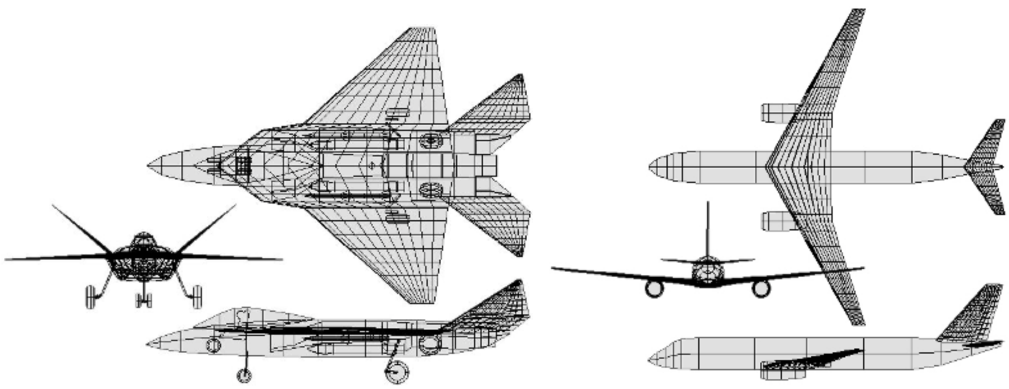
Summary of:

ENHANCING AIRCRAFT CONCEPTUAL DESIGN USING MULTIDISCIPLINARY OPTIMIZATION



By
Daniel P. Raymer

Doctoral Thesis
Report 2002-2, May 2002
ISBN 91-7283-259-2

Department of Aeronautics
Royal Institute of Technology
SE-100 44 Stockholm
Sweden

ABSTRACT

Research into the improvement of the Aircraft Conceptual Design process by the application of Multidisciplinary Optimization (MDO) is presented. Aircraft conceptual design analysis codes were incorporated into a variety of optimization methods including Orthogonal Steepest Descent (full-factorial stepping search), Monte Carlo, a mutation-based Evolutionary Algorithm, and three variants of the Genetic Algorithm with numerous options. These were compared in the optimization of four notional aircraft concepts, namely an advanced multirole export fighter, a commercial airliner, a flying-wing UAV, and a general aviation twin of novel asymmetric configuration. To better stress the methods, the commercial airliner design was deliberately modified for certain case runs to reflect a very poor initial choice of design parameters including wing loading, sweep, and aspect ratio.

MDO methods were evaluated in terms of their ability to find the optimal aircraft, as well as total execution time, convergence history, tendencies to get caught in a local optimum, sensitivity to the actual problem posed, and overall ease of programming and operation. In all, more than a million parametric variations of these aircraft designs were defined and analyzed in the course of this research.

Following this assessment of the optimization methods, they were used to study the issue of how the computer optimization routine modifies the aircraft geometric inputs to the analysis modules as the design is parametrically changed. Since this will ultimately drive the final result obtained, this subject deserves serious attention. To investigate this subject, procedures for automated redesign which are suitable for aircraft conceptual design MDO were postulated, programmed, and evaluated as to their impact on optimization results for the sample aircraft and on the realism of the computer-defined "optimum" aircraft. (These are sometimes called vehicle scaling laws, but should not be confused with aircraft sizing, also called scaling in some circles.)

This study produced several key results with application to both Aircraft Conceptual Design and Multidisciplinary Optimization, namely:

- MDO techniques truly can improve the weight and cost of an aircraft design concept in the conceptual design phase. This is accomplished by a relatively small "tweaking" of the key design variables, and with no additional downstream costs. *In effect, we get a better airplane for free.*
- For a smaller number of variables (<6-8), a deterministic searching method (here represented by the full-factorial Orthogonal Steepest Descent) provides a slightly better final result with about the same number of case evaluations
- For more variables, evolutionary/genetic methods get close to the best final result with far-fewer case evaluations. The eight variables studied herein probably represent the practical upper limit on deterministic searching methods with today's computer speeds.

- Of the evolutionary methods studied herein, the Breeder Pool approach (which was devised during this research and appears to be new) seems to provide convergence in the fewest number of case evaluations, and yields results very close to the deterministic best result. However, all of the methods studied produced similar results and any of them is a suitable candidate for use.
- Hybrid methods, with a stochastic initial optimization followed by a deterministic final "fine tuning", proved less desirable than anticipated.
- Not a single case was observed, in over a hundred case runs totaling over a million parametric design evaluations, of a method returning a local rather than global optimum. Even the modified commercial airliner, with poorly selected initial design variables far away from the global solution, was easily "fixed" by all the MDO methods studied.
- The postulated set of automated redesign procedures and geometric constraints provide a more-realistic final result, preventing attainment of an unrealistic "better" final result. Especially useful is a new approach defined herein, *Net Design Volume*, which can prevent unrealistically high design densities with relatively little setup and computational overhead. Further work in this area is suggested, especially in the unexplored area of automated redesign procedures for discrete variables.

1 INTRODUCTION

1.1 Overview

Aircraft designers have always tried to make their newest design the "best ever", and have eagerly used the latest tools at their disposal to determine the combination of design features and characteristics that will produce that "best." The Wright Brothers performed parametric wind tunnel trade studies of wing aspect ratio and camber, and part of their eventual success was due to this early form of optimization[1]. Subsequent generations of aircraft designers have learned how to make "carpet plots" for two-variable optimizations, and have laboriously extended that to a dozen or so variables by repetition and cross-plot. When electronic computers became available, aircraft designers gladly accepted their help in the repetitive calculations required for aircraft design optimization (see Ashley[2] for a definitive survey of aerospace optimization as of the late 1970's).

Today improved techniques for the optimization of complicated engineering problems are emerging from universities and research laboratories. These are being applied to the aircraft design process as soon as the designers perceive that the methods have become mature and practical enough to help to find a better "best", in a reasonable amount of time.

These new techniques usually go by the generic title "Multidisciplinary Optimization", or MDO. They are suitable for optimization of entire systems including aircraft vehicle configurations. As defined by a leading MDO researcher and proponent, MDO is "A methodology for design of complex engineering systems that are governed by mutually interacting physical phenomena and made up of distinct interacting subsystems (suitable for systems for which) in their design, everything influences everything else" (Sobieski[3]).

MDO permits optimization of a number of design variables affecting disparate functional disciplines, using system-level measures of merit, and in the presence of multiple system design constraints. As applied to aircraft, this should result in reduced acquisition and operating costs and/or better system performance..

There are a wide variety of MDO methods in development, and the current debate is quite heated as to which ones are best for various applications. Even within a general form of MDO, different researchers prefer different combinations of specific features. It remains difficult or impossible to draw general conclusions from the literature as to which methods one should use for a particular application. One thing seems clear – the "best" MDO method depends on the problem being solved.

MDO methods fall into several categories. Many of them are based on classical mathematical optimization involving definition of governing equations of an objective function, and calculation of derivatives to find the optimum. Other methods do not – they rely solely on calculation of actual values of objective functions and use some form of direct comparisons to find an optimum.

This research concentrates exclusively on the latter types of MDO, called *zeroth-order* or *non-gradient* methods because they do not involve determinations of derivatives or slopes. Such methods permit optimization using existing aircraft analysis software, and furthermore, such methods permit extreme complexity in the aircraft analysis.

The present research focuses solely on the earliest phase of aircraft design, namely conceptual design, in which the broadest design features are being determined such as number and size of engines, wing area and planform shape, and fuselage length and arrangement.

A suite of time-tested, conceptual-level analysis tools is employed in this research, and the MDO methods employed are restricted to those zeroth-order methods that can be implemented with virtually no additional set up beyond the input data already required for aircraft design analysis.

Another issue of importance to the use of MDO for aircraft conceptual design optimization is the actual selection of variables, constraints, and measures of merit. In the literature of aircraft MDO, these key parameters are often selected with little formal consideration, and sometimes bear little resemblance to the design parameters commonly used in industry design offices. An attempt is made herein to address these issues, offering a framework for selection and a suggested suite of variables, constraints, and measures of merit for various types of aircraft.

In addition to a study of which MDO methods seem best for aircraft conceptual design, this research addresses the manner in which the computer routine changes the representation of the aircraft design as a result of changes in the design variables. For example, an increase in fuselage length usually requires an increase in landing gear length to allow the same tail-down angle for takeoff.

For MDO results to have meaning and utility in the real world of aircraft conceptual design, procedures for automated redesign must be defined and employed that approximate to a reasonable degree the changes that an experienced human designer would make to an existing layout were the variables in question revised. Research reported herein addresses this issue with a postulated set of automatic redesign procedures.

1.2 Objectives and Unique Aspects of this Research

Objectives:

- Development and implementation of Aircraft Conceptual Design principles and methods in a PC-based system, incorporating vehicle analysis and system-level multivariable optimization.
- Development and implementation of advanced Multidisciplinary Optimization (MDO) routines including Evolutionary, Genetic, and Monte Carlo algorithms.
- Comparative assessment of optimization methods during aircraft conceptual design.

- Definition and assessment of procedures for geometrical constraints and automated air vehicle redesign to enhance optimization realism.
- Application of methods and optimization techniques to four notional aircraft design concepts, and use of them for comparative studies of MDO methods and options.

Unique Aspects of this Research and Contributions to the Field:

- Development and test of MDO routines based on the design variables, constraints, measures of merit, and analysis methods typically used by aircraft designers in industry.
- Development of a tool permitting study of MDO methodologies using exactly the same aircraft inputs and analysis methods, thus removing those potential sources of "noise" from comparative studies.
- Definition and validation of *Net Design Volume*, a measure of the packaging density of an aircraft design layout and a geometric constraint for MDO routines that avoids unrealistic configurations being defined by the optimizer.
- Definition and study of *Bit-String Affinity*, a measure of MDO convergence that is simple to implement and gives a useful and clear indication of convergence even for MDO routines that do not follow a mathematically pristine convergence rate.
- Definition and study of the *Breeder Pool* Genetic Algorithm, an apparently novel variant of the basic *GA* method.

1.3 Summary of Major Results and Conclusions

Probably the most important conclusion of this study is that the aircraft conceptual design process can be improved by the proper application of Multidisciplinary Optimization methods. Such MDO techniques can reduce the weight and cost of an aircraft design concept in the conceptual design phase by fairly minor changes to the key design variables, and with no additional downstream costs.

In effect, we get a better airplane for free.

These methods are shown to be superior to traditional carpet plots as used in the aircraft conceptual design process for many decades, and can become a normal and integral part of the definition of a new aircraft design.

The realism of MDO methods is shown to improve by the use of the geometric constraints and automated aircraft redesign procedures defined in this research and added to the MDO routine. A new geometric constraint approach defined herein, *Net Design Volume*, proved to credibly adjust the design to ensure sufficient volume for fuel and internal equipment.

Comparisons between the different MDO methods studied found that all of the methods produce reasonable results. For a smaller number of variables the deterministic full-factorial Orthogonal Steepest Descent searching method provides a slightly better final result with about the same number of case evaluations. For more variables, evolutionary/genetic methods get nearly the same final result with fewer case evaluations.

Of the evolutionary methods studied herein, the Breeder Pool approach devised during this research seems to provide convergence on a good solution in the fewest number of case evaluations.

Hybrid methods combining a stochastic initial optimization with a deterministic final optimization proved to work no better than either alone.

2 BACKGROUND (removed – see full thesis)

2.1 Aircraft Design Optimization – Purpose and Importance

2.2 Outline of Aircraft Design Process

2.3 Classical Aircraft Optimization Methods

2.4 Historical Review of Engineering Optimization

2.5 Overview of Multidisciplinary Optimization (MDO)

2.6 The MDO Realism Problem – Automating Aircraft Redesign

2.7 Observations Concerning Variables, Constraints, & MOMs

3 OBJECTIVES AND SCOPE OF RESEARCH

3.1 MDO Methodologies for Aircraft Conceptual Design

The overriding objective of this research is to offer improvements to the aircraft design process through the application of Multidisciplinary Optimization methods. This research focuses exclusively on the conceptual design process where new aircraft concepts are being developed, assessed, and selected for further design effort.

A deliberate decision was made to focus on zeroth-order optimization methods that iterate to a solution based solely on parametric evaluations of the measure of merit and design constraints. Such methods seem most capable of dealing with discontinuous and highly irregular objective and constraint functions, and also appear most suitable for incorporation into existing aircraft analysis codes such as this author's RDS-Professional[4].

A key objective is the ability to make direct comparisons between and among the various MDO methods programmed, using the same sample aircraft and the exact same analysis methods and executable code. In this manner some attempt at identifying a "best" method could be made, at least for the classes of aircraft studied and the optimization variables chosen.

In addition to the step searching method already programmed into RDS, a decision was made to focus on MDO methods in which the parametric variations to the aircraft design are all done with a chromosome-based scheme. This led to the selection of Monte Carlo, Evolutionary, and Genetic Algorithms. These could all be programmed as a related family of methods.

Another objective is the investigation of the relative importance of design variables and constraints common to aircraft conceptual design projects, including performance constraints and geometric constraints such as wingspan. To improve acceptance by practicing aircraft designers, this research is based on the design variables, constraints, measures of merit, and analysis methods typically used in industry.

3.2 Procedures for Automated Aircraft Redesign

This research actually started from a personal interest in this particular topic – how to have a computer program automatically redesign the aircraft during an optimization as design variables are parametrically changed, such that the resulting optimum aircraft is closer to being feasible when a human aircraft designer turns a computational optimum to a real configuration layout. This author previously developed a simple set of such techniques for the RDS-Professional program optimizer, but clearly more work was called for.

Thus, an objective of this research is to define and assess a set of procedures for the automated aircraft redesign that others can incorporate into their MDO routines to enhance optimization realism. Hopefully this can help to make MDO more useful to industry aircraft designers working on real aircraft design projects.

Most important, and apparently original, was the definition and validation of *Net Design Volume*, a measure of the packaging density of an aircraft design layout and a geometric constraint for MDO routines that avoids unrealistic configurations being defined by the optimizer. This addresses the issue of maintaining a realistic internal volume with allowance for fuel, payload, avionics, propulsion, and the numerous smaller subsystems that are properly designed only long after conceptual design.

3.3 Validation Models and Limits of Research

Four notional aircraft design concepts were prepared during this research. These were used as validation models to assess the MDO routines and automated aircraft redesign procedures. These are intended to span the spectrum of current design thought, and include a conventional jet transport, an F-16 replacement export fighter, a tactical unmanned air vehicle, and an asymmetric general aviation twin, as shown in figure 1.

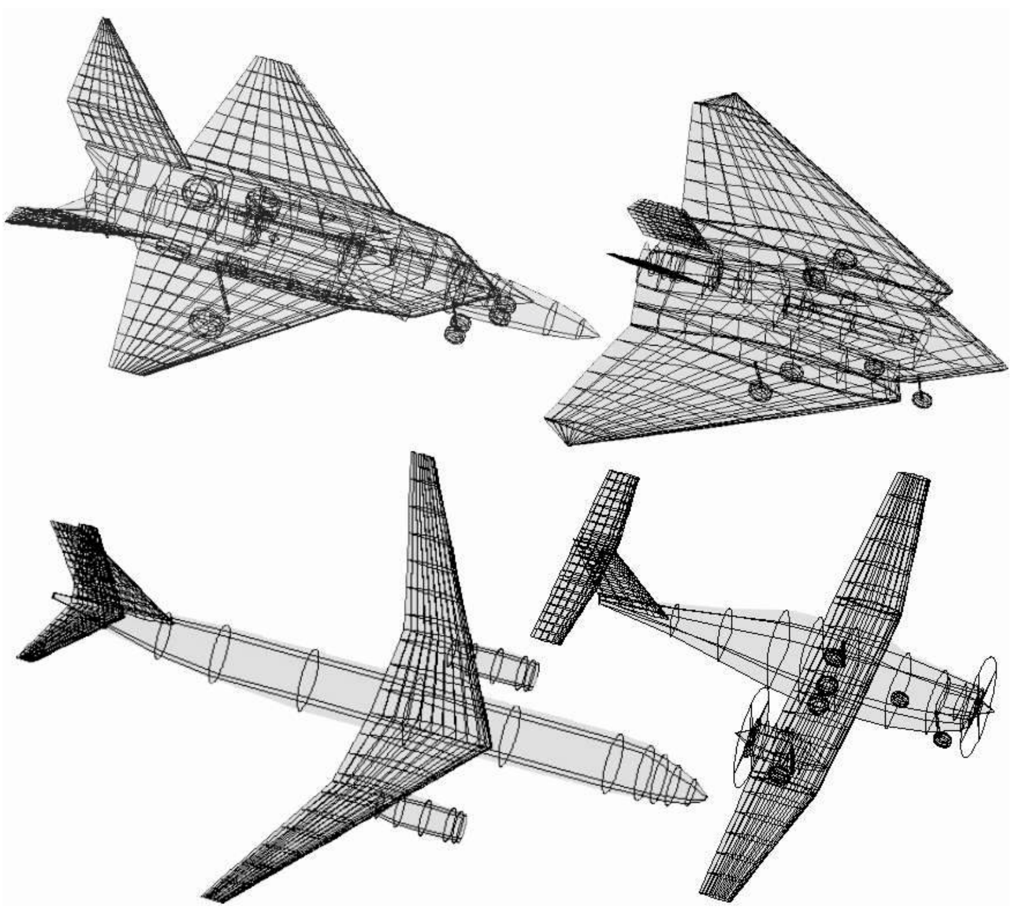


figure 1. Validation Models: Four Aircraft Notional Concepts

Each was designed and analyzed using the RDS-Professional program, and the results were compared to existing aircraft to ensure reasonableness and credibility of the data. MDO verification tests were conducted to determine any design-specific problems with the methods or the code.

4 APPROACH AND METHODS

4.1 Overview of Approach

The fundamental approach to the research described herein was to develop a sophisticated aircraft conceptual design computer program featuring a wide variety of MDO methods and options, incorporating a variety of design variables and automated vehicle redesign procedures, and then to run comparisons for four different notional aircraft design concepts. In all, over a million parametric aircraft designs were generated and analyzed in this research, and well over a hundred MDO runs were conducted along with numerous two-variable carpet plots for comparison.

In work previously reported on, this author developed the RDS computer program including Professional and Student versions. RDS[5] includes sophisticated implementations of the classical analysis methods[6] used in industry for many years, and incorporates a CAD module for initial 3-D layout of design concepts. RDS also includes a multidisciplinary optimizer based on the full-factorial Orthogonal Steepest Descent method described herein. RDS-Professional is available through Conceptual Research Corporation (PO Box 923156, Sylmar, CA, 91392, USA).

To conduct comparative optimizations using different aircraft conceptual design optimization methods, a highly flexible optimization module was programmed into the RDS-Professional program. This allows optimization, based on exactly the same inputs and analysis methods, using a variety of methods. These include the Orthogonal Steepest Descent Search, random Monte Carlo method, a collection of Genetic Algorithms, and an Evolutionary technique, as described below.

For these MDO methods, the program allows selecting from numerous options which, taken together, largely span the range of methods in use. These options, defined and detailed below, include:

- Number Of Individuals per Generation/Gang
- Total Number Of Generations/Gangs
- MOM Weighting Schemes:
    1. Linear
    2. MOM Rank Percentage-Squared
    3. MOM Rank Percentage-4th Power
    4. MOM Rank Percentage-Sine Wave
- Performance Penalty Factor and Variation (allows simulated annealing)
- Elitism (Best Survive Unchanged Into Next Generation)
- Option To Replace Individuals In Population After Breeding
- Breeder Pool Size (Percent Of Total Population)

- Mutation Probability Factor
- Breeding Crossover Options:
    1. Single-Point Crossover
    2. Uniform Crossover
    3. Parameter-Wise Crossover
- Geometric constraint holds including
    1. Fuselage maximum length
    2. Fuselage minimum diameter
    3. Wing maximum span
    4. Wing Aspect Ratio vs. Sweep to avoid pitchup
    5. *Net Design Volume* (described below)

Note that every option is not appropriate for every MDO method.

In operation, the optimizer begins by prompting the user for the analysis input files to use. These are normally the defaults for the design being optimized, which have previously been created by the user during the normal course of design evaluation. These include the inputs defining the performance constraints, which can include takeoff (ground roll, total takeoff distance, FAR 25 takeoff distance, or balanced field length), landing (landing ground roll, total landing distance, FAR 25 landing distance, or no-flare landing distance), rate of climb, time to climb, P_s at a given load factor, instantaneous turn rate, and acceleration time or distance.

The user then selects the MDO algorithm and options to employ. Next the user selects the objective function (Measure of Merit) which can be Takeoff Gross Weight (W_o), Empty Weight, Fuel Weight, Purchase Price, Life Cycle Cost, Net Present Value, or Internal Rate of Return. For designs with a fixed-size engine, the objective Measure of Merit is Range based on the user-defined mission. Also, the design space is defined by user inputs as to the maximum and minimum values of the design variables (with defaults of plus and minus 20%).

Following user selection of the appropriate optimization options as listed above, the program commences with parametric or random variations about the user-defined baseline design, depending on the MDO algorithm being employed. Each design variation is analyzed as to aerodynamics, weights, propulsion, sizing, performance, and cost. Sizing results (weight or range) or cost are used as the MOM, as selected by the user. Optimization then proceeds as detailed below.

Four validation models (aircraft test cases) were developed to test these methods and determine relative suitability for different classes of aircraft. Since the subject of this research is aircraft conceptual design, the test cases are notional aircraft designs. These include a jet transport, a single-engine fighter, an unmanned air vehicle, and a general aviation twin. These are defined in detail in the full thesis.

4.2 Orthogonal Steepest Descent Search

Orthogonal Steepest Descent, a full-factorial stepping search, has been successfully running in RDS-Professional for a number of years and has been used on various research projects such as that reported in Raymer and Burnside Clapp[7].

The Orthogonal Steepest Descent optimizer relies upon defining ratio multipliers for each of the design parameters, and adjusting those ratios until an optimum is found. These ratios are used to modify the analysis input data. For example, if the baseline wing loading is 100, the baseline design is represented by a wing loading multiplier ratio of 1.0. This is changed during the optimization until the best design is found with, say, a wing loading multiplier ratio of 0.88 (i.e., the optimal wing loading was found to be (100 x 0.88=88).

Optimization is done using step searching by a simple comparison method. Starting from a baseline aircraft definition, each variable is parametrically varied using these ratios by plus and minus some selected step size, in the same exhaustive manner as a full-factorial design of experiments. The resulting 3^n aircraft (where n = number of design variables) are all analyzed for aerodynamics, weights, sizing, cost, and performance. Propulsive thrust is merely ratioed to the defined *T/W* since, as of yet, no propulsion system design variables such as bypass ratio or propeller diameter have been included which would substantially change thrust or fuel consumption characteristics.

The "best" variant, that with the lowest value of the selected measure of merit that also meets all performance requirements, is remembered and when all parametric variations about the initial baseline are exhausted, becomes the center point baseline for the next iteration loop. This continues until no better variant is found, then the stepping distance is shortened and the process repeated until some desired level of resolution is obtained.

The Orthogonal Steepest Descent method is so simple and direct that it cannot get stuck in a loop or fail to find any solution at all unless the baseline aircraft is so poorly designed that neither it nor any parametric variations of it can meet all performance requirements. Also, it is deterministic, always finding the same solution to many decimal places when starting from the same baseline design. Therefore, it makes a good benchmark for study of other methods, especially those stochastic methods that may seem to converge but may actually fail to find the “true” best design.

4.3 Definitions and Operations for Chromosome-based Methods

The remaining MDO methods coded for this research are all related in that they are all stochastic in nature, and they all rely on a chromosome/gene bit-string to define the parametric variations of the aircraft being optimized. They also share many optimization options and parameters, as discussed below. As with the *OSD* optimizer, the MDO methods below all optimize for eight variables consisting of *T/W*, *W/S*, aspect ratio, taper ratio, sweep, thickness, fuselage fineness ratio, and wing design lift coefficient.

The following sections define this chromosome gene bit-string and the various operators used in the chromosome-based methods developed for this research.

4.3.1 Chromosome/Gene Bit-String Definition

In nature, the characteristics of an individual of a species are defined by *Genes*, which are connected together in a specified order forming *Chromosomes*. A similar scheme is employed for the Monte Carlo, Evolutionary, and Genetic algorithms as used herein. Specific values of design variables defining an individual aircraft are based on chromosome-like bit-strings comprised of ones (1s) and zeros (0s). Different values of those binary digits define a variety of alternative design permutations.

The following chromosome/gene bit-string definition is used:

T/W	W/S	A	taper	sweep	t/c	fuselage l/d	$C_{L\text{-design}}$
000000	000000	000000	000000	000000	000000	000000	000000

Each of the eight parameters is defined by a gene consisting of six binary digits that represent position on a spectrum from lowest to highest permitted value of that design variable, as input by the user. Thus, if the user allows wing loading to range from 40 to 100, the string 000000 represents 40, the string 111111 represents 100, and 001010 for example represents {40+(100-40)(10/63)=49.52}.

This chromosome scheme relies upon a user-defined baseline aircraft design that provides a point of departure for defining an initial population of designs. This design is created using normal aircraft design practice, and must have previously been analyzed as to aerodynamics, weights, propulsion, performance, sizing, range, cost, etc… The input data files developed to do that analysis are modified by the optimizer routine to develop different designs according to the codes in the chromosome string. If, say, the baseline design has a wing loading of 60 but the particular "individual" being created is supposed to have a wing loading of 90, then the aerodynamics and weights inputs for wing loading would be multiplied by 90/60=1.5. Other effects such as a change in tail size would also be made, again by changing the inputs to the appropriate analysis.

There is a subtle but important terminology issue for this study. The chromosome scheme has a "baseline design" that is used to develop the analysis input data, but that design is not a "starting design" or "basepoint". These optimizers do not start with this initial design and then search around for improvements – that is how the *OSD* method works. In the chromosome-based schemes, the baseline is only used to generate the initial population, which may or may not include that original baseline! It could be said that the baseline design concept in a chromosome-based scheme is really an analysis calibration device rather than an initial design.

In all the chromosome-based routines, an initial population of up to 500 designs is created by using a digital random number generator to create each bit in the chromosome string. Then, this string is used to change the input variables of the baseline design, creating a unique "individual" for each chromosome string defined. Where the optimizers differ is how they proceed after this initial population is created.

4.3.2 Selection - MOM Weighting (removed-see full thesis)

4.3.3 Selection - Performance Penalty Function

An essential part of engineering optimization is the use of constraints. These are typically "must-meet" performance requirements or real-world geometric constraints such as a maximum permitted wingspan (if violated, the airplane won't fit into the terminal gate). In classical carpet plot optimization, the constraints are lines on the graph, shaded to represent the "don't-go" (infeasible) direction. In most cases the optimum solution is found where two of the constraint lines intersect or where the objective function is tangent to a constraint line.

In the first version of a Genetic Algorithm developed for this research, a similar "don't-go" strategy was employed. Aircraft variants that violated one or more constraints were "killed", with no chance of reproduction or continuation into the next generation. This method worked, but typically led to the immediate elimination of about 65% of the population for the first generation. Later a subtler and less brutal strategy was incorporated as an option based on the *Penalty Function Method*.

In the *Penalty Function Method*, constraints are turned into adjustments to the measure of merit. If a constraint is violated, some function related to the amount of constraint violation is applied to degrade the calculated value of the objective function (measure of merit). For example, an aircraft with a takeoff distance in excess of the required value would have its weight (if that is the measure of merit) increased in some fashion from the actual calculated value.

The simplest possible Penalty Function Method was tried first, namely, a scalar penalty factor that is multiplied times the objective function (measure of merit) for each constraint that is violated. No attempt is made to decide by how much the constraint was missed, nor the relative importance of, say, missed takeoff distance vs. missed turn rate. If a design misses two performance constraints, its objective function is twice multiplied by the penalty factor in use.

Furthermore, provisions were made to allow starting this penalty factor multiplier at one value and linearly increasing it to another value as the optimization proceeds. By starting at 1.0 (no penalty) and increasing to a high value such as 2.0, a form of Simulated Annealing is obtained. By starting and ending at a high value such as 2.0, the "immediate-kill" of classical aircraft optimization is obtained.

This simple version of a Penalty Function Method proved to work well.

4.3.4 Elitism and Replacement

During the execution of Evolutionary/Genetic algorithms, a counter-convergence effect is sometimes seen. Literally, the next generation is worse than its predecessor generation, or at least, the best individual in the next generation is sometimes worse than the best of the prior generation. This is to be expected due to the stochastic nature of such optimizations.

A simple means of preventing such "backsliding" is to take the best individual of each generation into the next generation. Then, if none of the new generation is any better, that generation's best is unchanged from the prior generation. This is called "Elitism", and is implemented herein by allowing the user to specify up to 50 top members of each generation to be inserted into the next generation.

Another option separating various evolutionary and *GA* schemes is the decision as to what to do with chosen parents after they have "bred". Some favor discarding them, others favor replacing them in the "pool" to be selected again (if lucky). The implementation herein allows either option, termed *With Replacement* and *Without Replacement*.

4.3.5 Chromosome String Crossover

Essential to Genetic Algorithms is the concept of crossover, equivalent to mating in the real world of biology. Crossover is the method of taking the chromosome/gene strings of two parents and creating a child from them. Many options exist, allowing a nearly limitless range of variations on *GA* methods. The following options were coded into the RDS-Professional MDO module.

Single-Point Crossover: Performs the combination of genetic information from two parents by breaking their chromosomes into two pieces, sticking the first part of one parent's chromosome with the second part of the other's. The point where the chromosome bit-strings are broken can be either the midpoint or a randomly selected point.

Uniform Crossover: Combines genetic information from two parents by considering every bit separately. For each bit, the values of the two parents are inspected. If they match (both are zero or both are one), then that value is recorded for the child. If the parents' values differ, then a random value is selected.

Parameter-Wise Crossover: Combines parent information using entire genes defining the design parameters such as *T/W*. Here, each gene is six bits. For each gene, one parent is randomly selected to provide the entire gene for the child. Mutation (see below) is especially important for this crossover method because otherwise, only design parameter values found in the original population would ever be found in the final population.

4.3.6 Mutation

Mutation is applied to the offspring immediately after the crossover (mating) operation is performed. Mutation is done by considering every bit in the new chromosome, and multiplying a random number (0-1) times a probability factor constant. If this product is less than one, the bit in question is "flipped" from zero to one or vice versa. Therefore, the numerical value of this probability factor is simply the inverse of the percent likelihood of the bit being "flipped" – a high value offers a low chance of mutation.

4.3.7 Convergence Measure – Chromosome *Bit-String Affinity*

Evolutionary and Genetic algorithms are iterative in nature, with a (hopefully) better and better result appearing as the solution progresses. This approach to the final "best" answer is called *convergence*, and is both an indication as to whether a solution is emerging, and an aid to the decision to stop the optimization and declare a solution.

In the evolutionary and genetic algorithms used in this research, convergence can be seen on the output graphs of measure of merit vs. iteration number. The convergence ratio was calculated for each run but was of little use because the convergence of these methods does not tend to follow a sure and steady trend. Instead, it tends to jump around, sometimes flattening out as several generations go by without a better solution being found, and sometimes even reversing unless elitism is used as defined above. For this reason, a different measure of convergence was defined and employed in this research.

As the routine goes through generation after generation, it should be expected that "good" traits would begin to emerge. Furthermore, many individuals in the population should start to possess those good traits and thus, should begin to resemble each other. This should appear mathematically as an emerging similarity in chromosome bit-strings, and should be visually observable in the bit-strings. For example, after several generations one may note that the sixth bit positions in the individuals' bit-strings are now mostly ones, whereas the eighth bit positions may be mostly zeros.

When starting such an evolutionary method, the bits should initially have a random distribution. When the bits become completely nonrandom (all individuals have identical bits), the population is identical and the method can go no further unless mutation is introduced. This progression from randomness to non-randomness provides a clear indication of the progression towards convergence.

To calculate this bit-string indication of convergence, a *Bit-String Affinity* term is defined in which a calculated value of zero (0) indicates a random population whereas a calculated value of 100 indicates an identical population. This is determined from the average distance of each of the bit positions in the entire population from either one (1) or zero (0).

Bit-String Affinity is calculated by taking an average of the first bit position value for all the individuals, then an average of the second bit position value for all the individuals, and so on for all the bits in the bit-string as defined for the optimization. For each of these resulting averages, the distance from either 1 or 0 is determined as that average itself if less than 0.5, and as 1.0 minus that average if greater than 0.5. (obviously, the distance is 0.5 if the average is exactly 0.5). Then, these distances for each bit position are averaged for a total averaged distance.

This calculation yields a value in a range from 0.5 if purely random to exactly zero if all bits are identical. This is converted by a simple transformation to a more-intuitive measure spanning a range from zero if purely random, to 100% Bit-String Affinity if all individuals are identical.

This Bit-String Affinity calculation is trivially simple to implement yet has been found to be a powerful and intuitively useful measure of convergence. This concept seems to be new to the field.

Bit-String Affinity has been run on numerous optimization cases using a variety of Evolutionary/Genetic routines. After observing the usefulness of the Bit-String Affinity value, it was coded as an alternative stopping criterion (stop when >98%). This has, in some cases, terminated execution many generations before the intended stopping point thus saving unneeded execution time. The best airplane that would be found had already been found.

4.4 Monte Carlo Random Search

A Monte Carlo optimization method was programmed using the chromosome/gene string definition detailed above. This works by randomly creating and analyzing thousands of different aircraft and testing for the one with the best measure of merit that also meets all required performance points. To simplify coding and reduce memory requirements, the total population desired is generated and analyzed in "gangs" of 500, but there is no evolutionary component to the optimization. Each gang is produced purely by application of random numbers to create chromosome/gene bit-strings, and the best of all gangs is the selected best aircraft. Typically, 20 gangs of 500 would be created yielding a total of 10,000 individuals.

4.5 Genetic Algorithms

Genetic Algorithms are stochastic Evolutionary Algorithms with a close analogy to real-world biology. Essential to a Genetic Algorithm is the selection of the parents and the combination of their genes to produce the next generation (crossover). Coding for crossover as used in this research is detailed above. Methods employed for the selection of "parents" are described below.

4.5.1 Roulette Selection

Holland popularized the use of Roulette Selection to determine the "lucky parents". This is like the gambling device, but the sizes of the "slots" into which the random "ball" can fall are determined by the calculated values of the measure of merit as shown in figure 2 (based on actual data from a fighter optimization run conducted for this research).
Sizes of the slots are calculated as:

$$SlotSize_i = \frac{MOM(i)}{\sum_i MOM(i)}$$

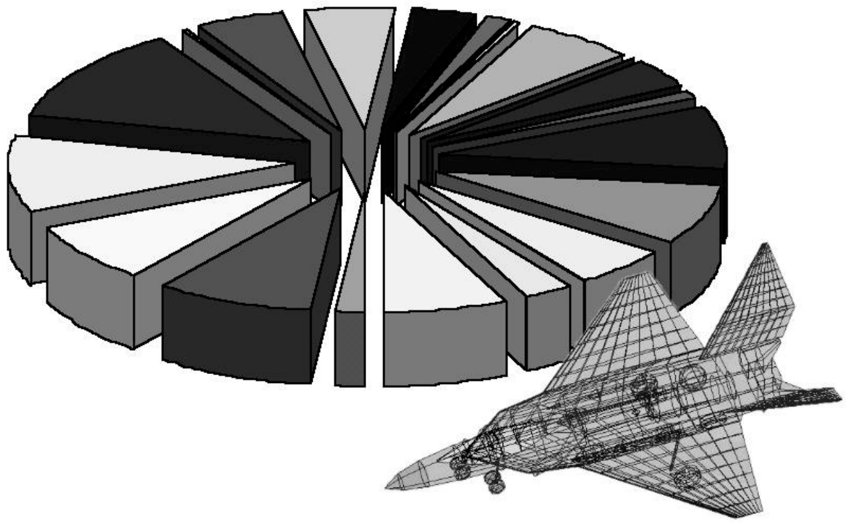


figure 2. Roulette Selection

If a penalty function method is in use to handle performance constraints, the slot size is based on the performance-adjusted MOM. If performance is not met, that aircraft's slot is made smaller.

4.5.2 Tournament Selection (1v1)

Tournament Selection, preferred by many recent researchers, selects four individuals at random. They "fight" one-vs.-one, with the superior of each pairing being allowed to reproduce with the other "winner", as shown in figure 3.

As implemented herein, each "winner" pair produces two offspring by two independent crossover operations. This creates a new population that is as large as the previous population.

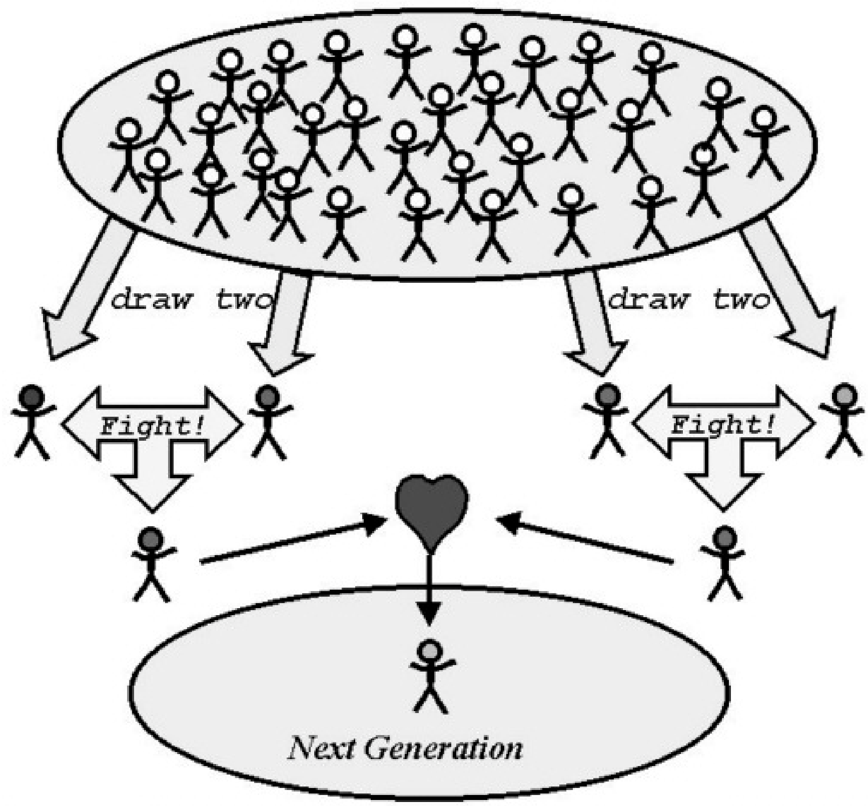


figure 3. Tournament Selection

4.5.3 Breeder Pool Selection

A selection scheme based on real-world biological reproduction was defined, and seems to be original to this research. In nature, the survival process is usually decoupled from the selection process. In many species, breeding selection is a fairly random event from among those who have survived long enough to reach the reproductive age of the species.

To mimic this in an MDO routine, the population of aircraft is analyzed and stacked as to fitness according to their value of the selected measure of merit (MOM). A user-specified percentage (default 25%) of the total population is then placed into a "breeder pool". The smaller the percentage used, the more "elite" the optimization becomes, favoring those with high values of the measure of merit but at the expense of reduced genetic diversity (and vice-versa). Use of 100% selection would allow all members of the parent generation to enter the breeder pool, essentially ignoring the MOM results and preventing any improvement with successive generations.

Then, two individuals are randomly drawn from the breeder pool and a crossover operation is used to create a member of the next generation. Once an individual is in the breeder pool there is no further competition except for the "luck" of being selected. The competition has already occurred in the selection to be included in the breeding sub-population. The breeder pool scheme is shown in figure 4.

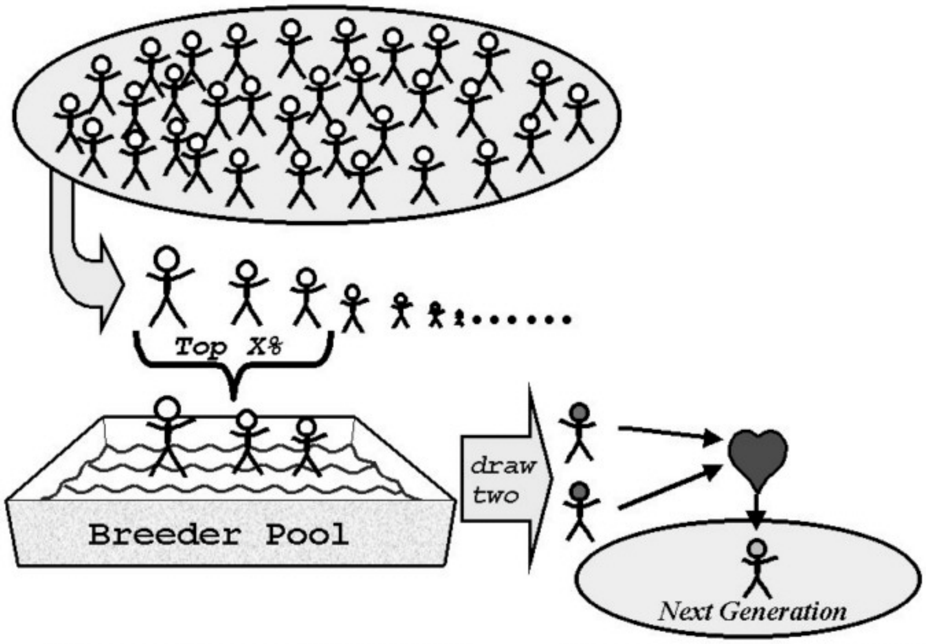


figure 4. Breeder Pool Selection

4.6 Evolutionary Algorithm – Best Self-Clones with Mutation

The final Evolutionary algorithm employed in this research can not be considered a Genetic Algorithm because crossover is not employed. This approach, a variant of *Evolutionary Programming*, is based more on the biology of ants. From an initial population, a best individual is found by MOM ranking, including application of the performance penalty method.

This best individual becomes the "queen" and sole parent of the next generation. This next generation is created by making copies (clones) of the queen's chromosome bit-string and applying a high mutation rate to generate a diverse next generation. The mutation rate is high enough that almost every child is mutated in some way, so the entire design space is being reconsidered during every iteration even as the method converges.

This method can be considered the ultimate in Elitism. Since the Queen alone reproduces, eliminating all other members of her population from reproduction, this author refers to the method as the "Killer Queen". Similar methods have been used by other researchers, especially in Europe.

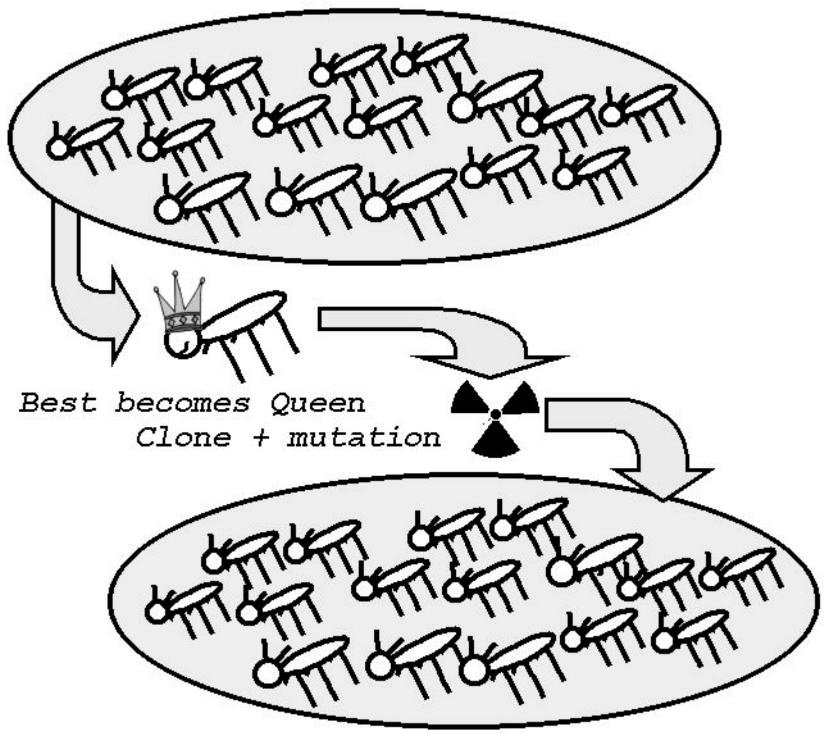


figure 5. "Killer Queen" – Best Self-Clones with Mutation

4.7 Hybrid Methods

The Orthogonal Steepest Descent method may find a local solution rather than a global optimum depending upon where it is started, and may take too long in getting to that optimum region. The other methods, all stochastic, offer a better hope of finding a global rather than local optimum but may never actually find the true best answer, and they may take many iterations to slightly improve the result.

A hybrid method may offer the best of both, so it was coded into the RDS-Professional program. Any of the stochastic methods (Monte Carlo, Evolutionary, or Genetic) can be used for a specified number of generations or gangs, followed by an Orthogonal Steepest Descent "fine tuning" starting from the best result from the stochastic method.

4.8 Analysis Methods Used for Optimization

The optimization methods reported herein all rely upon the well-proven RDS analysis modules developed by this author[5]. These include calculation of aerodynamics, weights, propulsion, sizing, range, performance, and cost. They represent a balanced collection of classical methods suitable for conceptual and early preliminary design and are described in detail in this author's aircraft design textbook, *Aircraft Design: A Conceptual Approach*.

Methods include component buildup for parasitic drag, leading edge suction and DATCOM charts for drag-due-to-lift and maximum lift, detailed empirical equations for weights, jet engine installation equations, propeller analysis from efficiency charts, industry-standard empirical cost equations, and physics-based equations for performance and sizing. These methods have been calibrated and tested in numerous studies over a ten-year period, and have been found to be quite reliable for most types of aircraft. Altogether, these analysis modules represent approximately 20,000 lines of source code and are further described in the Appendices (see full thesis).

4.9 Test-Case Run Matrix

To guide in the execution of test cases, a matrix was developed defining the test-case runs that would be conducted, including which validation model (aircraft notional concept) would be employed, which MDO method would be used, and which combination of options would be applied. This test-case run matrix is provided in full in the Appendices (see full thesis). In all, over a hundred optimizations were run totally over a million parametric aircraft design cases.

5 DESIGN VARIABLES, AUTOMATED REDESIGN PROCEDURES, AND GEOMETRIC CONSTRAINTS

An important issue for aircraft conceptual design MDO is the realism problem. To obtain a realistic revised design from an optimization routine, automated redesign procedures are required. These should approximate the changes that an experienced designer would make to an existing layout based on particular parametric revisions to the design variables. So, if some change to the parametric definition of the fuselage prevents the landing gear from working properly, a human designer would fix it – and so should the computer during MDO evaluations.

5.1 The Basic Six (or Five) Design Variables

In prior published work (see full thesis Appendices), this author identified the six most-important variables for aircraft conceptual design optimization as:

- *T/W or P/W (i.e., engine size defined by ratio)*
- *W/S (i.e., wing area defined by ratio)*
- *Aspect Ratio*
- *Taper Ratio*
- *Sweep*
- *Airfoil t/c*

These six variables include the performance-driving thrust and wing area, plus the parameters that define the basic wing geometry. These have at least 50 years of history behind them as key optimization variables, and in this author's opinion they should be the foundation of any optimization method intended for aircraft conceptual design.

If designing to an existing (fixed-size) engine, then engine scaling is not possible so a parametric variation of *T/W (*or *P/W)* is not possible, hence only five key variables remain.

In addition to the obvious direct changes to the analysis inputs as these design variables are changed, the aircraft analysis inputs are further modified as follows:

- Thrust and fuel flow vary by *T/W* or *P/W*
- Wing reference area varies based on *W/S*
- Wing exposed area varies based on *W/S*, adjusted for fuselage width cutoff
- Tail areas vary by the 3/2 power of wing area to hold constant tail volume coefficient
- Maximum cross-section area for wave drag calculation varies by wing area, *t/c*, and by cos(wing sweep), weighted to baseline percentage of total cross-section area
- Nacelle wetted area varies by *T/W*
- Wing fuel volume varies by 3/2 power of wing area
- Airfoil $C_{l\text{-}max}$ varies with *t/c* using empirical regression of NACA airfoils

- Airfoil leading edge sharpness parameter (ΔY) varies with *t/c*

5.2 Fuselage Fineness Ratio

The key top-level parameter for fuselage design is the fineness ratio (*f*), the fuselage length divided by its equivalent diameter (diameter that gives the actual cross-section area).

To find the true "best" fuselage fineness ratio, it must be included as a design variable in a multidisciplinary optimization. This was added to the RDS MDO routines, with the following automatic redesign procedures employed in addition to the obvious input revisions to fuselage diameter and length:

- To hold fuselage volume constant, diameter varies by cube root of f_{old}/f_{new}
- Tail areas vary inversely with fuselage length to maintain constant tail volume coefficient
- Landing gear length is scaled to maintain tail-down angle as fuselage length changes
- Maximum cross-section area for wave drag calculation varies by fuselage diameter as fineness ratio changes, weighted to baseline percentage of total cross-section area.

5.3 Design Lift Coefficient (Wing)

Another wing design parameter with great influence on the resulting aircraft is the wing Design Lift Coefficient ($C_{L\text{-}design}$). This is used during preliminary design as a target for optimization of twist, camber, and airfoil shape. Selection of a high design lift coefficient is equivalent to selection or design of an airfoil with high camber, which provides lots of lift at lower speeds but also lots of drag in cruising flight.

$C_{L\text{-}design}$ was added as the eighth variable in the RDS MDO routines. In addition to simply changing its value in the aerodynamic analysis inputs, the following effects were included:

- Airfoil leading edge sharpness parameter (ΔY) varies with design Cl via camber geometric approximation
- Airfoil Cl-max varies with design Cl using new empirical regression of data for several NACA airfoils[8] (which also includes variation with t/c)

5.4 Geometric Design Constraints

Geometric design constraints were added to the RDS MDO routines to permit searching for an optimal design with certain real-world requirements considered. These are treated in the optimization as additional performance constraints. Violations of them, like missing a takeoff distance requirement, are handled by multiplication of the calculated value of the measure of merit by the current value of the scalar penalty factor.

5.4.1 Fuselage Length and Diameter

Fuselage length and diameter limits can be input by the user at the initialization of the optimization. The length limit is an *upper* limit, often required in the design of military aircraft to ensure that the aircraft will fit in hardened shelters and on aircraft carriers.

The fuselage diameter limit serves as a *lower* limit. This prevents the optimization from making the fuselage smaller in cross-section than necessary to hold passengers, payload, or equipment as determined in the baseline configuration drawing.

5.4.2 Wingspan

The wingspan limit is an *upper* limit, based on a value input by the user, and mostly applied to large commercial transports to ensure usability of existing airport taxiways and gates. For military aircraft, span is constrained to allow the aircraft to fit in hardened shelters and on aircraft carriers. During conceptual design for the project that became the F-22, one thing that was known early was that the wingspan could not exceed that of the F-15, for just that reason.

5.4.3 Wing Geometry for Pitchup Avoidance

For a tailless aircraft or one with a tail positioned such that its effectiveness may be degraded at high angle of attack, it is important to avoid certain combinations of high aspect ratio and high sweep. Otherwise, near the stall the outflow from the high sweep will cause the tips to lose lift first. Due to the high aspect ratio this lost lift is located behind the center of gravity causing pitchup – an uncontrollable nose-up divergence leading to stall and spin.

A widely used pitchup avoidance criterion was detailed in NACA 1093. This gives a chart based on extensive wind tunnel testing that provides threshold curves of acceptable combinations of aspect ratio and sweep. Data for maximum allowed aspect ratio *(A)* were curve-fit for subsonic and transonic flight based on wing quarter-chord sweep (Δ_{QC}). These equations were added to the RDS MDO routines as an optional geometric constraint option.

5.5 Net Design Volume

The final geometric design constraint option added to the RDS MDO routines is intended to ensure that an optimization that makes the wing substantially smaller does not result in an aircraft that cannot hold its required fuel and internal equipment. This is done with the aid of a parameter called *Net Design Volume (NDV).*

Net Design Volume was defined by this author (Raymer[9]) as the internal volume of an aircraft less the volume dedicated to fuel, propulsion, and payload (including passengers and crew). *NDV* represents the volume available for everything else, including items that are not precisely known until well into the design process such as structural components, avionics, systems, equipment, landing gear, routing, and access provisions. Therefore, *NDV* can be used to assure that a design layout has a credible geometry such that the design, when finalized, will contain all required components without requiring excessively tight packaging, which can lead to fabrication and maintenance difficulties. Furthermore, *NDV* assessment can be used as a constraint in MDO optimization to help improve the design realism of the resulting optimized configuration.

5.5.1 **Definition Of Net Design Volume** (see full thesis)

5.5.2 Use Of *Net Design Volume* For MDO

NDV can be used to evaluate a just-completed aircraft configuration layout for historical reasonableness. Another usage, and a key objective of this study, is as a constraint factor in multidisciplinary design optimization (MDO).

NDV was applied to the RDS MDO routines to automatically "correct" the design geometry resulting from every parametric variation of the baseline, using the following steps:

- Calculate *NDV* density target from analysis of the baseline design layout prior to start of optimization
- During MDO optimization, analyze each design perturbation for *NDV* density
- Modify fuselage analysis inputs to photographically scale it in all directions to restore the target *NDV* density
- Scale landing gear length for new fuselage length
- Revise tail areas for new fuselage length
- Perform aircraft analysis and sizing
- Check other geometric constraints for violation (such as fuselage diameter too small)

5.6 Automated Redesign for Discrete Variables (see full thesis)

6 NOTIONAL AIRCRAFT CONCEPTS (see full thesis)

7 RESULTS

Development of the MDO modules incorporating features and options as described above was completed according to plan. The four notional aircraft concepts were designed, analyzed, and optimized using these routines. Numerous variations in MDO methods and options were run along with a number of trade studies of the use of various geometric constraints and automated aircraft redesign procedures. A test-case run matrix is presented in the Appendices (see full thesis) which also provides a summary of the results of the MDO analysis conducted for this research, including the final value of the selected Measure of Merit (price for the fighter, takeoff gross weight for the others). Also included, for the chromosome-based MDO routines, are the final value of Bit-String Affinity and the percent of the final population meeting all performance requirements.

7.1 Calibration Results: Orthogonal Steepest Descent Search (see full thesis)

7.2 Stochastic MDO Results

A total of 25 MDO runs were initially conducted, encompassing all of the chromosome-based stochastic methods developed for this research including Monte Carlo, three Genetic Algorithms (Tournament, Roulette, and Breeder Pool), and the Evolutionary scheme here called “Killer Queen” (runs 11-36). Altogether, this totaled about 250,000 parametric aircraft designs, each one defined by a chromosome gene bit-string and subjected to aerodynamics and weights analysis followed by sizing, performance, and cost calculations. All of the runs for the fighter aircraft were then completely redone to see if the results were similar. They were, providing some confidence that these results are repeatable in spite of the stochastic nature of these optimization methods.

Results are graphed in the following figures, showing the convergence of the Measure of Merit. In each case, the *OSD* results are included for comparison. The *OSD* final value of the measure of merit was superior to all of the other methods for all four aircraft, but by a fairly small amount. And, the *OSD* optimization generally took two to three times as many case evaluations to get to the optimum. However, even the long *OSD* optimizations took only about 10-30 minutes each on a 1 gHz personal computer, while the other methods averaged about 10 minutes each.

Optimization results for the Advanced Multirole Export Fighter are shown in figure 6. Note that an *OSD* best solution first appears after 6561 cases are evaluated. That is one full-factorial evaluation around the baseline design for eight variables, yielding 3^8 cases. The other methods have a first result that is simply the best member of a random initial population (default 500 in these calculations). Any differences in this starting value are pure luck and reveal nothing about the method employed!

By the time of the first full-factorial baseline parametric results with the *OSD* method, all of the stochastic methods (including Monte Carlo) have attained a result nearly as good as the *OSD* final solution. In the end, though, the *OSD* method finds a slightly better solution, and it gets the same result every time it is run.

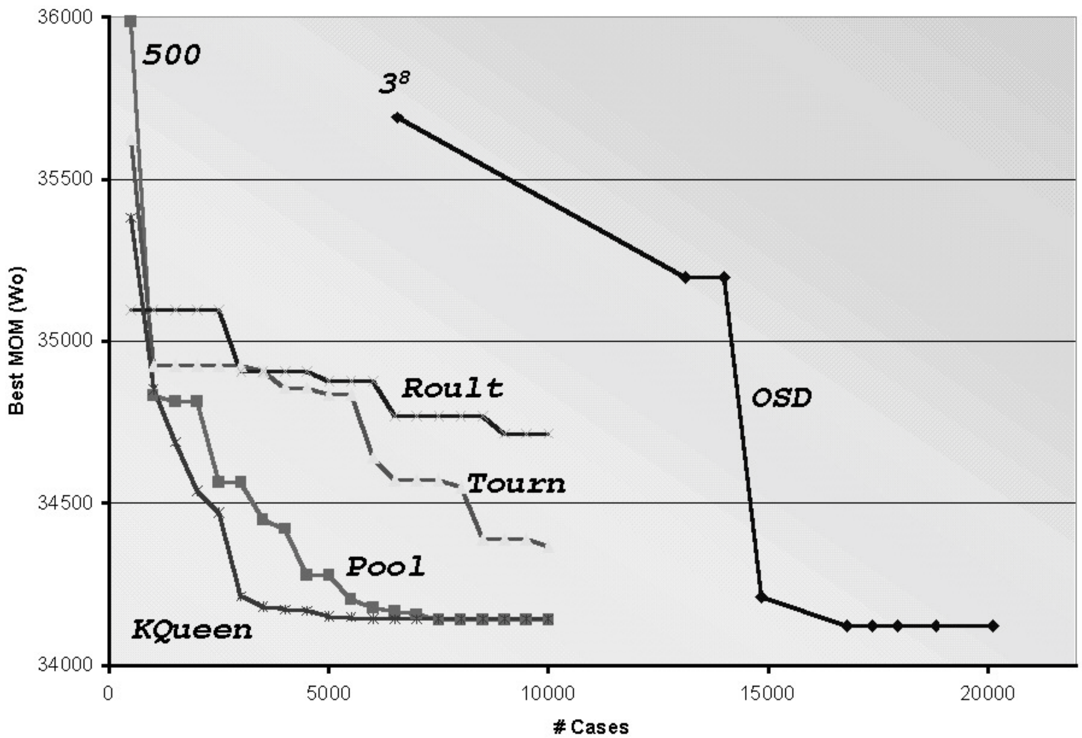


figure 6. MDO Solution Convergence: Fighter

Relative convergence rates of the stochastic methods favor the "Killer Queen", which creates a new generation simply by copying the best member of a generation and applying a high mutation rate. Close behind is the Breeder Pool, where a weighted Measure of Merit ranking is used to isolate a superior subpopulation from which individuals are selected at random for reproduction. Roulette selection appears to be the worst.

Bit-String Affinity was defined as an indication of the sameness of the members of a population or generation. Bit-String Affinity equals zero for a totally random population, and equals 100% for a completely identical population. This provides a useful and visual convergence criteria, and was included as an alternative stopping criterion in these routines. In at least six of the runs, Bit-String Affinity stopped the run early when all members of the population became virtually identical.

The following figure depicts the progression of Bit-String Affinity for the fighter aircraft using these MDO methods. Observe that the Killer Queen method begins, like the other methods, at virtually random (near zero) but immediately jumps to a high value where it remains for all subsequent generations. This is to be expected because after the first generation, all populations are created from mutated copies of the single best individual of the previous generation. The Bit-String Affinity in this case is just a reflection of the mutation rate being used.

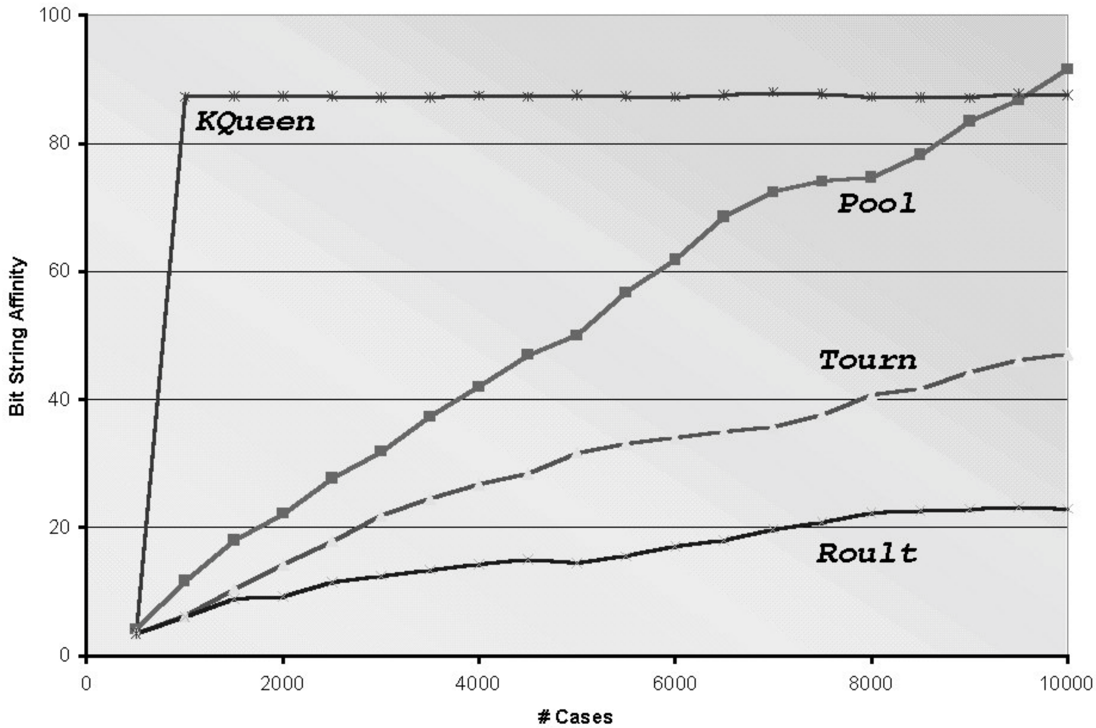


figure 7. Bit-String Affinity: Fighter

This Bit-String Affinity measure indicates that the Breeder Pool is producing the strongest convergence. As can be seen, the Roulette method seems to have a difficult time converging.

Results of the MDO runs for the civilian airliner are shown next. The *OSD* method begins with an initial value so high that it is almost off the scale. This is due to the “badness” of the initial baseline design, which was deliberately modified to reflect poor choice of design variables. So, the initial baseline and all variations around that baseline are poorly designed and hence, are heavy. Then, the *OSD* method must step away from this “bad” part of the design space, and that takes a large number of steps.

The same trends as to which stochastic method converge the fastest hold for this concept as well. Apparently the Tournament method got lucky in the first population, but was only able to slightly improve upon it afterwards. Bit-String Affinity for the transport is shown in figure 9.

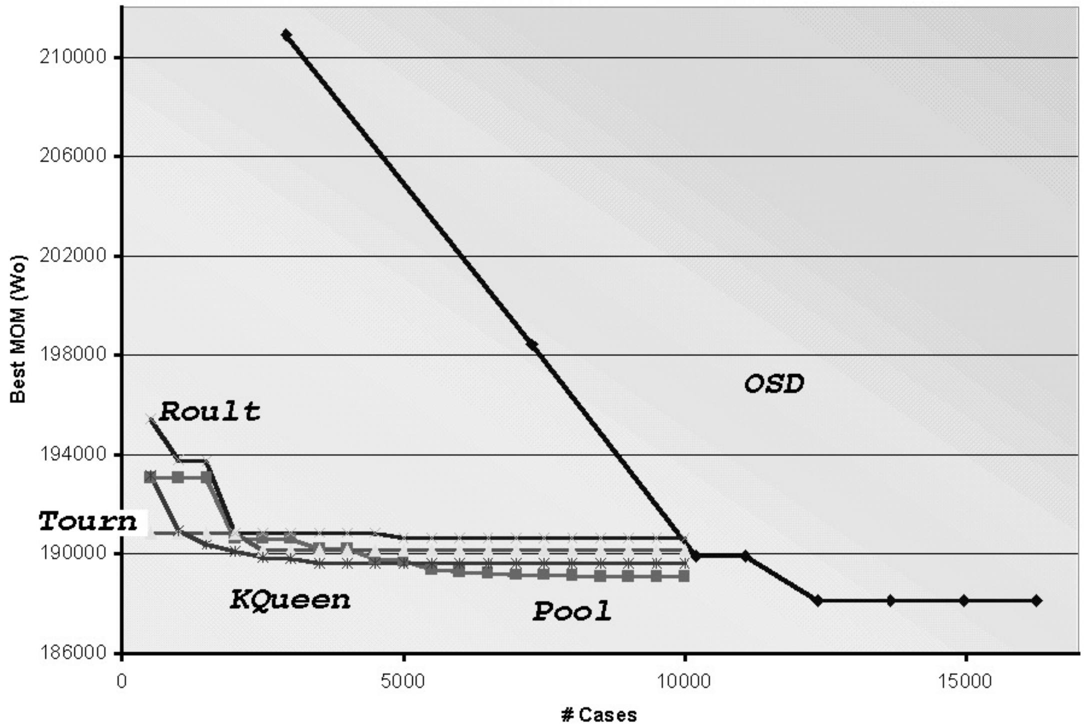


figure 8. MDO Solution Convergence: Transport

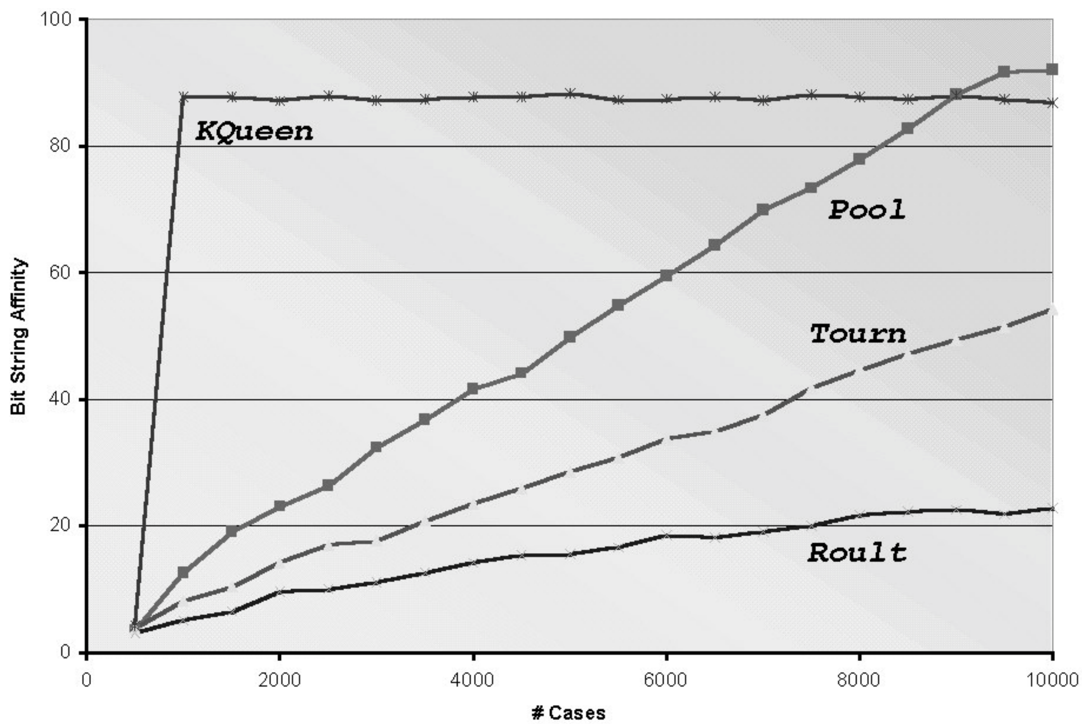


figure 9. Bit-String Affinity: Transport

The Asymmetric Light Twin is optimized using only seven variables since the *T/W* ratio cannot be used (fixed-size engine). The *OSD* method is very sensitive to the total number of variables. With fewer design variables, it managed to beat the stochastic methods to a solution, and found a better solution as well. Of the stochastic methods, the Breeder Pool performed the best (figure 10).

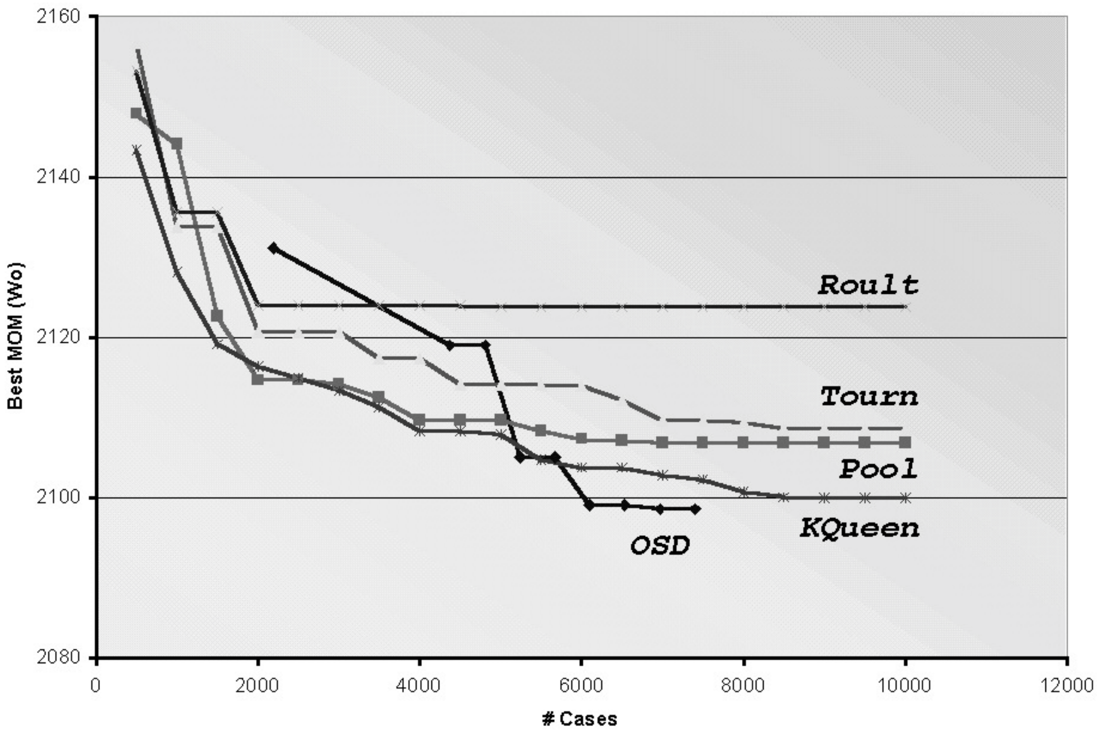


figure 10. MDO Solution Convergence: Light Twin

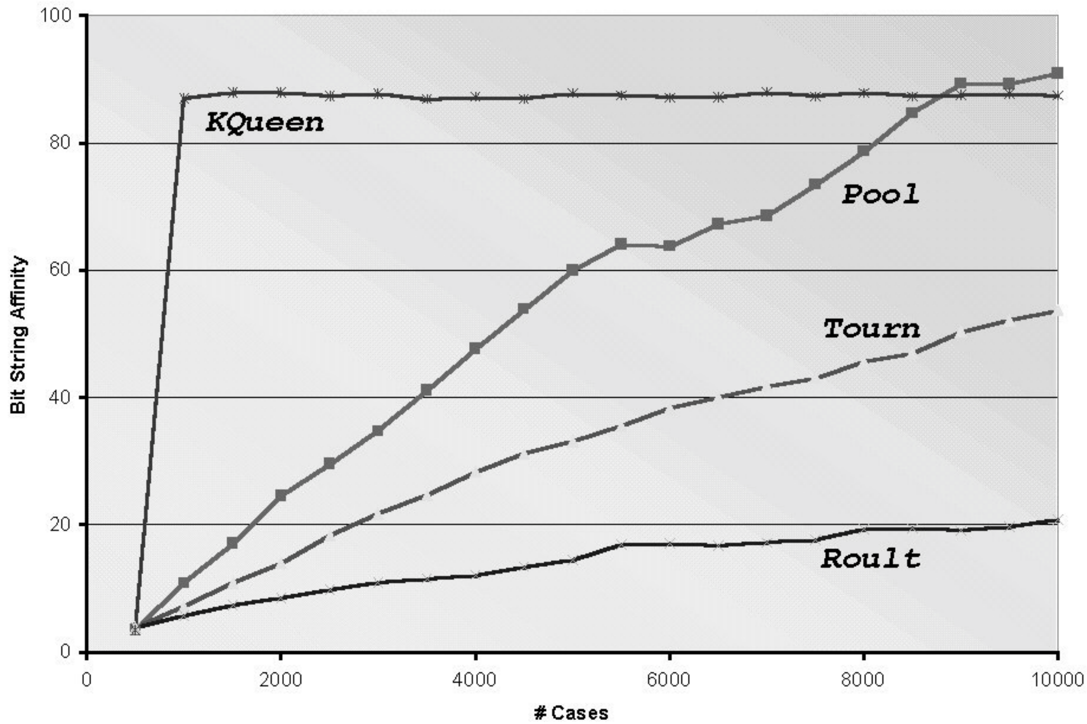


figure 11. Bit-String Affinity: Light Twin

Similarly, the UAV design uses only six variables (no *T/W* or fuselage fineness ratio), and the OSD method performs even better relative to the stochastic methods. This is depicted in figure 12.

This author expects an extrapolation in the other direction to follow this same trend. Increasing the number of variables beyond the eight used in this research would probably bring the *OSD* method almost to a halt with today's computers, while the stochastic methods would be less affected.

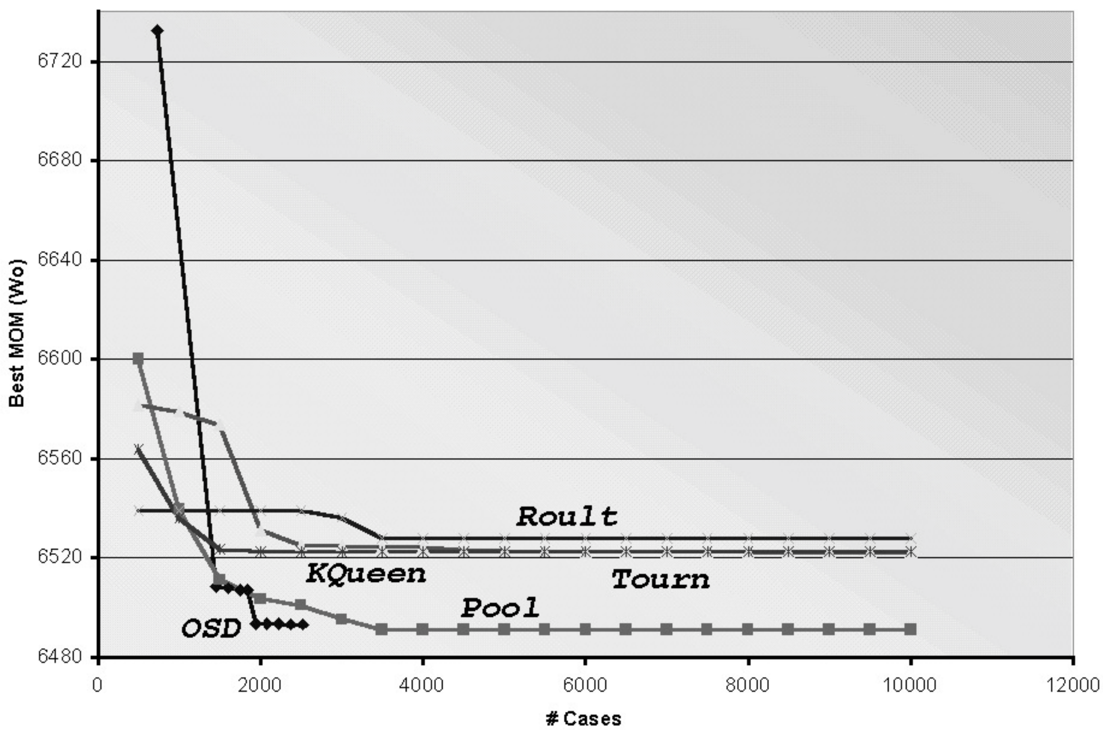


figure 12. MDO Solution Convergence: UAV

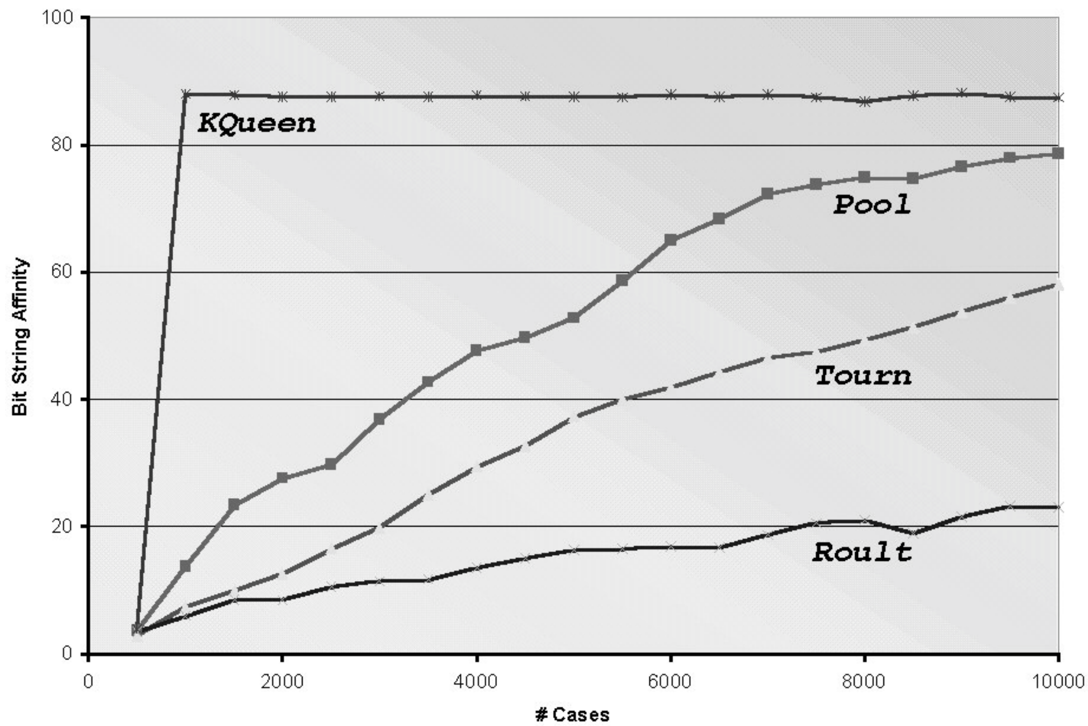


figure 13. Bit-String Affinity: UAV

To better understand the relative performance of these methods it is useful to know the number of parametric case evaluations required to attain a certain "goodness" of result. The deterministic *OSD* result can be used as a benchmark. In figure 14, the number of parametric evaluations required to find a result only 1% higher than the OSD solution is graphed. This is shown in figure 15 for a result that is 2% over the *OSD* solution. These charts seem to indicate that as few as four generations of 500 each will usually get to within 1-2% of the final result for the Breeder Pool and Killer Queen methods. This is, in some cases, a tenth of the number of case evaluations required to find (and know you have found) the best aircraft using *OSD*.

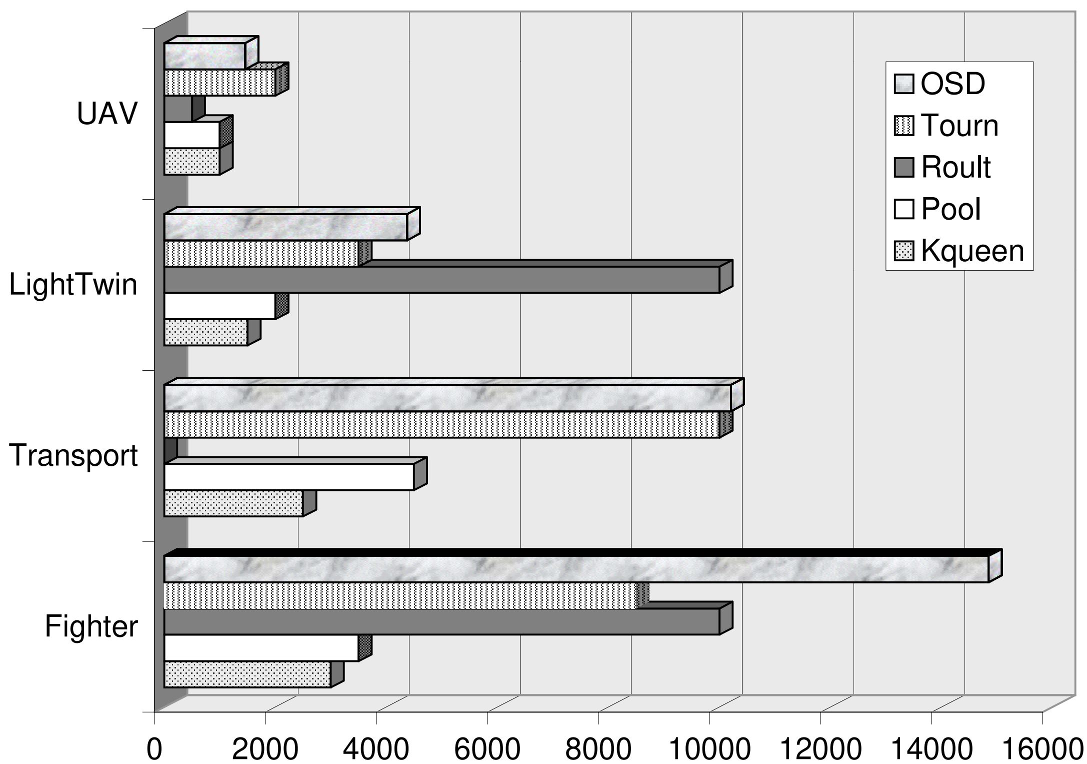


figure 14. Number of Runs Required to Come Within 1% of Best

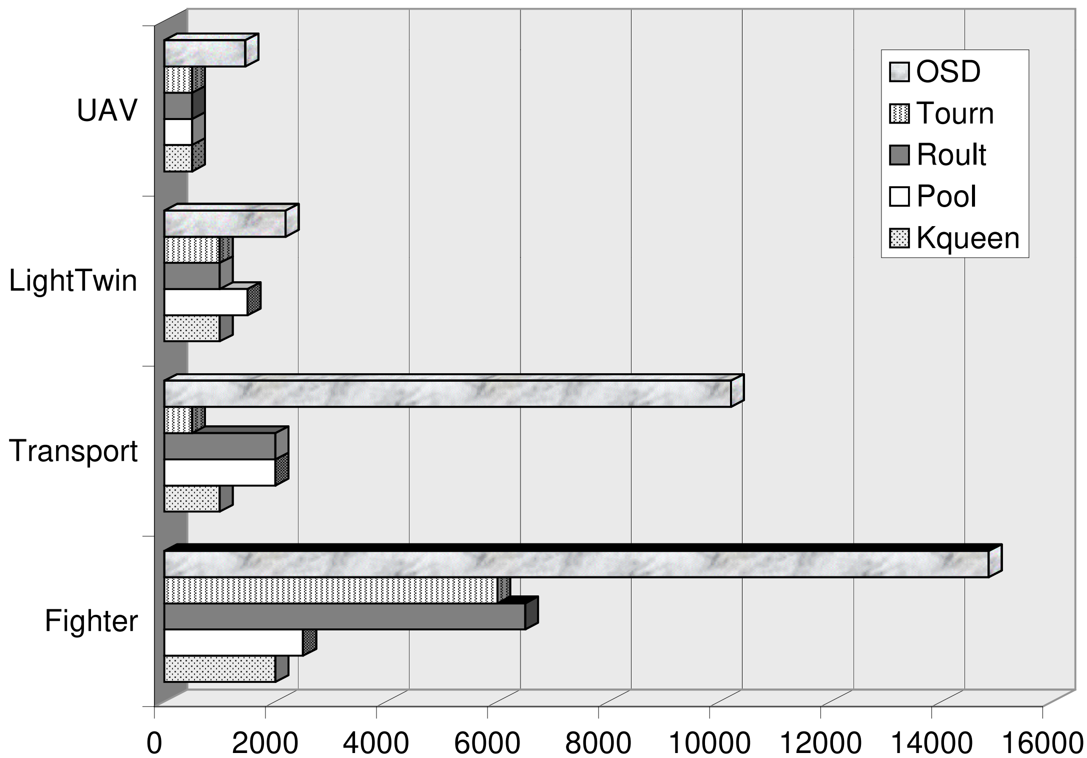


figure 15. Number of Runs Required to Come Within 2% of Best

8 SUMMARY AND CONCLUSIONS

Research has been conducted into the improvement of the Aircraft Conceptual Design process by the application of Multidisciplinary Optimization (MDO). Aircraft conceptual design analysis codes were incorporated into a variety of optimization methods including Orthogonal Steepest Descent, Monte Carlo, a mutation-based Evolutionary Algorithm, and three variants of the Genetic Algorithm with numerous options.

Four notional aircraft concepts were designed as test cases for evaluation of MDO methods and options, namely an advanced fighter, a commercial airliner, an asymmetrical light twin, and a tactical UAV. The commercial airliner design was deliberately modified for certain case runs using poorly-chosen design parameters including wing loading, sweep, and aspect ratio, to see if the MDO methods could “fix it.”

MDO methods and options were evaluated using these notional designs in over a hundred case runs totally more than a million parametric variations of these designs. These variations included application of automatic redesign procedures to improve the realism of such computer-designed aircraft. Each design variation was completely analyzed as to aerodynamics, weights, performance, cost, and mission sizing, and evaluated as to performance and geometric constraints.

The key conclusion – aircraft conceptual design *can* be improved by the proper application of such Multidisciplinary Optimization methods. MDO techniques can reduce the weight and cost of an aircraft design concept in the conceptual design phase by fairly minor changes to the key design variables. These methods proved to be superior to the traditional carpet plots used in the aircraft conceptual design process for many decades.

Evaluation of the different MDO methods for aircraft design optimization indicated that all of the methods produce reasonable results. For a smaller number of variables the deterministic Orthogonal Steepest Descent searching method provides a slightly better final result with about the same number of case evaluations. For more variables, evolutionary/genetic methods seem superior. The Breeder Pool approach defined herein seems to provide the best convergence in the fewest number of case evaluations.

The *Net Design Volume* approach defined herein to assure sufficient volume for fuel and internal equipment appears to work well and improves the design realism with little user effort. Other geometric constraints such as diameter, length, and span limits were also found to be useful for some design problems.

9 REFERENCES

[1] Wright, W., Wright, O., and McFarland, M. (ed.), ***The Papers of Wilbur and Orville Wright***, McGraw-Hill Co., N.Y., N.Y., 2001

[2] Ashley, H., **On Making Things the Best-Aeronautical Uses of Optimization**, AIAA Paper No. 81-1738 (Wright Brothers Lecture), August 1981

[3] Sobieszczanski-Sobieski, J., **Multidisciplinary Design Optimization: An Emerging New Engineering Discipline**, *Advances in Structural Optimization* (483-496), Kluwer Academic Publishers, the Netherlands, 1995

[4] Raymer, D., ***RDS-Professional Users Manual***, Conceptual Research Corporation, Sylmar, CA, 2000

[5] Raymer, D., **RDS: A PC-Based Aircraft Design, Sizing, and Performance System,** AIAA Paper 92-4226, Aug. 1992

[6] Raymer, D., **Aircraft Aerodynamic Analysis on a Personal Computer,** SAE Paper 932530, Aerotech 93, Sept 27, 1993

[7] Raymer, D., & Burnside Clapp, M., **Pioneer Rocketplane Conceptual Design Study**, *AIAA Journal of Aircraft* (summer 2002.)

[8] Abbott, I., and von Doenhoff, A., ***Theory of Wing Sections***, McGraw-Hill, New York, 1949

[9] Raymer, D., **Use Of Net Design Volume To Improve Optimization Realism**, AIAA Paper 2001-5246, 1st AIAA ATIO Forum, Los Angeles, CA, Oct. 2001