MULTIDISCIPLINARY OPTIMIZATION APPLICATIONS IN PRELIMINARY DESIGN
— STATUS AND DIRECTIONS

Ilan Kroo
Dept. of Aeronautics and Astronautics
Stanford University, Stanford, California

Abstract
Multidisciplinary design optimization (MDO) has played an important role in aircraft preliminary design for 30 years, yet it is far from a mature field. This paper discusses the increasingly widespread use of MDO for aircraft design, describing the evolution of computational tools and strategies, and summarizing some current research directions. The objective of this review is not to provide a comprehensive survey of MDO methods and applications, but rather to highlight some interesting aspects that suggest how this field is developing.

Introduction
When an aircraft designer hears that a new program will use multidisciplinary optimization, the reaction is often less than enthusiastic. Over the past 30 years aircraft optimization at the conceptual and preliminary design levels has often yielded results that were either not believable, or might have been obtained more simply using methods familiar to the engineers. Even 5-20 years ago, actual industry application of numerical optimization for aircraft preliminary design was not widespread. In 1982, Prof. Holt Ashley delivered an AIAA Wright Brothers Lecture entitled, "On Making Things the Best—Aeronautical Uses of Optimization."[1] For this lecture, he surveyed the relevant literature and found 4550 papers on optimal control, 2142 on aerodynamic optimization, 1381 on structural optimization. A total of 8073 papers, along with surveys, texts, etc. But Ashley had a hard time finding a single case where this formal procedure was employed by industry. In his paper he cites the results of an informal survey he conducted on the uses of optimization. Typical responses included:

- From an aeronautical engineer, experienced in civil and aeronautical structures, "One of the reasons that I stopped work in optimization was my dismay ... that there were so very few applications."

- From a Dean of Engineering who has known the field for over a quarter century: "I do not recollect any applications."

- From a foremost specialist on synthesis with aeroelastic constraints, 'I am sorry, but I don't really have any..."

- From a recently-retired senior design engineer, describing events at his aerospace company, "For fifteen years I beat my head against a stone wall ...

The end was: formal optimization techniques were never used in aircraft design (even to this day!). The company was forced to use them in its subsequent ICBM and space programs."

There have been several reasons for industry's hesitation in adopting formal optimization methods, especially MDO methods in preliminary design. One of the reasons has been that the fidelity of analysis methods appropriate for use with optimization was so poor that such studies rarely produced credible results. This was perhaps less true in the area of structural design where significant progress was possible using complex, but linear, analysis. Linear inviscid aerodynamic analyses were combined with optimization in the 1970's, but since drag, buffet, and maximum lift could not be predicted using these codes, integrated multidisciplinary synthesis programs such as FLOPS[2] and ACSYNT[3] continued to rely on more empirical 'handbook' methods. Since that time computational capabilities have increased by a factor of 10 to 20 every 5 years. Since efficient optimization requires on the order of $n^2$ analyses, a 100-fold increase in computational capability lead to an order of magnitude increase in the number of design variables that can be considered in an optimization problem. Thus, while early synthesis programs could deal with 5-10 variables, little more than could be managed by hand, the same programs could handle 50-100 parameters with more modern computers. Improvements in optimization algorithms, themselves permit several MDO programs for preliminary design to incorporate hundreds of design variables. Problems with thousands or even tens of thousands of nonlinear design variables have been solved in special cases involving sparse or nearly linear systems. This convergence between computational capability and computational requirements for interesting design problems is one of the reasons that MDO is no considered by many to be such a promising technology, despite the limited acceptance of pioneering MDO efforts.

This increase in computational capability has made possible advances in disciplinary optimization and detailed design, but the application of MDO in preliminary design is perhaps most significant. This is because preliminary design is where multidisciplinary trades matter most, where engineers understand the importance of interdisciplinary interactions, and where MDO is most likely to provide the most clear, near-term benefits. In a very real sense preliminary design is MDO. Preliminary design activities include multidisciplinary interactions and are aimed at deter-
mining a design that satisfied constraints and maximizes some figure(s) of merit. MDO is a formalization of the preliminary design process, enforcing rational trade-offs rather than ad-hoc or historically-mandated priorities. The MDO process encourages careful and explicit problem formulation, and although this sometimes constitutes a barrier to MDO application, it can reduce the likelihood of costly redesign later in the product development cycle.

Examples of MDO Applications in Preliminary Design
The applications of MDO to aerospace preliminary design in 1997 are very different from what they were when Prof. Ashley surveyed the field in 1982. The Sixth Symposium on Multidisciplinary Analysis and Design in 1996 [4] included over 200 papers including many industry papers on applications to preliminary design. Hundreds of significant multidisciplinary problems in automotive, marine, aeronautical, aerospace, and electronics industries were described and no attempt will be made here to provide a comprehensive survey. Instead, this section will briefly describe a few MDO applications in preliminary design with which the author has some familiarity, to highlight the growing number and utility of such efforts.

The first amusing example is illustrated in figure 1 which summarizes the evolution of commercial transport design over 30+ years. The 707 and A340 designs are not drawn at quite the same scale, but the author was surprised at the extent of the similarity that becomes apparent when the two designs are drawn with the same length and overlaid. The product of Mach number and maximum lift-to-drag ratio (M L/D), plotted in the figure (data from [5]) seems to confirm the lack of any progress in aerodynamic design. The figure is, however, misleading. The fuel consumption of aircraft over this period dropped by a factor of 2 and this was by no means entirely attributable to improved engine performance. In fact figure 1 is an ideal example of why MDO is so important. The surprisingly constant value of M L/D is not a reflection of stagnation in aerodynamic design, but rather a basic result of the complex physics of the aircraft system and an indication that the major aircraft companies do a good job of multidisciplinary design. If aerodynamicists strove solely to increase M L/D, the plot would look quite different. Improvements in design methodology and understanding of transonic flow phenomena could have produced gains in M L/D of at least 15% to 20%. Instead, advances in cruise aerodynamics are exploited in parameters that might be associated with another discipline. Supercritical airfoils may be used to increase cruise Mach number with little drag penalty. But as engine bypass ratios have increased, the benefits of this higher Mach number diminish. Thus, it makes more sense to maintain, or even lower, the Mach number and take advantage of the improved wing design to increase wing thickness and reduce structural weight, or reduce sweep, thereby requiring a simpler high-lift system and reducing the wing cost.

These kind of basic trades made in the conceptual and early preliminary design phases of a new aircraft involve fundamental compromises between aerodynamics, structures, and propulsion, with the primary interactions involving 10's of parameters. Even this part of the design process can be enhanced with MDO tools, but the most significant impact of modern MDO is to be found in more complex design problems.

MDO applications in the preliminary design of commercial and military aircraft have a long history, with computer-based optimization described in the 1960's. [6,7]. Even at that time, designers envisioned modal analyses, controlled by an executive program that managed optimization and sensitivity analysis. (Fig. 2)

Following the General Dynamics SYNAC program of 1966 [6], a number of conceptual design programs have evolved over the past 30 years. From the initial work on ACSYNT [3] to an ambitious integrated preliminary aircraft design system (IPAD)[8], to codes such as FLOPS [2], and WingMOD[9], these programs have steadily improved in usability and analysis fidelity. Figure 3 (from [9]) illustrates some of the considerations that are included in WingMOD, McDonnell-Douglas' Wing Multidisciplinary Optimization Design program. Analysis of wing (and tail) structures includes buckling, skin and substructure design, practical minimum gauge constraints, along with statistical as-built weight models. Nonplanar, multiple surface aerodynamics are based on linear solution methods, modified based on wind tunnel, flight, and CFD data to account for transonic and viscous phenomena. The system includes mission analysis and trajectory computations along with static aerelastic interactions. These modules interact with each other through a database management, optimization, and interface known as GenIE. Using a sequential quadratic programming (SQP) method and hundreds of design variables, the system has been applied to the design of conventional wing-tail combinations, winglets, composite wings, and unconventional aircraft [9-10]. Figure 4 illustrates some of the critical aerodynamic and structural constraints identified by WingMOD in the design of a composite wing for an MD-90 series of aircraft. Several flight conditions are associated with critical constraints at different locations on the wing and tail. The flexibility of this MDO process is evident as such constraints can be easily added or modified.

MDO has been employed in a number of recent unpuiloted air vehicle (UAV) programs. In the early stages of the Tier II+ program a number of MDO studies were undertaken to determine a government baseline design and establish ranges of expected performance [11]. Hundreds of mission scenarios, engine options, and payload combina-
tions were optimized using an analysis program that included a graphical user interface, SQP optimizer, and database management. Analyses included transonic laminar flow wing aerodynamics, routines for weight estimation of high aspect ratio composite wings, stability and trim computations, a mission and trajectory analysis routine, and six complete engine cycle decks, provided by several engine manufacturers. Typical cases involved optimization of twenty major design variables, including vehicle and trajectory-related characteristics to minimize the overall cost of fleet operation. A multiple window user interface permitted interaction with the database and optimizer, along with graphical representations of design results (Fig. 5). More recent UAV programs have included MDO-related design studies in a team-based, distributed design environment. Using standard web-browsing software, teams could develop and publish analysis models in their particular discipline and invoke remote models from other groups as part of their on analysis. Figure 6 shows the summary pages of several groups. Take-off performance computations, for example, use engine analyses developed, published, and documented by another team.

A more extensive application of multidisciplinary design methods for aircraft preliminary design is exemplified by recent work on advanced concepts for commercial transport aircraft at Boeing and McDonnell-Douglas [5]. The McDonnell-Douglas Blended Wing Body Concept (BWB) is shown in Figure 7, a rather unconventional solution to the large subsonic transport problem. The BWB is an unstable, tailless aircraft that houses passengers inside the center wing section and attempts to exploit boundary layer ingestion for improved fuel economy. Aerodynamics, structures, propulsion, stability, and control, are all tightly coupled in this design, making multidisciplinary analysis and design especially important. Furthermore, because it is unconventional, rather high-fidelity analyses are required for design, since established, empirical databases and statistical methods do not exist for this class of aircraft. Many aspects of this design are being developed using MDO. One of these is the cabin geometry and layout of passengers and accommodations. Although this task is generally accomplished by an experienced configurator using CAD, a study of the cabin design as a more formal optimization problem was undertaken. The problem was to determine the number of passenger and cargo bays (Figure 8), their length, and their location in the wing. Using a simplified structural model of the pressurized inner wing section, possible layouts were examined using an evolutionary algorithm, well-suited to this mixed continuous and discrete problem with many local minima. Results from this cabin design investigation were then incorporated into a planform optimization study that included more complete mission analyses and several primary wing design variables. Using the NPSOL optimizer [12] to minimize a measure of direct operating cost, several design were generated (with different span and cabin constraints) and used to help refine the baseline configuration.

**Challenges**

These limited examples of MDO use in aircraft preliminary design demonstrate that such methods have now become valuable tools for design; but many difficulties remain in the routine application of MDO. Users and developers of design frameworks are faced with several challenges.

**Increasing fidelity of disciplinary models**

It is sometimes said that multidisciplinary analysis, like a chain, is only as good as its weakest link. While this is not entirely accurate (as very approximate models of items with low sensitivities are adequate), there is often a need for higher fidelity analysis. One may build MDO systems with very impressive interfaces and sophisticated database management capabilities, but if the analyses are not well developed, the system may useless, or worse. In aircraft preliminary design, critically important considerations are often quite complex. For example, although aft-swept wings usually exhibit higher root stresses at high CL, for a given load factor, one of the critical loads is for the DC-8 wing is at low CL due to increased tail trim loads. Assumptions that are fine for one case (e.g., elliptic loading minimizes vortex drag) may not apply in another (e.g., two interfering wings, each with elliptic loading, may be far from optimally loaded). Particularly because optimization is known to exploit such simplifications, the analyses for MDO must be especially complete and robust. Improving the fidelity of disciplinary analyses is not straightforward, however. In addition to the likely increase in computation time, one must generally deal with an increasingly complex set of design parameters and results.

**Managing complexity**

As individual analysis accuracy is improved, it brings added complexity to the code itself. Furthermore, the number of analysis modules may increase as the system becomes more capable, making the process of integration more difficult. The complexity and dimensionality of the design parameterization becomes similarly more problematic. While a simple system, based on beam representations of structures, may be easy to manage, the number of design degrees of freedom and dimensionality of the system increase greatly when this model is replaced with a plate or finite element model.

**Communication**

As the complexity of the design task increases, communication between analyses, and between people involved in the design process, becomes more difficult. Large-scale projects often involve many individuals in different locations who must make different computer programs under-
stand the complex results from other programs and teams. Although much work is underway to reduce this problem, it remains a difficult one.

Optimization
Finally, the optimization process, itself, faces many challenges in the context of practical application to preliminary design. By their nature, conceptual and preliminary design often deal with discrete choices and topologically distinct configuration options. Even dealing with a large, but well-defined set of integer parameters poses difficulties. When the system involves legacy codes with poorly documented internal iteration and floating point branch tests, the situation becomes critical. Even well-written programs sometimes yield non-smooth results due to the discrete nature of modeling assumptions as shown in Figure 9 [13]. This has prompted renewed looks at non-gradient-based optimization including evolutionary algorithms [14] and the use of response surfaces to provide smooth design spaces. [15]

A great deal of research in the field of MDO has been undertaken to address these challenges. The following sections describe some of the many approaches and strategies that have been proposed.

Approaches and Strategies: Generations of MDO Solutions

One may divide the numerous approaches proposed for preliminary design into three generations of MDO strategies. The first generation involves integrating analysis into a monolithic, black box and combining it with optimization. In second generation architectures, analyses are distributed among multiple computers or engineering teams. Such systems employ a variety of tools for analysis management, helping to address some of the complexity problems, that plague larger integrated programs. Third generation systems decompose and distribute design tasks as well as analysis tasks. The structure of these architectures is outlined in Figure 10.

Generation 1: Integrated Analysis with Optimization
The SYNAC program, mentioned previously, is one of the earliest examples of first generation MDO systems. Since that time, projects such as ACSYNT, IPAD, OPTDOT, and FLOPS, along with a large number of conceptual design research codes [16] have demonstrated the utility of tightly integrated analysis routines called by a numerical optimizer. Such systems can often be assembled by a small team without formal inter-program communication protocols. This architecture is adequate for relatively small-scale projects, but quickly becomes problematic as the scope of the system increases. Large, monolithic MDO systems can be difficult to understand, manage, and extend.

New approaches to software development and management may permit larger, efficient systems, still based on the concept of tightly-coupled analyses. The use of object-oriented software and component architectures can dramatically reduce the complexity of such systems. In the example of Figure 11, a Java™-based textbook on aircraft preliminary design includes over 70 Java classes, providing disciplinary analyses, optimization, parametric trades, and drawing routines. [17] These components and the associated shared data are integrated through high-level database management classes. Students can access the text from an internet server with all computations performed on the client machines.

Generation 2: Distributed Analysis
Second generation MDO systems include modular, distributed analyses and often focus on interdisciplinary communication issues. These systems involve more than just parallel computation, but attempt to address some of the analysis complexity issues with a number of analysis management strategies. Examples of such systems include the Boeing Access Manager [18] with which an engineer may graphically assemble a set of analysis programs and create project-level procedures for execution. The system manages the inputs and outputs, substantially reducing the time required for code integration and permitting computation over a large network. NASA Langley's FIDO framework allows a user to script execution of programs over a network and provides a variety of process and data monitoring tools [19]. General Electric's Engineous system, and the more recent version, iSIGHT [20], developed by G.E. and Engineous Software, Inc. constitute well-developed, commercially-available examples of second-generation MDO systems.

In addition to scripted, or code-based analysis management, programs such as Rockwell's Design Sheet [21] and versions of GenIE [22] attempt to simplify analysis integration with dynamic, automated path generation, maintaining database consistency and inferring connections between analyses, based on knowledge of each routine's inputs and outputs. Design Sheet employs constraint propagation, while GenIE uses a quasi-procedural methods to assemble program logic. A recent version of GenIE is shown in Figure 12, where an aircraft design problem is decomposed into three parts, each accompanied by its own local copy of the quasi-procedural executive and its own local database. The groups communicate through a PVM network with a main program that includes another set of executive routines, optimizers, and user interface.

These second-generation systems have been applied to a large number of preliminary design problems over the past decade on currently constitute the most reliable strategies for modern MDO.
Complementing these systems are scheduling, planning, and decomposition tools such as DeMaid [23] and AGEN-DA [24] that assist the developer with problem formulation and implementation.

**Generation 3: Distributed Design**

The decomposition and distribution of analysis tasks among a group of computers and individuals can be carried a step further by decomposing the design problem and distributing design responsibility among several disciplines (or teams, or computers). This basic idea is appealing, as it provides disciplinary teams with design challenges, rather than relegating them to subordinate analysis tasks. However, just how this is accomplished is not obvious, and a number of schemes have been proposed.

Distributed design is currently undertaken in an often ad-hoc or informal way, with iterative or sequential design performed separately in each discipline. Except in the case of problems with special hierarchical character, such approaches offer no guarantees of convergence, or worse, convergence to incorrect solutions. Such criticism is not merely a technical quibble, as illustrated by the simple two-discipline wing design problem of figure 13. Here a high-level system optimizer varies the wing span in order to minimize a figure of merit consisting of a linear combination of wing structural weight and drag. In this sequential process (similar to some proposed MDO procedures), the aerodynamics group minimizes drag for a given span and the structures group uses the computed load distribution, minimizing weight while satisfying stress constraints. The result of this process is a wing with a smaller span and higher drag than the true optimum, as the system does not properly exploit interdisciplinary opportunities for drag reduction. (i.e. With a small increment in drag at a given span, the aerodynamics group could shift the loading board, reducing structural weight and allowing the system to increase the span. This reduces drag by more than the original drag increment as noticed by Prandtl.)

More carefully-constructed systems for distributed design optimization have been studied for some time and are classified as multi-level optimization methods. Several approaches to multi-level optimization have been proposed. (See [25] - [28].) At least to of these have been developed in some detail and applied to preliminary design problems.

Concurrent subspace optimization (CSSO) divides the design problem into several discipline-related subspaces, each of which shares some responsibility for satisfying constraints imposed on the system while trying to reduce a global objective (or objectives). A system level algorithm coordinates this process in different ways, depending on the implementation. Several versions of CSSO have been developed in recent years by researchers at NASA Langley, and Notre Dame (29). A commercial framework based on this concept (SysOpt) has been developed and is described in a number of papers [30].

A second approach to distributed design is termed collaborative optimization and also involves concurrent optimization of subproblems. In this case, disciplinary teams are charged with satisfying local constraints while trying to match target values specified by a system coordinator (Figure 14). The system adjusts these shared target values to minimize some objective while permitting the subspaces to match the targets. One of these system-level targets might be wing structural weight, which would be regarded by the structures group as a weight budget, while other groups regard it as the current estimate for wing weight. Each of the groups then tries their best to achieve the targets, but ensures that local constraints (such as buckling or stress constraints) are satisfied. The system can use information implicit in the subspace solutions (post-optimality gradient information) to improve the efficiency of the process.

This approach, and some related variations, have been applied to test problems by researchers at Stanford [31], NASA Langley [32], Brigham-Young University [26], Notre Dame, Lockheed-Martin, and Engenous Software. These test problems range from simple analytic functions in 2 variables to single-stage-to-orbit vehicle design and trajectory optimization in as many as 1000 variables. Aircraft sizing and aerelastic wing design have also been accomplished using this approach [33].

Although still in development, collaborative optimization is attractive in many respects. It greatly simplifies analysis integration and communication; it permits domain-specific selection of optimization algorithms (e.g. adjoint method for aero design); and it is especially well-suited for analyses that are already coupled with optimization, such as several structures and trajectory codes. Current implementations of collaborative optimization still have some difficulties including slow system-level convergence and sensitivity to subspace feasibility tolerances, but the strategy appears especially promising for large problems with low dimensionality interdisciplinary coupling.

**Approximate Models**

Another area of research with important implications for distributed design is the application of approximate models in high-fidelity MDO. Many recent studies have focussed on response surface generation, design of experiments, design of automated computer experiments, and neural network applications to MDO. Such approximation concepts may be used in many ways, as depicted in figure 15. In the left portion of the figure, response surface models of disciplinary analyses replace the more time-consuming and less smooth analyses for the entire multi-level optimization problem. A more interesting application is shown in the right portion of the figure, where RSM's are
used to model results of the subspace design problems. This approach may exploit extra information, available from the subproblems, and fits naturally into the collaborative optimization framework, restricting the model to target variables and helping to alleviate the 'curse of dimensionality'. This structure eliminates problems with slow system-level convergence and provides smooth subspace results to the system-level optimizer.

The nature of aerodynamic and structural characteristics for aircraft near optimal solutions is often very complex and nonlinear. It is likely that simple approximate models such as quadratic response surfaces can be regarded as acceptable approximations only in a narrowly restricted region of the design space. Thus, careful consideration of model updating schemes is extremely important in these applications. Recent research by Dennis and Alexandrov [43], along with others at NASA Langley and Boeing provide some guidance in approximate model refinement. Initial results using this type of approach in the context of multi-level design (Figure 16) have been very encouraging.

Conclusions

The development and application of MDO in preliminary design is far from a mature field. In addition to research on multidisciplinary design strategies in government laboratories and universities, several commercial and industrial frameworks for distributed analysis and design are under development. The include new versions of ISIGHT, the DARPA RaDEO and ASOP programs, new versions of GenIE, and many others.

These developments provide some responses to the challenges of improved fidelity and complexity management. Eventually, this process may blur some of the distinctions between conceptual, preliminary, and detailed design, enabling better decisions early in the design cycle and reducing time required for product development. Still, preliminary design is where interest in MDO began and where it is likely to remain most useful.

Larger scale MDO and collaborative design methods are still being developed, but current results are encouraging and promise to make this an exciting field in the future.

References


Figure 1. Transport Aircraft Evolution

Figure 2. Early View of Aircraft MDO System

Figure 3. WingMOD Design Conditions

Figure 4. Constraints from WingMOD

Figure 5. HALE MDO Interface

Figure 6. Distributed UAV Analysis
Cabin Layout Optimization

Variables:
- Total number of spanwise bays \(5 \leq \text{Nbays} \leq 12\)
- Single or double deck
- Passenger or cargo bay
- Length of bay

Figure 7. BWB Concept

![Blended Wing Body](image)

Figure 8. BWB Cabin Layout

![Cabin Layout Diagram](image)

Figure 9. Non-smooth Results from Aero Analysis

![Graph showing variations of leading-edge suction and its sensitivity](image)

Variations of the leading-edge suction and its sensitivity with respect to \(X^{lB}\) vs. sweep angle for the HSCT planform, calculated on a 26\(\times\)20 mesh.

Figure 10. Generations of MDO Architectures

![Diagram of MDO generations](image)

Figure 11. Java\textsuperscript{TM}-Based Aircraft Design Text

![Java-based design interface](image)

Figure 12. Distributed GenIE Architecture

![Diagram of distributed architecture](image)
Figure 13. Sequential Optimization of a Wing

Figure 14. Collaborative Optimization

Figure 15. RSM's in Multilevel Optimization

Figure 16. RSM Update in Collaborative Design