JOURNAL OF AIRCRAFT
Vol. 37, No. 6, November–December 2000

Aerodynamic and Aeroacoustic Optimization of Rotorcraft Airfoils via a Parallel Genetic Algorithm

Brian R. Jones,* William A. Crossley,† and Anastasios S. Lyrintzis‡
Purdue University, West Lafayette, Indiana 47907-1282

A parallel genetic algorithm (GA) methodology was developed to generate a family of two-dimensional airfoil designs that address rotorcraft aerodynamic and aeroacoustic concerns. The GA operated on 20 design variables, which constituted the control points for a spline representing the airfoil surface. The GA took advantage of available computer resources by operating in either serial mode, where the GA and function evaluations were run on the same processor or "manager/worker" parallel mode, where the GA runs on the manager processor and function evaluations are conducted independently on separate worker processors. The multiple objectives of this work were to minimize the drag and overall noise of the airfoil. Constraints were placed on lift coefficient, moment coefficient, and boundary-layer convergence. The aerodynamic analysis code XFOIL provided pressure and shear distributions in addition to lift and drag predictions. The aeroacoustic analysis code, WOPWOP, provided thickness and loading noise predictions. The airfoils comprising the resulting Pareto-optimal set exhibited favorable performance when compared with typical rotorcraft airfoils under identical design conditions using the same analysis routines. The relationship between the quality of results and the analyses used in the optimization is also discussed. The new airfoil shapes could provide starting points for further investigation.

Nomenclature

c = airfoil chord length
c_d = sectional drag coefficient
c_f = skin-friction coefficient
c_ℓ = sectional lift coefficient
c_m = section moment coefficient about $\frac{1}{4}$ chord
c_p = pressure coefficient
f = fitness function
g_j = constraint function
M = chord Mach number
n = number of CPUs used during a parallel run
n_{con} = number of constraints
n_{fc} = number of design flow conditions
n_{obs} = number of observer locations
P^* = scaled penalty factor
P_j = penalty function
Re = Reynolds number based on chord
r_j = penalty drawdown coefficients
u_e = boundary-layer-edge velocity
x/c = normalized airfoil station
x_s/c = normalized total surface separation, projected along ordinate axis
y/c = normalized airfoil ordinate
α = geometric angle of attack
Ψ = azimuth angle
$\forall_i$ = for all instances in the set i
$\exists_i$ = there exists in the set i

Subscripts

lower = lower surface of airfoil
upper = upper surface of airfoil
0012 = reference value to NACA 0012 airfoil section

Presented as Paper 98-4811 at the AIAA/NASA/ISSMO 7th Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, 2–4 September 1998; received 23 March 1999; revision received 6 April 2000; accepted for publication 7 April 2000.

*Graduate Student, School of Aeronautics and Astronautics; currently Analyst, Aerodynamics and CFD, Lockheed Martin Aeronautics Company, Fort Worth, TX 76101. Student Member AIAA.

†Assistant Professor, School of Aeronautics and Astronautics, 1282 Grissom Hall. Senior Member AIAA.

‡Associate Professor, School of Aeronautics and Astronautics, 1282 Grissom Hall. Associate Fellow AIAA.

Introduction

THIS paper discusses the application of a parallel genetic algorithm (GA) to a multiobjective airfoil design problem. Airfoil shapes were designed to minimize the drag coefficient and the overall averaged sound-pressure level (OASPL) for three flow conditions representative of those experienced by a helicopter rotor airfoil, when the helicopter is in forward flight. An angle of attack, a Mach number, and a Reynolds number describe these flow conditions.

Direct airfoil design perturbs an initial airfoil shape to improve performance of the airfoil. Although successful, this approach generally produces airfoils deviating only slightly from the initial design. Calculus-based search methods especially encounter this limitation because they find the nearest local optimum to the original design. Airfoil features like trailing-edge tabs, droop-snoots, and complex camber would be difficult to discover using a traditional method that perturbs a known shape.

In contrast, inverse airfoil design produces an airfoil whose pressure distribution matches a desired distribution. The inverse approach risks defining a distribution that no physical shape can produce. The inverse approach suffers further disadvantages when applied to multiobjective design. To obtain a family of airfoil shapes, the designer using inverse methods must specify multiple pressure profiles that represent the many possible compromises between objectives.

This research uses a direct problem formulation by defining an airfoil's surface using the location of spline control points. These control points constitute the design variables. Employing the GA to solve the direct problem allows the discovery of shapes unlikely to be found using other methods.

Genetic Algorithms

Since its first description, the genetic algorithm has been applied to many engineering optimization problems.[1] Based on Darwin's survival-of-the-fittest concept, the GA performs optimization tasks by "evolving" a population of highly fit designs over many generations. A GA has the ability to search highly multimodal, discontinuous design spaces. The GA also locates designs at, or near, the global optimum without requiring an initial design point.

The GA represents design variables as strings of binary numbers, which serve as chromosomes. Initially, the GA randomly generates a population of individuals. After decoding the chromosome of each individual into the corresponding design variables, each design is analyzed to determine a fitness value. Individuals with better fitness values are considered more optimal so that this fitness value must reflect both the objective of the design problem and any constraints imposed upon the design.

The GA employs selection, crossover, and mutation operators to perform its search. The selection routine performs the survival-of-the-fittest function that allows better individuals to survive and serve as parents. Crossover combines portions of chromosomes from the surviving parent designs to form the next generation of designs; combining features of good designs on average, but not always, results in better designs. This gives the GA its optimization-like capability. The mutation operator is used quite infrequently, as in nature, but this operator can mutate a binary bit in a chromosome to its opposite value (e.g., 0 to 1), which can introduce beneficial design traits that did not exist in the current population. If the mutated trait is poor, the design with this mutation will be unlikely to survive. This process transforms an initial population of randomly selected designs into a population of individuals that have adapted to their environment by becoming more optimal. Additional details of the genetic algorithm can be found in several texts, such as Ref. 1.

Related Research

Applications of the GA to single-objective aerospace design problems include airfoil aerodynamics for fixed-wing applications,[2–4] rotor applications,[5] and rotor system design for aeroacoustics.[6] GA applications to the multiobjective and multidisciplinary design of airfoils have also been investigated.[7] Airfoil design has been a popular application for GAs, although most of these efforts attempted to solve an inverse problem,[8–11] frequently for transonic airfoils. Of those that attempt the direct problem,[4,12] many perturb an existing shape, thus limiting the chances of locating nontraditional airfoil shapes. Further GA applications have started to explore multiobjective and multidisciplinary problems, including airfoil aerodynamic/structural design,[5,7] turbine cascade design,[13] and wing design.[14,15] Rotor blades have also been optimized to reduce vibratory loads using a parallel GA.[16] Additional discussion about GA applications for rotorcraft can be found in Ref. 17.

There are many variations of the GA and its operators. The GA used in this work utilized uniform crossover, tournament selection, and elitism.[18,19] Parallel execution provides efficiency benefits to the GA, and many implementations have been studied. These included distributed coarse-grain parallelization schemes,[20,21] demes,[2] and island methods. The coarse-grain scheme was adapted for this application. Finally, the use of the GA to solve multiobjective design problems has been investigated; the n-branch tournament, a generalization of the two-branch tournament approach,[22] has been used here.

Problem Formulation

For a set of three flow conditions, a family of low-drag, low-noise helicopter rotor airfoils representing the Pareto-optimal set[23] was generated. The airfoil design problem addresses a two-dimensional shape, but the flow conditions used in this work correspond to different azimuth angles of a helicopter rotor in forward flight. Aeroacoustic predictions are difficult using a two-dimensional airfoil, and so a specialized three-dimensional rotor model, described next, was developed to calculate a noise measure for each airfoil. This noise measure included information from two observer locations and made use of the aerodynamic conditions associated with each of the three flow conditions.

Aerodynamic Analysis Methods

Because the quality of any optimization method's results depends on the analyses used, well-established codes were selected. The airfoil analysis code, XFOIL,[24] formed the core of the analysis. For physically valid airfoils (e.g., no crossing of the upper and lower surfaces) XFOIL predicts the aerodynamic coefficients c_ℓ, c_d, and c_m. XFOIL combines the Karmen–Tsien compressibility correction with a solution generated from closely coupled inviscid and viscous methods. The linear vorticity, stream function panel method supplies the inviscid solution, and an integral boundary-layer method provides the viscous solution. Additionally, XFOIL predicts low-Reynolds-number transitional separation bubbles and provides compressible analysis up to sonic conditions. Despite problems analyzing large regions of separated flow, the viscous capabilities of XFOIL allow it to predict stall-like conditions for reasonable aerodynamic shapes with rounded leading edges and gradual changes in surface curvature.

Aeroacoustic Analysis Methods

The airfoil's loading and thickness noise are evaluated using the aeroacoustic code, WOPWOP.[25] This program evaluates rotor thickness and loading noise at arbitrary observer locations. The noise calculations depend upon the airfoil thickness, pressure, and shear distributions as functions of distance from leading edge, which are provided by XFOIL for this work.

Because this work addresses the design of two-dimensional airfoil sections and WOPWOP natively handles a three-dimensional rotor system, the rotor model was simplified. The primary simplification reduces the rotor system to a single rectangular planform, untwisted blade extending in the radial direction from the 75% radius station to the tip. Tip effects are a three-dimensional phenomenon and are ignored here. The airfoil loading computed by XFOIL for a flow condition is assumed constant for all rotor radial stations and azimuthal positions in the WOPWOP rotor model. WOPWOP can then predict the loading noise, thickness noise, and OASPL for the simplified rotor model. The OASPL is expressed in decibels and can be computed by adding the thickness and loading noise signals at observer locations. Repeating this for the other two flow conditions and summing the OASPL values calculated at each observer location for all of the flow conditions provides a noise measure associated with the two-dimensional airfoil shape. This approach assumes that if the model associated with one airfoil generates a lower noise measure than the model associated with another airfoil then the first airfoil is a lower-noise design.

Airfoil Representation and Handling

A third-order B-spline with 23 control points described airfoil surfaces for this application.[26] To define an airfoil, the chordwise coordinates of the B-spline control points remain fixed in a cosine distribution along a unit chord. One point defined the leading edge and was located on the horizontal axis. Two more points defined a manufacturable trailing edge with a gap of 0.005 x/c. The ordinate stations of the remaining 20 points, 10 points each on the upper and lower surfaces, were the design variables for this problem.

Compared to other methods, a B-spline produces a smoother representation of irregular airfoils that are encountered by the GA, particularly in the initial randomly generated population. Figure 1 illustrates such an airfoil. Obviously, this shape fails to represent a practical airfoil. The smoothing effect of the B-spline allows some analysis to be conducted, even on irregular shapes. The surface

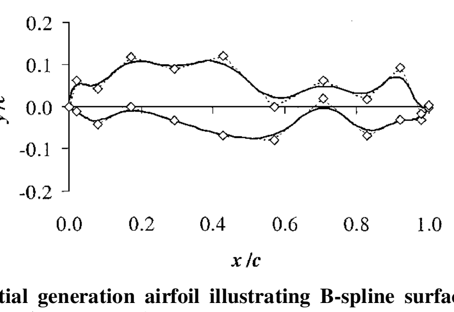


Fig. 1 Initial generation airfoil illustrating B-spline surface: ——, B-spline; and ◇, control points.

described by the B-spline is divided into 200 panels for aerodynamic and aeroacoustic predictions. This relatively high number of panels allowed traditional cosine spacing (with more panels near the leading and trailing edges) for the aerodynamic and aeroacoustic analyses and provided acceptable resolution for less regular shapes.

To keep the problem tractable, limits must be imposed on the position of each spline control point. Excessively limiting these parameters removes potentially beneficial designs from the search space, whereas excessive freedom wastes computing effort on very irregular designs. Figure 2 displays the limits used in this problem description. The figure also shows the small, finite gap at the trailing edge. The bold line represents the upper and lower limits of the 10 upper surface control points, and the thin line represents the limits for the lower surface. This search space still allows features like trailing-edge reflex and large amounts of camber, while eliminating pointed leading edges and other unsolvable features.

Problem Formulation

Building upon the ideas in Ref. 5, the two-dimensional airfoil flow conditions used here represent those experienced at the 75% radius position on a rotor blade of the example helicopter from Prouty[27] in forward flight with and advance ratio of 0.3. Figure 3 illustrates these flow conditions, and Table 1 presents the corresponding constraint limits. These constraints ensure that airfoils generated by the GA maintain a lift coefficient equal to, or greater than, that of the example rotor airfoil and maintain a moment coefficient smaller in magnitude than that of the example.

Addressing an aerodynamic objective and an aeroacoustic objective required two fitness functions. The aerodynamic objective sought to minimize the airfoil's drag coefficient at all three flow conditions. Because the GA performs its search using only a fitness value to represent a design, this value must reflect the objective

Table 1 Aerodynamic constraint limits

Coefficient	Flow condition 1 (Ψ = 90 deg)	Flow condition 2 (Ψ = 180 deg)	Flow condition 3 (Ψ = 270 deg)
c_ℓ	≥ 0.16	≥ 0.60	≥ 1.00
$\lvert c_m \rvert$	< 0.03	< 0.03	< 0.03

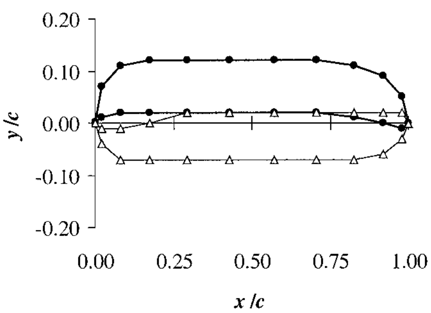


Fig. 2 Control point design variable limits: •, upper surface; and △, lower surface.

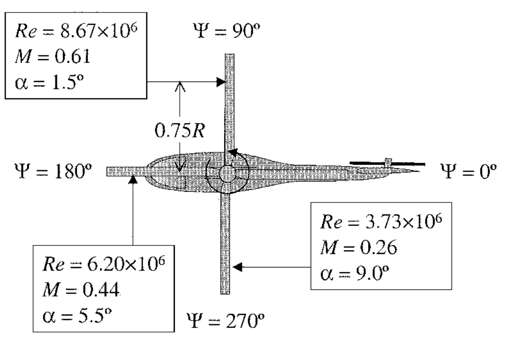


Fig. 3 Airfoil design condition description.

function value and any constraint violations. The aerodynamic fitness function was

$$f_1 = \sum_{i=1}^{n_{\mathrm{fc}}} \left[\left(\frac{c_d}{c_{d0012}} \right) + \sum_{j=1}^{n_{\mathrm{con}}} r_j P_j \right] \tag{1}$$

In this form drag was minimized, subject to constraints imposed via an exterior penalty method, as a sum including all three ($i = 1, 2, 3$) flow conditions. Because c_d is larger at high angles of attack, the drag coefficient at each flow condition was scaled by the NACA 0012's drag coefficient at the same flow condition. This scaling provided nearly equal consideration of all three flow conditions.

XFOIL determines boundary-layer properties using an iterative solution involving both an inviscid and viscous analysis. If the boundary-layer solution failed to converge within a specified number of iterations for a given flow condition, then any resulting aerodynamic data were deemed unreliable for this application. Because of this, two sets of constraint functions were used based on the convergence tolerance of XFOIL's viscous/inviscid boundary-layer iteration scheme. Constraints were enforced via linear penalty functions of the form:

$$P_j = \max(0, g_i) \tag{2}$$

where g_j is positive valued when violated.

The first set of constraints was used when the boundary-layer solution converged. In this case constraints were enforced for the lift and moment coefficients using two constraint functions, and the third function was set to zero:

$$g_{1i} = 1 - (c_\ell / c_{\ell 0012})_i \tag{3}$$

$$g_{2i} = (|c_m| / |c_{m0012}|)_i - 1 \tag{4}$$

$$g_{3i} = 0 \tag{5}$$

When the boundary-layer solution failed, the ratio (c_d / c_{d0012}) was assigned a value of 100 in order to be larger than the nominal value for airfoils with converged boundary-layer solutions by at least an order of magnitude. The fitness value will then have a large contribution from this (c_d / c_{d0012}) value, and so the airfoil design will not be selected in favor of an individual with a converged boundary-layer solution. In this situation the constraints took the form

$$g_{1i} = 0 \tag{6}$$

$$g_{2i} = 0 \tag{7}$$

$$g_{3i} = 1 - \lfloor (x/c)_{\max u_e \,\mathrm{upper}} + (x/c)_{\max u_e \,\mathrm{lower}} \rfloor / 2 \tag{8}$$

The third constraint function g_3 reflects how poor the solution is; this allows an airfoil with a nearly converged boundary layer to have a better fitness value than an airfoil whose iterative boundary-layer solution failed near the leading edge of the airfoil. This constraint uses the chordwise position of the last boundary-layer velocity prediction during initialization of the iteration scheme to measure how far the solution had proceeded for the upper and lower surfaces. The constraint is satisfied when the entire upper and lower surfaces have nonzero velocity predictions during initialization. Although this does not directly indicate the quality of the boundary-layer solution, this measurement provides the GA with a consistent means of comparing two individuals whose boundary-layer solution has failed.

Drawdown coefficients r_j in Eq. (1) scaled the penalty to the magnitude of the objective. The penalty associated with Eq. (8) received a larger value of r to encourage designs that would allow solution of the boundary-layer calculations.

The aeroacoustic fitness function incorporated the multiple flow conditions as well as multiple observer locations. The objective sought to minimize the airfoil's OASPL values, as summed over multiple flow conditions and as seen by multiple observers. Note that OASPL includes thickness and loading noise, but does not include

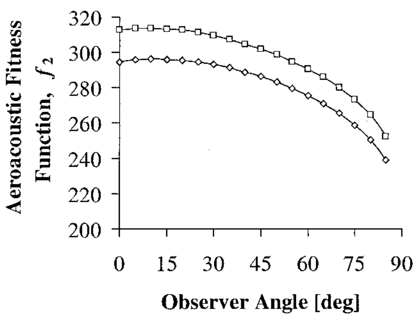


Fig. 4 Variation of f_2 with changing observer angle: □, airfoil number 1; and ◇, airfoil number 2.

the effects of high-speed impulsive (HSI) noise (i.e., shock-wave noise) and blade-vortex interaction (BVI):

$$\text{minimize } f_2 = P^* \sum_{k=1}^{n_{\text{obs}}} \left\{ \sum_{i=1}^{n_{\text{fc}}} [\text{OASPL} + r_a \max(0, g_{\text{sep}})]_i \right\}_k \tag{9}$$

Two observer locations were placed 50 m from the hub of the simplified WOPWOP rotor model, one in the rotor plane and one below the rotor along a ray angled at 45 deg to the rotor plane. Because OASPL combines loading and thickness noise, using multiple observer locations prevented a single source from dominating the fitness function. For example, observations in the rotor plane are dominated by thickness noise, whereas loading noise dominates observations below the rotor. The directivity of OASPL varies smoothly with the changing angle of an observer located below the rotor plane. Figure 4 illustrates the smooth variation of the aeroacoustic fitness function for two of the airfoils eventually generated by the GA. The simplified WOPWOP rotor model for the two-dimensional airfoil has no azimuthal directivity of the noise because of the constant loading assumptions just described. Thus, two observer positions were deemed adequate for this work, whereas two observers might not be enough for more complicated problems (e.g., including BVI or attempting a three-dimensional rotor-blade design).

A scaled penalty factor[22] P^* enforces aerodynamic constraints for the aeroacoustic fitness function so that penalties imposed on the aeroacoustic fitness have the same scale as those imposed on the aerodynamic fitness:

$$P^* = f_1 \Bigg/ \sum_{i=1}^{n_{\text{fc}}} \left(\frac{c_d}{c_{d0012}} \right)_i \tag{10}$$

The aeroacoustic fitness function in Eq. (9) contains one additional constraint not addressed in the aerodynamic fitness function. A linear penalty function enforces a boundary-layer separation constraint g_{sep}, when more than 30% of the total airfoil surface (both upper and lower) projected along the ordinate axis experiences separated flow. This reduces the artificial benefits to thickness noise predictions obtained when XFOIL sets the skin-friction coefficient to zero in areas of separated flow. The penalty is scaled to the order of the objective function using the drawdown coefficient r_a:

$$g_{\text{sep}_i} = (x_s/c)/0.3 - 1 \tag{11}$$

Implementation

To solve this multiobjective airfoil problem, a parallel GA was developed that incorporated some necessary additional features. Design variable encoding for the binary chromosomes followed a Gray-coding scheme to avoid Hamming distance issues between adjacent variable values.[1] Empirically derived relationships for the population size and mutation rate were used.[28] Features were developed to deal with the close coupling of the two objectives and with the difficulty in obtaining an initially viable population. These include an adaptive single objective to multiobjective fitness evaluation and the "reparenting" of unsolvable individuals. Reference 29 includes additional details of these special features.

Genetic Operators

The GA employed an n-branch tournament selection, where n corresponds to the number of objectives considered. This method is an extension of the two-branch tournament selection.[22] After placing the entire current population in a "pot," a user-defined number of individuals is randomly selected without replacement from this pot. These individuals compete, with the most fit individual surviving as a parent. Every member of the population competes on one of the n-fitness functions. Completing the selection process for a given branch leaves the pot empty. After the pot is refilled using the current population, selection continues using the next branch. Figure 5 illustrates the process when minimizing two fitness functions.

After selection two members of the parent pool are randomly selected, without replacement, and "mated" in order to pass on their traits to two children. This is implemented using uniform crossover.[30] Mutation occurs with a very low probability. This operation changes a bit in the chromosome to its opposite value (i.e., 1 to 0, or 0 to 1), which helps the evolution proceed toward a global solution by introducing binary patterns that may not exist in the present population. This process repeats until an new generation of individuals is created.

The implemented GA also included an elitism operator. This operator replaces an individual from each new population with the best individual of the previous generation. When performing a multiobjective optimization, multiple replacements occur as the best individual in each fitness function survives as an "elite." The addition of elitism to the GA was observed to benefit the solution of multiobjective problems.[22]

Pareto Set Determination

To be considered Pareto optimal, individuals must be both feasible and nondominated.[23] A design is considered nondominated if no

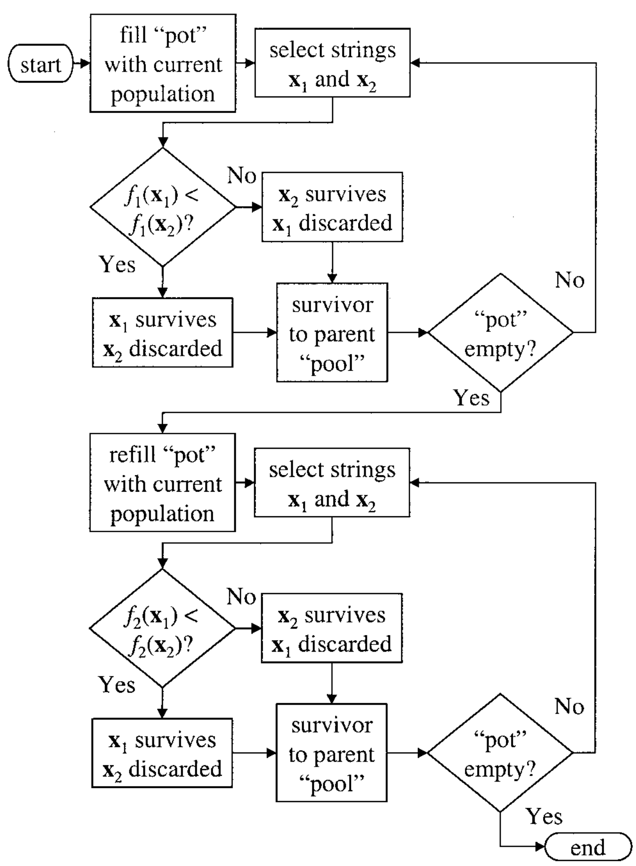


Fig. 5 Flowchart of n-branch tournament selection using two fitness functions.

other design exists that is better in all objectives. Mathematically, a design with a vector of objective values $\boldsymbol{u}$ is dominated by a design with a vector of objective values $\boldsymbol{v}$ if the following condition is met:

$$\forall_i v_i \leq u_i, \qquad \exists_i v_i < u_i, \qquad i = 1, 2, \ldots, n \tag{12}$$

The GA evaluates thousands of designs during a run, and any feasible nondominated design evaluated during the run is of interest. Consequently, the GA stores the set of feasible nondominated individuals in a linked list. This data structure best handles the fluctuating length of this set, as members are inserted and removed according to the conditions of Eq. (12). Evolution halts when either the maximum number of generations is reached or a stopping criterion is satisfied. Here, the stopping criterion is satisfied when the approximate Pareto-optimal set ceases to change.

Feasibility-Handling Features

Solving the direct airfoil design problem proved challenging because of the limits of the analysis tools, the design space size, and the degree of coupling between objectives. Consequently, a system of feasibility classes was developed. The range of possible airfoil shapes effectively required the GA to solve two separate problems. First, the GA needed to evolve a population of airfoils for which XFOIL could generate reasonable solutions. Only then could the GA solve the multiobjective problem because WOPWOP calculations need information from XFOIL. An airfoil whose upper and lower surfaces cross presents a physically impossible shape, whereas the shape in Fig. 1 is typical of an airfoil prompting numerical instabilities within XFOIL. Shapes of both kinds prevent successful aerodynamic analysis.

Six classes were defined to allow constraint handling: 1) unknown—initialization class, no analysis performed; 2) impossible—physically impossible shape; 3) unsolvable—solution aborted as a result of numerical instability in XFOIL; 4) unviable—aerodynamic solution attempted, but boundary-layer solution did not converge; 5) infeasible—converged solution returned with violated constraints; and 6) feasible—converged solution returned without violated constraints.

If the analysis routines deem a design impossible or unsolvable in the initial generation, the design is replaced with another randomly generated individual. During successive generations, the GA reapplies the crossover and mutation operators to the parent designs to form a different design. Because each parent design was at least solvable, their design chromosomes contain some valuable genetic material. This reparenting prevents the loss of this information. Random replacement after the initial generation would reduce the selection pressure that drives the evolutionary process.

Complementing reparenting, an adaptive fitness function evaluation scheme was developed. The GA first sought to minimize only the aerodynamic fitness until at least 60% of the population maintain convergent boundary layers, and then the GA began multiobjective optimization at the next generation. This proved advantageous here because the aeroacoustic analysis requires pressure and shear data from the aerodynamic analysis. By competing on a single objective first, the entire population can be used to reach a viable design space. After obtaining a usable population, both fitness functions can be evaluated, and multiobjective optimization can ensue.

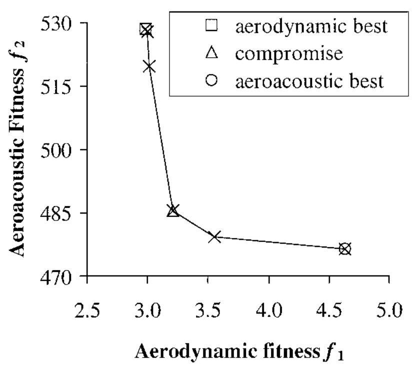
Parallelization

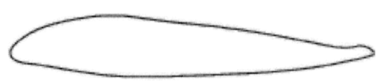
Increased computational cost accompanies using a GA to solve the direct airfoil design problem. Adequate resolution of the spline control points that determine the airfoil surface requires a long chromosome and a correspondingly large population. When evolved over enough generations to satisfactorily generate the Pareto-optimal set, solving the problem requires tens of thousands of fitness evaluations. Because the fitness evaluation of each individual in the population is independent of other individuals, the GA is well suited to coarse grain parallelization. The Message Passing Interface (MPI) was selected to parallelize the code.

A manager/worker model was employed. In this approach the manager node performs all GA operations and distributes individual airfoil designs to the worker nodes for analysis. The sole purpose of a worker node is to calculate the fitness value of its current individual. The manager and workers communicate via blocking communication, which dictates that neither the sending nor receiving processor can begin another task until the communication is complete. Because the GA begins with a random population, large differences in evaluation time can exist among different designs. Dynamic load balancing allows idle workers to begin new tasks without waiting for slower workers to finish.

Results

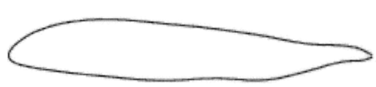
Because of limited access to computational resources, two separate computer systems were used to generate results. Employing the adaptive single-objective optimization, generating a population in which 60% of the individuals maintained converged boundary-layer solutions was completed on the PAPERS machine developed by the Purdue University Department of Electrical Engineering.[31] Seventeen Pentium II (300 MHz) PCs processed 450 generations over approximately 3 days before generating a population of designs meeting the viability condition. This population was then transferred to the 128-node San Diego Supercomputer Center IBM SP2 to generate multiobjective results. An additional 5 h and 20 min of computation across 57 available nodes created an approximate Pareto-optimal set consisting of six feasible nondominated individuals. Figure 6 displays the Pareto front of this set with three representative airfoils. The function space plot compares fitness values, which





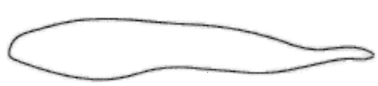
Flow condition	c_ℓ	c_m	c_d	f_2
1	0.3995	0.0123	0.0074	178.8989
2	0.8011	0.0209	0.0056	180.3315
3	1.0845	0.0265	0.0091	169.3829

Aerodynamic best airfoil



Flow condition	c_ℓ	c_m	c_d	f_2
1	0.3483	0.0183	0.0066	163.1024
2	0.8076	0.0144	0.0077	172.2863
3	1.1126	0.0157	0.0099	157.5834

Compromise airfoil



Flow condition	c_ℓ	c_m	c_d	f_2
1	0.4354	-0.0045	0.0093	158.1001
2	0.8557	0.0076	0.0112	160.2536
3	1.1330	0.0147	0.0145	158.2211

Aeroacoustic best airfoil

Fig. 6 Pareto front and representative airfoils.

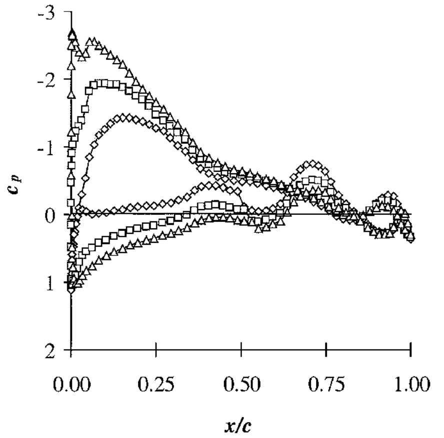


Fig. 7 Compromise airfoil pressure profiles: ◇, flow condition 1; □, flow condition 2; and △, flow condition 3.

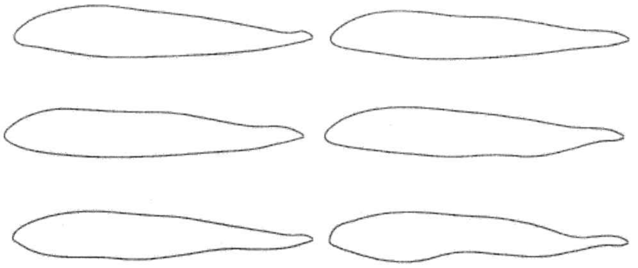


Fig. 8 Airfoil shapes in estimated Pareto-optimal set.

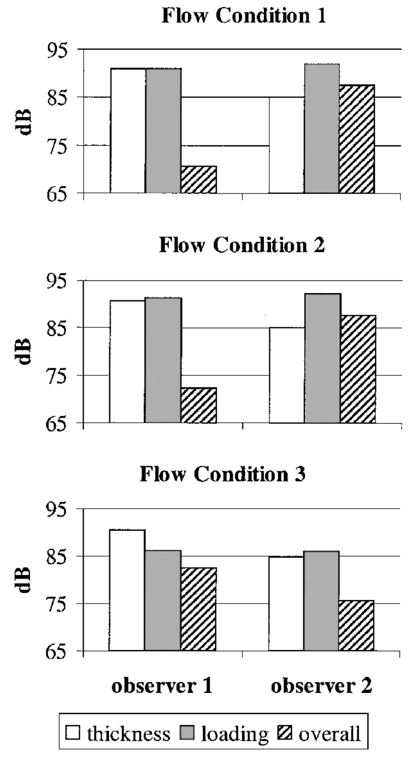


Fig. 9 WOPWOP noise values for compromise airfoil.

account for any penalties. The three highlighted airfoils exhibit the best predicted aerodynamic performance (based on the fitness function f_1), the best predicted aeroacoustic performance (based on f_2), and a compromise between aerodynamic and aeroacoustic performance.

Because the GA requires no initial starting point and does not perturb known shapes, the discovery of nontraditional airfoils is possible. This research intended to discover airfoil shapes that would not have been found via traditional optimization approaches, and the resulting airfoils are atypical. As a measure of validation, the pressure coefficient distributions at the various flow conditions of the compromise airfoil are displayed in Fig. 7. These profiles behaved as expected. Drops in the pressure profile correspond to dips in the airfoil surface. Similar investigations indicated that the flow would separate from the upper surface near the trailing edge of the best aeroacoustic airfoil in Fig. 6. A slight pressure recovery exists along the upper surface of the aerodynamic best airfoil. Additional validation was made by using the Euler-solver/integral boundary-layer code MSES[32] to predict pressure distributions and aerodynamic coefficients for unconventional airfoils. The predictions of XFOIL and MSES showed good agreement, varying by 2% or less. Reference 29 presents details of these comparisons.

Figure 8 shows the individuals comprising the estimated Pareto-optimal set. Scanning from left to right across each row relates the set as traversed from the best aerodynamic, but worst aeroacoustic, airfoil to the best aeroacoustic, but worst aerodynamic, airfoil. Many of the features evident on the airfoils appear reasonable when examined independently. Most airfoils have camber toward the leading edge. Increased camber benefits c_ℓ while positioning the camber toward the leading edge reduces the aerodynamic pitching moment. Reflexed trailing edges produce a restoring moment, further reducing c_m. Larger radius leading edges help prevent flow separation at higher angles of attack, whereas thinner airfoils have less profile drag and reduced thickness noise. Individuals near the aerodynamic edge of the Pareto front maintain laminar flow over 80% of the lower surface and 20% of the upper surface for the multiple flow conditions.

Perhaps the most unusual feature are the "waves" in the upper and lower surfaces that grow more pronounced in airfoils toward the aeroacoustic side of the Pareto front. This feature appears responsible for lower estimates of OASPL, even though thickness and loading noise predictions remain essentially constant or increase for airfoils closer to the aeroacoustic best airfoil. To illustrate this, Fig. 9 presents WOPWOP calculated noise values for the aeroacoustic best airfoil. The aeroacoustic fitness function [Eq. (9)] is the sum of the OASPL values from both observer locations for each flow condition. The OASPL value for this airfoil is lower than either the thickness or loading measures for most of the observers and flow conditions, suggesting that a cancellation occurs between the thickness and loading noise. This is true for airfoils with the wavy surface feature. Although this cancellation may be nonphysical, it is an interesting, atypical feature that was discovered with the GA. The noise cancellation of these waves needs further study. If the analysis technique, rather than a physical phenomenon, creates this cancellation, using the thickness and loading noise separately in the evaluation of the aeroacoustic fitness instead of OASPL can remedy this.

Comparisons with existing airfoils additionally increase the validity of these nontraditional rotor airfoil shapes. Figure 10 compares the GA-generated compromise airfoil against the NACA 0012 (Ref. 33) and the Boeing Vertol VR-7 (with 0 deg T.E. tab).[34] The NACA 0012 is the airfoil of Prouty's example helicopter.[27] The VR-7 airfoil is an airfoil shape developed specifically for a helicopter rotor. In the plots, the published results are taken from Refs. 33 and 34 for Reynolds numbers closest to those of the design flow conditions; this provides an idea of how closely the predictions agree with experimental data. No published data for the VR-7 matched the Reynolds number of flow condition 3. The combination of XFOIL and WOPWOP used to provide fitness values for the GA produced the remaining values shown in these plots. The GA compromise airfoil compares favorably in light of the specified design constraints. Before making detailed comparisons, one should note that XFOIL was unable to converge on a boundary-layer solution for the NACA 0012 at the second flow condition (corresponding to $\Psi = 180$ deg). This may suggest drag predictions of less than desirable accuracy. Observations taken over several runs indicate that the second flow condition was the most difficult of the three to obtain sufficient boundary-layer solutions.

Although the NACA 0012 performs well in pitching moment, it displays shortcomings in all other categories. The VR-7 performs

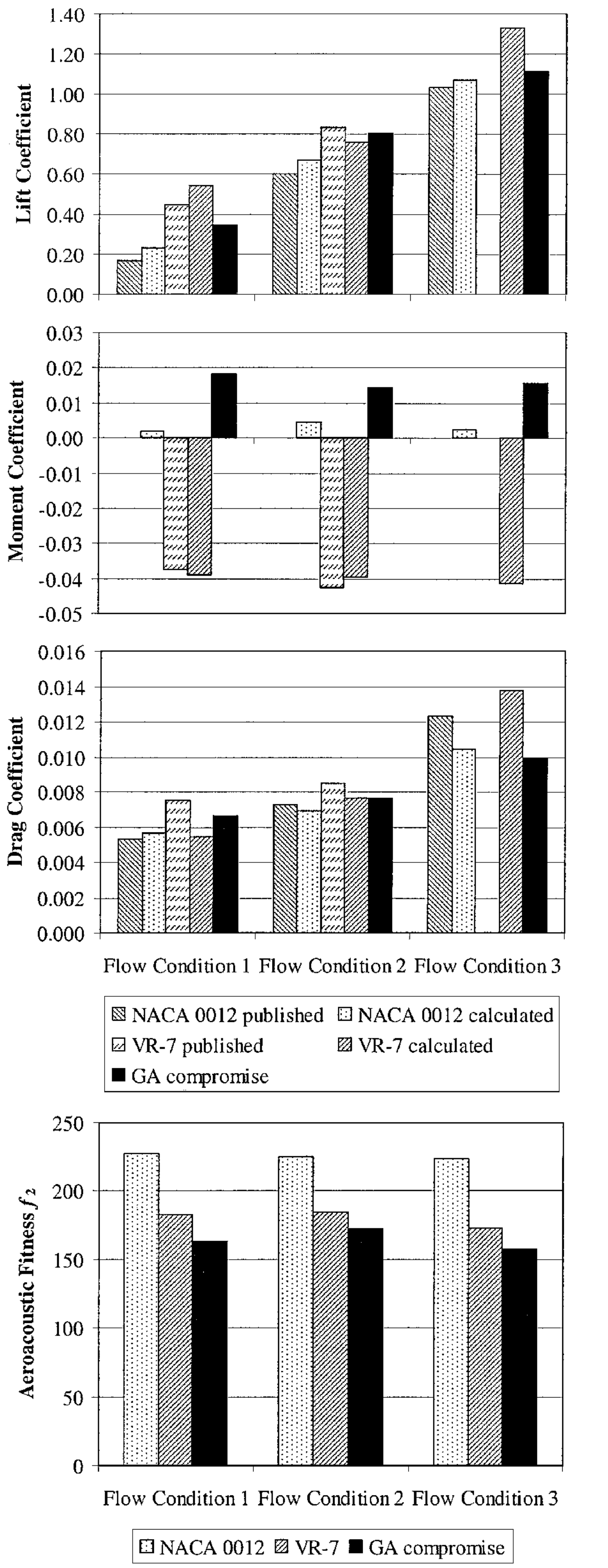


Fig. 10 Comparison of GA-generated compromise airfoil to existing airfoils.

well in all categories, with the exception of moment. The GA compromise airfoil is predicted to be quieter than the VR-7, while maintaining good aerodynamic performance. Relaxing the c_m constraint would allow a more direct comparison to the VR-7. Considering that the GA had no incentive to improve c_ℓ above the NACA 0012 c_ℓ constraints, the GA compromise airfoil compares favorably. Herein lies an advantage of the n-branch GA; developing the Pareto front allows a designer to consider tradeoffs between the different objectives and select an airfoil suitable to the stated needs. For example, the best feasible aerodynamic airfoil generated by the GA has a better predicted aerodynamic fitness function than either the NACA 0012 or the VR-7. Given the level of analysis tools employed, the GA-generated airfoils appear valid. However, these shapes should not be viewed as final airfoil designs but as starting points for further refinement.

Discussion

Even with feasibility classes and adaptive single-objective/multiobjective fitness evaluation, the GA expended most of its effort evolving a solvable population from a vast design space. At the end of a run, only 75% of the population were viable designs (XFOIL's boundary-layer solution scheme had converged), and only 25% of the population were feasible designs (viable designs with all constraints satisfied). These low percentages result because the search ability of the GA that allows for the discovery of nontraditional shapes also requires XFOIL and WOPWOP to evaluate designs that may be beyond their intended scope. The most likely solution to this problem is the development of more robust analysis tools capable of evaluating unusual shapes, although this would create additional computational expense.

Concerns for Application

The GA can generate nonintuitive shapes because of its population-based search and global optimization behavior. However, like most optimization techniques, the GA will exploit limitations of the analysis methods and shortcomings in the problem formulation. For example, airfoils can encounter transonic effects at some of the flow conditions, yet wave drag is not addressed with the current analysis. The drag calculations for unconventional airfoil shapes may be suspect; a higher-order computational fluid dynamics code including turbulence modeling may be needed. Similarly, the present aeroacoustic analysis neglects HSI, BVI, or broadband noise associated with separation-induced unsteady flow. The resulting airfoils are only as good as the ability of the analyses to predict the airfoils' performance. Because of the aforementioned concerns, additional analysis of the two-dimensional airfoils using higher-order tools and modeling and analysis of a three-dimensional rotor using these shapes are required before the GA-generated airfoils should be considered for application.

Parallel Execution

The additional capability obtained through parallel operation cannot be underestimated. The manager/worker parallel GA achieved a speed up close to the ideal $1/(n-1)$ value when solving the airfoil problem on an IBM SP2. The test documented in Fig. 11 plots the solution time for five generations of the rotor airfoil problem against an increasing number of single processor nodes. Although restricted in the number of generations to conserve computing allocation, this study shows a strong similarity between the actual and ideal speed-up values.

The performance displayed in Fig. 11 was expected for several reasons. The problem has a significantly high ratio of computation time to communication time. The communication times are on the order of microseconds, but airfoil evaluations averaged approximately 16 s each on this system. On average, a XFOIL analysis

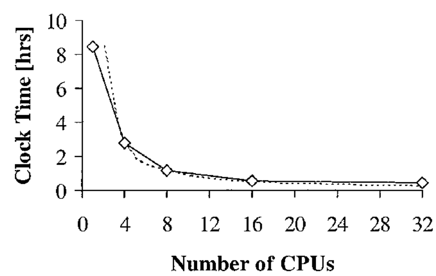


Fig. 11 Distributed system scalability study: ◇, actual; and · · · ·, ideal.

takes 3.1 times longer than a WOPWOP analysis. For some cases the actual and ideal wall clock times differ slightly. Toward the beginning of the trend line, the difference results from the lack of a two-node (one manager–one worker) computation. For this case the total wall clock time would actually increase above the serial (one node) computation because of the addition of communication time.

The remaining discrepancies result from varying numbers of fitness evaluations needed during the different runs. Although the same random seed was used in all runs to provide the same initial generation, the differing number of nodes used allowed the GA to process the individuals in different orders. The worker nodes returned individuals to the manager node as the fitness evaluations were completed, and the manager placed these in the population in the order they were returned. With different numbers of worker nodes, the return order varied so that the pairs of individuals selected as parents varied with the number of workers. For example, this reordering required the reparenting of 1724 individuals during the 32-node run, whereas the 16-node run replaced 822 individuals. This negated much of the expected speed up between the 16- and 32-node runs. The discrepancy resulting from workers idled between generations while the manager performs the genetic and statistical operations also remains. Despite these details, the dynamically load balanced parallel GA scales quite well as the number of CPUs increases.

Conclusions

Given a problem representing a helicopter rotor airfoil and using the codes XFOIL and WOPWOP, the GA generated a set of rotor airfoil shapes representing compromises between aerodynamic efficiency and minimum noise. Based upon predictions from the low-order analysis tools employed, the resulting nontraditional shapes appear to offer good aerodynamic and aeroacoustic performance. The n-branch tournament selection allowed a range of designs to be generated in a single run of the GA, which lets designers choose an airfoil best suited to their needs. Also, the inclusion of advanced feasibility handling and generationally adaptive fitness function evaluation allowed the GA to successfully evolve a population in a design space containing large regions of unsolvable airfoil shapes. Among the generated shapes, airfoils with waves in the upper and lower surfaces have predicted reductions in the overall averaged sound-pressure level. This feature would likely not have been discovered using a traditional design approach. In view of the analysis tools used, the GA-generated designs provide intriguing starting points that require further development prior to application.

Using MPI to implement a manager/worker parallel model made this computationally expensive problem tractable. The inherently parallel structure of the GA made the implementation straightforward, and dynamic load balancing added efficiency. The rotor airfoil problem maintained a very high ratio of computation to communication. Consequently, the problem's scalability closely matched the ideal value of $1/(n-1)$. The parallel GA generated solutions that would not have been obtained using other optimization methods and did so in reasonable times.

Acknowledgments

Some calculations were performed on the 128-node IBM SP2 at the San Diego Supercomputing Center, a member of the National Partnership for Advanced Computational Infrastructure. Additional calculations were performed on the PAPERS cluster in the School of Electrical Engineering at Purdue University with the assistance of Hank Deitz and his students. Mark Drela of the Massachusetts Institute of Technology supplied the code XFOIL. Kenneth Brentner of NASA Langley Research Center provided the code WOPWOP along with support and advice for this endeavor.

References

[1]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989.

[2]Doorly, D. J., and Peiro, J., "Supervised Parallel Genetic Algorithms in Aerodynamic Optimization," *Proceedings of the 13th AIAA Computational Fluid Dynamics Conference*, AIAA, Reston, VA, 1997, pp. 210–216.

[3]Obayashi, S., and Takanashi, S., "Genetic Optimization of Target Pressure Distributions for Inverse Design Methods," *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 881–886.

[4]Quagliarella, D., and Della Cioppa, A., "Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils," *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 889–891.

[5]Fanjoy, D. W., and Crossley, W. A., "Aerodynamic Shape Design for Rotor Airfoils via Genetic Algorithm," *Journal of the American Helicopter Society*, Vol. 43, No. 3, 1998, pp. 263–270.

[6]Wells, V. L., Han, A. Y., and Crossley, W. A., "Acoustic Design of Rotor Blades Using a Genetic Algorithm," *Proceedings of the AGARD Symposium on Aerodynamics and Aeroacoustics of Rotorcraft*, AGARD, Quebec, Canada, 1995, pp. 35-1–35-10.

[7]Doorly, D. J., Peiro, J., and Oesterle, J. P., "Optimization of Aerodynamic and Coupled Aerodynamic-Structural Design Using Parallel Genetic Algorithms," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Vol. 1, AIAA, Reston, VA, 1996, pp. 401–409.

[8]De Falco, I., Del Balio, R., Della Cioppa, A., and Tarantino, E., "Breeder Genetic Algorithms for Airfoil Design Optimisation," *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, ICEC '96*, IEEE Publications, Piscataway, NJ, 1996, pp. 71–75.

[9]Obayashi, S., and Takanashi, S., "Genetic Algorithm for Aerodynamic Inverse Optimization Problems," *Proceedings of the 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA '95*, Inst. of Electrical Engineers, Stevenage, England, U.K., 1995, pp. 7–12.

[10]Obayashi, S., Jeong, S., and Matsuo, Y., "New Blunt Trailing-Edge Airfoil Design by Inverse Optimization Method," *Journal of Aircraft*, Vol. 34, No. 2, 1997, pp. 255–257.

[11]Mantel, B., Periaux, J., Sefrioui, M., and Stoufflet, B., "Evolutionary Computational Methods for Complex Design in Aerodynamics," AIAA Paper 98-0222, Jan. 1998.

[12]De Falco, I., Del Balio, R., Della Cioppa, A., and Tarantino, E., "A Parallel Genetic Algorithm for Transonic Airfoil Optimisation," *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, ICEC'95*, Vol. 1, IEEE Publications, Piscataway, NJ, 1995, pp. 429–434.

[13]Obayashi, S., Tsukahara, T., and Nakamura, T., "Cascade Airfoil Design by Multiobjective Genetic Algorithms," *Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Inst. of Electrical Engineers, London, 1997.

[14]Obayashi, S., Takahashi, S., and Fejtek, I., "Transonic Wing Design by Inverse Optimization Using MOGA," *Proceedings of CFD98, The Sixth Annual Conference of the Computational Fluid Dynamics Society of Canada*, CFD Society of Canada, Ottawa, ON, Canada, 1998, pp. II-41–II-46.

[15]Obayashi, S., "Multidisciplinary Design Optimization of Aircraft Wing Planform Based on Evolutionary Algorithms," *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, IEEE Publications, Piscataway, NJ, 1998, pp. 3148–3153.

[16]Lee, J., and Hajela, P., "Parallel Genetic Algorithm Implementation in Multidisciplinary Rotor Blade Design," *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 962–969.

[17]Celi, R., "Recent Applications of Design Optimization to Rotorcraft—A Survey," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 176–189.

[18]Carroll, D. L., "Chemical Laser Modeling with Genetic Algorithms," *AIAA Journal*, Vol. 34, No. 2, 1996, pp. 338–346.

[19]Obayashi, S., Takahashi, S., and Takeguchi, Y., "Niching and Elitist Models for MOGAs," *Parallel Problem Solving from Nature—PPSN V*, Lecture Notes in Computer Science, Vol. 1498, Springer-Verlag, Berlin, 1998, pp. 260–272.

[20]Punch, W. F., Averill, R. C., Goodman, E. D., Lin, S.-C., Ding, Y., and Yip, Y. C., "Optimal Design of Laminated Composite Structures Using Coarse-Grain Parallel Genetic Algorithms," *Computing Systems in Engineering*, Vol. 5, No. 4–6, 1994, pp. 415–423.

[21]Henderson, J. L., "Laminated Plate Design Using Genetic Algorithms and Parallel Processing," *Computing Systems in Engineering*, Vol. 5, No. 4–6, 1994, pp. 441–453.

[22]Crossley, W. A., Cook, A. M., Fanjoy, D. W., and Venkayya, V. P., "Using the Two-Branch Tournament Genetic Algorithm for Multiobjective Design," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 261–267.

[23]Vincent, T. L., and Grantham, W. J., *Optimality in Parametric Systems*, Wiley, New York, 1981, pp. 71–101.

[24]Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," *Proceedings of the Third International Conference on Aerodynamics at Low Reynolds Numbers*, Springer-Verlag, New York, 1989, pp. 1–12.

[25]Brentner, K. S., "Prediction of Helicopter Rotor Discrete Frequency Noise," NASA TM 87721, Oct. 1986.

[26]De Boor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 1978, pp. 108–164.

[27]Prouty, R. W., *Helicopter Performance, Stability and Control*, Krieger, Malabar, FL, 1990, pp. 140–162.

[28]Williams, E. A., and Crossley, W. A., "Empirically-Derived Population Size and Mutation Rate Guidelines for a Genetic Algorithm with Uniform Crossover," *Soft Computing in Engineering Design and Manufacturing*, edited by P. K. Chawdhry, R. Roy, and R. K. Pant, Springer-Verlag, London, 1998, pp. 163–172.

[29]Jones, B. R., "Aerodynamic and Aeroacoustic Optimization of Rotor Airfoils via Parallel Genetic Algorithm," M.S. Thesis, School of Aeronautics and Astronautics, Purdue Univ., West Lafayette, IN, Dec. 1998.

[30]Syswerda, G., "Uniform Crossover in Genetic Algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1989, pp. 2–9.

[31]Deitz, H. G., Muhammad, T., Sponaugle, J. B., and Mattox, T., "PAPERS: Purdue's Adapter for Parallel Execution and Rapid Synchronization," School of Electrical Engineering, Purdue Univ., TR-EE 94-11, West Lafayette, IN, March 1994.

[32]Drela, M., *A User's Guide to MSES 2.8*, MIT Computational Aerospace Sciences Lab., Massachusetts Inst. of Technology, Cambridge, MA, May 1995.

[33]Abbott, I. H., and Von Doenhoff, A. E., *Theory of Wing Sections*, Dover, New York, 1959, pp. 462, 463.

[34]Dadone, L. U., "U.S. Army Helicopter Design Datcom: Volume I—Airfoils," U.S. Army Air Mobility R&D Lab., USAAMRDL CR 76-2, Moffett Field, CA, May 1976.