

Multi-Disciplinary Hybrid and Evolutionary Optimization

George S. Dulikravich*

Multidisciplinary Analysis, Inverse Design and Optimization (MAIDO) Center
MAE Dept., Box 19018, The University of Texas at Arlington, Arlington, TX 76019, U.S.A.
e-mail: gsd@mae.uta.edu

Brian H. Dennis

Institute of Environmental Studies, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
e-mail: dennis@garlic.q.t.u-tokyo.ac.jp

Thomas J. Martin

Turbine Module Center, Pratt & Whitney Aircraft Company
400 Main St., MS 169-20, East Hartford, CT 06108, U.S.A.
e-mail: martintj@pweh.com

Igor N. Egorov

IOSO Technology Center, Milashenkova 10-201, Moscow 127322, Russia
e-mail: optim@orc.ru

Key words: hybrid optimization, multidisciplinary design optimization, genetic algorithms

Abstract

When solving multi-disciplinary optimization problems the most important issue is the robustness of the optimization algorithm. Gradient-based optimization algorithms are known to terminate in a local feasible minimum that is the closest to the initial guess. Besides their inability to cope with multiple minima, gradient-based optimizers are also known for their computational inefficiency when dealing with a large number of design variables. On the other hand, non-gradient optimization algorithms like genetic algorithm, simplex method, and simulated annealing are recognized for their ability to either avoid or escape from a local minimum. However, genetic algorithms, for example, are known for their computational inefficiency especially when dealing with a relatively small number of design variables. Logical remedies are stochastic optimizers and hybrid constrained optimization with automatic switching among different gradient-based and non-gradient based optimization algorithms. Examples, illustrating the multi-disciplinary applicability of these algorithms, are given for design optimization of sizes, shapes and surface roughness of coolant flow passages, efficiency of multistage axial gas turbines, steady and unsteady flow through airfoil cascades, a magneto-hydrodynamic diffuser, and a freezing protocol for organ preservation.

1 Introduction

Numerical optimization algorithms have been known to be very sensitive, often stalling in local minima, becoming stationary on constraint boundaries, wasting computer resources, and otherwise failing to provide the best possible design (global minimum). It is very beneficial to incorporate several optimization algorithms or backup strategies so that, if one optimization method fails, another algorithm can take over [1]. Many different numerical optimization methods exist in the open literature and each behaves differently on each individual problem. Various optimization algorithms have been shown to provide faster convergence over others depending upon the size and shape of the mathematical design space, the nature of the constraints, and where they are during the optimization process. Sequential algorithms converge faster than others at different periods during the computer-automated optimization process, while they are slow and sometimes fail at achieving the ultimate objective. To achieve faster convergence and to provide greater robustness without the need for continuous monitoring of the process, the optimization algorithm should simultaneously utilize several optimization algorithms with automatic switching among them [2] or a self-adaptive search based non-gradient (semi-stochastic or stochastic) optimization should be pursued [3].

2 Hybrid Optimizer

The hybrid optimization algorithm developed by our MAIDO program incorporates some of the most popular optimization algorithms; the Davidon-Fletcher-Powell or the Sequential Quadratic Programming gradient searches, a genetic algorithm, the modified Nelder-Mead simplex method, and simulated annealing. Each technique provides a unique approach to optimization with varying degrees of convergence, reliability and robustness at different cycles during the iterative optimization procedure. A set of analytically formulated rules and switching criterion were coded into the program to automatically switch back and forth among the different optimization algorithms as the iterative minimization process proceeded [2].

The evolutionary hybrid algorithm handles the existence of equality and inequality constraint functions in three ways: Rosen's projection method, feasible searching, and random design generation. Rosen's projection method provided search directions that guided descent-directions tangent to active constraint boundaries. In the feasible search [4], designs that violated constraints were automatically restored to feasibility via the minimization of the active global constraint functions. If at any time this constraint minimization failed, random designs were generated about the current design until a new feasible design was reached.

Gradients of the objective and constraint functions with respect to the design variables, also called design sensitivities, were calculated using either finite differencing formulas, or by the much more efficient method of implicit differentiation of the governing equations [5,6]. The population matrix was updated every iteration with new designs and ranked according to the value of the objective function. During the optimization process, local minimums can occur and halt the process before achieving an optimal solution. In order to overcome such a situation, a simple technique has been devised [7,8]. Whenever the optimization stalls, the formulation of the objective function is automatically switched between two or more forms that can have a similar purpose.

The population matrix was updated every iteration with new designs and ranked according to the value of the objective function. The optimization problem was completed when the maximum number of iterations or objective function evaluations were exceeded, or when the optimization program tried all individual optimization algorithms but failed to produce a non-negligible decrease in the objective function. The latter criterion was the primary qualification of convergence and it usually indicated that a global minimum had been found.

2.1 Hybrid Optimization of Cooling Effectiveness of a Cooled Gas Turbine Blade

The hybrid design and optimization system was demonstrated on the second high-pressure turbine blade of the Pratt & Whitney F100 engine [9,10]. The geometry of the coolant passages and internal heat transfer enhancements were optimized for maximum cooling effectiveness. The external aerodynamics and coolant supply pressure were held fixed during the optimization process, while parameters such as rib turbulator height, turbulator pitch, pin fin diameter and internal rib positions were allowed to vary. This optimization test case was fully three-dimensional.

The entire three-dimensional blade was redesigned with 24 optimization design variables at five spanwise sections. The two internal ribs of the actual design were initially vertical. During the optimization, five two-dimensional design sections controlled the rib positions. The die pull angles of both ribs were also part of the design variable set in order to account for the manufacturing feasibility. The internal coolant walls were enhanced with two sets of ribbed turbulators (trip strips) for each leading and mid-body coolant passages. The strips of boundary layer turbulators were placed on the suction and pressure sides along the entire radial span of the two passages up until the first tip turn. The heights, streamwise pitches, and skew angles of each pair of trip strips were controlled by the optimization design variable set. The trailing edge coolant passage was cooled with pin fins that were shaped and positioned for cooling purposes, as well as to provide increased stiffness against vibration. In this optimization, the relative dimensions of the pin fins were fixed.

Table 1 lists the optimization design variables while their actual dimensions have been omitted because they are company proprietary. The minimum and maximum bounds on these variables are also given in the table. For example, the bounds on the turbulator heights were limited by the coolant passage height, H , between the suction and pressure sides, and the turbulator pitches were limited by a factor of turbulator heights, ε . The bounds of the rib positions were set in order to produce a geometrically feasible design. In order to set the manufacturing constraints, the coolant passage walls were filleted, draft was included in the ribs, and the die pull angles, θ_s , were constant along the span. In serial processing mode, the entire optimization process utilized a full week of computing time on a Sun Ultra60 workstation. About 630 objective function analyses were required, resulting in approximately 1800 simulations of the temperature field in the turbine blade using the BEM.

Table 1. Design variables of the F100 second high-pressure turbine blade optimization for increased cooling effectiveness [10].

Design Variable	Number in Set	Min.	Max.
Die pull angle, \square_s	1 per rib (2)	-10°	65°
Rib position, x_{rp}	5 per rib (10)	variable	variable
Turbulator height, ε	2 per cavity (4)	0.04* H	0.25* H
Turbulator pitch, p	2 per cavity (4)	5* ε	70* ε
Turbulator skew, α	2 per cavity (4)	0°	90°
Total number	24		

The optimization starting point, which was the production-version of the F100 second turbine blade design, had an integrated average cooling effectiveness of 25.85%. The cooling effectiveness of the three-dimensional optimized design was 29.7% [10]. Figure 1 shows the external wall temperature variation predicted by the coupled aero-thermo-fluid analysis. Note that the temperatures have been normalized in order to protect company proprietary information.

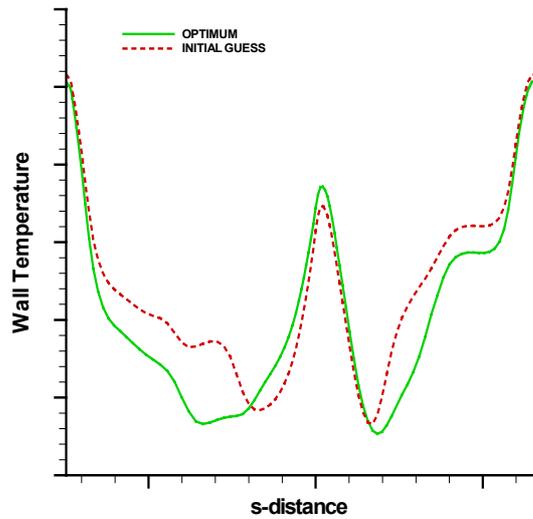


Figure 1: External blade temperature at the midspan of the second HPT blade of the F100 engine: initial (dashed curve), and optimized (solid curve) [10].

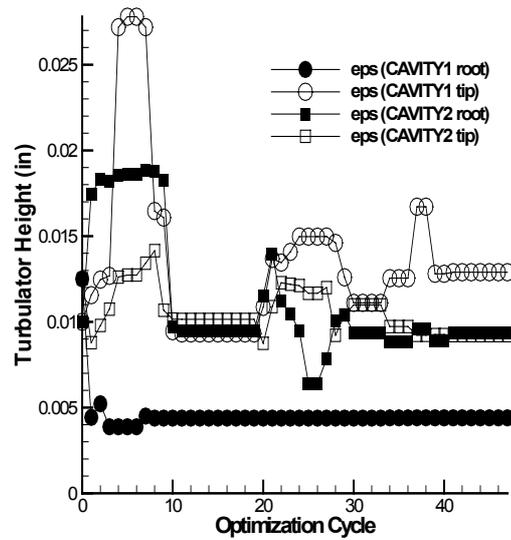


Figure 2: History of turbulator height design variables during cooling effectiveness optimization of F100 second turbine blade [10].

Figure 2 shows the history of the turbulator height design variables during the optimization run. Remember that the design variable set included four turbulator heights, ϵ , two in each coolant passage, and in each coolant passage, one constant height from the root to about the 2/3 span, and the other from 2/3 span to the tip. The former are shown as filled symbols and the latter are shown as open symbols. The turbulator heights in the leading edge coolant passage (1) are circles and the midbody coolant passage (2) are squares. The internal turbulator heights of the actual F100 second blade design were relatively tall, but the optimization algorithm reduced them to their smallest allowable value as set by the lower design variable bound. That bound was set by the range of validity of the correlation in the internal coolant flow model. Notice that the optimization algorithm reduced the leading edge root values of ϵ to its lower bound, and stayed on that lower bound, indicating that the optimizer was trying to remove that internal heat transfer enhancement. The reason for this is now clear. By removing the leading edge root turbulators, less heat will be absorbed by the blade so the coolant air would be cooler downstream. Unfortunately, the reduced internal heat transfer coefficients at the leading edge root had a penalizing effect. The stagnation point at the leading edge of the blade was hotter in the optimized configuration, and this was more pronounced at the blade root. The optimization objective was a globally integrated function, so the localized heating at the leading edge had only a small effect.

2.2 Optimization of a Multi-Stage Axial Flow Gas Turbine Efficiency

Very fast and accurate gas flow calculation and performance prediction of multi-stage axial flow turbines at design and off-design conditions can be performed using a compressible steady state inviscid axi-symmetric (through-flow) code with high fidelity loss and mixing models that account for turbulence, mixing, flow separation, etc. [11]. An example of entropy minimization (efficiency maximization) optimizes hub and shroud radii and inlet and exit flow-field for each blade row of a multi-stage axial flow gas turbine. The optimized shapes of hub and shroud indicate relatively minor differences as compared to the original shapes (Fig. 3a). The comparison of computed performance of initial and optimized designs shows significant improvement [12,13] in the optimized two-stage

turbine efficiency over the entire range of operating conditions (Figure 3b). This entire optimization process consumed less than two hours on a 500MHz processor.

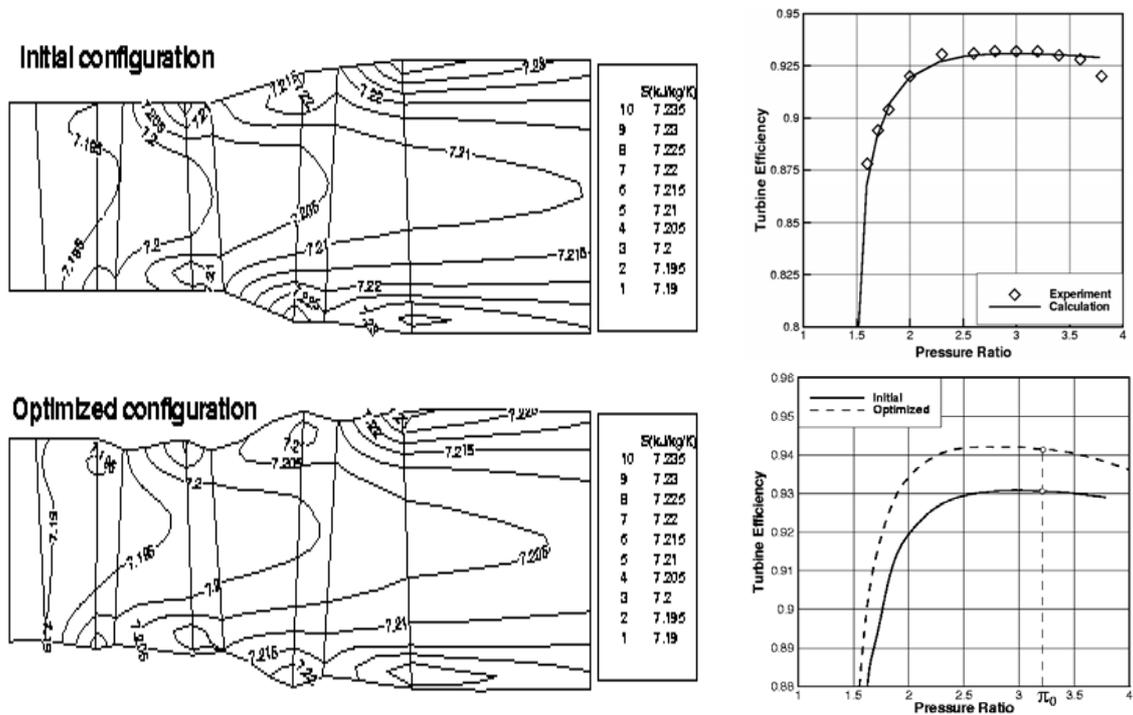


Figure 3: Two-stage axial gas turbine entropy fields and total efficiencies before and after optimization of hub and shroud shapes using a hybrid optimizer [12].

2.2 Parallel and Distributed Processing for Optimization

Constrained optimization of large, three-dimensional and complex engineering problems can be extremely slow with serial processing. The way to overcome time and size constraints is to parallelize the computational effort through multi-tasking or distributed processing. Multi-tasking involves the use of parallel compiler options and a workstation or mainframe with one or more central processing units (CPU). Distributed processing is a more explicit method of computing where several jobs are distributed across a network of workstations or personal computers. Multi-tasking usually involves shared run-time or random access memory (RAM) where the system memory spans a single address space. Distributed processing uses the RAM and hard disk space on each individual computer, called distributed memory. In shared memory multiprocessors, all of the data is accessible by all of the processors. Fast cache memories next to the processors are used in order to speed up the memory access. Cache coherency protocols are then needed to insure that all processors receive the same piece of data. Distributed memory networks are a popular architecture that is well suited to most parallel workloads. The address spaces of each processor are separate, thus communication between processors must be implemented by some form of message passing or file copying. The latter method is the easiest, but it can result in bottlenecks in the data transfer if too many processors ask for the same data at the same time. Parallel and distributed computing is a key technology in networked and high performance computer systems. By sharing the workload on an N -processor system, the optimization problem will be solved up to N times faster than a single processor system. Although an N -times speedup is difficult to achieve in practice, optimization algorithms can nearly achieve this factor because the individual jobs are nearly independent of each other.

Our hybrid optimizer, OPTRAN, was re-written to take full advantage of distributed processors over a network. Networked computers, also called clusters, have a lower cost than multi-tasking systems because they make use of existing local networks of idle CPUs. OPTRAN thus takes advantage of this kind of computer processing environment because it is like other evolutionary optimization algorithms in that it operates on a population of designs simultaneously and independently. In fact, OPTRAN was programmed specifically for a distributed parallel system to take advantage of every possible situation where parallel processing could be used. This includes parallel calculations of the gradients of the objective and constraint functions, parallel objective and constraint line searches, parallel genetic programming, parallel simplex contractions and reflections and parallel random design generation. Figure 4 illustrates the parallel architecture that is used by OPTRAN.

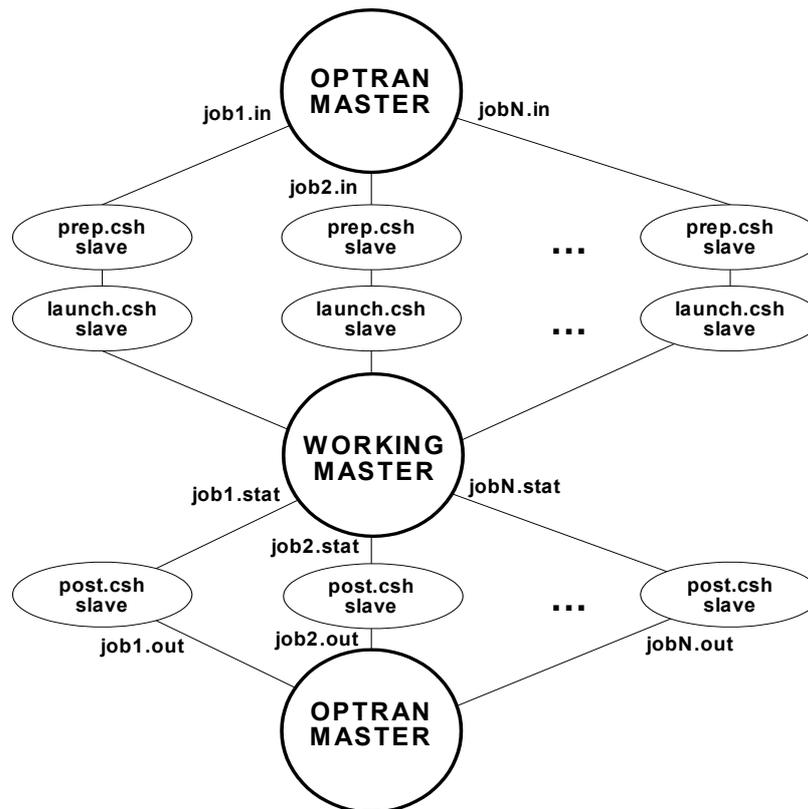


Figure 4: Parallel architecture of OPTRAN optimization algorithm.

The OPTRAN driver program was the master process that managed the design population and handed out individual processes or jobs to a number of slave processes (Figure 4). As a FORTRAN program, OPTRAN used system calls to launch UNIX shell programs. During the optimization run, OPTRAN would request one or more (N) objective and/or constraint function analyses. These requests were made parallel wherever possible. As the master processes, OPTRAN would execute N pre-processor UNIX scripts, called prep.csh. These slave process were responsible for creating a job directory on the remote workstation, copying all necessary files to that directory and executing any other (relatively quick) pre-processing tasks such as grid generation. Information about the job was stored in the job*.in file, which represents the block of distributed memory governing by the set of optimization design variables. The launch.csh shell script was forced to wait until its pre-processor task completed before it was executed. Then, slave processes with the most CPU intensive program were executed in the background. Execution control would then return to the master process while the N jobs kept

running. The working.csh master process was responsible for searching for keywords or key files in each N job directories on each remote host. This master script was responsible for counting the number of completed jobs, as well as determining if the job was successful or if it bombed, or was killed by a remoter user. A series of flags was sent back to the optimization driver program to tell what the status was of each job. This status flag indicated either running, complete, bombed or killed. If the job was killed, the working master process would wait a number of seconds before attempting to launch another identical job. Once all jobs were either completed or bombed, the master process was then be responsible for executing another series of UNIX shell commands, called post.csh. These scripts extracted information from the remote memory, executed any post-processing programs or calculations and then respond by writing N files, called job*.out. These N files contained values for the objective, F, and/or constraint functions, G_m & H_n , corresponding to the set of optimization design variables.

3 Genetic Algorithm Based Optimization

In our MAIDO Laboratory, we have developed a classical genetic optimizer that utilizes real numbers instead of binary number logic and that enforces constraints using Rosen's projection method [4]. We have also developed a parallelized micro-genetic algorithm [13,14] that enforces constraints using sequential quadratic programming [15]. Here are some illustrative examples of the results of both of these genetic-based optimizers.

3.1 Optimum Star-Shaped Hypersonic Missile

Specifically, the geometry of an axisymmetric body was optimized to reduce compression wave drag at zero angle of attack at hypersonic speeds and zero angle of attack. Optimal bodies of revolution that minimize wave drag have been analytically determined by Von-Karman and Sears-Haack half a century ago. These two bodies yield the minimum wave drag under two different sets of constraints. The Von-Karman body assumes that the body terminates with a flat plane, that the base area in this plane is known and that the total length of the body is specified. The Sears-Haack body assumes that the body is pointed at both ends and that the total volume and length of the body are given.

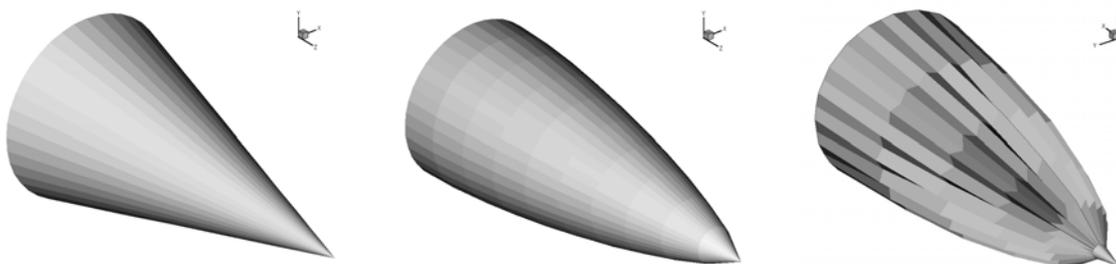


Figure 5: Initial cone configuration (left), optimum ogive shape (center) and converged star-shaped shape (right) of a hypersonic missile with a fixed length and a fixed volume [4].

The design variables were the radii of the body at 10 cross sections. Each design variable (the local cross sectional radius) was allowed to vary from 0 to 10 meters. During the optimization process, the length was kept fixed and the total volume of the body was constrained (with an equality constraint) not to change by more than 1.3 percent from its initial value. The constrained optimization converged to the smooth 'bulged' axisymmetric body called an *ogive* (Figure 5). The base area of the optimized

body, and the total volume (fixed) were then used to compute Von-Karman and Sears-Haack bodies from analytical expressions [16]. The analytically and numerically optimized shapes were in excellent agreement. The optimum ogive body drag was 58 percent of the original cone drag.

Next, all of the body surface nodes on the first cross section could move together radially and were controlled by one design variable. On the other five cross section planes, all of the 38 surface nodes had two degrees of freedom except for the two ‘seam’ points (the points on the vertical plane of symmetry) that were allowed to move only in the vertical plane. Thus, there were 78 design variables per each of the five cross sections and one design variable (radius) at the sixth cross section giving a total of 391 design variable in this test case. The constrained gradient-based algorithm could not converge to anything better than the smooth ogive shape which happened around 30th iteration. Therefore, the optimization was switched to a constrained GA algorithm. The convergence after that point again dramatically increased leading to the development of the body surface ‘ridges’, or ‘channels’, and the narrowing of the spiked nose. The final drag was 28 percent of the drag of a cone.

3.2 Single-Objective Constrained Optimization of a Cascade of Airfoil Shapes

A micro-genetic shape design optimization algorithm was applied to a redesign of an existing two-dimensional cascade of turbine airfoils having supersonic exit flow [15]. The single objective was to minimize the total pressure loss across the cascade row. A constrained micro-genetic optimizer was used for minimization of this single objective function. The following equality constraints were specified and iteratively enforced: aerodynamic lift force, mass flow rate, exit flow angle, and airfoil cross-section area. In addition, axial chord and gap-to-axial chord ratio were kept fixed, while enforcing an inequality constraint where the airfoil thickness was greater than or equal to the specified minimum allowable thickness distribution. The sequential quadratic programming optimizer was used for enforcement of the computationally inexpensive equality constraints like the specified airfoil cross-section area. Analysis of the performance of intermediate cascade shapes was performed with an unstructured grid compressible Navier-Stokes turbulent flow-field analysis code. The airfoil geometry was parameterized using nine conic section parameters and eight B-spline control points, thus keeping the number of design variables to a minimum while achieving a high degree of geometric flexibility and robustness.

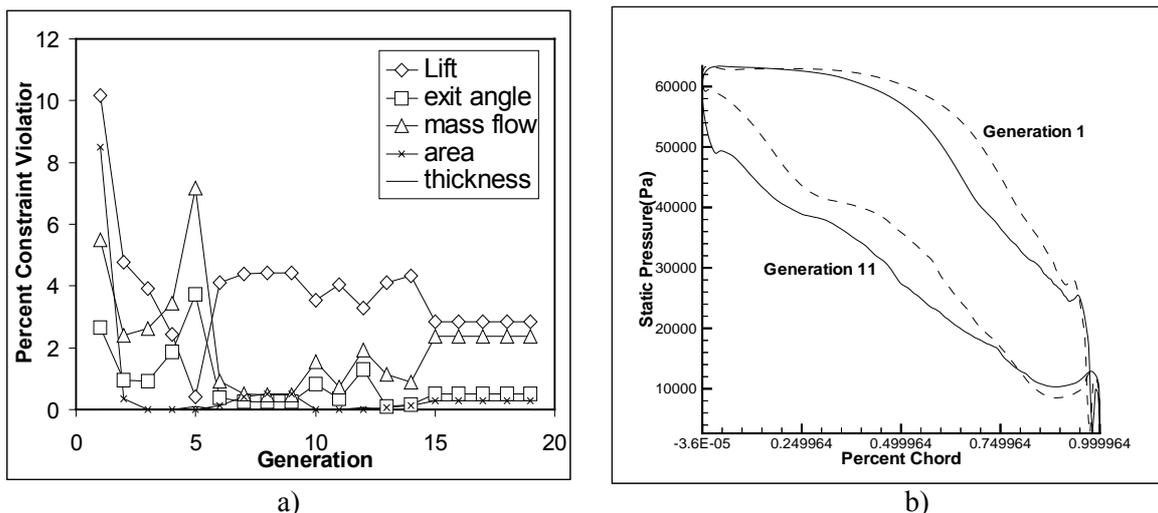


Figure 6. a) Violation of constraints when using penalty function, and b) surface pressure distribution for the best airfoils of generations 1 and 11, with SQP, minimum thickness distribution constraint, and 9 conic sections with 8 B-spline geometry perturbation points [15].

The optimization code proved to be very robust since it found the narrow feasible domain and converged to a minimum that satisfied all the constraints within the tolerances specified (Figure 6a). This type of shape design optimization is feasible on an inexpensive single processor workstation, it requires no changes to the existing flow-field analysis code, and even a semi-skilled designer can operate it. The surface pressure distribution (Figure 6b) that corresponds to the optimized airfoil cascade shape would be practically impossible to know in an *a priori* fashion even by the most experienced of the aerodynamics designers.

3.3 Rotor Cascade Optimization With Unsteady Passing Wakes

An axial turbine rotor cascade shape optimization with unsteady passing wakes was performed to obtain improved aerodynamic performance using an unsteady Navier-Stokes flow-field analysis code [17]. The objective function was defined either as minimization of total pressure loss or as maximization of lift, while the mass flow rate was fixed during the optimization. The design variables were geometric parameters characterizing airfoil leading edge, camber, stagger angle, and inter-row axial spacing (Figure 7a). Penalty terms were introduced for combining the constraints with the objective function. A genetic algorithm with a population of 32 designs was used as the optimizer. During each optimization iteration, the objective functions of the 32 new population members were computed simultaneously by using a 32 processor distributed memory parallel computer. The optimization results indicated that only minor improvements were possible in the unsteady rotor/stator aerodynamics by varying these geometric parameters (Figure 7b).

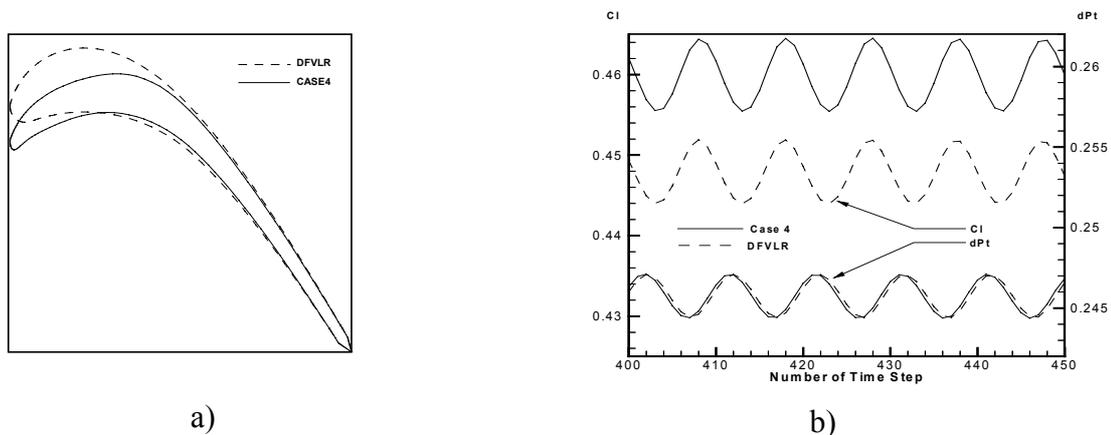


Figure 7: Original DFVLR and optimized DFVLR (Case4) rotor linear cascade airfoil shapes (a) and time variation of lift and total pressure loss for these two cascades (b) [17].

3.4 Multi-Disciplinary Design Optimization Applied to Magneto-Hydrodynamics

Most realistic design problems involve not only aerodynamics, but also other interacting disciplines. One such multi-disciplinary design optimization example involves magneto-hydrodynamics [18]. When a viscous liquid flows from a narrow passage into a suddenly wider passage, there are significant flow separation zones (Figure 8a) that will significantly reduce the efficiency of such flow fields. One possibility to reduce the flow separation would be to perform a straightforward wall shape optimization. But, if the shape of the passage walls is not to be altered for whatever reason, it is still possible to affect the flow-field pattern if the fluid is electrically conducting. It is well known that electrically conducting fluids respond to externally applied magnetic or electric fields. In this

situation, the objective is to find the proper distribution and orientation of the externally applied magnetic field along the passage walls so that the fluid flow separation is minimized.

Using a two-dimensional magneto-hydrodynamics analysis code based on the least squares finite element method and a parallel micro-genetic optimizer, it was recently shown [18] that such optimized magnetic fields can be used to significantly reduce flow-field separation (Figure 8b) and increase the static pressure rise for a fixed length of a diffuser.

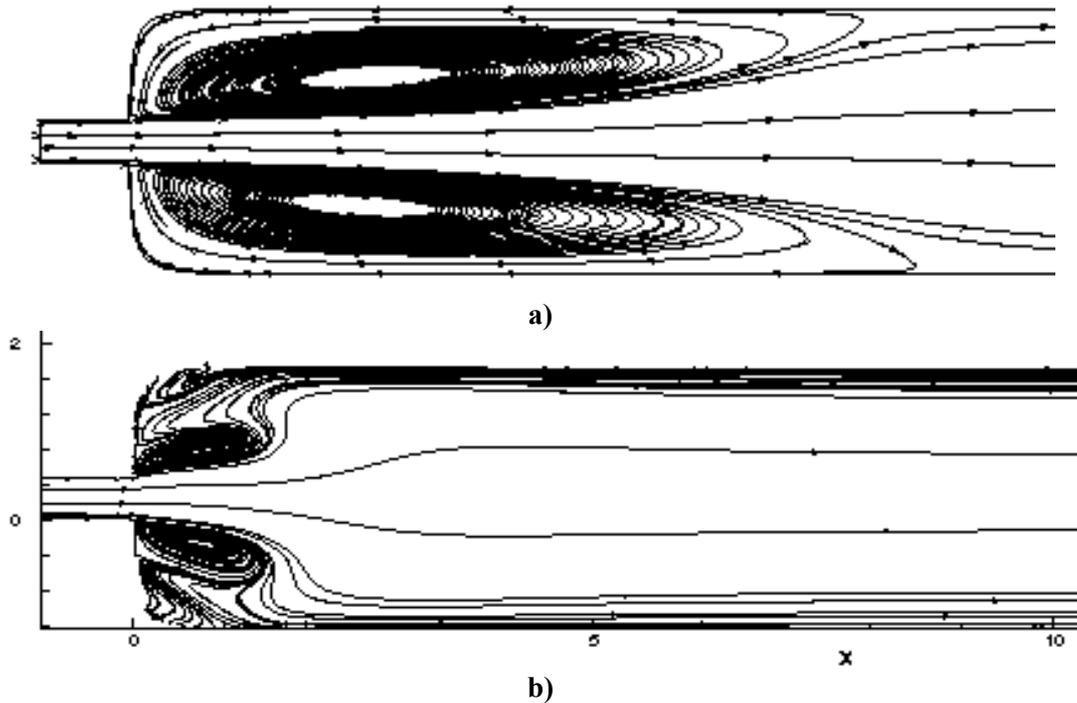


Figure 8: Streamlines for diffuser flow without magnetic field (a) and with an applied magnetic field (b) optimized to suppress laminar steady incompressible flow separation [18].

3.5 Optimization of Freezing Protocols for Preservation of Organs for Tissue Banking

One concept that offers a possible practical solution to freezing and thawing of organs is to immerse them in a cryo-protective gelatin in order to assure that the heat transfer from the outer surface of the organ to the gelatin occurs by pure conduction. The optimization objective is then to find the proper time variation of thermal conditions on the surface of the freezing container so that the optimal local cooling rates are achieved at each instant of time at every point inside the heterogeneous organ that was simulated as been composed of four different types of tissues. Transient temperature distribution was computed at every point of the organ using a three-dimensional linear thermo-elasticity finite element method analysis code subject to initially guessed 26 parameters describing temperature distribution on the spherical freezing container surface. From this, the actual local temperature gradients and thermal stresses were determined at each point in the organ.

A nonlinear constrained maximization method based on a micro-genetic algorithm [19,20] was used after a certain time interval to optimize these 26 parameters at each of the control points on the spherical container surface. Thus, such time evolution of temperature distribution (Figure 9a) on the container surface was determined that it maximizes the local cooling rates in the organ while keeping the local thermal stresses in the organ below user specified maximum allowable values (Figure 9b).

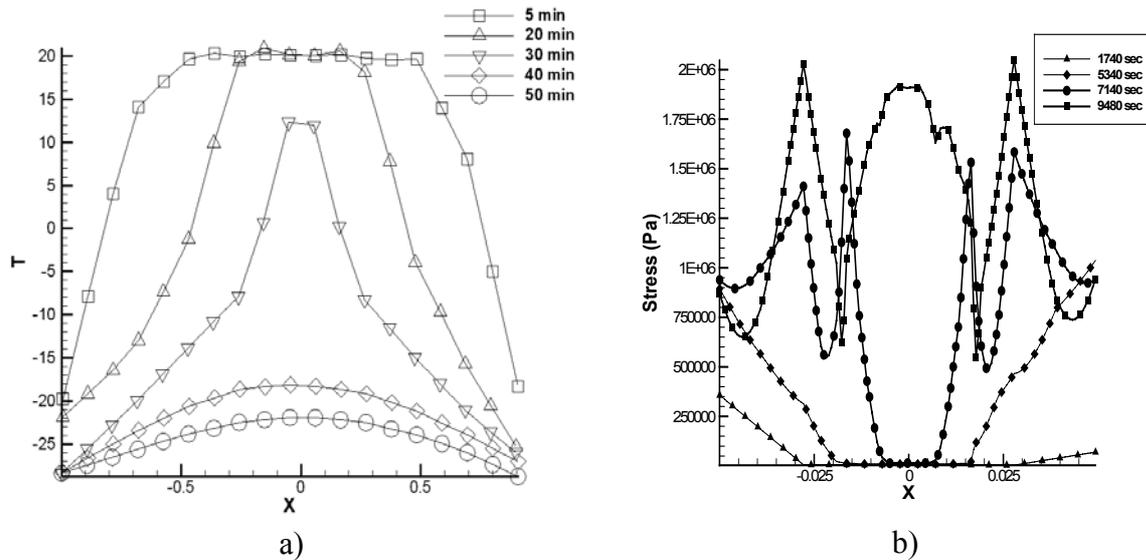


Figure 9: Temperature (a) and von Mises stresses (b) time evolutions along the intersections of x-y plane at $z = 0$ and x-z plane at $y = 0$ using periodic optimization of a spherical container wall temperature distribution during optimized freezing of a dog kidney [19,20].

4. Optimization Based Upon Self-Organization and Evolutionary Simulation (IOSO)

In multi-objective optimization we strive to compute the group of the *not-dominated* solutions, which is known as the Pareto optimal set, or Pareto front. These are the feasible solutions found during the optimization that cannot be improved for any one objective without degrading another objective. The multi-objective constrained optimization algorithm that we used was a modified version of an indirect method of optimization based upon self-organization (IOSO) and evolutionary simulation principles for parallel computation [21]. Each iteration of IOSO consists of two steps. The first step is the creation of a local approximation of the objective functions. In this step, the initial approximation function is constructed from a set of simple approximation functions resulting in a final response function that is a multi-level graph. The second step is the optimization of this approximation function. This approach allows for corrective updates of the approximation to make it more accurate in regions of the design space that promise rapid convergence.

The distinctive feature of this approach is an extremely low number of trial points to build the initial approximation (typically 30-50 points for the optimization problems with nearly 100 design variables). During each iteration of IOSO, the optimization of the response function is performed only within the current search area. This step is followed by a direct call to the mathematical analysis model for the obtained point. During the IOSO operation, the information concerning the behavior of the objective function in the vicinity of the extremum is stored, and the response function is made more accurate only for this search area. Thus, a series of approximation functions for a particular objective of optimization is built at each iteration. These functions differ from each other according to both structure and definition range. The subsequent optimization of these approximation functions allows us to determine a set of vectors of optimized variables, which are used for the computation of optimization objectives on a parallel computer.

4.1 Multi-Objective Aerodynamic Shape Optimization of Turbine Cascade Airfoils

As a practical example, a constrained multi-objective shape optimization was performed on a linear cascade of gas turbine airfoils that had a finite length, thus a finite number of airfoils. The original airfoil shapes were designed by expert aerodynamicists at the von Karman Institute of Fluid Dynamics (VKI) using a highly sophisticated inverse shape design code. Thus, this initial airfoil cascade shape was already highly efficient. This way it was possible to observe if the multi-objective constrained optimization was capable of creating realistic results that were better than the initial finite cascade configuration. The objectives were to simultaneously minimize the total pressure loss, maximize total aerodynamic loading (aerodynamic force component tangent to the airfoil cascade), and minimize the number of airfoils in the finite cascade row. The equality constraints were fixed mass flow rate, axial chord, inlet and exit flow angles, and blade cross-section area. The inequality constraints were the minimum allowable airfoil thickness distribution, minimum allowable lift force, and a minimum allowable trailing edge radius. This means that the entire airfoil cascade shape was optimized including its stagger angle, thickness, curvature, and solidity resulting in 18 design variables, 5 nonlinear constraints, and 3 objectives. The analysis of the performance of intermediate airfoil cascade shapes were performed using an unstructured grid based compressible Navier-Stokes flow-field analysis code with a $k-\epsilon$ turbulence model.

It is interesting to notice that although the VKI airfoil was designed by experienced aerodynamicists using sophisticated inverse shape design software, the optimizer found an entire family of feasible solutions that were better than the inversely designed VKI airfoil cascade for all three objectives. Specifically, cascade No.1 offers reduction of 7 percent in total pressure loss, needs 1 airfoil less than the VKI cascade, and creates about 1 percent higher loading (Figure 10). The designer ultimately must choose the best compromise solution among the optimized solutions that form the Pareto front [22].

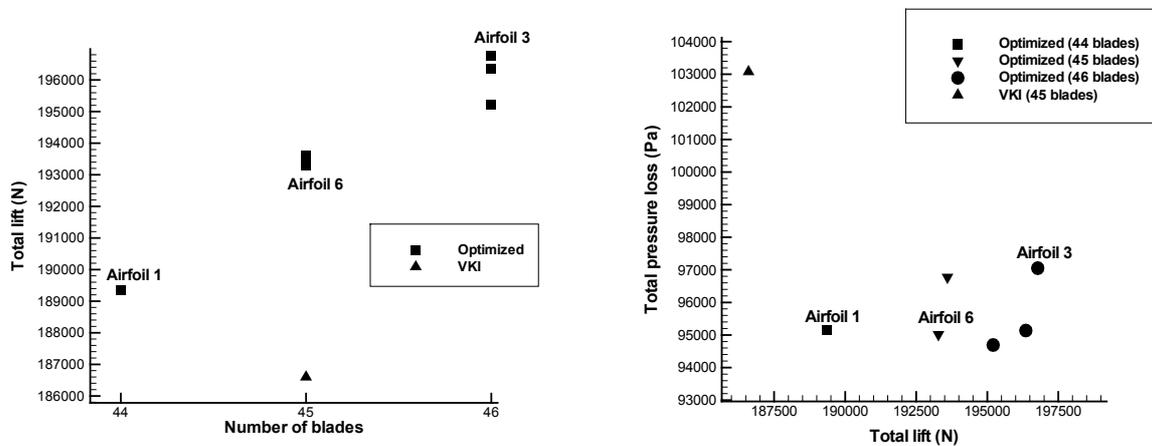


Figure 10: Comparisons of total loading produced, total pressure loss generated, and number of airfoils for optimized finite length cascades and the original VKI airfoil cascade [22].

Summary and Recommendations

A number of standard optimization algorithms can be assembled in a hybrid optimization tool where a set of heuristic rules can be used to perform automatic switching among the individual optimizers in order to avoid local minimums, escape from the local minimums, converge on a minimum, and

reduce the overall computing time. The constraints were enforced either via penalty function or via Rosen's projection method. It was demonstrated that hybrid optimization is a very robust and cost-effective optimization concept. Automatic switching among the individual optimizers can be further improved by incorporating certain aspects of neural networks. Use of simplified models (surrogates) for evaluation of the object function is highly cost-effective, although progressively more complete physical models should be used as the global optimization process starts converging. Otherwise, ludicrous results are possible where the deficiencies of the surrogate models are fully exploited by the optimizer. Parameterization of the design space plays a crucial role in the hybrid constrained optimization. Coarse parameterization usually, but not always, leads to a converged result at an acceptable cost in computing time. A refined parameterization definitely widens the feasible region in the case of a highly constrained optimization. Finally, a gene correction method based on sequential quadratic programming could be effectively used to enforce certain inexpensive constraints while penalty terms could be used to enforce the remaining constraints.

References

- [1] G. S. Dulikravich, *Design and Optimization Tools Development*, Chapters no. 10-15 in H. Sobieczky, ed., *New Design Concepts for High Speed Air Transport*, Springer, Wien/New York, (1997) 159-236.
- [2] G. S. Dulikravich, T. J. Martin, B. H. Dennis, N. F. Foster, *Multidisciplinary Hybrid Constrained GA Optimization*, Chapter 12 in K. Miettinen, M. M. Makela, P. Neittaanmaki and J. Periaux, eds., *EUROGEN'99 - Evolutionary Algorithms in Engineering and Computer Science: Recent Advances and Industrial Applications*, John Wiley & Sons, Ltd., Jyvaskyla, Finland, May 30 - June 3, 1999, (1999) 231-260.
- [3] N. Egorov, *Indirect Optimization Method on the Basis of Self-Organization*, Curtin University of Technology, Perth, Australia, Optimization Techniques and Applications (ICOTA'98), vol.2, (1998) 683-691.
- [4] N. F. Foster, G. S. Dulikravich, Three-dimensional aerodynamic shape optimization using genetic evolution and gradient search algorithms, *AIAA Journal of Spacecraft and Rockets*, 33, 3 (1997) 33-33.
- [5] R. T. Hafka, D. S. Malkus, *Calculation of sensitivity derivatives in thermal problems by finite differences*, *International Journal of Numerical Methods in Engineering*, 17, (1981) 1811-21.
- [6] T. J. Martin, G. S. Dulikravich, *Implicit BEM sensitivity for aero-thermo-structural optimization of coolant passages in turbine blades*, 9th AIAA/ISSMO Multidisciplinary Analysis & Optimization Symposium, Atlanta, GA, Sept. 4-6, 2002.
- [7] G. S. Dulikravich, T. J. Martin, Design of proper super elliptic coolant passages in coated turbine blades with specified temperatures and heat fluxes, *AIAA Journal of Thermophysics and Heat Transfer*, 8, 2 (1994) 288-294.
- [8] G. S. Dulikravich, T. J. Martin, *Inverse Shape and Boundary Condition Problems and Optimization in Heat Conduction*, Chapter no. 10 in W. J. Minkowycz and E. M. Sparrow, eds., *Advances in Numerical Heat Transfer - Volume I*, Taylor and Francis, (1996) 381-426.
- [9] T. J. Martin, G. S. Dulikravich, *Aero-Thermo-Elastic Concurrent Design Optimization of Internally Cooled Turbine Blades*, Chapter 5 in A. J. Kassab and M. H. Aliabadi, eds., *Coupled Field Problems, Series on Advances in Boundary Elements*, WIT Press, Boston, MA, (2001) 137-184.

- [10] T. J. Martin, G. S. Dulikravich, *Analysis and multi-disciplinary optimization of internal coolant networks in turbine blades*, ASME IMECE2001/HTD-24316, New York, November 11-16, 2001; also to appear in AIAA J. of Propulsion and Power.
- [11] M. V. Petrovic, W. Riess, *Through-flow calculation in axial flow turbines at part load and low load*, First European Conference on Turbomachinery - Fluid Dynamic and Thermodynamic Aspects, Erlangen, Germany, March 1-3, 1995.
- [12] M. V. Petrovic, G. S. Dulikravich, T. J. Martin, *Maximizing multistage turbine efficiency by optimizing hub and shroud shapes and inlet and exit conditions of each blade row*, International Journal of Turbo & Jet-Engines, 17 (2000) 267-278.
- [13] K. Krishnakumar, *Micro-genetic algorithms for stationary and non-stationary function optimization*, SPIE: Intelligent Control and Adaptive Systems, Vol. 1196, Philadelphia, PA, 1989.
- [14] D. L. Carroll, *Chemical laser modeling with genetic algorithms*, AIAA Journal, 34, 2 (1996) 338-346.
- [15] B. H. Dennis, G. S. Dulikravich, Z.-X. Han, *Constrained optimization of turbomachinery airfoil cascade shapes using a Navier-Stokes solver and a genetic/SQP algorithm*, ASME paper 99-GT-441, ASME Turbo Expo99, Indianapolis, Indiana, June 7-10, 1999; also in AIAA Journal of Propulsion and Power, 17, 5 (2001).
- [16] J. D. Anderson, Jr. *Hypersonic and High Temperature Gas Dynamics*, McGraw-Hill, New York (1989).
- [17] E. S. Lee, G. S. Dulikravich, B. H. Dennis, *Rotor cascade shape optimization with unsteady passing wakes using implicit dual time stepping and genetic algorithm*, Proceedings of the 9th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery (ISROMAC-9), Honolulu, HI, February 10-14, 2002.
- [18] B. H. Dennis, G. S. Dulikravich, *Optimization of magneto-hydrodynamic control of diffuser flows using micro-genetic algorithm and least squares finite elements*, Journal of Finite Elements in Analysis and Design, 37, 5 (2001) 349-363.
- [19] B. H. Dennis, G. S. Dulikravich, Y. Rabin, *Optimization of organ freezing protocols with specified allowable thermal stress levels*, ASME IMECE-2000, Orlando, FL, November 5-10, 2000, HTD-Vol. 368/BED-Vol. 47, (2000) 33-48.
- [20] B. H. Dennis, G. S. Dulikravich, *Determination of unsteady container temperatures during freezing of three-dimensional organs with constrained thermal stresses*, in M. Tanaka and G. S. Dulikravich, eds., *International Symposium on Inverse Problems in Engineering Mechanics – ISIP'2k*, Nagano, Japan, March 7-10, 2000, Elsevier Science Ltd, Amsterdam, (2000) 139-148.
- [21] I. N. Egorov, *Indirect optimization method on the basis of self-organization*, Curtin University of Technology, Perth, Australia, *Optimization Techniques and Applications (ICOTA'98)*, 2, (1998) 683-691.
- [22] B. H. Dennis, Z.-X. Han, I. N. Egorov, G. S. Dulikravich, C. Poloni, *Multi-objective optimization of turbomachinery cascades for minimum loss, maximum loading, and maximum gap-to-chord ratio*, AIAA Multidisciplinary Analysis and Optimization Conference and Exhibit, Long Beach, CA, Sept. 5-8, 2000; also in International Journal of Turbo & Jet-Engines, 18, 3 (2001) 201-210.