposed (Fig. 2, *FITNESS* text box). For each generation the population is divided into three parts. Each one of the parts would be treated as a distinct process, running in parallel. For instance, part I would evaluate the fitness from chromosomes 1 to 10, part II from chromosomes 11 to 20 and part III from 21 to 30. Ideally speaking, this would reduce the *CPU* time required to solve one generation by a factor of 3. A *UNIX* shell script was developed to synchronize the executions. Parts I thru III should start and finish at the same time so that the other parts of the *GA* code can be correctly performed. If one of the parts finishes before the others it will wait until all parts have been completed. Executions were successfully performed in parallel with minimum overhead.



INITIALIZATION
The initial generation of chromosomes is generated at random

DO WHILE I ≤ maxgen

DECODE
Convert the string of binary bits to real values of the design parameters

FITNESS
Divide the population into three parts. For each part a different process will execute the following loop in parallel:

```
DO I = 1, Population size/3
    Read decoded design parameter file
    Create grid system file for CFX4.4
    Create boundary conditions file for CFX4.4
    Submit CFX4.4 solver to the batch queue
    After completing the solver execution evaluate η
ENDDO
```

After completing the loop, check if other processes have finished as well. If not, wait.

STATISTICS
Write the output for the current generation

GENERATION

1) SELECTION (Tournament Method)

```
DO I = 1, Population size/2 – 1
  DO J = 1, 2
      Randomly pick a pair of chromosomes
      Compare the fitness between each other
      Chromosome with highest fitness = Parent(I,J)
  ENDDO
ENDDO
```

2) ELITISM
Select the best individual in the population. Create two cloned children from that individual. Store the children information in the last two lines of the chromosome file.

4) MUTATION

```
DO I = 1, population size – 2
  DO J = 1, chromosome length
    Flip a biased coin (p_heads = 0.033)
    If heads → mutate
  ENDDO
```

3) RECOMBINATION (One-point Crossover)

```
Randomly choose the crossover site
Flip a biased coin (p_heads = 0.06)
If heads → crossover
```

Figure 2 - Genetic algorithm optimizer flowchart.

NUMERICAL SIMULATION

Grid Generation

A FORTRAN code was developed by the authors in order to automate the computational grid generation. For every chromosome, the code reads the decoded real values for the design parameters, generates a multi-block grid system and exports it to the *CFX-4.4* solver through an *ASCII* file. Figure 3 shows a mesh for $R_I = R_D = 20$ mm, $R_P = 20$ mm and $H_P = 50$ mm.

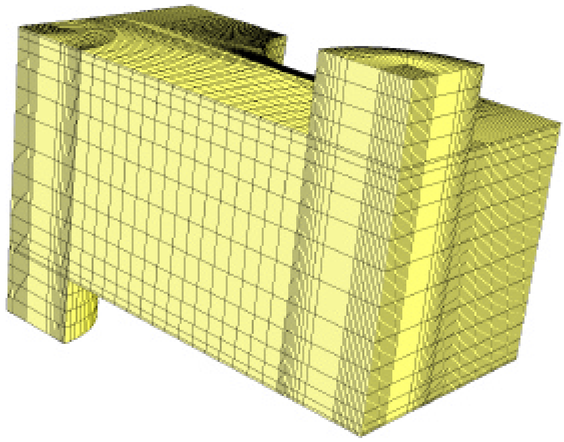


Figure 3 - Grid system.

The grid system shown in Fig. 3 was highly optimized in order to reduce the number of nodes. Several coarse grids were run and results were compared to finer grid ones and experiments. After a reasonable agreement was found the final configuration was defined. The number of grid points was kept constant (59,520) so that the *CPU* time for one run would be approximately one hour.

Computational Code

The three-dimensional, steady, Reynolds-averaged, incompressible Navier-Stokes and energy equations were solved with the finite difference *CFX-4.4* computational code. A second-order differencing scheme using a body-fitted coordinate system was employed based on the Rhie-Chow algorithm [14]. That is, the code performs curvilinear transformations to map the complex flow domain in physical space to a simple (rectangular) flow domain in computational space. A non-staggered grid is used when solving the velocity components of the momentum equations. Considering the velocity-pressure algorithm, the *SIMPLEC* method was adopted. In spite of the implementation of Rhie-Chow algorithm, checkerboard oscillations in the velocity and pressure usually associated with non-staggered grids was eliminated. All equations were solved using the Algebraic Multi-grid (*AMG*) method, described by Lonsdale [15]. This method solves the discretized equations on a series of coarsening meshes, internally produced by *CFX-4.4*. Detailed information on the theoretical basis of the software can be found in the *CFX-4.4 Solver* documentation [16].

Convergence was improved by specifying 6 inner iterations for the energy equation (expressed in terms of enthalpy), 2 iterations for the turbulence equations and one iteration for the other variables. The number of outer iterations was found to be the most critical aspect related to the compromise solution between good accuracy and short computational time. After several trials it was concluded that 500 outer iterations was a good measure.

Boundary Conditions

For the impingement hole surface, normal velocity was calculated so that the Reynolds number (based on the inlet hydraulic diameter) would equal 10,000. The other quantities prescribed at the inlet were temperature T_I = 323K, turbulence intensity Tu_I = 3.7% and dissipation length scale e = 0.01. Non-slip boundary conditions were applied to the top wall, the target plate, the pin and the surfaces contouring the impingement and discharge holes (yellow surfaces in Fig. 1). The temperature on these surfaces was assumed as constant and equal to 303K. At the discharge hole, mass flow was prescribed. On all side boundaries (except the pin) symmetry condition was applied. The working fluid used in the calculations was air at a reference temperature of 288 K (molecular viscosity m =1.969 x 10^{-5} kg/s·m, density r = 1.088 kg/m^3, thermal conductivity k = 0.02759 W/m·K, specific heat at constant pressure c_p = 1.008 x 10^3 kJ/kg·K).

Turbulence Model

The Menter modified low Reynolds number *k-w* model was employed. In this model, the equations switch from the standard *k-w* model close to the walls to equations equivalent to the *k-e* model away from the walls, but for independent variables *k* and ***w***.

RESULTS

The *GA* code solved 29 generations in 11.5 days using the Origin 3800 machine at the Supercomputing Center of Iwate University. The convergence history shown in Fig. 4 indicates that the best fit individual was first born in generation 22, with an outstanding performance of h = 0.010946. The optimizer keeps searching in the domain for other possibilities but saturates, as shown in the graph by the straight line from generation 22 to 29.

Figure 5 presents the search domain for each one of the design parameters. These graphs contain all points searched during the whole execution of the code. Thus, Fig. 5 provides a report on how well the search domain was explored by the *GA* and the tendency of the individuals to develop themselves according to the best fit chromosome. The red circles indicate the location of the maximum point.

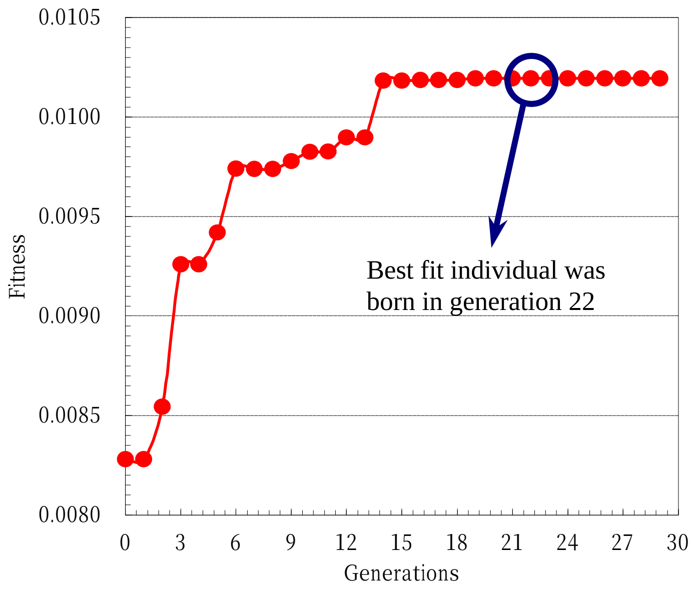
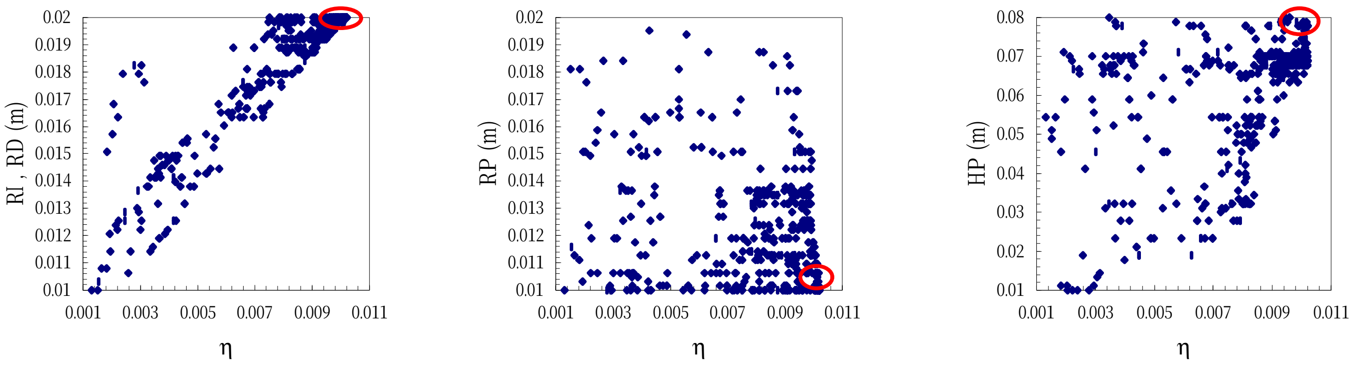


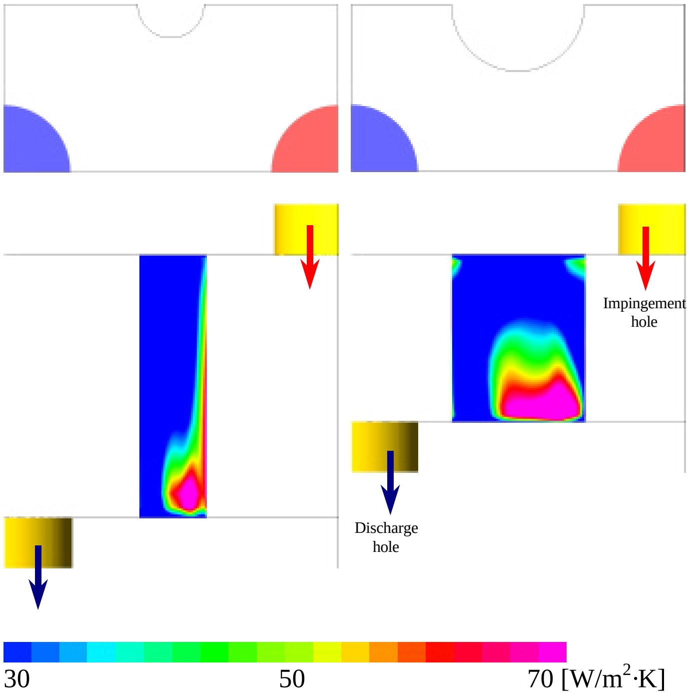
Figure 4 - Convergence history (Maximum fitness).



a) Impingement, discharge hole radius b) Pin hole radius c) Pin height

Figure 5 - Search domain for each one of the design parameters (red circles indicate the location of the best fit individual).

The best fit individual corresponds to a cooling system with $R_I = R_D = 20$ mm, $R_P = 10$ mm and $H_P = 77.7$ mm. For the given set of constraints the optimum efficiency would be obtained for a long pin with a small radius and large impingement and discharge holes. The efficiency was defined by correlating the heat transfer coefficient, the wetted area and the pressure drop. Therefore, the geometrical configuration which provides the maximum efficiency will not necessarily have the highest heat transfer coefficient on the pin surface. Gas turbine designers are usually seeking for a cooling system which provides the highest heat transfer between the pin surface and the impinging flow, but are also concerned about reducing the loss.



a) Optimized b) Datum

Figure 6 - Comparison between the heat transfer coefficient for the optimized and datum configurations.

Figure 6 presents a comparison between the heat transfer coefficient h for the optimized and datum ($R_I = R_D = 20$ mm, $R_P = 20$ mm and $H_P = 50$ mm) configurations. The surface averaged h indicates a negligible favorable tendency for the datum configuration (Table 3). This trend is overturned when efficiency is calculated, showing an improvement of almost 32% from the original design to the optimized one. It is more clearly understood if one takes into account that the wetted area for the optimized case is almost 20% larger for the optimized case and the pressure drop around 30% smaller than the datum. In both cases, the impingement hole radius is exactly the same, i.e., the magnitude of the velocity prescribed at the inlet does not change. The outstanding performance of the optimized case can be attributed to the small pin radius, which reduces the blockage effect of the pin to the impinging flow. The larger pin height is an attempt of the optimizer to retain the same surface area as the datum, thus maintaining h as high as possible.

Table 3 - Comparison between the performance of the optimized configuration with the datum.

	Optimized	**Datum**
Efficiency (η)	0.010946	0.007448
Pin surface h	28.46 W/m^2·K	28.51 W/m^2·K
ΔP	19.46 Pa	27.56 Pa

CONCLUDING REMARKS

The present study described the development of an optimization tool and its application to the design of an internal cooling system for turbine nozzles. The genetic algorithms have proven to be a useful tool for tackling the challenging design tasks of modern gas turbines. The development of a hybrid *FORTRAN*/*UNIX* shell script program enabled the authors to combine the *GA* code with a commercial software.

The maximum efficiency was obtained for a configuration with large impingement and discharge hole radius, large pin height and small pin radius. The large pin height was an attempt of the optimizer to retain the heat transfer coefficient values from the datum design by increasing the pin area while the small pin radius was beneficial for reducing the pressure drop.

The methodology adopted in this study may be applied to many other practical engineering problems. However, the authors acknowledge the necessity of improvements in the code. A multi-objective *GA* will be developed in the near future in order to extend the range of applications of the present code to more sophisticated analysis such as fluid-structure interaction problems.

ACKNOWLEDGMENTS

The second author acknowledges the financial support of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

[1] Funazaki, K., Imamatsu, N. and Yamawaki, S., 1999, "Heat Transfer Measurements of an Integrated Cooling Configuration Designed for Ultra-High Temperature Turbine Blades", Proceedings of the International Gas Turbine Congress 1999 Kobe, Vol. II, pp. 833-839.

[2] Funazaki, K., Tarukawa, Y., Kudo, T, Matsuno, S., Imai, R. and Yamawaki, S., 2001, "Heat Transfer Characteristics of an Integrated Cooling Configuration for Ultra-High Temperature Turbine Blades: Experimental and Numerical Investigations". ASME Paper 2001-GT-148.

[3] Funazaki, K. and Hachiya, K., 2003, "Systematic Numerical Studies on Heat Transfer and Aerodynamic Characteristics of Impingement Cooling Devices Combined with Pins," ASME Paper GT-2003-38256.

[4] Vicini, A., Quagliarella, D., 1997, "Inverse and Direct

Airfoil Design Using a Multiobjective Genetic Algorithm", AIAA Paper, Vol. 35, No. 9, pp. 1499-1505.

[5] Obayashi, S., 2000, "Multiobjective Evolutionary Computation for Supersonic Wing Design", In Périaux, J., Degrez, G., Deconinck, H., editors, "Genetic Algorithms for Optimisation in Aeronautics and Turbomachinery", Lecture Series 2000-07, von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium.

[6] Trigg., M.A., Tubby, G.R., Sheard, A.G., 1999, "Automatic Genetic Optimization Approach to Two-Dimensional Blade Profile Design for Steam Turbines", ASME J. Turbomachinery, Vol. 121, pp. 11-17.

[7] Akmandor, I.S., Öksüz, O., Gökaltun, S., Bilgin, M.H., 2002, "Genetic Optimization of Steam Injected Gas Turbine Power Plants", ASME Paper GT-2002-30416.

[8] Benini, E., Toffolo, A., 2002, "Towards a Reduction of Compressor Blade Dynamic Loading by Means of Rotor-Stator Interaction Optimization", ASME Paper GT-2002-30396.

[9] Elliot, L., Ingham, D.B., Kyne, A.G., Mera, N.S., Pourkashanian, M., Wilson, C.W., 2002, "The Optimisation of Reaction Rate Parameters for Chemical Kinetic Modelling Using Genetic Algorithms", ASME Paper GT-2002-30092.

[10] Sampath, S., Gulati, A., Singh, R., 2002, "Fault Diagnostics Using Genetic Algorithm for Advanced Cycle Gas Turbine", ASME Paper GT-2002-30021.

[11] Ferreira, S., Pillidis, P., Widell, H., "Optimization of Biomass Fuelled Gas Turbines Using Genetic Algorithms", ASME Paper GT-2002-30131.

[12] Goldberg, D.E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, Reading, Massachusetts, USA.

[13] Krishnakumar, K., 2000, "Building Blocks of Evolutionary Algorithms", In Périaux, J., Degrez, G., Deconinck, H., editors, "Genetic Algorithms for Optimisation in Aeronautics and Turbomachinery", Lecture Series 2000-07, von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium.

[14] Rhie, C.M., Chow, W.L., 1983, "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation", AIAA Paper, Vol. 21, pp. 1527-1532.

[15] Lonsdale, R.D., 1993, "An Algebraic Multi-grid Solver for the Navier-Stokes Equations on Unstructured Meshes", Int. J. Num. Meth. Heat an Flow, 3.

[16] CFX-4.4 Solver, 1999, AEA Technology.