THE PROMISE (AND REALITY) OF MULTIDISCIPLINARY DESIGN OPTIMIZATION[†]

SUSAN L. BURGEE[*] AND LAYNE T. WATSON[*]

Abstract. 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. This paper discusses the obstacles to MDO, and presents a parallel MDO paradigm using variable-complexity modeling and multipoint response surface approximations 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.

Keywords. aircraft design, optimal design, parallel computation.

AMS (MOS) classification. 65Y05, 65Y15, 90C06, 90C30, 90C90.

1. The promise. The design of modern aerospace vehicles involves multidisciplinary interactions and presents a formidable challenge to the designer in a competitive marketplace. Multidisciplinary optimization (MDO) technology is a major tool for addressing this need, but its current use in the design process is limited by its enormous computational burden. To address this challenge, Unger et al. recently developed a variable-complexity modeling approach, involving the use of refined and computationally expensive models together with simple and computationally inexpensive models [29]. This approach was applied to the combined aerodynamic-structural optimization of transport wings and focuses on the aerodynamic-structural optimization of the High Speed Civil Transport (HSCT) [18].

The goal of this paper is to emphasize the practical difficulties of MDO, and to present a parallel variable-complexity MDO paradigm as applied to the aerodynamic and structural design of the HSCT. In this multipoint approximation method, the design space is limited through the use of simple, less computationally expensive analyses for the evaluation of both the objective function and constraints for numerous HSCT configurations. Here the simple analyses are used to screen out HSCT designs which are not close to a possible optimum design. A genetic algorithm then is used to determine a set of points within the limited (nonconvex) approximation

[†] This work was supported by NASA Grant NAG-1-1562.

[*] Multidisciplinary Analysis and Design Center for Advanced Vehicles, Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0106.

domain which satisfy the D-optimality criterion. For these points, the refined, more computationally expensive analyses are performed to provide a more accurate assessment of the merit of the selected HSCT configurations. Prior to the use of the refined analyses, response surfaces are constructed and analyzed statistically to determine the structure of smooth approximations to the aerodynamic drag calculations and to the structural weight estimations.

A coarse grained parallelization of both the simple and refined analysis codes used for HSCT design was implemented to reduce the time required to evaluate the numerous HSCT configurations. Results on an Intel Paragon have shown reasonable speedup and efficiency. This paper presents an HSCT design problem involving four variables and the multipoint approximation technique previously used by Giunta et al. [15], speedup and efficiency results obtained for the parallelization of both the simple and refined aerodynamic analysis codes, and parallelization results for two structural analysis codes.

Because an integrated design should produce a global optimum for the entire system rather than merely a feasible union of optimized subsystems, MDO should result in overall better products. The economic and marketing advantages of a shorter design cycle are clear, and perhaps the purported ability of MDO to enhance economic competitiveness is largely responsible for its popularity. The massive computational requirements of realistic MDO are also clear, as is parallelism, achievable in several different organizations of the computation and at different levels of granularity. In summary, the promises of MDO are

- better products,
- to enhance economic competitiveness, and
- to leverage parallel supercomputer power.

2. The reality. The obstacles to MDO becoming a production methodology, in the same sense as quality control, are numerous and formidable. In aircraft design, for instance, typical disciplines involved would be aerodynamics, structures, thermodynamics, controls, propulsion, manufacture, and economics. Detailed analyses in each of these disciplines could involve tens to hundreds of subroutines and tens of thousands of lines of code. Managing the software libraries and data alone is a daunting task.

Codes from different disciplines typically are grossly incompatible, but even within disciplines, data structures and solution representations may be incompatible, requiring "translation" routines or recoding. This incompatibility is particularly acute when stand-alone packages with interactive interfaces are involved. As an example, the Craidon geometry wing representation for aerodynamic codes is quite different from a NASTRAN-like structural representation of a wing. Most disciplinary codes, designed years ago for small serial computers, are very ill-suited to modern parallel architectures, even with a coarse grained approach.

The accuracy of physical models is always an issue, particularly when new materials (e.g., composites or plastics) or new construction techniques (e.g., adhesive bonding) are employed. Clearly the validity of the optimal design depends on the validity of the underlying physical models.

Stand-alone disciplinary analysis packages with rich, interactive operating environments are almost surely (in the technical sense of these words, also) useless in the MDO context. Integrating such packages into a single parallel MDO program is extremely difficult, and likely to result in considerable inefficiency. The overhead time lost in unnecessary file I/O and translating between different solution representations is *not* insignificant.

Physical quantities governed by differential equations or implicitly determined are subject to numerical and discretization errors, resulting in very noisy (but as accurate as the physical constants involved) function values. This invalidates or foils many numerical algorithms, and requires special handling.

Detailed, highly accurate disciplinary analyses are very expensive, requiring sometimes hours on a supercomputer, even when run in parallel. The import of this is that, regardless of the dimension of the design space, it can be sampled for accurate function values at only a small number of points (order of 10 but not order 10^2). Clearly current optimization algorithms will not directly apply. This high cost for accurate function values is the motivation for the variable complexity modeling approach, which attempts to match the cost of information accuracy with the usefulness of that accuracy.

Several current approaches to MDO involve a decomposition across disciplines, whereby disciplinary optimizations are performed simultaneously and independently, guided by a "cooperative" or "collaborative" global phase [2], [3], [10], [22], [28]. While this paradigm is intuitively appealing and a direct descendant of the manual, sequential interdisciplinary design process, there is no theoretical proof of convergence. Counterexamples are known for some of these approaches [27], but are difficult to construct because of the complexity of the MDO schemes. A careful study of the nature of the counterexamples suggests that any approach based on independent subsystem optimizations is not provably convergent; for this reason the MDO paradigm proposed here uses pipelined rather than concurrent optimizations. Regardless, there is an urgent need for convergence theory for MDO algorithms.

In summary, the obstacles to achieving true MDO include

- sheer number and complexity of codes,
- incompatibility of codes with each other and with parallel computers,
- model verification,
- stand-alone interactive disciplinary analysis packages,
- noisy function values,
- high cost of disciplinary analyses,
- flawed parallel optimization methodologies.

The remaining sections in this paper discuss MDO at Virginia Tech for HSCT design, a core project of the Multidisciplinary Analysis and Design Center for Advanced Vehicles. Not all of the obstacles enumerated above are being addressed (e.g., software engineering tools to handle the subroutine and data libraries, or experimental work to validate the models), but some (for instance, noisy function values, expensive analysis values, parallel optimization algorithms) are being addressed in substantive, deliberate ways. Sections 3–8 summarize the HSCT mission, some model problems, and some attacks on the obstacles mentioned above. Sections 9–11 describe the concept of D-optimality, and a practical approach to it based on genetic algorithms. Section 12 presents, in some detail, a new parallel MDO paradigm that is provably convergent. Sections 13–15 give parallelization results, and Section 16 concludes.

3. HSCT configuration and mission. Successful aircraft configuration optimization requires a simple yet meaningful mathematical characterization of the geometry. The model used here completely defines the wing-body-tail configuration, using twenty-eight design variables. The wing planform is described by eight design variables, and the airfoil thickness distribution by an additional five. The engine nacelles move axially with the trailing-edge of the wing, and two parameters define their spanwise locations. The axisymmetric fuselage requires eight parameters to specify both the axial positions and radii of the four fuselage restraint locations. Two additional variables describe the size of the horizontal and vertical stabilizers. Details of the geometry specification appear in [17] and [20]. 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 [9].

The optimization problem is to minimize the takeoff 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. Fifty-five constraints which include performance/aerodynamic constraints and geometric constraints, such as fuselage volume and tail scrape angle, prevent the optimizer from creating physically improbable designs [17].

4. Four design variable problem. The HSCT wing planform is parameterized using eight design variables x_1–x_8 as shown in Figure 1. 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

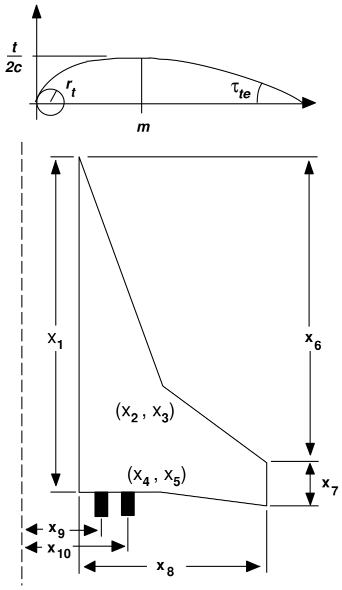

FIGURE 1. *Airfoil thickness distribution parameters (top) and wing planform and nacelle placement parameters (bottom).*

few design variables, say four: two of the original planform variables, root chord and tip chord, along with two new design variables (Fig. 2). 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 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 example problem considered here, the aerodynamics of the wing tip

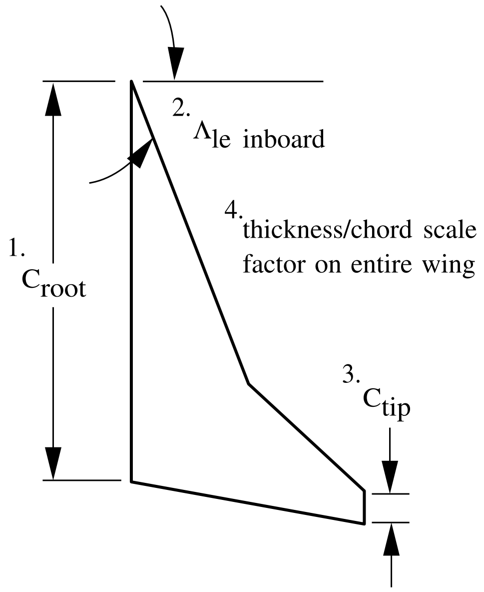

FIGURE 2. *Wing design variable definition for the four variable problem.*

region can strongly influence the design of a particular aircraft. Therefore, the tip chord is not a design variable that can be ignored.

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 $\sqrt{x_2^2 + x_3^2}$ is fixed at the baseline value, so that the inboard leading edge sweep angle alone, with the tip cord 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.

5. Variable-complexity modeling. 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. 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. 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 [20].

The detailed aerodynamic analysis code utilizes the Harris program [16] for the supersonic volumetric wave drag, a Mach-box [7] type method for supersonic drag due to lift, and a vortex-lattice program for landing performance. When compared to the computational costs of current 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 quickly becomes substantial. Therefore, a simple, less computationally intensive analysis in a variable-complexity modeling approach [18] is employed. In earlier work, this methodology was used within a sequential approximate optimization technique whereby the overall design process was composed of a sequence of optimization cycles. At the beginning of each cycle, approximations to the wave drag and drag due to lift were constructed using either linear, scaled, or global-local approximations [18]. The scaled approximation method was utilized in this study and employed a constant scaling function, σ, given as $\sigma(x_0) = f_d(x_0)/f_s(x_0)$, 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, $f(x)$, were calculated as $f(x) = \sigma(x_0)f_s(x)$. Move limits were then imposed on the design variables to avoid large errors.

6. Multipoint approximation. This section focuses on the use of multipoint approximations to fit response surfaces to the noisy objective function and constraints of the example problem. The efficient construction of a response surface requires several steps:

(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.

The term *multipoint* refers to the practice of concurrently analyzing many different HSCT configurations. The multipoint analyses yield a database containing both the independent design variables and objective function and constraint data. *Response surfaces* are smooth functions

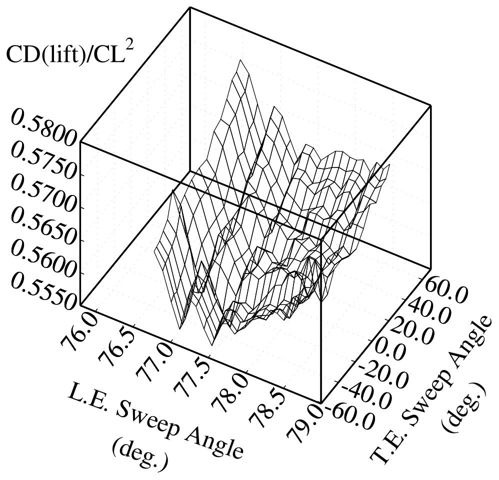

FIGURE 3. *Noisy drag due to lift objective function data in the design space.*

which define the relationship between the independent and dependent variables [21].

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 (Fig. 3), 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 objective function.

Because of the computational costs associated with evaluating the large number of point designs, the present work focuses on methods to determine the minimum number of calculations required to construct various response surfaces. This section discusses the situation with two design variables motivated by Figure 3.

Response surface approximations are constructed in [14] using both polynomial functions and rational functions. The bilinear tensor product model has the form

$$f(x,y) = \bigl(a_1x + a_2\bigr)\bigl(a_3y + a_4\bigr).$$

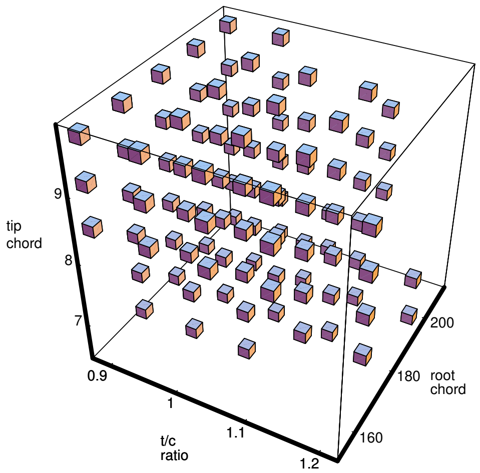

FIGURE 4. *Approximation domain projected along inboard leading edge sweep angle axis.*

Also employed by [14]were the quadratic polynomial

$$f(x, y) = a_1x^2 + a_2xy + a_3x + a_4y^2 + a_5y + a_6$$

and the biquadratic tensor product

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

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

$$f(x, y) = (a_1x + a_2y + a_3)/(a_4x + a_5y + a_6)$$

was utilized. For the bilinear tensor product, quadratic polynomial, and biquadratic tensor product the response surfaces were found by performing a linear least squares fit to the drag due to lift data. For the rational function approximation, a Levenberg-Marquardt nonlinear least squares fitting routine was used. Pictures of the response surfaces and a discussion of the results are in [14].

7. Approximation domain definition. The approximation domain is constructed around a nominal design point, usually obtained from the previous approximate optimization cycle. The domain is bound 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.

Using the constraint data obtained from simple analyses for each of the 1296 HSCT designs in a $6 \times 6 \times 6 \times 6$ 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 ≥ 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. Figure 4 shows a projection of the approximation domain for the four design variable problem along the inboard leading edge sweep angle axis.

8. 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.

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

$$1, x_1, x_2, \ldots, x_p, x_1^2, x_1x_2, \ldots, x_p^2,$$

corresponding to a quadratic model. The regression model is

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

which leads to the overdetermined matrix problem

$$Y \approx \mathbf{X}c, \tag{8.1}$$

where $Y = (Y_1, ..., Y_m)$, $c = (c_1, ..., c_n)$, and $\mathbf{X}$ is the $m \times n$ $(m \geq n)$ 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 $||Y - \mathbf{X}c||_2$. The regression or response surface model of the data is then taken as

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

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

$$\Upsilon = \begin{cases} \dfrac{100|\sigma_i|\sqrt{\mathrm{Var}(\epsilon)}}{c_i}, & c_i \neq 0, \\ 100, & c_i = 0, \end{cases} \tag{8.2}$$

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. Terms in the polynomial model having large coefficients of variation, typically over 0.10, 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. The higher order terms involving c_{tip} have coefficients of variation greater than ten percent and can safely be dropped from the response surface model. Thus, the number of terms

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

TABLE 1. *Calculated errors.*

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 1 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.

9. D-optimal criterion. One method for selecting points to be used in response surface construction is to choose m points from a set of $l > m$ candidate points existing in the design space that will maximize the determinant $|\mathbf{X}^T\mathbf{X}|$, known as the D-optimality criterion [1]. The model here is $Y \approx \mathbf{X}c$, where Y is the $m \times 1$ vector of objective function values, c is the $k \times 1$ vector of coefficients to be estimated, and $\mathbf{X}$ is an $m \times k$ matrix of constants having rank k.

Create a set of $l > m$ points by forming a grid over the design space. Points of intersecting grid lines become candidate points. Conceivably, one could consider each of the $\binom{l}{m}$ 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. 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 $|\mathbf{X}^T\mathbf{X}|$.

10. Genetic algorithm for D-optimality. The genetic algorithm (GA) employed here is roughly modeled after the algorithm described in [12]. 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 candidates are selected as parents for breeding based on the fitness $|\mathbf{X}^T\mathbf{X}|$. 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 $|\mathbf{X}^T\mathbf{X}|$. After two parents have been selected, the breeding process begins. A genetic string (sequence of genes) could be either a sequence of integers $i_1 i_2 \ldots i_m$ identifying the m points in the design, or a sequence of mp real numbers corresponding to the coordinates of the m points. 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 last $G-j$ genes from the second parent. Note that this process (*crossover rule* in GA parlance) only makes sense for point coordinates when the approximation domain is a Cartesian product. Once the child is generated it goes through a mutation process. Each child has a 15% chance of mutating, whereby a randomly selected gene is changed. Finally the child is checked for uniqueness and duplicate points. If it is a duplicate or has duplicate points, it is destroyed.

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 $a_1 a_2 \ldots a_m$ 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.

Let $a_1 a_2 \cdots a_m$ and $b_1 b_2 \cdots b_m$ be two distinct parents chosen for breeding. Delete from $b_1 b_2 \cdots b_m$ any indices also occurring in the string $a_1 a_2 \cdots a_m$, resulting in

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

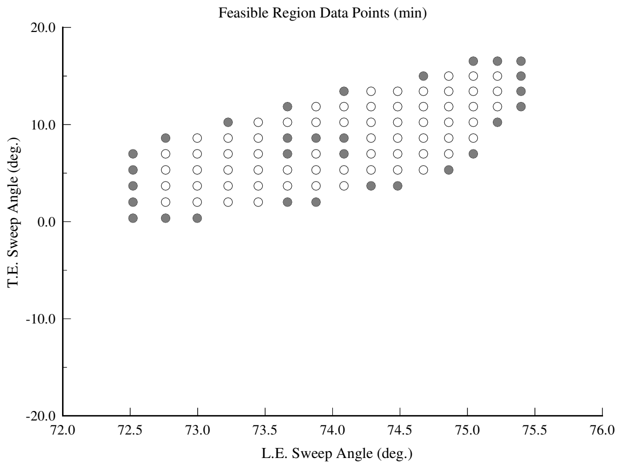

FIGURE 5. *Thirty D-optimal points (grey) selected by by the GA for a quadratic response surface fit.*

Now choose integer $k, 1 \leq k \leq r$, at random and delete k randomly chosen genes (indices) from $a_1 a_2 \cdots a_m$, 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 $k, 1 \leq k \leq r$. The random choices of k objects are made by generating random j 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.

11. D-optimal points in the example. Figure 5 shows an approximate D-optimal set of thirty points obtained by the GA for a quadratic

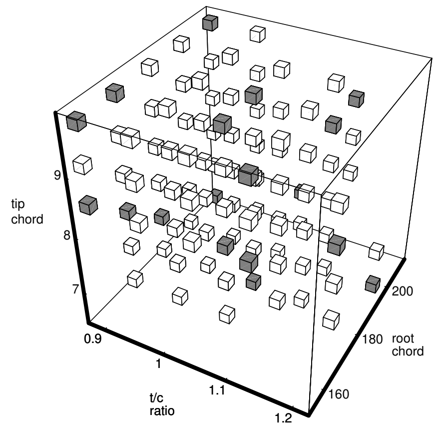

FIGURE 6. *Twenty D-optimal points selected by the GA, inboard leading edge sweep angle projection.*

polynomial response surface model to the data in Figure 3 (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 6 shows the projection along the inboard leading edge sweep angle axis 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 8 and Table 1). 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 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 5, the grey D-optimal points lie predominately along the perimeter of the approximation domain.

12. 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.

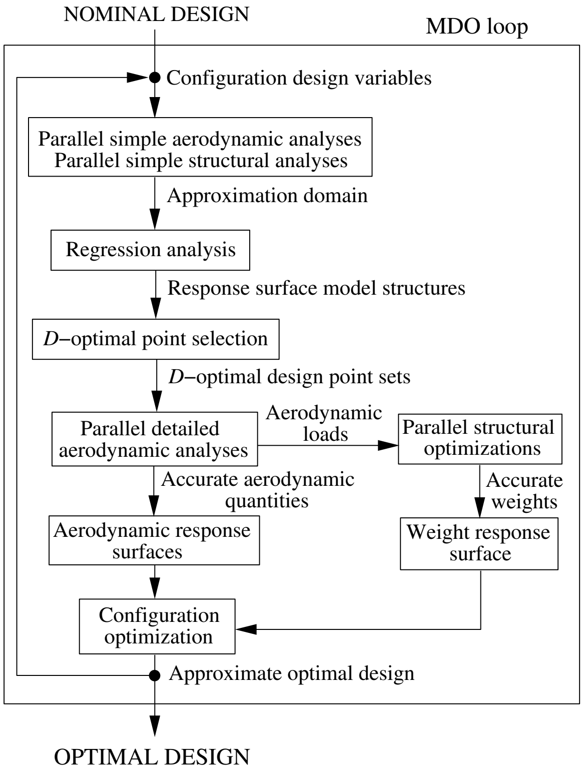

FIGURE 7. *MDO Paradigm.*

- *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. 7). 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 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 8 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

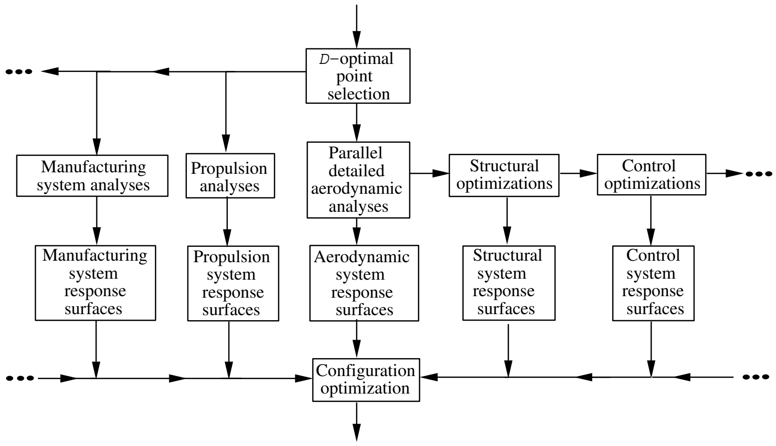

FIGURE 8. *General MDO paradigm.*

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 [27]. 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.

13. Parallelization of aerodynamic code. 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 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 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

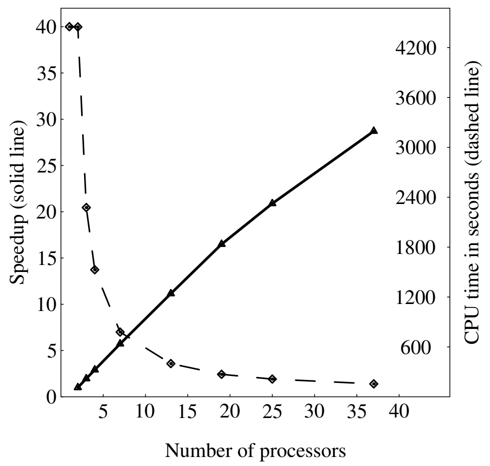

FIGURE 9. *Speedup results for the coarse grained parallelization of the simple aerodynamic code.*

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:

$$
\begin{aligned}
T_1 = &\text{parallelizable computation (PC)} + \text{I/O (I)} \\
&+ \text{inherently serial computation (SC)}.
\end{aligned} \tag{13.1}
$$

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

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

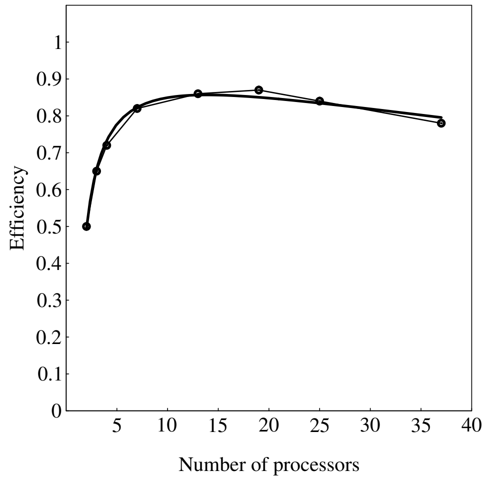

FIGURE 10. *Comparison of theoretical and actual efficiency for coarse grained parallelization of the simple aerodynamic code.*

Normalizing T_1 to one, the efficiency ω is given by

$$\frac{1}{n\ \max\left\{\dfrac{PC}{n-1}+SC,\ 1296*COM+I+X\right\}}. \tag{13.2}$$

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

$$\frac{1}{n\left(\dfrac{PC}{n-1}+SC\right)}. \tag{13.3}$$

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

$$\frac{1}{n(1296*COM+I+X)} \tag{13.4}$$

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 shown in Figures 9 and 10, and given in Table 2. The results for the speedup are reasonable, trailing off slightly as the number of processors increases above 20. Figure 9 also provides information about the CPU time in relation to the speedup. Since n here is relatively small, the efficiency model (13.2) was used for a nonlinear least squares fit to the

n	time (sec)	speedup	ω
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

TABLE 2. *Speedup and efficiency results for the simple aerodynamic analysis code.*

n	time (sec)	speedup	ω
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

TABLE 3. *Speedup and efficiency results for the detailed aerodynamic analysis code.*

data in Table 2. Figure 10 shows excellent agreement between the data (the circles) and the model (thick solid line).

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 3.

14. Parallelization of the 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.

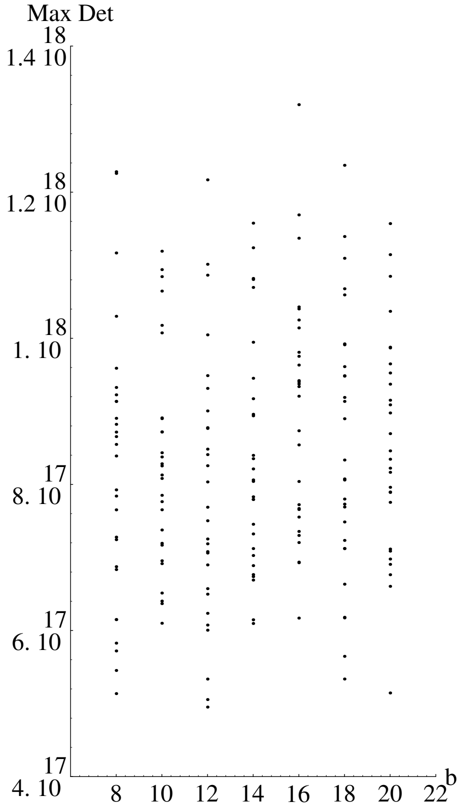

FIGURE 11. *Maximum determinant values versus the population size b.*

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 11 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 11 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 12. The coarse grained

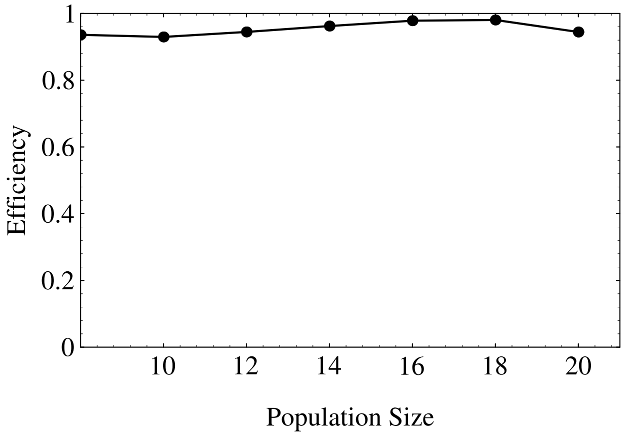

FIGURE 12. *Efficiency results for the GA.*

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.

15. Parallelization of 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 [13] is a general purpose finite element structural optimization code. Maestro [23] 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 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

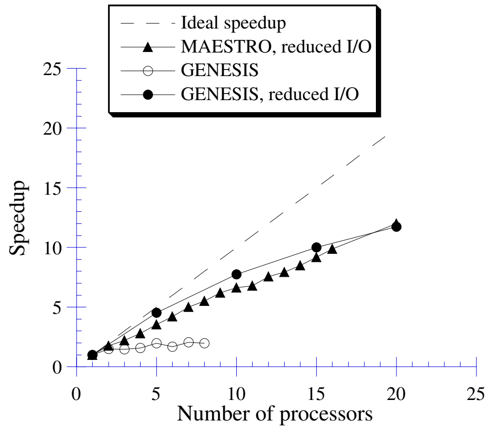

FIGURE 13. *Speedup results for the coarse grained parallelization of the structural code.*

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 13 and 14. 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 13 and 14 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.

16. Conclusions. To date much of the MDO work has been lacking in mathematical rigor or even sophisticated numerics. The work described here represents a modest but systematic attack on the obstacles listed in Section 2. Specifically, variable-complexity modeling (Section 5) addresses the high cost of detailed disciplinary analyses. Response surface methodology (Sections 6–9) deals with noisy function values, and also with the incompatibility of disciplinary computer codes. The latter ensues from integrating response surfaces generated by the disciplinary codes (parallel multipoint approximations, Sections 6–7), rather than directly integrating

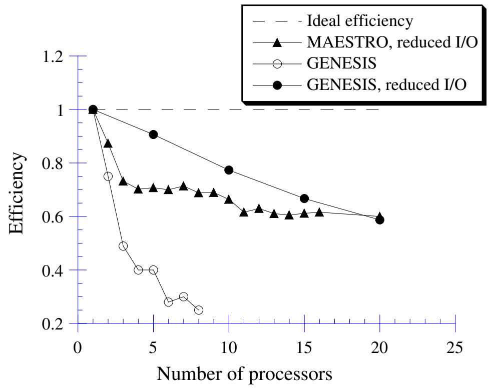

FIGURE 14. *Efficiency results for the coarse grained parallelization of the structural code.*

the codes themselves. Sections 10–11 concern algorithms particularly well suited to parallel computation and MDO. The pipelined MDO algorithm (Section 12) specifically addresses the potential nonconvergence of many current MDO schemes.

Unfortunately, some of the unavoidable but less glamorous obstacles to MDO (such as experimental model verification and the incorporation of software engineering practices) are not being fully addressed by the MDO community (including Virginia Tech). An important observation is that the *pipelined MDO algorithm proposed here is provably convergent*: the entire process is ultimately expressed in terms of aerodynamic design variables, and each outer iteration of the MDO loop improves the performance criterion in the configuration optimization. Still, tightly coupled interactions such as aeroelasticity or control-structure integration are not handled well by the pipeline, and one could argue that the pipelined MDO is a provably correct solution to the wrong problem.

Finally, in a nutshell, the essential new ideas here for MDO are

- variable complexity modeling,
- response surface approximations to noisy and/or expensive functions, and
- a pipelined MDO flowchart.

Acknowledgement. The HSCT design work reported here summarizes the efforts of many individuals over a period of several years, in particular, Vladimir Balabanov, Anthony A. Giunta, Bernard Grossman, Raphael T. Haftka, Matthew Kaufman, William H. Mason, and Robert Narducci. The opinions and conclusions here are solely those of the authors, and do not necessarily represent the views of other members of the Multidisciplinary Analysis and Design Center for Advanced Vehicles at Virginia Tech.

REFERENCES

[1] M. J. Box, and N. R. Draper, *Factorial Designs, the* $|\mathbf{X}^T\mathbf{X}|$ *Criterion, and Some Related Matters*, Technometrics, vol. 13, No. 4, 1971, pp. 731–742.

[2] R. Braun and I. Kroo, *Use of Collaborative Optimization in a Multidisciplinary Design Environment*, SciCADE '95, March 31, 1995.

[3] R. Braun and I. Kroo, *A Collaborative Approach to Trajectory Optimization*, ICASE/Langley Workshop on Multidisciplinary Optimization, Hampton, VA, March 13, 1995.

[4] S. L. Burgee, L. T. Watson, A. A. Giunta, B. Grossman, R. T. Haftka, and W. H. Mason, *Parallel Multipoint Variable-Complexity Approximations for Multidisciplinary Optimization*, Proceedings of the IEEE Scalable High-Performance Computing Conference, June 1994, pp. 734–740.

[5] S. L. Burgee, A. A. Giunta, R. Narducci, L. T. Watson, B. Grossman, and R. T. Haftka, *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.), SIAM, Philadelphia, PA, 1995, pp. 96–101.

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

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

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

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

[10] E. J. Cramer, J. R. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *Problem Formulation for Multidisciplinary Optimization*, SIAM J. Optimization, vol. 4, No. 4, 1994, pp. 754–776.

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

[12] H. Furuya and R. T. Haftka, *Locating Actuators for Vibration Suppression on Space Trusses by Genetic Algorithms*, ASME Winter Annual Meeting, 1993.

[13] *GENESIS User Manual*, Vanderplaats, Miura and Associates, Version 1.3, 1993.

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

[15] A.A. Giunta, R. Narducci, S. Burgee, B. Grossman, W.H. Mason, L.T. Watson and R.T. Haftka, *Variable-Complexity Response Surface Aerodynamic Design of an HSCT Wing*, AIAA Paper 95-1886, Proceedings of the 13th AIAA Applied Aerodynamics Conference, San Diego, CA, June, 1995

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

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

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

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

[20] M. G. Hutchison, E. R. Unger, W. H. Mason, B. Grossman, and R. T. Haftka, *Variable-Complexity Aerodynamic Optimization of an HSCT Wing Using Structural Wing-Weight Equations*, J. Aircraft, vol. 31, No. 1, 1994, pp. 110–116.

[21] A. I. Khuri and J. A. Cornell, *Response Surfaces: Designs and Analyses*, Marcel Dekker, Inc., New York, N. Y., 1987.

[22] Kroo, et al., *MDO for Large-Scale Aerospace Design*, ICASE/Langley Workshop on Multidisciplinary Optimization, Hampton, VA, March 13, 1995.

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

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

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

[26] R. H. MacNeal, *The NASTRAN Theoretical Manual*, The MacNeal-Schwendler Corporation: Los Angeles, CA, 1972.

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

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

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

[30] E. R. Unger, M. G. Hutchison, M. Rais-Rohani, R. T. Haftka, and B. Grossman, *Variable-Complexity Design of a Transport Wing*, Int. J. Sys. Automation: Res. and Appl., to appear.