

THE POTENTIAL OF GENETIC ALGORITHMS FOR CONCEPTUAL DESIGN OF ROTOR SYSTEMS

William A. Crossley
Graduate Student

Valana L. Wells
Associate Professor

David H. Laananen
Associate Professor

*Department of Mechanical and Aerospace Engineering
Arizona State University
Tempe, Arizona*

AUG 05 1994

NAG-2-882
Semi-Annual

ABSTRACT

The capabilities of genetic algorithms as a non-calculus based, global search method make them potentially useful in the conceptual design of rotor systems. Coupling reasonably simple analysis tools to the genetic algorithm was accomplished, and the resulting program was used to generate designs for rotor systems to match requirements similar to those of both an existing helicopter and a proposed helicopter design. This provides a comparison with the existing design and also provides insight into the potential of genetic algorithms in design of new rotors.

Key Words: Helicopter Rotor Design, Helicopter Rotor Optimization, Genetic Algorithm Applications

INTRODUCTION

Conceptual design of rotor systems often relies on partially qualitative decisions made by the members of a design team. The selection of airfoil sections for the rotor blades provides one example of a partially qualitative decision. Decisions such as this, by their very nature, can be biased by the experience and personal preferences of the design team. The

(NASA-CR-196813) THE POTENTIAL OF
GENETIC ALGORITHMS FOR CONCEPTUAL
DESIGN OF ROTOR SYSTEMS Semiannual
Report (Arizona State Univ.) 29 p

N95-11699

Unclass

G3/05 0022360

HSI

quantitative decisions also made during rotor system design can be quite complex, and numerical optimization techniques have been introduced to assist a design team in finding the "best" rotor system design. By introducing a design tool that would not only be useful in rotor system sizing where the numerical optimization methods are effective, but that would also provide a logical means for making configuration decisions, the process of conceptual design for rotors may be made simpler and faster. Further improvement in the process results if several design concepts could be evaluated simultaneously by the tool. Design configurations that might have been ignored or dismissed early in the design process by the design team may actually prove to be better design choices. Genetic Algorithms (GA) possess features that can form the basis for such a design tool.

The rapidly growing use of genetic algorithms in engineering problems can be extended to include rotor system design, sizing and selection. A genetic algorithm-based code to design a low-power, light-weight rotor system demonstrates the feasibility of using this type of methodology in conceptual design of rotor systems and suggests extension to the entire realm of rotorcraft design.

The following section provides a brief overview of the major principles of genetic algorithms. This is followed by a discussion of the application of genetic algorithms to rotor system design. Ensuing sections discuss the development and validation of the analysis methodology used in this exercise, and the results of using the GA-based code to design rotor systems for requirements of both an existing and a proposed rotor system. Finally presented are some remarks about further applications and research for genetic algorithms, as well as conclusions reached in the work discussed in this paper.

GENETIC ALGORITHMS-AN OVERVIEW

Computer scientists developed genetic algorithms during the 1960s and 1970s as a programming technique for constructing computer programs¹. It was later realized that this type of algorithm also had applications in other fields, in particular, as an optimization tool. Only recently have applications of GA appeared in aerospace engineering problems². This concept originated in work presented by Holland³, was expanded by Goldberg⁴ and many others, and has now developed into an accepted search and optimization technique. Details of the workings and theory of genetic algorithms are described in a vast collection of references (see Reference 5), so only a brief overview is given here.

Genetic algorithms work by mimicking the "survival of the fittest" patterns of natural selection and reproduction that are displayed in biological populations. Design variables, the "genes" of a given design, are mapped into binary strings. These strings are then concatenated to form the "chromosome" for a combination of variables that represent an individual design point. The genetic algorithm works with these binary chromosomes, rather than with the actual design parameters. This coding feature allows for a simultaneous use of discrete, integer and continuous design variables⁶.

An initial generation is created by randomly placing "1"s and "0"s along the chromosomes for a given population size. For example, a population size of 30 individuals will have 30 different chromosome strings. This population generally maintains the same size throughout the execution of the genetic algorithm. The values of the design variables in each of these individuals is decoded from the binary string through mapping relationships. From these values, a *fitness* value is assigned to each individual. This fitness is analogous to the objective function value in a numerical optimization problem. Individual chromosomes

with high fitness are more likely to survive and be used as "parents" for a subsequent generation of designs.

The search procedure used by GA follows a structured, probabilistic information exchange among the population of design points instead of the more familiar calculus-based search commonly applied in numerical optimization problems. The search includes a crossover process, where a "child" inherits traits from both of its "parents", and a mutation, where a bit in the chromosome string is changed, introducing a trait not seen in either parent. Figure 1 displays the crossover and mutation operators.

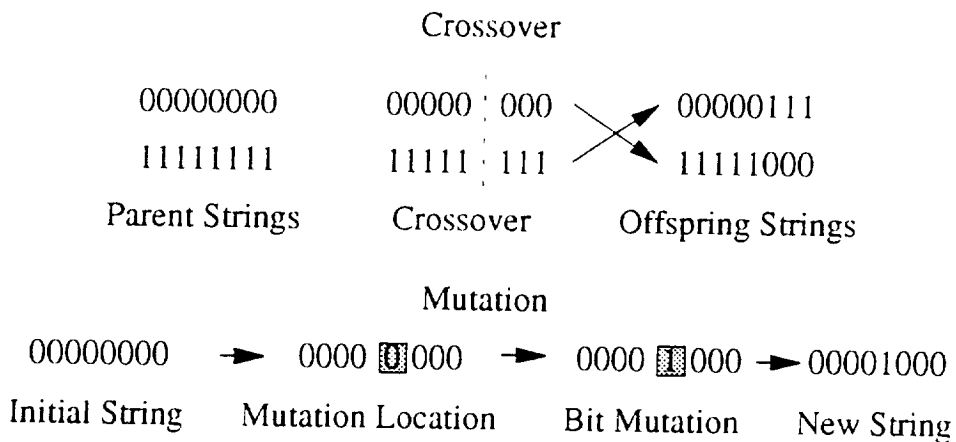


Figure 1. Crossover and Mutation Operators

This procedure benefits from a feature termed *implicit parallelism*⁴. Sections of the binary string, or patterns of "1"s and "0"s in the chromosome, represent characteristics of the design which provide a better than average fitness to the design; for example, a pattern representing low disk loading in a rotor system would be associated with a low-power rotor system. These sections of the string, called schema, are the building blocks of good designs. Individuals in the genetic algorithm population exchange their information, but the building blocks associated with the highly fit individuals propagate through the population with an increasing frequency, because individuals with these schema will have higher fitness

and be more likely to survive and reproduce. Researchers in GA have found that this processing of schema acts at a level near n^3 per generation, where n is the population size³. This implicit parallelism occurs even though the GA requires only n fitness evaluations per generation, implying that the important schema are processed at a rate of the number of fitness evaluations cubed.

GENETIC ALGORITHMS IN ROTOR SYSTEM DESIGN

Rotor system design encompasses many different engineering disciplines and can often be quite complex in nature. Desirable rotor systems require minimum power under various flight conditions, are controllable, have low weight, have low complexity, have low acoustic signature, and possess several other characteristics. The problem of designing a low-power, low-weight rotor system demonstrates the potential for using genetic algorithms as part of a conceptual design tool. Features unique to genetic algorithms provide the capability for this kind of design tool. As with many applications of genetic algorithms to engineering problems, some additions and variations of the basic GA approach were necessary to complete this design task.

Rationale for Genetic Algorithms

Many current efforts to improve rotor systems have revolved around numerical optimization of rotors for various objectives⁷. The genetic algorithm code does not replace these approaches, rather, it augments these other methods. Genetic algorithms were chosen for use in conceptual design for two major reasons. The first of these is the GA's ability to simultaneously consider the three types of design variables (continuous, integer, and discrete), and the second relates to GA's search properties as opposed to those of a calculus-based procedure. Additionally, the search from a population of points, rather than

from a single point, reflects the concept of parallel design efforts for different rotor configurations.

As mentioned before, conceptual design involves decisions, such as selection of airfoil section, based on the experience of the designer. Using discrete variables in a systematic way to represent design choices such as this provides vast capability in a design code. In the more general sense of rotorcraft design, the configuration of the aircraft itself could be considered a design variable. For example, a single variable could represent a single main rotor, a front-to-back tandem rotor, a side-by-side tandem rotor, or a coaxial rotor configuration. The fitness evaluation of individual designs drives the genetic algorithm's search through the design space. By using configuration-type variables in the problem to represent those design features that are normally subjected to qualitative or partially qualitative decision making, the GA helps to give numerical values to some of these traditionally non-quantitative design decisions. The set of variables examined in this work includes a discrete one representing airfoil section along with the integer variable for number of blades, as well as several variables that are continuous (tip speed, disk loading, solidity, linear twist and taper ratio). This provides an automated selection of blade airfoil section as well as determining the number of blades and values for the continuous variables. Simultaneous evaluation of these three types of design variables by the genetic algorithm suggests that GA can attempt a wider problem domain than most traditional numerical optimization methods.

Much of the recent work toward optimal rotor design has yielded promising results, but has relied on calculus-based methods for finding optimal design points. The genetic algorithm, because it is not calculus based, has some advantage as a conceptual design tool. GA can be used as a global optimizer of highly non-smooth and discontinuous functions because it

does not require evaluation of gradients and has no requirement for functional continuity. For a highly complex, and multi-modal design space, the GA provides a rapid search in the direction of the global optimum. The requirement of low power and low weight for the problem examined in this paper results in a complex design space that eludes simple description with respect to its modal nature, continuity, or convexity. Hence, this problem of designing a light-weight, low-power rotor system provides a good test for a genetic algorithm based-design code, and finding a design solution displays the ability to use GA in complex rotorcraft design problems during the conceptual phase.

Genetic algorithms search through the design space using a population of points, rather than proceeding from point to point. The evaluated population, especially in early generations, contains a diverse selection of possible designs; in essence, several different designs are evaluated in parallel. Several studies of engineering designers suggest that when presented with a design problem, the designers tend to work using a single-concept design strategy, continually re-working one design until either the design requirements are satisfied or the concept proves futile⁸. The ability to evaluate several different designs at once may allow consideration of designs that would otherwise never have been investigated.

As with most design tools, genetic algorithms have some disadvantages. The relative newness of the approach means that several problems associated with the theories of population size and control parameter values have yet to be resolved. Researchers examining genetic algorithms have undertaken several studies to determine appropriate control parameters^{9, 10}. Unfortunately for the users of GA, the findings have occasionally been contradictory at worst and confusing at best. Much of the work is empirical, suggesting that appropriate control parameters may be problem dependent. This requires experimentation on the part of the user, and in this paper some rules-of-thumb were

followed regarding population size, crossover probability and mutation probability. Other values for these parameters may provide better results and shorter run time than those used in this exercise.

Genetic algorithms search using a guided probabilistic search rather than gradient methods. Because of this, GA do not complete the search at local optima and can solve problems that cannot be attempted using calculus-based methods. However, since no gradient information is associated with the results, genetic algorithms have no guarantee of convergence. In this sense, the ability to guarantee an optimal point is traded for the ability to search the large, complex domain of the design problem.

The genetic algorithm also can encounter *premature convergence*⁴, a condition where the population of strings has become uniform in character, but is lacking near-optimal individuals. In this condition, the best individual no longer improves in fitness from one generation to the next. This can occur as either the result of the problem statement being difficult for the GA to approach (GA-hard)¹¹, or as the result of sampling error, especially in the initial randomly generated population. The concept of premature convergence causes different design results to be generated for different samples of the design space. One author has stated, "Traditional genetic algorithms, though robust, are generally not the most successful optimization algorithms on any particular domain."¹² Figure 2 presents a graphical representation of this concept. On a given, narrow problem domain, calculus-based methods are highly effective; however, when the problem domain is expanded, the genetic algorithm shows better performance. In the work to be discussed in this paper, the addition of discrete and integer variables to the design problem expands the domain typically associated with rotor design optimization problems.

In spite of the difficulties associated with genetic algorithms, the abilities of the code to be used in a conceptual design tool can still produce meaningful results. The addition of the discrete variables representing airfoil sections into the design problem has increased the domain of the problem from that typically attempted by traditional numerical optimization schemes. The effectiveness of the genetic algorithm as an optimizer may be less than that of a calculus-based method, but considered as a search technique, the genetic algorithm is quite efficient. The genetic algorithm narrows the vast design space to more interesting areas, which is the goal of conceptual design.

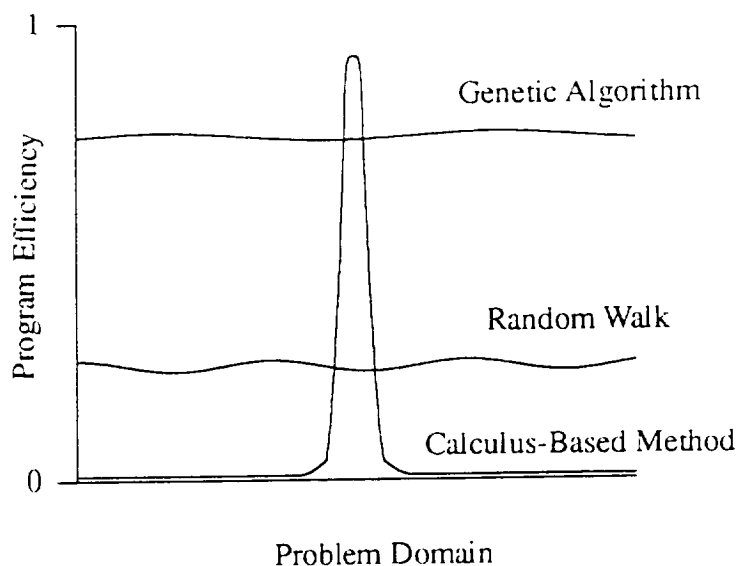


Figure 2. Program Efficiency versus Problem Domain

The GA Rotor Design Code

The rotor system design problem addressed in this research can be described in the manner of an optimization problem. Using a set of design variables, an objective function is to be minimized subject to certain constraints. Design variables used in this exercise are airfoil section, number of blades, solidity, linear twist, tip speed, disk loading, and taper ratio. A global function, combining the weight of the rotor system and the power required of the

rotor system, forms the multi-objective goal function. To keep the scope of the problem reasonably simple, power is minimized for the hover condition. Weight of the rotor system consists of the structural weight of the rotor blades, hub and hinge, drive system and flight controls based on statistical relations.

This GA-based rotor design code contains the three basic operators of genetic algorithms: selection, crossover and mutation. However, as in other GA applications, variations of these operators and "higher-order" operators alter the performance of the GA to make it better suited to the problem at hand.

To use the genetic algorithm, the design variables are coded in binary strings. For example, in this exercise, the airfoil section variable represents eight discrete airfoil sections and is coded into a three-bit binary string. The integer number of blades follows similar coding. Table 1 presents the coding of the discrete and integer design variables to the appropriate binary strings.

Table 1. Discrete and Integer Design Variable-to-Binary String Mapping

Airfoil Section	Binary String	Number of Blades	Binary String
NACA 0012	000	2	000
NACA 0015	001	3	001
NACA 23012	010	4	010
NACA 23015	011	5	011
NACA 63A012	100	6	100
NACA 63A015	101	7	101
Vertol VR-7	110	8	110
Vertol VR-8	111	9	111

To keep the problem small in scope, all of the continuous design variables used were mapped to binary strings of fairly short length. Table 2 contains a summary of the continuous variable-to-string mappings. The continuous design variables in this case have

large values of resolution. To make the variables more "continuous" longer binary strings could easily be used, thereby improving the resolution. For example, improving the resolution of the linear twist variable to 0.5 degrees for the same range of values requires a five bit string. If the problem warranted a resolution of about 0.001 degrees for linear twist, a string of 14 bits could be used. This experiment used high tolerance values for design variables; the resulting short string length makes the code faster to run and easier to debug, and requires less computer memory for storage of the strings and chromosomes. This results from a trade-off of short string length versus higher accuracy. The authors feel that the using a relatively short string length successfully demonstrates the capability of GA in this conceptual design exercise. With this variable-to-string mapping, the binary string representing one design point has a total length of 27 bits. Even with this seemingly coarse resolution, the mapping translates to a design space of 2^{27} or over 134 million different designs. In actuality, the entire design space and variables are discretized, but with appropriate string lengths, variables can be made continuous to within some tolerance.

Table 2. Continuous Design Variable to Binary String Mapping Parameters

Variable (units)	Minimum Value	Maximum Value	Resolution	String Length
Solidity	0.050	0.125	0.005	4 bits
Linear Twist (degrees)	-16	-1	1	4 bits
Tip Speed (ft/sec)	625	780	5	5 bits
Disk Loading (lb/ft ²)	3.0	18.5	0.5	5 bits
Blade Taper Ratio	0.3	1.0	0.1	3 bits

The chromosome of an individual design is built up from the strings relating to the design parameters. An example design for this problem using a NACA 0015 airfoil (represented by the string "001" as the first three bits of the chromosome), 5 blades (011), a solidity of 0.065 (0011), linear twist of -9 degrees (0111), tip speed of 680 ft/sec (01011), disk loading

of 5.5 lb/ft² (00101), and a taper ratio of 0.8 (101) will be represented by the 27-bit concatenated string:

"001011001101110101100101101"

For the multi-objective function in the rotor system design problem, a single global objective function with a quadratic penalty function provides the means to evaluate the fitness of the rotor system designs for minimum power and weight.

$$f = A_1 \left(\frac{p - p_{\min}}{p_{\min}} \right)^2 + A_2 \left(\frac{w - w_{\min}}{w_{\min}} \right)^2 + \sum_i C_i (\max[0, g_i(x)])^2$$

To set up the global function, the genetic algorithm was first run to find the minimum, unconstrained power solution for the rotor system. This provides the p_{\min} value in the following equation. Next, the unconstrained minimum weight solution was found, resulting in the w_{\min} value. Because these solutions are found without constraints, these rotor designs may not be practical designs; they only provide a scaling factor for the global function. Coefficients were chosen for A_1 , A_2 and C_i based mostly on observation. It was intended that the power and weight portions of the equation contribute equal amounts to the fitness value. Coefficients for the penalty function were also chosen to reflect the acceptability of a slightly violated constraint; a larger constant results in a higher penalty for a slight violation.

The single global objective function method to evaluate fitness was chosen for its simplicity in developing the GA code. Several researchers have found that Pareto optimality criteria

using vector-evaluated¹³ or niched¹⁴ genetic algorithms have been quite successful for multiobjective optimization, but these approaches require some difficult modifications to the genetic algorithm code.

A "tournament" selection method is used to choose the appropriate individuals to contribute to the next generation. In this selection approach, two individuals are selected without replacement from the current population. These individuals are evaluated for fitness, and the individual with the better fitness of this pair survives for the crossover step. A second pair is picked, evaluated and one individual is selected and mated with the individual from the previous pair. This continues until the new population is filled. Figure 3 provides a flow chart describing the tournament selection process. This method possesses several advantages over the more traditional roulette wheel or rank order selection methods⁴. As compared with other section methods which rely on the maximum numerical value for fitness to be considered the best fitness, the tournament selection method allows for a minimum fitness value to be considered the best fitness. This more naturally follows the intent of minimizing power and weight. Additionally, this tournament selection directly compares two individuals at a time, rather than comparing the relative fitness of one individual against the entire population; this avoids problems with fitness scaling. Unfortunately, even with fitness scaling problems avoided, the genetic algorithm can still encounter premature convergence. To help avoid this, an elitist tournament selection was used, in which the best individual from the current generation is kept unchanged and replaces the worst member of the next generation.

With the elitist tournament selection method helping to stall premature convergence, the GA still encounters situations where the current population cannot improve the best individual. When the algorithm begins to show signs of premature convergence in this

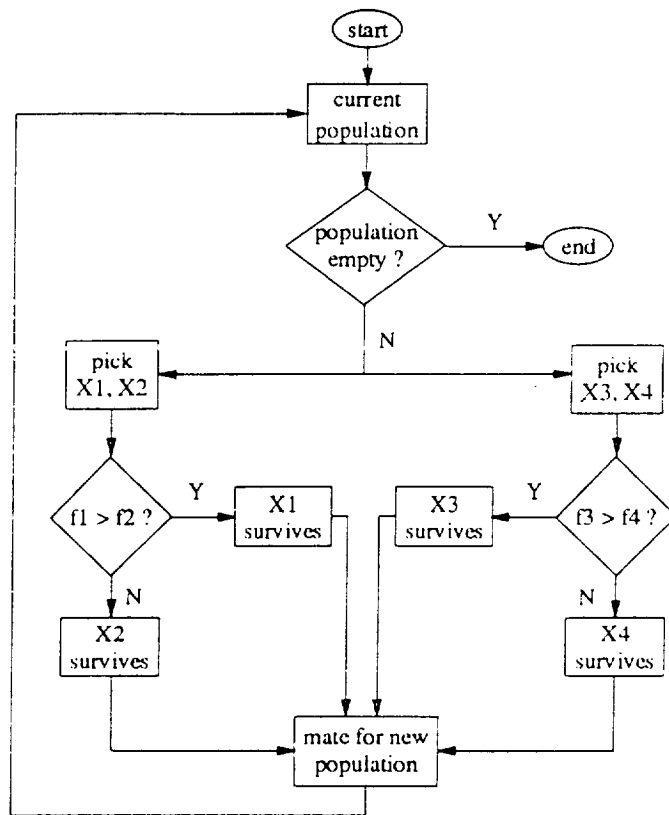


Figure 3. Tournament Selection Process

work, e.g., no improvement in the best individual after five generations, a G-bit improvement⁴ scheme is used. The G-bit improvement is a gradient-like bitwise improvement approach. For the current best individual in the population, the G-bit improvement routine varies one bit at a time in the chromosome. The single bit change producing the best fitness string is saved, and entered into the population, replacing the current generation's poorest member. This G-bit improvement forces a mutation upon the best individual to try to introduce new schema to the population that are needed to improve the best individual.

Attempts to avoid premature convergence also act to lessen the effectiveness of the global search properties of the genetic algorithm. To assist the global search, the population size

is increased to 100 members even though the string size would suggest that a smaller population would be adequate⁴. For the increased population size, the random sample of the search space taken by the initial population is more diverse, and therefore, more likely to contain high quality schema that will lead to a good solution. Based on the population size of 100 individuals, the crossover probability was set at 70%, meaning that each pair of strings selected for mating have a 70% chance of exchanging information and a 30% chance of the offspring being "clones" of the parents, and the mutation probability was set at 0.1%, meaning that a bit in the string has a 0.1% chance of being changed from its current value of "1" or "0" to the opposite value.

FITNESS ANALYSIS METHODOLOGY

For this study of genetic algorithms, the best rotor system combines a minimum amount of required power and a minimum system weight. Because genetic algorithms require a fairly large number of fitness evaluations during their operation, simple analysis techniques with a reasonable amount of accuracy and low computational cost provide the means for evaluating the rotor systems.

Analysis Methods

To determine the power required in hover, a combined blade element/momentum code, as suggested by Prouty¹⁵, calculates the power needed for a specified thrust. The lift curve slope, zero-lift angle of attack and a drag coefficient look-up table relating drag coefficient to angle of attack provide aerodynamic characteristics of the airfoil section. All airfoil data for this work was taken from Dadone¹⁶ to provide a single-source reference in order to maintain consistency. Empirical weight equations, as derived by Shinn¹⁷, provide a method for evaluating rotor system weight. Although a rigorous structural analysis of the rotor

system would furnish a more accurate weight estimate, this statistical, parametric analysis was utilized in order to keep computation time as low as possible.

A penalty method approach enforces constraints on the design space which keep the rotor designs realistic and feasible. One such constraint, that forbidding any stalled rotor sections in hover, is handled in this way. Because of the fairly simple representation of the airfoil section aerodynamic properties in the blade element method, a solution which provides enough thrust for hover may require angles of attack that exceed the airfoil's stall angle. If an angle is required which exceeds that for stall, the fitness function is penalized. For forward flight, the advancing blade at 90 degrees azimuth position experiences the highest local velocities. If any elements on this blade encounter a local Mach number which exceeds the drag divergence Mach number of the airfoil, a penalty is added to the fitness function for each element that encounters this condition.

Because the weight analysis for the rotor systems is based on a statistical model of existing rotor systems, the weight equations have difficulty when required to predict the weight of infeasible rotor designs, similar to the problems in existing design methods. In particular, the blades for a rotor system being evaluated may have an aspect ratio that is physically improbable, but the weight equations calculate a weight for this blade without regard to its improbability. Concerns about structural and dynamic response of the blades are included in a constraint on the blade aspect ratio to account for this problem. The aspect ratios of blades for 26 different helicopters were used to find a mean and standard deviation for the aspect ratio of a "typical" helicopter blade with the assumption that the blade aspect ratios follow a normal distribution. The constraint is imposed if the blade design has an aspect ratio which exceeds 23.1, the mean value plus two standard deviations ($\mu + 2 \sigma$) from the

database. This limit represents 98% of the rotor blades actually built and flying, implying that these blades meet structural and dynamic requirements.

A maneuver capability for both steady (no loss in speed or altitude) and transient (with change in speed and/or altitude) turns must be maintained. The steady turn analysis evaluates the retreating blade at an azimuth of 270 degrees. The portion of the blade outboard of the reverse flow region must produce thrust equal to the total thrust of the rotor system divided by the number of blades. If the blade cannot meet this requirement, a penalty is assessed. Because the transient maneuver involves dynamic blade stall and other complex issues, the analysis makes use of an empirical curve relating advance ratio to blade loading. At a given advance ratio and required rotor thrust, this curve provides a maximum available blade loading for comparison with the required blade loading. If the required blade loading exceeds the maximum available blade loading, the fitness is penalized.

Validation of Fitness Analysis

In conceptual design, between 10 and 15% analysis error is generally considered acceptable¹⁸. As the design process progresses, the allowable error decreases, and more sophisticated analysis techniques are required. For the application considered here, analyses with 10 to 15% error are appropriate. As discussed previously, reduced computational time motivated the methods used to calculate the fitness of a rotor system. Evaluating an existing rotor with the fitness evaluation routines and comparing known characteristics to calculated characteristics checks the validity of these methods. To provide this comparison, the light, single turbine MD 500E helicopter was used as the baseline rotor system.

The routines used to calculate the fitness value were decoupled from the genetic algorithm to calculate a fitness for the MD 500 rotor system. This rotor has simple planform

geometry and uses the NACA 0015 airfoil section. A description of the rotor can be found in the appendices of Reference 15. This aircraft weighs 3000 lb, and is evaluated in hover for sea-level, standard atmospheric conditions. Using the blade element code, the predicted power to hover is 286 hp, and the weight of the rotor system (blades, hub and hinge, drive system, and flight controls) is 373 lb. This predicted power differs by less than 5% from flight test data for hover out-of-ground effect. The predicted weight differs by less than 3%. These levels of error are more than acceptable for the conceptual phase of design.

Because the MD 500E was chosen as the baseline to evaluate the fitness methodology, it is important to note that the rotor system of the MD 500E violates the aspect ratio constraint assigned to the design problem. The 500E blade aspect ratio of 23.4 just exceeds the limit imposed of 23.1. Because the MD 500 violates this but is actually a successfully flying rotor system, a "weak" constant is multiplied to the penalty to allow blades like the 500's, which barely violate this constraint, to be acceptable. Using the fitness evaluation for the actual MD 500E rotor, the fitness value is 0.4872, the penalty for exceeding the aspect ratio constraint contributes only 0.0038 to this overall fitness. No other constraints are violated by the MD 500 rotor.

"DESIGN" OF AN EXISTING ROTOR SYSTEM

The MD 500E's design requirements were used to generate a rotor system using the GA-based code. The rotor system is to provide a combination of minimum power to hover and minimum system weight for an aircraft with a 3000 lb gross weight, 5.5 ft² of flat plate drag area, and an assumed hover download of 5%. The aircraft must be capable of a maximum forward speed of 135 knots, and must perform a steady maneuver (no change in altitude or flight speed) of 1.1 g at 70 knots and perform a transient maneuver (allowing change in altitude and/or flight speed) of 1.5 g initiated at 120 knots.

To solve this design problem, the genetic algorithm-based code generated nine designs, each time using a different initial set of 100 randomly selected design points. Each of the nine different runs produces a slightly different solution. A summary of the results forms Table 3, which also includes the MD 500E rotor system for comparison. The "GA seed" is simply a starting point for the initial random selection process.

Table 3. Genetic Algorithm Based Design of the MD 500E Rotor System

GA seed	Airfoil Section	No. of Blades	Solidity	Linear Twist (deg)	Tip Speed (ft/s)	Disk Loading (lb/ft ²)	Taper Ratio
MD 500E	NACA 0015	5	0.068	-9	680	5.48	1.0
9	Vertol VR-8	5	0.070	-6	690	5.0	0.5
9353	Vertol VR-8	5	0.075	-7	700	5.5	0.5
11939	Vertol VR-8	4	0.055	-6	685	5.0	0.9
5139	Vertol VR-7	4	0.055	-6	670	5.0	0.5
12934	Vertol VR-8	4	0.060	-6	705	5.0	0.6
566	Vertol VR-7	3	0.050	-9	665	5.0	0.7
4465	Vertol VR-8	5	0.070	-8	645	5.5	1.0
10270	Vertol VR-8	4	0.055	-6	730	4.5	0.5
12333	Vertol VR-7	5	0.070	-6	630	5.5	0.5

All of these rotor system designs are different from each other and from the actual MD 500E. It is important to see that the GA-generated rotor designs are actually fairly close together in the available design space, and that all have reasonable parameters. Because of the known effects of premature convergence and the lack of a guarantee of convergence at an optimum, these differences should be expected.

To display the search process that the genetic algorithm uses, Figure 4 presents the planform for several interim designs in the search using the initial seed value of 9. The MD 500E rotor system planform is included for comparison. The rotor system for generation 50 is the final design listed in the above table; all runs were continued for 50 generations, even though several runs had reached a convergent-like point before the 50th generation

was reached. Most of the major changes to the rotor system occur in the first generations as the search space is rapidly narrowed.

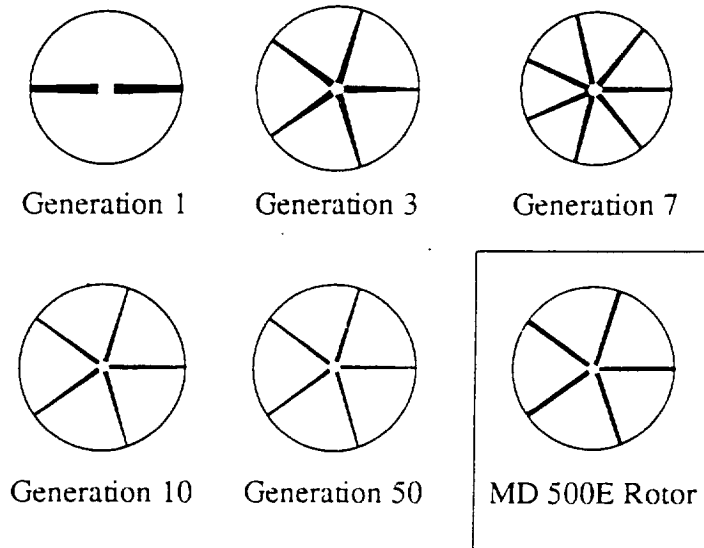


Figure 4. Planform Evolution of the Existing Rotor Design Problem

Results for Existing Rotor System Design

The objective values for the rotors produced by the GA-based code are compared with the actual 500E rotor system in Table 4. Fitness values for all of the rotor systems are shown along with the power and weight requirements. All of the new rotors have a lower fitness value than the existing rotor system, suggesting that they are better designs according to the criteria given in this problem. The rotors generated by the genetic algorithm all have lower power requirements than the actual MD 500E rotor. The rotor system weights are generally fairly close to the current rotor weight; some are slightly higher, others are slightly lower. As discussed previously, the MD 500E rotor blades violate the aspect ratio constraint, but this is only "slightly" violated and has a small effect on the fitness value. Other rotors with violated constraints have similar situations, and the effect of the constraint violation on the fitness value is very small, as seen in the last column of Table 4.

Table 4. Rotor System Comparison

GA seed	Power Required (hp)	System Weight (lb.)	Fitness Value	Violated Constraints (fitness contribution)
MD 500E	286	373	0.4872	aspect ratio (0.0038)
9	249	397	0.3388	none
9353	260	383	0.3682	steady turn (0.0036)
11939	255	356	0.2987	none
5139	252	355	0.2816	none
12934	252	373	0.3112	steady turn (0.0013)
566	256	350	0.2925	steady turn (0.0005)
4465	265	370	0.3630	none
10270	238	384	0.2835	Mach number, steady turn (0.0038)
12333	262	370	0.3469	none

The use of different design parameters accounts for some of the differences between the generated designs and the existing design. Allowing the Vertol VR-7 and VR-8 airfoils, which are specifically designed for low power, to be used as design variables leads to lower power designs. Similarly, taper ratio was probably constrained for manufacture in the MD 500E. Because of the rather complex nature of the rotor design problem, the genetic algorithm did reach some prematurely convergent conditions, resulting in the nine different blade designs. All of the resulting designs can be considered good designs, especially when compared with the existing rotor system. Obviously, some of the generated designs are better than others, and because of the lack of a guarantee on optimal convergence, this condition is comparable to (though not the same as) a calculus based method finding a local minimum. The important difference here is that the genetic algorithm does not find a *local* optimum; rather it progresses toward the *global* optimum until a convergent-like state is reached. This behavior supports the claim that these generated rotor systems are good designs, but does not allow the claim that any of these is the actual globally-optimal solution.

In this exercise the nature of the fitness function is rather important in affecting the solutions generated. The fitness function used in this problem placed a higher emphasis on reducing the power required to hover than on reducing rotor system weight. The effect of this is seen in the resulting designs always having a lower power requirement than the existing rotor system, but also having, in some instances, a higher weight than the existing system.

DESIGN OF A NEW ROTOR SYSTEM

To mimic one of the steps taken in the design of a new rotorcraft, the GA code was used to generate a rotor system with no existing comparison point. For this part of the exercise the rotor system for a notional 20,000-lb attack helicopter was designed. The aircraft was assumed to have 45.0 ft² of equivalent flat plate drag area and a hover download of 5% of the gross weight. Performance constraints imposed on the rotor included a maximum cruise speed of 185 knots, the ability to perform a steady turn of 1.25g at 70 knots (to approximate minimum power velocity), and the ability to perform a 2.0g transient turn initiated at 165 knots. Constraints were also imposed on the blade stall in hover and blade aspect ratio. Because this is meant to be an attack helicopter, a ballistic tolerance constraint was imposed; the mean aerodynamic chord of the blade must be greater than or equal to 18 in.

The GA-based code created nine different rotor designs using nine different initial random seeds. Table 5 displays the resulting designs. As with the MD 500E rotor design exercise, the different designs are in the same neighborhood of the design space. All rotors have between five and seven blades, and most have the Vertol VR-8 airfoil section. Other design variables have high levels of similarity.

Table 5. Genetic Algorithm-Based Design of the Attack Helicopter Rotor System

GA Seed	Airfoil Section	No. of Blades	Solidity	Linear Twist (deg)	Tip Speed (ft/s)	Disk Loading (lb/ft ²)	Taper Ratio
3406	Vertol VR-8	6	0.090	-8	655	5.0	0.7
4577	Vertol VR-8	6	0.085	-7	665	5.0	0.7
9330	Vertol VR-8	7	0.100	-6	665	5.0	0.5
14172	Vertol VR-8	6	0.085	-6	635	4.5	0.7
2612	NACA 63A012	6	0.090	-10	655	5.0	0.6
3677	Vertol VR-8	5	0.070	-8	660	4.0	0.8
706	Vertol VR-8	5	0.080	-4	640	4.5	0.7
12510	Vertol VR-8	5	0.070	-4	645	4.0	0.7
6813	Vertol VR-8	5	0.080	-8	670	5.0	0.8

Figure 5 displays the planforms for several intermediate designs found during the run using the seed value 3677. As with the existing design problem, most major changes are made in the first few generations of the design run.

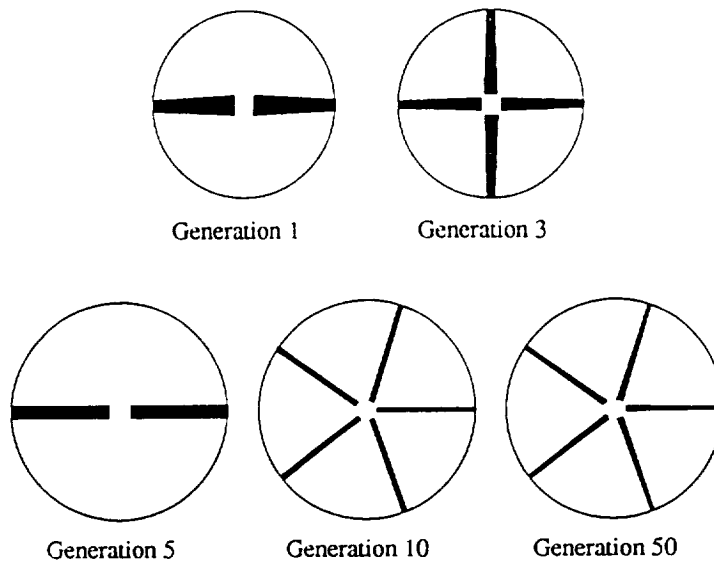


Figure 5. Planform Evolution of the New Rotor System Design Problem

Results for New Rotor System Design

The fitness values, power required and system weights for the attack helicopter rotor designs are compared in Table 6. While the fitness values for all the designs are around 0.35, the power and weight values actually differ from each other. Because a multi-objective global function was used to evaluate fitness, a low-power rotor with a higher system weight will have the same fitness as a higher-power, low-weight rotor system. In this particular exercise, none of the resulting rotors violated any of the constraints imposed upon the design space.

Table 6. Attack Helicopter Rotor System Comparison

GA seed	Power Required (hp)	System Weight (lb)	Fitness Value	Violated Constraints
3406	1830	4718	0.3586	none
4577	1835	4561	0.3372	none
9330	1815	4910	0.3824	none
14172	1740	4995	0.3601	none
2612	1879	4761	0.3934	none
3677	1666	5013	0.3298	none
706	1760	4943	0.3604	none
12510	1659	5033	0.3344	none
6813	1847	4509	0.3362	none

The difference in the designs can be attributed to the occurrence of premature convergence. Again, if the genetic algorithm is viewed strictly as an optimization technique, this premature convergence is undesirable, but viewed as a high-speed global search technique, these designs are all reasonable. The search of the genetic algorithms approaches the global optimum, so all of these designs may be considered good but not optimal designs.

FURTHER APPLICATIONS AND RESEARCH

This paper has demonstrated the potential for genetic algorithms in conceptual design of rotor systems. The work has, however, generated many questions about genetic algorithms as well as having demonstrated possibilities for GA application in conceptual design. There are several areas where this work could be elaborated, the objectives modified, and the application of GA extended to larger design problems.

Several areas exist for extensions of this work. Multiobjective optimization supplies a significant way to improve designs. In the context of genetic algorithms, the multiobjective function of low power and low weight in this discussion could be evaluated using different global functions from that used here. The tournament selection could be made a multiobjective tournament by having each branch represent each criteria. Other researchers have applied Pareto optimality concepts to the genetic algorithm; concepts of niching and sharing have also been investigated. Relative performance and results of these methods should be evaluated and compared for using multicriteria fitness in the genetic algorithm for conceptual design.

Other objective functions for the rotor system could also be included in the conceptual design process. Low noise requirements, minimum power at several flight conditions, minimum control load requirements, reduced vibration and shaft load transfer could all be used as objective functions, either alone or combined in some multiobjective approach. Typically, many of these features of rotor systems are addressed by rules-of-thumb early in the design process, or are not even addressed until later in the process when changes to the design become more difficult and costly.

Because of the simplicity of the demonstration presented in this paper, some accuracy in analysis was knowingly sacrificed. A more rigorous structural analysis would increase the confidence in accounting for the rotor structural and dynamic response. Weight prediction using a volume integration approach may also reduce uncertainty associated with using parametric weight equations.

The two demonstrations of rotor system design in this paper exhibited occurrence of premature convergence, where the genetic algorithm reached a point that could not be improved upon, but is not necessarily the global optimum. As suggested by Goldberg⁴, as well as others, a hybrid method for rotor system design could be employed. In one scenario for a hybrid method, the genetic algorithm first generates one or more conceptual rotor system designs. Then using more rigorous numerical optimization techniques and more accurate analysis methods, the preliminary design of the rotor could be developed using the genetic algorithm result as the starting point. Typically, these numerical optimizations are fairly costly. Reference 19 reports CPU times as high as 18 hr on a DEC VAX system, using a known rotor system as the initial design point. The genetic algorithm based code used in this paper takes about 10 min to run on an PC machine with a 486 processor. Because the numerical optimization routines, such as CONMIN, produce results that can vary widely with the initial design point, it seems that for a new rotor system, where no physical example exists, the GA-based code could provide quality starting points for the more rigorous design optimization procedures. This would reduce computational time which may be required when a poor initial design point is chosen.

Finally, it appears that using genetic algorithms can be applied to aircraft conceptual design. In this example, only one discrete variable was used in the problem statement. Using configuration selections as discrete variables, the GA-based code can search a still wider

design space than examined in this paper. These discrete variables could include rotor configurations, as discussed earlier in the paper, conceivably the same could be done for rotorcraft configurations; one variable could represent a helicopter, a tilt-rotor, a tilt-wing, a stopped rotor, or a lift jet, for example. Incorporating engine selection as a discrete variable could allow a designer to choose from "off-the-shelf" powerplants, and use an integer design variable to choose the number of the powerplants. It appears that there is great potential for genetic algorithms in the conceptual design of entire aircraft, not just rotor systems.

CONCLUSIONS

The development of a design code that incorporates genetic algorithms has potential capability to perform both selection and optimization tasks in the conceptual design of rotor systems. The work in this paper demonstrates the feasibility of incorporating GA into the conceptual design process of a rotor system. Additionally, the capability of the code to evaluate the multiobjective function of low power and low weight is demonstrated. With this potential demonstrated, the conceptual design process for entire rotorcraft may be greatly enhanced in both speed and scope with the application of genetic algorithms to this process.

ACKNOWLEDGMENTS

Mr. Crossley would like to thank McDonnell Douglas Helicopter Systems for the support of his work through a graduate education fellowship with Arizona State University. The authors would also like to acknowledge Mr. Tony Glinka for his contributions to the code. This work was partially sponsored by NASA Ames Research Center Grant GR NAG 2882.

REFERENCES

1. Holland, J. (1992) Genetic Algorithms. *Scientific American*, July, pp. 66-72.
2. KrishnaKumar, K. (1993) Genetic Algorithms - A Robust Optimization Tool. *AIAA 93-0315*.
3. Holland, J. H. (1992) *Adaptation in Natural and Artificial Systems*. MIT Press.
4. Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.
5. Goldberg, D. E., Milman, K., and Tidd, C. (1992) Genetic Algorithms: A Bibliography. IlliGAL Report No. 92008, Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois at Urbana-Champaign, 1992.
6. Lin, C.-Y., and Hajela, P. (1992) Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables. *Engineering Optimization*, (19), pp. 309-327.
7. Adelman, H. M., and Mantay, W. R. (1991) Integrated Multidisciplinary Design Optimization of Rotorcraft. *Journal of Aircraft*, (28), January, pp. 22-28.
8. Finger, S. and Dixon, J. R. (1989) A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes. *Research In Engineering Design*, (1), 1, pp. 51-67.
9. Schaffer, J. D., Caruana, R. A., Eshelman, L. J., and Das, R. (1989) A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June 4-7, Morgan Kaufmann Publishers, Inc., San Mateo, California.
10. Goldberg, D. E. (1992) *Optimal Initial Population Size for Binary-Coded Genetic Algorithms*, (TCGA Report No. 85001), The Clearinghouse for Genetic Algorithms, University of Alabama, Tuscaloosa, Alabama, 1985.s, MIT Press.
11. Goldberg, D. E. and Segrest, P. (1993) Finite Markov Chain Analysis of Genetic Algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, Massachusetts Institute of Technology, July 28-31.
12. *Handbook of Genetic Algorithms*, Davis, L. (editor), Van Nostrand Reinhold, 1991.

13. Schaffer, J. D. (1985) Multiple Objective Function Optimization with Vector Evaluated Genetic Algorithms. *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, Carnegie-Mellon University, July 24-26.
14. Horn, J. and Nafpliotis, N. (1993) Multiobjective Optimization Using The Niche Pareto Genetic Algorithm. IlliGAL Report No. 93005, Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois at Urbana-Champaign.
15. Prouty, R. W. (1990) *Helicopter Performance, Stability and Control*, Reprint Edition. Robert E. Krieger Publishing Company.
16. Dadone, L. U. (1976) *U.S. Army Helicopter Design Datcom, Volume I - Airfoils*, (Report Number USAAMRDL CR 76-2), U.S. Army Air Mobility Research and Development Laboratory, Moffett Field, California, May.
17. Shinn, R. A. (1981) Group Weight Estimation for the Advanced Scout Helicopter Design Study. SAWE Paper 1445, 40th Annual Conference of Mass Properties Engineering, Dayton, Ohio, May 4-7.
18. McMasters, J. H. (1985) Some Thoughts on Design Education. *AIAA 85-3079*, AIAA/AHS/ASME Aircraft Design, Systems and Operations Meeting, Colorado Springs, Colorado, October 14-16.
19. Callahan, C. B. and Straub, F. K. (1993) Design Optimization of Rotor Blades for Improved Performance and Vibration. *Journal of the American Helicopter Society*, (38), 4, pp. 62-70.