A COARSE GRAINED PARALLEL VARIABLE-COMPLEXITY MULTIDISCIPLINARY OPTIMIZATION PARADIGM

Susan Burgee

DEPARTMENT OF COMPUTER SCIENCE
VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY
BLACKSBURG, VA 24061-0106

Anthony A. Giunta, Vladimir Balabanov, Bernard Grossman, William H. Mason, Robert Narducci

DEPARTMENT OF AEROSPACE AND OCEAN ENGINEERING
VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY
BLACKSBURG, VA 24061-0203

Raphael T. Haftka

DEPARTMENT OF AEROSPACE ENGINEERING, MECHANICS,
AND ENGINEERING SCIENCE
UNIVERSITY OF FLORIDA
GAINESVILLE, FL 32611-6250

Layne T. Watson

DEPARTMENTS OF COMPUTER SCIENCE AND MATHEMATICS
VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY
BLACKSBURG, VA 24061-0106

Summary

Modern aerospace vehicle design requires the interaction of multiple disciplines, traditionally processed in a sequential order. Multidisciplinary optimization (MDO), a formal methodology for the integration of these disciplines, is evolving toward methods capable of replacing the traditional sequential methodology of aerospace vehicle design by concurrent algorithms, with both an overall gain in product performance and a decrease in design time. A parallel MDO paradigm using variable-complexity modeling and multipoint response surface approximations is presented here for the particular instance of the design of a high speed civil transport (HSCT). This paradigm interleaves the disciplines at one level of complexity, and processes them hierarchically at another level of complexity, achieving parallelism within disciplines, rather than across disciplines. A master-slave paradigm manages a coarse grained parallelism of the analysis and optimization codes required by the disciplines showing reasonable speedups and efficiencies on an Intel Paragon.

Introduction

In the design of modern aerospace vehicles there is a need for the integration of multiple disciplines, such as aerodynamics, structures, control systems, propulsion, etc. Design of aerospace vehicles has traditionally been sequential, through the interaction of specialists from many different disciplines. This sequential interdisciplinary collaboration is barely acceptable for simple designs, but as the size and complexity of the aerospace vehicle increases, the collaboration between disciplines becomes more difficult, resulting in a longer product design-cycle and higher design cost. In order to remain competitive in the aerospace vehicle design marketplace, designers must reduce this product development cycle time and design cost. Multidisciplinary optimization (MDO) is a formal methodology for the integration of these disciplines. Each discipline involves many complex analyses over a large design space, and therefore the codes needed for each discipline are usually large and complex. These codes must communicate data in an iterative process, creating a large computational burden. The goal of MDO technology is to use numerical optimization, with the aid of high performance computing, as a means of coupling these disciplines to reduce the computational time and design cost. MDO technology will provide aerospace vehicle designers with a variety of new tools and algorithms to aid in the design of modern aerospace vehicles.

Current research in the field of MDO involves the investigation of methods which will replace the sequential methodology of aerospace vehicle design, without an increase in complexity yet with an overall increase in performance. A nonsequential approach requires the tight coupling of two or more disciplines in an optimization environment. In order to attain the optimal design, these tightly coupled disciplines must iterate until they converge on the optimal design. While iterating, a large amount of computation within disciplines and communication between disciplines is required. Aerospace vehicle designers are researching methods to reduce the real computation time resulting from the tight coupling of two or more disciplines. These methods include both hierarchical and nonhierarchical strategies.

Cramer et al. (1994) have developed a hierarchical strategy in which aerodynamic and structural disciplines are tightly coupled. Two methods of reducing the huge computation time created by the tight coupling have been developed: Multidisciplinary Feasibility (MDF) and Individual Discipline Feasibility (IDF) (Cramer et al., 1994). Both methods use an optimizer to control the necessary computation and communication between disciplines. This optimizer provides control at a higher level, reducing the amount of communication needed between disciplines. Since the MDF method requires complete multidisciplinary feasibility after each iteration of the optimization, it is typically much more costly than IDF, which only requires individual discipline feasibility after each iteration. IDF also allows for easy substitution of one analysis code for another, which is often necessary in the design of an aerospace vehicle.

There has also been research on nonhierarchical MDO strategies by Sobieski et al. (1988). A variation on Sobieski's nonhierarchical strategy is provided by the work of Renaud et al. (1991). Nonhierarchical strategies would seem to work well in an MDO environment, given the uni-level interactions between separate disciplines. However, these nonhierarchical strategies do not work well in general (Shankar, 1992).

Kroo et al. (1995) have developed a MDO strategy known as "collaborative optimization," which involves decomposition of the aerospace vehicle design problem into subproblems. Each of these subproblems correspond to an aerospace discipline (e. g., aerodynamics, structures, and performance) and consists of two parts: analysis and optimization. A system-level optimizer is used to pass parameters to the different subproblems. These parameters are adjusted by the system-level optimizer to maximize the objective function while satisfying the system-level constraints. Theoretical convergence of this approach remains an open question.

This paper introduces a MDO paradigm, which is not a "pure" MDO strategy, but does accomplish many of the same goals as traditional MDO strategies. A hierarchical strategy is employed— interleaving structural optimization into the aerodynamic design —to generate estimates of structural weight for a configuration optimization. The structural optimization is coupled with algebraic weight equations to eliminate the need for performing structural optimization for every aerodynamic analysis. This paradigm is presented for the particular instance of the High Speed Civil Transport (HSCT) design and two disciplines (aerodynamics and structures), but is intended to be applicable to general MDO problems.

The nonsequential MDO paradigm presented here uses the concept of multipoint approximation, response surface methodology, and a parallel variable-complexity modeling strategy to achieve optimization. The parallel variable-complexity modeling involves a three-levels-of-complexity strategy. At the lowest level, the least expensive models are utilized to attain a reasonable initial design. These inexpensive models are then used to determine approximation domains for the more complex models. Finally, the more complex models are used in the construction of a parallel multipoint variable-complexity approximation.

Multipoint approximations and response surface methodology are required to deal with noisy and nonsmooth functions, which are unavoidable. Numerical noise results from various discretizations, finite dimensional approximations, and floating point arithmetic. Mathematically nonsmooth functions naturally occur as, e. g., constraint optima as a function of design parameters. Smooth response surfaces provide a computationally tractable mechanism for handling the nonsmooth components of the constraints and objective function.

Multipoint approximation does not present problems when dealing with a relatively small number of design variables. A large number of design variables, however, requires a large number of analyses, even for a low order approximation. Therefore simple analysis procedures are used to limit the approximation domain in the design space. Once the approximation domain is determined, points are selected within the domain where a more refined (second level) analysis is performed. The execution of the refined analyses at these selected points is done in parallel, a very coarse grained parallelism. As the optimization progresses, some of the analyses are done at the third level of complexity, using, e. g., Navier-Stokes or Euler equations instead

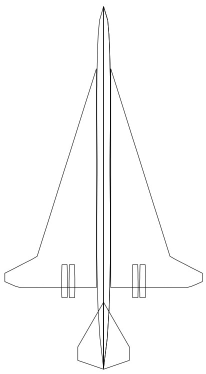

Fig. 1 Typical HSCT configuration.

of panel methods for the fluid flow, and more refined finite element meshes for the structural mechanics calculations. Details of computational requirements for high-fidelity analyses involving Navier-Stokes solutions and refined finite-element methods are addressed by Byun and Guruswamy (1994).

A coarse grained parallelization of both the aerodynamic and structural codes was implemented on an Intel Paragon to reduce the time required to evaluate the numerous points needed for the multipoint approximation. Coarse grained parallelization was chosen over fine grained parallelization because it naturally allows for the execution of the same analysis program many times in parallel, which fits well in a multipoint approximation environment. Results from the coarse grained parallelization of the aerodynamic code have shown reasonable speedup and efficiency, although further improvements will require a combination of fine and coarse grained parallelism.

In Section 1, the HSCT mission and the optimization problem are described. In Section 2, the variable-complexity modeling paradigm is explained. Section 3 describes multipoint approximations, the approximation domain, polynomial modeling, and regression analysis. Strategies for selecting points within the approximation domain are explored in Section 4, followed by an analysis of these strategies in Section 5. Section 6 provides a detailed high level description of the MDO paradigm. Section 7 details both the coarse grained parallelization of the individual steps of the MDO loop and the linking together of the steps into a single parallel program. Section 8 provides an analysis of the results as well as concluding remarks.

1 HSCT PROBLEM DESCRIPTION

1.1 HSCT CONFIGURATION

Figure 1 shows a typical HSCT baseline geometric configuration, a starting point for design optimization. Successful aircraft configuration optimization requires a simple yet meaningful mathematical characterization of the geometry. A model that completely defines the wing-body-nacelle configuration has been developed, using twenty-eight design variables. The wing planform is described by eight design variables. The leading and trailing edges of the wing are defined by the root chord, the two planform breakpoints, the axial location of the wing tip, the tip chord, and the wing semi-span. Figure 2 illustrates the definition of a wing planform using these eight variables.

The airfoil thickness is defined by five variables, which are the leading-edge radius size parameter, r_t, the thickness-to-chord ratio, t/c, the chordwise location of maximum thickness, m, and the trailing-edge half angle, $\tau_{\rm te}$. These parameters are illustrated in Figure 2. The nacelles move axially with the trailing-edge of the wing, and two parameters define their spanwise locations. Figure 2 illustrates the nacelle location parameters (x_9 and x_{10}).

The axisymmetric fuselage requires eight parameters. The axial location and radius of each of four *restraint* locations along the fuselage length are the design variables. Table A1 of Appendix A lists the 28 design variables, including their baseline values. While the configuration is defined using this set of parameters, the aircraft geometry is actually stored as a discrete numerical description in the Craidon format (Craidon, 1970).

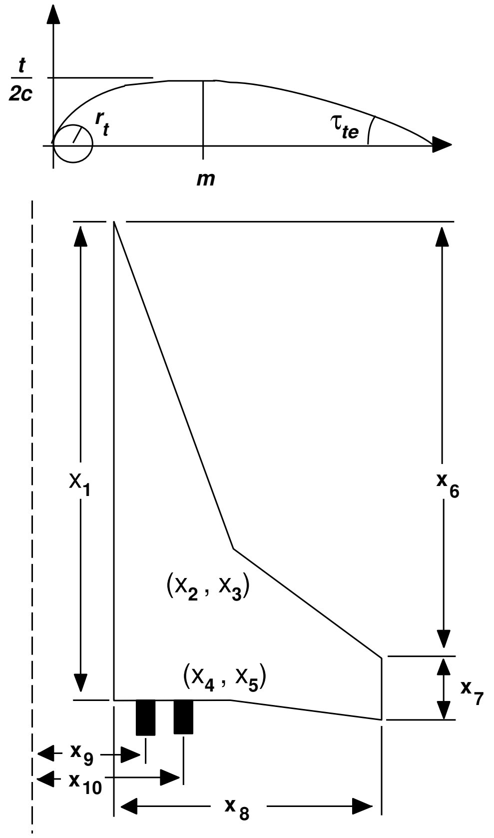

Fig. 2 Airfoil thickness distribution parameters (top) and wing planform and nacelle placement parameters (bottom).

1.2 HSCT MISSION

The optimization problem is to minimize the take-off gross weight of an HSCT configuration with a range of 5500 nautical miles and a cruise speed of Mach 2.4 while transporting 251 passengers. For this mission, in addition to the geometric parameters mentioned above, three variables define the idealized cruise mission. One variable is the mission fuel and the other two are the initial cruise altitude and the constant climb rate used in the range calculation. Sixty-six constraints which include performance/aerodynamic constraints and geometric constraints, such as fuselage volume and tail scrape angle, prevent the optimizer from creating physically impossible designs (Hutchison, 1993). These constraints are listed in Table A2 of Appendix A.

1.3 FOUR DESIGN VARIABLE PROBLEM

The HSCT wing planform is parameterized using eight design variables x_1–x_8 as shown in Figure 2. In addition, three variables describe the thickness-to-chord (t/c) ratios at the root chord, leading-edge break, and the tip chord; one variable is used to specify leading-edge radius; one variable defines the location of maximum t/c for all airfoil sections.

To develop and test the variable-complexity response surface optimization strategy it suffices to construct an example problem involving only a few design variables, say four: two of the original planform variables, root chord and tip chord, along with two new design variables (Fig. 3). The first new design variable is the inboard leading-edge sweep angle, which can be calculated from the original planform variables x_2 and x_3. The second new variable requires a more elaborate explanation. In this example design problem t/c values at the wing root, leading-edge break, and wing tip are obtained from a baseline HSCT design. The new variable is simply a constant scaling factor by which the three baseline t/c ratios are modified. By using

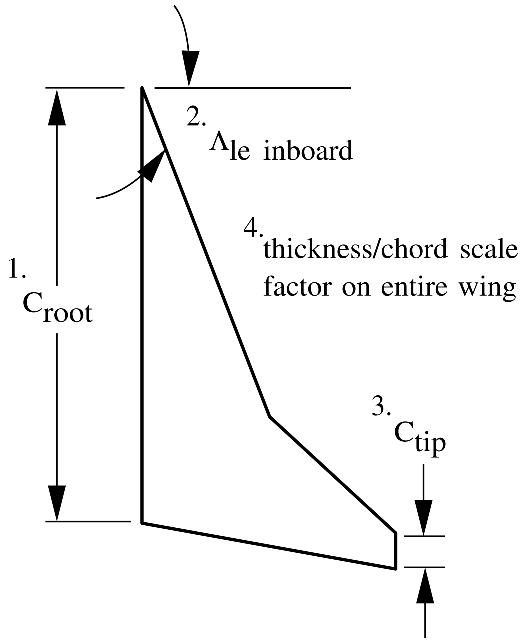

Fig. 3 Wing design variable definition for
the four variable problem.

this t/c scaling factor a single variable replaces the three t/c ratio variables. In addition to the new variable definitions, planform variables x_4 and x_5 are eliminated so that the trailing edge of the wing is straight.

Variations in the root chord have a significant effect on both the structural weight of the wing and on the volume available within the wing for fuel stowage. These characteristics directly influence the gross weight of the HSCT. Perturbations in the leading-edge sweep and the t/c scaling factor primarily affect supersonic drag due to lift, the volumetric wave drag, and wing weight. Thus, the range and gross weight of the HSCT are affected through variations in the drag components and structural weight. Since part of the paradigm proposed here uses regression analysis to eliminate terms in the response surface approximation, the tip chord variable is selected specifically because it is expected to have only a minor impact on the weight and performance of the HSCT. Although this is true for the small supersonic example problem considered here, the aerodynamics of the wing tip region can strongly influence the design of subsonic aircraft. Therefore, the tip chord is not a design variable that can be ignored in general.

For this design problem the takeoff gross weight is the objective function to be minimized, subject to sixty-six aerodynamic/performance and geometric constraints. The design space for this four variable problem is defined by allowing the root chord and tip chord to vary ± 20 percent from the values on the baseline HSCT. The t/c scaling factor also is permitted to vary ± 20 percent from a nominal value of unity. The leading-edge sweep is allowed to range only ± 9 percent from its baseline value. The inboard leading edge segment length $\sqrt{x_2^2 + x_3^2}$, outboard leading edge sweep angle, and wing semispan x_8 are fixed at their baseline values, so that the inboard leading edge sweep angle alone, with the tip chord and absence of trailing edge break, completely determine the wing shape. Variations in the sweep angle outside of this range produce configurations which are completely nonsensical.

2 VARIABLE-COMPLEXITY PARADIGM

The term "variable-complexity modeling" means the process by which simple, computationally inexpensive analysis techniques are used together with more detailed, expensive techniques in the design optimization process. Originally, this methodology was developed for gradient based optimization techniques in which the overall design process is composed of a sequence of optimization cycles. With this method the detailed analyses are employed at the beginning of each optimization cycle while the simple analyses, scaled to match the initial detailed results, are performed in subsequent calculations during each cycle (Hutchison, 1994). In a broader context, variable-complexity modeling is a mundane version of the concept of scale, which forms the basis of such fundamental mathematical tools as renormalization groups, fractals, and wavelets.

At the beginning of each optimization cycle, approximations to the aerodynamic drag components are constructed using either linear, scaled, or global-local approximations (Hutchison, 1993). The scaled approximation method employs a constant scaling function, σ, given as

$$\sigma(x_0) = \frac{f_d(x_0)}{f_s(x_0)}, \tag{1}$$

where f_d represents a detailed model analysis result, and f_s represents a simple model analysis result, both evaluated at a specified design point, x_0, at the beginning of an optimization cycle. During an optimization cycle, the scaled approximate analysis results, $f(x)$, are calculated as

$$f(x) = \sigma(x_0) f_s(x). \tag{2}$$

Move limits then are imposed on the design variables to avoid large errors, and the optimization is performed. At the end of the optimization cycle the scaling function σ is recalculated and the above process is repeated.

There are several precursors to the Hutchison (1994) version of variable-complexity modeling. In Unger et al. (1992), the aerodynamic/structural design optimization of a subsonic transport wing is performed efficiently by first performing a design using a rudimentary weight estimation method along with detailed aerodynamic analyses. The results from the initial design are then used as initial data for an optimization using detailed aerodynamics and detailed structural modeling of the bending material weight. Another implementation of variable-complexity modeling is reported in Hutchison et al. (1992,1993) for the aerodynamic/structural design of an HSCT. Simple and refined models of the various components of the aerodynamics analysis are combined to generate inexpensive, high quality approximations and to greatly reduce the computational cost. The results suggest that significant improvements in the design can be obtained using variable-complexity modeling.

In the variable-complexity paradigm proposed here, the detailed aerodynamic analysis utilizes the Harris program (Harris, 1964) for the supersonic volumetric wave drag (drag due to the generation of shock waves in supersonic flow), a Mach-box (Carlson, 1974, 1978, 1979) type method for supersonic drag due to lift, and a vortex-lattice program for landing performance (Kay et al., 1993). When compared to the computational costs of current advanced computational fluid dynamics analysis techniques, the aerodynamic analysis methods used in this study are relatively inexpensive. However, when implemented in design optimization, where the same calculation may be repeated thousands of times, the cost associated with these techniques (Harris, Mach-box, vortex-lattice) quickly becomes substantial. Therefore, even simpler, less computationally intensive aerodynamic analysis methods are employed for the present variable-complexity modeling approach (Hutchison, 1994).

The variable-complexity modeling paradigm proposed here is tailored for use with response surface based optimization techniques. Here, the simple analysis methods are used to evaluate many different HSCT configurations within a prescribed design space. By applying constraints to the design variables and objective function data, "nonsense" regions of the design space are excluded. The computationally expensive detailed analysis models are then used to more accurately evaluate a small, carefully chosen number of the remaining configurations. From these objective function data, response surface approximations are created which model the objective function over a sliver of the feasible design space. The (approximate) optimal design within this approximation domain is then easily identified using the response surface function. More precisely, the objective function and all computationally expensive constraints are approximated by response surfaces. Since numerous HSCT configurations are independently analyzed in this optimization method, it is particularly advantageous to apply coarse grained parallel computing to both the simple and detailed analyses.

Similarly, variable-complexity modeling is implemented in the HSCT structural analysis. Starting with a large number of candidate HSCT configurations, the designs are screened using algebraic weight equations to eliminate impossible design points. Accurate finite element based optimization is then applied to carefully selected configurations in the remaining design space. These accurate weight values are then used to construct a response surface, which provides a more accurate weight estimation for the HSCT configuration optimization.

3 RESPONSE SURFACE METHODOLOGY

This work focuses on the use of multipoint approximations to fit response surfaces to the noisy objective function and constraints of the HSCT design problem. The efficient construction of a response surface requires several steps:

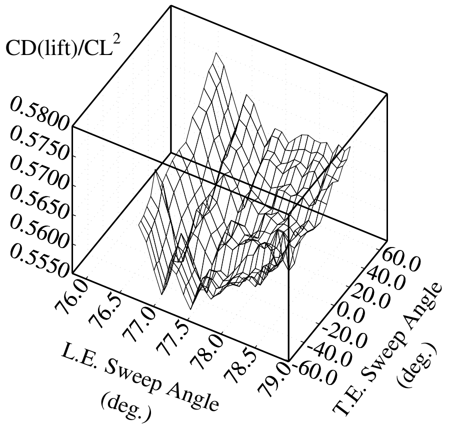

Fig. 4 Noisy drag due to lift objective

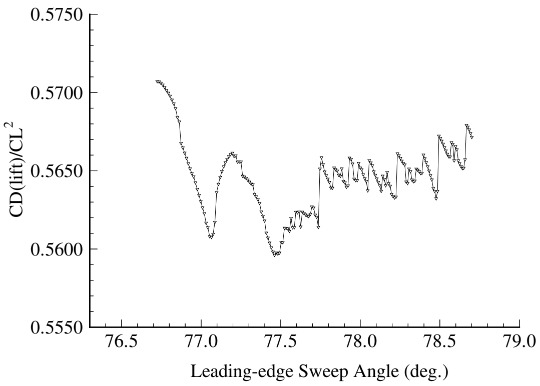

Fig. 5 Noisy drag due to lift vs. leading-edge sweep angle. Trailing-edge sweep angle approximately three degrees.

(1) Many simple analyses are performed on a grid of design points centered at a nominal initial design, and grossly infeasible design points are discarded.
(2) Regression analyses and analysis of variance are performed to eliminate unimportant terms from the response surface models.
(3) Some point selection process is then conducted to find a *small* number of design points at which detailed, expensive analyses will be done.
(4) Response surfaces are constructed based on the detailed analysis data.

These steps are elaborated in the following sections.

3.1 MULTIPOINT APPROXIMATIONS

The term *multipoint* refers to the practice of concurrently analyzing many different HSCT configurations. The multipoint analyses yield a database containing the independent design variables, constraints, and objective function data. *Response surfaces* are smooth functions which define the relationship between the independent and dependent variables (Khuri and Cornell, 1987).

The goal of multipoint approximation is to model the objective function and constraints either on the entire design space or portions of it using smooth response surfaces. Since the topography of a multidimensional objective function generally is unknown and may have many local minima, the smooth functions are selected so that the prominent features of the objective function are retained. Thus, in the optimization process, the region where the global minimum exists may be readily found while spurious local minima may be avoided.

As a simple illustration, consider drag due to lift as the objective function and the inboard leading-edge and trailing-edge sweep angles as the (only) two independent variables. Although the drag due to lift calculations are noisy (Figs. 4 and 5, where only two sweep angles are varied), the variation in the drag is only 2.5 percent. This allows low order response surfaces based on linear and quadratic functions to be used to model the topography of the drag due to lift objective function.

Because of the computational cost associated with evaluating the objective function and constraints at a large number of design points, the present work focuses on methods to determine the minimum number of calculations required to construct various response surfaces. This work with four design variables will be extended later to design problems involving the full 28 variables where computational costs are significant.

3.2 APPROXIMATION DOMAIN

The approximation domain is constructed around a nominal design point, usually obtained from the previous approximate optimization cycle. The domain is bounded by limits on the values of the constraints and the objective function obtained from simple inexpensive approximations. When a simple analysis is several orders of magnitude cheaper than a refined analysis, then the simple analyses define directly the approximation domain. Otherwise, linear approximations to the simple analyses are used. Constraints can bound the approximation domain by requiring that the approximation to the constraint not be violated by more than a certain margin. Similarly, the objective function can bound the approximation domain by limiting the change in the objective function.

Specifically, consider the four variable HSCT design problem described in Section 1.1. Using the constraint data obtained from simple analyses for each of the 1296 HSCT designs in a $6\times6\times6\times6$ grid around the nominal design, obvious "nonsense" configurations are eliminated from consideration. Here, designs are excluded if any of the aerodynamic/performance constraints (e.g., landing angle-of-attack $\leq 12°$) are violated by more than twenty percent, and if any geometric constraints (e.g., minimum airfoil chord lengths $\geq$ 7 ft.) are violated by more than five percent. In addition, gross takeoff weight (GTOW) is allowed to vary within ±20 percent of the baseline GTOW of approximately 650,000 lbs. and range is required to be greater than 5000 n.mi. Both of these constraints are imposed to remove from consideration any unrealistic designs which have not been eliminated previously. After applying these constraints, only 157 acceptable HSCT designs remain out of the initial 1296 designs. The *approximation domain* consists of these 157 design points. Figures 6 through 9 show projections of the approximation domain for the four design variable problem, each figure being the projection along one of the four design variable axes.

This same process for the full 28 variable design problem would result in 6^{28} analyses, clearly not practical. Currently this is being addressed by using partially balanced incomplete box designs, a standard statistical experimental design technique. Very sparse sampling techniques are the subject of active research within both the statistics and MDO communities.

3.3 POLYNOMIAL MODELING

Response surface methodology is a statistical technique in which smooth functions, typically polynomials, are used to model an objective function or constraints. For example, a quadratic response surface model (also known as a regression model) has the form

$$y = c_0 + \sum_i c_i x_i + \sum_{1\leq i\leq j\leq p} c_{ij} x_i x_j + \epsilon, \tag{3}$$

where the x_i are the design variables, the c_i are the polynomial coefficients, y is the measured response, and ϵ is a random error term. Typically, the polynomial (regression) coefficients are estimated using the method of least squares (cf. Section 3.4).

The construction of a response surface requires a minimum of n function evaluations where n is the number of coefficients in the polynomial model. Results from Giunta et al. (1994) confirm that typically $1.5n$ function values are required to produce response surfaces which accurately approximate the global trends of the data.

An example of a situation where response surface modeling techniques are applicable is contained in Giunta et al. (1994), where supersonic drag due to lift on an HSCT wing is calculated for various inboard leading-edge and inboard trailing-edge sweep angles (Fig. 4). Here, drag due to lift is calculated as

$$C_{Dlift} = (\frac{1}{C_{L\alpha}} - k_t \frac{C_T}{{C_L}^2}){C_L}^2, \tag{4}$$

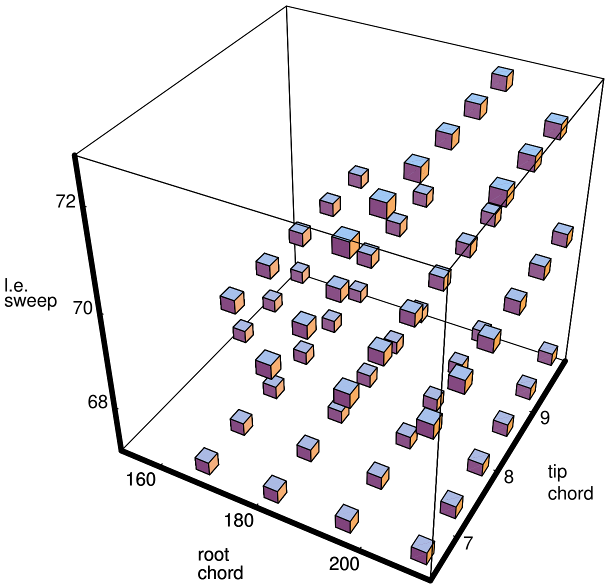

Fig. 6 Approximation domain projected
along t/c ratio axis.

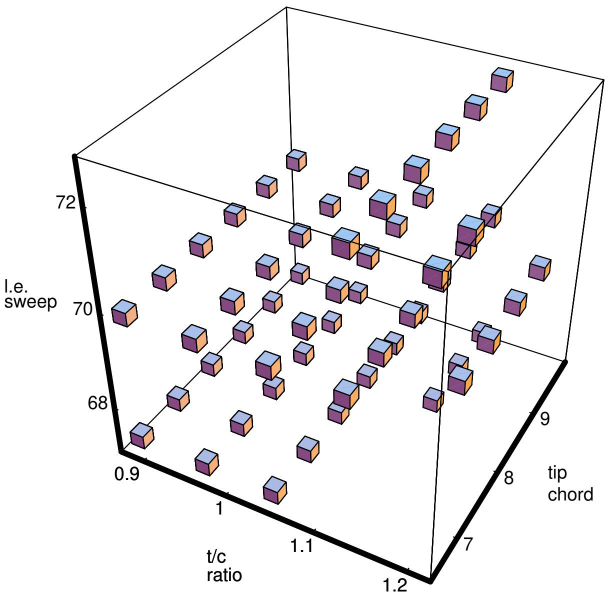

Fig. 7 Approximation domain projected
along root chord axis.

where $C_{L\alpha}$ is the lift curve slope, $C_T/{C_L}^2$ is the leading-edge thrust term, and k_t is an attainable leading-edge thrust factor. The numerical noise in the drag due to lift evaluation may be attributed to numerical noise in the lift curve slope and leading-edge thrust terms. The methods of Carlson et al. (1974, 1978, 1979) utilize a paneling scheme that is sensitive to planform changes. Thus, slight modifications to the leading and trailing-edge sweep angles, along with changes in the location where the Mach angle intersects the leading-edge, produce discontinuous variations in the predicted drag. Although the variations are small enough so that at all points the accuracy of the drag is acceptable, the oscillatory behavior creates difficulties

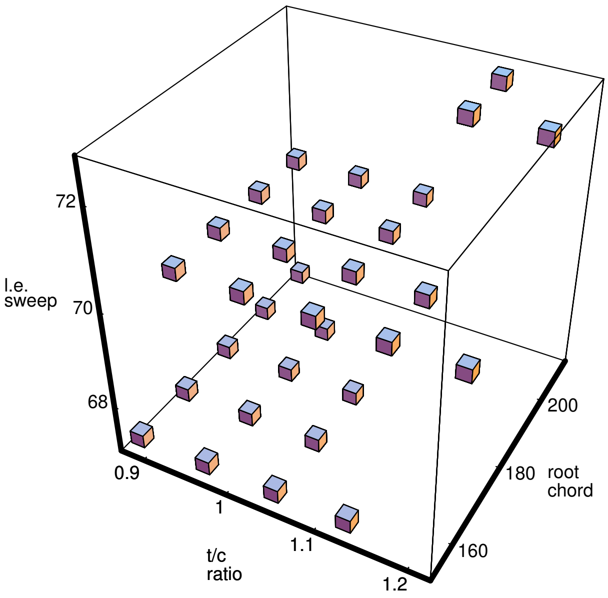

Fig. 8 Approximation domain projected along tip chord axis.

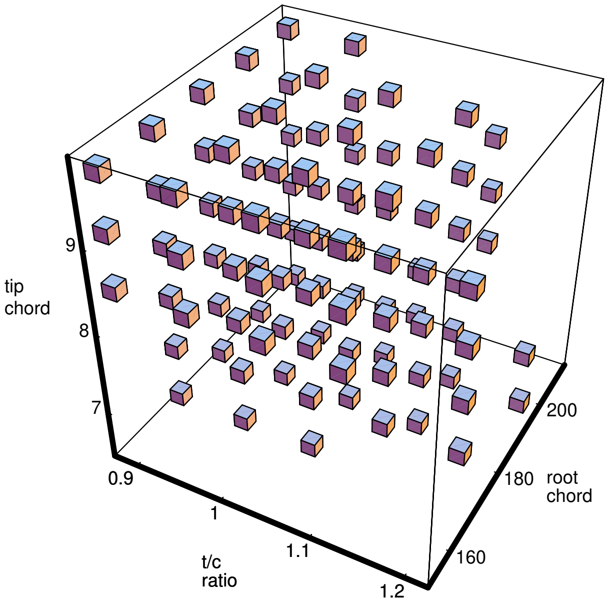

Fig. 9 Approximation domain projected along inboard leading edge sweep angle axis.

for gradient based optimization techniques.

Response surface approximations are constructed using both polynomial functions and rational functions. The bilinear tensor product model has the form

$$f(x,y) = (a_1 x + a_2)(a_3 y + a_4). \tag{5}$$

Also employed are the quadratic polynomial

$$f(x,y) = a_1 x^2 + a_2 xy + a_3 x + a_4 y^2 + a_5 y + a_6 \tag{6}$$

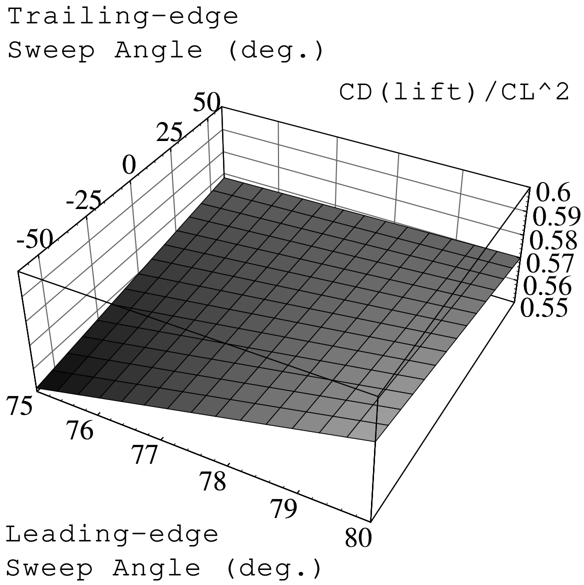

Fig. 10 Example bilinear tensor product response surface fit to the noisy objective function data.

and the biquadratic tensor product

$$f(x,y) = \left(a_1x^2 + a_2x + a_3\right)\left(a_4y^2 + a_5y + a_6\right). \tag{7}$$

In addition to these models, the (1,1) rational function

$$f(x,y) = (a_1x + a_2y + a_3)/(a_4x + a_5y + a_6) \tag{8}$$

is utilized.

Figures 10, 11, and 12 demonstrate the use of linear, quadratic, and biquadratic polynomial response surfaces to approximate the noisy drag due to lift in Figures 4 and 5. The response surface figures are generated using 100 randomly selected points in the design space, which is well above the minimum number of points needed to construct a response surface which adequately models the objective function. Initially, the rational function (8) produced response surfaces similar to those of the bilinear tensor product. Therefore, further analysis using the rational function was not conducted.

Although the bilinear response surface provides a reasonable prediction of the overall trend of the computed noisy surface in Figure 4, it is evident that the quadratic and biquadratic response surfaces best capture the trends of the noisy surface. The global minimum on the exact, noisy surface occurs for a leading-edge sweep angle of 77.6° and a trailing-edge sweep angle of $-20.0°$, which is roughly predicted in Figures 11 and 12.

3.4 REGRESSION ANALYSIS

In many design problems the designer encounters the *curse of dimensionality*, in which the number of analysis points needed to construct a response surface model increases exponentially with the number of design variables. Although this issue is not a problem in using response surface methodology with a four design variable problem, it is sure to arise when response surface methodology is applied to the full HSCT design problem, which has twenty-eight design variables. For this reason, it would be advantageous if less significant terms in the response surface model could be eliminated. As an example, a quadratic polynomial in p variables has $(p+1)(p+2)/2$ terms. A response surface in ten variables then would have sixty-six terms in the polynomial model and would require approximately one hundred analysis points to fit the model. However, if one third of the polynomial terms are insignificant, then the function could be represented with a forty-four term polynomial without a significant loss in accuracy. This reduced-term polynomial could then be constructed using approximately seventy analysis points. In multidisciplinary design optimization where each analysis may incur significant computational expense, the computational savings associated with the reduced-term polynomial may be substantial.

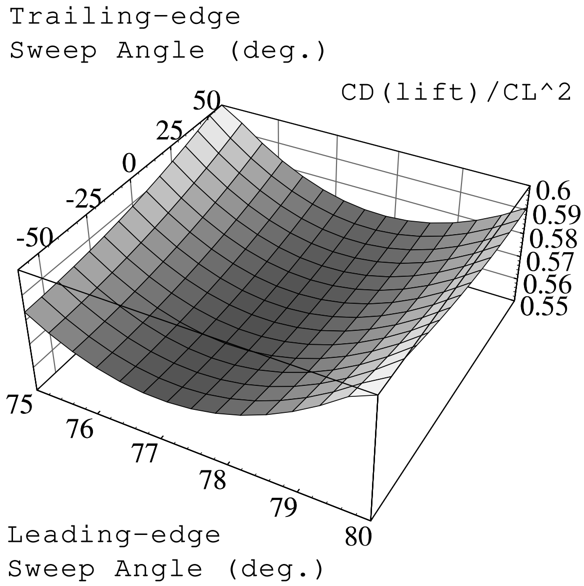

Fig. 11 Example quadratic polynomial response surface fit to the noisy objective function data.

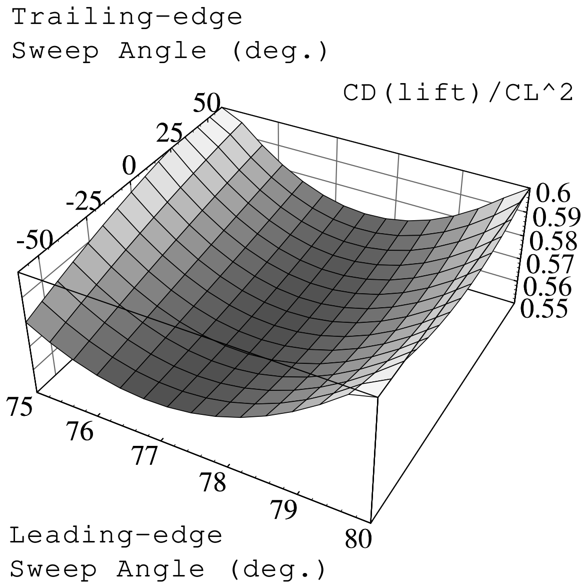

Fig. 12 Example biquadratic tensor product response surface fit to the noisy objective function data.

Statistical techniques known as regression analysis and analysis-of-variance provide the means to identify and remove the less important terms in the response surface polynomial models based on simple analyses, thereby reducing the number of detailed analyses needed for later construction of accurate response surfaces. Regression analysis is the procedure by which the c_i coefficients for the response surface model are obtained and typically involves the method of least squares, as follows. Let there be m measurements Y_i at m distinct design points $(x_1^{(i)}, \ldots, x_p^{(i)})$, $i = 1, \ldots, m$, and let the basis functions for the regression model be

$\varphi_1(x), \ldots, \varphi_n(x)$, where $x = (x_1, \ldots, x_p)$. A choice for the φ_i might be

$$1, x_1, x_2, \ldots, x_p, x_1^2, x_1 x_2, \ldots, x_p^2,$$

corresponding to a quadratic model. The regression model is

$$y \approx c_1 \varphi_1(x) + \cdots + c_n \varphi_n(x),$$

which leads to the overdetermined matrix problem

$$Y \approx \mathbf{X} c, \tag{9}$$

where $Y = (Y_1, ..., Y_m), c = (c_1, ..., c_n)$, and $\mathbf{X}$ is the $m \times n \quad (m \geq n)$ matrix

$$\begin{pmatrix} \varphi_1(x^{(1)}) & \ldots & \varphi_n(x^{(1)}) \\ \vdots & \ddots & \vdots \\ \varphi_1(x^{(m)}) & \ldots & \varphi_n(x^{(m)}) \end{pmatrix}.$$

The least squares solution is that (unique, assuming rank $\mathbf{X} = n$) vector $\tilde{c}$ which minimizes the 2-norm $||Y - \mathbf{X}c||_2$, or the sum of squares of the components of $Y - \mathbf{X}c$, the "errors." The regression or response surface model of the data is then taken as

$$\tilde{c}_1 \varphi_1(x) + \cdots + \tilde{c}_n \varphi_n(x).$$

The matrix $(\mathbf{X}^T \mathbf{X})^{-1}$ is called the variance-covariance matrix. The n diagonal elements in this matrix are the variance values associated with the n respective coefficients (Khuri and Cornell, 1987). The standard deviation σ_k of each coefficient is the square root of its variance.

Analysis of variance (ANOVA) involves estimating the variance of the predicted polynomial coefficients and uses the variance-covariance matrix $(\mathbf{X}^T \mathbf{X})^{-1}$. The diagonal terms ${\sigma_i}^2$ in this square matrix multiplied by the variance $\mathrm{Var}(\epsilon)$ in the measured function values Y_i are the variances of the respective coefficients c_i in the response surface polynomial model. The coefficient of variation Υ for each term in the polynomial is calculated as

$$\Upsilon = \begin{cases} \dfrac{100|\sigma_i|\sqrt{\mathrm{Var}(\epsilon)}}{c_i}, & c_i \neq 0, \\ 100, & c_i = 0, \end{cases} \tag{10}$$

where the factor of 100 expresses the coefficient of variation as a percentage. The term $\sqrt{\mathrm{Var}(\epsilon)}$ is usually estimated by the RMS error of the least squares approximation at the m data points. For the coefficient of variation calculations given below, $\mathrm{Var}(\epsilon)$ is taken to be unity. Terms in the polynomial model having large coefficients of variation, typically over 10%, may be dropped from the polynomial without significantly affecting the fidelity of the response surface fit.

Returning to the four variable HSCT design problem, step (1) of the response surface construction process yields 157 design points from an initial $6 \times 6 \times 6 \times 6$ grid. With the data from these 157 simple HSCT analyses a fifteen term quadratic polynomial response surface model is fit to the aircraft range data. Using the regression analysis and ANOVA methods described above, the coefficients of variation for the fifteen terms in the response surface model are calculated (Table 1). Here, the abbreviations ζ, $\mathrm{c_r}, \mathrm{c_t}$, and Λ_{LE} correspond to the t/c scaling factor ζ, root chord, tip chord, and leading-edge sweep angle, respectively. As shown, the higher order terms involving $\mathrm{c_t}$ have coefficients of variation greater than ten percent and can safely be dropped from the response surface model. Thus, the number of terms in the response surface model has been reduced to eleven and the modeling of the tip chord variable has been simplified from quadratic to linear.

Table 2 shows that the accuracy of the response surface fit is only slightly impaired after removing terms from the polynomial model for which the coefficient of variation is large. Here, the errors are calculated from the difference between the response surface prediction for the range and the actual value for the range at each of the 157 remaining HSCT design points.

Since range is directly affected by numerical noise in the supersonic drag due to lift and volumetric wave drag calculations, a response surface model for range is used to determine the reduced-term polynomial model structure for all of the drag components. An alternative approach, and one that will be considered in future work, would be to apply regression analysis and ANOVA separately to response surface models for each of the drag components.

Table 1. Regression analysis and ANOVA data.

Variable	Coefficient	Std. Dev.	Υ (%)
const.	0.058	0.197	3.416
ζ	-1.200	0.268	0.224
$\mathrm{c_r}$	-0.555	0.206	0.372
$\mathrm{c_t}$	-0.054	0.133	2.443
Λ_{LE}	0.755	0.261	0.346
$\zeta\mathrm{c_r}$	-0.221	0.627	2.835
$\zeta\mathrm{c_t}$	0.009	0.372	40.616
$\zeta\Lambda_{LE}$	0.170	0.848	4.990
$\mathrm{c_r c_t}$	-0.025	0.290	11.621
$\mathrm{c_r}\Lambda_{\mathrm{LE}}$	0.058	0.572	9.907
$\mathrm{c_t}\Lambda_{\mathrm{LE}}$	-0.021	0.358	16.955
ζ^2	-0.095	0.623	6.588
$\mathrm{c_r}^2$	-0.147	0.352	2.399
$\mathrm{c_t}^2$	-0.006	0.234	42.143
$\Lambda_{LE}{}^2$	-0.107	0.700	6.526

Table 2. Calculated errors.

Avg. Error	RMS Error	Max. Error
15 Term Polynomial		
0.034395	0.044025	0.126625
11 Term Polynomial		
0.035187	0.045363	0.113389

4 POINT SELECTION TECHNIQUES

To construct a response surface for an objective function or constraint involving p independent variables, a large number of individual configuration analyses are required. To use computational resources wisely, the designer must consider both the quantity and the distribution of the design points in the p-dimensional design space. In this regard, three design point selection techniques are examined here: central composite design, random, and D-optimal. The validity of these depend on statistical assumptions about the nature of the computational noise. It may be that other statistical estimation procedures, such as robust M- or LMS-estimation (Mainguy et al., 1995), are more appropriate for the computational noise.

4.1 CCD POINT SELECTION

One method for selecting the points used in response surface construction is called central composite design (CCD). This criterion, which is based on design of experiments theory (Mason et al., 1989), permits the estimation of both the quadratic and linear coefficients in equations (5–8) using a total of $2^p + 2p + 1$ points, where p is the number of design variables. This is in contrast to the 3^p points required for a full factorial design. For p greater than two, the number of CCD points is smaller than the number of factorial points. Unfortunately, for twenty-eight design variables, even CCD requires an unacceptably large number of points. However, the CCD criterion provides a useful standard of comparison to the other point selection techniques.

For the case $p = 2$, CCD requires nine point analyses, which is the same as that required for a 3^p full factorial design. However, the location of the CCD points in the design space is different than the location of the full factorial points. The parameter $\beta = (2^p)^{1/4}$ determines the spacing for the CCD points, as illustrated in Figure 13. Regardless of the computational cost of a CCD, one very serious practical problem is that it would be difficult to apply a CCD in an irregular region like the approximation domain here.

4.2 RANDOM POINT SELECTION

One way to deal with the irregular approximation domain generated by step (1) (cf. Section 3) is to simply pick m points at random from the l points identified by step (1). List the points as $x^{(1)}, x^{(2)}, \ldots, x^{(l)}$, and let u be a uniform random variate on the interval (0,1). Then

$$i = \lceil lu \rceil$$

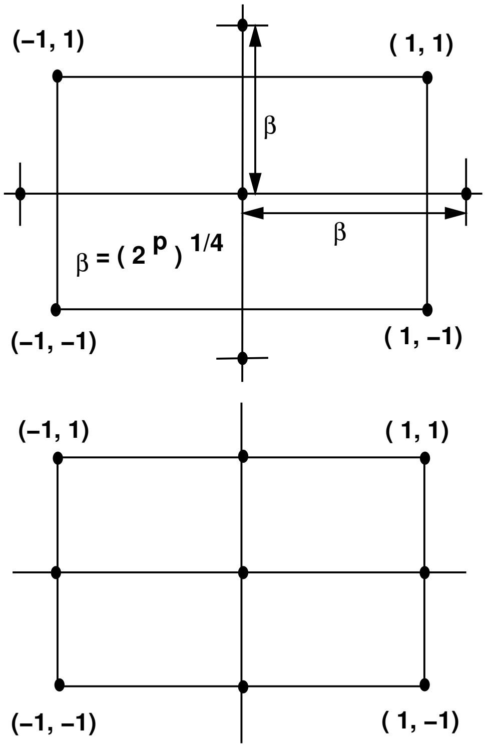

Fig. 13 CCD (top) and full factorial design (bottom) point distributions for a two variable problem.

picks a random index between 1 and l, and $x^{(i)}$ a random point from the approximation domain. Continue picking random points, eliminating duplicates, until m distinct points from the list have been selected. This random selection is fast and elegant, but also with high probability can generate a "lopsided" set of points within the approximation domain, biasing the resultant response surface. This phenomenon was verified in Giunta et al. (1994) for the set of $l = 861$ drag due to lift data points used to construct Figure 4 and $m = 50$.

4.3 D-OPTIMAL POINT SELECTION

A third candidate point selection technique is to choose points in the approximation domain which satisfy D-optimality (Box and Draper, 1971). Intuitively, D-optimality seeks the m points from the set of $l > m$ candidate points comprising the approximation domain that will yield the best fidelity between the polynomial response surface model and the true underlying function. A genetic algorithm is used here to estimate such a set of m points.

4.3.1 D-OPTIMALITY CRITERION

The D-optimality criterion states that the m points to choose are those which maximize the determinant $|\mathbf{X}^T\mathbf{X}|$, with $\mathbf{X}$ defined by (9). Several properties of this criterion are (Box and Draper, 1971):

- the set of points that maximizes $|\mathbf{X}^T\mathbf{X}|$ is also the set of points that minimizes the maximum variance of any predicted value from the regression model,
- the set of points that maximizes $|\mathbf{X}^T\mathbf{X}|$ is also the set of points that minimizes the variance of the parameter (polynomial coefficient) estimates,
- the design obtained is invariant to changes in scale.

The problem of finding the m points that will give the best fidelity approximation in the above sense is the problem of finding the m points that will maximize $|\mathbf{X}^T\mathbf{X}|$. Conceivably, one could consider each of the $\binom{l}{m} = l!/(m!(l-m)!)$ combinations of m points from the set of l candidate points, evaluate $|\mathbf{X}^T\mathbf{X}|$, and identify the set with the largest determinant. A small problem in two design variables may be to pick twenty-five points from 121 possible points (discretizing the design domain into ten sections in both directions leads to an 11×11 mesh). This leads to a total of $5.26 \cdot 10^{25}$ possible combinations, one or more of which are D-optimal.

It is clear that a heuristic method must be applied to estimate the optimum $|\mathbf{X}^T\mathbf{X}|$. This nonconvex combinatorial optimization problem is difficult because:

- the discrete feasible set is huge, and
- $|\mathbf{X}^T\mathbf{X}|$ may have local maxima.

For lack of a better alternative, a genetic algorithm (GA) is used to maximize $|\mathbf{X}^T\mathbf{X}|$.

4.3.2 GENETIC ALGORITHM

The GA employed here is roughly modeled after the algorithm described in Furuya and Haftka (1993). In general, a GA works with a population of b designs, each representing in this case a choice of m out of l possible points. (Here, *design* refers to an experimental design, not an HSCT configuration.) The population of designs goes though a selection process for breeding, whereby designs with a higher merit function, $|\mathbf{X}^T\mathbf{X}|$ here, have a higher probability of being selected for breeding. Breeding is performed in such a manner that the child designs maintain some likeness to the parent designs. The next generation of designs is comprised of $b-1$ children created during the breeding process plus the best parent design of the parent generation. Thus, the population size remains constant throughout the generations. This is known as an "elitist" strategy. The process continues for many generations, and terminates after a fixed number of generations, although there are many other reasonable stopping criteria which may be applied. The design with the best merit function in the end is used. The GA for seeking D-optimality is described next.

The intent of the GA is to select the group of m points that satisfy D-optimality for the purpose of constructing a least squares approximation to detailed analysis values at the m D-optimal points. Any group of m distinct points among the l points is called a candidate. To start the algorithm, an initial population of b candidates is created. Each candidate is formed by randomly selecting m distinct points from the set of l points comprising the approximation domain.

Consider first the situation where the approximation domain is a Cartesian product (a regular grid of points in the p-dimensional space), where the GA is easy to describe. A genetic string, made from the coordinates of the m points, is used to describe the candidate. For example, in a three dimensional design space, if $m=4$, the genetic string of a particular design would be

$$x_{1,1}\, x_{2,1}\, x_{3,1}\, x_{1,2}\, x_{2,2}\, x_{3,2}\, x_{1,3}\, x_{2,3}\, x_{3,3}\, x_{1,4}\, x_{2,4}\, x_{3,4}.$$

The length G of the genetic string (the number of genes) is the dimension p of the design space multiplied by m ($G = 12$ in the above example).

The candidates are selected as parents for breeding based on the fitness $|\mathbf{X}^T\mathbf{X}|$. Candidates with a higher fitness will be selected more often than candidates with lower fitness. The rank fitness is defined as $b+1-r$, where r is the rank of the candidate in the population in terms of the value of $|\mathbf{X}^T\mathbf{X}|$. The probability of selection as a parent is proportional to the rank fitness, so that the probability of the rth ranked design being selected as a parent is

$$p_r = \frac{(b+1-r)}{b(b+1)/2}. \tag{11}$$

The selection process for parenting is completed by generating a uniformly distributed random number x between zero and one, and selecting the rth ranked design satisfying $P_r \le x < P_{r+1}$, where

$$P_r = \sum_{i=1}^{r-1} p_i. \tag{12}$$

After two parents have been selected the breeding process begins. A random integer j between one and $G-1$ (the number of genes-1) is generated. The child is constructed from the first j genes of the first parent and the $G-j$ genes from the second parent. Note that this process (*crossover rule* in GA parlance) only makes sense when the approximation domain is a Cartesian product. Once the child is generated it goes through a mutation process. Each gene has a chance of mutating, and a fifteen percent chance is used in the present work. If a gene is selected for mutation, the gene is replaced with a gene coming randomly from the set of allowable values for that gene. Finally the child is checked for uniqueness and duplicate points. If it is a duplicate or has duplicate points, it is destroyed.

Now consider the general situation where the approximation domain is an irregular (possibly even unconnected) region. Number the points from 1 to l, and take the genetic sequence of a candidate design to be the string $a_1a_2 \ldots a_m$ of integer indices corresponding to the m points comprising the design, and assume

$$a_1 < a_2 < \ldots < a_m$$

for convenience. If $m \ll l$, the above crossover rule works reasonably well, and was used for the GA results reported here. (For a regular domain, the genetic string of point coordinates is more effective than the genetic string of point indices, in the sense that fewer generations are required to find a good design.) Children with duplicate points (indices) must be discarded, and as m grows relative to l, this discarding occurs more frequently. A more efficient crossover rule, which avoids creating children with duplicate indices altogether, is described next.

Every other detail of the proposed GA for irregular domains is the same as above, except for the crossover rule. One of many plausible crossover rules is described here. Let $a_1a_2 \cdots a_m$ and $b_1b_2 \cdots b_m$ be two distinct parents chosen for breeding. Delete from $b_1b_2 \cdots b_m$ any indices also occurring in the string $a_1a_2 \cdots a_m$, resulting in

$$b_{i_1}b_{i_2} \cdots b_{i_r}.$$

Now choose integer $k, 1 \le k \le r$, at random and delete k randomly chosen genes (indices) from $a_1a_2 \cdots a_m$, resulting in

$$a_{j_1}a_{j_2} \cdots a_{j_{m-k}}.$$

Into this string of length $m - k$ merge k randomly chosen genes from $b_{i_1} \cdots b_{i_r}$, giving a child with m (guaranteed distinct) genes. A second child is similarly generated with another random $k, 1 \le k \le r$. The random choices of k objects are made by generating random j, as described above, in the appropriate range repeatedly until k distinct indices have been selected. Note that although a child cannot have duplicate genes, there may still be duplicate children, which must be eliminated as described before.

After $b - 1$ children are created, the parent generation is replaced, retaining only the best parent for the next generation. After some stopping criterion has been satisfied, the candidate with the highest $|\mathbf{X}^T\mathbf{X}|$ is used for forming the response surface. There is no guarantee that the output of the genetic algorithm is truly D-optimal. However, known D-optimal sets of points for test cases have been recovered using the GA.

For the small two and four-dimensional HSCT design problems considered here, a population size of $b = 10$ suffices, with the number of generations varying from 2000 to 25000 depending on the number of points m needed for constructing the various response surfaces. The values $|\mathbf{X}^T\mathbf{X}|$ are monitored during the breeding process and when $\max |\mathbf{X}^T\mathbf{X}|$ remains unchanged for approximately 1000 generations, the breeding process is terminated. As stated above, each gene has a fifteen percent chance of mutating during each generation.

5 ANALYSIS OF POINT SELECTION TECHNIQUES

5.1 REGULAR APPROXIMATION DOMAIN

It is clear that a regular experimental design is ill suited to the irregular approximation domains occurring in the HSCT design problem. Nevertheless, it is instructive to compare the various point selection strategies on a regular Cartesian product domain. To that end, consider the 2-dimensional data illustrated in Figure 4, which came from a uniform 21×41 grid of 861 points.

The biquadratic tensor product response surface, using the D-optimal point selection technique, provides the most accurate fit to the noisy surface. The same type response surface based on CCD points provides an adequate fit, while that based on the random point selection technique is unacceptable, unless a much larger number of points is used for the fit. See Appendix B for a detailed analysis of the point selection techniques for each type of response surface fit.

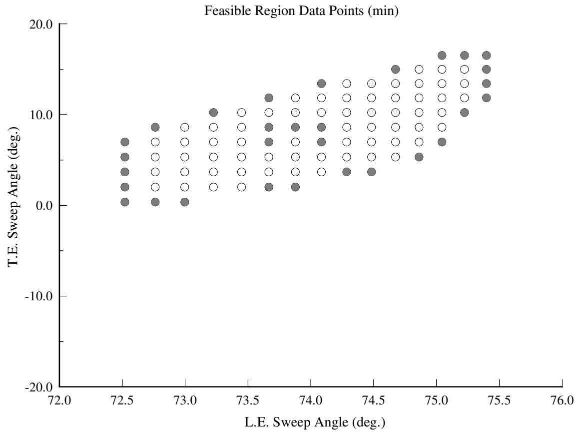

Fig. 14 Thirty D-optimal points (grey) selected by the GA for a quadratic response surface fit.

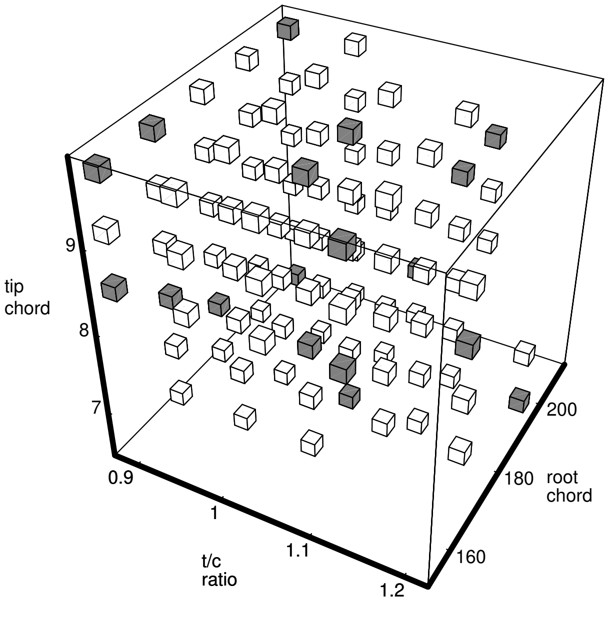

Fig. 15 Twenty D-optimal points selected by the GA, inboard leading edge sweep angle projection.

5.2 IRREGULAR APPROXIMATION DOMAIN

Figure 14 shows an approximate D-optimal set of thirty points obtained by the GA for a quadratic polynomial response surface model to the data in Figure 4 (a two design variable problem). Note that the D-optimal points (shown in grey) lie predominately along the perimeter of the irregular approximation domain (the circles). For a bilinear tensor product response surface model, all points lie on the perimeter. This behavior is expected since points near the perimeter of the approximation domain have the strongest influence on $|\mathbf{X}^T\mathbf{X}|$.

Figure 15 shows the projection along the inboard leading edge sweep angle axis (as in Fig. 9) of an approximate D-optimal set of twenty points (grey cubes) obtained by the GA for the 11 term quadratic polynomial response surface models of the wave drag and two components of drag due to lift (cf. Section 3.4 and Table 2). The response surface functions are then used to calculate the range constraint in the four design variable problem. The cubes show the 157 points comprising the approximation domain. Since

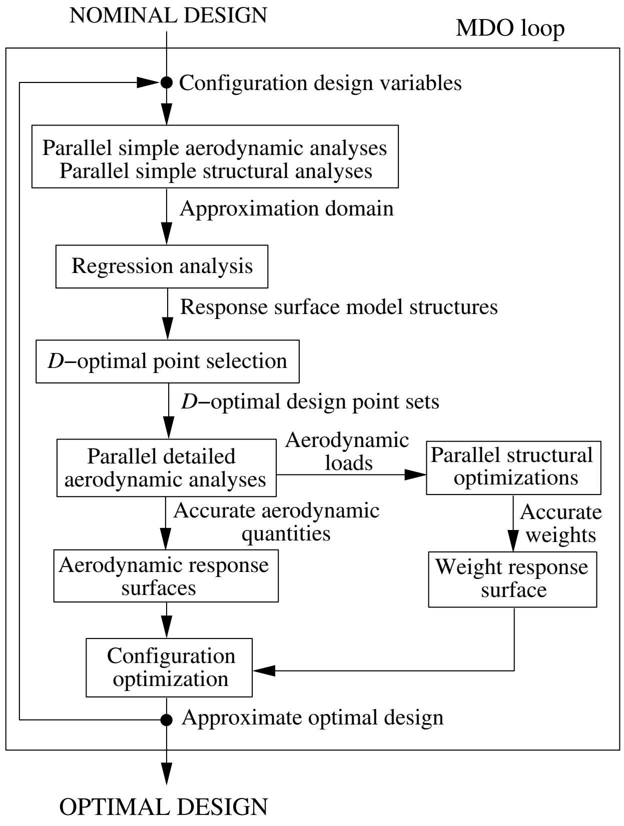

Fig. 16 MDO Paradigm.

some of the points in the approximation domain differ only in the sweep angle, the sweep angle projection shows fewer than the 157 total cubes and 50 grey ones. Similar to Figure 14, the grey D-optimal points lie predominately along the perimeter of the approximation domain.

6 MDO PARADIGM

The previous sections have detailed the individual components of the present MDO paradigm. At this point, a recapitulation of the salient concepts from these sections is useful.

- *Variable-complexity modeling* makes use of simple, computationally inexpensive models together with detailed, computationally expensive models in the optimization design process.
- *Multipoint approximations* are used to construct smooth response surface models of the objective function and expensive constraints, and fit well in a coarse grained parallel computing environment.
- In order to manage the number of multipoint approximations, the design space is limited to an *approximation domain* which excludes grossly infeasible design points.
- *Regression analysis* is performed to reduce the number of coefficients needed by the response surface models.
- A *genetic algorithm* is used to find sets of approximate *D-optimal points,* which are used for constructing polynomial response surface models.

The MDO paradigm presented here incorporates all these individual concepts.

The MDO algorithm is a repeat loop, with a nominal design as its starting point, approximate optimal designs as loop iterates, and an optimal design as its ending point (see Fig. 16). At the start of each loop, aerodynamic shape and mission variables are obtained from either the nominal starting design or the intermediate approximate optimal design. These shape and mission variables are then used in the parallel simple aerodynamic and structural analyses.

The simple aerodynamic analyses are performed on a regular grid of points in the design space. Simple aerodynamic calculations evaluate the (aerodynamic) feasibility of each grid point using tolerances on the

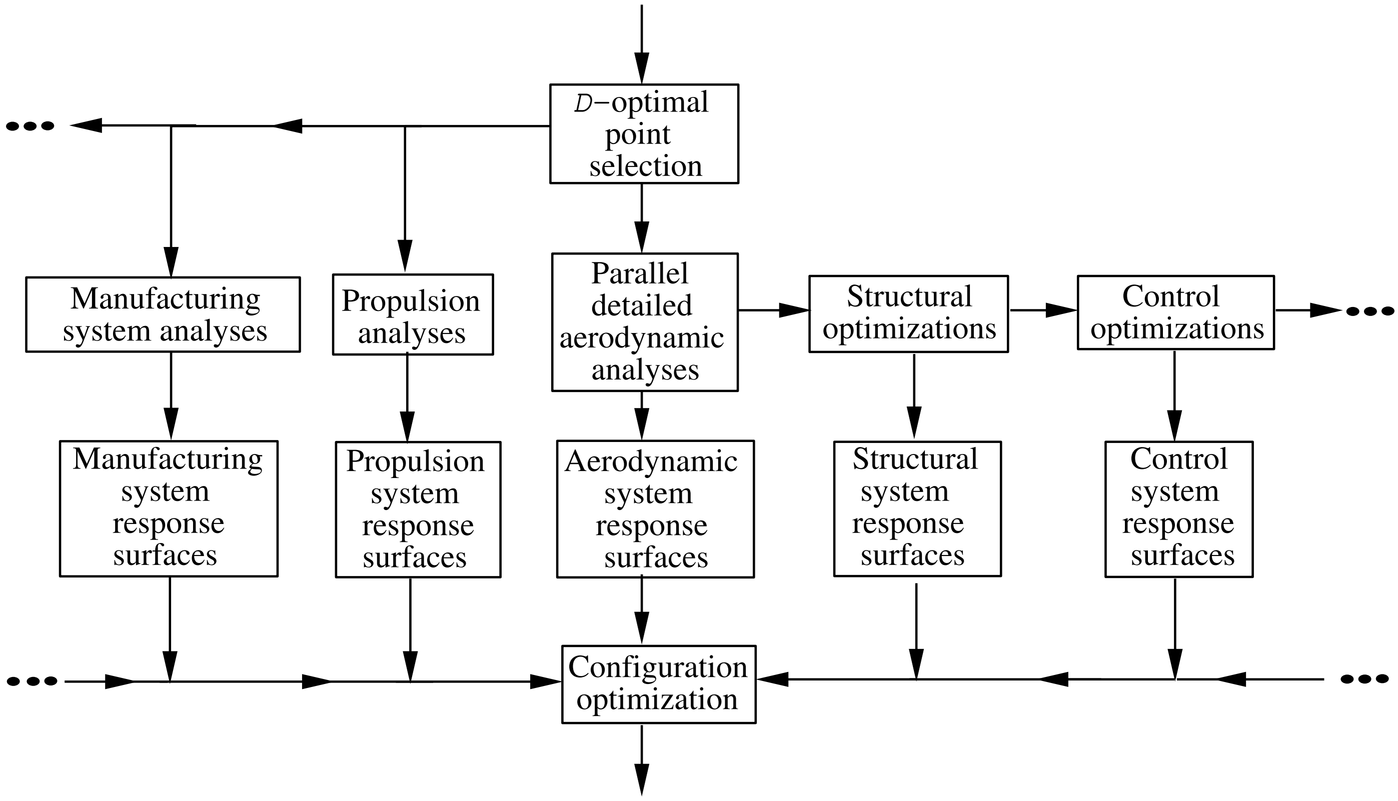

Fig. 17 General MDO paradigm.

constraints and move limits on the objective function, eliminating grossly infeasible points, and generating an approximation domain. The simple structural analyses use the aerodynamic shape and mission variables in basic weight equations to calculate approximate weights needed by the objective function and constraints, further refining the approximation domain.

Using the relatively abundant data from the simple analyses, regression analysis and analysis of variance are used to identify less important terms in the polynomial response surface models. Once the less important terms are eliminated, the structure of the reduced-term polynomial regression models is known, and can be used later in the generation of response surface approximations of the optimal weight and necessary aerodynamic quantities over the approximation domain.

A genetic algorithm is used to find sets of approximate D-optimal design points in the approximation domain obtained from the parallel simple analyses. The structure of a response surface model is embodied in the matrix $\mathbf{X}$, which defines the GA merit function $|\mathbf{X}^T\mathbf{X}|$. These D-optimal design points are input to the detailed aerodynamic analysis code, which performs detailed analyses at each of the D-optimal design points in parallel. The analyses result in accurate aerodynamic quantities, such as wave drag and other drag components (as in the four design variable problem), and accurate aerodynamic loads.

The accurate aerodynamic quantities are used to generate reduced-term polynomial response surface models for each of the expensive quantities (such as wave drag). An aerodynamic load calculated in the detailed aerodynamic analyses is used in a detailed structural optimization to calculate an accurate optimal weight for that particular aerodynamic load. This structural optimization is done (in parallel) for each aerodynamic load generated in the detailed aerodynamic analyses. The accurate optimal weights calculated in the structural optimization are used to generate a reduced-term polynomial response surface model for the optimal weight.

All the response surface models are then used in a configuration optimization to generate an approximate optimal design, which will be used as the starting design for the next iteration of the MDO loop. The grid spacing may possibly be refined for the simple analyses. When some convergence criterion is satisfied, the MDO loop exits with an optimal design.

Note that the *source of parallelism in the present MDO paradigm is the multipoint approximations within each discipline,* where the disciplines are visited sequentially in a pipeline. This contrasts sharply with other MDO paradigms, where the *source of the parallelism is processing the disciplines in parallel.*

Figure 17 shows how the MDO paradigm would be generalized to include other disciplines. Simple analysis for all the disciplines would be done concurrently, as the precursor to obtaining D-optimal point sets for all the relevant functions involved in all the disciplines. Those disciplines requiring optimization and input from other disciplines would be added to the right of the structural optimizations in a *pipeline.* Notice that the disciplinary optimizations occur in a pipeline, and circular coupling or dependencies are not permitted; to do so might result in nonconvergence of the entire process (Shankar, 1993). Disciplines which are independent, needing only the configuration and D-optimal point sets to perform their analyses, would be added to the left of the D-optimal point selection and could be executed in parallel. Note that these are *analyses*, not optimization, and are not pipelined nor coupled. Any strong couplings, such as aeroelasticity or interaction between fabrication and structural design, will have to be accommodated by the global configuration optimization module.

7 PARALLELIZATION

MDO often requires the linking together of black-box codes from different disciplines. Furthermore, the linking process has to allow for occasional updating of some of these codes and the linking of additional disciplines. Thus, *coarse grained parallelization*, which can be orchestrated with minimal knowledge of the individual disciplinary codes, is implemented in the MDO paradigm presented here.

The overall parallelization task has two distinct subtasks: parallelizing the individual steps (the boxes in Fig. 16) of the MDO loop, and linking everything together so that the entire MDO loop is a single parallel program. The practical difficulty of this latter step should not be underestimated. At the beginning of this MDO effort, a complete MDO outer loop (one configuration optimization step) involved programs in several languages (FORTRAN 66, FORTRAN 77, C, Mathematica), nonportable execution environments on several different machine architectures, extensive and interactive I/O for some program components, manual transfer of data between formats and between machines, and human intervention to “interpret” output and “tune” input parameters. Just to automate the outer MDO loop, without regard to efficiency or parallelizability, is a rather formidable undertaking. Results from each of these parallelization efforts are given in the following sections.

7.1 GENETIC ALGORITHM

A coarse grained parallelization of the GA code was implemented on 28 nodes of a 96 node Intel Paragon, for even population sizes ranging from 8 to 20. A master-slave paradigm was implemented such that each node performed its own genetic algorithm calculation and then sent its maximum determinant value and D-optimal set of points to the master node, which selected the highest maximum determinant and stored the corresponding set of D-optimal points to be used later in the detailed aerodynamic analysis code.

Two stopping criteria were used for the GA: fixed (stopping after 1,000 generations) and variable (stopping after the maximum determinant did not change for 200 generations). The fixed stopping criterion produced the better maximum determinant values for all but one of the population sizes, and was thus chosen for use in the GA for the MDO loop. Figure 18 shows the maximum determinant values for the 28 runs using the fixed stopping criterion for each of the population sizes. Note the rather wide variation in the “optimal” GA values, which was also observed for the variable stopping criterion. It is clear from Figure 18 that aggregating the results from all 28 runs and taking the best determinant value yields a fairly stable and reliable estimate of the optimum. The important conclusions are: (1) regardless of the population size and the number of generations, the result of a single GA run is an unreliable estimate of the global optimum, (2) population size is not crucial, and (3) using the aggregate of many independent parallel GA runs *is* reliable.

For each population size, the sum of the individual node times is divided by the maximum node time to get the speedup, which is then divided by 28 to get the efficiencies shown in Figure 19. The coarse grained parallelization with the variable stopping criterion results in severe load imbalance. Fortunately, the fixed stopping criterion produces both better D-optimal point sets and better efficiencies.

7.2 AERODYNAMIC CODE — SIMPLE

The aerodynamic code performs either detailed or simple analyses at the selected points in the approximation domain. These analyses are then used to compute a least squares approximation (the response surface) to

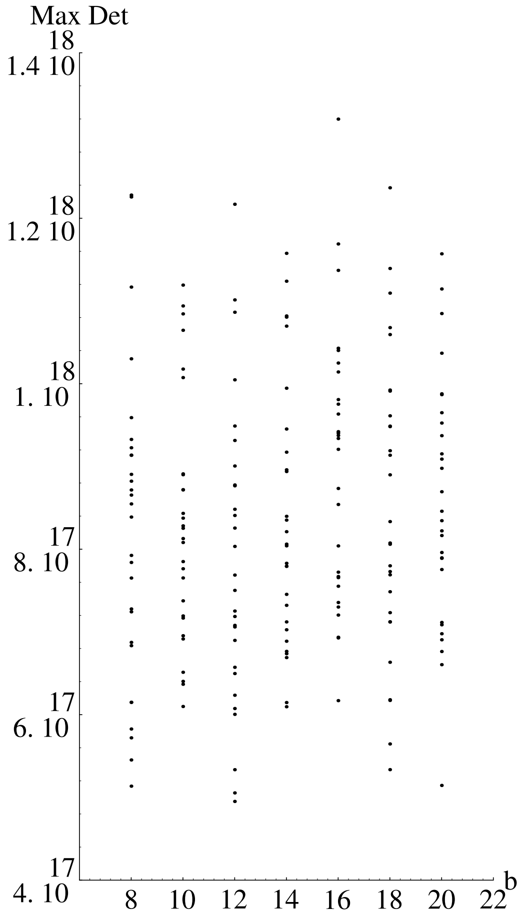

Fig. 18 Maximum determinant values versus the population size b.

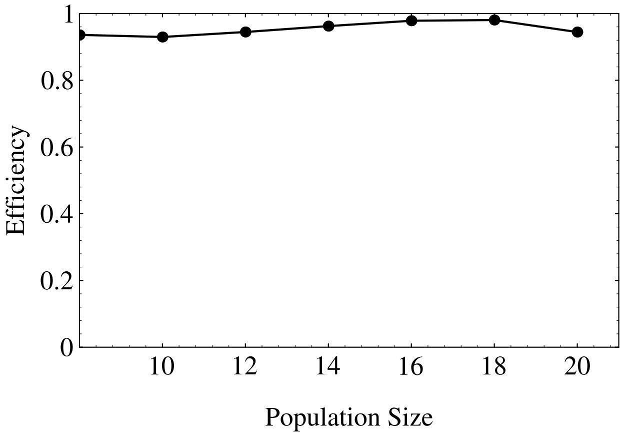

Fig. 19 Efficiency results for the GA.

some criterion function (e.g., drag), which is then used for optimization over the approximation domain. The coarse grained parallelization of the aerodynamic analysis modules within the full HSCT analysis code makes use of a master-slave paradigm on the Paragon, whereby one designated master node controls the data transfer and file input/output (I/O) of the remaining slave nodes.

The parallel versions of both the simple and detailed aerodynamic analysis codes are composed of two sections: an inherently serial portion of code at the beginning and end, and a parallelizable section which consists of computation and I/O. Since there is no communication or I/O performed in the serial portions

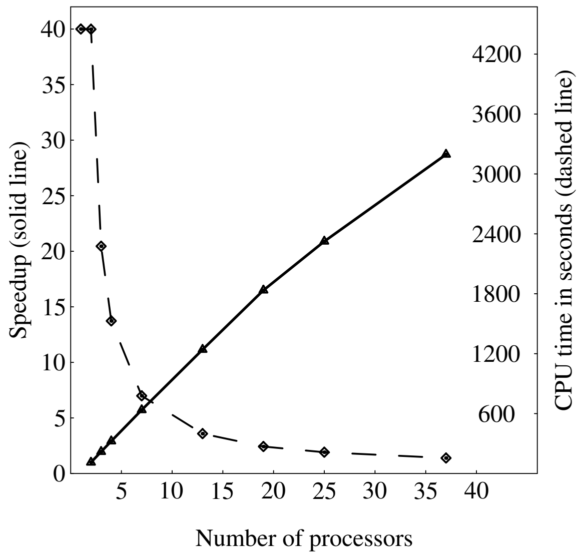

Fig. 20 Speedup results for the coarse grained parallelization of the simple aerodynamic code.

of the code, each node can perform the serial portions of the code without threat of extra overhead. The parallel multipoint analyses begin with a group of predetermined analysis points broadcast from the master node. The slave nodes then compute the subset of the points which each slave node will analyze, and then analyze their respective subsets of the selected points.

For each of the 1296 analysis points, the simple aerodynamic analysis code generates 88 pieces of output information. This large amount of I/O can substantially decrease speedup. In order to alleviate the I/O burden, the master node is designated as an I/O node, responsible for the output of the data from each node, and not computation. The master node receives the results of the point analyses from the slave nodes and immediately outputs the results to a file. Since I/O is inherently serial, it makes sense to have one node (or more, depending on the number of I/O channels the Paragon has) perform I/O, while allowing the other nodes to continue working, thereby overlapping the parallel computation time with the serial I/O time.

By using the master node as an I/O node to overlap parallel time with serial I/O time, a nontraditional model for speedup and efficiency must be used. In this model, total serial time T_1 is modeled as:

$$\begin{aligned} T_1 = &\text{ parallelizable computation (PC)} \\ &+ \text{I/O (I)} \\ &+ \text{inherently serial computation (SC)}. \end{aligned} \tag{13}$$

Using $n-1$ slave nodes, the slave time is $PC/(n-1)+SC$, and the master node time is $1296*COM+I+X$, where COM is the communication time for a single message, and X is the amount of time the master node must wait for communication to begin. The true parallel time T_n, reflecting the overlap of parallel time with serial I/O time, is then

$$T_n = \max\left\{\frac{PC}{n-1} + SC,\ 1296 * COM + I + X\right\}.$$

Normalizing T_1 to one, the efficiency ω is given by

$$\frac{1}{n\ \max\left\{\dfrac{PC}{n-1} + SC,\ 1296 * COM + I + X\right\}}. \tag{14}$$

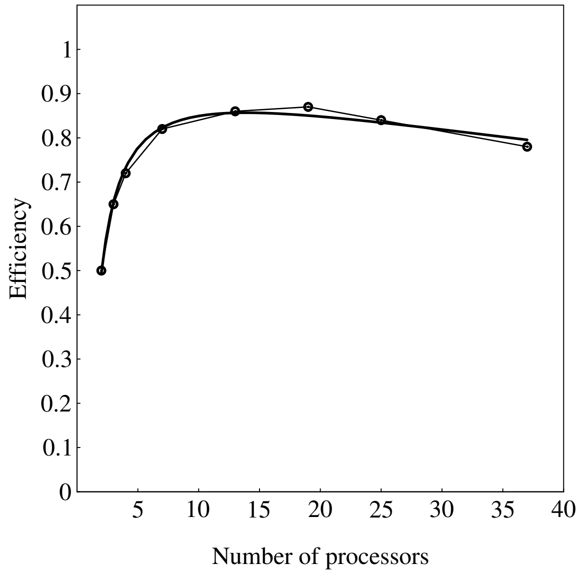

Fig. 21 Comparison of theoretical and actual efficiency for coarse grained parallelization of the simple aerodynamic code.

Table 3. Speedup and efficiency results for the simple aerodynamic analysis code.

n	time (sec)	speedup	efficiency
1	4451	1	1
2	4449	1	.5
3	2276	1.96	.65
4	1528	2.91	.72
7	778	5.72	.82
13	399	11.15	.86
19	270	16.49	.89
25	213	20.89	.84
37	155	28.71	.78

For small n, the communication and I/O term is dominated by $PC/(n-1)$, which leads to the simpler expression for efficiency:

$$\frac{1}{n\left(\dfrac{PC}{n-1}+SC\right)}. \tag{15}$$

For large n, the communication and I/O term dominates $PC/(n-1)$, and the efficiency

$$\frac{1}{n(1296*COM+I+X)} \tag{16}$$

deteriorates to zero as $n \to \infty$.

The coarse grained parallel algorithm was implemented on a 96 node Intel Paragon, on a range of 1 to 37 nodes. The CPU time, speedup, and efficiency results are given in Table 3, and shown in Figures 20 and 21. The results for the speedup are reasonable, trailing off slightly as the number of processors increases above 20. Figure 20 also provides information about the CPU time in relation to the speedup. Since n here is relatively small, the efficiency model (15) was used for a nonlinear least squares fit to the data in Table 3. Figure 21 shows excellent agreement between the data (the circles) and the model (thick solid line).

Table 4. Speedup and efficiency results for the detailed aerodynamic analysis code.

n	time (sec)	speedup	efficiency
1	743	1	1
2	403	1.84	.92
4	206	3.60	.90
5	179	4.15	.83
10	90	8.26	.83
20	45	16.51	.83

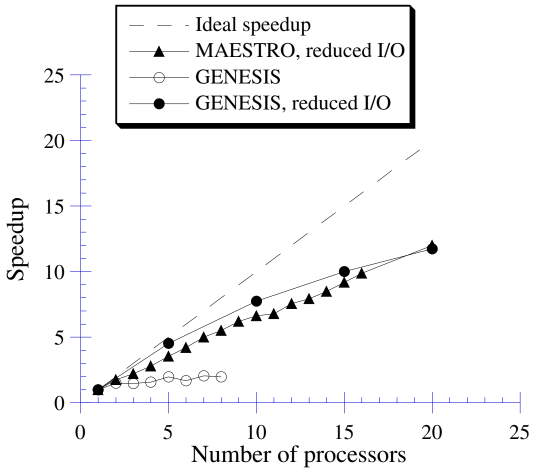

Fig. 22 Speedup results for the coarse grained parallelization of the structural code.

7.3 AERODYNAMIC CODE — DETAILED

The detailed aerodynamic analysis code uses the D-optimal analysis points (in this case, 20) determined by the simple aerodynamic analysis code to generate reduced-term polynomial response surface models for each of the expensive quantities (such as wave drag). The data needed for each analysis point is read in from a file by the master node, who then calculates each slave's subset of the points (including the master node's subset), and sends that information to each of the slave nodes. The slave nodes then do their analyses on their subset of the set of 20 analysis points, and output their own information. The small number of analyses and the significantly longer computation time for each analysis ameliorates the I/O burden encountered in the parallelization of the simple aerodynamic analysis code. Therefore, the master node does the same amount of analysis computation as each of the slaves, as well as managing the slaves.

The coarse grained parallelization of the detailed aerodynamic code was also implemented on a 96 node Intel Paragon, on a range of 1 to 20 nodes. This range was chosen because the number of nodes which can be effectively utilized in the multipoint strategy for parallelization of the detailed aerodynamic code is limited to the number of analysis points. The results are listed in Table 4.

7.4 STRUCTURAL CODE

Structural optimization requires parallelization because it is repeated many times within the overall HSCT optimization. The first step in the application of parallel computing to the HSCT structural analysis was to choose a finite element program that could be efficiently run on the Paragon. Two software packages were considered: GENESIS and MAESTRO.

GENESIS (1993) is a general purpose finite element structural optimization code. MAESTRO (1993) is a computer program for optimum design of large complex thin-walled structures, developed by Owen Hughes of Virginia Tech, and is extensively used for ship design.

Like most current finite element programs, both GENESIS and MAESTRO rely heavily on disk I/O, which limits their parallel performance. For the distributed memory architecture of the Paragon, the disk

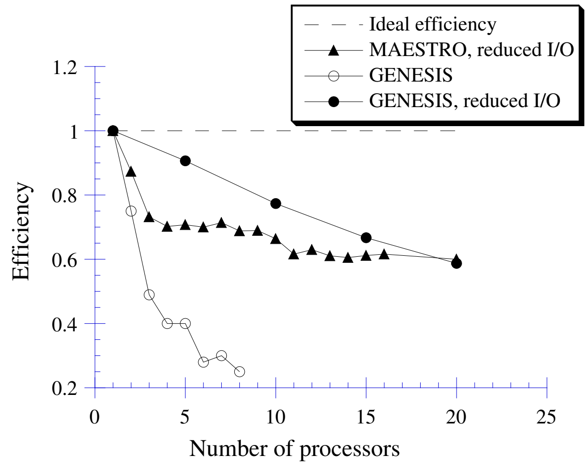

Fig. 23 Efficiency results for the coarse grained parallelization of the structural code.

I/O proved to be a bottleneck as the number of processors increased. To relieve this bottleneck, a version of GENESIS was obtained from the developer which had reduced I/O. For MAESTRO disk I/O was systematically replaced with memory usage, by converting all the temporary files into common blocks. It was necessary to add or replace about 2500 executable operators in the original MAESTRO code, requiring about 600 man-hours. However, the extensive changes in MAESTRO have not been fully debugged to date, and at present, the analysis capability of the reduced I/O version of MAESTRO has been validated, but problems remain in the optimization mode.

The effect of reducing the disk I/O in MAESTRO manifested itself even when the program was executed on a single node of the Paragon. The serial version took 501 seconds to execute a finite element analysis of the HSCT model, while the reduced I/O version required only 132 seconds for the same job.

Performance results for the coarse grained parallelization of GENESIS and MAESTRO are shown in Figures 22 and 23. With the standard version of GENESIS it was not possible to obtain speedup beyond 2.3 even with a large number of processors. The performance of the reduced I/O version is better and a speedup of 11.7 was achieved with 20 processors. Similar performance was observed for the reduced I/O version of MAESTRO.

From Figures 22 and 23 it appears that MAESTRO is expected to perform better for a larger number of nodes, but GENESIS was chosen for implementing structural optimization within the MDO loop. There are two major reasons for choosing GENESIS over MAESTRO. The primary reason for choosing GENESIS is the remaining difficulty with debugging the optimization capability in the reduced I/O version of MAESTRO. A second reason for choosing GENESIS is that both its input and output formats are similar to the I/O formats for NASTRAN, which is the most popular finite element program, and which serves as an industry standard. NASTRAN is being used to check the results of both GENESIS and MAESTRO, and the commonality of input and output between NASTRAN and GENESIS is attractive.

7.5 MDO LOOP

The automation of the MDO loop provided a significant programming challenge resulting from the need for three different “black box” engineering codes to interact, each of which was written in a different programming language. For each box of the MDO loop (Figure 16), any data required by a subsequent box was stored by a “master” node which then sent the data to the following box via message passing. The program executables were run on overlapping nodes, with each executable having a different process type in order to ensure reliability in the message passing. All executables were loaded onto a predetermined partition of nodes on the Intel Paragon at the same time using a UNIX shell script. Those boxes which needed data to begin processing simply waited for a message, relinquishing control of their processor until the message was received.

All the boxes in the main trunk in Figure 16 are done in parallel, except for the regression analyses and configuration optimization, which should consume a small fraction of the total time. The parallel structural optimizations and calculation of the weight response surface boxes of the MDO loop were not included in the automation of the MDO loop. As described in the previous section, the structural optimization code, GENESIS, has a significant I/O burden, resulting in speedup of only 11.2 on 20 nodes. Improvements must be made in this speedup for meaningful results to be obtained in the parallelization of the entire MDO loop. Work is currently underway to reduce the I/O burden incurred by GENESIS, but it will take a significant number of man hours to reduce the amount of I/O enough to affect the speedup (there are over 4500 separate I/O instances).

Table 5. Speedup results for the parallelization of the MDO loop.

n	actual speedup	potential speedup	actual efficiency	potential efficiency
1	1	1	1	1
7	5.04	6.44	.72	.92
13	7.43	11.32	.57	.87
20	9.44	17	.47	.85
37	12.04	27.86	.30	.75

Results from the parallelization of the main trunk of the MDO loop (i.e., excluding the structural portions) have been obtained for up to 37 nodes and are given in Table 5. In the parallelization of the MDO loop, a significant portion of compute time (20 minutes) is spent in the configuration optimization box of Figure 16. This serial time is the largest contributor to the speedup times, which is completely unexpected for a problem with four design variables and 66 smooth (response surface) constraints. To either parallelize the existing optimization code (a SUMT variant) or replace it with a better code will entail a substantial programming effort. Table 5 provides two values for speedup: actual speedup and potential speedup. These correspond to the speedup obtained for the actual time to run the entire loop and the speedup obtained for the time to run the MDO loop with a perfectly parallelized configuration optimization box. The actual speedup shows poor results for each value of n, even for small n. The potential speedup shows an upper limit on what is possible with a better optimization step. In order to improve the actual speedup a fine grained parallelization of the configuration optimization, the regression analysis, and the response surface boxes of Figure 16 must be performed, as the time in these serial boxes will increase at least cubically with the size of the problem.

Finally, Figure 24 shows the optimal HSCT design actually obtained by the parallel MDO loop. Slight changes in the wing shape (planform) shown in the figure result in a more aerodynamically efficient configuration, which translates to less fuel needed. The planform view of the wings does not show that the optimal wing is 22% thinner than the initial wing, which also contributes to making the optimal wing more aerodynamically efficient. The fuel weight saved by the improved aerodynamics is approximately 30,000 lb. However, the thinner optimal wing must have stronger, and thus heavier, structural support than the initial wing. The optimal wing is about 7,500 lb heavier than the initial wing in this case. The result of this tradeoff is that the optimal HSCT is about 22,500 lb lighter than the initial HSCT. For the record, the initial and optimal values of (c_r, c_t, Λ_{LE}, ζ) are (150.0, 7.0, 72.5°, 1.1) and (177.0, 7.1, 73.3°, 0.86), respectively, and the fuel weight changed from 340,000 lb to 309,379 lb.

8 Summary

A parallel variable-complexity multidisciplinary optimization paradigm for the particular instance of High Speed Civil Transport design on an Intel Paragon has been presented in this paper. Beyond the conceptual question of how best to accomplish MDO, the practical implementation of MDO requires the integration of heterogeneous, mismatched disciplinary codes. It is interesting that many of the features of existing software artifacts that make MDO difficult are precisely the same features that make parallelization difficult. These features include

- use of disk as extended memory,
- interactive code ill-suited for inclusion in a larger code,
- nonstandard programming constructs,
- dynamic creation of noncontiguous data structures,
- mismatched data structures for subtasks.

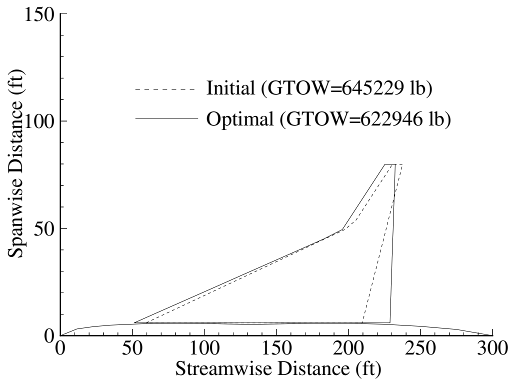

Fig. 24 Initial and optimal HSCT wing planforms.

Nonstandard constructs include obsolete and nonANSI standard vendor specific features of programming languages, and idiosyncratic programming styles. When data structures contain pointers to dynamically allocated memory, passing the structure in a message is meaningless because the pointers are invalid for the receiver's memory. Unravelling a dynamic tree data structure in a message passing environment is a programming nightmare. All the aerodynamic codes use a Craidon description of the aircraft geometry, which is qualitatively different from the finite element description required by the structures codes, and control and manufacture analyses require yet different representations. Translating between representations is a source of nonsmoothness, approximation error, and inefficiency.

Results from the coarse grained parallelization of the detailed and simple aerodynamic codes, as well as the genetic algorithm code, have shown reasonable speedups and efficiencies. The coarse grained parallelization of the structural code requires a greater reduction in I/O in order to attain reasonable speedups, which will require many man hours. Results from the parallelization of the entire MDO loop are poor, but the potential speedup is reasonable. This shows the need for a more fine grained parallelization of those codes (particularly the configuration optimization) which are currently executed serially.

ACKNOWLEDGEMENT

Use of the Intel Paragons at the NASA Langley Research Center and at Oak Ridge National Laboratory is gratefully acknowledged, as are comments from the referees.

BIOGRAPHIES

Susan L. Burgee is a Research Assistant in the Department of Computer Science at Virginia Polytechnic Institute and State University, Blacksburg. Her research interests include parallel computation and human computer interaction. She received a B.S. degree (magna cum laude) in Computer Science and Mathematics from the College of Notre Dame of Maryland, Baltimore, MD, in 1993 and is currently pursuing a M.S. in Computer Science at Virginia Polytechnic Institute and State University.

Anthony A. Giunta is a Research Assistant in the Department of Aerospace and Ocean Engineering at Virginia Polytechnic Institute and State University (Virginia Tech) in Blacksburg. He received the B.S. degree (magna cum laude) in Aerospace Engineering in 1993 from Virginia Tech and he is currently pursuing a Ph.D. degree in the same field. His research interests include multidisciplinary design optimization of flight vehicles and parallel computing.

Vladimir Balabanov is a Research Assistant in the Aerospace and Ocean Engineering Department at Virginia Polytechnic Institute and State University, Blacksburg. His current research interests include structural analysis and optimization, parallel computation and response surface design and analysis. He received a M.S. degree in Aerospace Engineering from the Moscow Institute of Physics and Technology (MFTI), Moscow, Russia in 1989. He has worked for the Central Aerohydrodynamic Institute (TsAGI), Moscow, Russia and is currently pursuing a Ph.D. degree in Aerospace Engineering at Virginia Polytechnic Institute and State University.

Bernard Grossman is presently Professor and the Department Head of Aerospace and Ocean Engineering at Virginia Tech. He also serves as the Director of the Multidisciplinary Analysis and Design (MAD) Center for Advanced Vehicles. His research interests include computational fluid dynamics, high-speed aerodynamics, theoretical fluid dynamics, chemically reacting flows and multidisciplinary design optimization. He is presently an Associate Editor of the AIAA Journal and a member of the AIAA Fluid Dynamics Technical Committee. He as been at Virginia Tech since January 1982. Previous to coming to Virginia Tech, spent 12 years at Grumman Aerospace Corp., where his last position was the Head of the Theoretical Aerodynamics Laboratory. All his degrees are from the Polytechnic Institute of Brooklyn, with a B. S. (Aerospace Eng.) in 1964, a M. S. (Astronautics) in 1965, and a Ph.D. (Astronautics) in 1969.

William H. Mason received his degrees from Virginia Tech, obtaining the B.S in 1970, the M.S. in 1971 and a Ph.D. in 1974, all in Aerospace Engineering. He worked for the Grumman Aerospace Corporation for 15 years before joining the faculty of the Aerospace and Ocean Engineering at Virginia Tech in 1989, where he is a professor. At Grumman he worked in aerodynamic methodology development and aerodynamic design. His last five years were spent in the advanced systems area working on the conceptual and preliminary design of proposed new aircraft. His research work is in design methodology and applied aerodynamics.

Robert Narducci received a dual B.S. degree (cum laude) in Aerospace and Ocean engineering from Virginia Tech in 1991. While working towards his B.S., he enrolled in the cooperative education program where he worked as a project engineer at the Naval Air Test Center Patuxent River, MD. In 1995, he received a Ph.D. in Aerospace Engineering from Virginia Tech. Robert currently works for McDonnell Douglas Aerospace, Long Beach, CA.

Raphael T. Haftka received a BSAE and MSAE from the Technion, Israel Institute of Technology, a Ph.D in Aerospace Engineering from the University of California at San Diego, and did Post Doctoral research as the NASA Langley Research Center.

He worked for the Israeli Aircraft Industries and for Structures Research Associates. He served on the faculties of the Technion, Israel, Illinois Institute of Technology and the Virginia Polytechnic Institute and State University, before coming to the University of Florida, where he is currently a professor of aerospace engineering, mechanics and engineering science. His current research interests include structural and multidisciplinary optimization.

Professor Haftka is president of the International Society for Structural and Multidisciplinary Optimization.

Layne T. Watson received the B. A. degree (magna cum laude) in psychology and mathematics from the University of Evansville, Evansville, IN, in 1969 and the Ph.D. degree in mathematics from the University of Michigan, Ann Arbor, in 1974.

He has worked for USNAD Crane, Sandia National Laboratories, and General Motors Research Laboratories and has served on the faculties of the University of Michigan and Michigan State University before coming to Virginia Polytechnic Institute and State University, Blacksburg, where he is currently a professor of computer science and mathematics. His current research interests include fluid dynamics, structural mechanics, homotopy algorithms, parallel computation, mathematical software, and image processing.

REFERENCES

Bertin, J. J., and Smith, M. L. 1989. *Aerodynamics for Engineering Students*, 2nd Ed. Englewood Cliffs: Prentice Hall. pp. 261–288.

Box, M. J., and Draper, N. R. 1971. Factorial Designs, the $|\mathbf{X}^T\mathbf{X}|$ Criterion, and Some Related Matters. *Technometrics* 13(4): 731–742.

Braun, R., and Kroo, I. 1995. Use of Collaborative Optimization in a Multidisciplinary Design Environment. *SciCADE '95.*

Braun, R., and Kroo, I. 1995. A Collaborative Approach to Trajectory Optimization. In *Proc. ICASE/Langley Workshop on Multidisciplinary Optimization.* Philadelphia: SIAM.

Burgee, S. L., Watson, L. T., Giunta, A. A., Grossman, B., Haftka, R. T., and Mason, W. H. 1994. Parallel Multipoint Variable-Complexity Approximations for Multidisciplinary Optimization. In *Proc. IEEE Scalable High-Performance Computing Conference.* Los Alamitos: IEEE, pp. 734-740.

Burgee, S. L., Giunta, A. A., Narducci, R., Watson, L. T., Grossman, B., and Haftka, R. T. 1995. A coarse grained variable-complexity approach to MDO for HSCT design. In *Parallel Processing for Scientific Computing*, D. H. Bailey, P. E. Bjørstad, J. R. Gilbert, M. V. Mascagni, R. S. Schreiber, H. D. Simon, V. J. Torczon, and L. T. Watson (eds.). Philadelphia: SIAM, pp. 96–101.

Byun, C., and Guruswamy, G. P. 1994. Wing-Body Aeroelasticity Using Finite-Difference Fluid/Finite-Element Structural Equations on Parallel Computers. AIAA Paper 94–1487.

Byun, C., and Guruswamy, G. P. 1995. An Efficient Procedure for Flutter Boundary Prediction on Parallel Computers. AIAA Paper 95–1484.

Carlson, H. W., and Miller, D. S. 1974. Numerical Methods for the Design and Analysis of Wings at Supersonic Speeds. NASA TN D-7713.

Carlson, H. W., and Mack, R. J. 1978. Estimation of Leading-Edge Thrust for Supersonic Wings of Arbitrary Planforms. NASA TP-1270.

Carlson, H. W., Mack, R. J., and Barger, R. L. 1979. Estimation of Attainable Leading-Edge Thrust for Wings at Subsonic and Supersonic Speeds. NASA TP-1500.

Craidon, C. B. 1970. Description of a Digital Computer Program for Airplane Configuration Plots. NASA TM X-2074.

Cramer, E. J., Dennis, J. R., Jr., Frank, P. D., Lewis, R. M., and Shubin, G. R. 1994. Problem Formulation for Multidisciplinary Optimization. *SIAM J. Optimization* 4(4):754–776.

Dudley, J., Huang, X., MacMillin, P. E., Grossman, B., Haftka, R. T., and Mason, W. H. 1995. Multidisciplinary Optimization of the High-Speed Civil Transport. AIAA Paper 95–0124.

Furuya, H. and Haftka, R. T. 1993. Locating Actuators for Vibration Suppression on Space Trusses by Genetic Algorithms. In *Structures and Controls Optimization*, AD-Vol. 38, N.S. Khot and R.T. Haftka, (eds.). New York: ASME, pp. 1–12.

GENESIS. 1993. *GENESIS User Manual*, Version 1.3. Goleta, CA: Vanderplaats, Miura and Associates.

Giunta, A.A., Dudley, J.M., Narducci, R., Grossman, B., Haftka, R.T., Mason, W.H., and Watson, L.T. 1994. Noisy Aerodynamic Response and Smooth Approximation in HSCT Design. In *Proc. 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 94-4376. Panama City Beach, FL: pp. 1117–1128.

Guruswamy, G. P., and Byun, C. 1993. Fluid-Structural Interactions Using Navier-Stokes Flow Equations with Shell Finite-Element Structures. AIAA Paper 93–3087.

Harris, R. V., Jr. 1964. An Analysis and Correlation of Aircraft Wave Drag. NASA TM X-947.

Hutchison, M. G., Huang, X. M., Mason, W. H., Haftka, R. T., and Grossman, B. 1992. Variable-Complexity Aerodynamic-Structural Design of a High-Speed Civil Transport Wing. AIAA Paper 92-4695.

Hutchison, M. G., Unger, E. R., Mason, W. H., Grossman, B., and Haftka, R. T. 1993. Aerodynamic Optimization of an HSCT Wing Using Variable-Complexity Modeling. AIAA Paper 93-0101.

Hutchison, M. G. 1993. Multidisciplinary Optimization of High-Speed Civil Transport Configurations Using Variable-Complexity Modeling. Ph.D. Dissertation, Blacksburg, VA: VPI&SU.

Hutchison, M. G., Unger, E. R., Mason, W. H., Grossman, B., and Haftka, R. T. 1994. Variable-Complexity Aerodynamic Optimization of an HSCT Wing Using Structural Wing-Weight Equations. *J. Aircraft* 31(1): 110–116.

Kay, J., Mason, W. H., Durham, W., and Lutze, F. 1993. Control Authority Assessment in Aircraft Conceptual Design. AIAA Paper 93-3968.

Khuri, A. I., and Cornell, J. A. 1987. *Response Surfaces: Designs and Analyses*. New York: Marcel Dekker.

Kroo, et al. 1995. MDO for Large-Scale Aerospace Design. In *Proc. ICASE/Langley Workshop on Multidisciplinary Optimization*. Philadelphia: SIAM.

MAESTRO. 1993. *MAESTRO Basic Features Manual*, Version 6.2. Stevensville, MD: Ross-McNatt Naval Architects.

Mainguy, Y., Birch, J. B., and Watson, L. T. 1995. A Robust Variable Order Facet Model for Image Data. *Machine Vision Appl.* 8: 141–162.

Mason, R. L., Gunst, R. F., and Hess, J. L. 1989. *Statistical Design and Analysis of Experiments*, New York: John Wiley & Sons, pp. 215–221.

Narducci, R., Grossman, B., and Haftka, R. T. 1994. Sensitivity Algorithms for an Inverse Design Problem Involving a Shock Wave. AIAA Paper 94-0096.

Renaud, J. E., Gabriele, G. A. 1991. Sequential Global Approximation in Non-Hierarchic System Decomposition and Optimization. *Advances in Design Automation* 1:191–200.

Shankar, J., Ribbens, C. J., Haftka, R. T., Watson, L. T. 1993. Computational Study of a Nonhierarchical Decomposition Algorithm. *Computational Optimization and Applications* 2:273–293.

Sobieszczanski-Sobieski, J. 1988. Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization.* Hampton, VA.

Unger, E. R., Hutchison, M. G., Rais-Rohani, M., Haftka, R. T., and Grossman, B. 1992. Variable-Complexity Design of a Transport Wing. *Intl. J. Systems Automation: Res. and Appl. (SARA)* 2:87–113.

APPENDIX A

Table A1. Design variables and baseline values.

#	Value	Description
1	150.00	Wing root chord (ft)
2	142.93	LE break, x (ft)
3	45.07	LE break, y (ft)
4	173.89	TE break, x (ft)
5	63.11	TE break, y (ft)
6	170.97	LE of wing tip, x
7	7.00	Tip chord (ft)
8	73.88	Wing semi-span (ft)
9	0.40	Chordwise location of max. t/c
10	2.63	Airfoil LE radius parameter, r_t
11	3.10	Airfoil t/c at root (%)
12	2.77	Airfoil t/c at LE break (%)
13	1.92	Airfoil t/c at tip (%)
14	2.00	Fuselage restraint 1, x (ft)
15	0.83	Fuselage restraint 1, r (ft)
16	12.09	Fuselage restraint 2, x (ft)
17	3.25	Fuselage restraint 2, r (ft)
18	132.86	Fuselage restraint 3, x (ft)
19	5.29	Fuselage restraint 3, r (ft)
20	249.36	Fuselage restraint 4, x (ft)
21	4.43	Fuselage restraint 4, r (ft)
22	20.92	Nacelle 1, x_9 (ft)
23	27.68	Nacelle 2, x_{10} (ft)
24	340,000	Mission fuel (lbs)
25	70.00	Starting cruise altitude (ft)
26	0.00	Cruise climb rate (ft/min)
27	690.00	Vertical tail area (ft^2)
28	713.00	Horizontal tail area (ft^2)

Table A2. Optimization Constraints.

#	Description
1	Range $\geq 5,500$ n. mi.
2	Landing angle of attack $\leq 12°$
3	Landing CL ≤ 1.0
4-21	Landing section Cl ≤ 2.0
22	Fuel volume $\leq$ half of wing volume
23-40	Wing chord $\geq$ 7.0ft
41	LE break, $y \leq$ wing semi-span
42	TE break, $y \leq$ wing semi-span
43	Root $t/c \geq 1.5\%$
44	LE break $t/c \geq 1.5\%$
45	TE break $t/c \geq 1.5\%$
46	Fuselage: $x_{rest_1} \geq 5ft$
47	Fuselage: $x_{rest_1} + 10\text{ft} \leq x_{rest_2}$
48	Fuselage: $x_{rest_2} + 10\text{ft} \leq x_{rest_3}$
49	Fuselage: $x_{rest_3} + 10\text{ft} \leq x_{rest_4}$
50	Fuselage: $x_{rest_4} + 10\text{ft} \leq 300\text{ft}$
51	Nacelle 1, $y \geq$ side-of-body
52	Nacelle 1, $y \leq$ nacelle 2, y
53	Engine out stability criterion
54	Minimum airfoil section spacing at wing tip
55-56	No engine scrape at landing angle-of-attack
57-58	No engine scrape at landing angle-of-attack, with 5° roll
59	No wing tip scrape at landing angle-of-attack, with 5° roll
60	No wing trailing-edge break point scrape at landing, with 5° roll
61	Crosswind landing capability with aileron deflection only
62	Crosswind landing capability with aileron and rudder deflection
63	Tail deflection $\leq 22.5°$ for landing
64	Takeoff rotation must occur prior to reaching 90% of takeoff velocity
65	No negative sweep of wing inboard trailing-edge
66	Root trailing-edge must not overlap root leading-edge of horizontal tail

APPENDIX B

From the residual data the average error, RMS error, and maximum error of the residual are calculated to evaluate the quality of the response surface fits (Tables B1, B2, B3). Here, m is the number of data points on which the response surface calculation is based. The data in the tables represents the average of three trials for each value of m.

The errors associated with the bilinear tensor product response surfaces are given in Table B1 and show that the error for the D-optimal points decreases only slightly as the number of points ranges from four to twenty. This is to be expected since the D-optimality criterion specifies points on the perimeter of the design

Table B1. Calculated error for the bilinear tensor product response surface fit.
*Average of three trials for each value of m.

m	Avg. Error	RMS Error	Max. Error
D-optimal Points			
4	0.00413	0.00502	0.01490
5	0.00403	0.00492	0.01468
10	0.00381	0.00470	0.01416
15	0.00409	0.00496	0.01502
20	0.00378	0.00464	0.01414
Random Points*			
4	0.06568	0.09958	0.34724
5	0.00506	0.00685	0.02310
10	0.00278	0.00355	0.01171
15	0.00260	0.00339	0.01171
20	0.00268	0.00347	0.01222
CCD Points			
9	0.00587	0.00662	0.01579

Table B2. Calculated error for the quadratic polynomial response surface fit.
*Average of three trials for each value of m.

m	Avg. Error	RMS Error	Max. Error
D-optimal Points			
6	0.00189	0.00247	0.00933
10	0.00215	0.00284	0.01102
20	0.00192	0.00252	0.01034
30	0.00192	0.00234	0.00855
40	0.00193	0.00244	0.00892
50	0.00211	0.00258	0.00779
Random Points*			
6	0.02178	0.03308	0.10160
10	0.00270	0.00357	0.01198
20	0.00227	0.00283	0.00906
30	0.00197	0.00251	0.00925
40	0.00193	0.00247	0.00906
50	0.00191	0.00240	0.00881
CCD Points			
9	0.00270	0.00331	0.00972

space when bilinear response surface functions are used to form $\mathbf{X}$ in equation (9). Thus, the points on the perimeter are effective in capturing the overall trend of the noisy surface, but adding additional points on the perimeter does little to change the overall quality of the fit. In contrast to the D-optimal error, the error for the randomly selected data points decreases considerably as m increases. This trend also is expected since response surfaces constructed using points mainly in the interior of the design space are more likely to be affected by the noise in the computed data. As the number of randomly selected points increases, the effects of the noise are canceled out and the surface fits improve. This explains why the error for the randomly selected points is lower than the errors for the D-optimal points when more than ten points are used to construct the response surfaces. The calculated errors for the nine CCD points are worse than those for the randomly selected points and for the D-optimal points. For the relatively simple bilinear tensor product model, all three point selection techniques provide point distributions which result in sufficiently accurate response surfaces comparable to that shown in Figure 11.

Table B3. Calculated error for the biquadratic tensor product response surface fit.
*Average of five trials for each value of m.

m	Avg. Error	RMS Error	Max. Error
D-optimal Points			
9	0.00199	0.00259	0.01044
10	0.00201	0.00263	0.00996
20	0.00202	0.00251	0.00774
30	0.00199	0.00240	0.00810
40	0.00183	0.00232	0.00823
50	0.00184	0.00233	0.00893
60	0.00189	0.00234	0.00782
70	0.00188	0.00240	0.00875
80	0.00182	0.00231	0.00907
90	0.00181	0.00231	0.00900
100	0.00180	0.00226	0.00855
Random Points*			
9	0.04653	0.08443	0.51657
10	0.01706	0.02994	0.14234
20	0.00300	0.00414	0.01657
30	0.00251	0.00363	0.01793
40	0.00219	0.00292	0.01172
50	0.00200	0.00253	0.00995
60	0.00193	0.00241	0.00869
70	0.00192	0.00244	0.00868
80	0.00185	0.00234	0.00878
90	0.00184	0.00231	0.00889
100	0.00186	0.00231	0.00845
CCD Points			
9	0.00310	0.00399	0.01262

Similar trends for the D-optimal and random points again occur in the error for the quadratic polynomial response surface as m increases from six to fifty points (Table B2). However, the error achieved using ten D-optimal points is comparable to that attained using twenty randomly selected points. Further, all response surfaces produced from the D-optimal points are similar to the quadratic response surface shown in Figure 11. In contrast, the response surfaces produced by the randomly selected points are significantly different from Figure 11 until at least twenty randomly selected points are used. The RMS error from the response surface fit for the CCD points is comparable to that of the D-optimal and randomly selected points. For the quadratic polynomial response surface the D-optimal and CCD point selection strategies require fewer points to accurately model the noisy surface and therefore are superior to the random point selection scheme.

The error for the biquadratic tensor product response surfaces is shown in Table B3 for the three point selection techniques. As demonstrated with the previous response surface functions, the error for the D-optimal points varies little as m increases and the response surfaces are qualitatively in good agreement with Figure 12. Approximately fifty randomly selected points are required to provide the same quality of fit as ten D-optimal points. This further demonstrates that random point selection is not an efficient point selection method for response surface formation. The CCD points provide a quality of fit that is somewhat less accurate than that from the D-optimal points, and the response surface based on the CCD points does not qualitatively agree well with Figure 12. For the biquadratic tensor product, the D-optimal points provide the most accurate fit to the noisy surface, while the CCD points provide an adequate fit. The random point selection technique is clearly unacceptable, unless a much larger number of points is used for the fit.