AIAA JOURNAL
Vol. 40, No. 12, December 2002

Adaptive Experimental Design for Construction of Response Surface Approximations

Victor M. Pérez* and John E. Renaud†
University of Notre Dame, Notre Dame, Indiana 45556
and
Layne T. Watson‡
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

Sequential approximate optimization (SAO) is a class of methods available for the multidisciplinary design optimization of complex systems that are composed of several disciplines coupled together. One of the approaches used for SAO is based on a quadratic response surface approximation, where zero- and first-order information are required at the current design. In this method, designers must generate and query a database of order $\mathcal{O}(n^2)$ to compute the second-order terms of the quadratic response surface approximation. As the number of design variables grows, the computational cost of generating the required database becomes a concern. An adaptive experimental design (AED) approach that requires just $\mathcal{O}(n)$ parameters for constructing a second-order approximation is presented. This is accomplished by transforming the matrix of second-order terms into the canonical form. The AED method periodically requires an order $\mathcal{O}(n^2)$ update of the second-order approximation to maintain accuracy. Results show that the proposed approach dramatically reduces the total number of calls to the simulation tools during the SAO process.

Nomenclature

f	=	objective function
$\tilde{f}^k$	=	approximation to f near $\boldsymbol{x}^k$
$\boldsymbol{g}$	=	inequality constraint vector
g_i	=	ith inequality constraint
$\tilde{\boldsymbol{g}}^k$	=	approximation to g near $\boldsymbol{x}^k$
$\boldsymbol{H}_{(\,)}$	=	Hessian matrix of ( )
r	=	penalty parameter
$\boldsymbol{U}$	=	orthogonal eigenvector matrix
$\mathcal{U}^s$	=	transformed design space
$\boldsymbol{x}$	=	vector of design variables
$\boldsymbol{x}^k$	=	current design point at kth iteration
Δ	=	trust region radius
$\boldsymbol{\lambda}$	=	vector of Lagrange multipliers
ρ	=	trust region ratio
Φ	=	augmented Lagrangian function
Ψ_i	=	alternative form for inequality constraint g_i

I. Introduction

APPLYING nonlinear optimization strategies directly to complex multidisciplinary systems can be prohibitive when the complexity of the simulation codes is large. Increasingly, response surface approximations (RSAs), or surrogate functions, are being integrated with nonlinear optimizers to reduce the CPU time required for the optimization of complex multidisciplinary systems. RSAs provide a computationally inexpensive lower-fidelity representation of the system performance.

Two trends have emerged when integrating RSAs within nonlinear optimization tools: 1) the use of global approximations, where an RSA over the entire design space is developed, and 2) the use of local approximations, where RSAs are built within a local region around the current design. In general, a single optimization is performed when employing global approximations. The cost of developing a good global RSA is obviously higher than for local approximations because a more complex model is required to mimic the system. When using local RSAs, a sequential approximate optimization (SAO) methodology can be used. In SAO, the design space is sampled around each design iterate to generate the database required for constructing a low-order polynomial using regression analysis.[1]

The authors have investigated two different approaches for sampling the design space in SAO frameworks. The first is an optimization based sampling, which has roots in the original concurrent subspace optimization (CSSO) algorithm of Sobieszczanski-Sobieski.[2] It was later modified for RSA optimization (CSSO-RS) by Renaud and Gabriele[3,4] and Wujek et al.[5] and expanded to a formal framework for trust region model management by Rodríguez et al.[6] In this approach, each of the disciplines perform an optimization subject to move limits. The required inputs from other systems are computed by linear approximations. The design points visited through the subspace optimizations are stored and serve as the database for the RSA construction. The other approach is a statistically based sampling using design of experiments (DOE) arrays as reported by Rodríguez et al.[7,8] At each SAO iteration, a set of design points is selected for sampling using a DOE array. The design points are evaluated using the local disciplinary design tools, where linear approximations are used for the nonlocal input states. The resulting database is used to build an RSA. Many other research studies have combined DOE techniques and RSA for optimization.[9–13]

Rodríguez et al.[8] performed a comparison between the optimization-based data generation (CSSO-RS) and a statistical based DOE approach using orthogonal arrays (OA). Results of that study show that, whereas low-strength OAs seem to perform well compared to the CSSO-RS approach, the CSSO-RS approach is still more robust in driving the optimization. An attempt to overcome the natural advantages of the optimization-based sampling was investigated in Pérez et al.[14] In the Pérez et al. study, the DOE-based sampling strategy was modified by projecting the OAs onto the linearized descent feasible region. There is evidence that the RSA constructed with such a database provides a better approximation of the system when constrained optimization is performed. One common goal of these studies is the reduction of

Received 30 October 2001; revision received 13 June 2002; accepted for publication 19 June 2002.

*Graduate Research Assistant, Department of Aerospace and Mechanical Engineering; vperez@nd.edu. Student Member AIAA.
†Professor, Department of Aerospace and Mechanical Engineering, 365 Fitzpatrick Hall; jrenaud@nd.edu. Associate Fellow AIAA.
‡Professor, Departments of Computer Science and Mathematics, 660 McBryde Hall (0106); ltw@cs.vt.edu.

computational cost when an optimization problem is solved using SAO.

In the SAO implementation studied by the authors, a quadratic function is computed at each iteration to mimic the response of the system. The main drawback of using SAO for problems with more than a few design variables is that the size of the database required to compute the coefficients of the polynomial grows quadratically with the number of design variables. In this paper, a methodology is presented that requires the user to perform a much smaller sampling at each iteration by fitting a reduced model during some iterations. The methodology has been named adaptive experimental design (AED).

The organization of the paper is as follows: In Sec. II, the basics of SAO are presented. Section III describes in detail the proposed methodology. A full description of the algorithm is presented in Sec. IV. Test problems and their corresponding results are presented in Sec. V and concluding remarks in Sec. VI.

II. SAO

The SAO framework solves a constrained optimization problem of the following form:

Minimize

$$f(\boldsymbol{x}) \tag{1a}$$

subject to

$$\boldsymbol{g}(\boldsymbol{x}) \geq 0 \tag{1b}$$

$$\boldsymbol{h}(\boldsymbol{x}) = 0 \tag{1c}$$

$$\boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max} \tag{1d}$$

The main idea of the SAO is to build simple RSAs of the objective function and constraints, valid for a local region. An optimization is performed over this approximation within the local limits. The approximation is updated every iteration until convergence is achieved.

The algorithm begins at iteration $k = 0$ with a starting point $\boldsymbol{x}^k$. The objective function, constraints, and their gradients are evaluated in the design characterization step. Then local move limits are defined. The move limits define the region where the approximation will be valid. Within these move limits, a database is constructed. Once an RSA is constructed using the information in the database, an optimization is performed with the approximations. The new candidate point is evaluated and either accepted or rejected based on a trust region test. After this, new move limits are set, and the optimization proceeds. A flowchart of the general SAO is shown in Fig. 1.

In the SAO strategies of Wujek and Renaud[15,16] and Rodríguez et al.,[6,7] which were modified for DOE-based sampling by Rodríguez et al.,[8] Pérez et al.,[14] and Pérez and Renaud,[17] the framework constructs second-order RSAs of the objective function and constraints within a local trust region. The approximations are constructed using exact first-order information at the current design point. Therefore, the work of constructing a response surface (RS) involves only the fitting of the second-order terms, which are referred to as the Hessian-RS matrix H, or simply the Hessian.

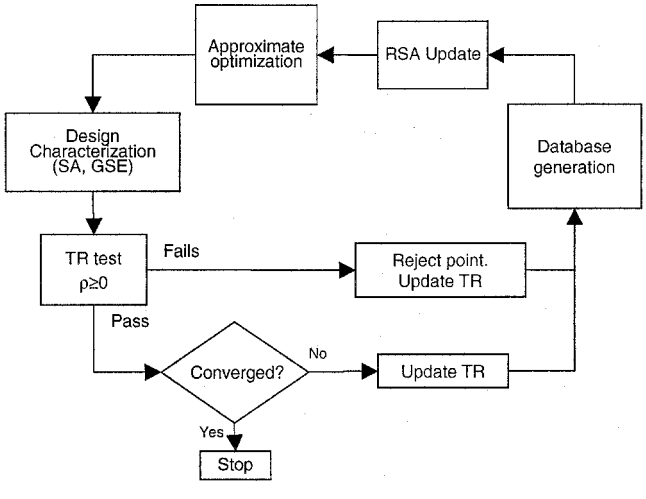

Fig. 1 SAO algorithm.

Alexandrov et al.[18] proved the convergence of a SAO framework in which the move limits are managed by a trust region approach, and the local approximation matches the function and the gradient at the current design point. The trust region augmented Lagrangian framework implemented in this study has these characteristics and has been proven to converge by Rodríguez et al.[6] This framework uses an augmented Lagrangian approach for driving the optimization and a trust region methodology to manage the move limits.

In the present research, the trust region augmented Lagrangian framework of Rodríguez et al.[6] is implemented. A more detailed description of the RSA, the trust region approach, and the database generation used in this study is presented in the following sections.

A. RSA

The quadratic function approximation used in the framework for f at the kth iteration is given by

$$\tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T \Delta \boldsymbol{x}^k + \tfrac{1}{2} \Delta \boldsymbol{x}^k \boldsymbol{H}_f^k \Delta \boldsymbol{x}^k \tag{2}$$

where $\Delta \boldsymbol{x}^k = \boldsymbol{x} - \boldsymbol{x}^k$. Both $f(\boldsymbol{x}^k)$ and $\nabla f(\boldsymbol{x}^k)$ are known, and only the second-order information $\boldsymbol{H}_f^k$ is computed using a response surface approach. Gradient information can be obtained in analytical form provided by the user, or computed via finite differences or using the global sensitivity equations[19] in the case of a multidisciplinary design optimization (MDO) problem.

Note here the difference between the true Hessian of the function and the Hessian-RS of the approximation $\tilde{f}^k$, denoted here as $\boldsymbol{H}_f^k$. The true Hessian is the second-order information evaluated at a given design point, in this case $\boldsymbol{x}^k$. The Hessian-RS, or simply Hessian, as it is referred to in this paper, is the Hessian matrix of a quadratic approximation of the function in the local region. The true Hessian is never computed nor used in this SAO framework. It is too expensive to compute by finite differences and assumed to be difficult to obtain in analytical form. Whenever Hessian is used in this paper, it refers to the matrix of second-order coefficients in the quadratic approximation.

The total number of coefficients in a full second-order model is $1 + n + n(n+1)/2$. However, in the quadratic model considered here, zero- and first-order information are known; therefore, the number of coefficients in $\boldsymbol{H}_f^k$ is $n(n+1)/2$, where n is the number of design variables in the problem. The minimum size of the database required to compute the approximate model (2) is, therefore, $n(n+1)/2$ or $\mathcal{O}(n^2)$ in general terms. The coefficients in the Hessian matrix are computed using the least-squares method.

The least-squares method assumes an independent and normally distributed error in the model to be fitted. This is debatably true for computer experiments. Sacks et al.[20] points out that in the absence of independent random errors, the rationale for least-squares fitting of an RS is not clear. However, he also indicates that least squares can be viewed as curve fitting. Simpson et al.[13] also underlines that, in deterministic computer experiments, the model should interpolate and not smooth along the data points. However, in realistic MDO applications, the analysis values are, in fact, noisy, and so least-squares fitting is more appropriate than interpolation. In this particular application, the quadratic model is used in the curve fitting sense, within a local region controlled by the trust region methodology. The general trend is captured by the approximation. When approaching convergence, the trust region method guarantees the adequacy of the quadratic model by shrinking the move limits. Furthermore, if variable fidelity data are used,[8] smoothing along the data points is a necessity. Other methods have been proposed to compute the coefficients of an RS, such as the minimum bias estimation method (see Myers and Montgomery[1]).

B. Trust Region Methodology

The trust region approach[15,16,18,21–23] is based on the use of a trust region ratio ρ to monitor how well the current approximation

is found to represent the actual design space. Consider an unconstrained optimization problem in which $\tilde{\phi}^k$ is an approximation to the function ϕ around the point $\boldsymbol{x}^k$ at the kth iteration. The move limits are defined by the region $\|\boldsymbol{x}-\boldsymbol{x}^k\|_p \le \Delta^k$ and the p norm defines the shape of the region. Δ^k is known as the trust region radius.

The minimization of $\tilde{\phi}^k$ subject to the defined trust region gives a new candidate point $\boldsymbol{x}_*^{k+1}$. The trust region ratio ρ is computed based on the information of the new candidate point:

$$\rho^k = \frac{\phi(\boldsymbol{x}^k) - \phi\left(\boldsymbol{x}_*^{k+1}\right)}{\tilde{\phi}^k(\boldsymbol{x}^k) - \tilde{\phi}^k\left(\boldsymbol{x}_*^{k+1}\right)} \tag{3}$$

The quantity ρ is simply the ratio of the actual change in the function to the change predicted by the approximation. The closer the value of ρ is to one, the better the approximation $\tilde{\phi}^k$ mimics the behavior in the descent direction of ϕ. The approximate minimization forces $\tilde{\phi}^k(\boldsymbol{x}_*^{k+1}) \le \tilde{\phi}^k(\boldsymbol{x}^k)$. A negative ρ indicates a poor approximation, and the new point does not decrease the function; therefore, the point is rejected and the trust region radius reduced. If the value of ρ is greater than zero, the point is kept: $\boldsymbol{x}^{k+1} = \boldsymbol{x}_*^{k+1}$. The trust region radius Δ is updated according to the following rules:

$$\Delta^{k+1} = \begin{cases} c_1\Delta^k & \text{if} \quad \rho^k < R_1 \\ c_2\Delta^k & \text{if} \quad \rho^k > R_2 \\ \Delta^k & \text{otherwise} \end{cases} \tag{4}$$

Typical values for the limiting range constants are $R_1 = 0.25$ and $R_2 = 0.75$. The trust region multiplication factors c_1 and c_2 are commonly set to be 0.25 and 2, respectively.

C. Database Generation

In SAO, at every iteration a new database is computed. The points to be evaluated can be generated using an optimization based sampling as in Refs. 4, 6, 15, and 16 or using more traditional DOE strategies as in Refs. 7, 8, 14, and 17. Each design point that is sampled can be evaluated by computing the system analysis (SA). In the case of multidisciplinary optimization problems, one could sample the linearly decoupled simulation codes or contributing analysis (CAs), as in Refs. 8 and 14, generating variable fidelity data.

Several techniques have been developed to sample the design space and generate an RSA efficiently. Among the common techniques to generate an experimental design are the traditional DOE arrays, such as full factorial experiments (FF), central composite designs (CCD), fractional factorials,[1] Latin hypercubes and their extensions (see, for example, McKay et al.[24] and Tang[25]), and OAs.[26,27] Some quality improvement computations such as D-optimality[28,29] are too large or too complex for an SAO framework. OAs are an excellent choice for computer experiments because they are easy to generate. Owen[26] has compiled a suite of programs to generate a broad class of OAs for different levels and numbers of design variables. In the construction of RSAs for MDO, one important consideration is the dimensionality of the problem, where large problems may impose the curse of dimensionality. Small problems can be easily dealt with by traditional sampling techniques (FF, CCD). However, as the number of variables increases, the size of the sampling does too. In the case of a second-order RSA, the number of the parameters to be fitted is $\mathcal{O}(n^2)$. FF and CCD generate design arrays with $\mathcal{O}(2^n)$, whereas some OAs have $\mathcal{O}(n^2)$ or even $\mathcal{O}(n)$. In this paper, OAs have been chosen as experimental arrays.

III. AED

The most important difference, from an experimental point of view, between traditional laboratory experiments and the computational experiments embedded in SAO, is that in the latter the experiment is repeated several times at different locations, up to convergence or stopping of the algorithm. At each new iteration, a new sampling is performed of the same system, but in a new sampling region. However, at each new iteration one is not completely blind to the behavior of the system because the earlier information about the nature of the RS, and the fitted coefficients of the preceding local approximation, are available. Because of the highly nonlinear nature of MDO problems, the design space is not expected to have the same response from the starting sampling region to the final one, and, therefore, the performance of a fixed experimental array may vary through the process. This was acknowledged in the research of Rodríguez et al.,[8] where the OAs were randomized to avoid having a fixed experimental array not capturing the true interactions of the design variables.

The main drawback for SAO is the size of the database required to construct the quadratic approximation. The number of parameters needed to fit a full second-order approximation is $\mathcal{O}(n^2)$. This problem is referred to as the aforementioned curse of dimensionality. An AED that takes advantage of the information from the preceding local approximation to modify the size of the experimental array required for the next sampling appears to hold promise for improving the efficiency of SAO algorithms. In this paper, information already available from the preceding approximation is used to reduce the size of the data required to construct a full quadratic model, while maintaining the quality of the approximation. As a result, the total cost of the optimization is reduced.

A reduction in the cost of the optimization can be obtained if, for some iterations, instead of an $\mathcal{O}(n^2)$ sampling, only an $\mathcal{O}(n)$ sampling is performed. This can be done by neglecting the off-diagonal terms in the Hessian matrix $\boldsymbol{H}_f^k$. Although this certainly reduces the order of data to be queried to $\mathcal{O}(n)$, the quality of the approximation might decrease considerably. These neglected terms could be an important component of the second-order information, and, thus, it would end up being a poor representation of the system response.

A different approach is proposed in this paper. Once the full matrix of second-order terms is approximated for a given design point, it can be transformed into its canonical form. In the canonical coordinates, the off-diagonal components of the Hessian vanish. Once the canonical transformation matrix has been found, it can be assumed, at least for some number of iterations, that the curvature of the function will be invariant, and so the transformation matrix is kept. In the next iterations, in the transformed design space, only the main diagonal of the second-order matrix is fitted [$\mathcal{O}(n)$ sampling]. As a result, back in the original space, a full matrix is obtained. The transformation matrix can be kept as long as the curvature of the function does not change too much. In that case, the full matrix of second-order terms has to be updated, and a new transformation matrix is computed. This technique is referred to as the AED mentioned earlier.

In the following section, a detailed description of the AED approach is presented.

A. Proposed Methodology

Let us assume that a single function f is being approximated. The quadratic approximation at the kth iteration is represented by Eq. (12). The matrix of second-order terms $\boldsymbol{H}_f^k$ is symmetric and, hence, similar to a real diagonal matrix via an orthogonal transformation matrix, whose columns are orthonormal eigenvectors. The eigenvalue decomposition of the Hessian matrix is defined as

$$\boldsymbol{H}_f^k = \boldsymbol{U}_f^k \boldsymbol{D}_f^k \boldsymbol{U}_f^{k\,T} \tag{5}$$

The matrix of normalized eigenvectors $\boldsymbol{U}_f^k$ defines a change of coordinates, whereas the diagonal matrix of real eigenvalues $\boldsymbol{D}_f^k$ defines the magnitude of the curvature along the new coordinate directions. This transformation and its significance is illustrated in Fig. 2.

With $\boldsymbol{U}_f^k$ as the transformation matrix, the design variables $\boldsymbol{x}$ can be transformed to the canonical variables $\boldsymbol{x}_U$. The canonical form of the quadratic approximation is

$$\tilde{f}_U^k(\boldsymbol{x}_U) = f_U\left(\boldsymbol{x}_U^k\right) + \nabla f_U^T\left(\boldsymbol{x}_U^k\right)\Delta\boldsymbol{x}_U^k + \tfrac{1}{2}\Delta\boldsymbol{x}_U^{k\,T}\boldsymbol{D}_f^k\Delta\boldsymbol{x}_U^k \tag{6}$$

where

$$\boldsymbol{x}_U^k = \boldsymbol{U}_f^{k\,T}\boldsymbol{x}^k \tag{7}$$

$$\Delta\boldsymbol{x}_U^k = \boldsymbol{x}_U - \boldsymbol{x}_U^k \tag{8}$$

$$f_U(\boldsymbol{x}_U) = f\left(\boldsymbol{U}_f^k\boldsymbol{x}_U\right) \tag{9}$$

$$\nabla f_U(\boldsymbol{x}_U) = \boldsymbol{U}_f^{k\,T}\nabla f(\boldsymbol{x}) \tag{10}$$

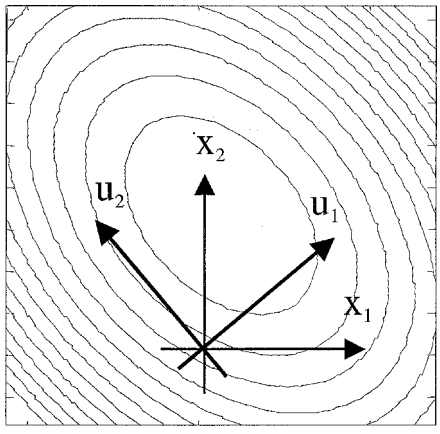

Fig. 2 Transformation to canonical form.

Now assume that a full Hessian is known for a preceding nearby iteration s, $s < k$. Its corresponding eigenvector matrix is $\boldsymbol{U}_f^s$. If the points x^k and x^s are close enough, it is safe to assume that $\boldsymbol{U}_f^k$ and $\boldsymbol{U}_f^s$ are similar. Therefore, to compute the Hessian matrix $\boldsymbol{H}_f^k$, one need only calculate the eigenvalue matrix $\boldsymbol{D}_f^k$ by sampling the design space with an $\mathcal{O}(n)$ experimental array. This experimental array is sampled in the original space. Then the database is transformed to the canonical space $\mathcal{U}^s$, where the eigenvalue matrix $\boldsymbol{D}_f^k$ is computed. Finally, a backtransformation returns the new full Hessian matrix in the original space:

$$\boldsymbol{H}_f^k = \boldsymbol{U}_f^s \boldsymbol{D}_f^k \boldsymbol{U}_f^{s\,T} \tag{11}$$

The eigenvector matrix $\boldsymbol{U}_f^s$ can be used for several iterations. If at any point the approximation is bad, then a new eigenvector matrix is computed. This is done by constructing a full Hessian using an $\mathcal{O}(n^2)$ experimental array (EA). In this paper, the criterion for a bad approximation is a failure in the trust region test as described in Sec. II.B. The trust region test fails when the new candidate point increases the value of the function. The bad approximation can be a result of a trust region being too large or a bad model. Therefore, a trust region reduction plus a transformation matrix update have to be performed.

In summary, the user requires two EAs one with $l_1 \geq n(n+1)/2$ $[\mathcal{O}(n^2)]$ design points and one with $l_2 \geq n[\mathcal{O}(n)]$ design points. At the first iteration, the local design space defined by the trust region is sampled using the $\mathcal{O}(n^2)$ EA. A full Hessian is approximated and its eigenvector matrix stored. For subsequent iterations the local design space is sampled using the $\mathcal{O}(n)$ EA. By the use of the AED technique described earlier, a full Hessian matrix can be approximated. When the trust region test fails ($\rho^k < 0$), then the algorithm resets the transformation matrix by sampling the $\mathcal{O}(n^2)$ EA.

B. Practical Implementation

In MDO, it is common to have several constraints in the formulation of a problem. The objective function and the constraints are often functions of the outputs of coupled simulation codes, and, thus, RSAs are used to approximate them. A direct application of the methodology just described, in a problem with m constraints, implies that $m+1$ least-squares estimations would have to be solved. For each function, an eigenvalue decomposition of its Hessian matrix, an individual database transformation, and a least-squares estimation would have to be performed. We refer to this as the individual transformation (IT) method.

It is common in SAO frameworks for the optimization to be performed over an unconstrained augmented Lagrangian, with variable bounds. In this case, one need only approximate the augmented Lagrangian, and, therefore, only a single transformation matrix is required. This method, referred to as the cumulative Hessian approach, requires that the Hessian of the augmented Lagrangian be derived from the functions approximations.

In the SAO framework employed in this research, the objective function and each of the constraints are approximated by second-order Taylor series. Define the approximate objective function and constraints by

$$\tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T \Delta \boldsymbol{x}^k + \tfrac{1}{2} \Delta \boldsymbol{x}^{k\,T} \boldsymbol{H}_f^k \Delta \boldsymbol{x}^k \tag{12}$$

$$\tilde{g}_i^k(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^T \Delta \boldsymbol{x}^k + \tfrac{1}{2} \Delta \boldsymbol{x}^{k\,T} \boldsymbol{H}_{g_i}^k \Delta \boldsymbol{x}^k \tag{13}$$

To simplify the derivation, only inequality constraints are included in this paper; however, the extension to include equality constrains is straightforward. The approximate augmented Lagrangian is

$$\tilde{\Phi}^k = \tilde{f}^k + \lambda^T \tilde{\Psi}^k + r \tilde{\Psi}^{k\,T} \tilde{\Psi}^k \tag{14}$$

where

$$\tilde{\Psi}_i^k = \min\left[\tilde{g}_i^k, -2(\lambda_i / r)\right] \tag{15}$$

is provably continuously differentiable (C^1). We are interested in the matrix of second-order terms $\boldsymbol{H}_{\tilde{\Phi}}^k$ in the Taylor series expansion of $\tilde{\Phi}^k$:

$$\tilde{\Phi}^k(\boldsymbol{x}) = \tilde{\Phi}^k(\boldsymbol{x}^k) + \nabla \tilde{\Phi}^k(\boldsymbol{x}^k)^T \Delta \boldsymbol{x}^k + \tfrac{1}{2} \Delta \boldsymbol{x}^{k\,T} \boldsymbol{H}_{\tilde{\Phi}}^k \Delta \boldsymbol{x}^k + \mathcal{O}(\|\Delta \boldsymbol{x}^k\|^3) \tag{16}$$

The reader should keep in mind that Hessian coefficients are estimated using RS techniques that fit the second-order response over the entire trust region. Substituting Eqs. (12) and (13) into Eq. (14) and matching the order of the terms, one obtains

$$\boldsymbol{H}_{\tilde{\Phi}}^k = \boldsymbol{H}_f^k + \sum_i \lambda_i \boldsymbol{H}_{\tilde{\Psi}_i}^k + 2r \sum_i \left(\nabla \tilde{\Psi}_i^k \nabla \tilde{\Psi}_i^{k\,T} + \tilde{\Psi}_i^k \boldsymbol{H}_{\tilde{\Psi}_i}^k \right) \tag{17}$$

Note that the expression for $\boldsymbol{H}_{\tilde{\Phi}}^k$ contains both first- and second-order terms. The first- and second- order terms can be combined into single terms

$$\boldsymbol{H}_{\tilde{\Phi}}^k = \boldsymbol{H}_{\tilde{\Phi}}^{k\,A} + \boldsymbol{H}_{\tilde{\Phi}}^{k\,B} \tag{18}$$

where

$$\boldsymbol{H}_{\tilde{\Phi}}^{k\,A} = 2r \sum_i \nabla \tilde{\Psi}_i^k \nabla \tilde{\Psi}_i^{k\,T} \tag{19}$$

$$\boldsymbol{H}_{\tilde{\Phi}}^{k\,B} = \boldsymbol{H}_f^k + \sum_i \left(\lambda_i + 2r \tilde{\Psi}_i^k \right) \boldsymbol{H}_{\tilde{\Psi}_i}^k \tag{20}$$

$\boldsymbol{H}_{\tilde{\Phi}}^{k\,A}$ is known because it is computed by first-order information only, which is already available. The matrix $\boldsymbol{H}_{\tilde{\Phi}}^{k\,B}$ is a linear combination of the objective function and constraint Hessians. If the eigenvector matrix of $\boldsymbol{H}_{\tilde{\Phi}}^{k\,B}$ is known, there is no need to compute the individual off-diagonal terms of the objective function and constraints. They will cancel with each other in the transformed space. Therefore, one can use the eigenvector matrix $\boldsymbol{U}_{\tilde{\Phi}}^{s\,B}$ of $\boldsymbol{H}_{\tilde{\Phi}}^{s\,B}$ as a unique transformation matrix. Then fit in the transformed space the main diagonals of all of the Hessian matrices of f and $\boldsymbol{g}$ as described in Sec. III.A. This approach is called the cumulative Hessian (CH) method.

Note that although the function $\tilde{\Psi}_i^k$ is not C^2, one can assume, in practice, that for an active constraint $\tilde{g}_i < -2\lambda_i / r$ and $\boldsymbol{H}_{\tilde{\Psi}_i}^k = \boldsymbol{H}_{g_i}^k$, and, for an inactive constraint, $\boldsymbol{H}_{\tilde{\Psi}_i}^k = 0$.

IV. Algorithm Description

The modified SAO using the AED methodology proposed is presented in Fig. 3. For simplicity, the algorithm mentions a single function to be approximated; however, it is equally applicable to a fully constrained problem using one of the techniques described in Sec. III.B.

Given a starting point $\boldsymbol{x}^0$, set $k := 0$:

1) Compute $f(\boldsymbol{x}^k)$ and $\nabla f(\boldsymbol{x}^k)$.
2) Initialize the trust region.
3) Sample the local design space using an $\mathcal{O}(n^2)$ EA.

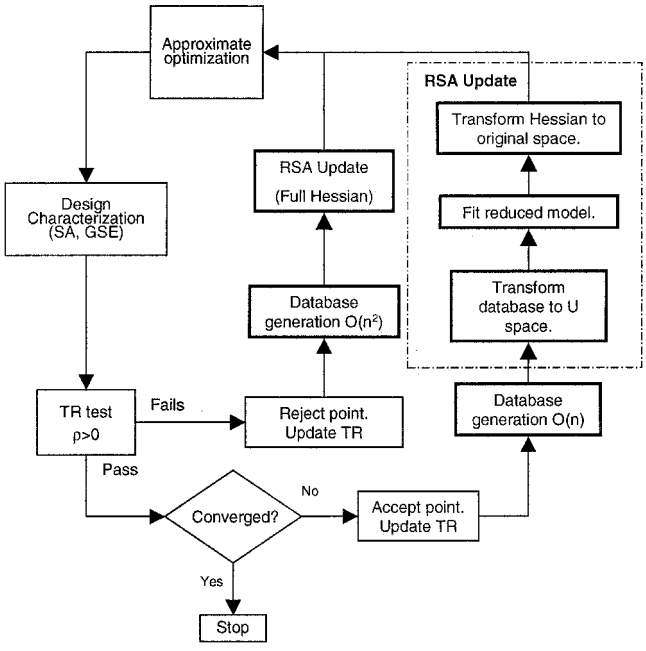

Fig. 3 SAO with AED.

4) Compute an approximation for the full Hessian matrix $\boldsymbol{H}_f^k$.

5) Compute the transformation matrix $\boldsymbol{U}_f^k$ as in Eq. (5). Set $s := k$.

6) Optimize the approximate augmented Lagrangian within the local trust region. Find a new candidate point $\boldsymbol{x}_*^{k+1}$.

7) Compute $f(\boldsymbol{x}_*^{k+1})$ and $\nabla f(\boldsymbol{x}_*^{k+1})$.

8) Compute the trust region ratio ρ.

9) If $\rho < 0$, go to 10; otherwise, go to 11.

10) Reject $\boldsymbol{x}_*^{k+1}$, update the trust region according to Eq. (4), and go to 3.

11) Set $\boldsymbol{x}^{k+1} := \boldsymbol{x}_*^{k+1}$ and $k := k+1$.

12) If algorithm convergence is achieved, stop; otherwise continue.

13) Update the trust region according to Eq. (4).

14) Sample the local design space using the $\mathcal{O}(n)$ experimental design.

15) By the use of $\boldsymbol{U}_f^s$, transform the database and the gradient information according to Eqs. (7) and (10).

16) In the transformed space(s) $\mathcal{U}^s$, compute the diagonal matrix $\boldsymbol{D}_f^k$ using the least-squares methodology.

17) Transform the diagonal Hessian approximation $\boldsymbol{D}_f^k$ back to a full Hessian matrix $\boldsymbol{H}_f^k$ using Eq. (11).

18) Update the r and λ_i in Eq. (14) as in Refs. 6 and 7, and go to 6.

V. Test Problems

To demonstrate the application of the decomposition technique within a SAO framework, three problems are shown. First a simple two-dimensional unconstrained problem is tested. Next, a two-dimensional constrained mathematical problem is tested. Finally, the strategy is applied to a two-discipline MDO test problem with 11 design variables. The complete implementation of the algorithm and test problems was done in MATLAB®. MATLAB's SQP optimizer was used to perform the approximate minimization at each iteration.

A. Optimization of an Unconstrained Function

A simple unconstrained minimization problem in two dimensions is used to show how dramatic the results can be:

Minimize

$$f(\boldsymbol{x}) = 3 + (x_1 - 40) + 2(x_2 - 40) + 0.75(x_1 - 40)^2 + 0.75(x_2 - 40)^2 + 0.5(x_1 - 40)(x_2 - 40) + 0.002x_1^2 x_2 \quad (21)$$

The optimum is located in $x^* = (36.2564, 38.1618)$. A simple SAO, with a fixed trust region radius, was performed using three methods to build the Hessian approximation: a full Hessian fit, the transformation method stated earlier, and fitting just the main diagonal without transformation. In the first two cases, the optimum was located in three iterations, starting from a nearby point. However, when fitting just the main diagonal, the algorithm required nine iterations to reach the optimum with a tolerance of 0.01. With a smaller tolerance (0.0001) it required more than 300 iterations to converge as compared to three for the full Hessian and the transformation method investigated in this paper.

Although actual engineering problems are more complex and restrictive than this unconstrained problem, it does illustrate the potential of the proposed method. The additional examples given hereafter represent more complex constrained problems that are closer to real engineering problems.

B. Barnes Problem

This is a purely artificial, two-dimensional problem originally formulated in Ref. 30. This problem was chosen because it is a highly nonlinear two-design variable problem. The problem is stated as follows:

Minimize

$$f(\boldsymbol{x}, \boldsymbol{y}) = a_1 + a_2 x_1 + a_3 y_4 x_1 + a_5 y_4^2 + a_6 x_2 + a_7 y_1 + a_8 x_1 y_1 + a_9 y_1 y_4 + a_{10} y_2 y_4 + a_{11} y_3 + a_{12} x_2 y_3 + a_{13} y_3^2 + a_{14}/(x_2 + 1) + a_{15} y_3 y_4 + a_{16} y_1 y_4 x_2 + a_{17} y_1 y_3 y_4 + a_{18} x_1 y_3 + a_{19} y_1 y_3 + a_{20} e^{a_{21} y_1} \quad (22a)$$

subject to

$$g_1 = y_1/700 - 1 \geq 0, \qquad g_2 = x_2/5 - y_4/25^2 \geq 0$$
$$g_3 = (y_5 - 1)^2 - (x_1/500 - 0.11) \geq 0$$
$$0.0 \leq x_1 \leq 75.0, \qquad 0.0 \leq x_2 \leq 65.0 \quad (22b)$$

where the coefficients a are constants given in Table A1 in the Appendix. The states $\boldsymbol{y}$ are calculated by the CAs as follows: For CA_1

$$y_1 = x_1 x_2, \qquad y_3 = x_2^2$$

for CA_2

$$y_2 = y_1 x_1, \qquad y_4 = x_1^2, \qquad y_5 = x_2/50$$

Figure 4 is a plot of the constraints and objective function contours.

1. *Implementation Details*

Two iterations of SAO were performed starting from several design points. In the first iteration, a full Hessian was fitted for the

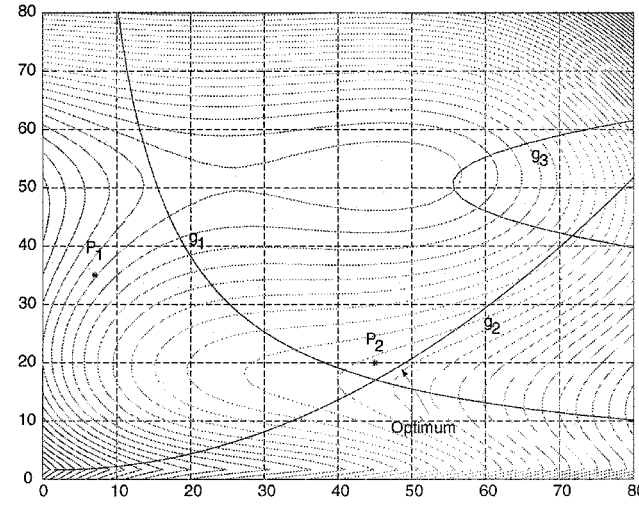

Fig. 4 Contours of f and g_i for the Barnes problem.

objective function and for each of the constraints. A new candidate point was found by solving a minimization over the approximate augmented Lagrangian.

In the second iteration, a new database was generated around the current design point. With this database, the Hessian matrix was computed using the following methods:

1) Estimate the full Hessian matrix (FH).

2) Estimate only the main diagonal (MD) terms of the Hessian without transformations in the original design space.

3) Estimate the MD in the transformed space.

a) Use one transformation matrix for each individual function involved (IT). The Hessian of each function is estimated in its transformed space.

b) Use one transformation matrix from the CH of the Lagrangian. The eigenvector matrix of the CH as in Eq. (20) is used to transform the design space. The MDs of each Hessian are computed in this transformed design space.

Once the Hessian information was estimated, an optimization was performed using each of the four approximations computed. The database for all approximations was generated by sampling with a 9-point, 3-level, FF array. The database was queried within the limits of the local variable bounds defined as a fraction of the whole design space by the trust region radius Δ.

To estimate the error of each approximation, a new database was queried. The mean square error (MSE) of what was computing by comparing the real value of the augmented Lagrangian to that obtained by the approximations [Eq. (14)].

2. Results for the Barnes Problem

The results for two initial design points are given in this section. The first run started with point $P_1 = (7, 35)$, which is an infeasible starting point that violates constraint 1. The sampling and optimization were performed with a trust region radius of $\Delta = 0.1$. Table 1 summarizes the results for each strategy used to fit the Hessian. In SAO, the optimization is performed with the augmented Lagrangian; therefore, the results are focused on the ability to capture the value of the Hessian matrix of the cumulative augmented Lagrangian as in Eq. (20).

The first three columns denote the Hessian terms of the cumulative augmented Lagrangian: x_1^2, x_2^2, and x_1x_2. We can see that, whereas the MD approach perfectly fits the x_1^2, x_2^2 terms, the cross term is not captured, although it has a somewhat negligible value. The individual transformation and the CH strategies capture, with an acceptable accuracy, the MD and off-diagonal terms. The small discrepancy in the results between the IT and the CH methods are due to the constraint violation; however, the difference is not significant. The next two columns show the final result for the two-step optimization: x_1^*, x_2^*. The final design is the same in all cases. When comparing the MSE of the four RSAs, it can be seen that the MD approach generates a greater error (10 times greater) than those of the individual transformation and the CH transformation. The magnitude of the MSE for the two transformation methods is comparable to that of the FH fit.

In the next run, the size of the trust region radius was increased to $\Delta = 0.3$, and the same starting point was used. Figure 5 shows the shape contours of the original augmented Lagrangian and the four approximations. We can see that a slight change in shape is found in the MD approach; however, the minimum for the five plots is located in the lower right corner [$x^* = (18.625, 24.925)$] of Fig. 5.

Results for the Hessian terms are found in Table 2. The maximum difference in Hessian terms is found in the off-diagonal terms because the MD approach does not compute them. The MSE shows a very small difference between the FH, IT, and CH. The MSE for the MD approach is more than twice the value of the other three. However, the four strategies end up at the same optimum.

Table 1 Point (7, 35), Δ = 0.1

Method	x_1^2	x_2^2	x_1x_2	x_1^*	x_2^*	MSE
FH	0.1141	−0.0112	0.0254	11.125	31.425	1.2615*e*−3
MD	0.1141	−0.0112	0	11.125	31.425	1.4201*e*−2
IT	0.1145	−0.0118	0.0251	11.125	31.425	1.2631*e*−3
CH	0.1145	−0.0117	0.0251	11.125	31.425	1.3205*e*−3

Table 2 Point (7, 35), Δ = 0.3

Method	x_1^2	x_2^2	x_1x_2	x_1^*	x_2^*	MSE
FH	0.1143	−0.0095	0.0282	18.625	24.925	0.8933
MD	0.1143	−0.0095	0	18.625	24.925	2.0462
IT	0.1173	−0.0138	0.0261	18.625	24.925	0.8989
CH	0.1172	−0.0137	0.0261	18.625	24.925	0.9115

Table 3 Point (45, 20), Δ = 0.1

Method	x_1^2	x_2^2	x_1x_2	x_1^*	x_2^*	MSE
FH	−0.0318	0.1549	0.0262	49.125	16.425	9.3201*e*−04
MD	−0.0318	0.1549	0	49.125	16.425	1.3077*e*−02
IT	−0.0316	0.1546	0.0264	49.125	16.425	9.3323*e*−04
CH	−0.0316	0.1546	0.0264	49.125	16.425	9.3323*e*−04

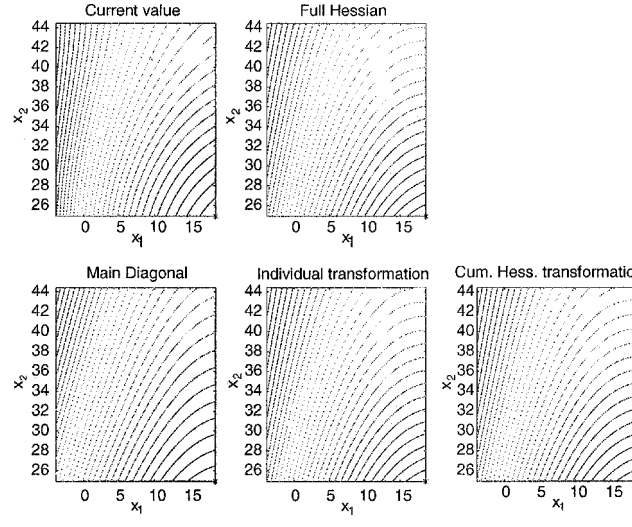

Fig. 5 Augmented Lagrangian for point (7, 35), Δ = 0.3.

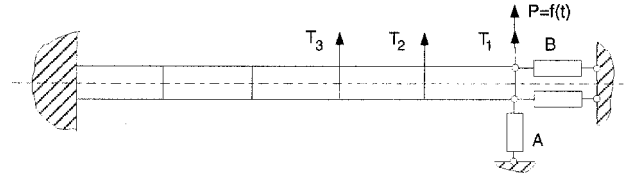

Fig. 6 Controls-augmented structure.

The second point evaluated is $P_2 = (45, 20)$, which is located near the global optimum. A trust region radius of $\Delta = 0.1$ is used in the same two-iteration study. At this design point, no constraint is violated, but it is close to the feasible design boundaries of g_1 and g_2. Results for this point are shown in Table 3. Results for IT and CH are the same because no constraint is being violated and, therefore, the augmented Lagrangian is reduced to the objective function. (We are performing a two-step SAO. The Lagrangian multipliers are initialized to zero.) The MSE for the MD strategy is two orders of magnitude greater than the other three approaches; however, the optimum with respect to the local variable bounds is found to be the same for each of the approximations.

C. Controls-Augmented Structure

In this section, an engineering design optimization problem called the controls-augmented structure (CAS) is detailed. This problem introduces more complexity with more design variables, more constraints, and a fully coupled SA. One important feature of this problem is that the data used to construct the RSAs come from the linearly decoupled subsystems (CAs), instead of the costly SA.

The CAS design problem shown in Fig. 6 was first introduced by Sobieszczanski-Sobieski et al.[31] The problem contains a total of

11 design variables and 43 states. The physical problem consists of a cantilever beam subjected to static loads along the beam and to a dynamic excitation force applied at the free end. Two sets of actuators are placed at the free end of the beam to control both the lateral and rotational displacement.

The SA comprises two fully coupled CAs. The structures subsystem CA_s consists of a finite element model of the beam where the natural frequencies and modes of the cantilever beam are computed. CA_s requires, in addition to the characteristics of the beam, the weight of the control system as input. The weight of the control system is calculated in the controls subsystem CA_c. The weight of the control system is a function of the dynamic displacements and rotations of the free end of the beam. These dynamic displacements and rotations are functions of the natural frequencies and modes obtained in the structures CA, thus subjecting these CAs to coupling.

The objective of the optimization is to minimize the total weight of the system W_t, comprising the weight of the beam W_s plus the weight of the control system W_c. The minimization is subject to seven constraints on the static stresses σ; static lateral and rotational displacements, d_l and d_r, respectively; the first two natural frequencies ω_1 and ω_1; and dynamic lateral and rotational displacements at the free end of the beam, d_{dl} and d_{dr}, respectively. The problem is posed as follows:

Minimize

$$W_t = W_s + W_c$$

subject to

$$g_1 = 1 - d_l/d_{l_a} \geq 0, \qquad g_2 = 1 - d_r/d_{r_a} \geq 0$$

$$g_3 = \omega_1/\omega_{1_a} - 1 \geq 0, \qquad g_4 = \omega_2/\omega_{2_a} - 1 \geq 0$$

$$g_5 = 1 - \sigma/\sigma_a \geq 0, \qquad g_6 = 1 - d_{\mathrm{dl}}/d_{\mathrm{dl}_a} \geq 0$$

$$g_7 = 1 - d_{\mathrm{dr}}/d_{\mathrm{dr}_a} \geq 0$$

1. *Implementation Details*

The trust region augmented Lagrangian framework for SAO developed by Rodríguez et al.[6,7] was implemented. The RSAs were constructed with medium-fidelity information queried by OAs of strength two.[8] This means that the sampling is performed over the decoupled CAs instead of sampling the costly SA. At the beginning of the optimization, FH approximations of the functions were computed. Transformation matrices were then computed for each of the decomposition strategies described earlier, that is, IT and CH. When a negative trust region ratio was encountered, an FH update was performed.

The problem has 11 design variables and, therefore, 66 Hessian terms. When an FH update was required, an OA with 128 points, 8 levels, and strength 2 was used. When only the 11 MD components were computed (either in the transformed or nontransformed space) smaller OAs were implemented. To asses the impact of the number of design points in the array, four OAs with an increasing number of design points were implemented: OA(27, 13, 3, 2), 27 design points, 13 design variables (2 were cut), 3 levels, strength 2; OA(36, 13, 3, 2), 36 design points, 13 design variables (2 were cut), 3 levels, strength 2; OA(50, 11, 5, 2), 50 design points, 11 design variables, 5 levels, strength 2, and the same OA(128, 11, 8, 2) used for the FH approximation. Note that an OA with more design variables than needed can be used because OAs keep their properties when factors are removed. In this case, a 13-design-variable OA is used to make an 11-design-variable array by ignoring two of its columns. [OA(27, 13, 3, 2), OA(36, 13, 3, 2), and OA(50, 11, 5, 2) can be found in Ref. 27. OA(128, 11, 8, 2) were generated using software by Owen.[26]]

As described by Owen,[26] one can randomize an OA to have different EAs. In this study, to avoid bias due to a particular EA from the randomization of the OAs, 10 runs were performed for each strategy and each OA. The results shown correspond to the average results of the 10 runs.

2. *Results for the CAS Problem*

The optimization was performed starting from two different points. The first is a feasible design $X_0 = (10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 0.1)$. The comparison of the four methodologies was done by evaluating the number of cycles required to converge (number of approximate minimizations) and the total number of CA calls required. The number of approximate minimizations indicates how the AED will impact the quality of the RSA. However, to evaluate the cost of the optimization, the number of CA calls is more important because each CA can be computationally expensive.

The number of approximate minimizations is shown in Fig. 7. There is no significant change in the number of approximate minimizations required to converge the SAO. The CH strategy with the OA(27) required more than 35 iterations to converge. On the other hand, using IT with OA(36) required the least number of approximate minimizations. We have to highlight that, contrary to what was expected, for the IT, the use of greater number of design points increased the number of approximate minimizations. This means more points did not increase the quality of the approximation and reduce the number of approximate minimizations. The results, in general, illustrate that an order $\mathcal{O}(n)$ update for this problem works well without effecting the overall performance of the optimization.

Figure 8 shows the results for the total cost of the optimization. One has to compare the cost of each run with the total cost of using an FH update. An overall decrease of up to 35% in the total number of CA calls required to converge can be seen. It is important to remember that no special heuristic, other than a negative trust region ratio, was used to decide when to perform an FH update. Thus, a further reduction could be possible with a more elaborated update criterion.

The second starting design point, $P_2 = (5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 0.1)$ is infeasible. Results are shown in Figs. 9 and 10. For each OA, the IT required more iterations to converge than the CH, except in the case of OA(128). The savings in the number of CA calls are more significant for this starting point.

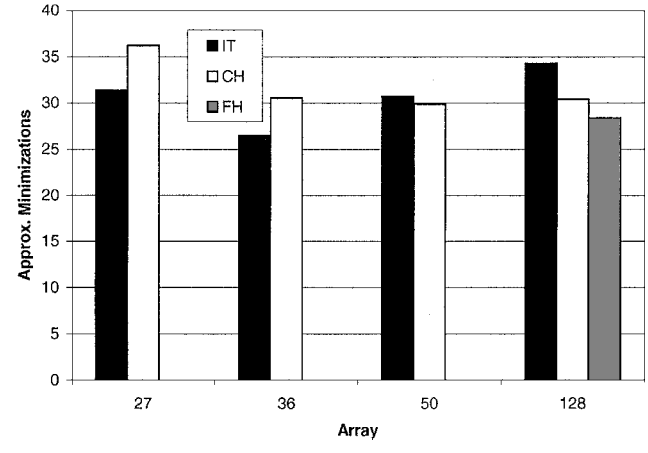

Fig. 7 Starting from a feasible point, approximate minimizations, CAS problem.

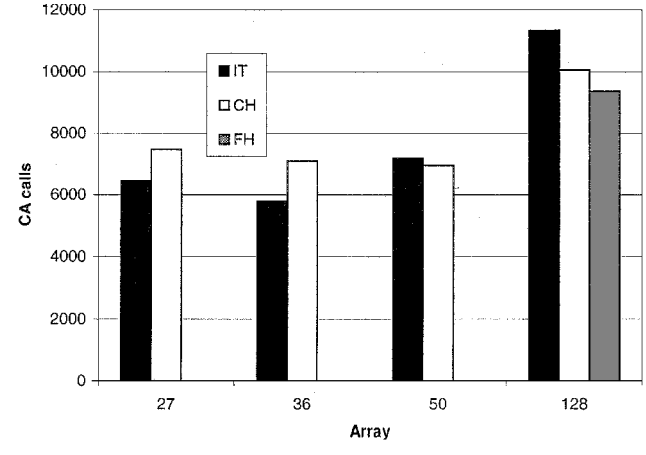

Fig. 8 Starting from a feasible point, CA calls, CAS problem.

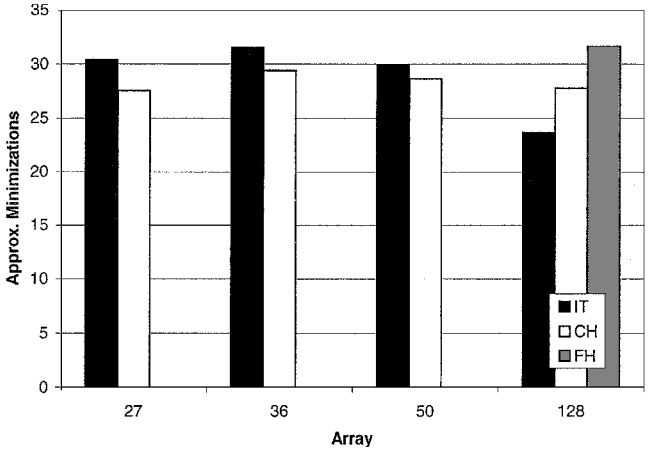

Fig. 9 Starting from an infeasible point, approximate minimizations, CAS problem.

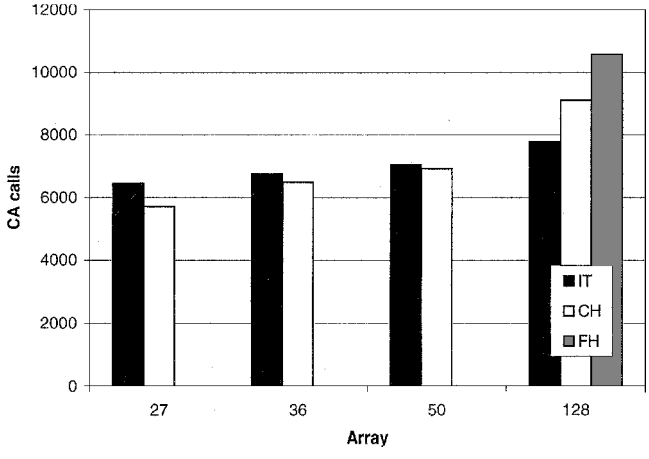

Fig. 10 Starting from an infeasible point, CA calls, CAS problem.

The smallest number of calls is required when using the OA(27); however, OA(36) and OA(50) do not significantly increase the number of CA calls required for SAO convergence.

VI. Conclusions

In this paper, a methodology that treats the curse of dimensionality by reducing the size of the database required for building a quadratic RSA is presented. The method, named AED, makes use of a canonical transformation of the preceding FH matrix to capture the geometric information. The AED only requires computation of the MD terms in the transformed space, thereby reducing the amount of data required to build a quadratic approximation. Two variants of the methodology are presented. In one, called the individual transformation method, the transformation is applied individually to the objective function and each of the constraints. An alternate method that computes the CH of the augmented Lagrangian and uses it to define the transformation is also presented. The methodology is applied to three test problems. The results are compared to those obtained by fitting the FH matrix at each iteration. Results show that the methodology can be applied to engineering problems, significantly reducing the amount of data required to fit a full quadratic function without an increase in the number of optimization iterations.

In a simple unconstrained example problem, the use of the proposed AED outperformed significantly the use of a Hessian with neglected off-diagonal terms. A fully constrained problem and an MDO problem showed an important reduction in the total cost of the optimization. The reduction is expected to increase with the number of design variables in the problem because the AED reduces the number of parameters to be fitted from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ for most of the iterations. No significant difference has been observed between the two decomposition strategies IT and CH. Future research will focus on generating a full-order reduction in the required database for all iterations in the SAO process.

Appendix: Coefficients for the Barnes Problem

Table A1 Coefficients and their values for the Barnes problem[30]

Coefficient	Value
a_1	75.196
a_2	−3.8112
a_3	0.12694
a_4	$-2.0567e-3$
a_5	$1.0345e-5$
a_6	−6.8306
a_7	0.030234
a_8	$-1.28134e-3$
a_9	$3.5256e-5$
a_{10}	$-2.266e-7$
a_{11}	0.25645
a_{12}	$-3.4604e-3$
a_{13}	$1.3514e-5$
a_{14}	−28.106
a_{15}	$-5.2375e-6$
a_{16}	$-6.3e-8$
a_{17}	$7.0e-10$
a_{18}	$3.4054e-4$
a_{19}	$-1.6638e-6$
a_{20}	−2.8673
a_{21}	0.0005

Acknowledgments

This multidisciplinary research effort is supported in part by the following grants and contracts: National Science Foundation Grants DMI98-12857 and DMI01-14975 and NASA Grant NAG1-2240. Victor Pérez was also partly funded by the Fulbright Commission and the Consejo Nacional de Ciencia y Tecnología, México.

References

[1]Myers, R. H., and Montgomery, D. C., *Response Surface Methodology. Process and Product Optimization Using Designed Experiments*, Wiley, New York, 1995.

[2]Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, Pt. 1, 1989, pp. 51–78.

[3]Renaud, J. E., and Gabriele, G. A., "Improved Coordination in Nonhierarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2367–2373.

[4]Renaud, J. E., and Gabriele, G. A., "Approximation in Nonhierarchic System Optimization," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 198–205.

[5]Wujek, B. A., Renaud, J. E., and Batill, S. M., "A Concurrent Engineering Approach for Multidisciplinary Design in a Distributed Computing Environment," *Multidisciplinary Design Optimization: State-of-the-Art*, edited by N. Alexandrov and M. Y. Hussaini, Vol. 80, Proceedings in Applied Mathematics Series, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 189–208.

[6]Rodríguez, J. F., Renaud, J. E., and Watson, L. T., "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data," *Structural Optimization*, Vol. 15, No. 3–4, 1998, pp. 141–156.

[7]Rodríguez, J. F., Renaud, J. E., and Watson, L. T., "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization," *Journal of Mechanical Design*, Vol. 120, No. 1, 1998, pp. 58–66.

[8]Rodríguez, J. F., Pérez, V. M., Padmanabhan, D., and Renaud, J. E., "Sequential Approximate Optimization Using Variable Fidelity Response Surface Approximations," *Structural and Multidisciplinary Optimization*, Vol. 22, No. 1, 2001, pp. 24–34.

[9]Balabanov, V., Kaufman, M., Knill, D. L., Haim, D., Golovidov, O., Giunta, A. A., Haftka, R. T., Grossman, B., Mason, W. H., and Watson, L. T., "Dependence of Optimal Structural Weight on Aerodynamic Shape for a High Speed Civil Transport," *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1996, pp. 599–612.

[10]Chen, W., Schrage, D. P., Allen, J. K., and Mistree, F., "Statistical Experimentation for Affordable Concurrent Design," *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1996.

[11]Giunta, A. A., Dudley, J. M., Narducci, R., Grossman, B., Haftka, R. T., Mason, W. H., and Watson, L. T., "Noisy Aerodynamic Response

and Smooth Approximations in HSCT Design," *Proceedings of the Fifth AIAA/USAF/NASA/ISSMO Symposium*, AIAA, Washington, DC, 1994, pp. 1117–1128.

[12]Giunta, A., Watson, L. T., "Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1998, pp. 392–404.

[13]Simpson, T. W., Peplinski, J., Koch, P. N., and Allen, J. K., "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," *Proceedings of the 1997 ASME Design Engineering Technical Conference* [CD-ROM], American Society of Mechanical Engineers, Fairfield, NJ, 1997.

[14]Pérez, V. M., Renaud, J. E., and Gano, S. E., "Constructing Variable Fidelity Response Surface Approximations in the Usable Feasible Region," *Proceedings of the 8th AIAA/NASA/USAF Multidisciplinary Analysis and Optimization Symposium* [CD-ROM], AIAA, Reston, VA, 2000.

[15]Wujek, B. A., and Renaud, J. E., "New Adaptive Move-Limit Management Strategy for Approximate Optimization, Part 1," *AIAA Journal*, Vol. 36, No. 10, 1998, pp. 1911–1921.

[16]Wujek, B. A., and Renaud, J. E., "New Adaptive Move-Limit Management Strategy for Approximate Optimization, Part 2," *AIAA Journal*, Vol. 36, No. 10, 1998, pp. 1922–1937.

[17]Pérez, V. M., and Renaud, J. E., "Decoupling the Design Sampling Region from the Trust Region in Approximate Optimization," *Proceedings of the International Mechanical Engineering Congress and Exposition*, Vol. 63, American Society of Mechanical Engineers, New York, 2000, pp. 205–214.

[18]Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing Use of Approximation Models in Optimization," *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16–23.

[19]Sobieszczanski-Sobieski, J., "A Linear Decomposition Method for Large Optimization Problems—Blueprint for Development," NASA TM-83248-1982, 1982.

[20]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435.

[21]Alexandrov, N. M., "Managing the Use of Surrogates in General Nonlinear Optimization and MDO," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium*, AIAA, Reston, VA, 1998, pp. 720–729.

[22]Dennis, J. E., and Torczon, V., "Managing Approximation Models in Optimization," *Multidisciplinary Design Optimization: State-of-the-Art*, edited by N. Alexandrov and M. Y. Hussaini, Proceedings in Applied Mathematics Series, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 330–347.

[23]Lewis, M. R., "Trust Region Framework for Managing Approximation Models in Engineering Optimization," *Proceedings of the 6th AIAA/NASA/USAF Multidisciplinary Analysis and Optimization Symposium*, AIAA, Reston, VA, 1996, pp. 1053–1055.

[24]McKay, M. D., Conover, W. J., and Beckman, R. J., "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[25]Tang, B., "Orthogonal Array-Based Latin Hypercubes," *Journal of the American Statistical Association*, Vol. 88, No. 424, 1993, pp. 1392–1397.

[26]Owen, A. B., "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, Vol. 2, No. 2, 1992, pp. 439–452.

[27]Hedayat, A. S., Sloane, N. J. A., and Stufken, J., *Orthogonal Arrays. Theory and Applications*, Springer Series in Statistics, Springer-Verlag, New York, 1999.

[28]St. John, R. C., and Draper, N. R., "D-Optimality for Regression Design: A Review," *Technometrics*, Vol. 17, 1975, pp. 15–23.

[29]Pukelsheim, F., *Optimum Design of Experiments*, Chapman and Hall, New York, 1995.

[30]Barnes, G. K., M.S. Thesis, Univ. of Texas, Austin, TX, 1967; also *Applied Nonlinear Programming*, edited by D. M. Himmelblau, McGraw–Hill, New York, 1972, pp. 344–395.

[31]Sobieszczanski-Sobieski, J., Bloebaum, C. L., and Hajela, P., "Sensitivity of Control-Augmented Structure Obtained by a System Decomposition Method," *AIAA Journal*, Vol. 29, No. 2, 1991, pp. 264–270.

A. Messac
Associate Editor