Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design

Murray B. Anderson*, Glenn A. Gebert**
Sverdrup Technology Inc./TEAS Group

Abstract

This study examines the use of pareto genetic algorithms to determine high efficiency wing geometries. Applying genetic algorithms to aerodynamic shape optimization is a fairly new concept. This study demonstrates the capability of pareto genetic algorithms to determine highly efficient and robust wing designs given a variety of design goals and constraints. The design study executed by the genetic algorithm and presented in this report document the learning capability of genetic algorithms, and the power of pareto genetic algorithms for multi-disciplinary optimization. Results indicate that the pareto genetic algorithm is able to develop high efficiency wing geometries while maintaining an adequate lifting performance and load-bearing structure.

Introduction

In recent years researchers have applied gradient-based optimization schemes to aerodynamic design[1-3], but these methods are subject to several undesirable restrictions. First, these methods require knowledge of the aerodynamic derivatives for each parameter, or combination of parameters for higher order coupling. These derivatives may not be easily determined and must be continually recalculated as the design moves through the design space since few aerodynamic design problems have linear derivatives, especially in multiple dimensions. Second, gradient-based optimizers must start with a specified set of initial parameters, which can bias future solutions toward a local optimum in the viscinity of the starting point. Third, gradient-based methods cannot be applied to problems where there are discontinuities in the design space because the derivatives in these regions are not defined. Discontinuities are common in aircraft design. As Gage[4] points out, there are no partial engines or partial seats on an aircraft.

Previous Work

Only recently have attempts been made to couple artificial intelligence (i.e. learning) techniques to aerodynamic design. The research of Gage and Kroo[4] was focused on applying genetic algorithms to the topological design of nonplanar wings. In their work the wing was broken into several segments which were allowed to vary in incidence and dihedral angles. The goal of the optimization was to minimize induced drag given a fixed lift. Their approach used both a penalty and repair approach to deal with solutions not achieving the fixed lift value. Bramlette and Cusic[5] have also applied genetic methods to the parametric design of aircraft and Tong[6] has used genetic algorithms to conduct preliminary design of turbine engines after gradient methods had stalled in a local optimum. More recently Anderson[7] has applied genetic algorithms to subsonic wing design with the goal of producing good aerodynamic shapes, but also given the additional constraint that the structure must not break. Anderson used penalty weights to combine Lift, L/D ratio, and structural design margins into one global objective function. Anderson, like Gage[4], pointed out that the achieved solution is strongly dependent upon the values of the penalty weights, a very unappealing result. This current work was designed to eliminate the need for penalty weights by treating each design goal separately using a Pareto genetic algorithm. The genetic software used for this

* Senior Engineer, Member AIAA
** Technical Fellow, Member AIAA

Copyright © 1996 by Murray B. Anderson. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

study was the IMPROVE (Implicit Multi-Objective Parameter Optimization Via Evolution) code[8], a general purpose genetic algorithm package which can easily be interfaced to a variety of objective functions.

Geometry Optimization with Genetic Algorithms

Genetic algorithms are parametric search procedures that mimic the mechanics of natural genetics; a given population will propagate both good and bad characteristics to the next generation. The Darwinian theory of evolution states that good characteristics, which help a member of the population survive, will gradually dominate the population as the members with bad characteristics die off. The genetic algorithm used in this study preserves the Darwinian approach of species optimization, and is often referred to as a simple genetic algorithm because the population is governed by natural selection with crossover, reproduction, and mutation as the only operators on the population.

Many gradient search (i.e. sensitivity derivative) techniques have been applied to geometric shape optimization. Genetic algorithms have many advantages over gradient/sensitivity search methods:

a. they search from a population and not a single parameter set.

b. they can provide the global maximum (for maximization problems) and not a local maximum.

c. they can operate on irregular functions (step functions).

d. they do not require initial estimations of the parameters.

e. they operate on a coding of the parameter set, not the parameters themselves.

Unlike gradient search routines, which are deterministic based on a set of criteria, genetic algorithms are probabilistic because they work with a coding of the entire parameter set rather than individual parameters. For this study, a binary coding of the parameters of interest was chosen. Each parameter is discretized based upon its minimum and maximum value and its resolution (or accuracy requirement) into N-bit binary unsigned integers called substrings. For example, if a string length for a parameter were 4 bits long, the substring 0000 would form the lower bound and 1111 would from the upper bound. Each substring, representing a parameter, is strung together to form a single string which represents a member of the genetic algorithm population. Upon initialization, the total population is formed by randomly generating strings.

Following initialization, the genetic algorithm approach is governed by three operations: reproduction, crossover, and mutation. Reproduction is the process by which a string transitions into the next generation. Crossover can also be called mating because it is the process by which two strings exchange binary digits. Crossover is usually performed following reproduction and is a three-step process as follows:

a. The current generation is paired through tournament selection, and crossover is performed depending upon a preselected value of the crossover probability, P_c.

b. A break point, P, chosen at random for each pair of strings.

c. The bits to the right of position P are interchanged between the two strings as shown in figure 1.

For the current study, the crossover probability was set to 70 percent which allowed crossover of genetic material between two members of the population 7 out of 10 times that two strings were paired. A moderate probability of crossover is preferred for simple genetic algorithms[9].

Mutation is the random altering of one of the binary bits in a string (figure 1). Through mutation, the

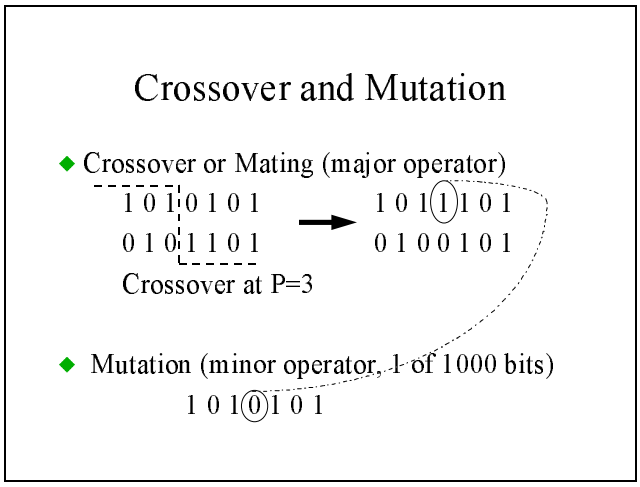

Figure 1. Crossover and Mutation

global maximum of a function is achieved rather than a local maximum. Mutation probabilities, P_m, are usually kept very low to prevent complete randomness in the solution process. For the current study, the mutation rate was set to 1 percent (1 out of 100 bits).

Pareto genetic algorithms[10,11] differ from generational genetic algorithms in that they operate on multiple goals or objectives simultaneously rather than having one "fitness" function. The goals are optimized individually to determine parameter sets which work well for each goal. Through the mating of good parameter sets for each individual goal, a family of parameter sets which work well across the spectrum of goals is obtained. This family of solutions is called the pareto optimal (p-optimal) set for the multi-objective/multi-goal problem. The operation of the pareto scheme in the genetic algorithm software is by domination. One goal set, defined as the collection of individual goal performances based on one parameter set, must clearly dominate another if the parameter set is to survive. Goal set **A** dominates set **B** if the following two statements are true:

$$\forall_i (\mathbf{A}_i \geq \mathbf{B}_i) \quad \text{and} \quad \exists_i (\mathbf{A}_i > \mathbf{B}_i)$$

When two members of the population are chosen for the tournament selection procedure, the domination rules are examined to see whether one member dominates the other. The clear winner is retained for the next generation. If there is no clear winner, a situation called nondomination, the winner is selected at random for survival.

Problem Definition

For this study, the genetic algorithm will attempt to evolve a family of wing designs with good aerodynamic and structural properties. Since finite wing theory[12] is used to provide the aerodynamic data for the wings, the flight conditions are subsonic (200 ft/sec at 1000 ft altitude) with a moderate angle of attack (10 deg). For this study the wing planform shape, wing twist, wing thickness distribution, and nonplanar wing dihedral/anhedral are the general wing characteristics that the genetic algorithm will manipulate. Each of these four characteristics are allowed to vary according to prescribed methods symmetric about the wing root chord.

Wing Planform. The first wing characteristic to be defined in this study is the wing planform. The physical constraints for the planform are defined as follows:

$$C_i = \eta B + B\left[\varepsilon\left(1 - x_i\right)^2 + \lambda\left(1 - x_i\right)\right] \quad (1)$$

where C_i is the chord length at spanwise stations x_i, η is the minimum chord length (in percent) of the wing span B, ε and λ are the variables which govern the variation of the chord with spanwise station. A planar wing, using finite wing theory, has one solution that produces the highest lift-to-drag ratio (L/D): a straight wing of minimal chord (i.e. highest possible aspect ratio). Finding this optimum solution has proven to be an easy task for a genetic algorithm in previous studies[7]. For this study the span was fixed to 20 feet and the minimum chord was set to 5 percent of the span ($\eta = 0.05$) or 1 foot.

Wing Twist. According to basic aerodynamic principles of finite wings, an improved aerodynamic solution can be obtained by twisting the wing tips

down to reduce the effective angle of attack at the wing tips, and therefore reduce the induced drag of the wing. The wing twist formulation is as follows:

$$\theta_i = \mu x_i^2 + \nu x_i \qquad (2)$$

where q_i is the local twist angle at spanwise stations x_i, and μ and ν are the variables which govern the variation of twist along the span. The optimum values of μ and ν should combine to twist the wing tips down into the incoming flow. This effect is called washout, and has been used to reduce induced drag on many aircraft wings.

Wing Thickness Distribution. For this concept demonstration study, a simple thin-skinned shell was used to define the wing[13] structure. It was assumed that the wing skin was made of 1/16 inch sheet aluminum with a density of 63 lbm/ft^3. Structural integrity was determined simply by calculating the shear stress and bending moment at each spanwise station and determining whether the structure available at that location was capable of bearing the load[13]. As expected, the structural integrity of a given wing design was essentially determined by examining the loads at the wing root, where the bending moment is the highest, and determining whether the existing cross sectional structural area at that location was capable of bearing the load. The genetic algorithm formulation had two mechanisms available to maintain structural integrity of any spanwise station while minimizing structural weight. First, the genetic algorithm could vary the chord length according to equation 1, thereby varying the structural distribution through chord length variations. Second, a new variable was added to the formulation to allow the structure to vary in thickness (T) with span according to the following equation:

$$T_i = 0.05C_i + \omega C_i\left(1 - \frac{x_i}{B/2}\right) \qquad (3)$$

This equation gives the genetic algorithm a second way to increase the cross sectional structural area at a spanwise station x_i, through increasing the parameter ω. According to equation 3, the thickness at any span station could not be less that 5 percent of the chord length at that station.

Nonplanar Geometry. An additional set of parameters was added to include out-of-plane possibilities in the design. The genetic algorithm could, if it proved valuable, arch the wing by varying the height of the wing along the span according to the following equation:

$$H_i = \gamma x_i^2 + \delta x_i \qquad (5)$$

where H_i is the height of station x_i relative to the wing root station. This equation gives the genetic algorithm the ability to vary the height (and also slope DH/Dx) at each spanwise station.

The γ and δ parameter bounds essentially allowed the height of each section to vary vertically with span from 0 (no variation) to both positive and negative values that exceed the actual span in magnitude. In order to keep the actual span from exceeding the 20 foot limit, the true span was computed by tracking the length along the curvature of the wing and stopping at 20 feet. For example, a wing having a δ value of 10 (linear 45 degree height variation with span) would have a semi-span of 10*cos(45), or 7.071 feet.

Design Goals. There are two classes of design goals considered under this study: aerodynamics and structures. Beyond the conceptual design phase, ease of manufacturing is one important factor which should also be considered, however, such considerations are not included in this study. The aerodynamic goals of this study are straightforward: produce a high lift-to-drag (L/D) ratio and a specified amount of lift. Likewise, the structural consideration is also simply stated: do not design a wing which will break under the applied loads. The genetic algorithm formulation for these two general classes of design considerations is manifested in two primary ways. First, the genetic algorithm penalizes solutions not achieving the desired lift value even if the resulting structural design would not fail due to loading. Second, the genetic algorithm punishes any solutions which fail the structural integrity test. Inclusive in these considerations is the need to minimize structural weight (W) and produce a high L/D ratio. Minimal structural weight considerations are not explicitly

stated as a goal, however, the implementation of the structural integrity goal contains penalties for solutions which "overbuild" the structure beyond a specified design margin. This mixing of attributes, goals, and design parameters calls for a learning procedure that can produce reasonable hybrid solutions which work well for each goal. The three explicit goals for this study are defined as follows:

Goal 1-Maintain Structural Integrity, and attempt to produce solutions with a 20 percent design margin by using the following relationship:

$$f = \frac{A_{cross}}{A_{req}} \qquad for \ \frac{A_{cross}}{A_{req}} \le 1.2$$

$$f = 1.2 - \left(\frac{A_{cross}}{A_{req}} - 1.2 \right) \qquad elsewhere$$

where A_{req} is the minimum structural cross sectional area capable of bearing the applied aerodynamic load. This formulation means that the peak structural performance value (*f*) is 1.2, and any genetic solutions which provide too little or too much cross sectional area will not achieve the optimum value.

Goal 2-Produce a specified lift value L_{des} (1000 lbf) by rewarding/penalizing genetic solutions using the following relationship:

$$g = \frac{L}{L_{des}} \qquad for \ \frac{L}{L_{des}} \le 1.0$$

$$g = 1 - \left(\frac{L}{L_{des}} - 1 \right) \qquad elsewhere$$

The g-function is optimal at a lift ratio of 1.0, with no margin, so solutions producing excessive lift are penalized as are solutions producing too little lift.

Goal 3-Produce the highest L/D ratio possible.

Genetic Setup. Since genetic algorithms only require the maximum, minimum, and required accuracy of each parameter, the global constraints shown in table 1 were used to define the bounds of the seven parameters used in this study.

ε	0.0	1.0	0.01
λ	0.0	1.0	0.01
μ	-1.0	1.0	0.1
ν	-1.0	1.0	0.1
ω	0.0	0.15	0.01
γ	-1.0	1.0	0.1
δ	-1.0	1.0	0.1

Table 1. Parameter Constraints

These parameter bounds dictate that each member of the genetic population will have 41 bits. This many bits translates into 2^{41} (over 2 trillion) possible solutions. The population size was set to 80 in order to keep the population size larger than the number of bits[8,9]. Recall that the mutation rate was set to 1 percent.

Results

Convergence History. Figures 2, 3, and 4 illustrate the convergence history of the three goals. Each goal shows a typical convergence history of the pareto genetic algorithm implemented in this study. The pareto genetic algorithm quickly finds niches of good performance in each goal (by generation 10),

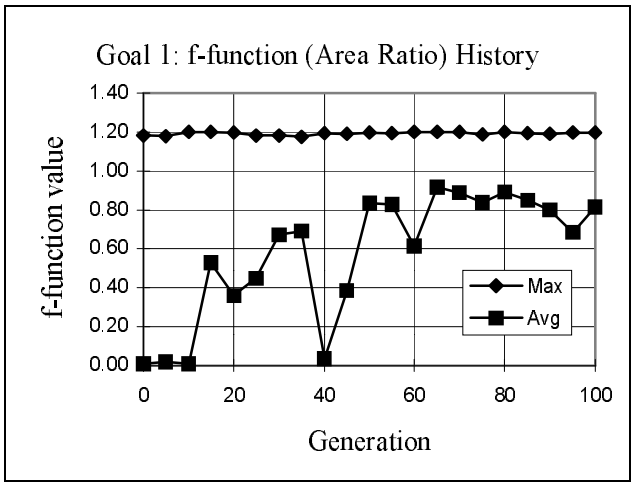

Figure 2. *f*-Function History

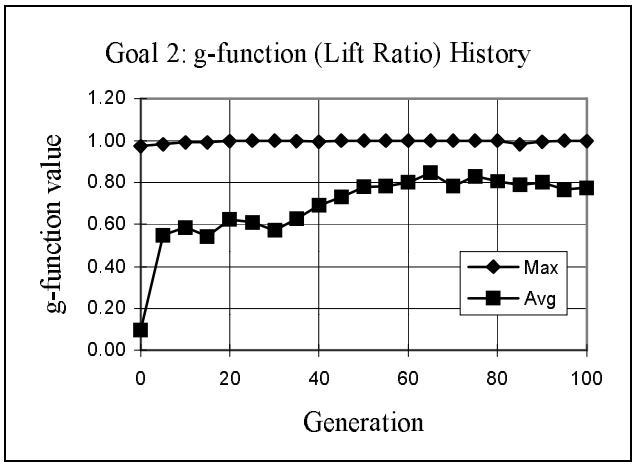

Figure 3. g-Function History

however it takes many more generations until the average population member exhibits reasonable performance. By generation 80 the genetic algorithm had essentially reached a steady state condition, with no clear improvement in the average population member. In a sense, the algorithm had "converged". As can be clearly seen in these figures, the average population member never reaches the maximum achieved value in each niche. The IMPROVE code was designed to prevent (as an option) exact chromosome duplicates in the population so that duplicate solutions would not run in the same generation. Rather, the detected duplicates would undergo random mutation of one bit prior to evaluation. One side effect of this approach, however, is that the effective mutation rate actually increases over time as higher selection pressure is placed on the population survivors. As a result, the average population member will never achieve the performance of the maximum performer in each goal. This fact is not a concern, however, since it is highly unlikely that a best performer in a niche will also be the best hybrid performer across multiple goals.

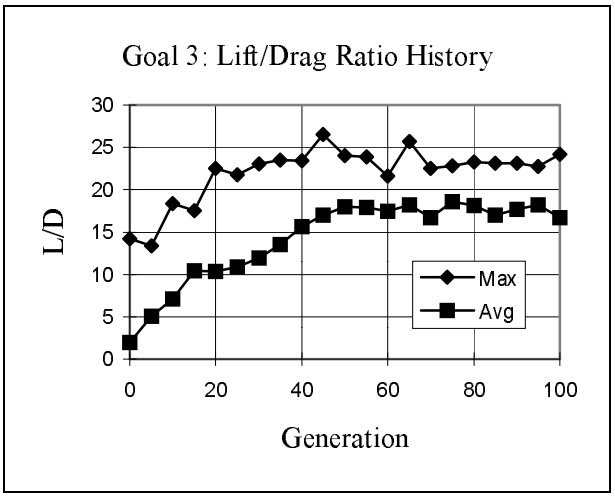

Figure 4. L/D Ratio History

The Evolution of the Design. The following figures illustrate the evolution of the wing from generation 0 (random initialization) through 80. Since pareto optimization produces a family of solutions, the authors selected one of the more desirable wing designs at the noted generations simply for illustration purposes. Non-domination is a basic tenant of pareto optimization, so selection of a single representative from a given population is subjective. All views are from 30 deg elevation and 20 degree yaw angle to provide perspective to the designs.

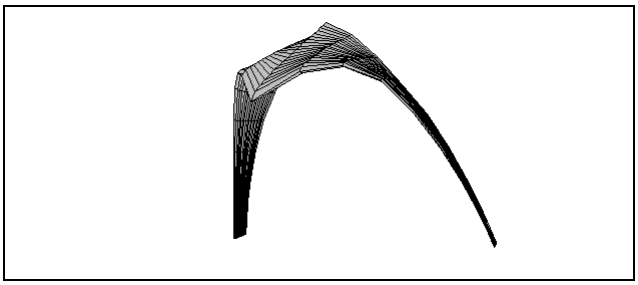

Figure 5. Generation 0

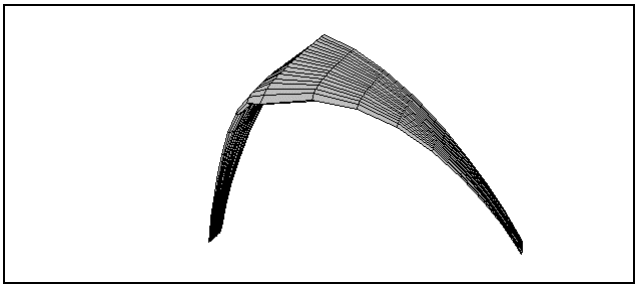

Figure 6. Generation 20

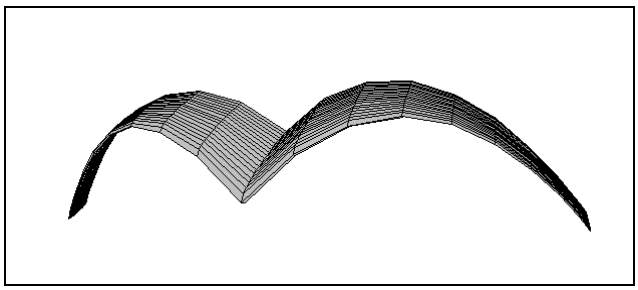

Figure 7. Generation 40

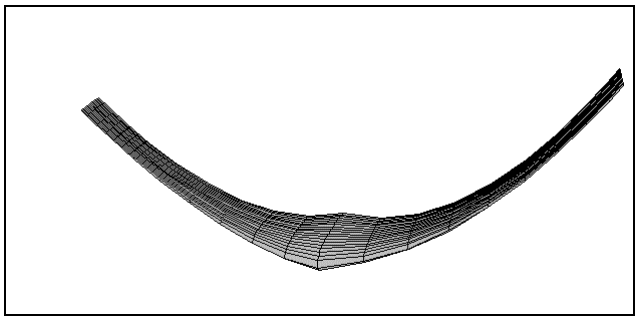

Figure 8. Generation 60

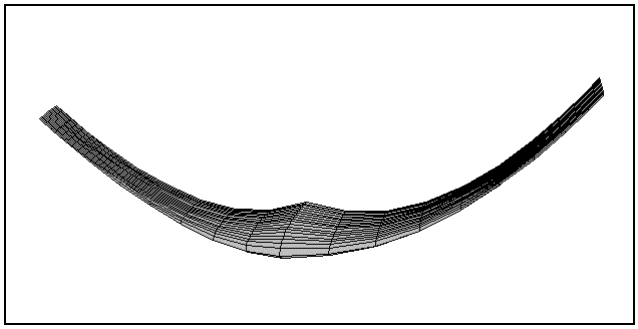

Figure 9. Generation 80

As these figures show, the genetic algorithm started with a very crude anhedral design in generation 0, and began refining this design through generation 40. These figures show that the genetic algorithm learned some important design lessons, such as washout and taper, and that nonplanar wing designs can help reduce bending moments. Figure 7 is especially illustrative of this fact given the unusual anhedral (outboard)/dihedral(inboard) shape of the wing. According to table 5, the genetic algorithm had developed a very good aerodynamic and structural design at this point in terms of the three design goals. The structure was strong enough to handle the loads, the desired lift was produced within a reasonable tolerance, the L/D ratio exceeded 18, and the bending moments had been reduced significantly from generations 20 to 40.

It is also very interesting to note that between generations 40 and 60 the genetic algorithm totally changed the design into an dihedral configuration, which produced a L/D ratio that was nearly two full counts larger. The structural weight and drag both dropped substantially, the wind lift coefficient increased significantly, and the bending moment remained relatively constant. This is illustrative of the power of genetic algorithms when compared to gradient-based optimization methods. The genetic algorithm was able to radically shift its focus from one part of the search space to a portion of the search space which was completely different.

From generations 60 to 80 the dihedral design underwent some smoothing (notice the rounded planform edges compared to generation 60) and slightly relaxed the dihedral angle variation at the inboard stations. These changes increased the L/D ratio to a value of 21.46 and increased the wing lift coefficient to 0.81.

Parameter	Gen 0	Gen 20	Gen 40	Gen 60	Gen 80
Area Ratio	1.24	1.21	1.19	1.21	1.16
Lift (lbf)	1102	913	1062	1006	1069
L/D Ratio	1.72	8.99	18.82	20.41	21.46
Drag (lbf)	640	101	56	49	50
Weight (lbf)	127	152	154	129	132
Bending Moment	1801	2275	1663	1615	1703
Wing C_L	0.79	0.58	0.66	0.77	0.81

Table 5. Summary of Evolving Wing Designs

Figure 10 shows the geometric wing twist variation and associated effective and induced angles of attack

of the wing shown in figure 9. As figure 10 shows, the genetic algorithm learned that washout (negative geometric twist at the wing tips) was a desirable design feature, and twisted the wing down into the flow at the wing tips. Since the wing had significant dihedral, which itself reduces the effective angle of attack of each wing station, the genetic algorithm did not have to twist the wing as much as might be expected given the specified 10 degree angle of attack of the wing root chord.

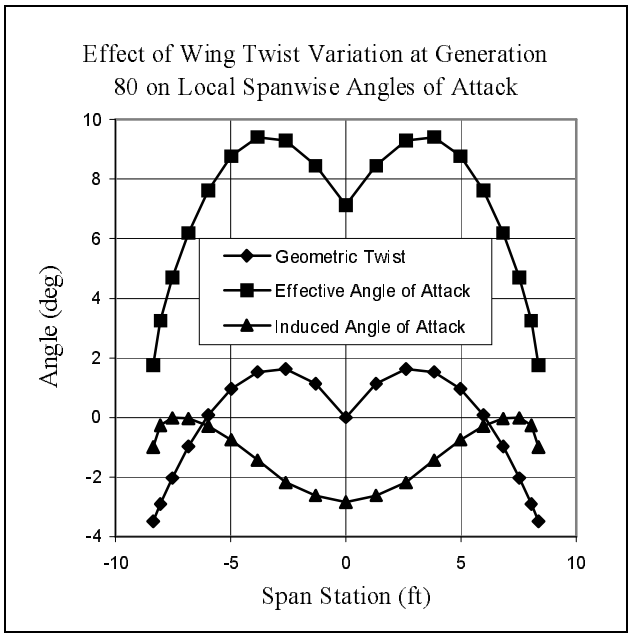

Figure 10. Wing Twist Variation

Conclusions

The genetic algorithm is capable of learning important lessons when designing wing shapes with finite wing theory. This study has shown that the genetic algorithm learned that wing twist can be used to effectively reduce induced drag, and that nonplanar wing shapes can reduce root chord bending moments, thereby reducing the structure required to support the achieved aerodynamic loads.

Given the wide design parameter constraints (i.e. wide geometric freedom) of this study, the resulting optimization space, with over 2 trillion possibilities, requires a significant number of wing solutions (8000 for this study). It is encouraging to note, however, that good wing solutions were obtained in spite of the algorithm sampling only a small fraction (3.6e-07 percent) of the total design space. Since finite wing theory was used for this study, the solutions only required approximately 10 minutes on a 100-megahertz Pentium® personal computer. Applying more sophisticated aerodynamic simulation methods (Euler or Navier-Stokes based solutions) would make applying simple genetic algorithms impractical for such large optimization spaces given current computer hardware.

References

1. Kroo, I., Takia, M., "A Quasi-Procedure, Knowledge-Based System for Aircraft Design," AIAA Paper No. 88-4428, Presented at the AIAA/AHS/ASEE Aircraft Design, Systems, and Operations Meeting, September 1988.

2. Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design,", AIAA Paper No. 94-4325.

3. Gage, P., and Kroo, I., "Development of the Quasi-Procedural Method for Use in Aircraft Configuration Optimization," AIAA Paper No. 92-4693, Presented at the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992.

4. Gage, P., and Kroo, I., "A Role of Genetic Algorithms in a Preliminary Design Environment," AIAA Paper No. 93-3933.

5. Bramlette, M., and Cusic, R., "A Comparitive Evaluation of Search Methods Applied to the Parametric Design of Aircraft", Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, 1989.

6. Tong, S.S., "Turbine Preliminary Design Using Artificial Intelligence and Numerical Optimization Techniques." Journal of Turbomachinery, Jan 1992, Vol 114/1.

7. Anderson, M.B., "The Potential of Genetic Algorithms for Subsonic Wing Design", AIAA Paper 95-3925, presented at the 1st AIAA Aircraft Engineering, Technology, and Operations Congress, Los Angeles, CA, September 1995.

8. Anderson, M.B., "Software and Users Manual for the IMPROVE Code", Copyright 1995.

9. KrishnaKumar, K., R.K. Renjith, and Seywald, H., "Genetic Algorithms Tutorial Course Notes," AIAA sponsored tutorial, Monterey, CA, August 1993.

10. Louis, Sushil J., and Rawlins, Gregory J. E., "Pareto Optimality, GA-easiness and Deception", Proceedings of the Fifth International Conference on Genetic Algorithms, pages 118-123, Morgan Kauffmann, San Mateo, CA, 1993.

11. Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.

12. Keuthe, A.M., and Chow, C., Foundations of Aerodynamics: Bases of Aerodynamic Design, Wiley, New York, 1976.

13. Shanley, F.R., Weight-Strength Analysis of Aircraft Structures, Dover Publications Inc., New York, 1960.