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
Kungliga Tekniska Högskolan
Royal Institute of Technology
SE-100 44 Stockholm
Sweden

ENHANCING AIRCRAFT CONCEPTUAL DESIGN USING MULTIDISCIPLINARY OPTIMIZATION

ISBN 91-7283-259-2

"Young man, in mathematics you don't understand things,
you just get used to them."

John von Neumann.

ACKNOWLEDGEMENTS

The author would like to thank Dr. William Crossley of Purdue University and Dr. Jaraslaw Sobieski of NASA-Langley for valuable information, advice, and review. Also, thanks to Ron Mairs, Douglas Robinson, William Axtell, Herschel Owens, and Harry Scott of North American Aviation (Rockwell, now Boeing) for on-the-job instruction in classical design optimization methods, many years ago.

Special thanks are given to Dr. Arthur Rizzi, thesis advisor and instigator of my pursuit of this research at KTH. Thanks also to Hannes Ross (EADS/DASA) for serving as the thesis review “Opponent”, and to Drs. Torsten Kjellberg (KTH), Jan-Gunnar Persson (KTH), Petter Krus (University of Linköping), and Billy Fredriksson (SAAB), the members of the thesis review committee.

Appreciation is extended to Conceptual Research Corporation for providing the funding and travel budget to support this research, and to the Swedish Royal Institute of Technology (KTH) for use of its facilities and for the kindness and friendship shown to the author by the faculty, staff, and students during residency there. Finally, thanks to Dr. David Raymer for critical text review and editorial suggestions.

DEDICATION

This work is dedicated to all those who taught me – and the list grows each day.

(This page intentionally blank)

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.

NOMENCLATURE

A	= Aspect Ratio ($span^2$/reference area, applied to wings and tails)
APU	= Auxiliary Power Unit
Breguet Range Equation	= Classical range calculation method (see Raymer[11])
C	= Specific Fuel Consumption
C_L	= Wing Lift Coefficient
$C_{L\text{-}design}$	= Wing Design Lift Coefficient (used to optimize camber, twist, & airfoil)
CAD	= Computer-Aided Design
CER	= Cost Estimating Relationship
CFD	= Computational Fluid Dynamics
DAPCA	= Development and Procurement Cost of Aircraft (cost model)
DATCOM	= Data Compendium (USAF handbook of classical aerodynamic analysis)
f	= fuselage fineness ratio=length/diameter
FAR	= Federal Aviation Regulations (USA equivalent of JAR)
fineness ratio	=Length/diameter (usually of fuselage)
GA	= Genetic Algorithm
JAR	= Joint Aviation Requirements (European equivalent of FAR)
kgsm	= kg/square meter
KTH	= Kungliga Tekniska Högskolan (Swedish Royal Institute of Technology)
L/D	= Lift-to-Drag Ratio
LE	=Leading Edge (wing or tail)
M	= Mach Number
Mcr	= Critical Mach Number
Mdd	= Drag Divergent Mach Number
MDO	= Multidisciplinary Optimization
MOM	= Measure of Merit (Objective Function in optimization)
NDV	= Net Design Volume
O&S	= Operations and Support
OSD	= Orthogonal Steepest Descent
P_s	= Specific Excess Power
psf	= pounds per square foot
P/W	= Power-to-weight ratio of aircraft (engine power/W_o)
r	= asymptotic convergence rate
RDS	= Aircraft design software package *("Raymer's Design System")*
RS	= Response Surface
SFC	= Specific Fuel Consumption
t/c	= Airfoil thickness/chord length
TE	=Trailing Edge (wing or tail)
TOGW	= Aircraft Takeoff Gross Weight
T/W	= Thrust-to-weight ratio
W/S	= Wing loading (weight/area)
W_e	= Aircraft Empty Weight
W_o	= Aircraft Takeoff Gross Weight
UAV	= Unmanned or Uninhabited Aerial Vehicle

(This page intentionally blank)

TABLE OF CONTENTS

(This page intentionally blank)

LIST OF FIGURES

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. Researchers are applying a number of these techniques to aircraft vehicle design with intriguing results (Macmillin[4], Blasi[5], Raymer[6]), and MDO has begun to find application to "real" design projects at major aircraft companies (Cassidy[34]). In the future, MDO promises to be a key part of the aircraft designer's toolbox.

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.

For example, in the real world of aircraft range calculation one does not simply employ the Breguet Range Equation based on a selected cruise speed. Instead, for each design variation one must usually determine optimal speed and altitude with consideration for factors such as continuous power limits, engine overspeed limits, stall margins, air traffic regulations, temperature limits, and the actual, non-simple variations of lift, drag, thrust, and fuel consumption with speed and altitude. Once the desired speed and altitude are found, the range must be calculated taking into account the fact that the Breguet Range Equation is only valid over a small change in aircraft weight (the usual assumption that the use of the cruise-climb technique fixes this problem is not completely correct because, in the act of climbing, the *L/D* and fuel consumption actually change by some small amount).

In this author's opinion, this complexity in real-world calculations reduces the realism of methods based on defining governing equations and taking mathematical derivatives of those equations. One simply cannot develop closed-form equations that include all the real-world factors. Warn other authors, "We do not advocate over-simplification or distortion in formulation simply in order to be able to solve the eventual optimization problem more easily" (Gill, Murray, & Wright[7]). Note that methods using numerically generated gradients rather than mathematical derivative calculations do not limit analysis complexity, but they suffer from numerical "noise" generated during repeated function calls.

As will be described below, there are distinct phases of aircraft design, and there are many perceptions as to what aspects of an aircraft design can and should be subjected to an MDO study. The present research focuses solely on the earliest phase of aircraft design, namely conceptual design, in which the broadest design features are being

---

[†] The terminology *zeroth-order* as applied to optimization methods should not be confused with the quality of the analysis methods, which some authors break into quality levels called *first order methods*, *second order methods*, etc… This author does not employ such terminology, preferring to encourage the use of the best quality methods that can be employed in an amount of time reasonable for each phase of the design process. This changes as the years go by and new methods are developed, new codes are offered, and new computers get better. In any case, the term *zeroth-order* should not somehow imply that these methods are even worse than *first-order* methods! The *zeroth-order* methods for optimization can be employed with so-called *first-order* analysis methods, or with so-called *second-order* analysis methods equivalent to those used herein, or with highly sophisticated methods such as CFD and structural FEM.

determined such as number and size of engines, wing area and planform shape, and fuselage length and arrangement. This is done in the context of a search among multiple configurations and alternative design concepts, so the analysis and optimization methods to be employed cannot have excessive set up time, nor requirements for highly complex geometric inputs. The designers simply don't have the time to do a full, detailed study of each proposed alternative, nor do they have enough time to employ analysis and optimization methods with a time-consuming set up process.

Nor does the industrial design environment lend itself to methodologies and programs requiring substantial project-dependent mathematical or procedural development. Any approach requiring, say, development and differentiation of governing equations specific to a certain design would probably be of little practical use, no matter how "theoretically" desirable it may be.

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[6].

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. In the literature of the field, this link between the computer optimization and the "real world" is often dismissed with a reference to parametric formulas in textbooks such as Nicolai[8], Roskam[9], Stinton[10], or Raymer[11].

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. This was commonly done in the huge sizing optimization codes of the major aircraft companies[12], but the set up time of nearly a month precludes use of such methods for many design efforts. Furthermore, these are expert programs requiring extensive training and continuous experience to obtain reliable results. Research reported herein addresses this issue with a postulated set of automatic redesign procedures.

To be realistic, though, *no* set of routines can ever hope to insert *complete* design realism into an MDO method. We cannot even anticipate all real-world problems with a team of experts directly involved in the design process. Boeing didn't learn of a flutter problem on the SST program until well into fabrication, and the only solution available was a weight increase totaling thousands of pounds. Any previous optimizations done without this information were therefore totally invalid.

As all designers know, good information about a design only comes late in the design process, when it is often too late to fix things in an optimal fashion. Early in the design process it is easy to make changes, but the information about the design is incomplete and sometimes incorrect. Sobieski[13] words this well - "as the design process advances the amount of knowledge about the object of design asymptotically tends to 100%, while the freedom to act of that knowledge asymptotically reduces to 0%!" Greater use of high-end analysis tools earlier in the design process should improve this, but is beyond the scope of this research.

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.

(This page intentionally blank)

2 BACKGROUND

2.1 Aircraft Design Optimization – Purpose and Importance

During the development of a new aircraft concept, the optimization of the design to provide the desired capabilities at a minimum cost is of paramount importance. Aircraft are incredibly expensive compared to almost any other single man-made item. A new four-seat aircraft costs an order of magnitude more than a normal four-seat automobile. A large commercial airliner costs roughly half a million dollars per passenger seat, or, looked at from another perspective, approximates the cost of a major new high-rise office building. The latest manned bomber, depending on how the accounting is done, costs on the order of a billion dollars per copy.

These per-plane costs, large though they may be, pale beside the development costs of a new aircraft. The recently-let System Design & Development contract for the F-35 Joint Strike fighter is officially given as $19 billion[14], and that does not include the billions spent on closely related technology developments in recent years. Citation X, the new business jet from Cessna, cost about a quarter of a billion dollars of company money to develop, and took a total of five years before the aircraft was certified at which point Cessna could begin to sell the aircraft and hope to recover their investment[15]. For small general-aviation aircraft the development costs are so large compared to the expected profits that many designs currently in production are over 40 years old. The development costs for a new commercial airliner are so great that the would-be producers literally "bet the company" for a major new start. Some companies have lost that bet, and are no more.

From the operator's point of view, the purchase price (which includes development costs) is large but is not usually the largest cost of owning and operating the aircraft. Fuel, maintenance, and crew costs over the expected life of a typical aircraft will dwarf the amortized purchase price. In the world of commercial aviation, a small percent improvement in operating costs will make a large effect on overall airline profitability because the profit is determined as a large expense subtracted from a hopefully-larger income.

US Congressional Budget Office analysis[16] of USAF data indicates that the total operating cost of major weapons systems averages over twenty times the capital value (equivalent to re-purchase cost). Their definition of operating cost is a broad one, and includes fuel, personnel, maintenance, spares, and operation costs of the associated air bases – all the expenses required to keep the aircraft operational.

So, aircraft are prodigiously expensive, and any approach that can reduce those outlays will be appreciated by the developers and their customers. Specific technologies that can reduce cost, such as improved engines, lightweight structures, and advanced control systems, are funded for development with just that in mind and are used just as soon as the developers and customers believe that the projected savings outweighs the potential risks.

However, for any selected suite of technologies the cost of a new aircraft can be further reduced simply by an improvement in the design process. There are a variety of such improvements that can be made, and such improvements have been pursued for as long as aircraft have been designed.

One improvement with great payoff has been, and remains, in the area of the actual layout of the design. In the early 1940's a mathematical lofting method based upon the application of conic curves was developed at North American Aviation. This so improved the design definition that the first aircraft developed with conics, the P-51 Mustang, proved to have lower drag than other, similar technology designs, even accounting for the other technological advances of the P-51[11].

More recently, the development of Computer-Aided Design (CAD) has improved the actual design layout process in all phases of aircraft design. CAD has especially improved the interface between design and fabrication, through better product definition and through computerized numerical control (CNC) machining directly from this digital product definition[17]. Newer aircraft including B-2, F-22, X-35, Eurofighter Typhoon, SAAB Gripen, Dassault Raphael, and Beech Premier have all benefited in both cost and quality from the application of CAD and CNC.

These examples of design process-driven savings are independent of the application of any specific new technologies to the actual aircraft concept, such as an improved material or engine.

Optimization methods, the subject of this research, present another area in which improvements to the design process can provide substantial savings in cost independent of the application of new technologies. Aircraft are designed to specific roles and missions, and an improvement in the design process that allows the aircraft to be better tailored to its intended usage will reduce cost by the elimination of unneeded "extra" capabilities. For example, if a new aircraft design has a takeoff distance that is substantially less than that required for operation at its intended airports, that "extra" capability provides no additional utility to the operator yet carries a price – probably in an overly-large wing and/or engine. An improved design process that would identify this early and allow the designers to drive out the excess capability would directly save cost.

Another way that an improved design process can reduce aircraft cost is in the early identification of the best possible balance between the disparate desires of the various design disciplines. The aerodynamics department generally prefers a thinner wing to reduce drag, whereas the structures department prefers a thicker one to reduce weight. Identification of the best balance must be done in the context of the aircraft's roles and missions, and has the potential for a substantial overall cost savings.

Specific aircraft optimization methods are discussed below, and span a spectrum from simple one-variable parametric trades or even closed-form solutions, to highly sophisticated, mathematically based multivariable/multidisciplinary optimization methods. Properly applied, aircraft design optimization offers reduced cost in all phases of design. It is the specific intention of this research to improve the aircraft conceptual design process by the improved usage of such sophisticated optimization techniques.

2.2 Outline of Aircraft Design Process‡

The actual process of aircraft design must drive the selection of the methods of optimization as well as the assumptions and constraints to be observed. Aircraft design is, by its very nature, multidisciplinary, and the notion that there is something new about the development of optimizations that consider multiple disciplines is viewed with wry amusement by those experienced in aircraft conceptual design. In this field, optimizations have *always* included aerodynamics, structures, propulsion, controls, systems, and a host of other disciplines. However, emerging MDO techniques provide a more-formalized structure to the design optimization process and allow better management of the large number of trades necessary to find the optimum design.

Aircraft design can be broken into three major phases, namely *Conceptual Design, Preliminary Design*, and *Detail Design*. Each of these has different tasks and objectives, and the design process in each phase is quite unique from other phases. Thus, the tools to be employed differ, and even the people involved are usually different (at least in the big companies). These three phases of design are depicted in figure 1. Conceptual and early Preliminary Design are the focus of this research.

In *Conceptual Design*, the basic questions of configuration arrangement, size, weight, and performance are answered. Numerous alternative design concepts are prepared in response to the design requirements, and numerous variations on those concepts are also studied. All design options are "fair game", and the design space extends as far as the designers' imaginations.

In conceptual design, the design requirements are used to guide and evaluate the development of the overall aircraft configuration arrangement. A mathematical process called "sizing" is used to calculate what the aircraft takeoff gross weight, empty weight, and fuel weight must be for the design to reach the range as specified in the design requirements. This calculated weight is used as the starting point in making a design arrangement drawing, determining the overall size, wing and tail area, required fuel tank volume, and many other aspects of the design.

‡ Portions of this section excerpted and edited from Raymer, *AIRCRAFT DESIGN: A Conceptual Approach,* 1999[11]. For permission to copy contact the author.

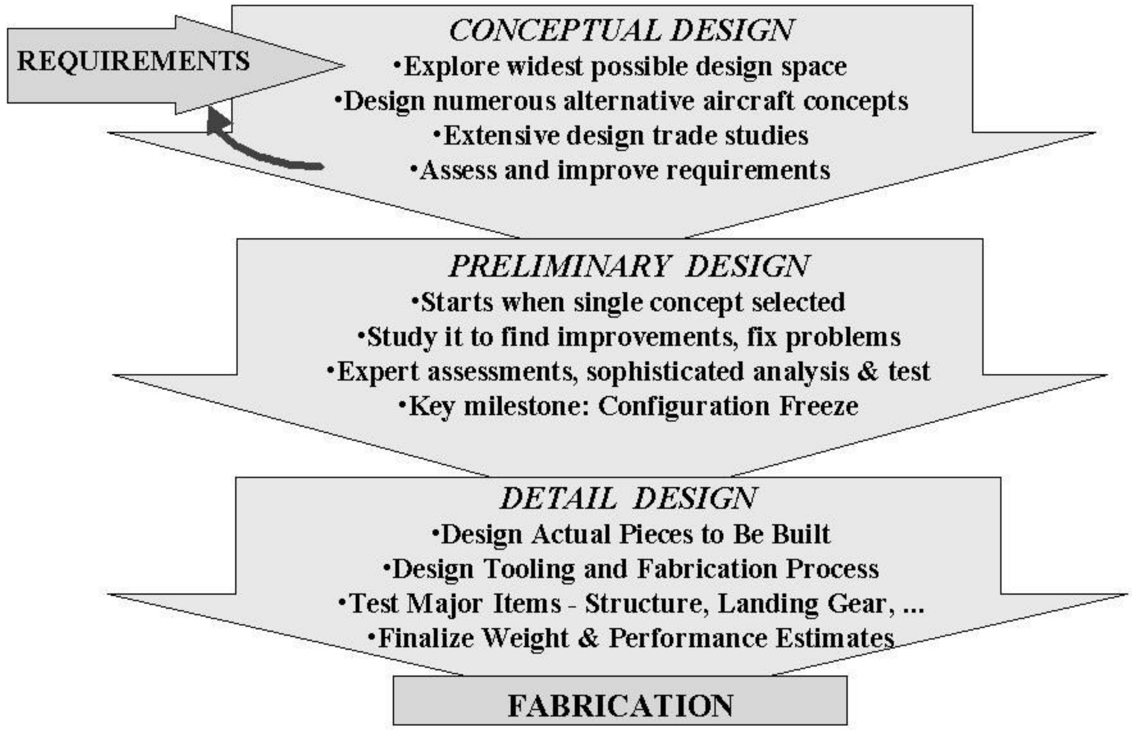


figure 1. Three Phases of Aircraft Design

Calculated aircraft weight is commonly used as the measure of merit (MOM) in aircraft design optimizations, so the implementation of a reliable procedure for calculating it is critical to any optimization result. If cost is used as the MOM, the calculated weight is a key input to the cost calculation so again, this sizing calculation is critical.

This design arrangement includes wing and tail overall geometry (areas, sweeps, etc.), fuselage shape and internal locations of crew, payload, passengers, and equipment, engine installation, landing gear, and other design features. The level of detail in configuration design is not very deep, but the interactions among all the different components are so crucial that it requires years of experience to create a good conceptual design.

This initial layout is analyzed to determine if it will perform the design mission. Aerodynamics, weights, and installed propulsion characteristics are analyzed and subsequently used to do a detailed sizing calculation. Furthermore, the performance capabilities of the design are calculated and compared to the design requirements.

A key aspect of conceptual design is that it is a very fluid process, and the design layout is always being changed, both to incorporate new things learned about the design and to evaluate potential improvements to the design. Trade studies and an ever-increasing level of analysis sophistication cause the design to evolve on almost a week-by-week basis, and changes can be made in every aspect of the design including wing geometry, tail arrangement, and even the number of engines. Furthermore, during conceptual design a number of alternative designs are studied to determine which design approach is preferred.

Optimization techniques are used to find the lightest or lowest-cost aircraft that will both perform the design mission and meet all performance requirements. The results of this optimization include a better estimate of the required total weight and fuel weight to meet the mission. The results also include required revisions to the engine and wing sizes. It is this optimization during conceptual design that is the key focus of the research described herein.

Optimization methods during conceptual design focus on the overall design characteristics rather than finer details of the concept. Typical parameters being optimized include thrust to weight ratio, wing loading, wing aspect ratio and sweep, and fuselage fineness ratio. Design aspects such as the exact airfoil shapes are generally not included in optimizations at the conceptual design level because the extra time involved is better spent looking at more design concepts and gross design alternatives. Instead, one can optimize the overall wing design lift coefficient ($C_{L\text{-}design}$) and have that optimum value used as the input to wing design during later efforts.

Due to the fluid nature of the conceptual design process and the rapidity with which the design changes and progresses, it is generally concluded that sophisticated analysis tools such as CFD and structural FEM are inappropriate at this stage due to the expense, but more-importantly, due to the time involved (see Jameson[18]). A single Reynolds-averaged Navier-Stokes analysis for a reasonably complicated aircraft configuration only takes a few hours with today's computers[19]. This does not count the weeks of set up time to develop a workable analysis grid. While conceptual design is usually done on the same high-end CAD systems used in later phases of design, the fact that the designers are developing and assessing a large number of alternative configurations prevents them from spending enough time on each design to assure even zeroth-order continuity between all parts and panels, let alone first or second-order continuity as required by CFD.

Instead, robust classical methods such as panel codes and the DATCOM[20] are used for aerodynamics calculations, and structural weights are estimated by time-proven statistical equations. In the aerodynamic analysis, even if CFD could be applied from the first design layout it would still be inappropriate because the actual airfoil shapes and the distributions of twist and camber have not yet been properly determined. That will not occur until the next phase of design, so in conceptual design we need a tool that will, in the words of other researchers, "predict the wing drag which the detailed aerodynamic design will achieve but even *before* the detailed aerodynamic design is started[21]."

One widely used technique to predict aerodynamic performance prior to final wing design is the semi-empirical method of *Percent Leading-Edge Suction* (see Raymer[11]). This provides a reasonable estimate of the aerodynamic characteristics of a chosen wing planform as they will be *after* the aerodynamics department has completed its optimization effort during the later *Preliminary Design* effort. Another method involves the a-priori use of a sophisticated analysis in a parametric fashion. For the development of the Airbus A-380, a mathematical surface was fit to a limited number of detailed evaluations to create such a tool[21].

Another unique aspect of conceptual design is that it involves design alternatives as well as improvements to a specific design concept. This extends to features of the design such as number of engines, number of seats across a row, and type of tail. These are *discrete*, or *integer* variables as opposed to the continuous variables that predominate later stages of aircraft design. Inclusion of these in design optimization methods is further discussed below.

The area of Conceptual Design has received relatively little attention from the MDO development community. In a recent survey paper sponsored by the AIAA MDO Technical Committee summarizing industry MDO applications, Giesing and Barthelemy[22] identified *no* applications in conceptual design. Less than half the applications identified were in early preliminary design.

Preliminary Design can be said to begin when the major changes are over. The big questions such as whether to use a canard or an aft tail have been resolved. The configuration arrangement can be expected to remain about as shown on current drawings, although evolutionary revisions will occur. Preliminary design is characterized by a maturation of the selected design approach. The design evolves over a period of many months, with an ever-increasing level of understanding of the design, an ever-increasing level of design and analysis detail, and an ever-increasing level of confidence that the design will work.

During Preliminary Design, the one selected aircraft design concept is subjected to a continued refinement and optimization. Early Preliminary Design resembles the optimizations of Conceptual Design, with analysis and optimization tools being applied to revise and improve the design layout in an iterative process. Since only one design is being studied in Preliminary Design, it is possible to apply more costly, time-consuming methods such as CFD, structural FEM, wind tunnel testing, six-degree-of-freedom flight simulation, and sophisticated weight and cost analysis tools. These are phased in as the design matures enough that large changes to the configuration become less likely.

During early Preliminary Design, optimization continues on top-level parameters such as thrust to weight ratio, wing loading, wing aspect ratio and sweep, and fuselage fineness ratio. As the design progresses and major revisions become less likely, optimization proceeds towards finer design aspects such as the exact airfoil shapes and the distribution of twist and camber, or the best shape for the fuselage to promote laminar flow. This is often done by defining *shape functions* - geometric equations which control the shape and are themselves controlled by parametric inputs. Alternatively, aerodynamic optimization can be done by specifying desired pressure distributions and searching for a shape that will produce it (see Jameson[18]). By this phase of the design process, the top-level parameters mentioned above are locked in and will not be further changed or optimized unless major problems are uncovered.

Also during Preliminary Design, specialists expert in the various design disciplines and aircraft subsystems are given the overall design concept and asked to evaluate it and to refine the design in their area of expertise. They commonly find areas in which they request design modifications, requiring further iterations and refinements of the design concept. Following such revisions, the design optimizations must be redone because any change to the design layout will likely affect the inputs, and hence the outputs of an optimization.

Assuming a favorable decision for entering full-scale development, the *Detail Design* phase begins in which the actual pieces to be fabricated are designed. This last part of the design process is characterized by a large number of designers preparing detailed drawings or CAD files with actual fabrication geometries and dimensions. Thousands of little pieces not considered during conceptual and preliminary design must be designed during the detail design phase[§]. These include flap tracks, brackets, structural clips, doors, avionics racks, and similar components. Every single piece of the aircraft's structure and its hydraulic, electrical, pneumatic, fuel, and other systems must be designed in detail – hence the name.

Optimization in Detail Design tends to be subsystem or part specific, not system-wide. Design procedures for structural parts, equipment, wiring, and other areas typically include the minimization of weight of those items, but not tradeoffs with other parts or systems. Such tradeoffs should have been accomplished during Conceptual and Preliminary Design.

As the design progresses through conceptual, preliminary, and detail design, the level of detail of the design steadily increases. This is illustrated in figure 2 for a typical piece of aircraft geometry, the front wing spar. The top of figure 2 depicts the design of a front wing spar in the amount of detail typical of conceptual design, usually nothing more than a straight line in top view at the desired location of the spar. The spar is assumed to be approximately the depth of the wing.

While this seems crude, keep in mind that the entire aircraft arrangement is being determined at this stage of design, and the interactions between components are more important than the exact geometry of any one part. This simple definition answers the key questions for the initial conceptual layout: How big can the wing box, wing fuel tank, and leading-edge flaps be? If this front spar is moved forward then the wing box gets larger, reducing structural weight, but the leading-edge flaps get smaller, reducing their lift contribution and requiring a bigger wing to meet, say, a landing distance. The designer

---

[§] In addition to the thousands of pieces that need to be designed, modern high-end aircraft also require the development of millions of lines of computer code, with a large impact on development cost. An interesting study beyond the current scope would be an optimization that includes an estimate of additional software costs as design features are considered that require unique coding, such as the use of relaxed stability or integrated propulsion controls.

must attempt to trade off these conflicting desires and find a reasonable compromise for spar location.

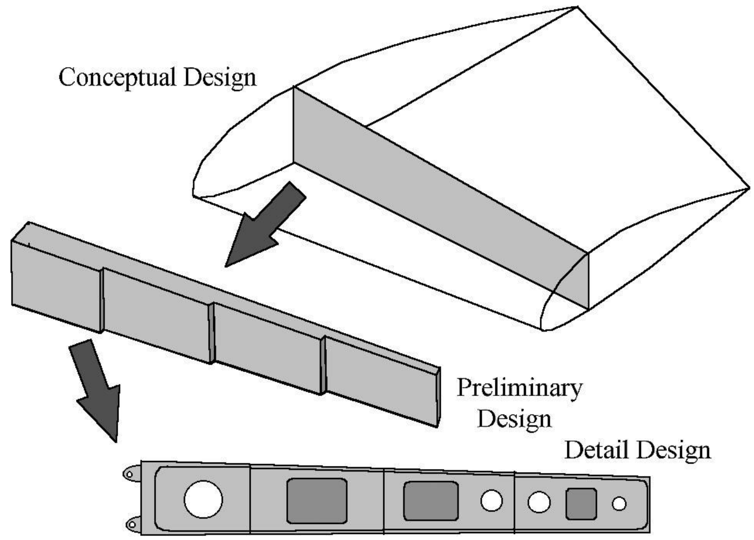


figure 2. Wing Spar as Defined in Conceptual, Preliminary, and Detail Design

In preliminary design, the example wing spar's overall geometry is refined, including the actual shaping of the spar's cross section (middle of figure 2). Fairly sophisticated methods are used to perform a structural analysis of the overall spar, with the objective of determining the thickness (or number of composite plies) required to handle the expected loads. The spar is only one element of the overall structure of the aircraft that will be defined in preliminary design, and extensive analysis will be done of the whole structural concept to assess and optimize the overall concept.

Note that the spar design in the preliminary design phase is still not in sufficient detail to be built. Full consideration has not yet been given to attachments, cutouts, access panels, flanges, manufacturing limitations, fuel sealing, and other real-world details. These are the subject of detail design (bottom of figure 2), and are typically considered only after the aircraft structural concept as a whole has been validated during the preliminary design phase.

Towards the end of detail design, the design drawings are of sufficient depth to determine the part weights from the volume of various materials as indicated on the drawings. This is the first time weight can be assessed by direct calculation – in conceptual and preliminary design, statistical methods must be employed.

For a detailed review of the aircraft design process, see Raymer[11] chapter three.

2.3 Classical Aircraft Optimization Methods**

An informal review of aircraft design textbooks and NACA reports from the 1920's through the 1940's did not find any mention of an aircraft design optimization method other than general advice which can be paraphrased as "hold down the weight, clean up the drag, and increase the horsepower." Planform geometry and powerplant size were picked largely by reliance upon prior successful designs with some modest extrapolation as better engines became available. Both Prandtl wing theory and the Breguet range equation were known by this time, but other than the obvious conclusion favoring elliptical lift distributions, no one seems to have attempted a determination of a combination of design parameters that would maximize range for a specific aircraft concept (or minimize weight and cost). Even as late as 1960, a widely used aircraft design textbook stated that "to obtain the optimum combination, the only solution is to design a series of three or four airplanes with different combinations and choose the one with the lowest direct operating cost" (Corning[23]).

This points to a methodology dichotomy that is still evident today – that between "equational" and "parametric" approaches. Equational approaches involve efforts to write meaningful governing equations and solve them mathematically or procedurally. In 1933 Prandtl[24] included wing weight effects in an optimization of spanwise lift distribution, yielding a greater loading towards the root than in his own classic elliptical aerodynamic optimization. Göthert[25] in 1939 developed analytical methods to optimize a wing, using span and area as variables. Typical modern analytical optimizations based on derivatives of governing equations can be found in Torenbeek[26] among others.

However, most industry applications of optimization have relied heavily on the parametric approach. These were well described in a 1970's design textbook (Nicolai[8]), but were in use in industry at least as early as the 1950s and probably before based on anecdotal information relayed to this author by design "old-timers".

In parametric optimization, the selected design parameters such as wing sweep or aspect ratio are varied about the baseline values as seen on the design layout. Estimates are made as to the impact of those variations on the design layout, either by actually having a designer redraw the aircraft for each variation or by applying some selected procedures for automated redesign. These attempt to determine the impacts of parametric design variations without a man-in-the-loop drawing revision. Then, the design is re-analyzed and re-sized, and all performance and cost estimates are recalculated. From this data, an optimum is found using methods ranging from a simple single-variable graph to the sophisticated MDO techniques described herein.

Classical aircraft optimization usually employs the "carpet plot" technique to display the results of the parametric calculations and to solve for the optimum aircraft meeting all performance constraints, as shown in figure 3.

** Portions of this section excerpted and edited from Raymer, *AIRCRAFT DESIGN: A Conceptual Approach,* 1999[11]. For permission to copy contact the author.

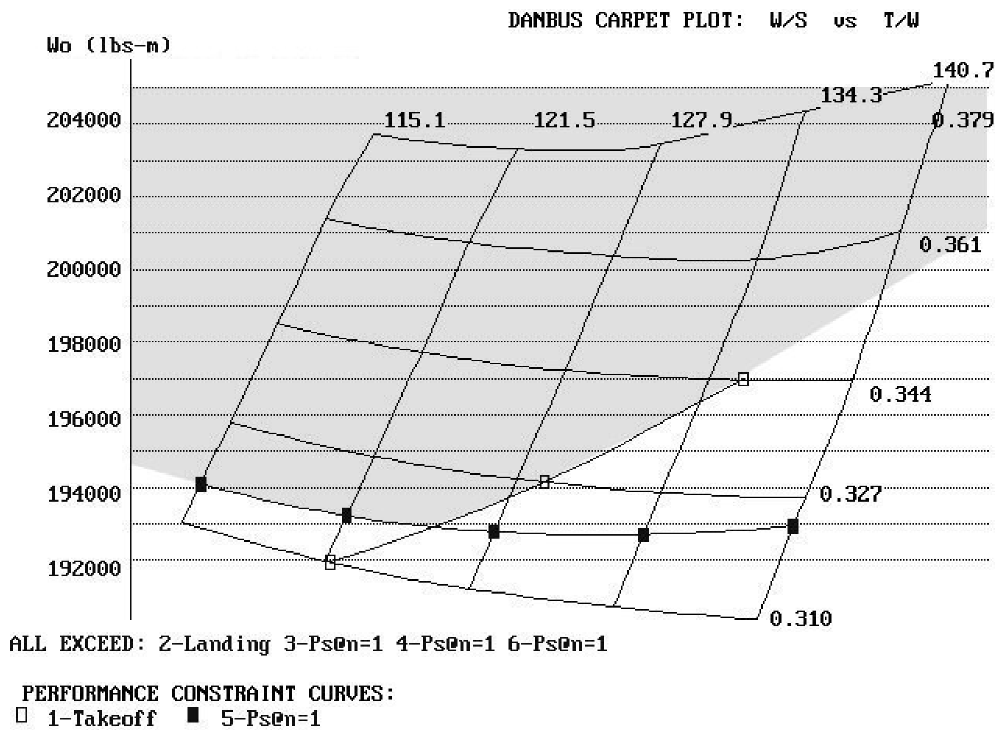


figure 3. Classical Optimization via Carpet Plot

The carpet plot of figure 3 represents the commercial airliner sample design described in Section 6.1.2. The chosen parametric variables, thrust-to-weight ratio *(T/W)* and wing loading *(W/S)* are arbitrarily varied from the as-drawn baseline values by some percentage. Each combination of *T/W* and *W/S* produces a different airplane, with different aerodynamics, propulsion, and weights. These different airplanes are separately sized to determine the takeoff weight of each to perform the design mission. If cost is to be the measure of merit, it is calculated from the weight. The different airplanes are also individually analyzed for performance. If the *T/W* and *W/S* variations are wide enough, at least one of the aircraft will meet all performance requirements although it will probably be the heaviest airplane when sized to perform the mission.

Graphical interpolation techniques (see Raymer[11]) are applied to produce the "carpet" lines, showing the effect on measure of merit for these parametric variations. Then the performance constraint lines are superimposed, showing what combinations of those design variables provide the required performance values.

In the sample of figure 3 the original baseline design layout is at the middle of the crossing "carpet" lines, and exceeds all performance constraints as indicated by the various labeled diagonal lines. The "optimum" solution according to this plot is found at the lowest point (best MOM) where all requirements are met or exceeded, and typically occurs where two or more of them cross (here at about 193,000 lbs {8,7543 kg}).

At the optimum point it is possible to move in a direction of lower aircraft weight (better measure of merit) only by violating the performance constraint lines. This epitomizes the definition of the Kuhn-Tucker Theorem as described below, which is used in the development and proof of MDO methods.

This carpet plot technique directly allows only two parametric variables, but designers have long used crossplots of numerous carpet plots to optimize for four and more variables. It is common to use *T/W* and *W/S* as the main variables of an aircraft design optimization because collectively they have a large effect on sized takeoff gross weight and performance[††]. To add additional variables to the optimization, the classical optimization method selects additional variables such as wing Aspect Ratio and Sweep, and makes parametric variations of them. For each combination of those additional variables, a separate *T/W* - *W/S* carpet plot is created and used to find the optimal aircraft. Then, a carpet plot of the Aspect Ratio and Sweep variables is created in which every data point is a "best" combination of *T/W* and *W/S*.

Needless to say, the workload becomes immense especially when 1960's-era computers are used and the graphing is all done by hand. To optimize T/W, W/S, aspect ratio, taper ratio, sweep, and thickness (the basic set of six design parameters commonly used in aircraft conceptual design – see below, also see Raymer[27]) requires a minimum of 3^6, or 729 data points (5^6, or 15,625 data points would improve accuracy). Each data point represents a different airplane and requires full analysis for aerodynamics, propulsion, weights, sizing, and performance.

To better optimize an aircraft at the conceptual level, additional design parameters such as fuselage fineness ratio, wing design lift coefficient (or camber), and engine bypass ratio or propeller diameter could be included in a simultaneous optimization. One could attempt to simultaneously optimize all of these and many more, and also have the computer optimally change the actual shape of the design including wing planform breaks, nacelle locations, and tail locations, and perhaps optimize the airfoils and the APU installation at the same time.

Such "everything-optimization" is neither feasible nor desirable. As to feasibility, the example above indicates how quickly the number of required data points (i.e., aircraft parametric evaluations) spirals out of control as additional design variables are added. Some MDO researchers label this the "curse of dimensionality", and it leads to unacceptable execution times plus additional set up effort.

Nor is "everything-optimization" desirable. After a certain point, excessive time spent on defining, executing, and understanding an optimization computer program is just time taken away from other pressing design tasks. Such optimization would probably stretch beyond the applicability or sensitivity of the chosen measure of merit, which after all is

[††] Assuming a new engine will be developed for the design, thus allowing the designers full freedom to specify *T/W*. If an existing engine must be used, then the *T/W* will be defined by the available thrust and the sized aircraft weight and so cannot be used as a parametric design variable. In such cases the next-most-critical design parameter, usually wing Aspect Ratio, is used in the Carpet Plot

just an assumed approximation for how the aircraft will actually be operated. Also, the analysis may be rather insensitive to some selected design variables producing ambiguous or misleading results. Finally, many of the design parameters that could be included are not very interdependent ("weakly-coupled") and can be optimized separately.

But, if design parameters are carefully selected and evaluated with considerations for the "real world" of aircraft design, a truly better aircraft can be found in a reasonable amount of computational time. Such multivariable/multidisciplinary optimization methods are the focus of this research, and selection of design variables is discussed in a section below.

2.4 Historical Review of Engineering Optimization

Optimization involves the pursuit of the "best" – or a significant "better". Better what? A better value of some defined *"measure of merit"* or *"objective function"*. For aircraft conceptual design, the measure of merit is typically weight and/or cost for some specified capability, or capabilities such as range or payload at a specified weight or cost. This pursuit of better/best is limited by specified conditions involving real-world operational aspects or must-meet capabilities, which in mathematical terms are the *"constraints"* of the optimization. Fundamentally, we can define optimization as the determination of a minimum or maximum of one or more objective functions such that no constraints are violated. While equality constraints weigh heavily in other applications of optimization, in aircraft design optimization the constraints are almost always of the inequality sort – it is acceptable to be better than the required value, just don't be worse!

Optimization is nothing new – it is inherent in the laws of physics. A massive collection of particles, floating freely in space, will form a sphere that is the optimum shape for minimizing surface area for an enclosed volume. A ball rolling down a hill will automatically, under the direction of nothing more than the laws of gravity and motion, find the fastest way down from a given starting point. Pebbles will, over time, pack themselves into the smallest possible volume.

Human efforts at optimization go back as far as humans have existed. Even a primitive man tries to find a better way to kill prey, gather foods, carry water, defend loved ones, and provide shelter from the elements. In fact, optimization by a-priori thought rather than instinct is a key factor that makes us human (although some animals look pretty thoughtful at times – like a dog trying to get to an out-of-reach bone!).

Prior to the last few hundred years, optimization was largely by trial-and-error, with good results passed down as heuristic folklore. The great cathedrals of Europe were designed with every intent to minimize material (for cost) and column size and number (for aesthetics), but the only available tools were the study of prior successes and failures and the construction and test of portions of the design under consideration.

In the world of shipbuilding, quite close to the world of aircraft design, the disaster of the Swedish warship Vasa is instructive concerning the problems of attempting to optimize with insufficient analytical tools to assess the design constraints. Vasa was ordered during a time of war (1625) as a single-deck warship with a keel length of 108 ft and a width and ballast load suitable for such a length, based on prior experience. The customer - the king who was away fighting in Germany - sent an order to make the ship “more optimal” for its military purpose, namely by adding guns which required the ship to be longer (135 ft.) and to have an unplanned-for second gundeck and bigger sails.

Since there were no technical means to calculate the stability or ballast requirements except by past experience, nobody could prove that it *wouldn’t* work (i.e., the stability constraint could not be calculated to determine an upper limit on the design variables “number of guns” and “number of gundecks”). So, they built it that way rather than incur the delay needed to start over with a broader hull and more ballast space. When the hull was floated and the guns installed, they performed the usual stability test in which 30 men would run from side to side to see if the rolling motion would grow excessively. The Boatswain later said “If they had run across the ship one more time she would have capsized." Unfortunately, the king had sent clear instructions: "Vasa shall be ready by next (25 July), and if not, those responsible would be subject to His Majesty's disgrace." They finished it, launched it, and watched it roll over and sink in 100 ft of water[28].

Optimization by mathematical analysis became possible in the 1600’s when Isaac Newton and Gottfried Leibniz independently developed calculus. About the same time, Pierre de Fermat defined a general approach to compute local minimums and maximums of functions by solving for the derivative and setting it to zero – the basis of most analytical optimization today[40]. Fermat, along with Blaise Pascal, founded the theory of probability that is critical to Monte Carlo techniques and the recently developed evolutionary/genetic optimization algorithms. Interestingly enough, Fermat and Pascal became involved in probability theory when a gambler asked Pascal for advice as to how to best divide game winnings[29] - and even today game theory provides a powerful optimization tool.

In the 1700’s, Leonhard Euler developed methods to find the extreme values of functions, along with many other contributions to mathematics and physics including definition of a basic equation of hydrodynamics still used in computational aerodynamics. Joseph Lagrange, together with Euler, developed the calculus of variations. This remains highly useful in optimizing real-world problems such as those that are time-dependent. Lagrange also developed generalized equations of motion and developed the concept of partial differential equations, two of the foundations of engineering dynamic analysis.

In the early 1800’s, Adrien-Marie Legendre and Carl Friedrich Gauss developed the method of least-squares curve fit that is often used in optimization, especially the modern Response Surface method. Later Pierre Laplace developed a formal proof of the least-squares method, on which the estimation of curve fit errors is based. In the mid-1800’s, William Hamilton developed theorems concerning differential equations, dynamic analysis, and imaginary numbers which have great application for the solution of optimum design problems.

Andrei Markov in the early 1900’s developed the theory of stochastic processes and pioneered the study of what became known as Markov Chains. These are sequences of random variables in which the future value of the variable is determined by the present value but is independent of the way in which the present value was derived from its predecessors. In other words, a Markov Chain has no history and no after-effects[30], which is typically true of iterative optimization processes.

Vilfredo Pareto, an economist in the early 1900's, developed the principle of multiobjective optimization for use in allocation of economic resources. His concepts became known as "Pareto optimality", defined as a situation in which you cannot make someone better off without making someone else worse off. A graphical representation of Pareto optimality is widely used to depict two-objective optimality. An aircraft design example might be a requirements trade study in which you attempt to maximize both range and payload weight, and plot a curve showing the optimum tradeoff between the two (figure 4).

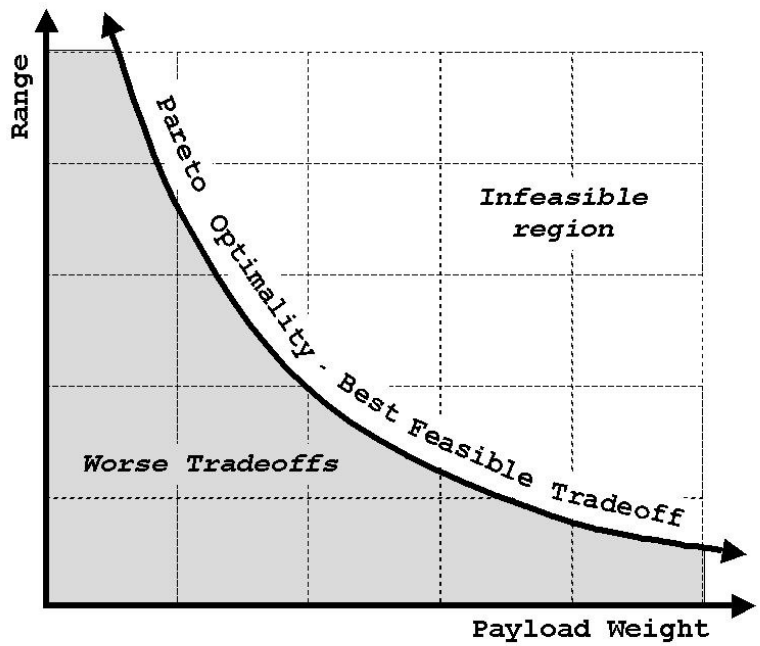


figure 4. Pareto Graph: Range-Payload Optimality

In 1947 George Dantzig developed the Simplex Method to optimize problems involving scheduling of training, supply and deployment of personnel for the U.S. Air Force. In the military terminology of the day such planning was known as "programming", and since the equations were linearized, this became known as "Linear Programming" (not to be confused with computer programming, which didn't exist at that time). A key aspect of linear programming is its ability to deal with constraint functions independent of the objective function. Linear programming has become widespread in its usage, especially for business decision making.

The Kuhn-Tucker Theorem (Albert Tucker and Harold Kuhn) of 1950 is considered to have launched the modern field of nonlinear programming (although it was apparently defined twice previously, by William Karush in 1939 and by Fritz John in 1948). Kuhn-Tucker gives necessary and sufficient conditions for the existence of an optimal solution to a nonlinear objective in the face of constraints. Fundamentally it says that at the optimum, the only direction you can move to improve the objective function is one that will violate one or more constraints. Kuhn-Tucker is widely used in the proofs of analytical optimization methods. As described above, the classic aircraft design carpet plot is actually an excellent illustration of Kuhn-Tucker (see figure 3).

2.5 Overview of Multidisciplinary Optimization (MDO)

Multidisciplinary Optimization, or MDO, can be described as a collection of mathematical techniques for multi*variable* optimization in which the optimization clearly crosses *disciplinary* boundaries. An essential feature of MDO is the presence of design constraints and measures of merit which are of system-level concern. In a typical aircraft conceptual design application, the measure of merit (MOM) is either cost or its surrogate, weight, where the aircraft is sized[‡‡] to some specified mission which includes the range and payload requirements. The design constraints are typically the aircraft's required performance values such as takeoff distance and climb rates, plus any geometric or operational constraints such as a wingspan limit.

MDO grew out of prior multivariable optimization methods as a natural consequence of the attempt to apply optimization to more system-level problems. To the cynic, MDO is just the new "buzzword" for multivariable optimization, but MDO should be recognized rather as a distinct system-level subset of multivariable optimization. Furthermore, the fairly recent application of MDO as a serious topic for research has led to the development of new and distinct optimization methods due precisely to that system-level focus.

A typical application for MDO in the aircraft design field is the simultaneous aerodynamic and structural optimization of a wing. The wing is defined in terms of some geometric variables, and the effects on aerodynamics and structural strength are determined as the geometry is varied. Results are assessed versus some defined measure of merit, and in the presence of constraints which can be based on performance, safety, operability, or practicality. Other applications of MDO in aerospace include such diverse areas such as launch vehicle geometric design, composite materials design, coupled wing-body integrated analysis, advanced structural weights estimation, and aerothermal and sizing optimizations[31,32,33]. Organizations such as Boeing's Phantom Works are using MDO techniques in aircraft conceptual design on a fairly routine basis, and report great success at quickly "weeding through" numerous design alternatives[34].

‡‡ "Sizing" is a mathematical iterative process which determines the aircraft takeoff gross weight, empty weight, and fuel weight required such that a given aircraft concept layout can perform a specified mission (range) at a specified payload weight. This calculated size is used to redraw the aircraft with a revised wing area, fuselage length, etc…, appropriate to the determined weight. Note that range is a given (independent variable) whereas aircraft weight (thus aircraft physical size) is the calculated, dependent variable. For a detailed description see Raymer[11] chapter three.

A wide variety of approaches are being used for defining and solving multidisciplinary optimization problems including the following, described below:

- Finite Difference
- Implicit Function Theorem
- Stepping Search Methods*
- Response Surface
- Monte Carlo*
- Random Walk and Simulated Annealing
- Evolutionary Algorithms and Evolution Strategy*
- Genetic Algorithms*
- Decomposition

(used in this research)*

In figure 5 the contours of the measure of merit of a two-variable design optimization problem are shown to illustrate the various optimization methods in the following sections. The sought-after global optimum point is indicated by a star.

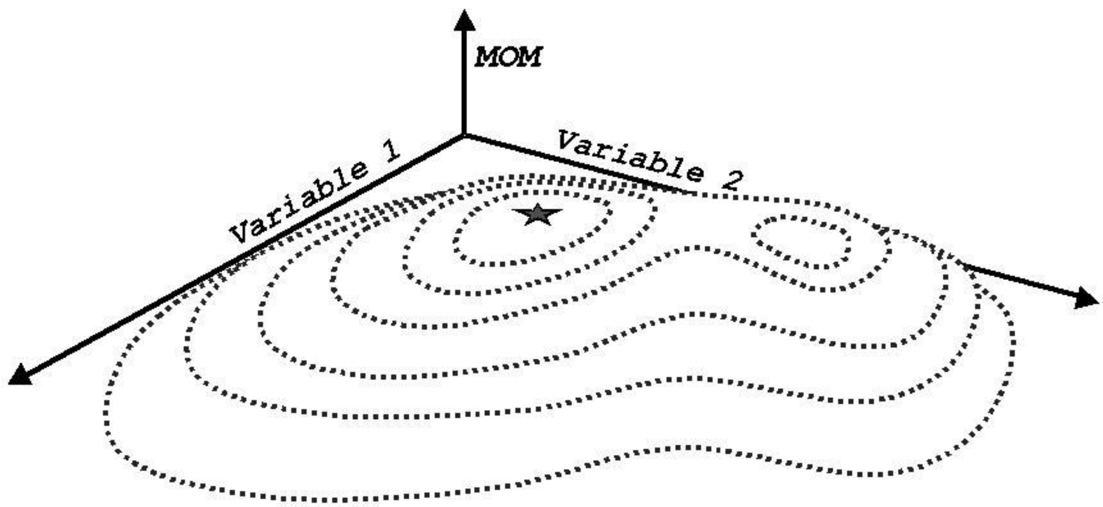


figure 5. Contours of MOM vs. Design Variables

Constraints are not shown in this figure, but would indicate infeasible regions in the design space. In most aircraft optimization problems, the best usable answer is not the unconstrained global best. Instead it is usually the point closest to the global best on one or more constraint boundaries, where one can get no closer without violating a constraint (as defined by the Kuhn-Tucker Theorem mentioned above).

2.5.1 Finite Difference

A widely used technique for multivariable or multidisciplinary optimization uses a *Finite Difference* approach. Small parametric changes are made to the system (aircraft) one at a time, and the change in the measure of merit is used to define a slope (first derivative) which represents the system response (sensitivity) to a change in that variable. These derivatives are then used to predict the optimum solution, and iteration is used to drive out the obvious linearization errors. Finite Difference methods were successfully used in conceptual design optimization of a new business jet as reported by Gallman et al.[35]

Some have reported that the calculation of the finite differences can be computationally expensive and may produce inaccurate gradient approximations (Newman, et al[36]). For each step of the iteration, the number of system analyses required is at least one more than the number of design variables. Also, the difference between the "real world" and the linear assumption of the method requires that the solution found for each iteration be kept fairly close to the values of the parameters used in determining the slopes, which increases the number of steps required and introduces error.

2.5.2 Implicit Function Theorem

The *Implicit Function Theorem* differentiates the various governing equations to obtain sensitivity equations. These are used to set up simultaneous linear algebraic equations, which are then solved for derivatives of the objective function as the design variables are changed. These are used to find a solution.

Unfortunately, the governing equations of a real aircraft design problem are both complicated and design-specific. In the real world, design analysis is limited by such factors as stall margin, service ceiling, max-continuous throttle setting, FAR/JAR-restricted climb profiles, and peculiarities of the selected engine's thrust and SFC curves. These do not lend themselves to an equational representation, especially considering that there are often discontinuities in the derivatives of the real-world data.

2.5.3 Stepping Searches

There are a variety of methods that can be labeled *Stepping Searches*, in which the objective function is evaluated and a decision is made as to what direction to move to find a better value of the objective function without violating any constraints. This direction of maximum local improvement to the objective function can be found using derivatives of governing equations or finite difference methods based on actual calculations of the measure of merit, and is called the direction of *Steepest Descent*. After determining the best direction to move, a "step" of some defined distance is made in that direction to a point which becomes the origin for the next calculation and step. This is shown in figure 6, starting from the point labeled (1).

Steepest descent is computationally intensive especially when constraints are used. The "best" direction must be found in a direction that doesn't violate the constraints, requiring a large number of evaluations.

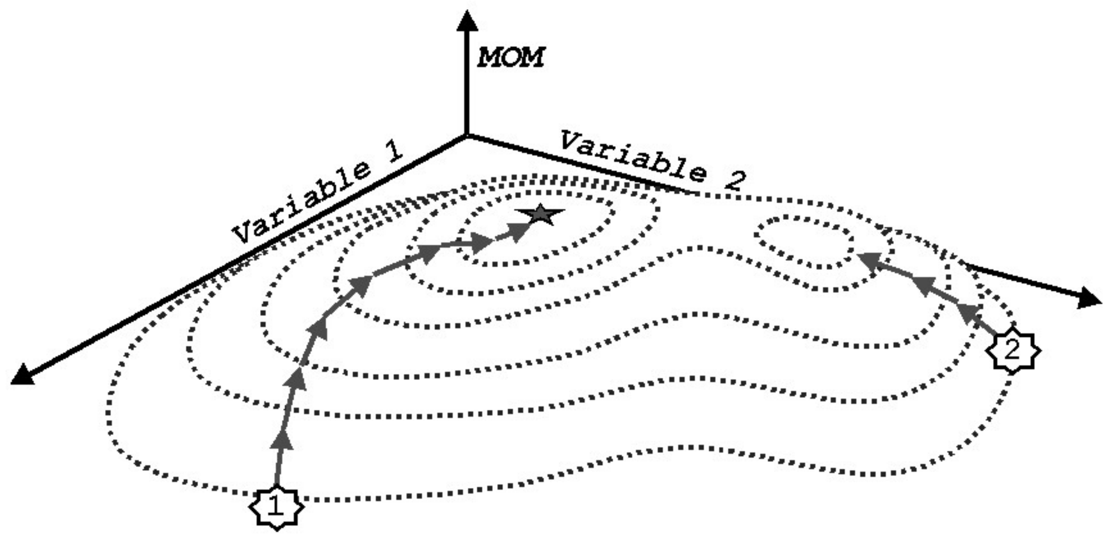


figure 6. Stepping Search

Stepping searches are prone to finding a local optimum rather than the global “best”, depending on the location of the starting point. This can be seen in figure 6, starting at the point labeled (2). One way to avoid this is called *Multi-start*, in which the optimization is rerun a number of times from different starting points.

Stepping searches are frequently employed due to their robust and deterministic nature. Herbst and Ross[37] used a stepping search routine coupled to a computerized sizing code for early design optimization of the F-15. For other examples see Schick et al.[38] or Crawford et al.[39].

2.5.4 Response Surface

The Encyclopedia of Optimization states that *“the evaluation or approximation of derivatives is a central part of most nonlinear optimization calculations”*[40]. This is typically in many MDO methods including the Finite Difference method and the Implicit Function Theorem described above. Unfortunately, there is a critical problem with such methods – the presence of numerical noise, which results in incorrect gradients and can delay or prevent convergence[41].

In the following methods, such derivatives are avoided by optimizing directly from evaluations of the objective function (measure of merit). These methods are called *zeroth-order* or *non-gradient* methods, and proceed by calculating specific values of the MOM at various combinations of the design variables. Non-gradient methods are described in detail in Hajela[42]. The research described herein is exclusively restricted to such methods for the reasons described in the Introduction.

In the widely-used *Response Surface* method (*“RS”*), the design variables are repeatedly changed to create a number of different designs, using either a simple parametric scheme or employing Design of Experiments to determine the best combination of variables for optimization purposes[41]. The resulting system (i.e., aircraft) variations are analyzed as to measure of merit and performance constraints. This creates a database of specific combinations of the variables and the resulting measures of merit and performance values. Sample calculated MOM data are illustrated in figure 7 using evenly spaced parametric combinations of the variables.

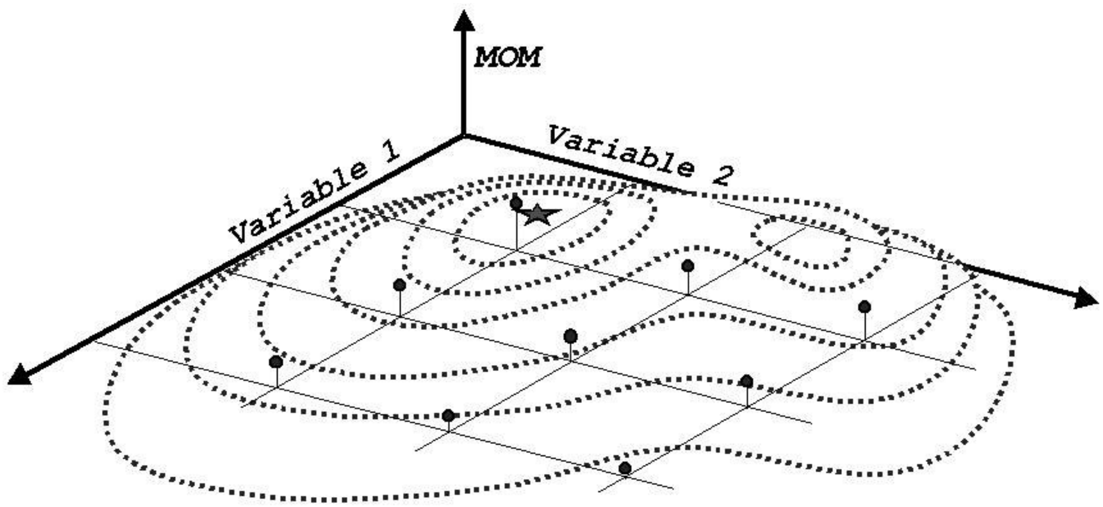


figure 7. Parametrically-created Measure of Merit Data Points

This parametrically produced data is then fit by least-squares methods to an approximating multidimensional surface equation, the Response Surface (figure 8). This is mathematically or numerically solved for an optimum, shown in the figure as a circle. Performance constraints are used as limitations on the allowable solution space. Note that the optimum on the response surface may not exactly match the optimum in the real objective function (figure 5) – but it should be close, and the "hill" is flat at the optimum so the result should have almost exactly the same value of the measure of merit.

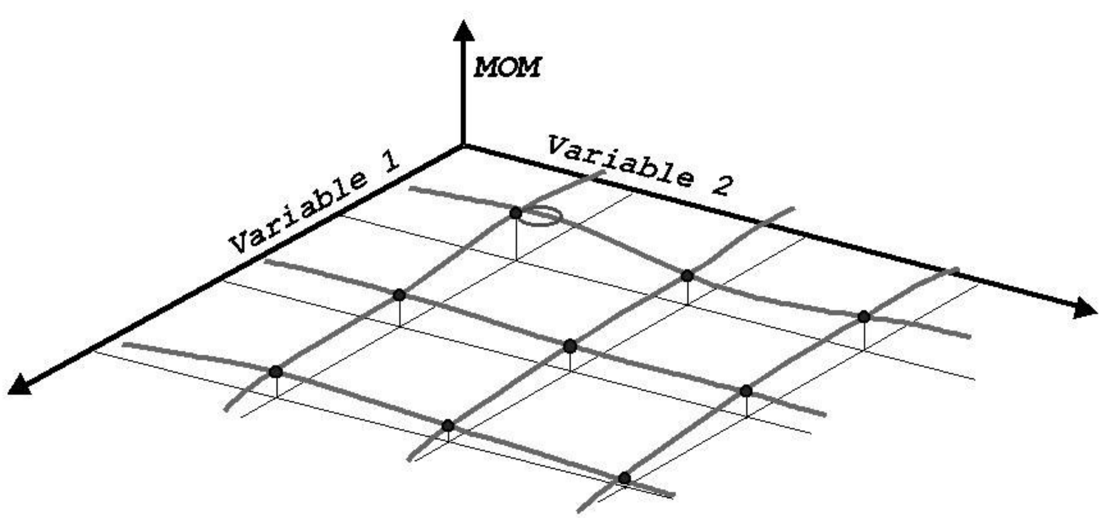


figure 8. Response Surface Fit to Data Points

Response Surface is one of the leading MDO methods, and has the advantage of bounding the number of alternative values of each variable that must be made to find an optimum. Essentially, a response surface can be created from any selected number of parametric variations, and the only question is this – how well does the surface represent the reality? Here, the number of calculations is traded against the validity of the result. If, by chance, the "true" optimum is located between and away from any of the points calculated, then the value obtained from the response surface may not be very close to the true result.

There is also the problem of "goodness of fit". The response surface is just a curve fit – it may fit well at the given data points, but poorly match the actual results between those points. Typically, a response surface is a multidimensional polynomial of only second degree, because higher degrees increase the computational workload and increase the chance of obtaining a non-real, "wiggly" fit. However, use of a second-degree polynomial prevents proper fit to a reflexed region of the design space such as shown in the curve closest to the Variable 2 axis in figure 8.

But, these problems are not severe and many researchers and industry designers report excellent results with Response Surfaces (Mavris and DeLaurentis[43], Cassidy[34]). In addition to the large reduction in the number of full design evaluations that must be calculated, RS has a further advantage of naturally smoothing out numerical noise resulting from the parametric analysis. Sevant et al[44] employed Response Surfaces to optimize flying wing designs primarily due to this noise-smoothing characteristic. Also, once the RS is fit it is quite easy to develop numerous design tradeoff graphs[45].

Another benefit of the Response Surface method is that the design points are selected and evaluated external to, and prior to the optimization. This makes it possible to select design points and have real engineers working offline do the design and analysis work to calculate the system-level response to changes in the design variables. In one company they go so far as to have designers prepare initial layouts of dozens of different aircraft concepts spanning the range of parametric design variables. These are then analyzed, fit to a response surface, and an optimum is determined[34].

2.5.5 Orthogonal Steepest Descent (Non-Gradient Stepping Search)

A mathematically simple method labeled *Orthogonal Steepest Descent ("OSD")* uses a full-factorial stepping searching method. It is similar to the Steepest Descent search described above, but uses neither derivatives nor finite differences to find the direction of maximum local improvement to the objective function. Instead, the region around the current best is investigated only along the variables' axes, and a step of pre-determined size is made in the best direction found (which is probably not the best direction there could be, but a few more step iterations will reach the "mountain top"). This is shown in figure 9, starting from the point labeled (1) and proceeding to the region of the optimum.

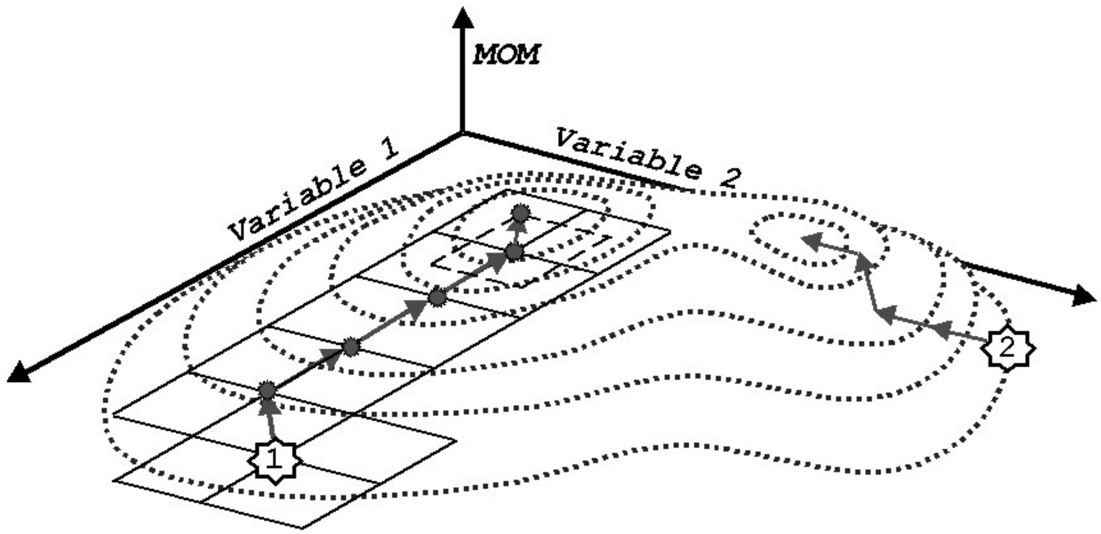


figure 9. Orthogonal Steepest Descent Full-Factorial Stepping Search

Each variable is parametrically varied by the selected step size (plus and minus), and the resulting aircraft are all analyzed. The aircraft having the lowest value of the selected measure of merit that also meets all performance requirements 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[6]. No derivatives or finite differences are required because no attempt is made to find exactly the best "direction" to move – motion is always along the orthogonal axes of one or more variables.

Because it is robust and deterministic, always finding the same optimum, this method is employed as a baseline technique in the research herein. *OSD* requires a large number of steps and a large number of calculations for each step, implying a lengthy calculation time. The amount of calculation goes up by at least 3^n, where n represents the number of design variables making *OSD* a poor choice for optimization of dozens or hundreds of variables. Also, by its nature *OSD* may find a "local" optimum as shown in the starting point labeled (2). This allows the possibility of an unfound better solution on a "different mountain". This method is further described in Section 4.

2.5.6 Monte Carlo

In the stochastic *Monte Carlo* method, a random probability function is used to generate a huge number of potential designs, and all candidates are defined, analyzed, and compare to the other designs to find the "best" design. This is defined as the design that meets all performance constraints and has the best value of the selected measure of merit. There is no proof that the "best" design found is the best that could exist, or that it is even close to the possible global optimum. However, since the Monte Carlo method is randomly examining the entire design space, it is unlikely that it will return a local optimum only, and if the found best design is substantially better than the design obtained without optimization then the method has served a useful purpose. This method is also used in the research described herein.

2.5.7 Random Walk

A stochastic stepping method called *Random Walk* applies a probability function to define a random direction away from the current design variable. If it leads to a better design, a step is made in that direction and the process is repeated. This resembles the steepest descent search of figure 6 but rather than step in the best direction found, a step is made in the first randomly-chosen direction that offers an improved value of the measure of merit. This is also called *Drunkard's Walk*, for obvious reasons as can be seen in figure 10. Note that this too is prone to getting stuck on a local optimum.

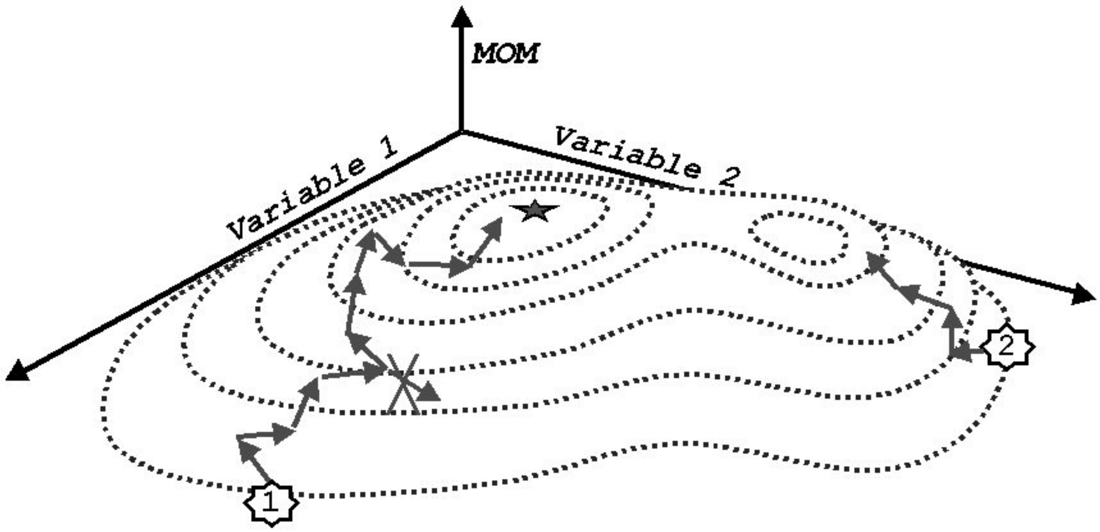


figure 10. Random Walk

2.5.8 Simulated Annealing

In *Simulated Annealing*, the tendency of Random Walk to get trapped in a local optimum is avoided by adding randomness to the acceptance of a "better direction". Early in the optimization, a probability function is applied such that some times, a step direction is accepted that actually leads to a worse value of the objective function. This probability is reduced as the optimization progresses so that, by the end of the optimization run, only "good" directions are accepted. To further ensure that the result is not a local optimum, the process can be restarted from this supposed optimum, with a high acceptance of worse values of the objective function reintroduced, then later driven out again[46].

Simulated Annealing is said to be analogous to the annealing of metals in which the temperature during cooling is sometimes increased then reduced again, to allow crystalline structures to settle into their lowest energy state prior to final solidification. Simulated Annealing was used by Pant and Fielding[47] to simultaneously optimize aircraft configuration and flight profile of a commuter/regional transport. The algorithm used employs random step directions with step size adjusted to keep the number of accepted configurations about the same as the number rejected.

2.5.9 Evolutionary Algorithm

An *Evolutionary Algorithm* works by applying a heuristic process of survival of the fittest to a defined population of potential solutions (i.e., aircraft). While Darwin is not normally associated with aircraft design, the modeling of aircraft characteristics as genes of design variables shows much promise. The design variables are coded into (usually) binary strings such that a collection of 1s and 0s defines a particular aircraft by its design variables (Crossley[48], Raymer and Crossley [49]).

Rather than starting with a single baseline design and trying to improve upon it, an evolutionary algorithm starts with a number of binary strings composed of random 1s and 0s defining some initial population of designs (figure 11). The measure of merit is evaluated for each of these designs. The optimum design is improved through a process involving selection and successive generations of alternative aircraft individuals as defined by the designs' bit-strings.

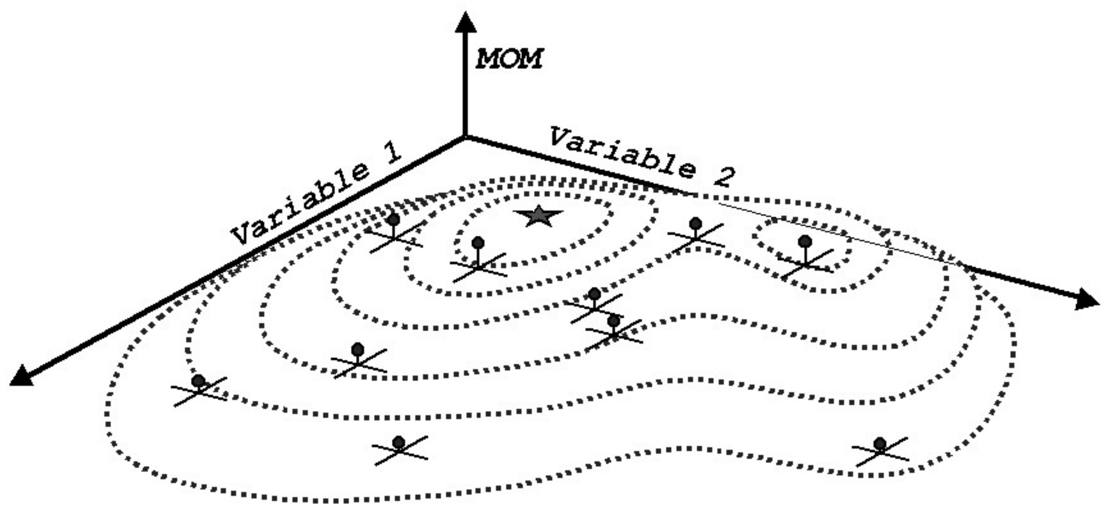


figure 11. Random Initial Population Generated from Design Variables

One of the earliest applications of an evolutionary algorithm was a demonstration using a simple hinged model, a wind tunnel, and a set of dice. Ingo Rechenberg and Hans Schwefel[50], students at Technische Universität Berlin, put the model in the tunnel and started with the non-aerodynamic shape shown at the top of figure 12. Then they threw the dice, and used the random results to adjust the angles between the different hinged segments. The better of different new "individuals" were kept as the "parents" of the next generation. Results from the first three generations are shown, and a slight progression towards a better shape can already be seen. After 200 generations, the obvious minimum drag result – a straight board – was obtained (almost). Rechenberg calls this the *Evolution Strategy (Evolutionstrategie).*

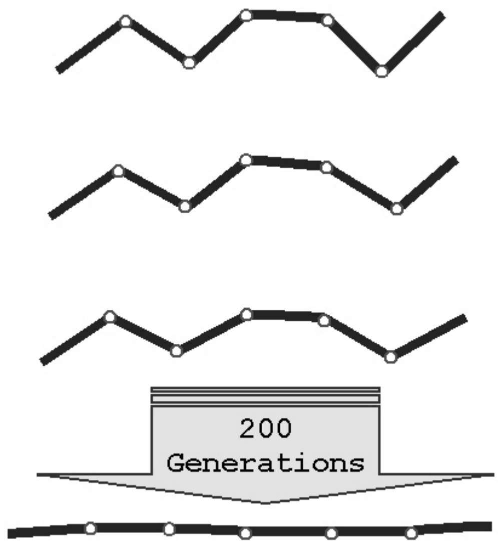


figure 12. Aerodynamic Optimization via Evolutionary Strategy[51]

A similar method called *Evolutionary Programming* does not attempt to "mate" parents of one population to create the next generation. Instead, the next generation is created by mutation applied to the most-fit of the previous generation. A variation of this method is employed in this research.

2.5.10 Genetic Algorithm

Another promising type of evolutionary algorithm is the *Genetic Algorithm (GA).* Members of a randomly generated starting population of aircraft are analyzed and evaluated as to fitness, based on the measure of merit, and the most fit are most likely to be permitted to reproduce. Aircraft variants are defined parametrically by the values of a chromosome-like genetic bit-string. Reproduction occurs by "crossing" their genes with those from another selected "parent". The next generation is evaluated as to fitness, and the process continues until the population all resemble each other or the measure of merit is no longer improving. This is presumed to represent an optimum.

The concept of a Genetic Algorithm is attributed to J. Holland in the early 1960's as described in his landmark book[52], and has been extensively explored for a variety of optimization problems. *GA* falls into the class of stochastic global optimization methods, and by its nature is not prone to falling into a local optimum. This makes it desirable for classes of problems with non-simple objective function shapes, and/or with complicated constraint functions. It is also useful for classes of problems with a large number of design variables, where traditional optimizers may simply be incapable of finding *any* usable solution.

An advantage of Genetic and Evolutionary Algorithms is their ability to incorporate non-continuous (*discrete*, or *integer*) variables such as number of engines in an optimization with continuous variables such as wing sweep. This is difficult to do with derivative-base optimization methods and generally has to be "faked" by what amounts to duplicate optimizations, one for each potential value of every discrete variable.

Genetic Algorithms have been applied to aircraft, rotorcraft, and spacecraft design by a number of researchers including Blasi et al.[53], Perez[54] et al., Roth et al.[55], Crossley[48], and Mosher[61]. A Genetic Algorithm has been applied to better predict the control inputs that may cause departure on the X-31 Post-Stall Maneuver aircraft[56]. *GA* has even been applied to assist in identification of weakly coupled submodules for a decomposition optimization[57].

Key concepts of the Genetic Algorithm include *Selection*, *Crossover*, and *Mutation*. *Selection* (to breed) can be done in many ways, but is always based on a calculated fitness measure that is related to the optimization objective function (measure of merit) and the constraint functions. One simple criterion would be, select the individuals with better values of the measure of merit (such as lower sized takeoff weight) provided that they meet all performance requirements. As this effectively "kills" all non-performing individuals, it may be too restrictive in that some individuals may have "good genes" otherwise, but those genes are lost because those individuals miss one performance value by some small amount.

Another selection criterion could be the calculated value of the measure of merit, post-multiplied by a factor based on violation of the performance constraints. So, the almost-good-enough individuals can still be picked for breeding. Selection of which individuals are allowed to breed can be based on a global stacking of all individuals or on a one-vs.-one tournament in which individuals are randomly paired to "fight it out", with the winner being allowed to breed with another winner.

A widely-used strategy, the "Roulette Wheel", assigns a likelihood of being selected based on the fitness measure then applies a random number generator to determine which individuals are actually picked. This is much like spinning a roulette wheel wherein the sizes of the "pie slices" are based on fitness.

Crossover refers to the actual "mating", the creation of a new individual in the next generation from (usually) two selected individuals. It too can be done in many ways. *Single-point Crossover* is done by bisecting the chromosome strings of the two parents into two parts at some (usually random) location. The first half of one parent's chromosome string is pasted to the second half of the other parent's chromosome string. Often, two children are created in this manner from each selected pair of parents by using the leftover halves for the second child[58].

Another scheme, *Uniform Crossover*, combines the genes bit-by-bit. Each bit is inspected for both parents – if they are the same, the child has that value (0 or 1). If they are different, a value is randomly selected. Another option is to treat the genes defining a particular characteristic, such as wing sweep, as a unit, and randomly pick them from one parent or the other.

Some researchers suggest using a weighted crossover scheme wherein a more-fit parent contributes a greater percentage of the genetic information of the child[61].

Mutation involves taking the chromosome strings of the children (i.e., the next generation) and generating a random number for each bit. If a defined low-probability result is obtained, the bit in question is "flipped" to the opposing value. This has the effect of creating new information in the population and serves to avoid premature or local convergence, but also interferes with the convergence to a certain extent. For example, the *GA* may have finally produced the ideal airplane only to have it lost when the random mutation operator changes a key design parameter.

An important formal analytical result with implications for the utility of the Genetic Algorithm is the *Schema Theorem*[52] (also called the *Fundamental Theorem of Genetic Algorithms*). This concerns the overall patterns in the chromosome string itself. Such patterns define "good" values of related design variables as the algorithm proceeds through many generations, and it is important that good patterns, or schemata, are not lost. Convergence finally occurs because the best schemata eventually dominate the population. The Schema Theorem is a mathematical expression of the number of schemata that will exist in the next generation based on the number in the current generation, the population size, fitness, crossover probability, mutation probability, length of the chromosome string, and length and order of the various schemata.

The Schema Theorem has several implications of special concern. When using a bisecting crossover operator, the position on the chromosome string of the various design variables becomes important. For example, it is well known that three-engine airliners are optimized with a lower thrust-to-weight ratio (*T/W*) than two-engine airliners because the effect of losing one engine is less catastrophic. A chromosome string with *T/W* right next to number of engines will likely preserve a good schemata involving them both, whereas if *T/W* is far from the number of engines an emerging good schemata will likely be lost during crossover. Other examples are less obvious. It is unlikely that those developing an optimization code will be able to anticipate every instance where the definition of the bit-string itself will affect the quality of the optimization result.

Also, long schemata are more likely to be disrupted than shorter ones by crossover. This may prevent the full development of multi-variable optimum schemata such as, "this wing loading plus this sweep plus this aspect ratio plus this taper ratio equals a good airplane."

In *GA* routines, there is a dichotomy between good "exploitation" of available schemata and good "exploration" of the design space. Single-point crossover "exploits" the good schemata of designs that are selected to reproduce, keeping approximately half of their schemata intact depending upon where the crossover occurs. Uniform crossover "explores" the design space by blending the schemata of the two selected designs. This often disrupts existing schemata, but introduces new schemata not present in either parent.

Probably because of these implications, many researchers believe that Uniform Crossover is preferable to a bisection crossover. However, Uniform Crossover itself tends to break up schemata by the random selection of a bit where the parents have different bit values.

Another implication of the Schema Theorem is that excess mutation can continuously destroy emerging schemata, causing the Genetic Algorithm to become more like a random, Monte Carlo optimizer. This affects both uniform and bisection-type crossovers.

Guidelines for mutation rate per bit suggest that for single-point crossover, the probability of mutation per bit should be between $(1/N)$ and $(1/NL)$, where N is the population size and L is the length of the chromosome bit-string. For uniform crossover, empirical evidence[59,60] indicates that a reasonable mutation rate equals $((1+L)/2NL)$.

The size of the population is important in the Genetic Algorithm method. If too large, the computational time is excessive and the number of generations may be limited. If population size is too small there may be insufficient "genetic information" in the initial population, or the random recombinations of genetic information may cause useful information to be lost. Crossley[59] suggests a population size of 30 or more if the chromosome string is less than 30 bits long, and if more than 50 bits long, the population should greater than 100. Mosher[61] suggests a population size four times as large as the number of bits in the chromosome bit-string. Values derived by Goldberg[62] suggest an optimal population size equal to $(1.65)\ 2^{0.21L}$.

As the Genetic Algorithm, like Monte Carlo, relies heavily on random probability factors and is hence not deterministic, there is little assurance of getting a repeatable result or knowing that the "optimum" discovered is actually the very best possible result. There is not even a mathematical proof of solution convergence in a Genetic Algorithm – but they do work, and often work well!

There are many variations on this basic *GA* scheme including the manner in which designs are selected for reproduction, the manner in which their genes are combined to produce the next generation, and the use of options such as mutation, replacement, and elitism, as discussed below. It has been said that there are as many Genetic Algorithms as there are researchers developing them. Many of these *GA* variations have been coded and evaluated in the research described herein.

2.5.11 Decomposition

Decomposition works by partitioning a large engineering design optimization problem into a number of smaller, solvable problems (sub-modules). During execution of the optimizer, top-level routines pass data between the submodules in a structured manner that retains their coupling and accommodates the defined system constraints (figure 13). For example, a wing analysis decomposition may have an aerodynamics module that knows how to calculate drag and airloads if it knows the wing shape, and a structures module that knows how to calculate weight and structural deflections if it knows the airloads. Each executes separately, passing their results to the other until they converge at an optimum for the measure of merit such as weight or drag, or a blended bit of both.

Decomposition should really be understood as a framework for MDO problem simplification, and once accomplished, the decomposed sub-problems are solved using one of the other optimization techniques.

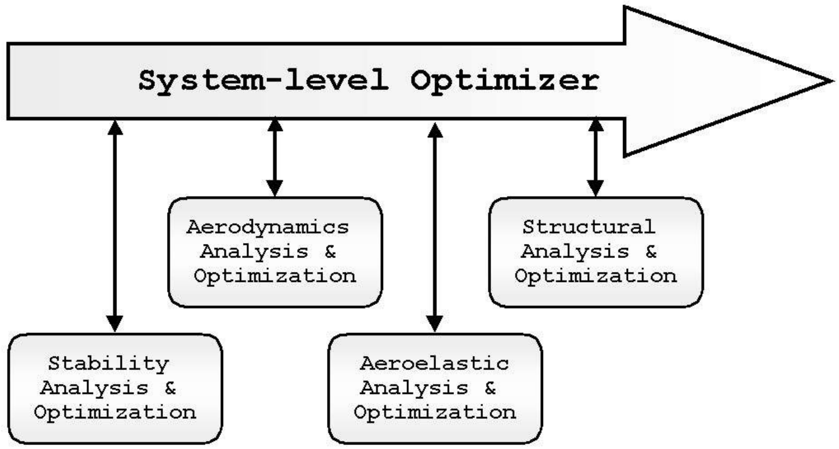


figure 13. Optimization by Decomposition

The key to successful application of decomposition is to separate the design variables, constraints, and/or analysis methods into groups that are only weakly interconnected. Then we can perform separate optimizations within these groups, coordinated and linked such that the entire system is optimized when the separate optimizations are brought together. The groups may also be broken into weakly-interconnected subgroups, creating a tree-like structure to the optimization process with a top level that is the entire system, and sub-levels below it representing groups and subgroups.

In textbooks, decomposition it often illustrated using connected structural members such as a door frame[46]. This is readily and obviously decomposed into the separate beams comprising the door frame. Applying such techniques to the problem of aircraft conceptual design optimization remains difficult, as there are not such obvious, weakly interconnected groupings to decompose. Sobieski[63] has developed an analytical method for assessing a multidisciplinary system and determining its sensitivity to its design variables. This method, called *Concurrent Subspace Optimization*, uses global sensitivity equations as a way to decompose an optimization problem analytically rather than by inspection and intuition.

Significant research continues into the use of decomposition for aircraft design problems, and successful industry utilization has been reported (for example see Hollowell et al.[64], Batill et al.[65] or Isikveren[66]).

2.5.12 Stopping Criteria

An important aspect of any optimization method is deciding when to stop. Poorly defined *stopping criteria* can cause the execution to run on far longer than required, or can stop execution before a better result can be found. Typical stopping rules include:

- Fixed number of iterations or populations
- Fixed amount of execution time/cost
- Objective function unchanged for specified number of generations
- Small percent improvement in objective function over last value
- Similarity in design variables (genes)
- Percent coverage of the possible design space

2.6 The MDO Realism Problem – Automating Aircraft Redesign

Multidisciplinary Optimization promises to help develop a significantly better aircraft. However, the optimized best design must be turned from a computer-generated collection of design parameters into a real design layout, either on a drafting table or more likely, on a CAD system. It is quite possible that when a real aircraft designer tries to turn optimized parameters into a buildable layout, unforeseen problems will arise. The fuel and landing gear may not actually fit, forcing a stretch in the fuselage. This would increase drag, Or, the center of gravity may have moved enough that the wing has to be moved to compensate, forcing redesign of everything from landing gear location to control surface sizes. Perhaps the computer–defined longer nose interferes with pilot overnose vision, or any of a hundred other real-world design requirements may have been violated.

Somehow, the initial baseline design must be automatically modified for each parametric variation in a realistic manner so that the final selected "best" aircraft as modeled in the optimizer is very similar to the design that will emerge on an actual, post-optimization design layout. If this is poorly done, the designer will have to make substantial changes to the design as optimized by the computer. These changes will in turn change the aircraft's aerodynamics, weights, and other analysis results, producing an aircraft different from and probably worse than the design promised by the optimizer.

As a simple example, consider the optimization of wing loading *(W/S)* to maximize the range of a propeller-powered aircraft. This can be analytically determined as:

$$\left(\frac{W}{S}\right)_{Optimum} = q\sqrt{C_{D0}/K}$$

where q is the dynamic pressure, C_{D0} is the parasitic drag coefficient, and K is the drag-due-to-lift factor. Clearly, if the designer must increase the fuselage size after the optimum *W/S* is determined, the parasitic drag will increase and the best *W/S* will be higher than that predicted by this simple optimization method. This may well happen if the “optimum” wing loading as found with this equation is much higher than assumed for the first design layout. A higher wing loading corresponds to a smaller wing, which may not hold the required amount of fuel – thus a bigger fuselage would be required.

Definition of automated redesign procedures for use within an optimization routine is therefore important for obtaining usable results. To continue our example, if an MDO optimization includes wing loading as one of the variables, we must parametrically vary the wing loading and calculate the effects upon the aerodynamics, structure and weights, stability, control, and any other functional disciplines that may be affected by the change.

One effect is obvious - the wing gets bigger or smaller. Others are subtler. As the wing area changes, its proportion of wetted to reference area changes. Its area and thickness change so it can hold more or less fuel, and if the landing gear is located in the wing the tires may no longer fit. As a result, a change in wing loading will often require a change in fuselage size. When this is done, the landing gear geometry may no longer work, requiring further geometric changes to the fuselage and/or wing. The tails, which must be resized to accommodate a larger or smaller wing, must also adjust in size depending on the resulting tail moment arm. A change in wing size could even impact the inlet duct length or external flow field causing a change in thrust and fuel consumption.

Altogether, a simple parametric change in wing loading may flow down to a large number of changes to the aircraft's representation in the input data for analysis, cutting across many of the functional disciplines incorporated in the MDO. These may in turn have an impact on the chosen measure of merit, or upon design constraints such as performance requirements or cost targets.

Routines for each of these can be postulated and included in an MDO code. This has been done in some cases, and the old optimization codes of the major aircraft companies had many such real-world effects included[12]. These programs had computer routines that would change the representation of the aircraft design as a result of changes in the design variables, in a fashion intended to represent what a human designer would do. However, they tended to be very design-specific and had a long set up time (weeks) and needed expert users to obtain good results.

At present, it appears that MDO methodology development is running ahead of our ability to implement reasonable automated aircraft redesign within those codes. Automated procedures are needed that will redesign the concept in response to changes in parametric variables much as an actual designer would do. These procedures must have minimal set up and *not* require an expert’s attention for every design being studied. Furthermore, it is important that only those areas that substantially influence the output of an optimization be included, to avoid excessive workload.

An attempt is made in this research to postulate and assess a suite of credible automated aircraft redesign procedures, as described in Section 5.

2.7 Observations Concerning Variables, Constraints, & MOMs

When crafting an MDO program or routine, the developers must select the design variables, measure of merit, and design constraints. The design variables are terms that represent physical features of the design that will be parametrically varied in some fashion to find an “ideal” set of features. Design variables can be smoothly changeable over a continuous spectrum, such as wing area, or can be discrete variables that have only integer values such as number of engines or number of aisles and seats across.

Selection of design variables completely drives the result, to a greater extent even than the optimization method employed. As an obvious example, failure to include the possibility of variation in the wing loading of an aircraft will make it very doubtful that the true optimal aircraft will be found. The full range of design variables is often called the "Design Space". It could include tens of thousands of design variables defining everything about the aircraft, but wise designers carefully select the most-relevant suite of design variables depending upon where they are in the design process.

Design constraints are aircraft properties that must be attained for design acceptability. Normally these are performance constraints such as takeoff distance, rate of climb, or cruise speed. Sometimes design constraints include cutoff values for physical features such as wingspan (can't be greater than existing commercial taxiways or military hardened shelters) or fuselage diameter (can't be less than minimum passenger or cargo envelope plus structural allowances). Design constraints can also include environmental restrictions such as noise propagation or creation of ozone-depleting chemicals.

If the chosen suite of design constraints is incomplete, incorrect, or poorly analyzed, the apparent best aircraft may not be usable, or a better aircraft may be missed. For example, if a "quick and dirty" takeoff calculation is used in an optimization, the "constraint line" in the solution space may be to the "left" or "right" of reality, causing the selection of either a design that doesn't actually meet the required takeoff distance, or a design that exceeds the required value and thus is heavier or more expensive than optimal (in either case, the design is actually "suboptimal" in the chosen measure of merit).

A measure of merit (MOM) is desired capability that the vehicle will be optimized to attain. In mathematical terms, the measure of merit is the objective function for the optimization. Typically the MOM is based on cost or weight if the vehicle is being sized to a required mission, or on range if the aircraft is being designed to a pre-determined weight, engine, or physical size.

A compound measure of merit can be created, blending several calculated measures of merit via some weighting scheme. For example, the true measure of merit may be ton-miles of cargo delivered at constant cost, but there may also be an intuitive "goodness" associated with a smaller aircraft. To bias the optimization away from extreme-size solutions, the calculated ton-miles cargo delivered may be reduced by some factor times the calculated aircraft empty weight. Alternatively, the MOM blending could occur by summing ton-miles and empty weight, with one of them multiplied by a factor equating them for an “apples-to-oranges” comparison. This can be useful, but the selection of the mysterious factor will completely drive the resulting design solution. Alternatively, the graphical methods of Pareto[29] (figure 4) can be applied to two diverse measures of merit, showing the reduction in one to obtain an improvement in the other.

Selection of the measure of merit is critical to the determination of an optimal design. If a design is optimized to the wrong measure of merit, say minimum weight, it will not be the best aircraft if another measure, say fuel, is actually more important. The aircraft should be optimized to the measure of merit that provides greatest value to the intended customer and results in the largest number of sales. Sometimes this is difficult to determine.

Another complexity to this discussion: some of the "design-to" performance requirements such as range, payload, or cruise speed may not actually be included in the constraint calculations. Instead, they may be embedded in the calculation of the measure of merit. The aircraft is usually sized to a required range carrying a required payload, and those values are used in the equations that generate the sized takeoff weight (and most other MOM's). Therefore, we do not need to include those parameters again in a constraint calculation even though they are clearly design constraints.

An important consideration in selection of criteria for all of these is that it is sometimes difficult to recognize just what is a design variable, what is a constraint, and what is a measure of merit. For example, some may consider takeoff speed or aircraft turn rate to be design variables and may perform design trade studies to attempt to optimize them. In this author's opinion, those should more properly be considered design constraints, and treated accordingly in the MDO methodology. An attempt is made below to provide precise definitions to avoid such confusion.

Following are recommended selection criteria for Design Variables, Design Constraints, and Measures of Merit based on this discussion. Selection criteria for Design Variables, which are the *independent variables* of an optimization, are given in table 1.

table 1. Recommended Selection Criteria: Design Variables

1. Represent clear physical or technical features of the design layout that can be expressed and parametrically modified by a single numerical value (ex. Wing Loading), or can be clearly explained as a discrete, "either-or" choice (ex. Tail Type).
2. Should *not* represent items of specific end worth or value to customers (ex. Range, Speed, Cost) - such items should be incorporated into the measure of merit or constraints.
3. Non-trivial and calculable impact on chosen MOM (typically related to weight or range).
4. Global impact - i.e., affect entire design such that variables cannot be optimized independently (ex. Inlet duct length of commercial airliners doesn't usually require global optimization).
5. Reflect current and historical design parameters and vocabulary wherever possible.

The last point needs further discussion. Some MDO research reports indicate use of design variables unlike those normally used in industry design offices (for example, Knill et al[45], Koch et al[67]). This is especially true in the optimization of wing geometry, where a "cloud" of arbitrary X-Y planform points is sometimes used to define and optimize the wing, typically with leading and trailing edge points at the root chord, tip chord, and a few “break” points. In such a scheme the wing area, sweep, aspect ratio, and other reference wing parameters are determined by the final locations of these X-Y planform points rather than being specified in the MDO routine. Results sometimes show wild "optimal" planforms with grossly exaggerated chord length changes, including tip chords substantially longer than inboard chords implying negative taper. This requires a laborious and design-dependent process to drive out those obvious bad answers.

Aircraft designers think and design in terms of the trapezoidal reference parameters, namely wing loading (or area), aspect ratio, taper ratio, sweep, thickness, and dihedral. These define the wing in a time-honored and useful fashion, and there is a wealth of historical and test data to assess and bound the wing planform selection based on those parameters. Non-trapezoidal changes to the basic wing shape are generally considered in terms of percent chord variation from the trapezoid at specified percent span locations.

Use of this "normal" way to define the wing has two major benefits. First, it makes the results immediately meaningful to experienced industry aircraft designers. Second, and perhaps more important, these traditional design variables can easily be checked for reasonableness against standard design practice (see, for example Batill et al.[65], Isikveren[66], Pant and Fielding[47], or Crossley et al.[48]).

Another desirable feature in a suite of design variables, consistent with historical practice, is to utilize ratios and non-dimensional parameters to the greatest extent possible. This is seen in the wing parameters discussed above, and is also true in parameters such as fuselage length-to-diameter ratio, engine bypass ratio, propeller disk loading, and tail volume coefficient. Not only is it easier to control wild excursions with ratio-based variables, but comparisons to other designs and prior design iterations are facilitated.

Another type of variable should also be considered and defined, the “noise” variable. These are not parametric variables that will control the characteristics of the aircraft. Instead, they are uncertainties in either the definition and analysis of the aircraft itself or in some external factor that will affect the optimization outcome. In classical aircraft design analysis, we subject the baseline aircraft design to parametric variations of drag coefficient, engine fuel consumption, empty weight, and similar factors to determine how sensitive our concept is to unexpected variances from our best-guess predictions in these areas. We also subject the design to variations in external factors such as fuel cost or passenger load factor to determine if our concept is still economically viable in the face of such uncertainty. In a similar fashion, MDO methods can assess sensitivity to such noise variables. A good example of such incorporation of noise and control variables into an MDO method can be seen in Mavris[68].

Recommended selection criteria for Design Constraints are provided in table 2. These can be considered the *constraint lines* in the optimization.

table 2. Recommended Selection Criteria: Design Constraints

1. Represent calculable properties of the aircraft given a particular set of values of the Design Variables.
2. Must-meet *numerical* values derived from commercial or military specifications, safety, operability considerations, customer desires, or overall good design practice.
3. Preferable if design constraints represent "firm" requirements that are unlikely to be relaxed in the face of cost/performance shortfalls.

Concerning the last statement, more flexible requirements such as cruising altitude could be folded into a compound MOM. This would allow the optimizer to "discover" that a slight reduction in cruise altitude makes the aircraft much cheaper. This approach is difficult and is not normally done with classical aircraft optimization methods such as carpet plots. Instead, we have demanded a particular cruising altitude (continuing the example), and when the aircraft was heavier and more expensive than we'd hoped, we started looking for something of lesser priority to trade away. By proper formulation of the MOM in an MDO routine we can learn that up front.

The required numerical values of the constraints are normally considered to be "firm". If the FAR/JAR climb requirement is 50 fpm, a design yielding 51 fpm gets no extra credit whereas a design yielding 49 fpm cannot be certified nor sold. In some cases, though, requirements are customer-driven and are not so firm. The customer may indicate a desire for, say, a 2000-ft field length but if offered a cheaper aircraft with a 2500-ft field length would probably be happy. These can be handled with some variation of "fuzzy logic" or via ramp-shaped weighting functions applied to the MOM based on the calculated constraint values. (Note the blurring of distinction between constraints and measures of merit.)

Some design constraints fall into a special category in that they can either be implemented as constraint lines bounding the optimization, or they can be used to adjust the design directly during the optimization. For example, it is usually wise to consider aircraft volumetric density as a constraint, not allowing it to rise above some selected practical value. This could be treated as a calculated property with an appropriate constraint line through the design space. Or, it could be used to redefine the aircraft geometry for each iteration (the preferred approach, discussed below). If the aircraft total volume reduces due to, say, a reduction in wing area, the optimizer could make up the volume shortfall by lengthening the fuselage. This will probably result in a faster solution than an optimization that merely penalizes a design having inadequate internal volume.

Recommended selection criteria for Measures of Merit are provided in table 3. The MOM is the *objective function* and represents the *dependent variable(s)* in the optimization.

table 3. Recommended Selection Criteria: Measures of Merit

1. Represent a non-trivial and calculable indication of the worth of the concept.
2. Materially affected by the design variables and constraints.
3. Clear meaning to designers and customers.
4. If blended, need clear rationale for method and factors used for blending

Worth is typically defined as cost or a cost-related measure such as weight. For a fixed-size design, worth may be defined by a selected performance result such as range. Some important sales-related aspects of a design cannot be included in a MOM – beauty, benefit to mankind, national pride, or "to infinity and beyond" can have no role in MDO unless they can be reduced to non-trivial and calculable numbers

Appendix A contains suggested design variables, constraints, and measures of merit based on the above discussion and on historical usage in the aircraft industry. During this research, certain of these suggested design variables, constraints, and measures of merit have been programmed into MDO routines and evaluated for their utility and relative importance to the design solution.

(This page intentionally blank)

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[69].

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.

In selecting these variables, it was decided to forgo inclusion of any propulsion system design variables such as jet engine bypass ratio or propeller diameter (other than, of course, engine size via the *T/W* ratio). This does not imply that this author considers those unimportant, and they may be studied at a later date. However, parametric engine models that permit such design variations are only approximations of the complexities of actual engine data, and often ignore factors such as overspeed limits, part-power installation drag changes, actual bleed and power takeoff effects, or continuous power limits. Those complexities are calculated by large cycle analysis programs at the engine companies, and their sophisticated data is normally provided as an input to the aircraft design analysis and optimization. Trades are done using alternative engines as provided by the engine company.

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.

In this research, this author's prior methods were expanded in several key areas. 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.

Automatic aircraft redesign procedures are discussed in Section 5.

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 14. Section 6 provides a full description of these designs.

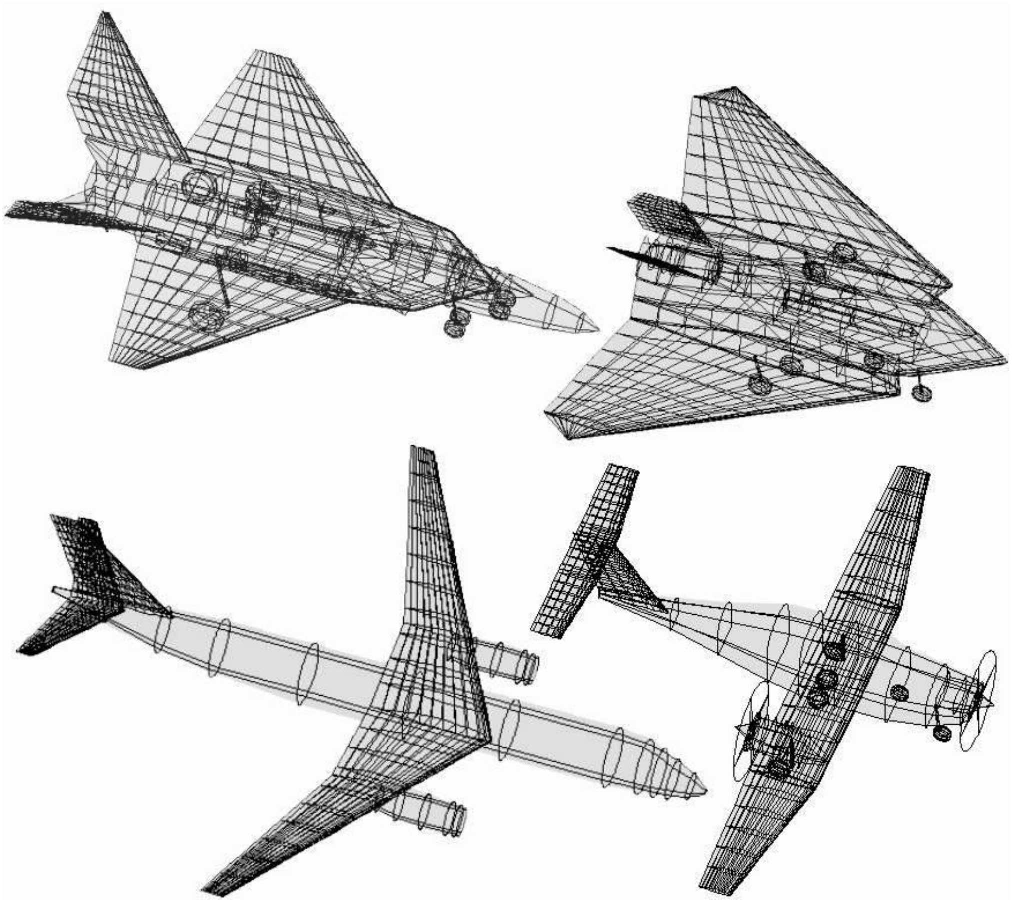


figure 14. Validation Models: Four Aircraft Notional Concepts

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 (these runs are not included in the run matrix defined in the Appendices, and total about 20 runs).

While spanning the spectrum of design concepts, these validation models represent only a tiny subset of the available options for aircraft design and will differ from the concepts considered in any particular industry design project. Thus, any conclusions reached in this research must be carefully reviewed when applied to other aircraft concepts (as well as other MDO approaches and aircraft analysis methods).

Furthermore, not all of the available MDO methods were included in this research – methods were limited to the Orthogonal Steepest Descent and the Chromosome-based methods as described above. Results herein cannot be extended to cover methods not studied, and the author's reasons and rationale for selecting particular MDO methods may not apply to others. However, within the methods studied and the limits of the methods and test models employed, these results should be of interest.

(This page intentionally blank)

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[70] includes sophisticated implementations of the classical analysis methods[71] 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[6]. 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.

Coding for these methods and options was added to the latest version of the RDS-Professional program, and quite possibly represents the widest variation of MDO capabilities ever programmed into one computer program.

Since it is intended to include these methods with the next commercial release of RDS-Professional, the coding had to be practical for industry usage. Requirements for industry practicality were well defined in the CFD context by Vos, Rizzi, Darracq, and Hirschel, as follows[19]:

1. *Assured Accuracy (fidelity)* in the sense that the engineer has confidence in the results,
2. *Acceptable costs* in terms of both computer run times, including set up and turnaround, and human effort to learn the skills to run the code,
3. *Robustness* so that it can be run by a non-specialist, and
4. Sufficient *Generality* in the data structures and objects allowing future code modifications, refinements and developments.

These guidelines were employed in the RDS implementation of MDO techniques, with reasonable success in this author's opinion. Final proof of this can only occur with customer acceptance and use of the end result.

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 Section 6.

These four validation models were optimized repeatedly using different MDO routines and different options of those routines. Results are detailed in Section 7. Below are descriptions of the MDO methods and options as implemented into RDS-Professional for this research.

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[72].

As originally programmed, this method would optimize an aircraft concept simultaneously using six key design variables, namely *T/W*, *W/S*, aspect ratio, taper ratio, sweep, and thickness. During the research described herein, this was expanded to include fuselage fineness ratio (*f*) and wing design lift coefficient ($C_{L\text{-}design}$). Section 5 discusses selection of these variables and the manner in which the aircraft concept is varied as these parameters are changed (automated aircraft redesign).

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. This was depicted in figure 9. The coding logic is shown in figure 15.

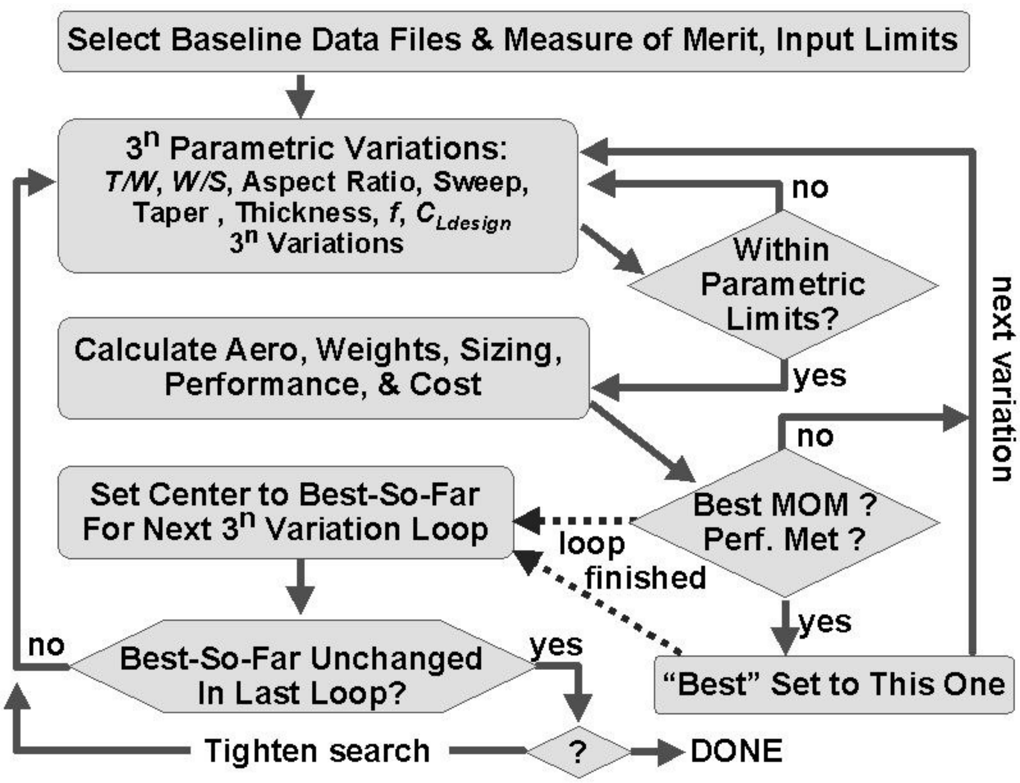


figure 15. OSD Optimizer Program Logic

This process has been termed *"Orthogonal Steepest Descent (OSD)"* since it changes design variables by plus/minus increments about the current "best", moving along the orthogonal parametric axes in the direction that produces the largest improvement in the measure of merit without violating the performance constraints. *OSD* does *not* attempt to compute the gradient vector to find the true direction of steepest descent, which is likely to be off the parametric axes – hence the name Orthogonal. But, it eventually finds the correct answer as the step resolution is decreased, and has proven to be very robust and reasonably fast. With eight design variables on a modern one-gHz computer, it typically finds an optimum in about 10-30 minutes depending on the resolution sought and the complexity of the design mission.

While this method can theoretically result in a "local" solution because it searches starting from a single point, the initial step size is so large that the whole design space is initially examined. Also, it is presumed that a well-designed initial aircraft concept used as a starting point for the optimization will not be "on the wrong mountain". The solution obtained should be the global solution for the given design space unless the designer is completely wrong about the chosen design approach.

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-based scheme actually turns the continuous variables into discrete variables. The resolution can be made as fine as desired by using more bits for each design variable. Resolution is calculated by

$$resolution = \frac{x_{\max} - x_{\min}}{2^l - 1}$$

where $x_{\max}$ is the upper bound, $x_{\min}$ is the lower bound, and l is the number of bits used to represent each gene (six as applied here). In the wing loading example above, the resolution is 0.95, equal to the total span of the variable (100-40=60) divided by the span of numbers represented by that variable's bits (60/63=.95).

While more digits than six would give greater fineness to the variable graduations, this author believes that such accuracy is greater than the noise level of aircraft conceptual analysis and hence is not worth the trouble and computational expense.

Put another way, this chromosome definition of eight gene parameters each represented by six bits gives 64^8, or 281,474,976,710,656 possible aircraft variations. That should include some concepts very close to the absolute optimum! Needless to say, exhaustive enumeration and analysis of all possible variations is out of the question.

Purely discrete variables such as tail type or number of engines have not been considered as of yet. It was originally intended to do so, but the problem of the definition of believable procedures for automated redesign for variations such as one engine vs. two proved insurmountable except for limited design-specific cases. This remains a fertile field for further research, and is discussed in section 5.6 below.

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 (see code snippet[§§]). 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.

```
FOR iAC=1 TO iPopultn
 FOR idum=1 TO iNumBits
  RNUM%=INT(ROUND(RND,0))            'creates random integer 0 or 1, RND is random real 0-1
  GA$(iAC)=GA$(iAC)+LTRIM$(STR$(RNUM%))
  NEXT idum
NEXT iAC
```

4.3.2 Selection - MOM Weighting

Selection of the "best" individual or individuals is based primarily on the calculated value of the Measure of Merit (objective function). This is determined from the aerodynamic, weight, propulsion, sizing, performance, and cost analysis of the revised aircraft per the design parameters as defined by that aircraft's chromosome bit-string. As implemented herein, measures of merit include takeoff gross weight, empty weight, fuel weight, purchase price, life-cycle cost, and internal rate of return (financial analysis). Alternatively, range can be used as the MOM if the aircraft weight is specified or fixed for some reason.

The MOM can be directly used for selection. The best aircraft is ranked first, the next best is second, etc… In some methods this may cause premature convergence due to the excessive influence of a few initial "super-individuals". To counter this, a random "stirring" can be applied to the calculated values of the MOM. The MOM's of all the individuals are mapped to a 0-1 scale, then those values are multiplied by a 0-1 random number. These random-adjusted MOM values are then used to determine which individuals are selected for defining the next generation. The ultimate selection of the final most-optimum aircraft is *not* random-stirred, for obvious reasons.

This random-stirring approach may excessively penalize the better individuals. Being "lucky" is equally weighted with being "good". To increase the importance assigned to MOM as opposed to the random factor, MOM weightings were added in which the 0-1 MOM scale is adjusted by a weighting function.

[§§] Code snippets are intended to illustrate the methods employed and to assist others in duplicating them. They are not themselves executable, nor do they illustrate the thousands of lines of code required to make them executable in the actual MDO routines (plus ~20,000 lines of analysis code). Code snippets are illustrated using conventions from *Powerbasic*, a high-end compiled extension of Basic that generates fast executables yet remains easy for a non-expert to read. For a full description of programming statements as used in these code snippets, see the *Powerbasic Reference Manual*, Powerbasic Inc., Carmel CA, 1997.

A linear weighting function represents the original, simple multiplication of MOM ranking scale times a random (0-1) number. Other options, which bias selection in favor of the better individuals, include MOM^2, MOM^4, and cos(MOM), as shown in figure 16. The cosine function seems especially desirable because it not only increases selection of the best individuals, but minimizes the probability that the bottom-most individuals will get "lucky" enough be selected for defining the next generation.

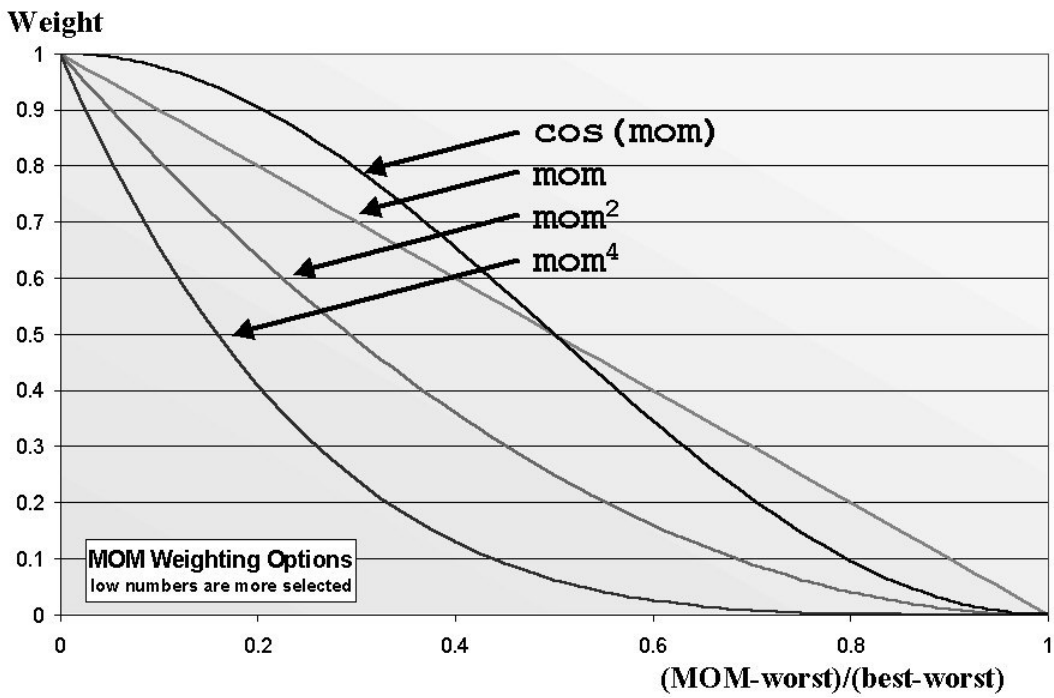


figure 16. MOM Weighting Functions

A high (good) MOM value returns a low MOM weight because the final selection criterion is a minimization (see code snippet below). When range is used as the MOM, it is subtracted from an arbitrary big number to convert optimization to a minimization problem.

```
xKmom=(xMOM(iAc)-WrstMOM0)/(BestMOM0-WrstMOM0)
IF ibWeight=1 THEN                    'linear
      WtMOM= abs(1.-xKmom)
     ELSEIF ibWeight=2 THEN      'MOM^2
      WtMOM= (1.-xKmom)^2
     ELSEIF ibWeight=3 THEN      'MOM^4
      WtMOM= (1.-xKmom)^4
     ELSE                                  'cosine wave
      WtMOM= (0.5+0.5*cos(3.1416*xKmom))
     END IF
```

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 (reported in Raymer and Crossley[49]), 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.

In mathematical terms, this approach can be expressed as the creation of a modified objective function f from the actual objective function f_a using penalty multipliers c_j (also called *draw-down coefficients* or *penalty weights*), as follows:

$$f = f_a + \sum_{j=1}^{\#\text{of constraints}} c_j \max\left[0, g_j\right]$$

The constraints are defined in the g_j functions, which are positive-valued if the constraint is violated and negative-valued if it is satisfied, as follows:

$$P \le P_{req} \rightarrow g = \frac{P}{P_{req}} - 1 \quad \text{else} \quad P \ge P_{req} \rightarrow g = 1 - \frac{P}{P_{req}}$$

If a constraint function g_j is not violated, the term max $[0, g_j]$ is zero so there is no adjustment to the objective function from that constraint.

Note that the Penalty Function Method turns a constrained optimization into an unconstrained one. Also, it is possible to relax the penalty early in the optimization then tighten it later to ensure that the final optimum does meet all constraints, much like in Simulated Annealing. This has special application in Evolutionary and Genetic algorithms because it allows useful schemata to be preserved into later generations even if the resulting solution does not exactly meet all constraints.

The key to successful use of the Penalty Function Method is the selection of the function used. Early in this research a ranking scheme was defined based on a user-input *penalty acceptance ratio* for each performance requirement, expressed as the percent of degradation in a given performance requirement acceptable for a percent improvement in the objective MOM[73]. The calculated MOM was to be increased or reduced by the MOM times the relative shortfall in performance value divided by its user-defined penalty acceptance ratio.

Upon further reflection it was decided to first try something simpler, because of the increased user workload and the uncertainties associated with using qualitative assessments for such a critical part of the optimization. Furthermore, many of the performance requirements used in aircraft design are based on the performance parameter Specific Excess Power (P_s) for which the required value is often zero. This makes it difficult to define a suitable ratio for violation of the constraint.

The simplest possible Penalty Function Method was therefore tried, 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 so well that the more-complicated scheme described above was abandoned.

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.

[***] The Penalty Function Method described here is more-properly termed an *Exterior* Penalty Function Method, because it is only active if the solution is outside the feasible design space. *Interior* penalty functions are also possible, wherein the objective function is penalized as the constraint barrier is approached from within the feasible design space. An interior penalty approaches infinity at the constraint barrier, preventing passing into the infeasible region. This method was not investigated because it does not provide the "forgiveness" properties that will allow relaxation of constraints early in the optimization to maintain desirable schemata, nor does it promote final solutions exactly on the constraint lines as is typical in aircraft design optimization problems.

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.

Elitism is contrary to the evolutionary nature of such routines because the Elite individual is unchanged, either through crossover or mutation. Implementations that are restricted to small population sizes suffer an evolution penalty if Elitism is employed, because one or more individuals in each population are not evolving. Elitism is a fix to a problem, not a desirable thing in of itself. But, with population sizes of up to 500 individuals, Elitism is a trivial penalty to the codes described herein and does serve to avoid “backsliding”, as is shown in test cases below.

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.

<u>Single-Point Crossover:</u> 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. In the implementation herein, a second child was not created from the leftover pieces as described in Section 2.5.10.

```
  ‘ibCross=3 is midpoint crossover, ibCross=4 is random point
  ‘MID$( GA0$(),1,idum1) finds bit value at idum1 position
IF ibCross=3 THEN idum1=INT(.5*iNumBits)   ‘midpoint
        ELSE idum1=INT(RND*iNumBits)  ‘random point
  RNUM%=INT(ROUND(RND,0))
  IF RNUM%=0 THEN
   GA0$(0)=MID$(GA0$(iParent1),1,idum1)+ MID$(GA0$(iParent2),idum1+1,iNumBits-idum1)
   ELSE
   GA0$(0)=MID$(GA0$(iParent2),1,idum1)+ MID$(GA0$(iParent1),idum1+1,iNumBits-idum1)
   END IF
```

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.

```
FOR ij=1 TO iNumBits
  b1$=MID$(GA0$(iParent1),ij,1)  'finds value at ij position
  b2$=MID$(GA0$(iParent2),ij,1)
  IF b1$=b2$ THEN
    GA0$(0)=GA0$(0)+b1$
    ELSE
    RNUM%=INT(ROUND(RND,0))
    GA0$(0)=GA0$(0)+LTRIM$(STR$(RNUM%))
    END IF
  NEXT ij
```

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.

```
FOR ij=1 TO iNumBits STEP iVarBits
  b1$=MID$(GA0$(iParent1),ij,iVarBits)
  b2$=MID$(GA0$(iParent2),ij,iVarBits)
  IF b1$=b2$ THEN
    GA0$(0)=GA0$(0)+b1$
    ELSE
    RNUM%=INT(ROUND(RND,0))
    IF RNUM%=0 THEN GA0$(0)=GA0$(0)+b1$ ELSE GA0$(0)=GA0$(0)+b2$
    END IF
  NEXT ij
```

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 (see code snippet).

```
FOR ij=1 TO iNumBits
  b1$=MID$(GA$(iAC),ij,1)
  RNUM%=INT(ROUND(RND*ProbFctr/2.,0))
  IF RNUM%<1 THEN
    IF b1$="0" THEN b1$="1" ELSE b1$="0"
    END IF
  GA$(0)=GA$(0)+b1$
  NEXT ij
```

Applied to all 48 bits of the chromosome string used herein, an individual's probability of mutation is found to be:

$$P_{mutaion} = 1 - \left(1 - \left(\frac{1}{probfctr}\right)\right)^{numbits}$$

In Section 2.5.10 it was suggested that, for uniform crossover, the mutation rate should equal *((1+L)/2NL)*. For L=48 bits and N=500 individuals per generation, this would be a bit mutation rate of one in a thousand (probfctr=1000). This was used as the default in this study, and provides a 5% chance of an individual having one or more mutations.

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 mathematical terms a sequence of steps *converges with order r* when *r* is the *asymptotic convergence rate* and represents the largest number for which

$$0 \le \lim_{k \to \infty} \frac{\left\| x_{k+1} - x^{*} \right\|}{\left\| x_k - x^{*} \right\|^{r}} \le \infty$$

where x^* is the "true" optimal solution and x_k is the calculated value for iteration k. If the convergence rate r equals one, the sequence exhibits linear convergence meaning that the solution is approached in a fashion in which the ratio between the current solution's error and the previous solution's error is either reducing or staying roughly constant, not increasing towards infinity as you approach the final solution. If $r=2$, the convergence is quadratic, etc.

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 as defined above 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 of any order r. 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 (see code snippet below) to a more-intuitive measure spanning a range from zero if purely random, to 100% Bit-String Affinity if all individuals are identical.

```
BitAffin=0
FOR iBitNum=1 TO iNumBits
 AvValBit=0
 FOR iAC=1 TO iPopultn
   AvValBit=AvValBit+
     val(MID$(GA$(iAC),iBitNum,1))
   NEXT iAC
 AvValBit=AvValBit/iPopultn
 IF AvValBit<.5 THEN
    BitAffin=BitAffin+AvValBit
    ELSE
    BitAffin=BitAffin+(1.-AvValBit)
    END
 next iBitNum
BitAffin=BitAffin/iNumBits
BitAffin=100(1.-2.*BitAffin)
```

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, and a journal article is being submitted detailing this concept.

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.

A final observation - the size of the population somewhat affects the initial calculated value of Bit-String Affinity. A fairly small population will never appear to be purely random (Bit-String Affinity=0) because the random fluctuations will cause averages of bit position values to be, not exactly 0.5 as in an infinitely-large random population, but values slightly under or over 0.5. In either case, the slight random distance from 0.5 is added to the Bit-String Affinity calculated value. This results in a value of perhaps 2-5% for a random population depending on the number of bits and the size of the population. This does not compromise the intended usage because convergence can be readily seen as the Bit-String Affinity goes from this small initial value to 50% or higher in a successful run.

4.4 Monte Carlo Random Search

A Monte Carlo optimization method as described in Section 2.5.6 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.

Using the binary design variable definition, there are 2^{48}, or about 2.815×10^{14}, different possible aircraft descriptions, so even 10,000 aircraft defined and analyzed represents only a small fraction of the total design space.

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 in Section 4.3.5. Methods employed for the selection of "parents" are described below.

4.5.1 Roulette Selection

Holland[52] 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 17 (based on actual data from a fighter optimization run conducted for this research). Sizes of the slots are calculated as:

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

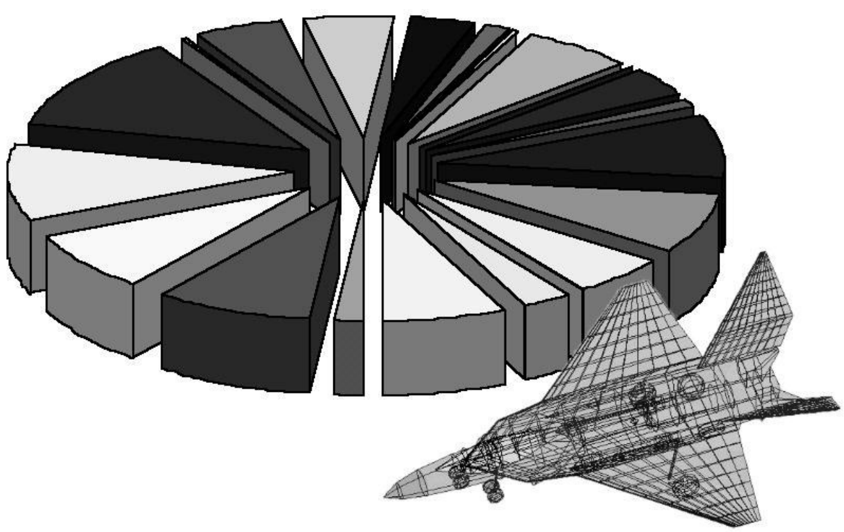


figure 17. Roulette Selection

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 by the weighting scheme described in Section 4.3.3.

In the implementation used herein, the MOM sizing of the "slots" can also be weighted to favor the higher values of MOM, as described in Section 4.3.2. This can be used to prevent the randomness of the selection process from overpowering the "goodness" of a favorable value of the measure of merit. Roulette selection is illustrated in the following code snippet.

```
SumPie=0.
 for iAc=1 to iPopultn
   GOSUB WT4MOM    'sets weight using linear, square, 4th or cosine
   SumPie=SumPie+(1-WtMOM)   '1- so that high mom=large pie piece
   next iAc
  WholePie=SumPie
  END IF
.
SumPie=0.
  X#=RND       '0-1 random #
  iAc=1        ' number of aircraft selected by roulette
  WHILE SumPie<X#
   GOSUB WT4MOM    'sets weight using linear, square, 4th or cosine
   SumPie=SumPie+(1-WtMOM)/WholePie
   iAc=iAc+1
   WEND
```

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 18. This can be done with or without replacement (see Section 4.3.4), and can include both a randomness and a MOM weighting as described above.

As implemented herein, each "winner" pair produces two offspring by two independent crossover operations (not by using "leftover" genes). This creates a new population that is as large as the previous population. Appendice C contains a complete printout of a sample Tournament optimization run (limited to two generations of 10 individuals each).

Tournament Selection as implemented herein can be seen in the code snippet below.

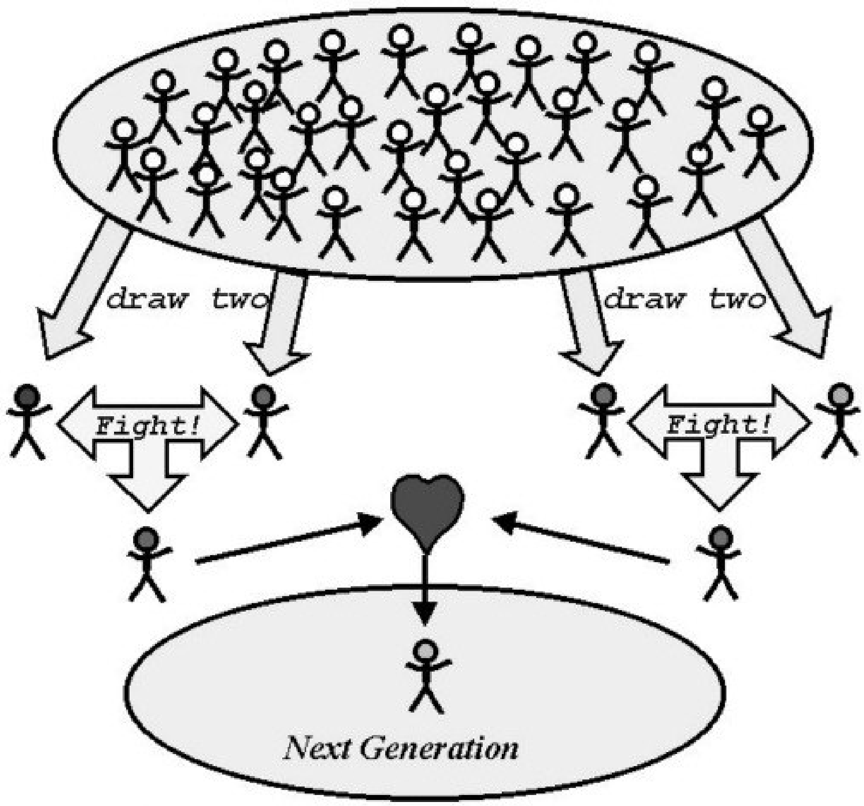


figure 18. Tournament Selection

```
FOR iAc=1 TO iPopultn STEP 2
  GOSUB FIGHTEM     '1v1 tournment – see below
  iParent1=idum1
  GOSUB FIGHTEM
  iParent2=idum1
  GOSUB Mate2       'crossover operator
  GA$(iAc)=GA0$(0)  'set child into next generation
  IF iAc<iPopultn THEN   'don't exceed iPopultn in GA$(iAc)
    GOSUB Mate2
    GA$(iAc+1)=GA$(iAc) 'second child from this pair
    END IF
  NEXT iAc
.
FIGHTEM:
  iHope1=INT(RND*(iPopultn-1-ibGone))+1
  iHope2=iHope1                    'ibGone is #removed so far
  WHILE iHope2=iHope1              'so iHope1 not picked twice
   iHope2=INT(RND*(iPopultn-1-ibGone))+1
   WEND
  IF abs(xMOM(iHope1))<abs(xMOM(iHope2)) then idum1=iHope1 _ 'winner
                                   ELSE idum1=iHope2
   RETURN
```

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.

In nature, the individuals selected to breed still remain in the breeder pool available for future selection (excepting species such as Black Widow spiders). This is simulated by using *replacement* as described above. The breeder pool scheme is shown in figure 19, followed by a code snippet.

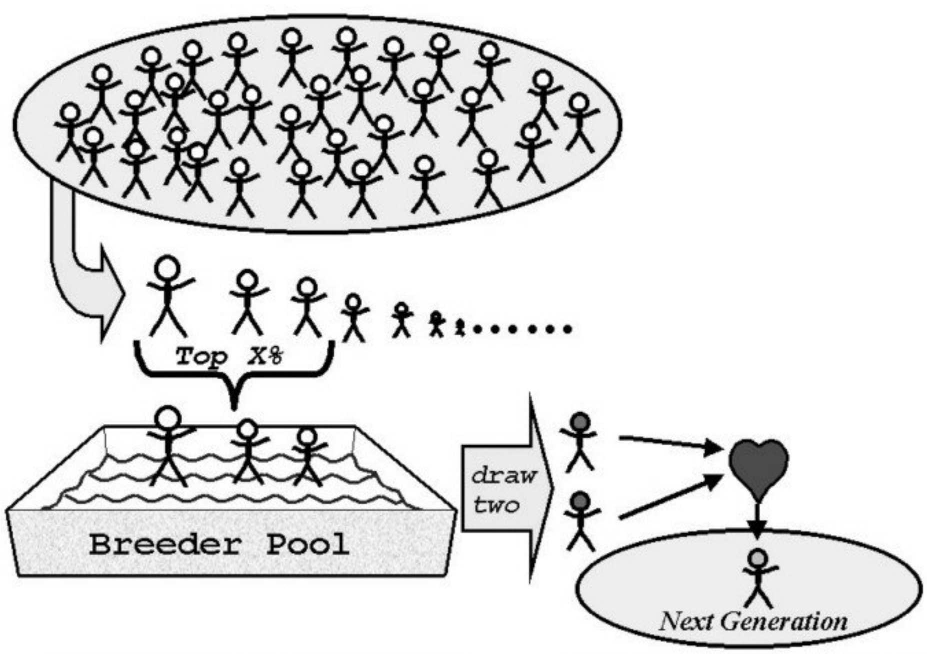


figure 19. Breeder Pool Selection

```
FOR ix=1 TO iPopultn          'stack best to worse
 BestMOM=BIGNUM               'initial best mom
 FOR iAc=1 TO iPopultn        'find best remaining
   GOSUB WT4MOM               'sets weight using linear, square, 4th or cosine
   IF CrntMom<BestMOM THEN BestMOM=CrntMom
   IF BestMOM=CrntMom THEN ibBest=iAc       'number of new best
   NEXT iAc
 IF BestMOM<BIGNUM AND BestMOM>0. THEN
   GA$(ix)=GA0$(ibBest)
   xMOM(ibBest)=-99      'so this one not picked again
   ELSE
   EXIT IF          'found all
   END IF
  NEXT ix

FOR iAc=1 TO iPopultn     'copy selected population strings to GA0$()
  IF iAc<=int(iPopultn*iTopPick/100) THEN
    GA0$(iAc)=GA$(iAc)    'only lets top iTopPick % go into breeding
    ELSE
    GA0$(iAc)=""
    END IF
  GA$(iAc)=""
  next iAc

FOR iAc=1 TO iPopultn                          'random pick from breeder pool
  iparent1=INT(RND*(iNumMate-1))+1             ' iNumMate is number in breeder pool
  iparent2=iparent1
  WHILE iparent2=iparent1
    iparent2=INT(RND*(iNumMate-1))+1
    WEND
  GOSUB Mate2
  GA$(iAc)=GA0$(0)
  NEXT iAc
```

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* as described in Section 2.5.9, 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[50]. This approach is depicted in figure 20, followed by a code snippet.

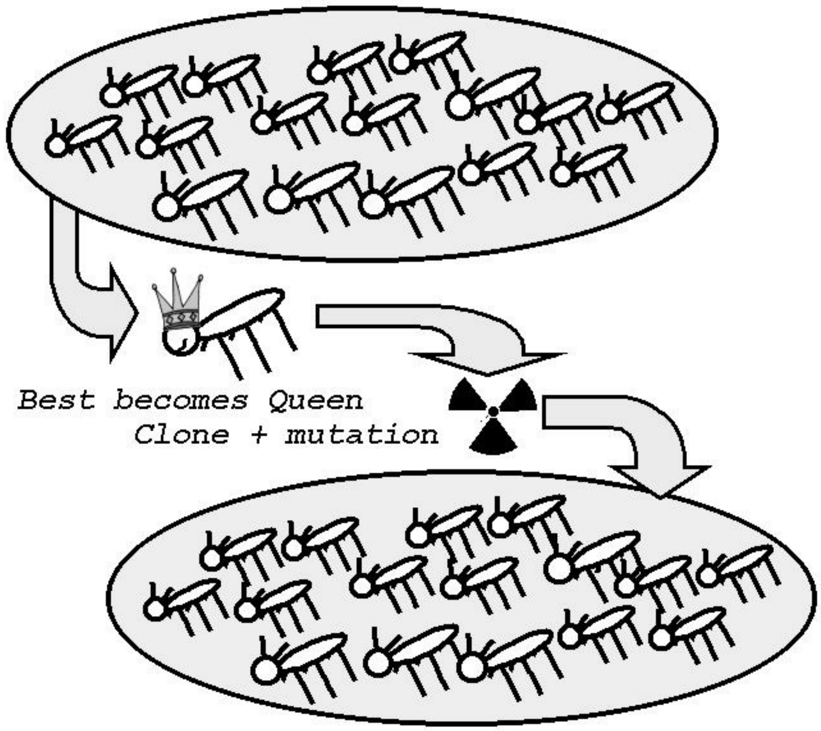


figure 20. "Killer Queen" – Best Self-Clones with Mutation

```
GOSUB Top10               'finds best one and puts in TopTen$()
FOR iAc=1 TO iPopultn      'make rest all like Queen, then mutate
  GA$(iAc)=TOPTEN$(1)     'entire next population are copies of Queen
  GOSUB MUTATE1           'mutates GA$(iAC), places in GA$(0)
  GA$(iAC)=GA$(0)
  next iAc
```

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[70]. 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*[11].

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[71,83,85,86] 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.

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. 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

As described in Section 2.6, 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. Such procedures were including in the huge sizing optimization codes of the major aircraft companies[74], but with set up times of nearly a month these methods do not seem feasible for conceptual-level MDO.

In the context of an MDO code, automated aircraft redesign is really a set of procedures for revising the analysis inputs. This could be done by actually modifying, via computer algorithm, the 3-D CAD file defining the aircraft geometry, then extracting from that revised geometry the inputs needed for analysis. This would be required if CFD or structural FEM were being used for analysis, but is probably not needed for the analysis methods typical of today's conceptual design efforts. Certainly, the analysis inputs of the RDS-Professional code as used herein do not require such a laborious procedure.

Instead, the automated aircraft redesign methods can be applied directly to the analysis input file data. For example, in the classical wing aerodynamic analysis of RDS the wing input data include wing reference area, actual exposed wing area, aspect ratio, taper ratio, thickness ratio, sweep, design lift coefficient, skin roughness parameter, estimated laminar flow, and key airfoil data ($C_{l\text{-}max}$ and leading edge parameter ΔY). For the fuselage, input data include length, equivalent diameter, wetted area, aft-end upsweep angle, and frontal areas of any windshield or aft-facing base areas.

These input data can be directly manipulated by automated aircraft redesign procedures of varying degrees of sophistication. In a simple implementation, wing area and fuselage diameter could be changed with no regard for mutual interactions. In a better implementation, the change in exposed wing area resulting from a change in fuselage width could be estimated and the input file revised.

In preliminary work in this area[6], this author defined a simple but reasonable set of procedures for such automated redesign. To improve realism, these automated redesign procedures have been expanded and enhanced. Even with a better set of such procedures this author believes that an experienced designer should always make a final layout following analytical optimization.

A related subject – in some cases the parametric variations in design variables may yield an aircraft that is not acceptable for practical reasons. For large airliners such as the Airbus A-380, the wingspan should not exceed the available ramp space at major airports, limiting the A-380 to a span of 262 feet. An MDO-optimized concept with wingspan in excess of this would probably be unacceptable to the customers, so some means of preventing the selection of such a design must be included in the optimizer. These geometric design constraints are in addition to the performance-based constraints previously discussed.

Selection of the design variables and the automated aircraft redesign procedures and geometric constraint methods used for this research are described below.

5.1 The Basic Six (or Five) Design Variables

In prior published work (Raymer[27], see also Appendices to this report), 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*

[†††] RDS uses the equivalent Sears-Haack body method to estimate supersonic wave drag, in which total aircraft length and maximum cross-sectional area are key inputs. This adjustment allows rapidly revising this drag estimate as wing geometry is changed.

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). Numerous books such as the classic Hoerner Fluid Dynamic Drag[75] indicate an optimum fineness ratio around 3-4 for minimizing drag in subsonic flight. However, this "optimal" fineness ratio assumes a constant frontal area or diameter - in other words, "what length minimizes total drag given a certain maximum diameter". A more important question for most aircraft is, "what fineness ratio minimizes drag for a given total *volume* enclosed?"

In a recent study[76] this author wrote a program to vary fineness ratio of a streamlined shape and calculate the resulting wetted surface area and the drag "form factor". This is a term in classical drag analysis that accounts for the pressure drags on the back of a body as a result of viscous separation. The product of wetted area and form factor, times a flat-plate skin friction coefficient, gives the total drag.

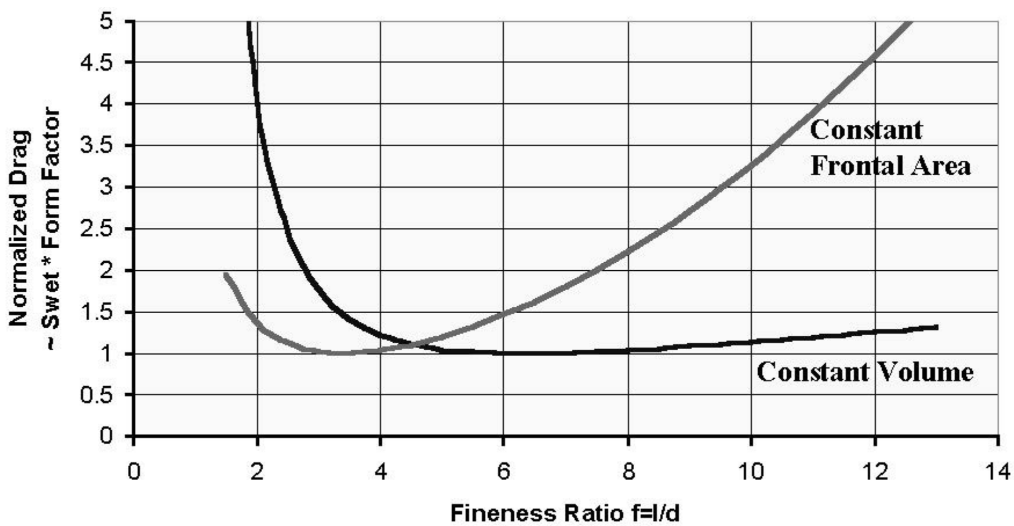


figure 21. Optimum Fuselage Fineness Ratios

Results are shown in figure 21 as fineness ratio is varied with two different assumptions, constant frontal area (i.e., diameter), and constant volume. The results are normalized to 1.0 to illustrate relative merit. As can be seen, the constant diameter assumption gives an optimum of 3-4 just as suggested by the old books and manuals. On the other hand, if volume is held constant then the optimum is somewhere between 6 and 8 - quite a different result! If an aircraft is volume-tight and is designed using the old suggested values of 3-4, the fuselage drag will be about 25-50% higher than possible with a fineness ratio of 6.0, according to this analysis.

This ignores structural effects on the fuselage, which may push the multidisciplinary optimum solution towards a lower value. 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.

During conceptual design $C_{L\text{-}design}$ has often been chosen by the designer or chief aerodynamicist based on experience with similar aircraft. Aircraft drag during cruise will be minimized if the aircraft cruises at approximately its wing design lift coefficient, calculated by:

$$C_{\text{L-design}} = \frac{W/S}{q}$$

Ideally, one could select wing loading *W/S* to provide the desired $C_{L\text{-}design}$. However, wing loading must often be set to a lower value (larger wing) to obtain the desired stall speed or takeoff/landing performance. Also, extreme values of $C_{L\text{-}design}$ provide poor values of airfoil lift-to-drag ratio. Since $C_{L\text{-}design}$ has an effect on maximum lift, it and *W/S* should be determined together.

Thus, $C_{L\text{-}design}$ 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[77] (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 (see Raymer[11]). 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}), resulting in the following constraint equations:

Subsonic: $$A_{max}=10.^{(1.047-0.552*\tan(\Delta_{QC}))}$$

Transonic: $$A_{max}=10.^{(0.842-0.435*\tan((\Delta_{QC}))}$$

These equations were added to the RDS MDO routines as an optional geometric constraint option. During optimization, the appropriate equation is used to calculate the maximum allowable aspect ratio for the design's wing sweep. If that value is exceeded, the design is penalized in the same manner as an airplane missing a performance requirement, as described above.

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[78]) 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.

In conceptual design, a layout is prepared that shows the overall aerodynamic configuration including fuselage, wings, tails, and the like, and also shows the major internal items such as the engine, landing gear, fuel tanks, avionics, and the payload, passenger compartment, and crew station. This drawing cannot and does not show every item that will be in the final, as-built aircraft. Many items are simply too small to worry about in the initial layout process, although taken together they add up to substantial volume. Often, internal components have yet to be designed, and will not be designed until the later "detail design" phase. Examples include the actual aircraft structure, equipment items such as actuators and environmental control, and the ducting and wiring that pass throughout the aircraft.

An experienced aircraft designer knows to provide a generous amount of "un-spoken-for" volume in the aircraft, spaced properly throughout the aircraft. The correct amount of extra space just "looks right". Such an intuitive measure of merit is difficult to duplicate or teach, and impossible to program. For this reason, attempts have been made to provide an analytical evaluation of the "right" amount of extra volume. Over 30 years ago, Cadell[79] looked at a volumetric density evaluation to determine a reasonable volume allocation.

More recently, O'Brimski[80] reviewed the volumetric density of fighter-type aircraft and determined a practical limit on total aircraft density to assist in evaluation of aircraft proposed to the US Navy. His methodology was simple, based on a graph similar to figure 22 of aircraft total internal volume (excluding inlet duct) versus takeoff weight with full internal fuel and load. The actual graph has distribution restrictions, but overall his data indicates a maximum practical density ranging from about 33 lb/ft^3 for small fighters to 31.5 lb/ft^3 for larger fighters.

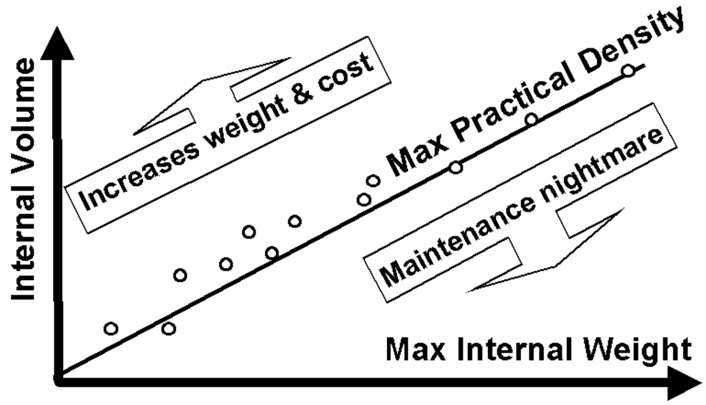


figure 22. Traditional Volumetric Density Graph

This methodology is suitable for analysis of aircraft proposals but less useful for conceptual design optimization purposes, for two reasons. First, it is based on detailed evaluation of the actual volume plot of the completed design, which is not normally available for each alternative of a parametric study. Second, it is insensitive to some of the likely optimization variables. For example, a jet engine with higher bypass ratio occupies more volume, requiring an increase in fuselage size when the concept is properly drawn. Such an engine burns less fuel, reducing the volume required. These would not be addressed by this earlier method.

5.5.1 Definition Of Net Design Volume

Net Design Volume is a design metric developed to provide a similar, simple method of ascertaining design realism, with enough extra detail to account for the differences between different aircraft and between parametric variations of a design undergoing optimization.

Net Design Volume is defined as the internal volume of an aircraft fuselage, nacelles, and wings, less the volume dedicated to fuel, propulsion, payload, passengers, and crew. *NDV* more closely represents this "un-spoken-for" extra volume that the designer knows will provide enough space for everything else, including aircraft structure and undefined items such as avionics, systems, equipment, landing gear, routing, and access provisions. Volume of the tails and pylons are not included because those items rarely provide usable volume (although some aircraft do put fuel in the tails). Also, tightly packed separate podded nacelles have no extra volume beyond that used for propulsion, and so separate nacelles are excluded from the evaluation of *NDV*.

$$\boxed{NDV = \{Vfus + Vwing\} - Vfuel - Vppc + \underbrace{Nengines\{Vnacelle - Veng - Vduct - Vtailpipe\}}_{\text{Ignore if separate nacelles}}}$$

Internal weapons bays, not addressed in figure 22, are accounted for in *NDV* by the subtraction of that internal volume. This makes it easier to compare designs with and without such bays, such as modern stealth fighters.

From *NDV*, a density is then calculated based on the weights associated with the *NDV* areas. Basically, this W_{ndv} is the empty weight less fuel and engine weight. Since tails, pylons, and separate podded nacelles are excluded from *NDV*, their weights are removed. Definition of fuel and payload are not required because both are excluded from *NDV*.

The proper calculation of *NDV* would use a detailed aircraft volume plot. This would be a suitable approach for evaluation of a conceptual design layout in industry or government design offices. For *NDV* to be a usable tool in MDO computer programs, a simpler method of calculating *NDV* is required, provided that it still provided a reasonable approximation of the effects of parametric variations in design concept.

A simplified volumetric calculation was defined as follows. Fuselage volume was estimated using three segments. The nose was assumed to be ellipsoidal in cross-section and planview, with length equal to 1.5 times height (2.5 for supersonic designs). A constant-section center section was assumed, based on a tail length defined as 2.5 times height, or 60% of length for supersonic designs. The tail was defined as an ellipsoidal shape unless the back is cut off as for a rear-mounted engine, in which case a straight taper was assumed. This is illustrated in figure 23, with a code snippet below.

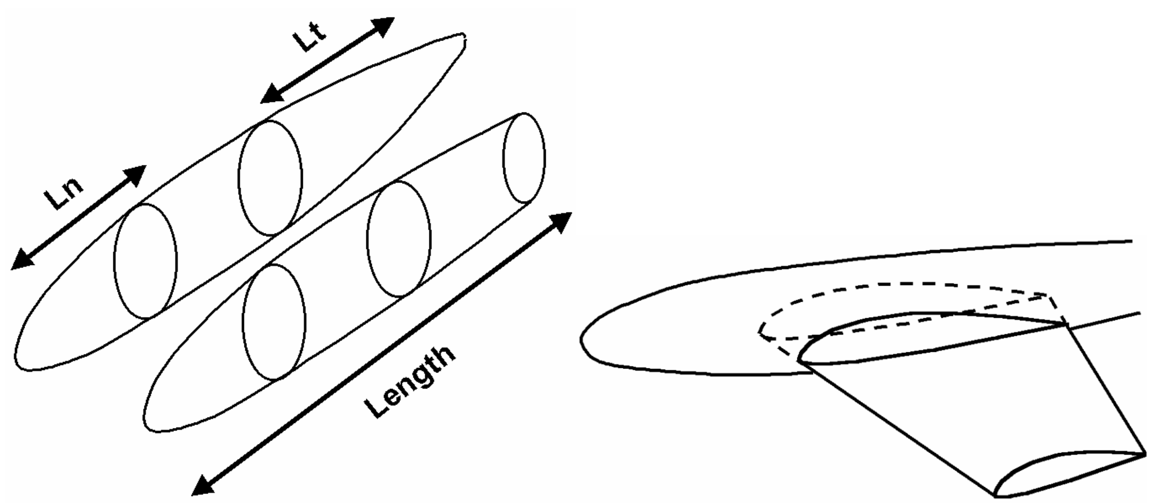


figure 23. Fuselage and Wing Volume Estimates

```
IF ibsuper=1 THEN
  LN=2.5*Htfus
  LT=.6*LENGTH
  IF LN>.4*LENGTH THEN LN=.4*LENGTH
  ELSE
  LN=1.5*Htfus
  LT=2.5*Htfus
  IF LN+LT>LENGTH THEN LN=LENGTH-LT
  END IF
Amax=Xcirc*Wdfus*Htfus/4.
Aeng=NumEng*3.14*Deng*Deng/4.
Vfus=(.48*Amax*LN) + (Length-LT-LN)*Amax
IF ibackeng=0 THEN   'tapered tailcone
   Vaft=0.48*Amax*LT
   ELSE
   Vaft=0.33*LT*(Amax+Aeng+SQR(Amax*Aeng))
   END IF
Vfus=Vfus+Vaft
```

Wing volume was estimated by determining the areas of the tip airfoil and the airfoil at the root, where the wing meets the fuselage (not the theoretical root at the center of the airplane). Airfoil area, based on geometric data for a number of NACA sections, was approximated as 0.67 times the chord length and thickness. Volume was determined by integration assuming a straight taper between these end airfoils (see code snippet).

```
Croot=2.*Swing/(span*(1.+taper))
Ctip=taper*Croot
Cside=Croot*(1-Wdfus/Span) + Ctip*Wdfus/Span
Sside=.67 * ToverC * Cside*Cside
Stip =.67 * ToverC * Ctip *Ctip
Vwing=0.333*(Span-Wdfus)*(Sside+Stip+SQR(Sside*Stip))
```

Propulsion volume for an internal jet engine was estimated from the engine diameter and length, and the inlet length. The engine front face diameter is typically about 80% of engine maximum diameter, and is the diameter of the back of the inlet. Thus, the area of the back of the inlet is roughly 0.64 times engine maximum area. To account for expansion of the inlet from front to back, this factor was reduced to 0.6 and used to estimate inlet volume.

Volumes of the payload (including internal weapons bays), passenger compartment, and crew compartment were directly assessed by measurement off design layouts. Fuel volume was estimated from fuel weight, applying standard fuel density values and an appropriate installation factor (88% installation factor was assumed).

To evaluate whether this definition of *Net Design Volume* and the approximate volumetric analysis as outlined above have validity for actual aircraft, seven representative modern fighter aircraft were subjected to comparative regression analysis using the author's least-squares multivariable nonlinear regression program (see Raymer[78]).

The selected aircraft are as follows:

- F-15A
- F-16A
- F-18A
- Gripen
- Rafale
- Typhoon
- F-22

These aircraft were modeled using data from *Jane's All the World's Aircraft*[81], *Aviation Week and Space Technology* weekly magazine (various issues), and miscellaneous other sources, as detailed in Raymer[78]. The earliest versions of these aircraft were deliberately used under the assumption that these are closest to the original design layout and hence are most representative of the original designers' definition of "un-spoken-for" volume. A concept that was designed with substantial unused growth volume would violate this assumption. The F-15A, which seems low in density in the following analysis, is known to have been designed with substantial growth in mind, including unused internal volume.

The data indicate that the average *NDV* density (W_{ndv} / *NDV*) equals about 34 pounds per cubic foot. Compared to data in O'Brimski[80], this *NDV* density is a bit greater than the broader density measure of volume less inlet duct. This is possibly due to the elimination of canopy and tails from *NDV*, both of which are less dense than the rest of the aircraft (canopy being full of air).

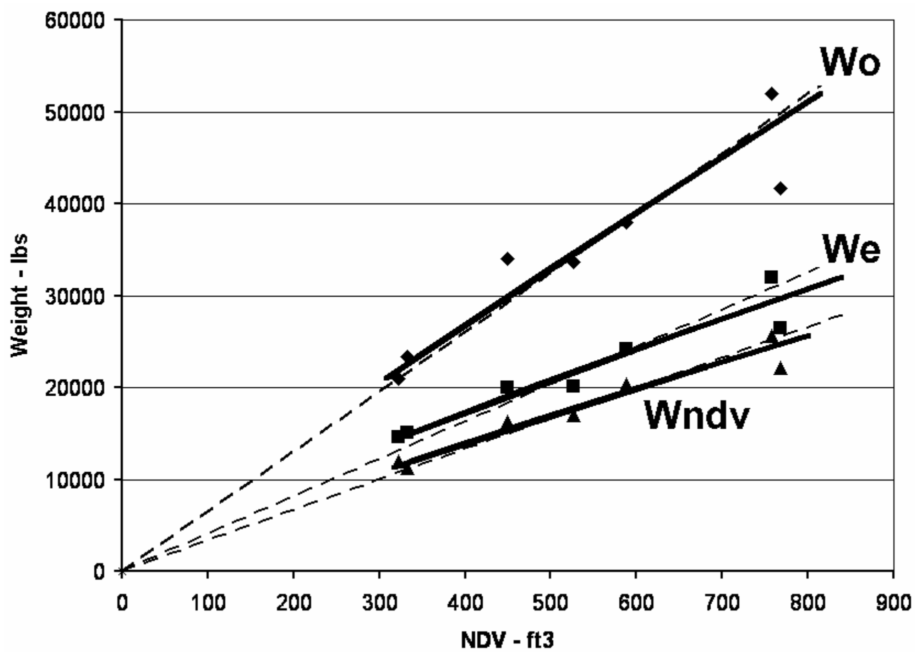


figure 24. NDV Regression and Ratio Trendline Comparison

Regression analysis created the curves illustrated in figure 24, which show weight versus calculated value of *Net Design Volume* as described above. Weights are shown three ways – total takeoff weight *(W_0)*, empty weight *(W_e)*, and *Net Design Volume* weight

(W_{ndv}). As discussed above, *Net Design Volume* weight excludes fuel and engine weight as well as tails, pylons, and separate podded nacelles.

Regression analysis indicated that W_{ndv} does closely predict *NDV* in spite of the simplifications employed. Also, W_{ndv} provides the best fit to estimate *NDV* with a correlation coefficient of 0.980. Using W_e to estimate *NDV* yields a coefficient of 0.971, while using W_0 gives a value of 0.959.

Simple ratio trendlines can also be seen in figure 24, showing that the regression results are very similar to the ratio trendlines. Also, the *NDV* density is seen to slightly decrease with increasing aircraft weight versus the constant-density trendlines (dotted). This seems odd, and this author would have expected the opposite, but the same result was seen in the trendline in O'Brimski[80].

In defining *NDV*, the treatment of avionics volume was of concern. Ultimately it was decided to include avionics in *NDV* (in other words, not attempt to subtract out avionics volume and weights). Certainly the opposite could have been done, because the avionics volumes and weights are included on the first design layouts. It was decided to include avionics in *NDV* to simplify the workload, and because, with an average avionics density of 30-45 lb/ft^3, avionics has about the same density as our estimated *NDV*. Thus, it can be included or not with little change in result. This may not be true for other classes of aircraft, but for them the total avionics weight is rarely as substantial a fraction of We as for fighters.

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, as was the intention of O'Brimski[80]. 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)

This application of *NDV* as a geometric design constraint entails a three-step process as suggested by figure 25. First, the parametric changes are made to design variables such as sweep, taper ratio, or in this case, wing loading (wing gets smaller). Second, the *Net Design Volume* is calculated and compared to the baseline value, and the fuselage is revised to hold it constant. In this case, the smaller wing requires a fuselage stretch to provide the same *Net Design Volume*, assuming that required fuel in the wing gets pushed into the fuselage. Finally, the revised design is analyzed and resized to the mission, and those results are used in the optimizer.

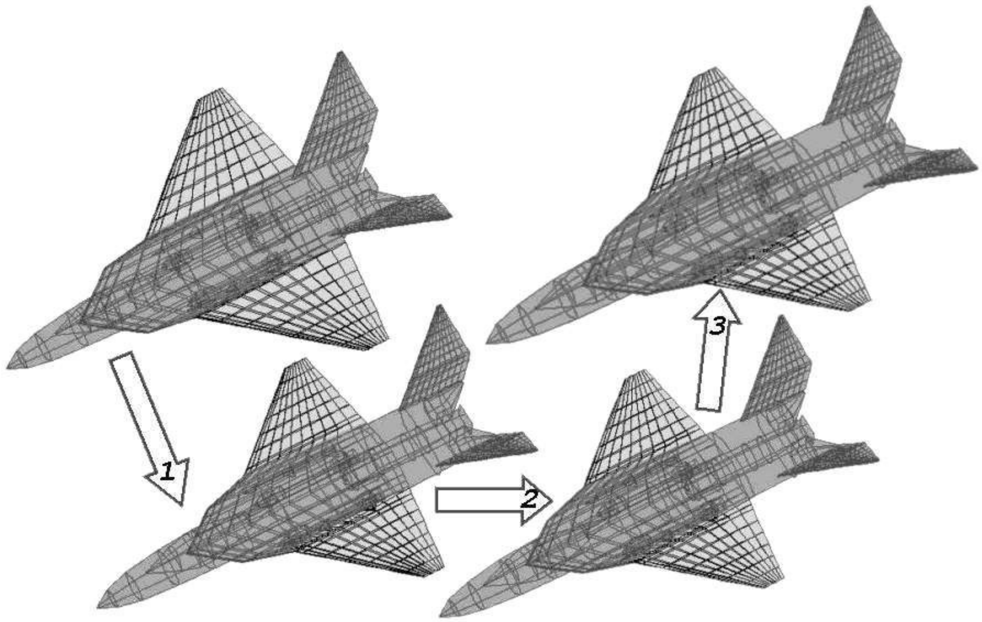


figure 25. Three-Step Process for Use of Net Design Volume

Net Design Volume is never used as a penalty function in the optimization, because it is always corrected prior to the analysis. The "penalty" is in what may be an excess fuselage size to attain the required *NDV* density, so it is already included in the calculation of the MOM and does not need to be further included as a penalty function multiplier.

This use of *Net Design Volume* may not be appropriate for some classes of aircraft. A general aviation aircraft it usually not volume-tight the way a fighter, bomber, or large airliner is. With an aircraft that is not volume-tight, there may be enough room in the wing for all the required fuel even if the wing is made smaller. In such a case, the hold of *Net Design Volume* should be turned off.

To avoid excessive fuselage "shrinkage" in the case of a large wing, a limit of 75% scaling of fuselage length was included in the method. It is also possible to specify that the fuselage can be made larger to maintain *NDV* density, but it should not be made any smaller even if the wing can carry more of the fuel. Often the designer has already made the fuselage as small as is practical even if some fuel can be moved to the wing.

5.6 Automated Redesign for Discrete Variables

More difficult to obtain are redesign procedures for the “discrete” variables, those that have only integer values such as number of engines or number of aisles and seats across. It was intended in this research to define automated procedures for aircraft redesign for use in MDO routines, but it quickly became apparent that any such procedures would be design-specific and limited to a single class of aircraft. Any routine capable of modifying a propeller-powered canard-pusher aircraft from single to twin engines would be useless for a design trade in which a twin-engine F-15-like fighter were to be studied with a single engine.

Similarly, an automated redesign procedure that would modify an Airbus-like transport from seven seats across to eight seats across would probably be inaccurate if applied to a corporate jet in which a study of three- vs. two-across seating were to be made.

To incorporate optimization of discrete variables into a general-purpose aircraft design program such as RDS-Professional would require a large number of alternative routines, one for each class and type of aircraft, and would still never be trustworthy for use in a type of aircraft not already considered. This is contrary to the philosophy employed in the rest of the RDS-Professional program and so was abandoned for now.

Traditionally these discrete variable design optimizations have been done by design trade studies in which an experienced configuration designer produced a new design layout incorporating the alternative selection. This will probably remain the best way to treat these discrete variables, although an organization doing numerous optimizations on similar designs may find it useful to code specific automated redesign procedures for the studies that they do often.

An alternative means of dealing with discrete variables in MDO is to have numerous quick layout studies done by experienced designers, then distill the results to numerical relationships via Response Surfaces and apply them to an MDO routine. Here, the question becomes – how well can designers do such alternative concepts in the limited amount of time available per concept?

The old approach – actually designing and analyzing alternative design concepts – is really not so bad. Each alternative concept can be designed and optimized separately, and the best of the best selected as the optimum design. However, the workload goes up exponentially as more discrete-variable trades are included.

(This page intentionally blank)

6 NOTIONAL AIRCRAFT CONCEPTS

To evaluate MDO codes for aircraft conceptual design, realistic samples of aircraft conceptual designs are required. One option would have been the use of data from actual aircraft, modeled as if they were in the conceptual design phase. This is problematic for several reasons. First, any attempt to reverse-engineer an actual aircraft and then perform MDO can have only two outcomes. One possibility is that the MDO finds no better of a design, in which case you were not able to fully exercise the MDO because the design was already optimal (assuming the code is working). Even worse, you may find a "better" design and thereby lower your own credibility or offend those who designed it in the first place, or both.

A more serious problem, though, is that in many cases it is difficult to determine the actual design mission and requirements of an aircraft previously designed. Such data is buried in company archives, and those who actually did the conceptual design are often unavailable. A design that was well optimized to one set of requirements would appear poorly done if evaluated to a different set of requirements. Without company data on both the initial requirements and the resulting design, the use of a "home-grown" model of an existing aircraft may be more misleading than helpful.

Instead, four new notional designs were developed for this research by this author. While not as fully developed as would be done for an actual design project, they were designed with reasonable concern for design realism and were developed and analyzed using the same methods and software that this author uses in his professional design contracts. Design requirements were defined by this author based on experience in similar projects.

6.1.1 Advanced Multirole Export Fighter

A single-engine multirole fighter was designed as an F-16 replacement for the USAF and as an export fighter for a worldwide market. It is an outgrowth of design approaches developed during the author's prior work for several corporations studying next-generation fighters (Raymer[82,83,84]) and further addressed in company studies of advanced carrier-based STOVL fighters (Raymer[85]). The actual design concept as shown below was specifically developed during this research as a test case for MDO (as were all of these concepts).

This aircraft has an as-drawn takeoff gross weight (W_o) of 45,000 lbs {20,412 kg} and an empty weight of 23,870 lbs {10,827 kg}, attaining a subsonic combat radius of about 850 nmi {1574 km} dropping two 1,000 lbs {454kg} weapons. The design has a length of 54 ft {16.5 m} and a span of 38.4 ft {11.7 m}, with a wing loading of 76.3 psf {372 kgsm} and aspect ratio of 2.5. It uses a single advanced technology afterburning turbofan of 32,000 lbs thrust {142 kN}.

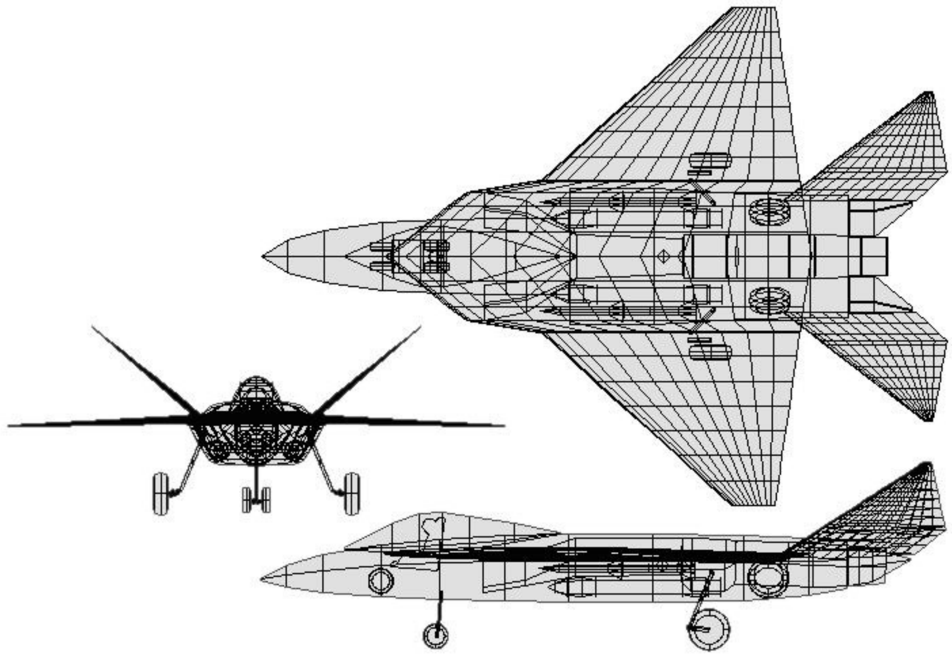


figure 26. Advanced Multirole Export Fighter

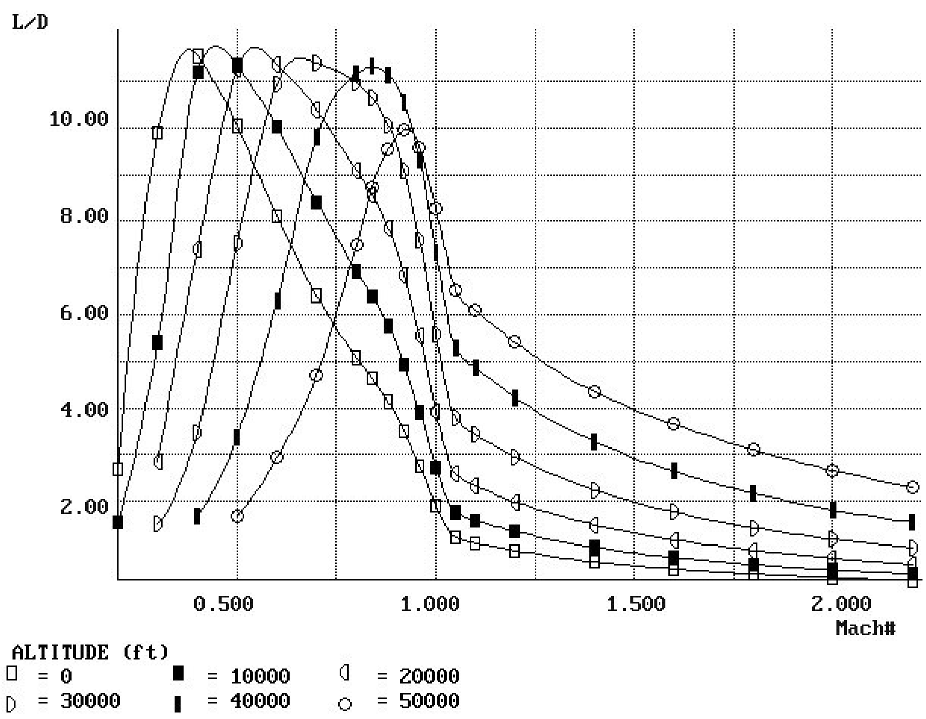


figure 27. Advanced Fighter - Estimated L/D

Estimated aerodynamic results are summarized in figure 27 showing lift-to-drag ratios (*L/D*) in level flight. Weights estimates are tabulated below. Payload includes two 1,000-lb air-to-surface weapons {454 kg} and two air-to-air missiles.

	Lbs	Kg
STRUCTURES GROUP	10229.5	4640.0
Wing	3089.7	1401.4
Vert. Tail	866.5	393.0
Fuselage	4627.6	2099.0
Main Lndg Gear	775.1	351.6
Nose Lndg Gear	318.1	144.3
Engine Mounts	63.3	28.7
Firewall	113.0	51.3
Engine Section	53.3	24.2
Air Induction	322.9	146.5
PROPULSION GROUP	6355.3	2882.7
Engine(s)	4930.0	2236.2
Engine Cooling	273.0	123.8
Oil Cooling	37.8	17.2
Engine Controls	21.2	9.6
Starter	66.4	30.1
Fuel System	1027.0	465.8
EQUIPMENT GROUP	4484.8	2034.3
Flight Controls	1020.8	463.0
Instruments	128.8	58.4
Hydraulics	171.7	77.9
Electrical	706.5	320.5
Avionics	1579.8	716.6
Furnishings	391.7	177.7
Air Conditioning	464.7	210.8
Handling Gear	20.7	9.4
MISC EMPTY WEIGHT	2800.0	1270.1
TOTAL WEIGHT EMPTY	23869.6	10827.1
USEFUL LOAD GROUP	21130.4	9584.6
Crew	220.0	99.8
Fuel	18000.4	8164.8
Oil	50.0	22.7
Payload	2860.0	1297.3
TAKEOFF GROSS WEIGHT	45000.0	20411.6
Design Gross Weight	36000.0	16329.3

figure 28. Advanced Fighter - Weights Estimates

As a design sample, this notional fighter offers a suitably complicated problem for sizing and performance analysis, with 13 mission segments and 6 must-meet performance points. The design mission is an out-and-back tactical attack mission, with a best-speed cruise at 40,000 ft {12,192 m} of 800 nmi {1482 km} each way. At mid-mission, a 50 nmi {93km} dash at 500 kts is done, the two attack weapons are dropped, and a three-turn maximum-afterburner combat is performed. Performance requirements include a 20 deg/sec corner speed turn at 15,000 ft {4572 m}, a sustained turn at 5 g's at Mach .8 and 15,000 ft {4572 m}, dash speed of M1.6 at 30,000 ft {9144 m}, and an acceleration from M.6 to M1 at 15,000 ft {4572 m} in 30 seconds. Takeoff and landing must be within 2000 ft {610 m}.

Optimization for this fighter concept was done using Purchase Price as the Measure of Merit, as estimated by the DAPCA equations described in the Appendices, Section 9.2.5.

The as-drawn baseline design has an estimated price of $34.7m, but does not yet meet all performance and range requirements.

Optimization using the traditional carpet plot technique is shown in figure 29. The shaded area represents the "feasible" region where all constraints are met. Increasing price is towards the top of the plot, so the best aircraft is found as the lowest point of the feasible region, which corresponds to a T/W of 0.72 and W/S of 70, with a purchase price (MOM) of $38.5 million. A separate calculation indicated that this optimum aircraft has a takeoff gross weight of 44,500 lbs {20,185 kg}, slightly below the as-drawn weight, when sized to the design mission using these performance-constrained optimal design parameters.

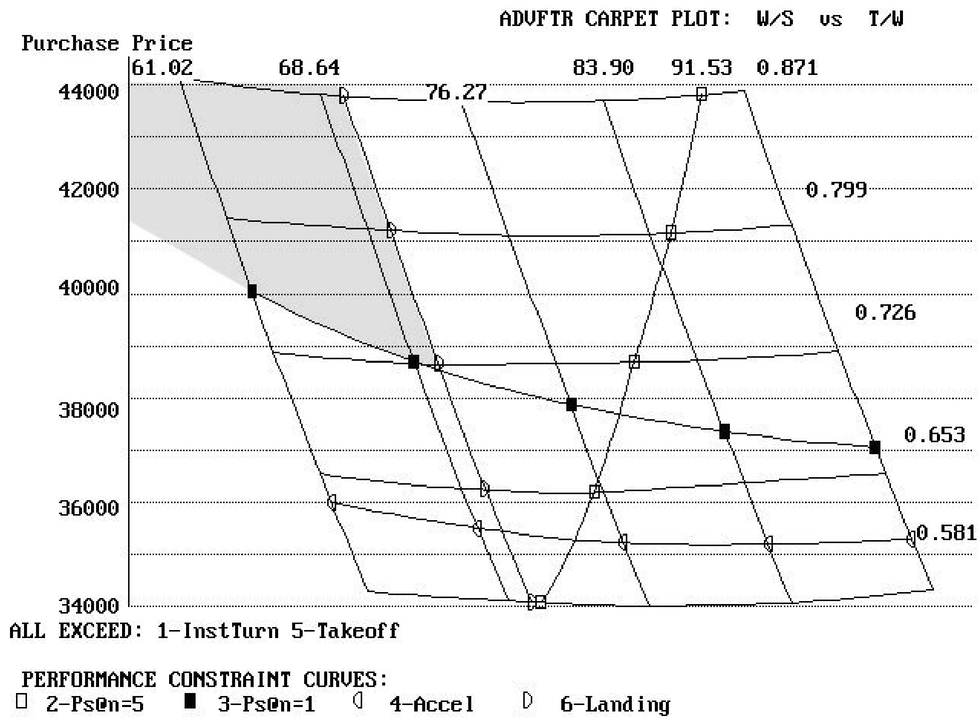


figure 29. Advanced Fighter - Carpet Plot Optimization

This classical optimization considered only two variables. A key question for this research is, "Would consideration of more design variables offer further improvements, and if so, which MDO methods are most efficient for such multivariable optimization?" Such results are presented in Section 7, below.

6.1.2 Civil Transport

A commercial transport intended to be reminiscent of the Airbus A321 was developed using published Airbus data and information from an analysis model previously used at KTH. Shown in figure 30, this concept is a twin-engine, single aisle design with a 13 foot wide {4m} diameter fuselage. It has an as-drawn takeoff gross weight (W_o) of 213,844 lbs {96,998 kg} and an empty weight of 114,210 lbs {51,804 kg}. It attains a range of about 3300 nmi {6112 km} plus reserves, carrying 188 passengers. The design has a length of 146 ft {44.5 m} and a span of 130 ft {40 m}, with a wing aspect ratio of 10.13

and an area of 1672 sqft {155 sqm}. Design requirements are typical for commercial transports and include takeoff and landing distances and various climb conditions, some with gear and flaps down.

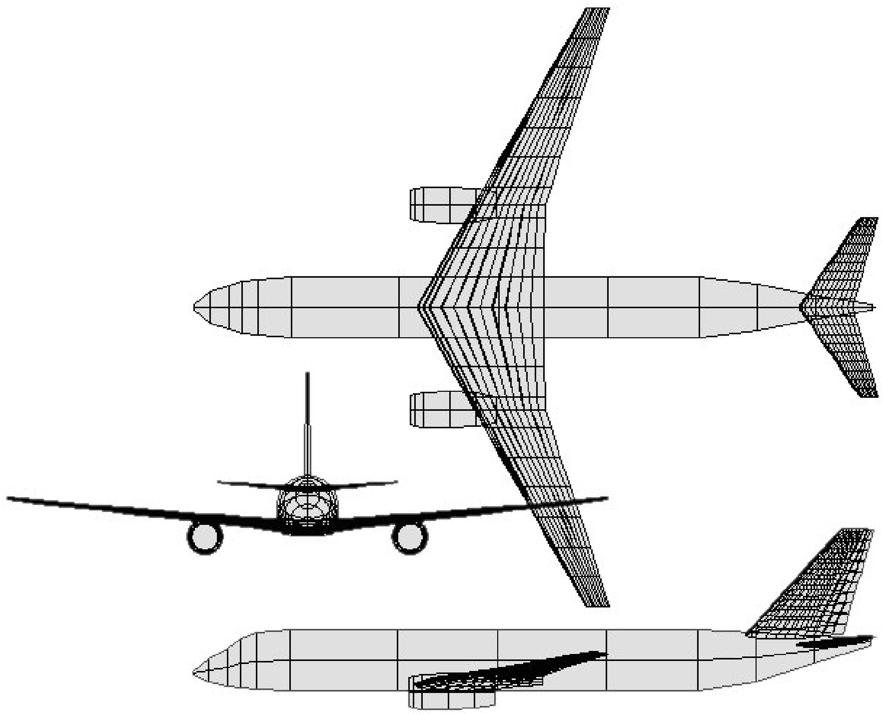


figure 30. Civil Transport

Engines were approximated by an existing commercial turbofan data set, adjusted to approximate the characteristics of the CFM-56 per data in Janes[81]. Nominal thrust per engine is 36,800 lbs {164 kN}.

As a deliberate test of the MDO routines, a badly optimized version of this commercial airliner was designed as shown in figure 31. The various MDO routines were tested for their ability to start from such a poorly-conceived design baseline and find a "normal", presumably optimal design. Specifically, the wing planform parameters were changed to values vastly different from the usual practice for jet airliners. Wing loading was changed from 128 to 107 psf {624 to 521 kgsm}, wing aspect ratio was changed from 10.13 to 5, and wing sweep was changed from 29 to 40 degrees. Such a wing is typical for an early jet fighter, not a transport!

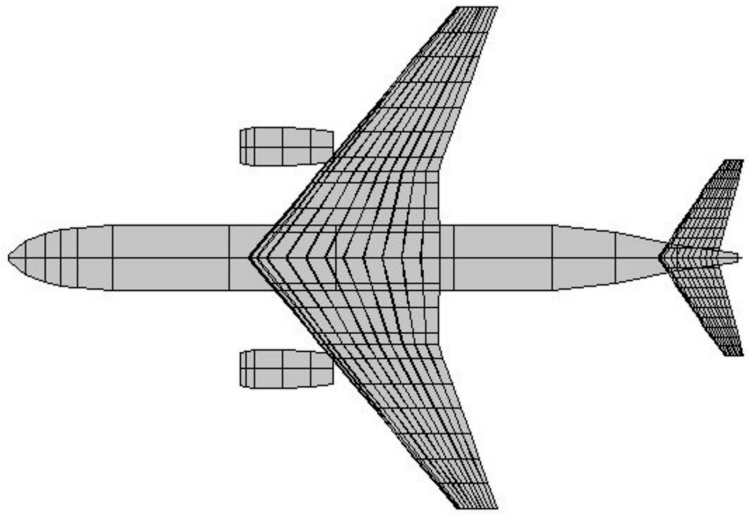


figure 31. "Bad" Civil Transport

Lift-to-drag ratios of both designs are shown below, indicating the poorness of the choice of wing design parameters for the "bad" design. It has a best L/D of only 15 compared with the "good" design's value of nearly 19, a 25% difference (recall that range is directly proportional to L/D). Weights estimates for the "good" design follow.

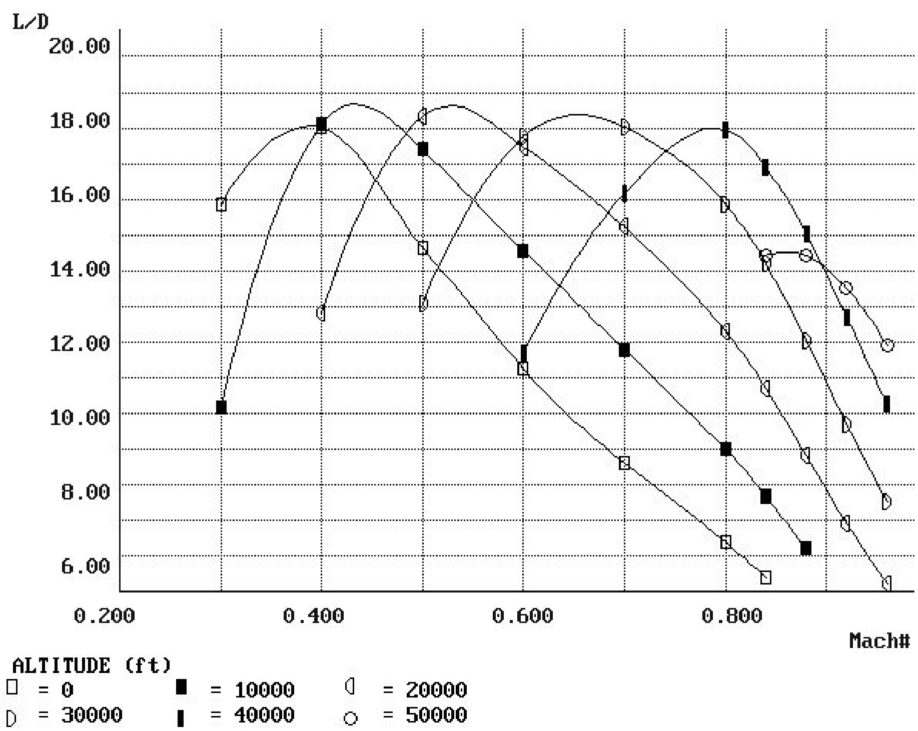


figure 32. "Good" Civil Transport - Estimated L/D

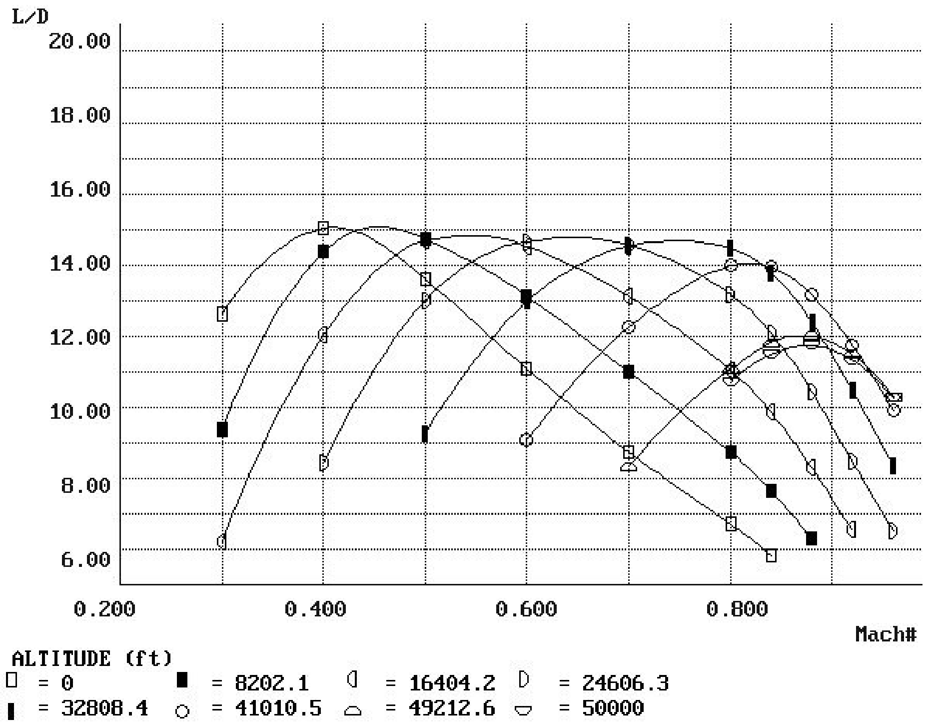


figure 33. "Bad" Civil Transport - Estimated L/D

STRUCTURES GROUP	58734.0	26641.3
Wing	21519.3	9761.0
Horiz. Tail	757.5	343.6
Vert. Tail	2833.9	1285.4
Fuselage	21957.5	9959.7
Main Lndg Gear	6600.8	2994.1
Nose Lndg Gear	1214.0	550.7
Nacelle Group	3851.2	1746.9
PROPULSION GROUP	17461.4	7920.3
Engine(s)	16755.1	7600.0
Engine Controls	45.2	20.5
Starter	226.0	102.5
Fuel System	435.0	197.3
EQUIPMENT GROUP	13102.7	5943.3
Flight Controls	2997.7	1359.7
Instruments	205.8	93.4
Hydraulics	279.0	126.5
Electrical	2834.8	1285.9
Avionics	2025.2	918.6
Furnishings	1570.4	712.3
Air Conditioning	2167.8	983.3
Anti Ice	427.7	194.0
Handling Gear	64.2	29.1
APU installed	530.2	240.5
MISC EMPTY WEIGHT	24912.3	11300.0
TOTAL WEIGHT EMPTY	114210.3	51804.9
USEFUL LOAD GROUP	99633.8	45193.1
Crew	1440.0	653.2
Fuel	65683.6	29793.6
Oil	110.2	50.0
Passengers	32400.0	14696.4
TAKEOFF GROSS WEIGHT	213844.2	96998.0

figure 34. "Good" Civil Transport - Weights Estimates

Classical carpet plot optimizations of these two designs are shown below. For the "good" design, an optimized takeoff gross weight of 193,000 lbs {87,643 kg} is found at the intersection of two performance requirements – takeoff distance and engine-out rate of climb.

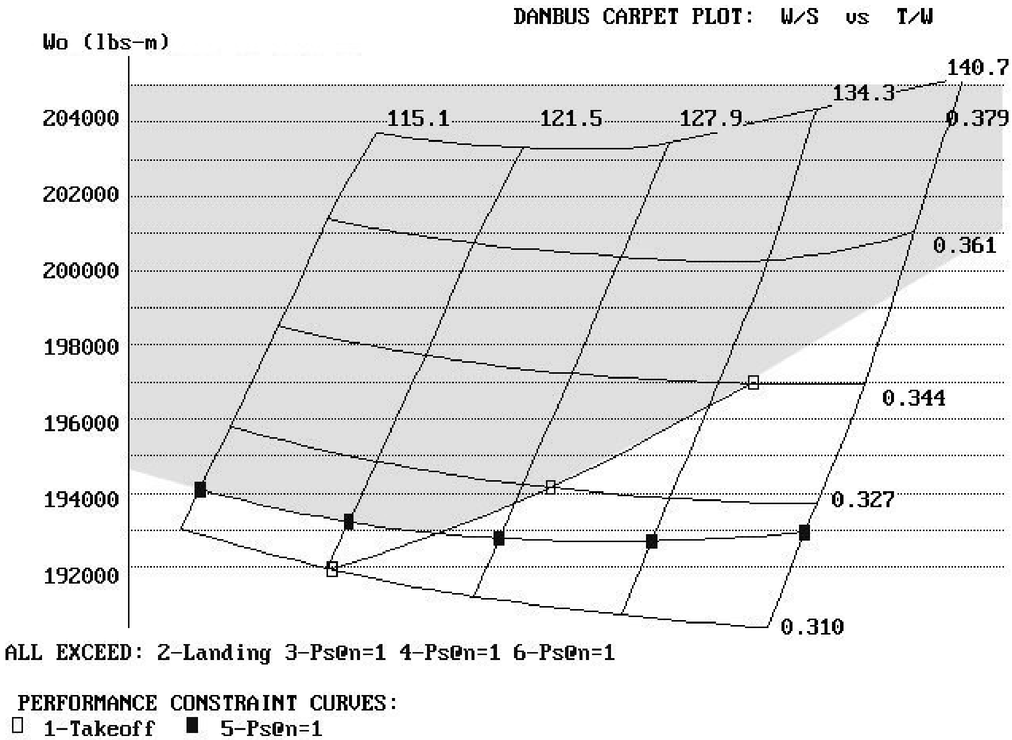


figure 35. "Good" Transport - Carpet Plot Optimization

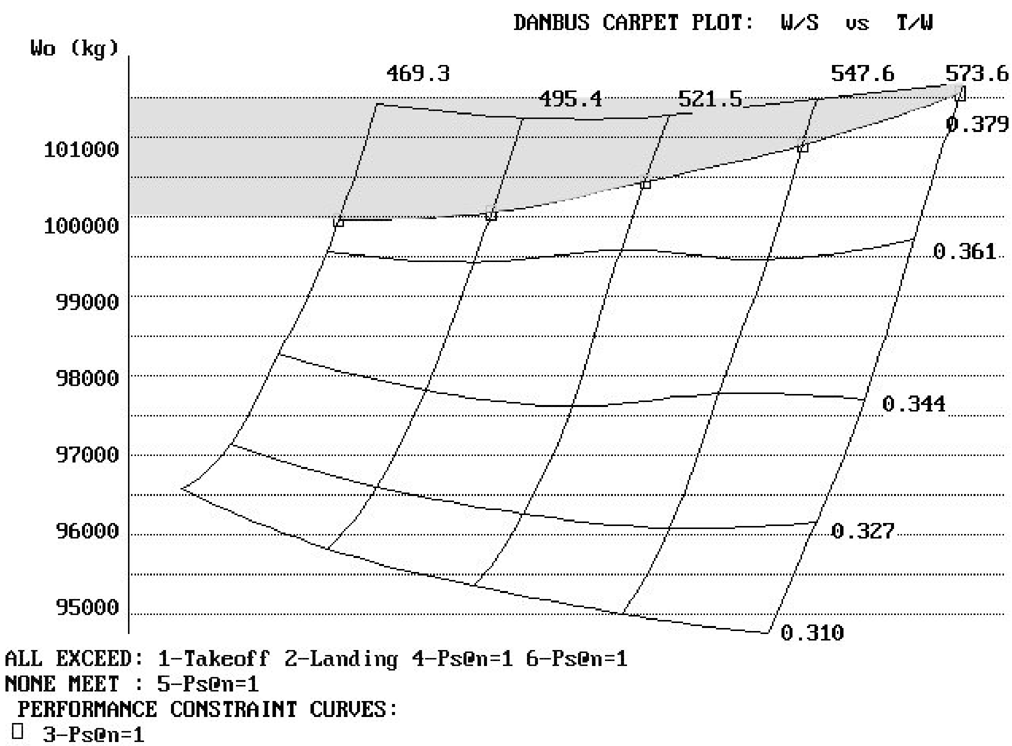


figure 36. "Bad" Transport - Carpet Plot Optimization

In this two-variable optimization, only the wing loading of the "bad" design can be fixed. The aspect ratio and sweep remain unchanged here, so the optimized "bad" design shown in figure 36 is still worse than the other design (220,462 lbs, or 100,000 kg). This occurs with a wing loading of 96 psf {469 kgsm} and a T/W of 0.365. Notice that even with optimization it is still incapable of meeting one of the performance requirements, an engine-out rate of climb condition (labeled "None Meet").

6.1.3 Asymmetric Light Twin

This aircraft is based on a novel variant of the concept of asymmetric design, with the objective of providing twin-engine redundancy without the engine-out controllability problems common to traditional designs.

Most twins have the engines so far apart that the loss of an engine at low speeds will result in a loss of control - the running engine will drag the aircraft over on its back or into a spin. P-effect makes this worse[‡‡‡].

In the 1930's, the Blohm-Voss Bv-141 was designed with the main fuselage (including the engine) offset to the left so that the pilot and gunner could be placed in a pod to the right, giving excellent visibility for both. Since the propeller was offset to the left, the yawing moment during climb was cancelled. More recently E. Rutan developed the asymmetric, twin-engined Boomerang, with a single fuselage with an engine in front, plus a second engine added alongside in a smaller engine nacelle which extends rearward to the horizontal tail. This configuration reduces the P-effect because the engines are placed close to the aircraft centerline.

Several years ago this author conceived of a variation on this asymmetric design philosophy which is a bit more "normal-looking" and which directly addresses the P-effect. This starts with a single-engine aircraft concept much like a low-wing Piper, which is morphed to an asymmetric twin by the addition of a second engine on the right wing in a pusher nacelle arrangement. To provide lateral balance, the main fuselage is moved to the left of the wing centerline. This has several beneficial effects. The front engine's P-effect is cancelled by its displacement to the left of center – the downward-moving blade is nearer to the wing centerline. The pusher engine, if the engine has the usual rotation direction, also has its downward-moving blade near the wing centerline so its P-effect is also cancelled. The wing is left-right symmetric providing normal lateral handling qualities, as is the horizontal tail.

A design was developed for this research using this concept, as shown in figure 37. This is intended as a homebuilt carrying 2 people with a jumpseat for a third. It is designed around two Jabiru 3300cc, 6-cylinder aircraft engines of 100 hp {75 kW}, based on data from the engine company's web site. A 2-bladed prop was used for thrust analysis.

[‡‡‡] P-effect is the yawing moment experienced by propellers when operated at an angle of attack, as during a climb. It is caused by the increased forward velocity and angle of attack of the downward moving blade, which therefore generates more thrust.

This strange-looking aircraft (it grows on you) has an as-drawn takeoff gross weight of 2200 lbs {998 kg} and an empty weight of 1413 lbs {641 kg}. It is 20 ft long {7 m} and has a span of 29 ft {8.8 m}. Wing area is 85 sqft {7.9 sqm}, and aspect ratio is 10.

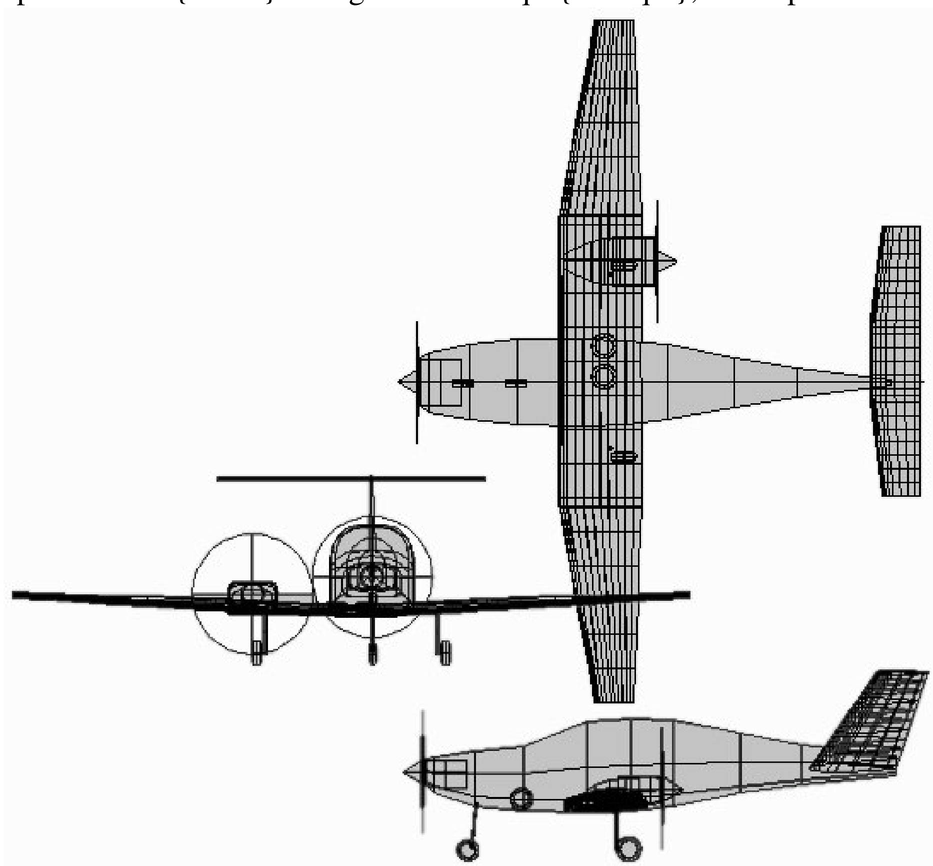


figure 37. Asymmetric Light Twin

Although the design is unusual looking, it can be modeled for analysis and optimization with the same parameters as a normal aircraft because the wing is symmetric about the true aircraft centerline. The fuselage, while offset, can be evaluated for drag as a normal fuselage. Weights were adjusted to include the extra engine and nacelle. Aerodynamic and weights results are presented below.

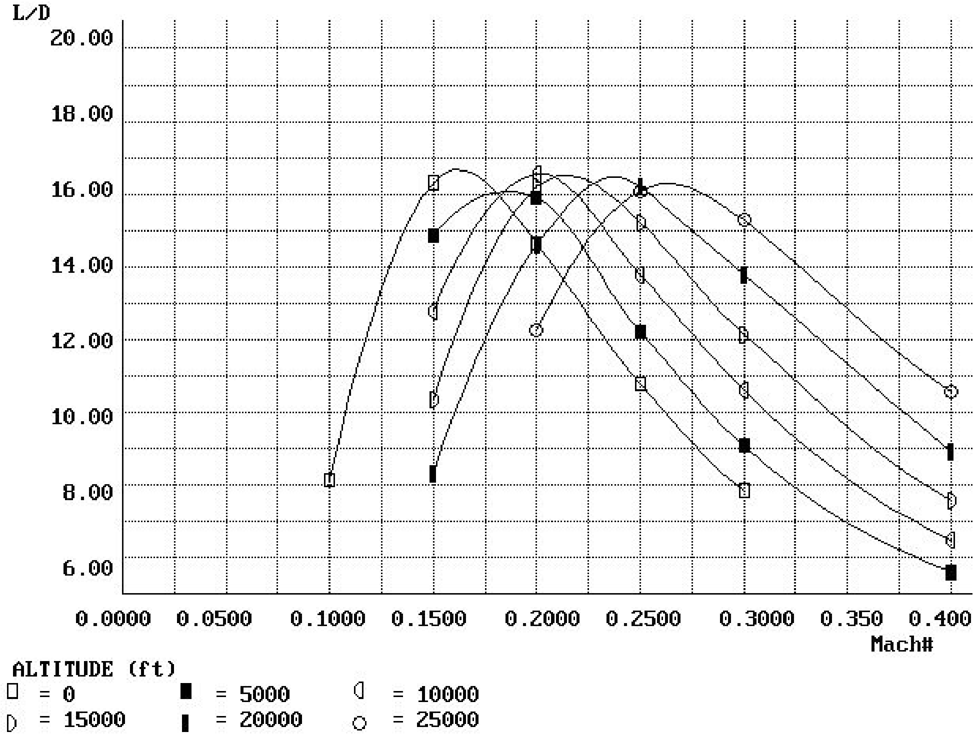


figure 38. Light Twin - Estimated L/D

STRUCTURES GROUP	550.9	249.9
Wing	186.5	84.6
Horiz. Tail	25.3	11.5
Vert. Tail	19.5	8.8
Fuselage	202.9	92.0
Main Lndg Gear	91.4	41.5
Nose Lndg Gear	25.4	11.5
PROPULSION GROUP	607.7	275.7
Engine(s)	322.0	146.1
Eng Installation	235.8	107.0
Fuel System	49.9	22.6
EQUIPMENT GROUP	179.0	81.2
Flight Controls	23.0	10.4
Electrical	63.3	28.7
Avionics	42.7	19.3
MISC EMPTY WEIGHT	75.0	34.0
TOTAL WEIGHT EMPTY	1412.6	640.8
USEFUL LOAD GROUP	787.4	357.1
Crew	400.0	181.4
Fuel	327.4	148.5
Oil	20.0	9.1
Payload	40.0	18.1
TAKEOFF GROSS WEIGHT	2200.0	997.9

figure 39. Light Twin - Weights Estimate

As with the last concept, the carpet plot optimization cannot be done with *T/W* and *W/S* so we use *W/S* and aspect ratio instead. In figure 40, the optimum aircraft is not found where two constraint lines intersect. Instead, one constraint line (rate of climb) forms a "bucket" and the optimum aircraft is found at its bottom. Its takeoff gross weight is 2180 lbs {989 kg}, found with wing loading of 25 and aspect ratio of 9.8.

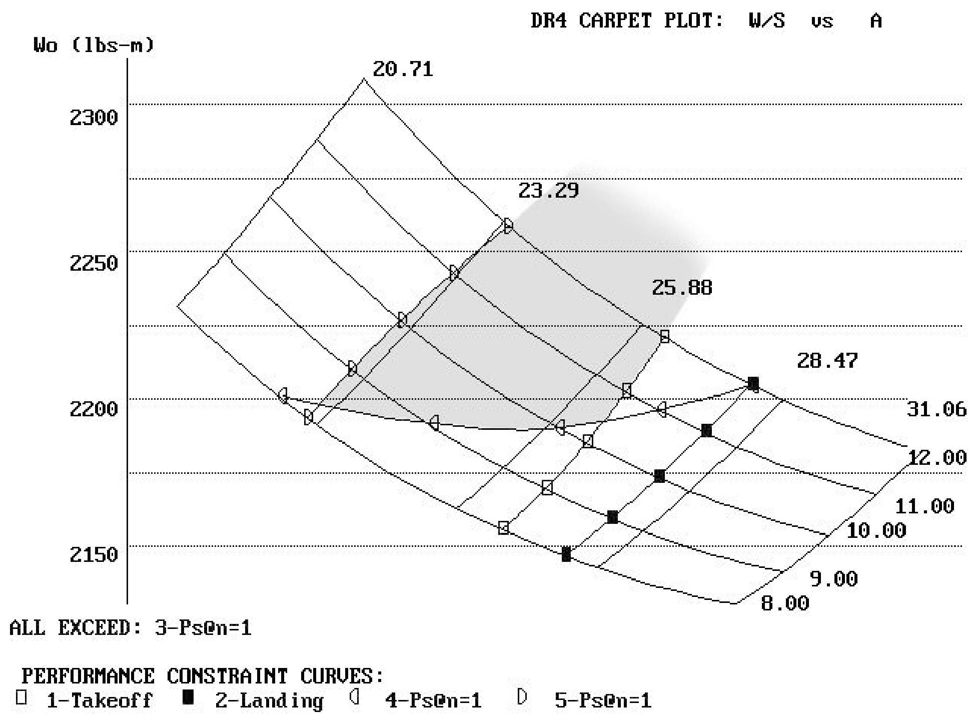


figure 40. Light Twin - Carpet Plot Optimization

6.1.4 Tactical UAV

Tactical Unmanned Air Vehicles (UAV's), those with bomb dropping or missile firing capabilities, are emerging as potent new weapon systems. The Predator, armed with small missiles, was recently used in combat for the first time. Several new tactical UAV's are now in development.

This author conducted design studies on Unmanned Air Vehicles in the mid-1990's (Raymer[86]), and later developed a novel concept for a stealthy, low-profile inlet duct for a UAV (illustrated in the third edition of Raymer[11]). This later work was used as the basis for development of the notional design concept illustrated in figure 41. A highly-swept flying wing, its intended mission is ground attack using a single 1,000 lb {454kg} "smart bomb".

As-drawn takeoff gross weight (W_o) of this notional UAV is 7,250 lbs {3,289 kg} with an empty weight of 4,157 lbs {1,886 kg}. Combat radius is 750 nmi {2778 km}. Length is 30 ft {9 m} with span of 36 ft {11 m}. Wing aspect ratio is 2.2 and the wing is highly swept – 60 degrees. The engine is an approximated version of the widely used JT-15D, of 2900 lbs thrust {12.9 kN}. L/D ratios are shown in figure 42.

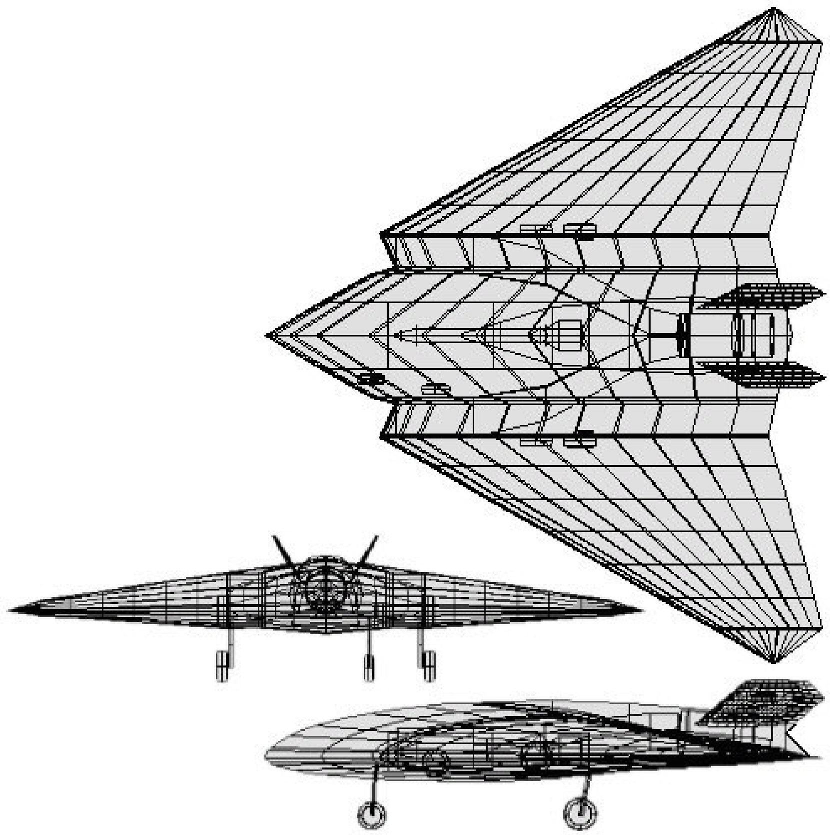


figure 41. Tactical UAV

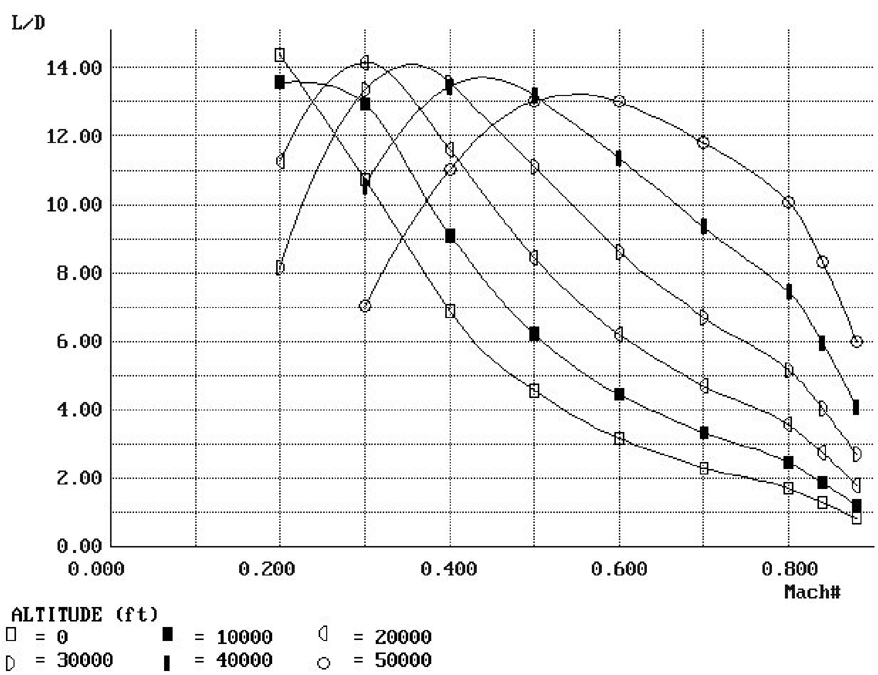


figure 42. Tactical UAV - Estimated L/D

UAV weights estimates are provided below, based on attack fighter weights adjusted to reflect the unmanned aspects of the design. Suitable miscellaneous allocations were also made for low-observables treatments and an internal weapons bay.

STRUCTURES GROUP	1235.9	560.6
Wing	713.4	323.6
Vert. Tail	65.2	29.6
Fuselage	110.6	50.2
Main Lndg Gear	248.1	112.6
Nose Lndg Gear	73.2	33.2
Engine Mounts	7.9	3.6
Firewall	11.3	5.1
Engine Section	6.1	2.8
PROPULSION GROUP	736.4	334.0
Engine(s)	632.0	286.7
Oil Cooling	37.8	17.2
Engine Controls	18.6	8.4
Starter	10.7	4.9
Fuel System	37.3	16.9
EQUIPMENT GROUP	1286.9	583.7
Flight Controls	180.2	81.8
Instruments	100.9	45.8
Hydraulics	108.4	49.2
Electrical	208.3	94.5
Avionics	567.9	257.6
Air Conditioning	121.1	54.9
MISC EMPTY WEIGHT	898.0	407.3
TOTAL WEIGHT EMPTY	4157.2	1885.7
USEFUL LOAD GROUP	3092.8	1402.9
Fuel	2072.8	940.2
Oil	20.0	9.1
Payload	1000.0	453.6
TAKEOFF GROSS WEIGHT	7250.0	3288.5

figure 43. Tactical UAV - Weights Estimates

Classical carpet plot optimization of this UAV is shown in figure 44. The carpet plot optimization cannot be done with the usual two variables, *T/W* and *W/S*, because the aircraft is designed and optimized assuming a fixed-size engine, the JT-15D. For the designs presented above, it was assumed that the engines could be scaled to provide whatever thrust level was required, and if that meant a new engine had to be built, we built it. For this and the next design, it is assumed that we must pick and use an existing engine. Thus, *T/W* cannot be defined parametrically to a specified value because the engine thrust is fixed. As aircraft weight changes, *T/W* changes as well. Since *T/W* is unavailable as a trade variable, the aspect ratio is typically used instead.

The optimum aircraft weighs about 7100 lbs {3220 kg} when sized to the design mission while meeting all performance constraints. These include a 28 degree per second instantaneous turn and a 3-g sustained turn, plus takeoff and landing distances of 1600 ft {488 m}. The optimum wing loading is about 11.5 psf {56 kgsm} and the optimum aspect ratio is about 1.8.

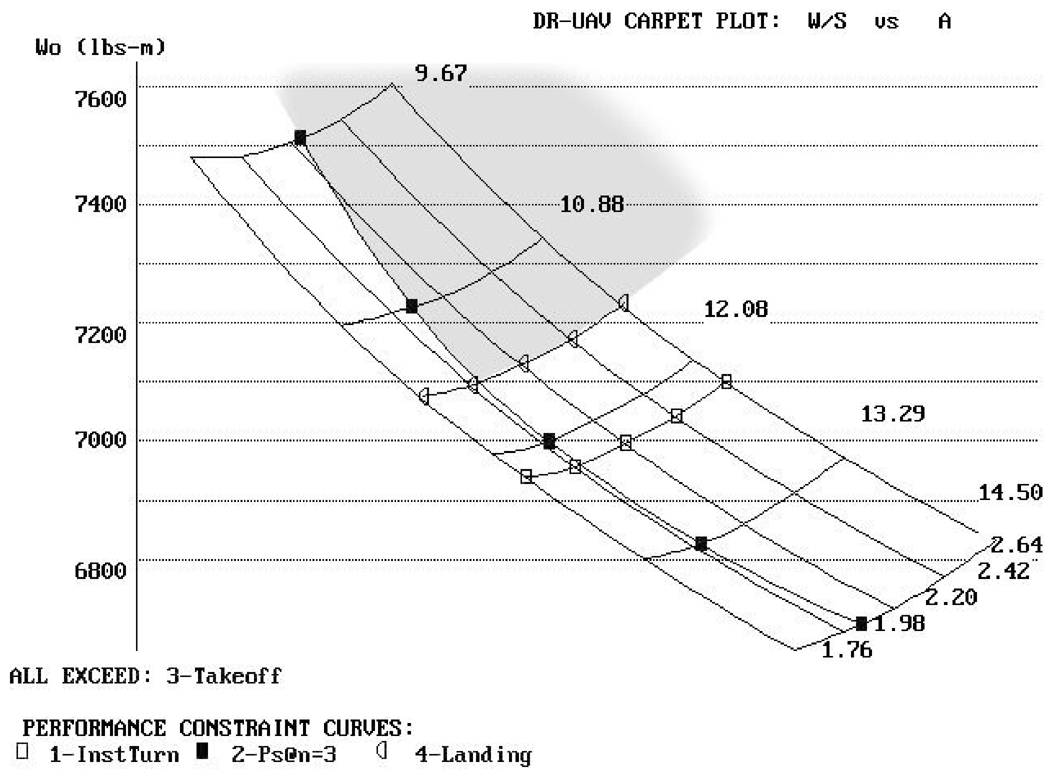


figure 44. Tactical UAV - Carpet Plot Optimization

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 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 (see 4.3.7) and the percent of the final population meeting all performance requirements.

Sample results for certain cases are also provided in the Appendices, but since each run produces several thousand pages of output detailing about 10,000 parametric design variations, complete data are not included herein. Results data are graphically summarized in the sections below.

7.1 Calibration Results: Orthogonal Steepest Descent Search

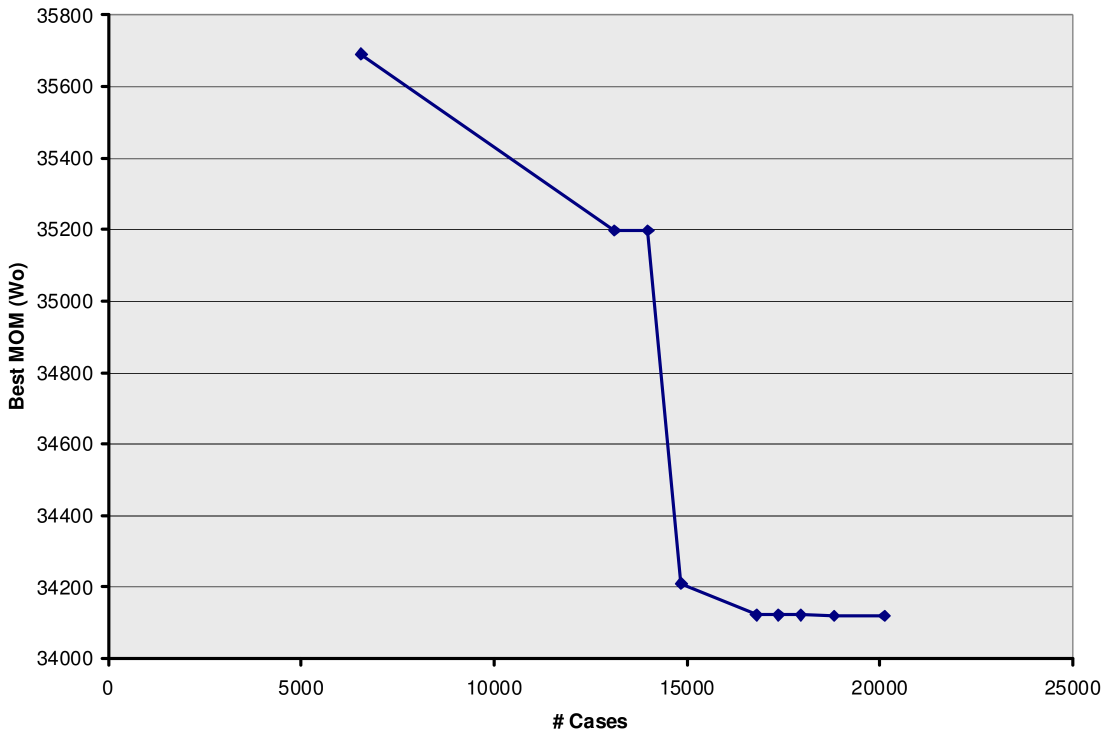
As detailed in Section 4.2, the Orthogonal Steepest Descent *(OSD)* method is both deterministic and determined, and will ultimately find the best-possible optimum given the selected ranges of parametric variables. While *OSD* is theoretically capable of converging on a local optimum, thus missing a better global solution, this was never actually observed during this research. This author believes that such problems are unlikely in aircraft design optimizations for two reasons. First, the design space is unlikely to be reflexed for any of the parametric variables in use. It is for this reason that second-degree polynomial response surfaces are widely accepted as representative of the design space. Second, and perhaps more important, in aircraft design problems we are optimizing a design layout prepared by a human designer, and can rely on the designer's experience to develop a good-enough concept that the optimal solution is found within fairly small excursions of the parametric variables.

Orthogonal Steepest Descent optimization was conducted for all four aircraft concepts, and in each case it found a better design than was found by the carpet plot method. These are summarized below, showing the solution convergence in graphical form followed by the optimum design variables and Measure of Merit.

These results can be considered a calibration baseline for the other MDO methods described later. It should be expected that these stochastic methods would never find a better result, but may find almost as good of a result in a fewer number of parametric case evaluations. Run numbers refer to the test-case run matrix in the Appendices.

7.1.1 Advanced Multirole Export Fighter

Results for the Advanced Multirole Export Fighter are shown in figure 47 (run 1). Recall that the carpet plot optimization of this design yielded a *T/W* of 0.72 and *W/S* of 70. This gives a purchase price (MOM) of $38.5 million, at a takeoff gross weight of 44,500 lbs {20,185 kg}. *OSD*, changing eight rather than two design variables, was able to find a revised design with a price of $34.1 million and weight of 36,600 lbs {16,600 kg}. It took about 17,000 parametric case evaluations to get this final number, but the method required a few thousand more cases to be sure that the best answer was actually found.



	Carpet Plot	OSD	
T/W	0.720	0.617	
W/S	70.000	68.65	psf
ASPECT	2.500	2.97	
SWEEP	48.000	38.4	
TAPER	0.120	0.096	
t/c	0.045	0.054	
Fus l/d	13.846	16.62	
CL-dsgn	0.200	0.235	
MOM	$38.5m	$34.1m	
Wo	44.5k	36.6k	lbs
Wo	20.2k	16.6k	kg

figure 45. OSD Optimization Results – Advanced Multirole Export Fighter

The best aircraft from the *OSD* optimization is shown in figure 46 as a two-variable carpet plot around the optimum. This is a useful way to visualize the impact of the performance requirements on the optimal solution. One can see that the *OSD* method worked because the design constraints cross exactly at the baseline aircraft (middle) of the graph.

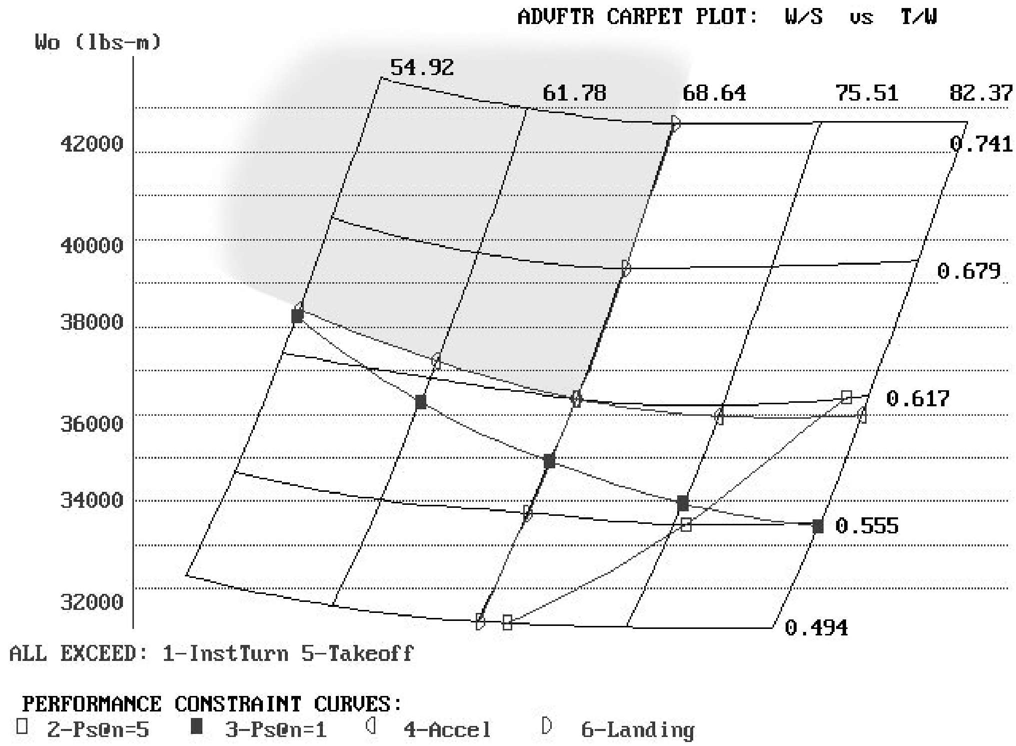
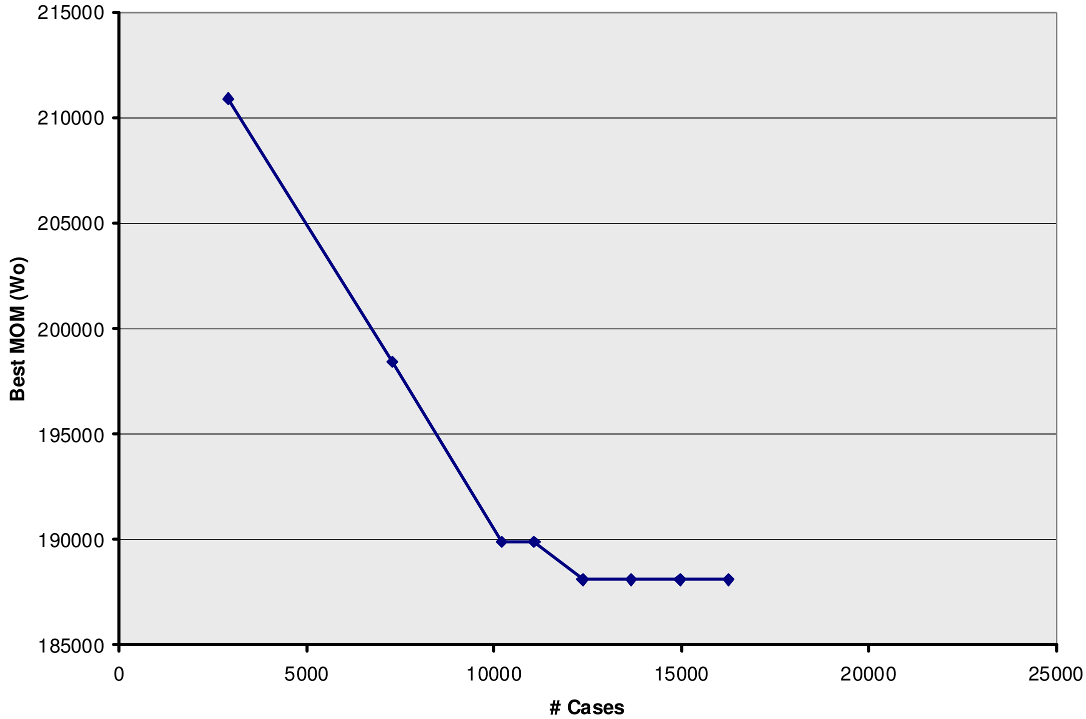


figure 46. Carpet Plot About Optimal Design: Fighter

7.1.2 Civil Transport

Optimization of the Civil Transport is shown in figure 47 (run 2). In this case, the starting design was the "bad" transport shown in figure 31, deliberately designed poorly to see if the optimizers could "fix" it. The carpet plot optimization of this "bad" design yielded a design takeoff gross weight of 220,462 lbs {100,000 kg}, and was unable to meet all performance requirements (figure 36). Starting with the original, "good" design, the carpet plot found a minimum takeoff gross weight of 193,000 lbs {87,643 kg}. This design did meet all performance requirements.

To test the *OSD* method, it was given the "bad" design as a starting point and had no trouble converting it back into a "good" design with a takeoff gross weight of 188,100 lbs (85,321 kg}. All performance requirements were met. 16,254 parametric case evaluations were required.



	Carpet Plot*	OSD*	OSD *(good)*	
T/W	0.365	0.30	0.31	
W/S	96.000	113.06	116.69	psf
ASPECT	5.000	12.00	11.52	
SWEEP	40.000	25.00	23.42	
TAPER	0.250	0.20	0.20	
t/c	0.120	0.14	0.14	
Fus l/d	11.266	12.96	12.26	
CL-dsgn	0.550	0.66	0.45	
Wo	220.5k	188.1k	187.9k	lbs
Wo	100.0k	85.3k	85.2k	kg

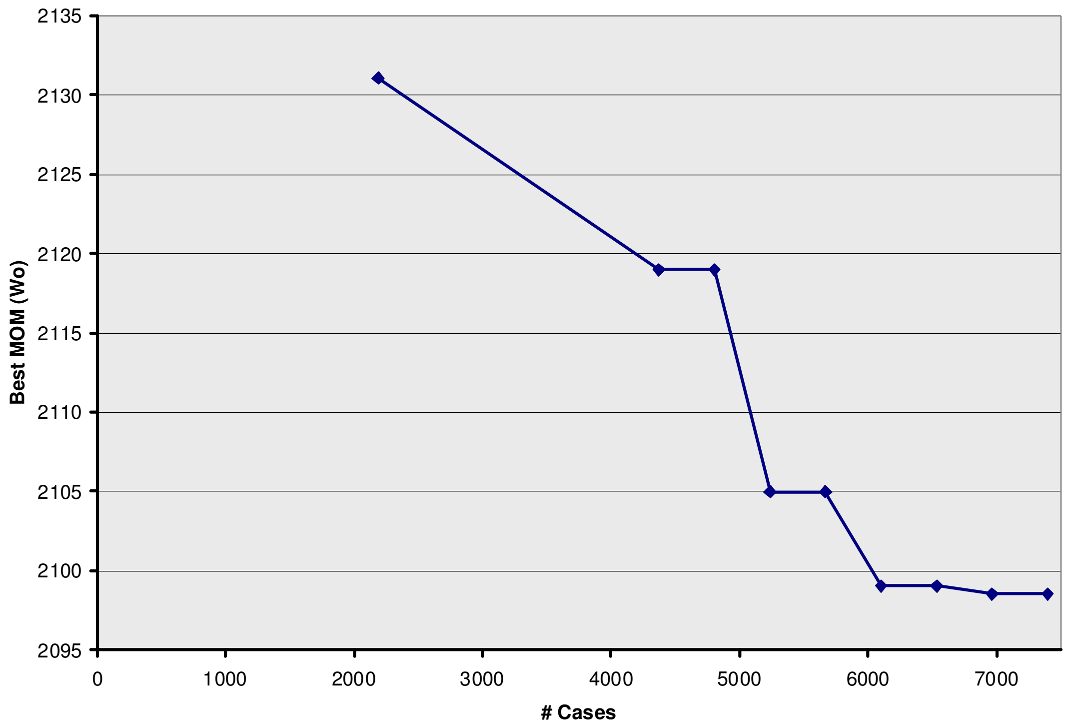
(**bad*)

figure 47. OSD Optimization Results – Civil Transport (starting with "bad" one)

A second *OSD* run was made (run 9), starting with the original "good" civil transport design to see how that would compare with the "bad" starting point. This found a minimum takeoff gross weight of 187,900 lbs {85,230 kg}, with very similar values of the design parameters. The slight weight difference is due to the fact that the "good" design was made "bad" simply by changing the wing, without adjusting the tail areas. The optimizer did adjust tail sizes per the automatic redesign procedures described in Section 5.1. In other words, the computer did a better job than the human designer – this author!

7.1.3 Asymmetric Light Twin

The carpet plot optimization of the Asymmetric Light Twin found a minimum takeoff gross weight of 2180 lbs {989 kg}. Since the design is based around existing engines, the two-variable carpet plot was done using wing loading and aspect ratio as parameters. Results using *OSD* are shown in figure 48 (run 3). Seven variables are used (*T/W* is still unavailable as a parametric parameter since an existing engine is used). The optimized design weights 2098 lbs {951 kg} and meets all requirements. 7398 parametric case evaluations were run to obtain this result.



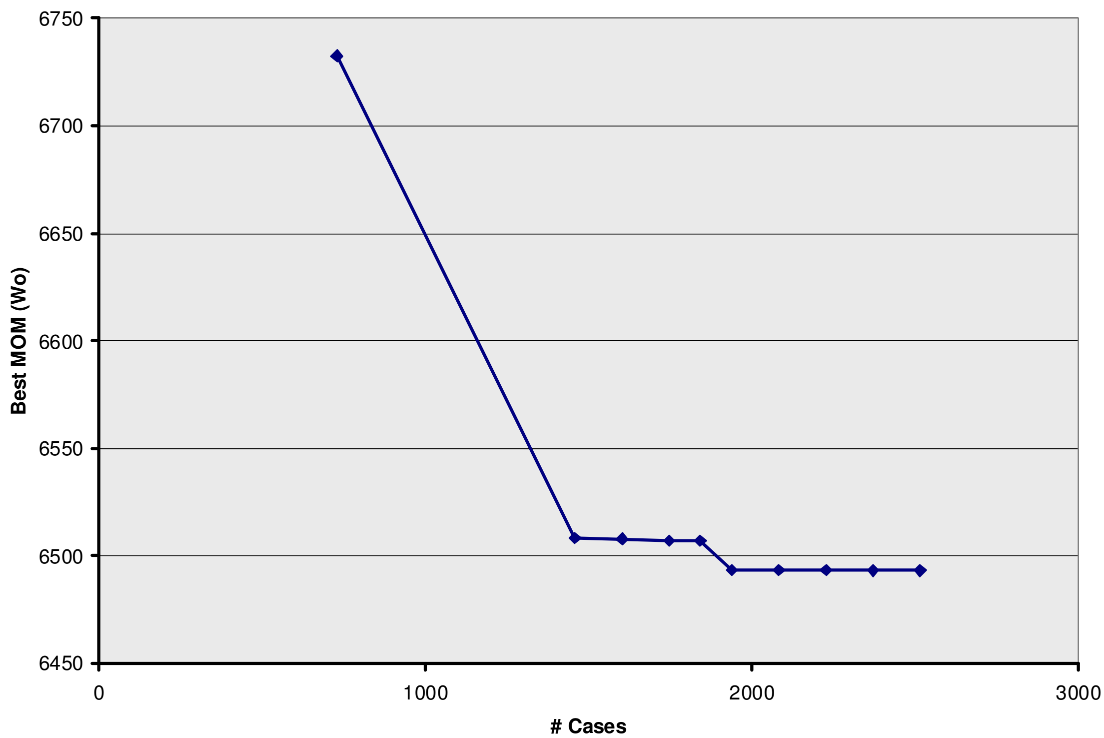
	Carpet Plot	OSD	
W/S	25.000	26.60	psf
ASPECT	9.880	8.75	
SWEEP	8.030	6.42	
TAPER	0.400	0.32	
t/c	0.150	0.18	
Fus l/d	5.000	4.81	
CL-dsgn	0.500	0.60	
Wo	2180	2099	lbs
Wo	989	952	kg

figure 48. OSD Optimization Results – Asymmetric Light Twin

7.1.4 Tactical UAV

The final calibration run for the Orthogonal Steepest Descent *(OSD)* method for MDO is for the tactical UAV (run 4). Again, the engine is fixed in size so the *T/W* cannot be used as a parametric variable. Also, there is no fuselage so its fineness ratio cannot be optimized.

Carpet plot optimization of this design yielded a *W/S* of about 11.5 psf {56 kgsm}, and an aspect ratio of 1.8, with a takeoff gross weight of 7100 lbs {3220 kg}. *OSD*, changing six rather than two design variables, was able to find a design with a weight of 6493 lbs {2945 kg}, requiring a total of 2514 design cases.



	Carpet Plot	OSD	
W/S	11.50	14.50	psf
ASPECT	1.80	1.98	
SWEEP	60.000	48.0	
TAPER	0.110	0.088	
t/c	0.150	0.129	
CL-dsgn	0.500	0.600	
Wo	7100	6493	lbs
Wo	3220	2945	kg

figure 49. OSD Optimization Results – Tactical UAV

A two-variable carpet plot of the best result from the *OSD* optimization is shown as figure 50. This shows an interesting result. The performance constraints of the carpet plot allow moving downwards from the *OSD*-optimized design (seen at the center of the carpet lines), into the crosshatched region. If allowable, this would reduce the weight from about 6500 lbs to about 6380 lbs {2948 to 2894 kg}, or about two percent.

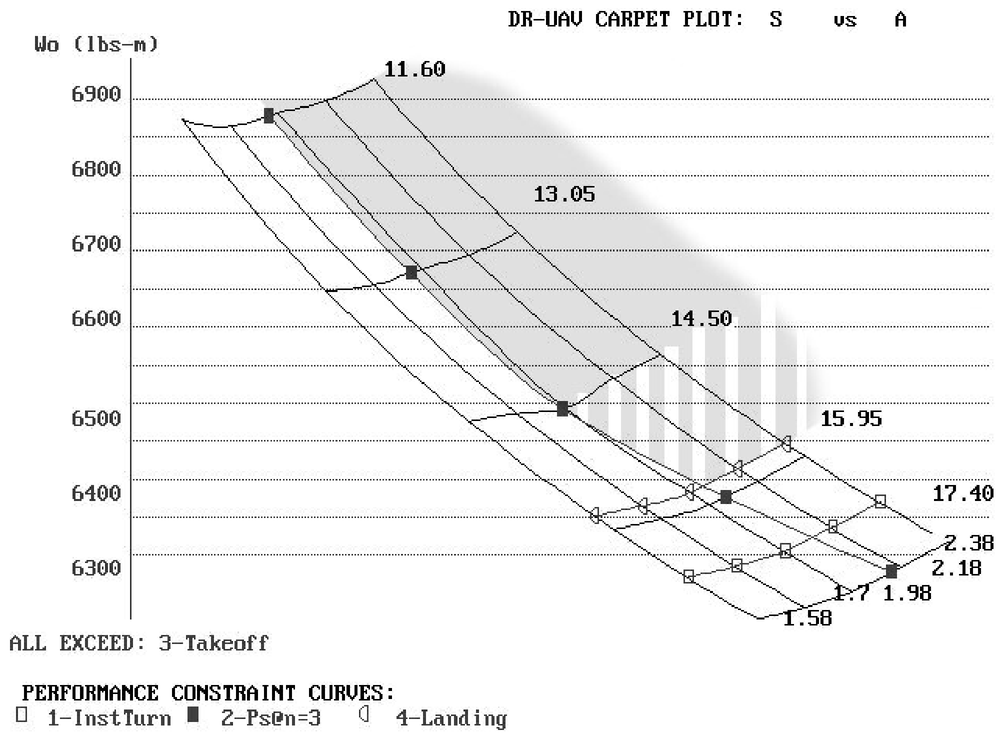


figure 50. Carpet Plot About Optimal Design: UAV

The reason that the *OSD* optimization didn't "discover" this possibility is that it wasn't permitted to. In the setup for this case, the highest permitted wing loading was restricted to 14.5 psf {70.8 kgsm} to ensure that the wing would remain large enough to contain the weapons bay. The carpet plot knows nothing of such constraints, and offers the crosshatched region in its ignorance. This does indicate to the designer that a price is being paid for attempting to keep the weapons bay entirely within the contours of the wing, and suggests that for this design a "stub" fuselage may be a more optimal design.

7.1.5 Comparison: 2-Variable *OSD* vs. Carpet Plot

The Orthogonal Steepest Descent optimization routine in RDS-Professional was used to cross check its results versus those obtained via classical carpet plot optimization, as described above. Results are reassuring. Restricting *OSD* to the same two variables used for the carpet plots of the four designs above gave the same results as found on the carpet plots (runs 5-8).

The only exception was the "bad" civil transport, where the *OSD* optimization could not find an answer. As shown on the carpet plot, no change in *T/W* and *W/S* could fix the engine-out climb requirement. The *OSD* was then rerun with that performance requirement removed (run 47), and was able to find the same result as the carpet plot.

These 2-variable OSD runs took about 60 parametric evaluations, versus the 25 (5^2) used for a standard carpet plot. In either case, total time is a few seconds.

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 shown in section 7.1 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.

The Monte Carlo method is not included in these graphs. It doesn't improve or evolve the design, it simply calculates the requested number of design cases and reports on the best one found. As can be seen in the Appendices, the Monte Carlo result was usually worse than that of the other approaches, but not by a significant amount. More than any other approach, its success depends upon pure luck and so general conclusions relative to other methods are difficult. However, it is simple to program and at least for these cases, gave credible results.

Optimization results for the Advanced Multirole Export Fighter are shown in figure 51. 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.

Others such as Wang and Damodaran[87] have reported the opposite – that deterministic solutions are faster than stochastic. However, this research finds that it depends on the number of design variables and that for a large number of design variables, the stochastic methods can be faster (see below).

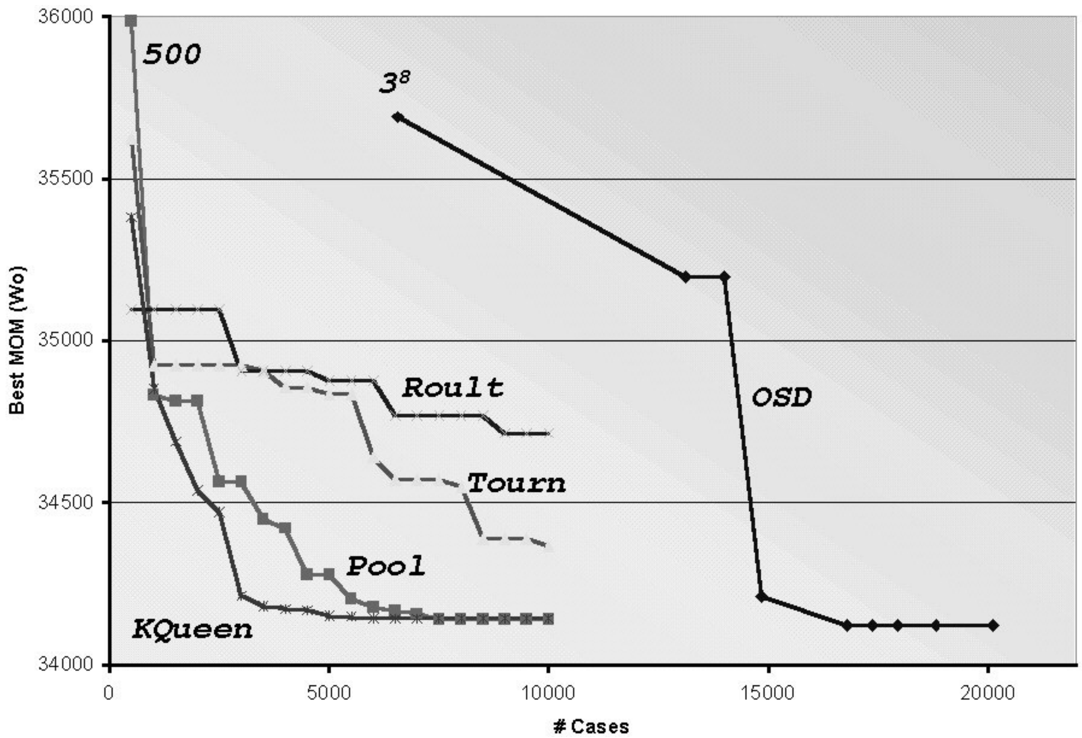


figure 51. MDO Solution Convergence: Fighter

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 in Section 4.3.7 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 listed in the Appendices, 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.

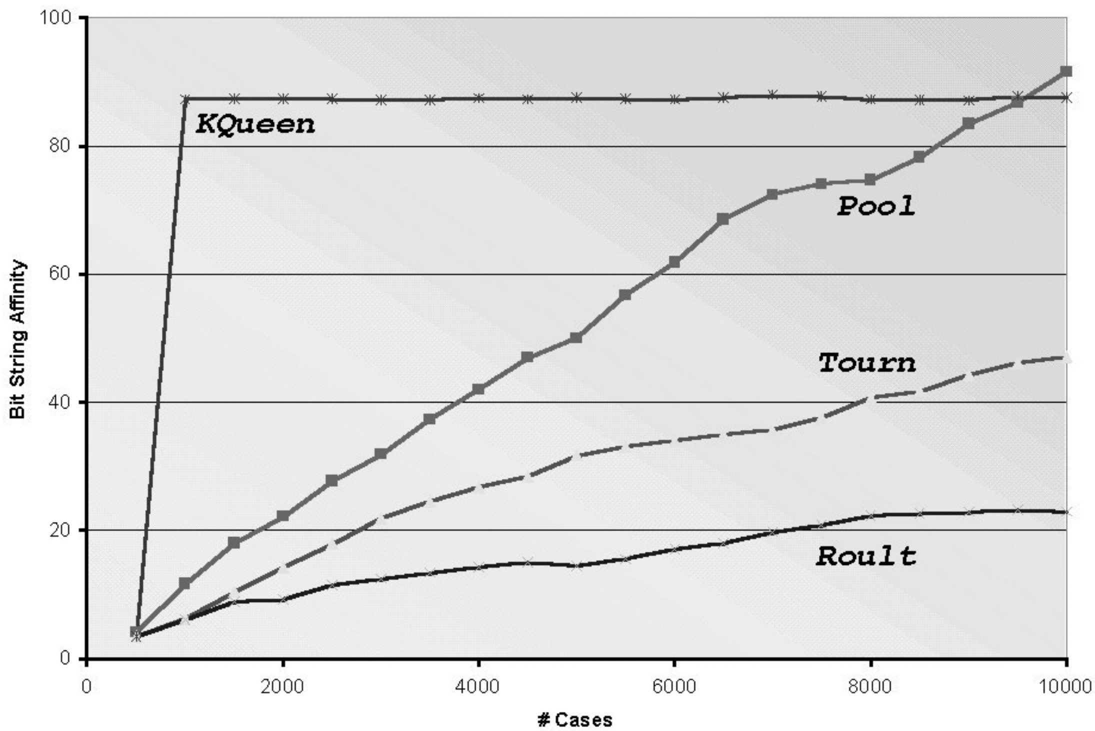


figure 52. Bit-String Affinity: Fighter

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 stochastic methods can initially examine design concepts throughout the design space. With a bit of luck, a fairly good design can be found even in the first population. No time is wasted stepping away from the bad region of the design space.

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 54.

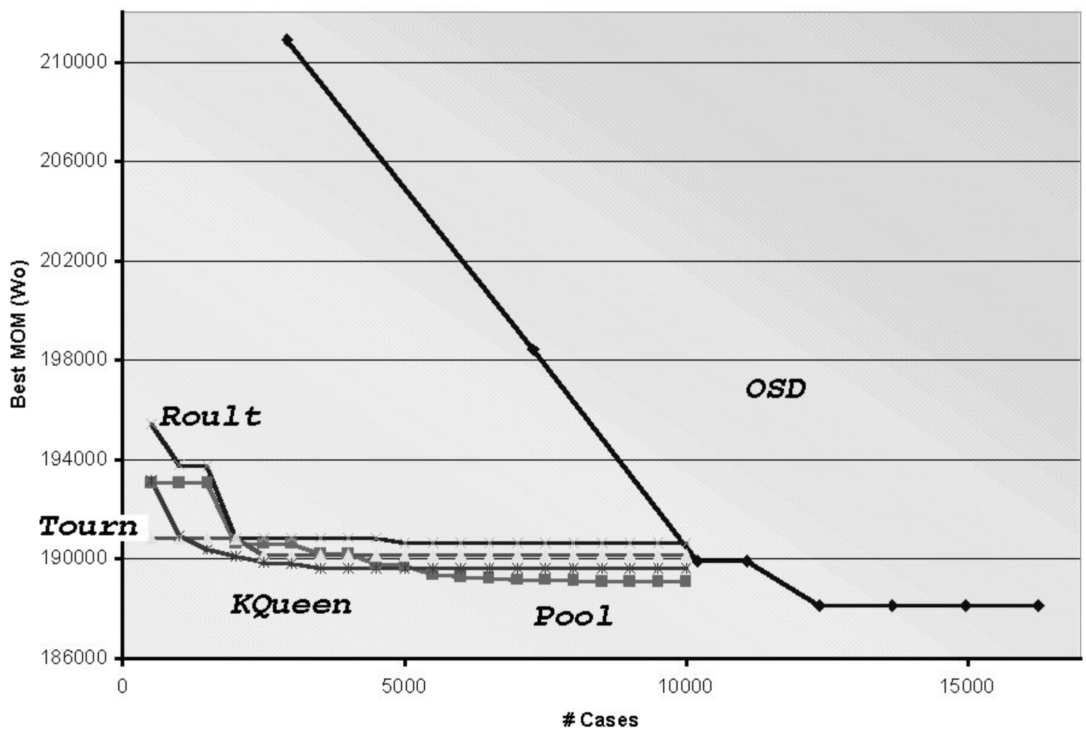


figure 53. MDO Solution Convergence: Transport

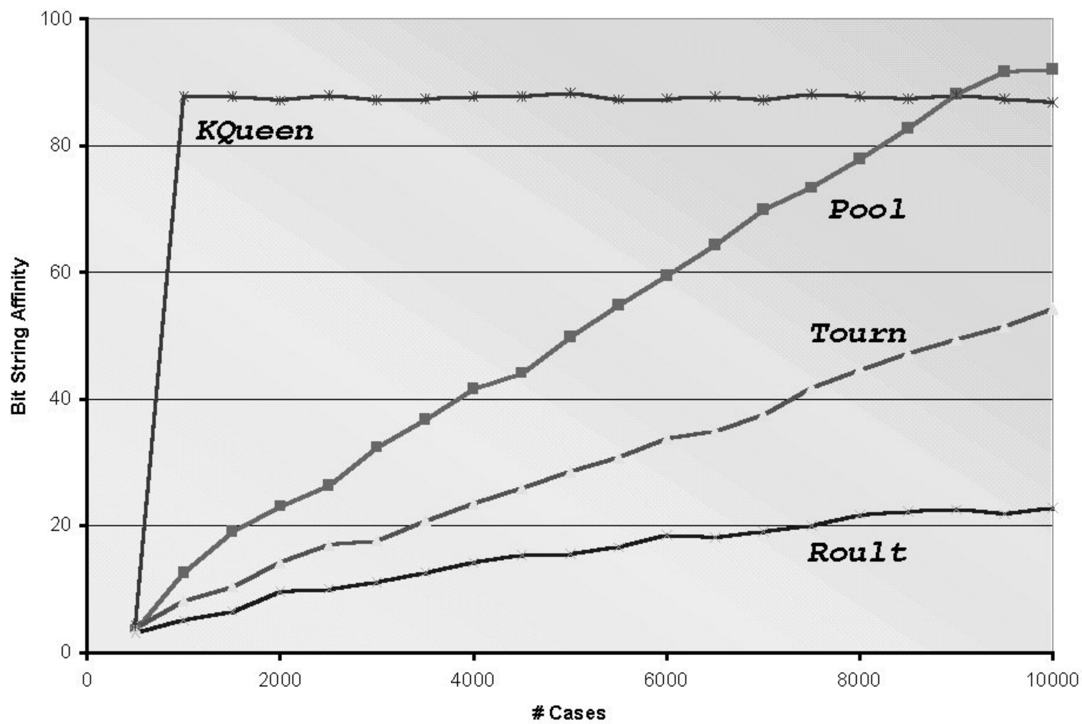


figure 54. Bit-String Affinity: Transport

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 55).

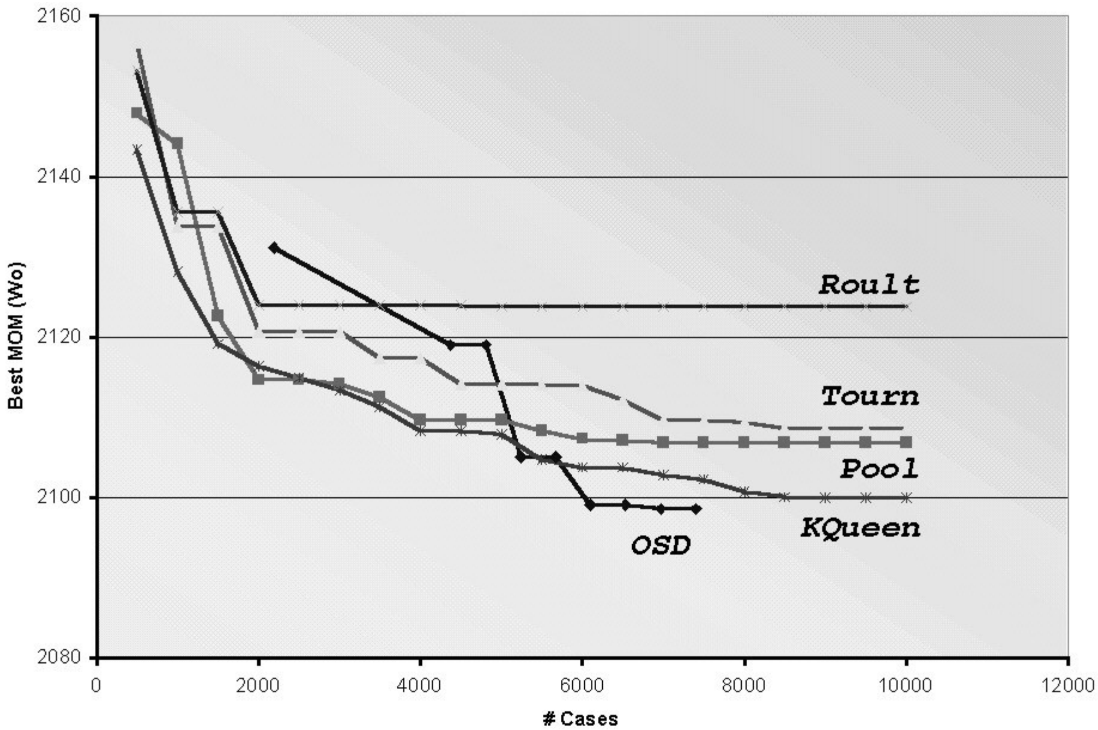


figure 55. MDO Solution Convergence: Light Twin

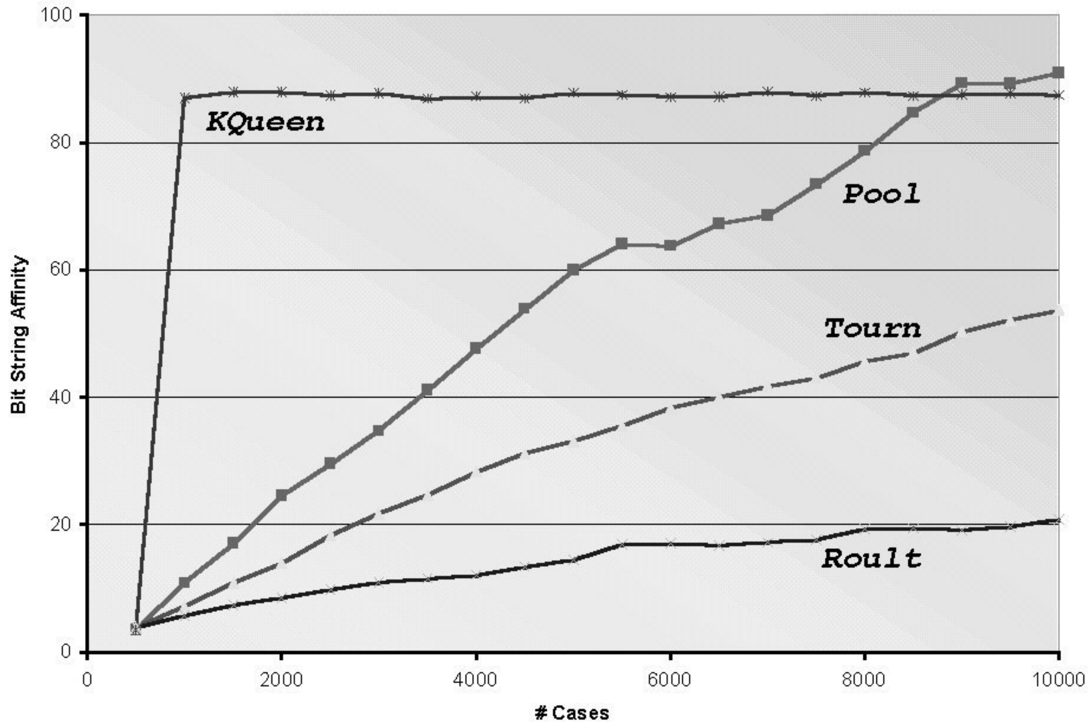


figure 56. Bit-String Affinity: Light Twin

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 57.

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.

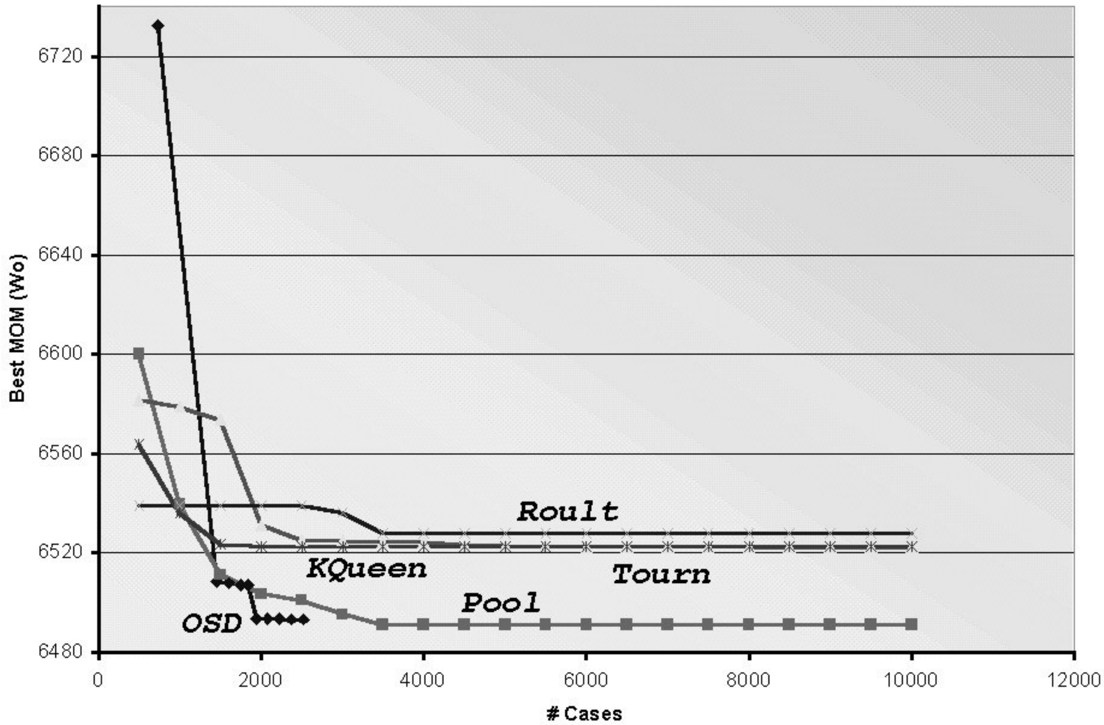


figure 57. MDO Solution Convergence: UAV

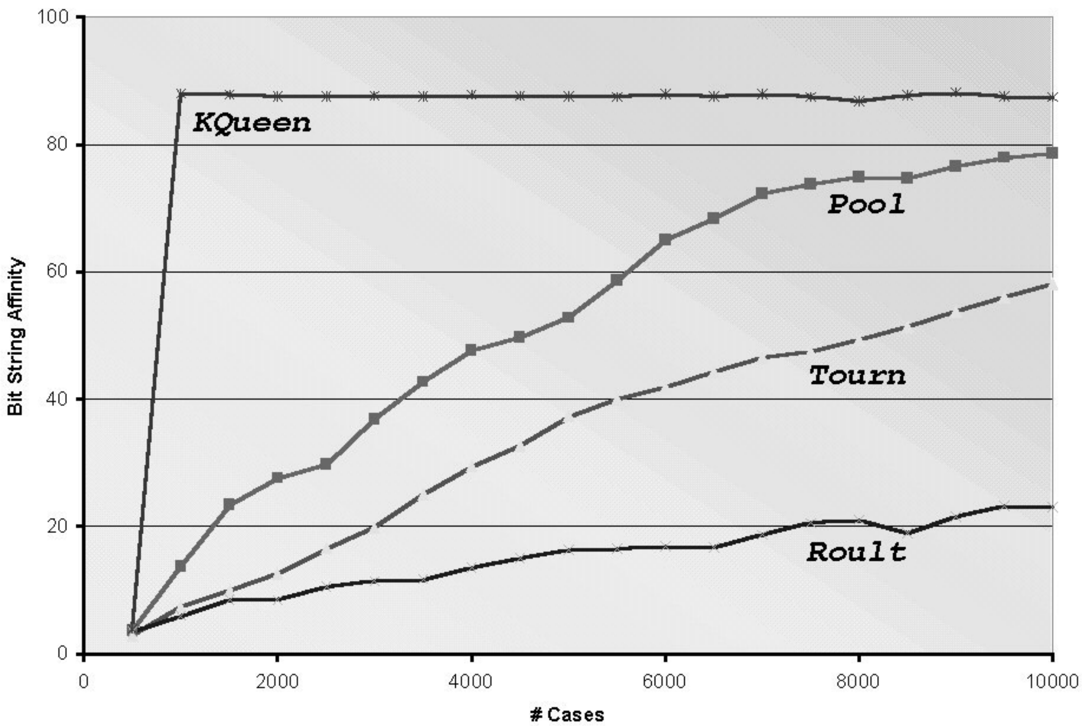


figure 58. Bit-String Affinity: UAV

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 59, the number of parametric evaluations required to find a result only 1% higher than the OSD solution is graphed. This is shown in figure 60 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*.

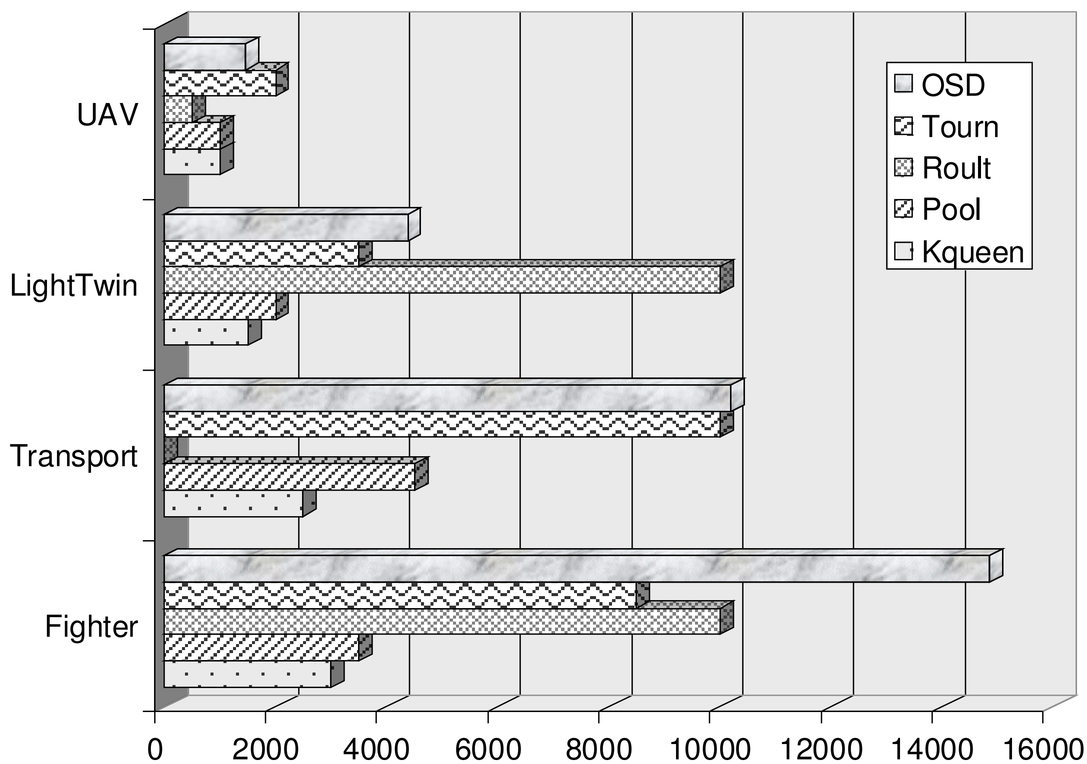


figure 59. Number of Runs Required to Come Within 1% of Best

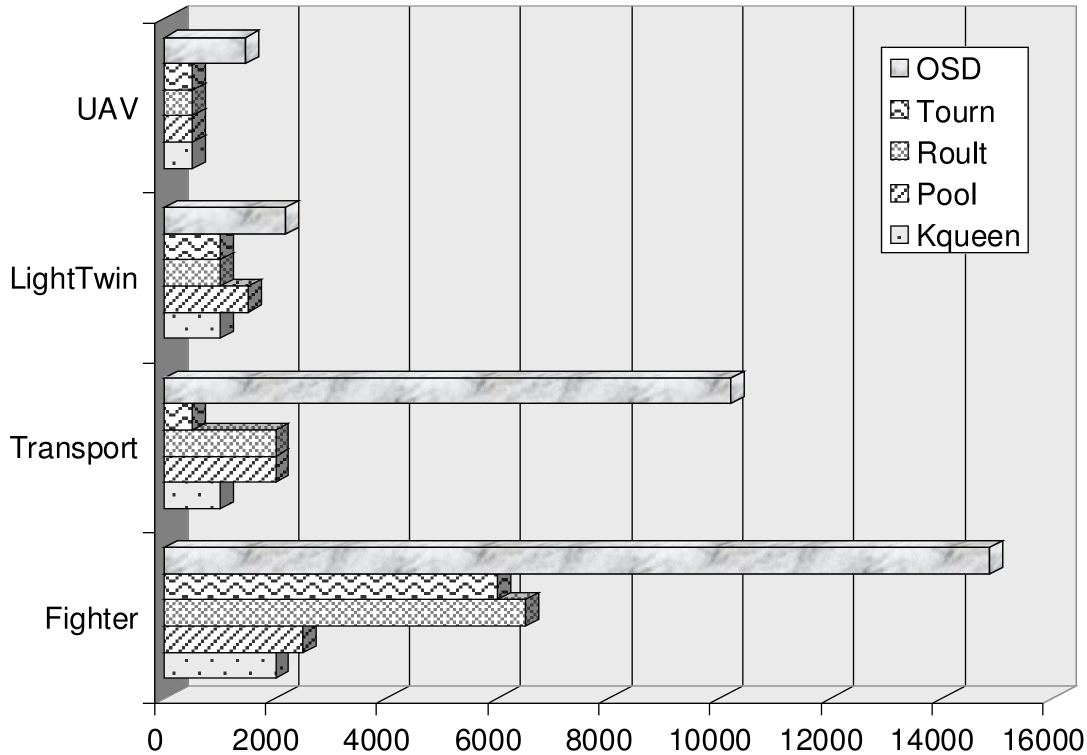


figure 60. Number of Runs Required to Come Within 2% of Best

7.3 Hybrid Methods

From these results it would seem that the stochastic methods are superior at quickly exploring the total design space, but that when you want to get the final "right" answer the Orthogonal Steepest Descent method is preferable. This led to the idea of a hybrid method. Code was developed to allow starting with any of the stochastic methods for a specified number of generations, then finishing with an Orthogonal Steepest Descent search beginning from the best found by the stochastic method.

In the test cases conducted for this research, the hybrid methods didn't function very well. The problem seems to be an inability to predict in advance how good the stochastic result would be and thereby determine the desired starting resolution for the deterministic search. If the starting resolution is set too coarse, the deterministic routine repeats thousands of full-factorial case evaluations around the stochastic method's solution while slowly tightening the resolution. If the resolution is set too fine, and the stochastic method's result wasn't very close to the final optimum, the deterministic routine requires a huge number of small steps to eventually find the correct optimum.

Runs 72-76 applied the hybrid method. In each case, the optimization took as long or longer than an *OSD* search starting from the as-drawn baseline design. In one case, the

Tournament-OSD hybrid, it took over 43,000 parametric case evaluations to find a solution.

7.4 Trade Studies of MDO Methods and Options

These MDO routines have many options, as described in Section 4.1. In this section, several of these are exercised to determine their impact on the solution convergence and final result[§§§]. Interestingly enough, the main lesson of these studies is that the details of these OSD methods do not change the results by much – just about any combination of such options would probably yield a workable method. Perhaps this explains why some people say that there are as many MDO methodologies as there are MDO researchers.

7.4.1 Measure of Merit Weighting

In Section 4.3.2, a scheme was described to allow increasing the importance of the calculated value of the measure of merit versus the randomness of the selection process. A linear weight implies that being "good" is equally weighted with being "lucky". Squared, fourth-power, and cosine wave weighting functions were applied to the fighter and transport designs. These were tested using the Roulette Genetic Algorithm which, based on results above, seems that it would benefit the most from an additional weighting applied to Measure of Merit.

Results were inconclusive. None of the weighting schemes had a substantial effect on the final calculated value of the measure of merit (cost for the fighter, weight for the transport). Convergence graphs were prepared to see if an improvement could be detected, but little change was observed other than a random-appearing scatter in the convergence lines.

In any event, these MDO methods all work reasonably well without such weighting. Perhaps that is why little effect was seen – they already work too well for an improvement to be noted.

7.4.2 Alternative Crossover Schemes

In the cases run above, uniform crossover was used exclusively (see 4.3.5) based on what seems to be a consensus in the recent literature. Alternative crossover schemes were tested, namely the bisection scheme and a parameter-wise scheme wherein genes are passed whole to the next generation, as previously described. For the fighter aircraft results are shown below which indicate that bisection crossover is slightly poorer and uniform crossover is best, with parameter-wise crossover very similar to uniform crossover. A similar evaluation for the transport aircraft revealed virtually no difference in outcome for all three schemes.

[§§§] The MDO methods and options implemented in this research, when applied to each other in a full factorial sense, define a huge number of alternative MDO approaches. To truly find the most-optimal combination of methods, one should apply MDO to the MDO! This is beyond the scope of this study, and as indicated in these results, would probably not offer any real improvement to the optimization.

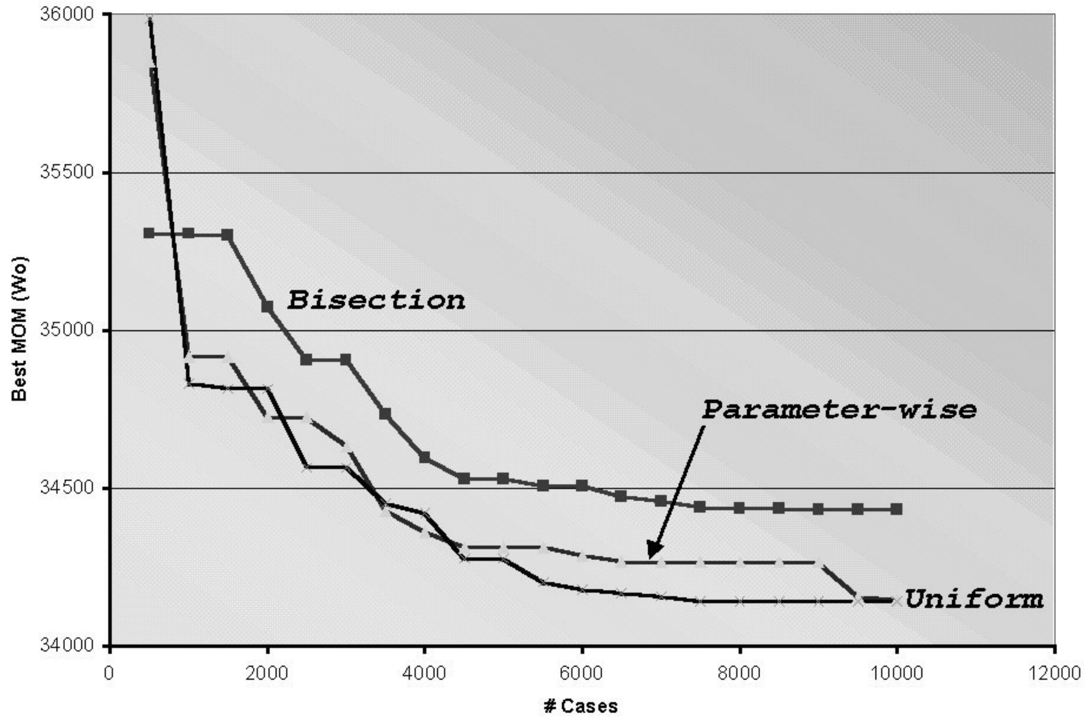


figure 61. Alternative Crossover Schemes: Fighter

7.4.3 Simulated Annealing

A form of simulated annealing was tested on the fighter and transport designs. As implemented here, this applies a linear scaling to the performance penalty factor from a low initial value to a higher value at the end of the optimization. Here the measure of merit is not being "annealed", only the penalties applied when performance constraints are missed. Results for the transport are shown below (run 49), and indicate some benefit to this approach. A better value of the measure of merit is found, despite starting from a worse value in the initial random population. However, a similar study for the fighter aircraft did not show a benefit (run 48).

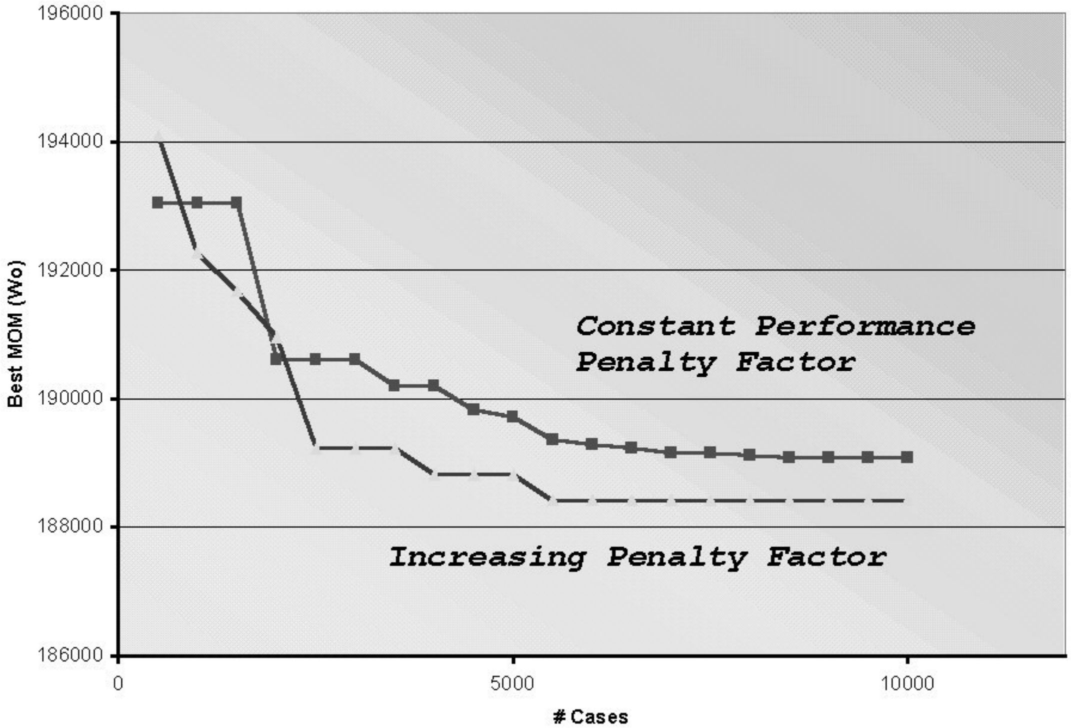


figure 62. Simulated Annealing: Transport

7.4.4 Elitism

Elitism was defined above as the preservation of the best individual or individuals of each generation. This upper elite is preserved without crossover or mutation and inserted into the next generation. By using Elitism, one is assured that the next generation will be at least as good as the current generation – backsliding is prevented. However, the elite individuals do not contribute to the evolutionary process because they are neither mating nor mutating, so convergence may be slowed.

Elitism of one individual was employed as the default in all these MDO runs. This was removed for three test runs. In none of these was the final result materially affected. In figure 63 one can see that elitism does serve to prevent the upward spike seen at generation 5 (after 2500 evaluations). But, the no-elitism solution managed to converge to essentially the same result even with this occurrence of backsliding.

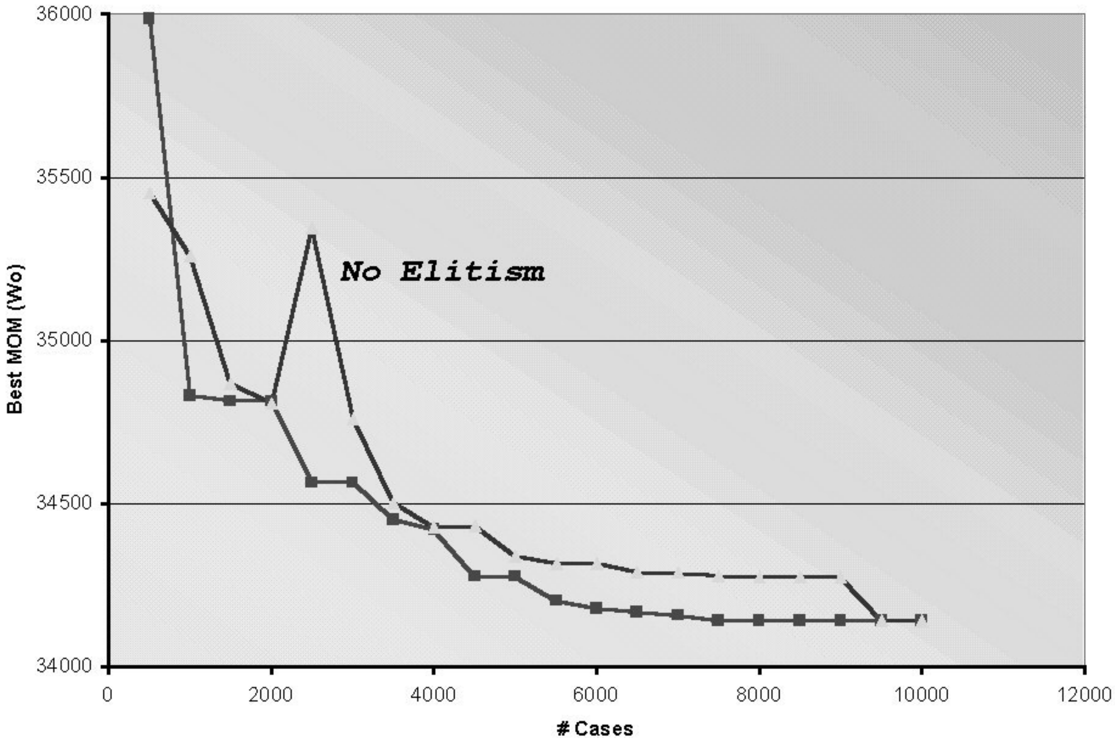


figure 63. Elitism Trade Study – On and Off (Fighter)

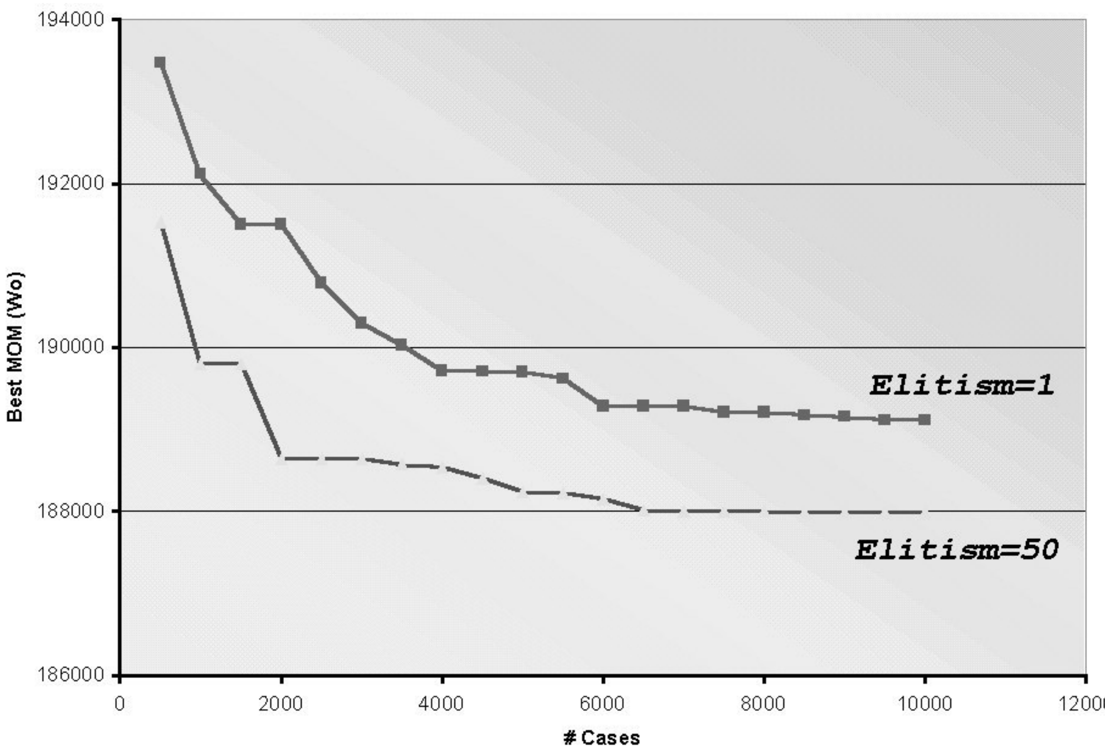


figure 64. Elitism Trade Study – 50 vs. 1 (Transport)

The other extreme of elitism is shown in figure 64. Here, an elite group of the 50 best individuals is selected for each generation and preserved into the next. This provides superb convergence in this example, with the final result virtually identical to the *OSD* best solution. This makes intuitive sense, because it provides each new generation with a cadre of hardened victors undiluted by mutation.

7.4.5 Miscellaneous Trade Studies

Runs 61-63 vary the size of the Breeder Pool from 5% to 75 % (the default value, determined by trial runs during code development, is 25%). Results indicate a better convergence and a better final result as pool size is reduced, making the optimization more elite. At the extreme, a pool size of one individual would be identical to the Killer Queen method except that, of course, the crossover operation would be problematic. 25% was left as the default on purely intuitive grounds.

The impact of population size was studied in runs 67-68. Other researchers often use population sizes much smaller than the 500 used by default in this research. This is probably a matter of necessity – their analysis methods do not permit conducting 10,000 parametric case evaluations on a routine basis. The population size rule-of-thumb offered in Section 4.5 would suggest that a population of 200 would be adequate for the chromosome bit-string used herein (4 times 48 bits in the string). Results indicate a faster convergence but slightly poorer results with a smaller population size. With a population of only 50 individuals, premature convergence is noted.

A mutation probability of 0.1% per bit (5% for each individual) was used as the default in these case runs, based on suggestions in Section 2.5.10 and on good results observed in early test runs. A trade study of mutation rate was done using the Tournament scheme, increasing mutation from 5% to 62% per individual (runs 64-66). The effect was minimal, but indicate worse results with high mutation which prevented convergence.

For the Evolutionary "Killer Queen" MDO method, mutation is essential as it is the only means of obtaining new genetic information in the next generation. It was varied from the default 95% down to 8 percent (runs 69-71). Surprisingly, this had little effect on the final best answer, but slowed the convergence somewhat.

To assess the impact of number of variables, a number of cases were run with the variables restricted from eight down to two (runs 77-87). Results are as expected – fewer variables results in a faster solution but, of course, with some design variables fixed the resulting aircraft cannot be quite as optimal.

7.5 Automated Aircraft Redesign Procedures

As discussed in Section 5, procedures were coded to revise the aircraft geometric inputs to analysis to properly reflect the changes to the aircraft design resulting from parametric changes to the selected design parameters. These were "on" during all of the MDO cases described herein, and inspection of the modified analysis input files indicates their reasonableness.

Runs 102-105 were done with these portions of the code deactivated to determine if these changes materially affect the resulting design. Results were not greatly significant, with W_0 differing by less than one percent for the cases studied. However, these design examples are good design examples which makes them bad test cases for automated redesign procedures. The as-drawn layout differs in takeoff gross weight by only 5-10% from the final optimized design and the optimized design parameters are also not greatly different from the baseline, so the effect is relatively small.

7.6 Net Design Volume

Net Design Volume (NDV) was defined in Section 5.5 as a method to prevent the optimizer from unrealistically reducing the wing size of a "tight" design without providing volume elsewhere for the fuel displaced. It was "off" for most of these MDO case runs, but was turned on for a specific evaluation of its effects (runs 88-92). Inspection of the results indicate that it is in fact making the fuselage a bit larger when the wing is parametrically made smaller in volume (area, thickness, or aspect ratio due to the fuselage cutoff effect). When permitted by user input, it also makes the fuselage smaller when the wing gets larger.

This last effect caused confusion because the use of *NDV* hold kept reducing the aircraft's takeoff gross weight by several percent, instead of increasing it as normally expected when another constraint is added to an optimization. The wing optimized larger so the *NDV* hold routine made the fuselage smaller, saving weight and drag.

A case run with "no-smaller" selected returned the unconstrained optimum indicating that the method was working properly. An attempt was made to force the optimizer to make the fuselage larger by relaxing the performance constraints that drove the wing loading (runs 90 and 91). The optimizer responded by making the wing area smaller as expected, but also made the wing thicker resulting in a smaller fuselage once again! When permitted, the optimizer *likes* to make the fuselage smaller and will grow the wing somehow to allow this.

For the small number of additional inputs and a minimal computational expense, *NDV* seems to provide a welcome note of realism in such optimizations – even if it outsmarts the user's attempt to demonstrate its effects.

7.7 Geometric Constraints

Options for constraining fuselage and wing geometry for real-world requirements were coded and tested in runs 95-100. For the fuselage, length can be constrained to a maximum input value while diameter can be constrained to an input minimum. For the wing, a span limit can be input. Also, the aspect ratio can be constrained to avoid pitchup as described in Section 5.4.3.

As should be expected, these constraints have no affect at all if the optimization moves away from them. A wingspan constraint will not affect the takeoff gross weight if the optimization is reducing the span anyway.

Run 93 shows a span limit on the transport design resulting in an 8% increase in takeoff gross weight. The optimizer was prevented from increasing the wing aspect ratio as much as it desired. In run 95 the fighter length was constrained to avoid exceeding the as-drawn baseline length (very common in fighter design such as for the F-22). This increased the aircraft price by 9% compared to the unconstrained optimum, which was lengthening the fuselage as a part of the optimization.

Application of the pitchup constraint to the commercial airliner was initially trivial because the optimum wing geometry was within the pitchup avoidance limits. To force a test case, the design maximum speed was increased to Mach.9, which leads to higher sweep. The pitchup-constrained M.9 optimum design is almost 50% heavier than the unconstrained design optimized for M.9, because the aspect ratio is held to a low value to avoid pitchup. Luckily, real airliners usually have large horizontal tails to avoid pitchup so the wing does not need to suffer this constraint. The other aircraft were already well within the pitchup limits so use of this constraint was trivial, and not recorded.

7.8 Comments on Ease of Programming

This author, who struggled for many months to perfect the classical carpet plot and Orthogonal Steepest Descent routines in the RDS-Professional program, found it remarkably easy to incorporate the chromosome-based optimization schemes into the existing aircraft analysis code. Also, it was much easier to add more variables into chromosome-based schemes than into *OSD* (not that it is trivial to get these methods to work properly – just easier!).

Easiest of all was the Monte Carlo routine, with literally a few dozen lines of code to create random chromosome/gene bit-strings and call the analysis routines. The only real logic required is simply to remember which of the designs is "best". Note that the other methods actually start with an initial population defined by the Monte Carlo routine.

The Evolutionary method used here ("Killer Queen") was also quite easy to program. Basically, all that was added to the Monte Carlo routine was an iterative loop of generations in which the best design found was copied and mutated.

The other schemes were all about equally easy (or difficult) to program. While the basic logic seems straightforward, allowing for numerous options and variations made the coding and debugging more difficult. Still, all of these chromosome-based methods together were easier to code and debug than the *OSD* method alone!

One interesting observation is that it is easy in these methods to allow the user to specify a range of a design variable that excludes the as-drawn baseline's value of that parameter (such as, the design is drawn with an aspect ratio of 8 but the user tells the code to investigate designs between 10 and 12). In the *OSD* coding this is forbidden because it must start from the as-drawn baseline design, and step away from it. But, perhaps the designer should draw it right the first time!

(This page intentionally blank)

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.

(This page intentionally blank)

9 APPENDICES

9.1 APPENDICE A – VARIABLES, CONSTRAINTS, & MOM's

Following are suggested design variables, constraints, and measures of merit as discussed in Section 2.7 above. In each category, the items are given roughly in order of this author's expectation as to importance and suitability for aircraft conceptual/preliminary MDO. Notice that some parameters of great importance, such as number of wing spars, are missing from these lists. Usually, such an inside-only change to wing structure would have minimal impact on the overall design other than in structural weight and cost, and hence can be optimized separately. In mathematical terms, it is expected that such parameters are weakly coupled with respect to the other design variables, and furthermore will have a much smaller impact on the objective function than those others. Such non-inclusion of non-global design variables is essential to hold down the computational time required for MDO.

Other parameters that might be noted as missing are, in this author's opinion, dependent variables. For example, the number of lavatories in a commercial airliner is driven by specifications based on number of passengers, so it should not be used as an independent design variable.

9.1.1 Recommended Design Variables

("independent variables" in the MDO)

THE BASIC SIX OR FIVE

- *T/W or P/W (unless fixed-size engine)*
- *W/S (area)*
- *Aspect Ratio*
- *Taper Ratio*
- *Sweep*
- *Airfoil t/c (constant or averaged)*

WING ENHANCEMENTS

- *Design Lift Coefficient or airfoil optimization*
- *Planform LE/TE breaks (% C at % b/2)*
- *Winglets (presence and geometry)*
- *Airfoil t/c distribution*
- *Twist or twist distribution*
- *Flap types*
- *Aileron / flap % span*
- *Aileron / flap % chord (vs. rear spar location)*

FUSELAGE VARIABLES

- *Length / diameter ratio ("fineness ratio") or diameter alone*
- *Nose or radome fineness ratio*
- *Upsweep angle or aft-end fineness ratio*
- *Volume distribution (optimize wave drag)*

PASSENGER CABIN / CARGO BAY

- *# Seats across*
- *# Aisles*
- *# Seats (if allowed by design objectives)*
- *Dimensions of cargo containers / pallets / hold*

ENGINE OPTIONS

- *# Engines*
- *Engine location(s)*

JET ENGINE VARIABLES

- *Bypass Ratio (BPR)*
- *Overall Pressure Ratio (OPR)*
- *Turbine Inlet Temperature (TIT)*
- *Capture Area*
- *Inlet lip radius*
- *Auxiliary door presence / size*
- *Inlet type (pitot, ramp, cone,other)*
- *Ramp or Cone angles (supersonic)*
- *Exhaust type*

PROPELLER VARIABLES

- *Propeller disk power loading*
- *Propeller blade power loading*
- *Blade Design Lift Coefficient*
- *Design Advance Ratio (fixed pitch)*
- *Blade Pitch Distribution*
- *Turbo/supercharging*
- *Cooling Options*

LANDING GEAR

- *# Mains*
- *Retracted Location (wing, fuselage, pods, other)*

TAIL GEOMETRY (HORIZ. & VERTICAL)

- *Tail Volume Coefficient (Area[****])*
- *Aspect Ratio*
- *Taper Ratio*
- *Sweep*
- *Airfoil t/c*
- *Location*
- *Tail Type*

9.1.2 Recommended Design Constraints

("constraint lines or surfaces" in the MDO)

PERFORMANCE REQUIREMENTS

- *Takeoff / landing distances*
- *Stall & approach speed*
- *Maximum speed*
- *Climb rate or time to climb*
- *Sustained & instantaneous turn rate*
- *Ps at given conditions*
- *Acceleration time/distance*
- *Glide ratio / sink speed*
- *Descent rate / time / distance*
- *Engine-out climb, takeoff, level flight*
- *Alternate mission range / endurance*

[****] Use of the tail areas as design variables is not the preferred practice, since required tail sizes are normally calculated from requirements for controllability, trim, and stability. Thus tail areas should be determined as constraints, not design variables. Or, better yet, the tail areas should be included as a part of the automated redesign so that every design variation meets requirements for controllability, trim, and stability.

GEOMETRIC LIMITATIONS

- *Cabin / payload minimum dimensions*
- *Wingspan*
- *Length*
- *Static Tail height above ground*
- *Spotting factor*
- *Gear span*
- *Tail-down angle (takeoff & landing)*
- *Stealth-driven angles, edges, etc.*

VOLUMETRIC

- *Fuel*
- *Avionics*
- *Subsystems & equipment*
- *Crew station*
- *Cabin or payload bay*
- *Net Design Volume Density (see Section 5.5)*

OPERATIONAL CONSTRAINTS

- *Weight-class limitation (ex: 12,500 for FAR23)*
- *Engine-out overwater flight time limits*
- *Carrier operability (weight, size, performance)*
- *Noise / sonic boom*
- *Environmental pollutant creation*
- *Signature*
- *Vulnerable area*

STABILITY & CONTROL CONSTRAINTS[††††]

- *Static longitudinal stability plus trim*
- *Static directional stability plus engine-out rudder trim*
- *Roll rate or time to roll*
- *6 DOF dynamic modes (spiral, phugoid,etc.)*
- *Above including flexible response*
- *Spin entry & recovery*

[††††] A better practice would be to include tail areas a part of the automated redesign so that every design variation meets requirements for controllability, trim, and stability.

9.1.3 Recommended Measures of Merit

("dependent variables" in the MDO)

WEIGHT-BASED

- *Sized takeoff gross weight*
- *Sized empty weight*
- *Fuel weight*
- *Structure weight*
- *AMPR / DCPR* weight*
- *Alternate mission useful load*

(*Airframe Manufacturer's or Defense Contractor's Planning Report - see Raymer[11])

COST-BASED

- *Development cost*
- *Procurement cost*
- *Purchase price*
- *Fuel cost*
- *Operating cost (yearly or per seat-mile)*
- *Life-cycle cost*

REVENUE-BASED

- *Gross or net revenue*
- *Break-even load factor*
- *Net Present Value*
- *Return on Investment*

UTILITY-BASED

- *Range / endurance (if fixed-size design)*
- *Ton-miles cargo delivered**
- *Combat Exchange ratio*
- *Percent of Targets "serviced" or "serviceable" **
- *Enemy assets destroyed**
- *Movement of Line of Troops**
- *Days to win the "war" **
- *Alternate mission range / endurance / payload*

(*at constant cost)

9.2 APPENDICE B – AIRCRAFT ANALYSIS METHODS[‡‡‡‡]

9.2.1 Overview of RDS Aircraft Design Software

Aircraft analysis for this research was done using the calculation modules of RDS-Professional, developed by this author over a ten-year period. RDS is a sophisticated PC-based aircraft design and analysis system developed for the conceptual design of new aircraft and the initial analysis of derivatives and alternate missions.

RDS comes in several versions. RDS-Student is available to assist students in the tedious analysis calculations of a typical "capstone" senior design class, leaving them more time to develop a design concept and perform the trade studies and optimizations that are a key and often-skipped portion of the design process. RDS-Homebuilt is aimed at the aircraft homebuilder community, providing them with a tool using industry methods at an affordable price. RDS-EZ is primarily for enthusiastic amateurs, offering simplified inputs consisting of only 20 inputs to define an aircraft for RDS analysis. RDS-Professional is the complete RDS program as developed for use by aircraft designers in industry, government, and academia for conceptual trade studies, technology evaluations, and preliminary performance predictions.

RDS features a 3-D CAD module for design layout, and has analysis modules for aerodynamics, weights, propulsion, and cost. Also included are aircraft sizing, mission analysis, and complete performance analysis including takeoff, landing, rate of climb, P_s, f_s, turn rate, and acceleration. RDS provides graphical output for drag polars, L/D ratio, thrust curves, flight envelope, range parameter, and more.

RDS follows the design and analysis methods of this author's textbook *AIRCRAFT DESIGN: A Conceptual Approach*[11]. These methods are distilled from the classical and time-proven first-order techniques commonly used in industry design groups for early visibility into design drivers and options. With RDS these methods are automated and incorporated into user-friendly modules that permit a tremendous quantity of calculation including trade studies and optimization in the early stages of design.

RDS is marketed through Conceptual Research Corporation (PO Box 923156, Sylmar, CA, 91392, USA), excepting RDS-Student which is marketed through AIAA (800 682 2422).

Following are brief discussions of the analysis methods employed by RDS as used in this research.

‡‡‡‡ Portions of this section excerpted and edited from Raymer, *AIRCRAFT DESIGN: A Conceptual Approach,* 1999[11], and Raymer, *RDS-Professional Users Manual*, 2000[69]. For permission to copy contact the author.

9.2.2 Aerodynamics & Stability

The RDS Aerodynamics Module estimates parasite drag (subsonic and supersonic), drag due to lift, lift curve slope, and maximum lift from a user-defined input matrix. Analysis methods are based upon classical techniques, as defined in Raymer[11].

Subsonic parasitic drag is estimated by the component buildup method, using a flat-plate skin friction drag coefficient including the effects of Reynolds Number and surface roughness, and a form factor to include the effects of viscous separation. Supersonic wave drag is determined by the equivalent Sears-Haack technique. Transonic drag is determined by empirical fairing between M_{dd} and the supersonic wave drag value, using a cubic polynomial curve.

Drag due to lift is calculated by the leading-edge suction method using lift-curve slopes ($C_{L\text{-}\alpha}$) calculated with DATCOM equations and graphs. The input design lift coefficient ($C_{L\text{-design}}$) is used to define the leading edge suction schedule. For subsonic, thick-wing designs the suction schedule is modified to provide 93% leading edge suction at all lift coefficients below $C_{L\text{-design}}$. For high aspect ratio wings, the program adjusts the leading edge suction schedule as a function of aspect ratio based on empirical data.

This analysis method provides a drag due to lift estimation based upon the wing planform as defined in the input file, but assuming that the airfoil, camber, and twist will be optimized during preliminary design by a competent design team. This feat is accomplished via the quasi-empirical nature of the leading-edge suction schedule.

Maximum lift is calculated using DATCOM charts for the "clean" wing, and a simplified flap analysis based on flapped wing areas for leading and trailing edge high-lift devices.

Longitudinal stability and control are evaluated using 1-DOF equations and methods such as DATCOM charts. RDS calculates the neutral point, pitching moment derivative, and other stability parameters, which are used to estimate trim drag. This is applied as an adjustment to the drag-due-to-lift factor *"K"*.

9.2.3 Weights

Weights and balance are calculated using long established and well-proven empirical equations. For fighter, attack, transport, and bomber aircraft, equations developed by Vought Aircraft[88,89] are employed with modifications and factors to make them more suitable for current and future designs. For General Aviation aircraft, equations developed at the AeroCommander Company (later Rockwell) are employed[90]. For other types of aircraft, these equations can be adjusted using "fudge factors" as was done for UAV's in a design study by this author[86]. Alternatively, the simple but effective "pounds per square feet" method can be employed.

The weights are estimated from an input design gross weight (W_{dg}). This may or may not be the same as the design takeoff gross weight (W_0), which is the total weight used for the group weight statement to calculate fuel available. For fighter aircraft especially, it is common to define W_{dg} as the aircraft weight after burning off some fuel. A typical W_{dg} is defined as the aircraft weight carrying some specified payload, with 50-60% of maximum internal fuel. This may be only about 85% of W_0. For further discussion and a full listing of weights equations, see Raymer[11].

No attempt is made herein to perform actual structural analysis. This follows the time-honored, classical means of aircraft weights analysis during the conceptual design stage. This is done for two reasons. First, the aircraft is not yet defined in enough detail to identify all the major structural components, nor are the loads known in any substantial detail. Second, even if it were possible to identify all the structural components and determine their critical sizing loads, that would only account for the portion of weights associated with structural strength.

There are many other things that cause weight, ranging from complicated aeroelastic effects to skin minimum gage requirements to thicknesses of finishes and coatings. To include these, *any* weight estimation method requires a substantial empirical aspect. There are reliable and proven statistical equations to estimate the entire weight of components such as wings and tails whereas this author is unaware of any proven equations to estimate the incremental weight of such components after the structural weight is determined separately.

Nor should the equations used herein be viewed as pure "blind" statistics. They were developed by starting with a simplified geometry of the various components, applying simplified loads, and determining skin thicknesses and finally required material volumes based on classical structural analysis equations. This resulted in purely analytical equations for calculating weight. To better reflect the realities of actual aircraft, the coefficients of these equations were adjusted by a least-squares process to better match data on dozens of existing aircraft.

9.2.4 Propulsion

Propulsion analysis includes installation analysis for jet engine thrust and specific fuel consumption, and propeller thrust and specific fuel consumption for piston-prop engines. Jet engine installation analysis takes an uninstalled engine file and applies corrections for pressure recovery, bleed, inlet drag, and nozzle drag. Defaults are provided for the standard MIL-E-5008B reference pressure recovery schedule and typical actual pressure recovery schedules for various types of inlets. Part-power data can be included as input tables, corrected for installation, or can be approximated by an analytical approximation developed for this author by J. Mattingly (see Raymer[11]).

Propeller analysis calculates thrust and specific fuel consumption from inputs for horsepower and brake specific fuel consumption as functions of altitude, propeller efficiency as a function of Advance Ratio (J), and static thrust coefficient ratio as a function of power coefficient. Analysis includes a blockage adjustment for the effect of the nacelle behind the propeller. A propeller tip Mach number correction is employed, along with scrubbing, cooling, and miscellaneous drag adjustments. Static thrust is estimated, then an empirical fairing is used between the static value and the forward flight thrust results.

9.2.5 Cost

For cost analysis of the development and procurement (purchase) phases of aircraft acquisition, equations developed by the RAND Corporation are used. These are known as *DAPCA*, the Development and Procurement Costs of Aircraft model[91]. *DAPCA* is notable in that it seems to provide reasonable results for several classes of aircraft including fighters, bombers, and transports.

DAPCA estimates the hours required for developmental research, design, and engineering and for production as broken out into engineering, tooling, manufacturing, and quality control groups. These are multiplied by the appropriate hourly rates to yield costs. Development support, flight-test, and manufacturing material costs are directly estimated by *DAPCA* but must be corrected to reflect inflation in these areas. As *DAPCA* is empirical in nature, it requires knowledgeable "fudge-factoring" to get good results for a specific class of aircraft.

For operating expenses RDS employs tabular buildup methods, defining expected costs in future years based on historical data and anticipated operations, and multiplying these costs by the expected inflation rates.

9.2.6 Performance

RDS uses equations derived from physics (primarily F=ma) to calculate the commonly-used aircraft performance measures including:

- Flight Envelope
- Range Performance (nmi/lb fuel)
- Loiter Performance (seconds/lb fuel)
- Climb (Rate, angle, and optimal schedule)
- f_s Calculation (measure of climb fuel efficiency)
- P_s (Specific Energy)
- Turn Rate
- V-n & Gust Response
- Takeoff (ground roll, obstacle clearance, or Balanced Field Length)
- Landing
- Acceleration (time and distance)

These can all be employed as performance constraints in optimization, with must-meet values specified by user input. Equation derivations can be found in Raymer[11].

9.2.7 Sizing

Sizing is the process of calculating the required takeoff gross weight of a new design to perform some given mission. Mission analysis determines the range capability of a design (new or existing) where the design takeoff gross weight is known and unchangeable. Both are done in RDS using a detailed mission model defined by the user. RDS mission and range analysis is highly sophisticated and can include factors such as headwind, stall margin, service ceiling, and maximum descent rate, and it allows in-the-loop optimization of speeds, altitudes, and climb schedules. Such mission complexity is required to be suitable for use on real aircraft projects.

Virtually all types of mission segments are available, allowing the user to build a mission model of any type desired (orbital insertion is not currently supported, but may be added soon!). Up to 50 mission segments can be used allowing great detail in the mission modeling. Default mission models are available including fighter, attack, general aviation, and commercial transport. Weight drops and refuels are also supported.

Velocities can be entered as true airspeed, calibrated airspeeds, or as Mach number. Alternatively, RDS-Professional can find optimal altitudes and velocities for cruise and loiter. RDS checks to ensure that speeds are neither below stall speed at that altitude, nor above maximum speed at that altitude and power setting. If either is found, RDS fixes it by changing velocity. This also permits setting a desired power setting and letting RDS find the resulting aircraft speed. Separate stall margins for cruise and loiter are supported. Headwinds and non-standard atmospheric models can be applied.

Sizing includes the use of a range credit for climbs and descents. In the case of climbs, the distance required to perform the climb is calculated, saved and subtracted from the next cruise segment. During a descent, the program can calculate descent rate and the range credit for descent, and can determine best descent speed to maximize descent range.

Rather than perform sizing in which the aircraft weight is parametrically varied until a desired range is attained, the reverse process can be performed using the same mission model. This finds the range that will be attained by the as-drawn aircraft.

9.2.8 Brief Survey of Other Aircraft Design Programs

This research effort has been done exclusively using the RDS-Professional computer program which is developed and marketed by this author. Other integrated design-analysis codes exist, some of which are widely used. A brief overview of several is provided below, based on publicly available information (this information may be incomplete or out of date, and the reader is referred to the organizations providing these codes for better information).

9.2.8.1 ACSYNT

ACSYNT (AirCraft SYNThesis) grew out of a sizing code started many years ago at NASA-Ames, and now includes modules and features developed by a variety of companies and universities. It runs on workstations and has sophisticated sizing and optimization capabilities, and includes full vehicle analysis including aerodynamics, weights, propulsion, performance, cost, etc. ACSYNT has a parametric design capability in its CAD module. It is considered an "experts" code requiring training and learning time to use it properly.

9.2.8.2 AAA

Built on the aircraft design textbook series of Jan Roskam, AAA (Advanced Aircraft Analysis) runs on workstations. There is also a PC version. AAA contains full vehicle analysis with special emphasis on stability and control. It uses a commercial CAD package. It offers design and analysis capabilities at a simplistic "Level One" where minimal inputs are required, and at a better "Level Two" more like RDS and the other codes described here.

9.2.8.3 FLOPS

FLOPS (Flight Optimization System) is a Fortran 77, workstation-based code with capabilities for conceptual and preliminary design and evaluation of advanced aircraft concepts. It was developed over a 20-year period with funding from NASA-Langley, and is available to organizations not competing with the US aerospace industry. It is widely used in university MDO development, as well as by aerospace firms and government agencies. It apparently does not have a CAD module but accepts inputs from them.

9.2.8.4 PIANO

A commercially-developed code from the UK, Piano is largely aimed at conventional configuration, subsonic commercial transports, especially in its CAD, weights, and aerodynamic modules.The aircraft is defined by roughly 240 input parameters in a tabular form. Piano includes some two-variable parameteric study capability, and a multivariable optimizer based on a form of zeroth-order stepping search using mass, wing area, thrust, aspect ration, and sweep as variables (*t/c*, taper, and flaps are also available but not recommended).

9.2.8.5 IDAS

Developed at Rockwell International during the 1980's, this program integrated earlier codes into a user-friendly, workstation-based package with full capabilities for 3-D design layout, analysis, sizing, and optimization. IDAS (Integrated Design and Analysis) incorporated the sophisticated Rockwell Vehicle Sizing Program (VSP) and this author's Configuration Development System[92] (CDS, later renamed CDM), plus aerodynamics, weights, and propulsion modules crafted from several sources. IDAS/CDM is noteworthy in its sophisticated implementation of a wide suite of conceptual design-specific routines. These unique tools allow rapid creation of a credible aircraft configuration including Bezier-lofted surfaces and complete internal definition (cockpit, payload, fuel tanks, engine, inlets, major subsystems, structure, etc). Both the X-31 and the B-1B were designed using CDM in the conceptual phase. Neither IDAS nor CDM are publicly available, and both are clearly "experts" codes.

9.3 APPENDICE C – TEST-CASE RUN MATRIX

Run	Aircraft	MOM	MOM result	# cases	Method	BtStrAf	%MeetPerf	MOM Wt	Penalty	Mutation%	Crossover	
1	Ftr	$	34.1m	20106	OSD			Abs				
2	CivB	Wo	188.1k	16254	OSD			Abs				Bad Civ Tr Design
3	GA	Wo	2099	7398	OSD			Abs				fixed-size engine -no T/W
4	UAV	Wo	6493	2514	OSD			Abs				No T/W, No fus FineNess
5	Ftr	$	38.6m	81	OSD			Abs				2 variables - like carpet
6	CivB	Wo	<see #47>		OSD			Abs				won't run - can't meet perf
7	GA	Wo	2165	54	OSD			Abs				2 variables
8	UAV	Wo	7104	63	OSD			Abs				2 variables
9	Civ	Wo	187.9k	20546	OSD			Abs	Civ="Good"			Original Good Design
10	CivB	Wo	211.6k	22842	OSD			Abs	CivB="Bad"			20% var range - too small
11	Ftr	$	34.6m	10000	Mont			Abs				
12	CivB	Wo	192.3k	10000	Mont			Abs				Expanded Variable Range
13	GA	Wo	2109	10000	Mont			Abs				
14	UAV	Wo	6512	10000	Mont			Abs				
15	Ftr	$	34.1m	10000	Pool	92	98	Lin	1.2	5	Unif	
16	CivB	Wo	189.1k	10000	Pool	92	99	Lin	1.2	5	Unif	Expanded Variable Range
17	GA	Wo	2107	10000	Pool	91	99	Lin	1.2	5	Unif	
18	UAV	Wo	6490	10000	Pool	77	88	Lin	1.2	5	Unif	
19	Ftr	$	34.4m	10000	Tmnt	47	91	Lin	1.2	5	Unif	
20	CivB	Wo	190.2k	10000	Tmnt	54	99	Lin	1.2	5	Unif	Expanded Variable Range
21	GA	Wo	2109	10000	Tmnt	54	87	Lin	1.2	5	Unif	
22	UAV	Wo	6522	10000	Tmnt	58	99	Lin	1.2	5	Unif	
23	Ftr	$	34.7m	10000	Roult	23	41	Lin	1.2	5	Unif	
24	CivB	Wo	190.6k	10000	Roult	23	73	Lin	1.2	5	Unif	Expanded Variable Range
25	GA	Wo	2124	10000	Roult	30	36	Lin	1.2	5	Unif	
26	UAV	Wo	6527	10000	Roult	23	58	Lin	1.2	5	Unif	
27	Ftr	$	34.1m	10000	KillQ	88	32	Lin	1.2	95	Unif	
28	CivB	Wo	190k	10000	KillQ	87	44	Lin	1.2	95	Unif	Expanded Variable Range
29	GA	Wo	2100	10000	KillQ	87	45	Lin	1.2	95	Unif	
30	UAV	Wo	6523	10000	KillQ	88	91	Lin	1.2	95	Unif	
31	Ftr	$	34.1m	20106	OSD			Abs				Repeat of baseline
32	Ftr	$	35.0m	10000	Mont			Abs				"
33	Ftr	$	34.1m	10000	Pool	85	94	Lin	1.2	5	Unif	"
34	Ftr	$	34.3m	10000	Tmnt	47	91	Lin	1.2	5	Unif	"
35	Ftr	$	34.6m	10000	Roult	19	38	Lin	1.2	5	Unif	"
36	Ftr	$	34.1m	10000	KillQ	88	30	Lin	1.2	95	Unif	"

Run	Aircraft	MOM	MOM result	# cases	Method	BtStrAf	%MeetPerf	MOM Wt	Penalty	Mutation%	Crossover	
37	Ftr	$	35.0m	10000	Roult	20	31	^2	1.2	5	Unif	test of MOM weights
38	Civ	Wo	191K	10000	Roult	19	69	^2	1.2	5	Unif	"
39	Ftr	$	34.8m	10000	Roult	21	32	^4	1.2	5	Unif	"
40	Civ	Wo	192.6k	10000	Roult	22	43	^4	1.2	5	Unif	"
41	Ftr	$	35.1m	10000	Roult	23	43	Cos	1.2	5	Unif	"
42	Civ	Wo	190.2K	10000	Roult	23	77	Cos	1.2	5	Unif	"
43	Ftr	$	34.4m	10000	Pool	88	99	Lin	1.2	5	Bisc	
44	Civ	Wo	188.8k	10000	Pool	95	98	Lin	1.2	5	Bisc	
45	Ftr	$	34.1m	10000	Pool	86	88	Lin	1.2	5	Prm	
46	Civ	Wo	189.2k	10000	Pool	92	98	Lin	1.2	5	Prm	
47	CivB	Wo	221k	57	OSD			Abs				2 variables, no perf#5
48	Ftr	$	34.3m	10000	Pool	90	84	Lin	1to2	5	Unif	Simulated Anneal
49	Civ	Wo	188.4k	10000	Pool	96	98	Lin	1to2	5	Unif	"
50	Ftr	$	34.4m	10000	Tmnt	41	88	Lin	1to2	5	Unif	"
51	Civ	Wo	189.8k	10000	Tmnt	43	83	Lin	1to2	5	Unif	"
52	Ftr	$	34.1m	10000	Pool	88	94	Lin	1.2	5	Unif	No Elitism
53	Civ	Wo	189.0k	10000	Pool	80	86	Lin	1.2	5	Unif	No Elitism
54	Ftr	Wo	36345	19054	OSD			Abs	{price $34.1m}			Weight as MOM
55	Ftr	Wo	36970	10000	Roult	22	43	Lin	1.2	5	Unif	"
56	Ftr	$	34.1m	11500	Pool	99	99	Lin	1.2	5	Unif	longer runs - 50k
57	Civ	Wo	189k	14500	Pool	98	99	Lin	1.2	5	Unif	both stopped by BitStAff
58	UAV	Wo	6495	10000	Pool	79	99	Lin	1.2	5	Unif	No Elitism
59	Ftr	$	34.1m	10000	Pool	88	94	Lin	1.2	5	Unif	Elitism=50
60	Civ	Wo	188.0k	10000	Pool	99	99	Lin	1.2	5	Unif	Elitism=50 - stopped
61	Ftr	$	34.1m	10000	Pool	99	98	Lin	1.2	5	Unif	Pool=5%-stopped at 6000
62	Ftr	$	34.3m	10000	Pool	55	88	Lin	1.2	5	Unif	Pool=50%
63	Ftr	$	34.8m	10000	Pool	37	95	Lin	1.2	5	Unif	Pool=75%
64	Civ	Wo	189.3k	10000	Tmnt	31	87	Lin	1.2	62	Unif	mutation 50 (62%)
65	Civ	Wo	188.9k	10000	Tmnt	45	89	Lin	1.2	21	Unif	mutation 200 (21%)
66	Civ	Wo	188.7k	10000	Tmnt	52	91	Lin	1.2	8	Unif	mutation 600
67	Ftr	$	34.2m	10000	Pool	98	94	Lin	1.2	5	Unif	Population 50 - stopped
68	Ftr	$	34.1m	10000	Pool	99	98	Lin	1.2	5	Unif	Population 250 - stopped
69	Civ	Wo	188.9k	10000	KillQ	96	75	Lin	1.2	62	Unif	mutation 50 (62%)
70	Civ	Wo	192.0k	10000	KillQ	99	92	Lin	1.2	21	Unif	mutation 200 -stopped
71	Civ	Wo	189.4k	10000	KillQ	99	98	Lin	1.2	8	Unif	mutation 600 -stopped

Run	Aircraft	MOM	MOM result	# cases	Method	BtStrAf	%MeetPerf	MOM Wt	Penalty	Mutation%	Crossover	
72	Ftr	$	34.3m	20360	Mont							Hybrid-4 gens then OSD
73	Ftr	$	34.2m	21764	Pool							Hybrid-4 gens then OSD
74	Ftr	$	34.5m	19334	Tmnt							Hybrid-4 gens then OSD
75	Ftr	$	34.2m	42562	Tmnt							Hybrid-20 gens then OSD
76	Ftr	$	34.2m	23708	KillQ							Hybrid-4 gens then OSD
77	Ftr	$	38.3m	10000	Pool	85	98	Lin	1.2	5	Unif	No Finess Ratio (7 var)
78	CivB	Wo	189.0k	10000	Pool	87	98	Lin	1.2	5	Unif	No Finess Ratio
79	GA	Wo	2106	10000	Pool	80	99	Lin	1.2	5	Unif	No Finess Ratio
80	Ftr	$	37.8m	10000	Pool	70	92	Lin	1.2	5	Unif	No f,cl-d (6 variables)
81	Ftr	$	38.2m	10000	Pool	75	97	Lin	1.2	5	Unif	No f,cl-d,t/c (5 variables)
82	Ftr	$	38.3m	10000	Pool	71	98	Lin	1.2	5	Unif	No f,cl-d,t/c,taper (4)
83	Ftr	$	38.5m	10000	Pool	62	88	Lin	1.2	5	Unif	No f,cl-d,t/c,taper,swp (3)
84	Ftr	$	38.6m	6561	OSD							No f,cl-d (6 variables)
85	Ftr	$	38.6m	2835	OSD							No f,cl-d,t/c (5 variables)
86	Ftr	$	38.4m	729	OSD							No f,cl-d,t/c,taper (4)
87	Ftr	$	38.6m	243	OSD							No f,cl-d,t/c,taper,swp (3)
88	Ftr	$	33.6m	10000	Pool	87	98	Lin	1.2	5	Unif	NDV hold
89	Ftr	$	34.1m	10000	Pool	91	98	Lin	1.2	5	Unif	NDV hold-no smaller
90	Ftr	$	33.7m	10000	Pool	96	87	Lin	1.2	5	Unif	Landing=3000,Ps=4
91	Ftr	$	32.9m	10000	Pool	99	99	Lin	1.2	5	Unif	NDV hold L=3000,Ps=4
92	Civ	Wo	187.6k	10000	Pool	91	99	Lin	1.2	5	Unif	NDV hold
93	Civ	Wo	202.9k	10000	Pool	91	0.4	Lin	1.2	5	Unif	Span limit=110ft
94	Civ	Wo	193.0k	10000	Pool	84	75	Lin	1.2	5	Unif	Diameter limit=13.5ft
95	Ftr	$	37.0m	10000	Pool	86	0.2	Lin	1.2	5	Unif	Length limit=54ft (base)
96	Ftr	$	34.5m	10000	Pool	66	75	Lin	1.2	5	Unif	Span limit=35ft base=38.4
97	Civ	Wo	202.6k	10000	Pool	91	97	Lin	1.2	5	Unif	Mcruise=.9 - high sweep
98	Civ	Wo	299.8k	10000	Pool	80	0.2	Lin	1.2	5	Unif	Pitchup-subsonic Mcr=.9
99	UAV	Wo	6493	10000	Pool	76	98	Lin	1.2	5	Unif	Pitchup - transsonic
a0	UAV	Wo	6493	10000	Pool	81	99	Lin	1.2	5	Unif	taper=.2
a1	GA	Wo	2130	10000	Tmnt	59	56	Lin	1.2	5	Unif	Diameter limit=4.2ft
a2	GA	Wo	2087	10000	Tmnt	59	88	Lin	1.2	5	Unif	Remove Tail Scaling
a3	GA	Wo	2097	10000	Tmnt	57	86	Lin	1.2	5	Unif	Remove Gear Scaling
a4	CivB	Wo	187.9k	10000	Pool	80	97	Lin	1.2	5	Unif	Remove Gear Scaling
a5	CivB	Wo	190.0k	10000	Pool	82	91	Lin	1.2	5	Unif	Remove Tail Scaling
a6	CivB	Wo	190.0k	10000	Pool	82	91	Lin	1.2	5	Unif	NDV hold

9.4 APPENDICE D – SAMPLE CASE

Tournament Optimization of Civil Transport

- **Two Generations, Population of 10**

```
RDS MULTIVARIABLE OPTIMIZATION HISTORY
METHOD: TOURNAMENT (1v1 competition to breed) [.GA3]
FPS Units
Required performance values
 9842.52  9842.52  5  8.1  6  1

Key to printout:
Variation #, Gene String
T/W, W/S, A, Sweep, Taper, t/c, Fuselae Fineness, CL-design
MEASURES OF MERIT (Wo,We,Wf,Price,LCC,NPV,IRR)
PERFORMANCE RESULTS (up to 20 items)   Yes if performance met

INITIAL POPULATION RANDOMLY GENERATED
1 , 110000110110101000011111100000000100011101001011 1
 .380259,  122.0692,  5.269842,  39.87302,  .2507937,  .1020952,  13.16134,
.5203175
 229355, 124993.6, 70411.15
 8517.375, 6733.47, 7.505166, 50.99729, 3.299648, 57.82096, No
Wing Span= 99.5061217492389
Fus Length= 165.775057275502
Diameter= 12.5956045361624

2 , 000000010000000001000101011101001000110010111010 0
 .27536,  96.29905,  4.063492,  34.53968,  .2920635,  .1089524,  11.65923,
.6215873
 262931.8, 123777.9, 105203.7
 8368.22, 5310.422,-13.67319, 16.65125,-10.45968, 35.84304, No
Wing Span= 105.331989719927
Fus Length= 160.019685329937
Diameter= 13.7247172275394

3 , 010101100110011101000011100011001101011000010100
 .3212533,  111.2186,  4.920635,  32.76191,  .2555556,  .1059048,  10.72936,
.5098413
 210415.3, 110274.5, 66190.55
 8610.917, 6021.699,-3.638832, 31.74504,-3.523738, 45.38945, No
Wing Span= 96.4851441744784
Fus Length= 140.563998647027
Diameter= 13.1008797397607

4 , 111100001100011111000000101111101011101000110010 1
 .4064838,  93.5864,  4.984127,  32.25397,  .2492063,  .1135238,  10.22865,
.5692064
 227611.3, 121652.2, 72008.78
 5387.423,  5213.629,  10.11452,  45.63326,  7.098994,  67.33411,  Yes,New
Best
Wing Span= 110.09942582291
Fus Length= 139.762648775157
Diameter= 13.6638388994439
```

5 , 10001001000000011110101011011001111100100001001
.3496635, 96.29905, 4.222222, 42.66667, .2857143, .119619, 11.5877, .4714286
224631.5, 117996.7, 72684.65
6916.048, 5661.359,-1.345271, 38.84706,-1.965849, 54.11435, No
Wing Span= 99.2418663256817
Fus Length= 151.224713845776
Diameter= 13.0504460898384

6 , 01001101110010101000100000000110100101100100 1110
.3168825, 104.437, 5.333333, 34.03175, .2015873, .1272381, 10.80088, .4888889
207225.9, 108765.5, 64510.14
8076.764, 5751.098,-2.346864, 29.22845,-1.832376, 46.24191, No
Wing Span= 102.871302772553
Fus Length= 140.472076888032
Diameter= 13.0056084753587

7 , 11001001000100100110110011111100101111000101010
.3846298, 96.97721, 4.285714, 43.17461, .3, .104381, 13.01828, .5866667
237148.2, 127407.4, 75790.58
6381.067, 5696.405, 5.351473, 47.8369, 2.300084, 62.75151, No
Wing Span= 102.373289950678
Fus Length= 166.412392712085
Diameter= 12.7829750080982

8 , 10010110111111101010011101101111011100000000 1001
.3562197, 117.3221, 5.84127, 41.90476, .2428571, .1379047, 9.012659, .4714286
213413.9, 113344.1, 66119.52
8866.854, 6607.014, 3.224084, 45.57382, 2.193734, 54.49562, No
Wing Span= 103.080162807091
Fus Length= 125.723017662626
Diameter= 13.9496027852347

9 , 11111011100001010100100011000010111101010110 0111
.4108546, 123.4255, 4.666667, 34.03175, .2761905, .1318095, 10.51477, .5761905
231952.1, 120479.4, 77522.48
7281.957, 6383.75, 5.590257, 55.21342, 2.624485, 63.0225, No
Wing Span= 93.6483257676556
Fus Length= 143.256643879244
Diameter= 13.6243270639625

10 , 01011000010011000011111110111100010100101110 1011
.3234387, 88.1611, 5.52381, 48, .2746032, 9.980952E-2, 9.799478, .5901588
220789, 120919.5, 65919.21
7300.002, 5586.925, 1.425374, 34.3455, .8175545, 52.15464, No
Wing Span= 117.616849490196
Fus Length= 134.453949748617
Diameter= 13.7205216463413

```
POPULATION (# 1) OF  10 CREATED AND ANALYZED
BEST MOM =  227611.3 FOR VARIATION # 4
DESIGN VARIABLES OF BEST:
.4064838,  93.5864,  4.984127,  32.25397,  .2492063,  .1135238,  10.22865,
.5692064
 10% MEET ALL REQUIREMENTS
GENE STRING BIT AFFINITY (0=RANDOM 100%=IDENTICAL):  22.91668
```

```
TOURNAMENT BEGINS ! (Hopeful number:weighted MOM)
Hopeful 1 # 5: 323469.5
Hopeful 2 # 8: 307316
Winner is  8
Hopeful 1 # 6: 298405.3
Hopeful 2 # 8: 307316
Winner is  6
   Parent 1: 1001011011111110101001110110111101110000000001001
   Parent 2: 0100110111001010100010000000011010010110010011 10
      Child: 1100010111101010101001000010111000110100000001001
      Child: 1001111011111110100010010010111011110010000001000
Hopeful 1 # 7: 284577.8
Hopeful 2 # 2: 378621.8
Winner is  7
Hopeful 1 # 9: 278342.5
Hopeful 2 # 4: 227611.3
Winner is  4
   Parent 1: 1100100100010010011011001111110010111110001010 10
   Parent 2: 1111000011000111110000010111110101110100011001 01
      Child: 1100000001000110010000011111110001111100011011 10
      Child: 1101100101010011110001001111110100110110001011 11
Hopeful 1 # 3: 302998
Hopeful 2 # 2: 378621.8
Winner is  3
Hopeful 1 # 1: 275226
Hopeful 2 # 5: 323469.5
Winner is  1
   Parent 1: 0101011001100111010000111000110011010110000101 00
   Parent 2: 1100001101101010000111111000000001000111010010 111
      Child: 1101001101100111000010111000010011010110100101 10
      Child: 1101011001100111010100111000000011000110000101 10
Hopeful 1 # 3: 302998
Hopeful 2 # 9: 278342.5
Winner is  9
Hopeful 1 # 1: 275226
Hopeful 2 # 9: 278342.5
Winner is  1
   Parent 1: 1111101110000101010010001100001011110101011001 11
   Parent 2: 1100001101101010000111111000000001000111010010 111
      Child: 1111001100001100010110111000001010011110011001 11
      Child: 1111101101100100010010111100001011101111110001 11
Hopeful 1 # 8: 307316
Hopeful 2 # 9: 278342.5
Winner is  9
Hopeful 1 # 7: 284577.8
Hopeful 2 # 1: 275226
Winner is  1
```

```
   Parent 1: 111110111000010101001000110000101111010101100111
   Parent 2: 110000110110101000011111100000001000111010010111
      Child: 111110111010110101001101100000101111011110000111
      Child: 110000110010010100001111110000000110001101110011 1
```

BEGIN SECOND GENERATION ANALYSIS

1 , 100111101111111010001001001011101111001000001000
 .3605905, 117.3221, 5.84127, 34.28571, .2174603, .1318095, 9.584891, .4679365
 211186.3, 112030.2, 65205.94
 8268.517, 6319.771, 4.24267, 43.2206, 2.843147, 55.14505, No
Wing Span= 102.54079627236
Fus Length= 130.53564987749
Diameter= 13.6188972349498

2 , 110001011110101010100100001011100011010000001001
 .3824444, 105.7933, 5.333333, 41.14286, .2174603, .1226667, 10.15712, .4714286
 221726.6, 119318.7, 68457.64
 6999.794, 6026.978, 7.786603, 48.79808, 4.775083, 62.02418, No
Wing Span= 105.725355994146
Fus Length= 137.90186483963
Diameter= 13.5768624292498

3 , 110110010101001111000100111111010011011000101111
 .3933714, 99.68986, 4.476191, 33.01587, .3, .1104762, 10.72936, .604127
 227766, 120184.7, 73631.06
 5845.558, 5425.399, 6.008014, 45.06517, 3.216, 63.27497, No
Wing Span= 101.128428517945
Fus Length= 144.322217342599
Diameter= 13.451154146012

4 , 110000000100011001000001111111000111110001101110
 .380259, 88.1611, 4.793651, 32.25397, .3, .1013333, 12.51758, .600635
 226319.2, 122420.8, 69948.21
 5510.879, 5066.134, 7.36365, 36.99208, 4.710142, 61.09959, No
Wing Span= 110.931635514749
Fus Length= 159.61183513913
Diameter= 12.7510132355799

5 , 110101100110011101010011100000000110001100001011 0
 .391186, 111.2186, 4.920635, 36.8254, .2507937, .1051429, 10.72936, .5168254
 227438.6, 121558.9, 71929.54
 7305.543, 6144.789, 7.305083, 49.97638, 3.738399, 61.98651, No
Wing Span= 100.312242291144
Fus Length= 144.253103816387
Diameter= 13.4447126104552

6 , 110100110110011100001011100001001101011010010110
 .3890006, 122.0692, 4.888889, 34.79365, .252381, .1059048, 10.87241, .5168254
 227827.2, 120577.2, 73299.72
 8038.021, 6526.692, 5.252568, 50.42739, 2.006806, 58.71058, No
Wing Span= 95.52229881834
Fus Length= 145.615891996229
Diameter= 13.3931526296248

7 , 1111101101100100010010111100001011101111111000111
 .4108546, 122.0692, 4.539682, 34.79365, .2761905, .1310476, 13.51899, .4644445
 235270.9, 124133.4, 77187.29
 7548.76, 6513.693, 7.098033, 55.35388, 2.685565, 62.6572, No
Wing Span= 93.5392292367575
Fus Length= 170.203582751528
Diameter= 12.5899660729315

8 , 1111001100001100010110111000001010011110011001111
 .4064838, 118.0002, 5.555556, 38.85714, .2507937, .1272381, 13.08981, .5761905
 229106.6, 125003.6, 70152.69
 7376.301, 6414.033, 12.58991, 56.21288, 7.359319, 65.74718, Yes,New Best
Wing Span= 103.858269856118
Fus Length= 165.113985288964
Diameter= 12.6139294976993

9 , 1100001100100101000011111100000011000110111001111
 .380259, 119.3566, 4.634921, 35.80952, .2761905, .1051429, 10.94394, .5761905
 226848, 119566.1, 73331.73
 7822.668, 6360.712, 3.040273, 48.17245, .2064644, 57.14243, No
Wing Span= 93.8567464423754
Fus Length= 146.044571666202
Diameter= 13.3447861367965

10 , 1111101110101101010011011000001011110111100001111
 .4108546, 124.7819, 5.68254, 35.30159, .2507937, .1318095, 11.15853, .4644445
 225844.3, 121364.1, 70529.94
 7835.091, 6680.297, 11.87431, 56.84501, 7.041446, 64.74426, Yes,New Best
Wing Span= 101.414503217791
Fus Length= 147.730105919846
Diameter= 13.2392095915408

POPULATION (# 2) OF 10 CREATED AND ANALYZED
BEST MOM = 225844.3 FOR VARIATION # 10
DESIGN VARIABLES: .4108546, 124.7819, 5.68254, 35.30159, .2507937, .1318095, 11.15853, .4644445
20% MEET ALL REQUIREMENTS
GENE STRING BIT AFFINITY (0=RANDOM 100%=IDENTICAL): 39.58334

Gen#	Best MOM	%Meeting Perf	Gene Bit-String Affinity
1	227611.3	10	22.91668
2	225844.3	20	39.58334

(This page intentionally blank)

10 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] P. E. MacMillin et al**, An MDO Investigation of the Impact of Practical Constraints on an HSCT Configuration** , AIAA paper 97-0098

[5] L. Blasi et al., **Conceptual Aircraft Design Based on a Multiconstraint Genetic Optimizer**, *AIAA Journal of Aircraft,* Vol 37 No. 2, March-April 2000

[6] Raymer, D., **Multivariable Aircraft Optimization on a Personal Computer**, SAE/AIAA Paper 9965609, World Aviation Congress, 24 Oct. 1996

[7] Gill, P., Murry, W., and Wright, M., ***Practical Optimization***, Academic Press (Harcourt Brace & Co.), London, UK, 1981

[8] Nicolai, L., ***Fundamentals of Aircraft Design***, E. P. Domicone Printing Services, Fairborn, OH, 1975.

[9] Roskam, J., ***Airplane Design,*** Roskam Aviation & Engineering Corp., Ottawa, Kansas, 1985

[10] Stinton, D. ***The Design of the Aeroplane***, BSP Professional Books, 1983.

[11] Raymer, D., ***AIRCRAFT DESIGN: A Conceptual Approach***, American Institute of Aeronautics and Astronautics, Washington, D.C., Third Edition 1999

[12] Robinson, D., *E-mail Communication*, Boeing/Rockwell/North American Aviation (retired), 9-12-2000

[13] Sobieszczanski-Sobieski, J., *E-mail Communication*, NASA-Langley, 3-25-02

[14] Wall, R. and Fulghum, D., **Lockheed Martin Strikes Out Boeing**, *Aviation Week*, 10-29-2001, page 32

[15] Meyer, R., **57th SAWE International Conference Keynote Address Transcript**, *Weight Engineering*, Society of Allied Weights Engineers, Fall 1998

[16] Hildegrandt, G., and Man-bing, S., ***An Estimation of USAF Aircraft Operating and Support Cost Relationships,*** RAND Note N-3062-ACQ, 1990

[17] Raymer, D., ***Aircraft Computer-Aided Design and Computer-Aided Manufacturing**,* RAND Corp. PM-611-AF, 1996

[18] Jameson, A., **Re-Engineering the Design Process through Computation**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[19] Vos, J., Rizzi, A., Darracq, D., and Hirschel, E., **Navier-Stokes Solvers in European Aircraft Design**, *International Journal of Progress in the Aeronautical Sciences*, Pergamon, Oxford, U.K. (publication expected 2002)

[20] Hoak, D., et al., ***USAF DATCOM***, Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH.

[21] Van der Velden, A., Kelm, R., Kokan, D., and Mertens, J., **Application of MDO to Large Subsonic Transport Aircraft**, AIAA Paper 00-0844, 38th AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 2000

[22] Giesing, J., and Barthelemy, J., **A Summary of Industry MDO Applications and Needs**, AIAA Paper 98-4737, 7th Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 1998

[23] Corning, G., ***Airplane Design***, (self-published), College Park, MD, 1960

[24] Prandtl, L., **Uber Tragflugel des Kleinsten Induzierten Widerstandes**, *Zeitshrift fur Flugtechnik und Motorluftschiffahrt,* 25 Jg, 1933

[25] Göthert, B., **Einfluss von Flachenbelastung, Flugelstreckung und Spanweitenbelastung auf die Flugleistungen**, *Luftfahrtforschung*, Vol. 6, 1939

[26] Torenbeek, E., ***Fundamentals of Conceptual Design Optimization of Subsonic Transport Aircraft***, Delft University of Technology Report LR-292, Delft, the Netherlands, 1980

[27] Raymer, D., **Vehicle Scaling Laws For Multidisciplinary Optimization,** AIAA Paper 01-0532, 39th AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 2001

[28] Anon, *Exhibit text*, Vasamuseet (Vasa Museum), Stockholm, Sweden

[29] Ball, W., ***A Short Account of the History of Mathematics*** (4th Edition, 1908), Transcribed by D. Wilkins, Trinity College, Dublin

[30] Anon., ***Encyclopedia of Mathematics***, Kluwer Academic Publishers, Dordrecht, the Netherlands, 2001

[31] Anon. (AIAA Technical Committee on Multidisciplinary Design Optimization), **Current State of the Art On Multidisciplinary Design Optimization (MDO)**, AIAA White Paper, ISBN 1-56347-021-7, September 1991

[32] Lewis, K., **Multidisciplinary Design Optimization**, *Aerospace America*, Dec. 2001

[33] Bartholomew, P., **The Role of MDO within Aerospace Design and Progress towards an MDO Capability**, AIAA Paper 98-4705 (Defence Evaluation and Research Agency, UK), 1998

[34] Cassidy, P., *Telecon*, Boeing Phantom Works, St. Louis, MO, 9 Oct. 2001

[35] Gallman, J., Kaul, R., Chandrasekharan, R., and Hinson, .M., **Optimization of an Advanced Business Jet**, *AIAA Journal of Aircraft,* Vol 34 No. 3,May-June 1997

[36] Newman, J., Taylor, A., Barnwell, R., Newman, P., and Hou, G., **Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[37] Herbst, W., and Ross, H., **Application of Computer Aided Design Programs for the Technical Management of Complex Fighter Development Projects**, AIAA Paper 70-364, AIAA Fighter Aircraft Conference, St. Louis, Mo., 1970

[38] Schieck, F., Deligiannidis, N., and Gottmann, T., **A Flexible, Open-Structured Computer Based approach for Aircraft Conceptual Design Optimization**, AIAA Paper 2002-0593, 40th AIAA Aerospace Sciences Meeting, Reno, NV, Jan 2002

[39] Crawford, C., and Simm, S., **Conceptual Design and Optimisation of Modern Combat Aircraft**, RTO MP-35, RTO AVT Symposium on Aerodynamic Design and Optimization of Flight Vehicles in a Concurrent Multidisciplinary Environment, Ottawa, Canada, 1999

[40] Anon., ***Encyclopedia of Optimization***, Kluwer Academic Publishers, Dordrecht, the Netherlands, 2001

[41] Giunta, A., ***Aircraft Multidisciplinary Design Optimization using Design of Experiments Theory and Response Surface Modeling Methods***, Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, A, 1997

[42] Hajela, P., **Nongradient Methods in Multidisciplinary Design Optimization – Status and Potential**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[43] Mavris, D., and DeLaurentis, D., **A Probabilistic Approach for Examining Aircraft Concept Feasibility and Viability**, *Aircraft Design (International Journal)* vol 3-No. 2, Pergamon, June 2000

[44] Sevant, N., Bloor, M, and Wilson, M, **Aerodynamic Design of a Flying Wing Using Response Surface Methodology**, *AIAA Journal of Aircraft,* Vol 37 No. 4, July-Aug 2000

[45] Knill, D., Giunta, A., Baker, C., Grossmann, B., Mason, W., Haftka, R., and Watson, L., **Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design,** *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[46] Haftka, R., and Gurdal, Z., ***Elements of Structural Optimization***, Third Edition, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1992

[47] Pant, R., and Fielding, J., **Aircraft Configuration and Flight Profile Optimization using Simulated Annealing**, *Aircraft Design (International Journal)* vol 2-No. 4, Pergamon, Dec 1999

[48] Crossley, W., Martin, E., and Fanjoy, D., **A Multiobjective Investigation of 50-Seat Commuter Aircraft Using a Genetic Algorithm,** AIAA paper 2001-5247, Oct. 2001

[49] Raymer, D., and Crossley, W., **A Comparative Study of Genetic Algorithm and Orthogonal Steepest Descent for Aircraft Multidisciplinary Optimization,** AIAA Paper 2002-0514, 40th AIAA Aerospace Sciences Meeting, Reno, NV, Jan 2002

[50] Schwefel, H., ***Numerische Optimierung von Computer-Modellen***, Birkhauser, Basel, 1974

[51] Anon., *Evolution Strategy,* (http://www.bionik.tu-berlin.de), Technische Universität Berlin, 2002

[52] Holland, J., ***Adaption in Natural and Artificial Systems***, MIT Press, Cambridge, Massachusetts, 1975

[53] Blasi, L., Iuspa, L., and Core, G**., Conceptual Aircraft Design Based on a Multiconstraint Genetic Optimizer**, *AIAA Journal of Aircraft,* Vol 37 No. 1, Mar-April 2000

[54] Perez, R. E., Chung, J., and Behdinan, K., **Aircraft Conceptual Design Using Genetic Algorithms,** AIAA Paper 2000-4938, Sep. 2000

[55] Roth, G., and Crossley, W. , **Commercial Transport Aircraft Conceptual Design Using a Genetic Algorithm Based Approach,** AIAA Paper 98-4934, Sep. 1998

[56] Ryan, G., ***A Genetic Search Technique for Identification of Aircraft Departures***, NASA Contractor Report 4688, contract NAS 2-13445, August 1995

[57] Rogers, J., ***DeMAID/GA USER'S GUIDE***, NASA Technical Memorandum 110241, NASA-Langley, April 1996

[58] Davies, W., et al., ***Twins*** (film script), Universal Studios, Hollywood, CA, 1988

[59] Crossley, W., *Lecture Notes* - Multidisciplinary Optimization, Purdue University, 2001

[60] Williams, E. and Crossley, W. , **Empirically-Derived Population Size and Mutation Rate Guidelines for a Genetic Algorithm with Uniform Crossover**, *Soft Computing in Engineering Design and Manufacturing*, P. K. Chawdhry, R. Roy and R. K. Pant (editors), Springer-Verlag, 1998

[61] Mosher, T., **Conceptual Spacecraft Design Using a Genetic Algorithm Trade Selection Process**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[62] Goldberg, D., ***Genetic Algorithms in Search, Optimization, and Learning***, Addison Wesley Longman, Inc., Boston, Massachusetts, 1989

[63] Sobieszczanski-Sobieski, J., **Optimization by Decomposition: A Step from Hierrachic to Nonhierrachic Systems,** *Recent Advances in Multidisciplinary Analysis and Optimization*; NASA CP-3031; Part 1, 1988.

[64] Hollowell, S., Beeman, E., and Hiyama, R., ***Conceptual Design Optimization Study***, NASA Contractor Report 4298, Contract NAS1-18015, 1990

[65] Batill, S., Stelmack, M, and Yu, X**., Multidisciplinary Design Optimization of an Electric-powered Unmanned Air Vehicle**, *Aircraft Design (International Journal)* vol 2-No. 1, Pergamon, March 1999

[66] Isikveren, A., ***Methodology for Conceptual Design and Optimization of Transport Aircraft***, Licentiate Thesis, Report 98-8, KTH-Royal Institute of Technology, Stockholm, Sweden, 1998

[67] Koch, P., Simpson, T., Allen, J., and Mistree, F., **Statistical Approximations for Multidisciplinary Design Optimization: The Problem of Size**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

[68] Mavris, D., Bandte, O., and DeLaurentis, D., **Robust Design Simulation: A Probabilistic Approach to Multidisciplinary Design**, *AIAA Journal of Aircraft,* Vol 36 No. 1, Jan-Feb 1999

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

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

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

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

[73] Raymer, D., *Genetic Algorithm Optimization Scheme,* Unpublished draft, 11-7-01

[74] Robinson, D., *E-mail Communication*, Boeing/Rockwell/North American Aviation (retired), 9-12-2000

[75] Hoerner, S., ***Fluid Dynamic Drag***, Hoerner Fluid Dynamics, Brick Town, N.J., 1965

[76] Raymer, D., ***Dynamic Lift Airship Design Study*** (contract report for Ohio Airships Inc.), Conceptual Research Corporation Report CRC-R-01-9, Sylmar, CA, 2001

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

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

[79] Cadell, W., **On the Use of Aircraft Density in Preliminary Design**, SAWE Paper 813, May 1969

[80] O'Brimski, F., **The Maximum Practical Density for Tactical Aircraft**, Internal Technical Note, NAVAIR AIR-5223, Washington, DC., 1993

[81] Taylor, J., ***Jane's All the World Aircraft***, Jane's, London, UK, (1976, 1992, 2000).

[82] Raymer,D., and Norris, M., **Next Generation Fighters: Roles & Concepts**, AIAA Aerospace Engineering Conference, Los Angeles, 1990

[83] Raymer, D., **Next-Generation Attack Fighter Conceptul Design Study**, *Aircraft Design,* Vol. 1 No. 1, Pergamon Press, March 1998

[84] Raymer, D., ***Next Generation Attack Fighter: Design Tradeoffs and Notional System Concepts*****,** The RAND Corporation, MR-595-AF, Santa Monica, CA, 1996

[85] Raymer, D., and McCrea, M**., Design of Jet STOVL Aircraft using RDS-Professional**, International Powered Lift Conference, Arlington, Virginia, Oct. 2000

[86] Raymer, D., ***Tactical UAV's: Design Requirements Tradeoffs & Notional System Concepts,*** RAND Corp., DRR-1763-AF, 1997

[87] Wang, X., and Damodaran, M**., Comparison of Deterministic and Stochastic Optimization Algorithms for Generic Wing Design Problems**, *AIAA Journal of Aircraft,* Vol 37 No. 5, Sept-Oct 2000

[88] Staton, R., ***Statistical Weight Estimation Methods for Fighter/Attack Aircraft***, Vought Aircraft Rept. 2-59320/8R-50475, 1968

[89] Staton, R., ***Cargo/Transport Statistical Weight Estimation Equations***, Vought Aircraft Rept. 2-59320/9R-50549, 1969

[90] Jackson, A**., Preliminary Design Weight Estimation Program**, AeroCommander Div., Rept. 511-009, 1971

[91] Hess, R., and Romanoff, H., ***Aircraft Airframe Cost Estimating Relationships***, Rand Corp., Rept. R-3255-AF, Santa Monica, CA, 1987.

[92] Raymer, D., **Developing an Aircraft Configuration Using a Minicomput**er, *Astronautics and Aeronautics (now Aerospace America),* Nov. 1979