A Coevolutionary Architecture for Distributed Optimization of Complex Coupled Systems*

Prasanth B. Nair and Andrew J. Keane

Computational Engineering and Design Center, School of Engineering Sciences, University of Southampton, Highfield, Southampton SO17 1BJ.

E-mail: {P.B.Nair, Andy.Keane}@soton.ac.uk

Submitted September 1999; revised January, 2001

This paper presents a coevolutionary architecture for distributed optimization of complex coupled systems. This architecture is inspired by the phenomena of coevolutionary adaptation occurring in ecological systems. The focus of this research is to develop flexible design architectures for addressing the organizational and computational challenges involved in optimization of large-scale multidisciplinary systems. In the proposed design architecture, the optimization procedure is modeled as the process of coadaptation between sympatric species in an ecosystem. Each species is entrusted with the task of improving sub-domain specific objectives and the satisfaction of sub-domain constraints. Coupling compatibility constraints are accommodated via implicit generalized Jacobi iteration, which enables the application of the proposed architecture to systems with arbitrary coupling bandwidth between the disciplines, without an increase in the problem size. A domain decomposition approach is presented for distributed structural optimization to construct a class of test problems. Numerical studies are presented to demonstrate that convergence to an optimal solution satisfying the sub-domain and coupling compatibility constraints can be readily achieved.

Key Words: Multidisciplinary design optimization, coupled systems, evolutionary computation, coevolutionary algorithms, structural optimization, domain decomposition.

1. INTRODUCTION

Multidisciplinary design optimization (MDO) is an enabling methodology for the design of complex systems, the physics of which involve couplings between various

*To appear in *AIAA Journal*, 2002. A preliminary version of this paper was published by the American Institute of Aeronautics and Astronautics as AIAA Paper 1999-1260

interacting disciplines/phenomena. The underlying focus of MDO methodology is to develop formal procedures for exploiting the synergistic effects of the coupling in the problem physics at every stage of the design process. One advantage offered by this methodology is the achievement of calendar time compression via concurrency in the product design and development cycle. It also allows for tradeoff studies between performance, manufacturing, supportability, economics, and life cycle issues to be conducted at all stages of the design process. Hence, the adoption of MDO methodology is not only expected to lead to better designs, but also the establishment of a more physically meaningful design practice as compared to the traditional sequential approach in which the synergistic effects of coupling are ignored.

A recent review of the state of the art in MDO has been presented by Sobieszczanski-Sobieski and Haftka.[1] It was noted in this review that "organizational challenges" and "computational cost" are generally perceived as the major obstacles to the application of MDO methodology in design practice.

A good deal of earlier research work in MDO has focused on system optimization approaches, also known as multidisciplinary feasible methods; see, for example, reference.[2] These approaches require a suite of highly integrated disciplinary analysis tools connected to a single optimizer. The major disadvantages of this line of approach include : (1) the high costs involved in software integration and maintenance of the integrated design system, (2) specialist disciplinary groups do not have any decision making power in the design process, and (3) the increase in computational cost and potential convergence problems due to the use of a single optimizer to handle all the design variables. Another important factor is the massive computational cost incurred in such MDO approaches due to the requirement of iterations between the disciplines to arrive at a multidisciplinary feasible solution at each function evaluation.

These drawbacks to system optimization approaches, coupled with the ever increasing requirement of addressing the organizational and computational challenges in MDO have motivated the development of distributed optimization architectures. Example architectures on which research has recently been pursued with particular vigor are concurrent subspace optimization (CSSO)[3,4] and collaborative optimization (CO).[5−7] These architectures enable the solution of large-scale MDO problems in a distributed fashion and retain the advantages of division of labor.

Such multilevel optimization approaches[8] to MDO in general involve the use of a system level optimizer, which guides a number of disciplinary optimizers to improve the overall system performance, satisfy the disciplinary constraints, and ensure multidisciplinary feasibility of the converged solution. Since multidisciplinary feasibility need not be enforced at each iteration, this approach circumvents the problem of "disciplinary sequencing", in which some of the disciplines have to wait for data from the other disciplines before carrying out their design studies. Hence, distributed optimization allows for the possibility of tackling MDO problems in a concurrent fashion (thereby achieving calendar-time compression), and also retains the autonomy of disciplinary specialists in the design process. The latter advantage is widely acknowledged to be a crucial factor in the acceptance of formal MDO methods by industry. Distributed MDO methods are also well suited to the hier-

archical organizational structure and heterogeneous computing platforms found in many large industries.[5]

Following the discussion presented in Alexandrov and Kodiyalam[9], MDO methods can be broadly classified on the basis of the design architecture or formulation, the optimization algorithms used for design space search (DSS), methods used for constructing approximation models to accelerate DSS, and the framework employed for managing variable-fidelity analysis models. The area of MDO architectures is closely related to the fields of system decomposition and optimization. The research work reported here focuses on the first two points, i.e., the MDO architecture and the optimization algorithm.

Recently, stochastic nongradient optimization techniques such as evolutionary algorithms (EAs) have been successfully applied to some nonconvex design optimization problems; see, for example, references.[10–12] Even though this class of optimization techniques offer many advantages as compared to gradient search, the number of function evaluations required to converge to an optimal solution is considered prohibitive for many problems. This problem of high computational cost has led to very few applications of EAs to MDO using high-fidelity analysis models. In reference[11], a feedforward neural network was first constructed to map the design space before the optimization studies, with inevitable restrictions on the scope and fidelity of the final optimum. Hence, when the difficulties of high computational cost are coupled with nonconvex design spaces, standard EA approaches are impractical and more advanced strategies are required. The development of computational frameworks for construction and management of approximation models to accelerate EAs has been the focus of some recent research; see, for example, references.[13,14] It is expected that the computational cost of using EAs could be significantly reduced by using such DSS strategies.

As a consequence of the high computational cost involved in using EAs for DSS, its application to multilevel optimization becomes computationally prohibitive. A genetic algorithm (GA) approach based on predator-prey coadaptation was applied to bilevel optimization problems by Venter and Haftka.[15] It was shown that this approach could lead to savings in the computational cost. The decomposition technique proposed by Lee and Hajela[11] could also be interpreted as a coevolutionary search strategy. These techniques fall under the broad banner of coevolutionary computation. An excellent description of the theoretical aspects of coevolutionary computation and results for some application areas can be found in the dissertation of Potter.[16]

This paper introduces the paradigm of coevolutionary adaptation in the context of distributed optimization of complex coupled systems. Inspired by this paradigm, it is shown that the design process can be modeled as the process of coadaptation between *sympatric* species in an ecosystem, i.e., species which live in the same place, rather than being geographically isolated. Each discipline is modeled as a species which collaborates with the other species to improve the discipline specific objectives and satisfy local constraints. Here, in order to allow applicability to systems with arbitrary coupling bandwidth, the *coupling variables* are not explicitly represented as design variables. The implicit, generalized Jacobi iteration strategy used in the present research allows for the satisfaction of the coupling compatibility con-

straints at the optima under rather mild assumptions. Further, it is assumed that the design variables are decomposed into disjoint sets, i.e., the *disciplinary species* are responsible for evolving independent sets of design variables. A *Coevolutionary Genetic Algorithm* (CGA) is then used to model the patterns of interaction between the various species.

A domain decomposition scheme is employed to construct test problems for studying the performance of the proposed coevolutionary MDO (CMDO) architecture. Results are presented for structural optimization problems, wherein the concept of substructuring can be used to reformulate the problem in terms of coupled systems. It is shown that the CMDO architecture successfully converges to an optimal solution satisfying the disciplinary as well as coupling compatibility constraints. The effects of coupling bandwidth and increase in problem size on the performance of the CMDO architecture are also examined.

2. THE PRESENT APPROACH

The main focus of the research work reported here is the development of flexible MDO architectures, with the specific aims of - (1) disciplinary autonomy in both analysis as well as optimization, and (2) applicability to systems with arbitrary coupling bandwidth without an increase in the problem size. The features of a canonical CGA are first described in order to delineate the fundamentals underpinning the CMDO architecture. In the subsequent sections, the issues involved in application of this coevolutionary search paradigm to distributed optimization of coupled systems are addressed.

2.1. Coevolutionary Genetic Algorithms (CGAs)

A CGA models an ecosystem consisting of two or more sympatric species having an ecological relationship of mutualism. Consider the problem involving maximization of $f(\mathbf{x})$, which is a function of n variables. Potter and Dejong[17] used an approach in which the problem is decomposed into n species - one species for each variable. Hence, each species contain a population of alternative values for each variable. *Collaboration* between the species involves selection of representative values of each variable from all the other species and combining them into a vector which is then used to evaluate *f(x)*. An individual in a species is rewarded based on how well it maximizes the function within the context of the variable values selected from the other species. This procedure is analogous to the univariate optimization algorithm wherein 1-D search is carried out using only one free variable. Note here that it is also possible to decompose the original problem variables into several blocks, with each species evolving a block rather than a single variable; see, for example, reference.[13]

The steps involved in a canonical CGA (see also Figure 1) may hence be summarized as:

Step 1 : Initialize a population of individuals for each species randomly or based on some previous design knowledge.

Step 2 : Evaluate the fitness of the individuals in each species. Fitness evaluation in a CGA involves the use of the following algorithm:

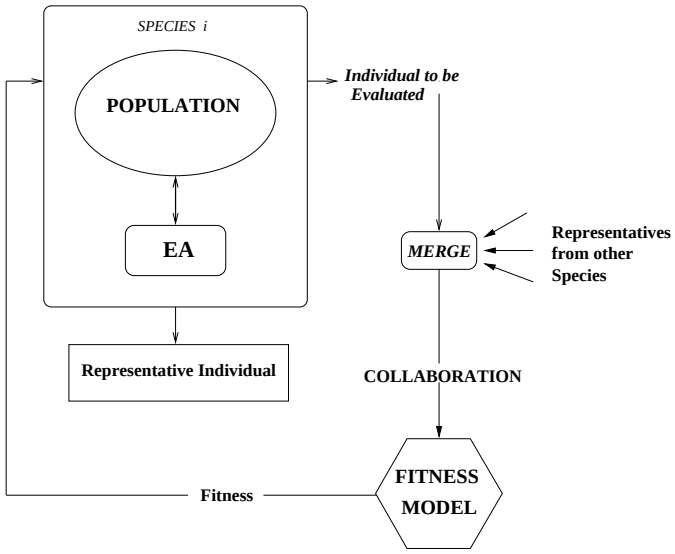

FIG. 1. Flow chart of Coevolution of a Species within a Collaborative Process.

Choose representatives from all other species.

FOR each individual i in the species being evaluated **DO**

- Form collaboration between individual i and the (fixed) representatives from other species.
- Evaluate the fitness of the collaborative design by applying it to the target problem, and assign it to the individual i.

ENDDO

Step 3 : If the termination criteria is not met, then apply a canonical GA involving the operators of reproduction and genetic recombination to arrive at a new population for each species. Go to Step 2.

Potter and Dejong[17] showed that a CGA approach to function optimization could lead to faster convergence as compared to a conventional GA, for low to moderate levels of variable interdependencies (also referred to as *epistasis*[‡] in the GA literature). This can be primarily attributed to the attendant reduction in the active search space due to coevolution of each variable concurrently, e.g., for the function optimization problem described earlier the original search space of size $(2^k)^n$ has

[‡]In the context of function optimization, epistasis can be defined as the extent to which the contribution of one variable to the fitness depends on the values of the other variables.

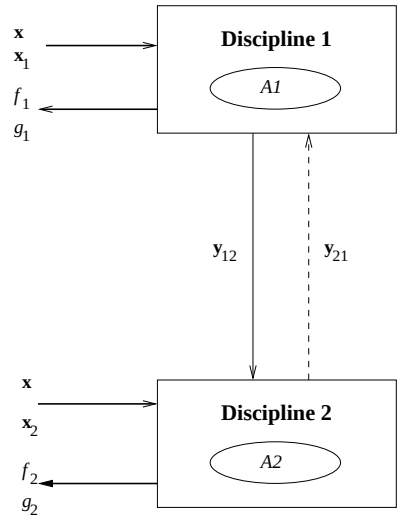

FIG. 2. Data Flow Diagram for a Typical MDO Problem.

been reduced to a series of n constrained search spaces, each of size 2^k, where k is the number of bits used to represent each variable. The underlying premise of the CGA approach is that, when the optimization problem under consideration has a moderate degree of epistasis, faster progress in the search can be made by decomposing the design space.

In CGA-based search, the term *generation* is used to refer to a cycle involving selection, recombination and fitness evaluation for a single species, and the term *ecosystem generation* refers to an evolutionary cycle through all the species being coevolved. The reader is referred to Potter[16] for a more detailed description of the theoretical aspects of CGAs. Further details, including applications of coevolutionary algorithms can be found in the literature.[13,18–20]

2.2. Analysis of Coupled Systems

To illustrate the design architecture developed here, consider a coupled system involving two disciplines (or subsystems) shown in Figure 2. The coupling shown here is typical of aerodynamics-structures interaction encountered in aeroelastic analysis of flexible aircraft wings. For this problem, $\mathbf{x}$ denotes the vector of multidisciplinary design variables (i.e., variables which are common to both disciplines). $\mathbf{x}_1$ and $\mathbf{x}_2$ are the vectors of disciplinary design variables corresponding to subsystem 1 and 2, respectively. f_1 and f_2 are the objective functions corresponding to discipline 1 and 2, respectively. g_1 and g_2 are the constraint functions corresponding to discipline 1 and 2, respectively. $\mathbf{y}_{12}$ is the vector of coupling variables computed in discipline 1 which is needed for analysis of subsystem 2. Similarly, $\mathbf{y}_{21}$ denotes the vector of coupling variables computed in discipline 2 which is required for analysis of subsystem 1.

Note here that arriving at a multidisciplinary feasible solution for given values of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}$, would involve iterating between both the disciplines using an initial

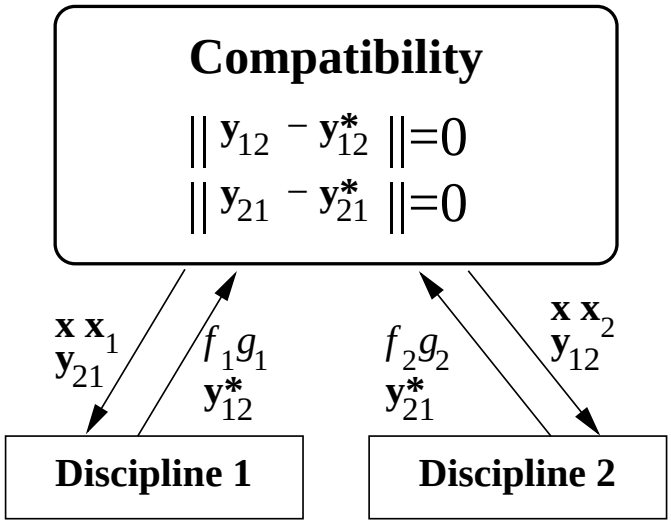

FIG. 3. Distributed Multidisciplinary Analysis.

guess for either $\mathbf{y}_{12}$ or $\mathbf{y}_{21}$. This sequential iterative scheme can be interpreted as a Generalized Gauss-Seidel (GGS) approach; see Arian[21] for a detailed exposition.

The state equations for the two disciplines can be mathematically expressed as :

$$A1(\mathbf{x}, \mathbf{x}_1, \mathbf{u}_1, \mathbf{y}_{21}(\mathbf{u}_2)) = 0, \tag{1}$$

and

$$A2(\mathbf{x}, \mathbf{x}_2, \mathbf{u}_2, \mathbf{y}_{12}(\mathbf{u}_1)) = 0, \tag{2}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the state variables for subsystem 1 and 2, respectively.

In order to eliminate the couplings between the disciplinary analysis modules, the original system can be decomposed as shown in Figure 3. For given values of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}$, multidisciplinary analysis now requires an initial guess for both $\mathbf{y}_{12}$ and $\mathbf{y}_{21}$. Iterations involving aerodynamic and structural analysis converge to a multidisciplinary feasible solution when the coupling compatibility constraints are satisfied, i.e., when $||\mathbf{y}_{12} - \mathbf{y}^*_{12}|| = 0$ and $||\mathbf{y}_{21} - \mathbf{y}^*_{21}|| = 0$. Here $\mathbf{y}_{12}$ and $\mathbf{y}_{21}$ are the values of the coupling variables used at the current iteration, and $\mathbf{y}^*_{12}$ and $\mathbf{y}^*_{21}$ are the new values of coupling variables after disciplinary analysis using the current values of $\mathbf{y}_{21}$ and $\mathbf{y}_{12}$, respectively. This decomposition of the original coupled system makes it possible to concurrently analyze a design and thus perform distributed optimization. As discussed in Arian[21], this parallel iterative scheme for multidisciplinary analysis can be interpreted as a Generalized Jacobi (GJ) approach. Further, it was also shown that the GJ scheme will take more iterations than the GGS scheme to converge.

2.3. Some Issues in Coevolutionary MDO

The issues which need to be addressed in order to apply a coevolutionary search strategy to optimization of this coupled system are discussed next.

[A] *Problem Decomposition* : This issue is mainly concerned with how the coupled system should be decomposed into various species. A natural procedure for decomposition would be to divide the design variables into groups on disciplinary lines. The chromosome of each species consists of the variables it is allowed to control. In general, it may be preferable to decompose the multidisciplinary design variable vector $\mathbf{x}$ into completely disjoint sets to circumvent the difficulty of arriving at a consensus on the independently evolved multidisciplinary variables.

[B] *Choice of Representative Individuals* : As mentioned earlier in the description of CGAs, the species interact with each other via representative individuals. The evolution of each species is thus constantly driven by evolutionary changes in the species it interacts with. In the genetics literature, this is referred to as the Red Queen hypothesis, wherein each species must constantly adapt just to remain in parity with the others. Hence, a fundamental issue in coevolutionary computation is how to choose the representative individuals from each species.

In reference[15], two strategies were suggested for choosing representative individuals. The first strategy was to select the individual with the highest fitness in a species as its sole representative. In the second strategy, two representative individuals were chosen. The first representative was the individual with highest fitness while the second representative was chosen randomly. Both the representatives are used for fitness evaluations, and the maximum fitness of the two possible collaborations is assigned to the individual under consideration. Notice that this will lead to the requirement of 2^{m-1} fitness evaluations for each individual in a discipline, where m is the total number of species. It was found via numerical experiments that the first strategy, although superior in terms of convergence speed, may not be robust for problems with high epistasis. In contrast, the second strategy is more robust at the expense of slower convergence. Of course the number of representatives need not be the same for each species nor remain fixed during optimization. However, the computational cost of fitness evaluation grows with increase in the number of representatives, since a domain specific analysis must be carried out for each combination of collaborating representatives considered.

A comparison study of strategies for choosing representative individuals in coevolutionary computation can be found in Bull.[22] In the present research, the individual with the highest fitness in a species is chosen as its representative.

[C] *Coupling Variables* : A fundamental issue in distributed optimization of coupled systems is how to ensure multidisciplinary feasibility of the optimal solution. Existing MDO architectures such as CSSO and CO explicitly treat the coupling variables as auxiliary design variables. The major disadvantage of this strategy is that the problem size grows significantly with increase in the coupling bandwidth. In the present research, the aim is to circumvent this fundamental difficulty by treating the coupling variables implicitly in the optimization process. Details of the strategy used for handling the coupling variables are described in subsequent sections. In contrast to other distributed MDO formulations, a major advantage

of the the present approach is that it can therefore tackle problems with arbitrary coupling bandwidth without an attendant increase in the problem size.

[D] *Fitness Evaluation* : The fitness of the disciplinary species is defined as a function of the respective disciplinary objectives and constraints. In the present research, a penalty function approach is used for handling constraints. Alternative constraint handling strategies for evolutionary algorithms are available in the literature.[23,24]

3. THE COEVOLUTIONARY MDO ARCHITECTURE

Based on the earlier discussion, various MDO architectures adopting different considerations for tackling the coupling compatibility constraints and problem decomposition characteristics can be developed within the framework of the coevolutionary genetic adaptation paradigm. However, to ensure applicability to systems with broad coupling bandwidth, here the *coupling variables* are not explicitly considered as design variables. The coupling variables are instead initialized to values computed from system analysis of a baseline design. In the subsequent stages of the coadaptation procedure, the disciplinary species exchange values of the coupling variables computed during the disciplinary analysis of their respective *representative individuals*. This implicit iteration scheme reduces the extent of violation of the coupling compatibility constraints as the coevolution reaches stasis.

Further, in the architecture developed here, the design variables are decomposed into disjoint sets, and hence no variables are shared between the disciplines. The chromosome of the disciplinary species are hence composed of independent sets of variables. The chromosome representation and fitness evaluation details for both the species are summarized below. Here, the variables $\mathbf{x}^1$ and $\mathbf{x}^2$ are the disjoint sets formed from decomposing the vector of multidisciplinary variables $\mathbf{x}$, i.e., all the elements of $\mathbf{x}$ are placed in $\mathbf{x}^1$ or $\mathbf{x}^2$.

Disciplinary Species 1

Genotype: $[\mathbf{x}^1, \mathbf{x}_1]$, i.e., the first part of the multidisciplinary design vector and the variables local to discipline 1.

Fitness Evaluation: Computation of the objective and constraint functions for discipline 1 (f_1 and g_1) requires the values of $\mathbf{x}^1$, $\mathbf{x}_1$, $\mathbf{x}^2$, and $\mathbf{y}_{21}$. The values of $\mathbf{x}^1$ and $\mathbf{x}_1$ are readily available since they are directly controlled by species 1. In the first ecosystem generation, the value of $\mathbf{y}_{21}$ is initialized either from system analysis of a baseline design or randomly. Similarly, $\mathbf{x}^2$ could be initialized randomly in the first ecosystem generation, or set to the value corresponding to a baseline design. In the subsequent ecosystem generations, the values of $\mathbf{y}_{21}$ and $\mathbf{x}^2$ provided by discipline 2 for their representative individual are used.

Disciplinary Species 2

Genotype: $[\mathbf{x}^2, \mathbf{x}_2]$, i.e, the second part of the multidisciplinary design vector and the discipline specific variables.

Fitness Evaluation: To compute the objective and constraint functions for discipline 2 (f_2 and g_2), the values of $\mathbf{x}^2$, $\mathbf{x}_2$, $\mathbf{x}^1$, and $\mathbf{y}_{12}$ are required. The values of $\mathbf{x}^2$ and $\mathbf{x}_2$ are readily available. The value of $\mathbf{y}_{12}$ computed via system analysis

of a baseline design (or randomly initialized values) is used in the first ecosystem generation. The subset of multidisciplinary variables controlled by discipline 1, i.e., $\mathbf{x}^1$ is either initialized randomly or set to the baseline values. In the subsequent ecosystem generations, the values of $\mathbf{y}_{12}$ and $\mathbf{x}^1$ provided by discipline 1 for their representative individual are used.

It can be observed that the implicit procedure used here to ensure satisfaction of the interdisciplinary coupling compatibility constraints turns out to be similar to the GJ scheme discussed earlier for distributed multidisciplinary analysis. It is known that the GJ scheme may fail to converge for some problems; see, for example, Arian.[21] Hence, it becomes important to examine when the the implicit procedure used in the CMDO architecture may fail. The approach used here for satisfying the coupling constraints may be interpreted as a *randomly restarted* generalized Jacobi iteration scheme. The term 'random' is used here to indicate that the coupling variables $\mathbf{y}_{12}$ and $\mathbf{y}_{21}$ are reset based on the progress of the coevolution procedure, i.e. they may change radically as the representatives change. However, as the coevolutionary process reaches stasis (i.e., as $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}$ converges), the implicit procedure reduces to the conventional GJ scheme. For the example problems considered in this paper, it is observed that an optimal solution satisfying the interdisciplinary compatibility constraints can be readily found. However, for problems where difficulties arise in satisfying the compatibility constraints, alternative iterative schemes (such as successive over relaxation or Newton-based algorithms[25]) may have to be employed in the later stages of the coevolutionary search.

3.1. Data Coordination, Surrogate Modeling, Decomposition, and Other Issues

A common blackboard model for the coupling variables is used for transferring data between the disciplines. The disciplinary species post the values of the coupling variables and multidisciplinary design variables corresponding to their representative individuals on this blackboard during the coadaptation procedure. It is to be noted here that, since EAs constitute a global search paradigm, high quality space-filling computational data can potentially be obtained as a by-product of the coevolution procedure. Such data can be used for constructing simulation metamodels to accelerate disciplinary DSS and archive the design space - see reference[14] for an approach to construction and management of metamodels in evolutionary optimization algorithms. A more detailed discussion of problem solving environments for multidisciplinary design which bring together such capabilities has been presented in Keane and Nair.[26]

It is also of interest to examine the possibility of using the information generated during the coevolutionary adaptation procedure to make decisions on how the set of multidisciplinary design variables should be decomposed. This would allow for the possibility of the optimal problem decomposition structure to emerge rather than be fixed *a priori* by the designer. However, it is not clear at this stage of the research how to develop such an algorithm. An alternative procedure would be to decompose the multidisciplinary variables into disjoint sets by computing their main effects for the disciplinary objectives via orthogonal array-based experimental designs before starting the optimization.

Since the various species can interact via a single representative individual, the optimization algorithms which may be used within the CMDO architecture are not restricted to evolutionary methods alone. In fact, arbitrary optimization formulations/algorithms may be used within the disciplines.

The CMDO architecture allows two avenues for massive parallelization of the optimization process. Firstly, the evolution of the disciplinary species may be carried out concurrently. Further, there also exists the possibility of computing the fitness of the individuals in each species concurrently.

4. DISTRIBUTED STRUCTURAL OPTIMIZATION VIA SUBSTRUCTURING

To illustrate the application of some of these ideas in practice, a distributed structural optimization formulation based on domain decomposition is presented in this section. Consider the finite element mesh of a general structural system which is decomposed into two non-overlapping substructures by a solid line, as shown in Figure 4. The solid line passes through nodes which lie in substructure 1. The dashed line is drawn through the nodes in substructure 2 which are directly connected to these nodes in substructure 1. In Figure 4, N1 and N2 are the total number of degrees of freedom (dof) in substructure 1 and substructure 2, respectively. $\mathbf{x}_1$ and $\mathbf{x}_2$ are the vectors of local design variables corresponding to substructure 1 and substructure 2, respectively (e.g., geometry variables, material properties, etc). Note that $\mathbf{x}_1$ and $\mathbf{x}_2$ are disjoint sets. The variables corresponding to the interface region between the two substructures are denoted by $\mathbf{x}$, e.g., $\mathbf{x}$ might correspond to the area and geometry of the elements in the interface region.

The equations for static equilibrium of the structure in Figure 4 can be written in partitioned form as

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \tag{3}$$

where $\mathbf{K}_{11} \in \mathbb{R}^{N1 \times N1}$, $\mathbf{K}_{12} \in \mathbb{R}^{N1 \times N2}$, $\mathbf{K}_{21} \in \mathbb{R}^{N2 \times N1}$, and $\mathbf{K}_{22} \in \mathbb{R}^{N2 \times N2}$ are the submatrices of the partitioned global stiffness matrix, $\mathbf{d}_1 \in \mathbb{R}^{N1}$ and $\mathbf{d}_2 \in \mathbb{R}^{N2}$ are the displacements corresponding to the nodes of each substructure, and $\mathbf{f}_1 \in \mathbb{R}^{N1}$ and $\mathbf{f}_2 \in \mathbb{R}^{N2}$ are the external forces acting at the substructure degrees of freedom.

Equation (3) can be rewritten as two simultaneous matrix equations of the form

$$\mathbf{K}_{11}\mathbf{d}_1 = \mathbf{f}_1 - \mathbf{K}_{12}\mathbf{d}_2 \tag{4}$$

$$\mathbf{K}_{22}\mathbf{d}_2 = \mathbf{f}_2 - \mathbf{K}_{21}\mathbf{d}_1 \tag{5}$$

The second term on the right hand side of the above equations indicates a coupling between the displacements in the two substructures. In general, $\mathbf{K}_{12}$ and $\mathbf{K}_{21}$ are highly sparse matrices. Hence, only a few components of $\mathbf{d}_1$ and $\mathbf{d}_2$ are required to compute the coupling terms.

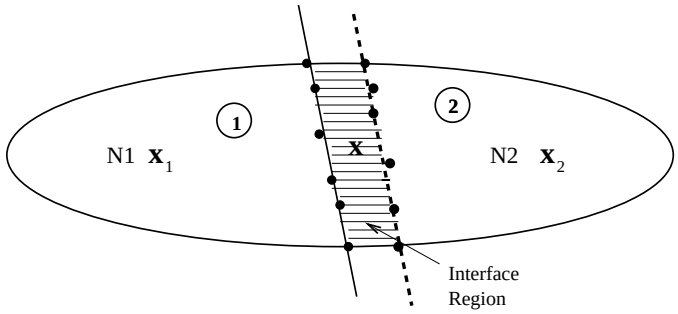

FIG. 4. Decomposition of Structural Domain.

The nature of the coupling between equations (4-5) is depicted more clearly by rewriting them as

$$\mathbf{A}1(\mathbf{d}_1, \mathbf{x}_1, \mathbf{x}, \mathbf{d}_{21}(\mathbf{d}_2)) = 0 : \mathbf{K}_{11}(\mathbf{x}_1, \mathbf{x})\mathbf{d}_1 = \mathbf{f}_1 - \mathbf{K}_{12}(\mathbf{x})\mathbf{d}_{21}(\mathbf{d}_2), \tag{6}$$

and

$$\mathbf{A}1(\mathbf{d}_2, \mathbf{x}_2, \mathbf{x}, \mathbf{d}_{12}(\mathbf{d}_1)) = 0 : \mathbf{K}_{22}(\mathbf{x}_2, \mathbf{x})\mathbf{d}_2 = \mathbf{f}_2 - \mathbf{K}_{21}(\mathbf{x})\mathbf{d}_{12}(\mathbf{d}_1), \tag{7}$$

where $\mathbf{A}1$ and $\mathbf{A}2$ symbolically denote the analysis equations to be solved for substructure 1 and 2, respectively. Solution of the analysis equations $\mathbf{A}1$ and $\mathbf{A}2$ lead to the displacement vectors $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively. $\mathbf{d}_{21}$ denotes the vector of displacements at those nodes in substructure 2, which are directly connected to the nodes in substructure 1, i.e., the nodes which lie on the dashed line in Figure 4. Similarly, $\mathbf{d}_{12}$ is the displacement vector at the nodes in substructure 1 which are directly connected to the nodes in substructure 2, i.e, the nodes which lie on the solid line in Figure 4. The coupling variables in equations (6-7) are $\mathbf{d}_{21}$ and $\mathbf{d}_{12}$. The coupling bandwidth resulting from the substructuring strategy would depend on the total number of dof corresponding to the elements in the interface region.

For given values of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}$, iterating between equations (6) and (7) from an initial guess for either $\mathbf{d}_{21}$ or $\mathbf{d}_{12}$ leads to a solution for the global displacement vector $\mathbf{d}$ which satisfies the physics of the problem. This implies satisfaction of force equilibrium at the interface region between the two substructures.

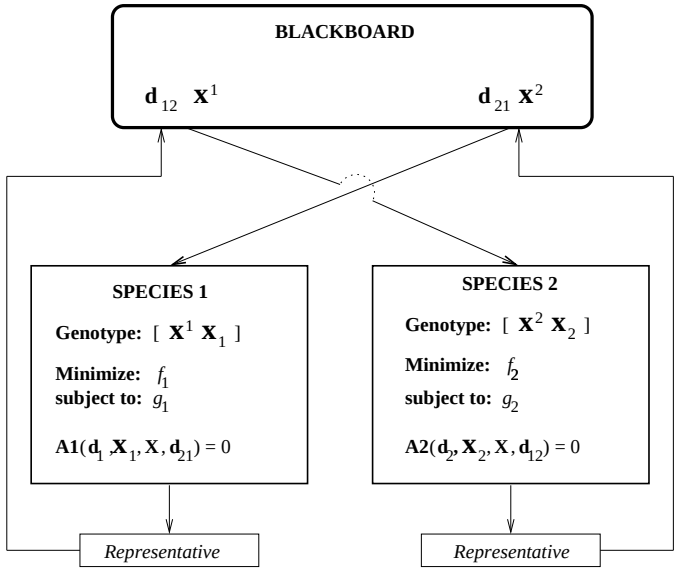

FIG. 5. Coevolutionary Optimization of Coupled Substructures

In the context of distributed analysis using the decomposition strategy discussed earlier (see Figure 3), the coupling compatibility constraints can be written as

$$||\mathbf{d}_{12} - \mathbf{d}^*_{12}|| + ||\mathbf{d}_{21} - \mathbf{d}^*_{21}|| = 0, \tag{8}$$

where $\mathbf{d}_{12}$ and $\mathbf{d}_{21}$ are the values of the coupling displacements at the current iteration, and $\mathbf{d}^*_{12}$ and $\mathbf{d}^*_{21}$ are the new values of the coupling variables after solving $\mathbf{A}1$ and $\mathbf{A}2$, respectively.

The domain decomposition approach used here enables the application of the CMDO architecture to distributed structural optimization. In particular, for the structure in Figure 4, the optimization problem can be posed as a simulated MDO problem with two coupled disciplines. The optimization procedure in this setting is shown in Figure 5. In the decomposition strategy chosen here, each species handle disjoint sets of design variables. This is achieved by decomposing the variables corresponding to the interface region ($\mathbf{x}$) into two disjoint sets $\mathbf{x}^1$ and $\mathbf{x}^2$. The first species controls the variables $\mathbf{x}_1$ and $\mathbf{x}^1$, while the second species controls the variables $\mathbf{x}_2$ and $\mathbf{x}^2$. The mesh corresponding to the interface region is modeled by both substructures, so as to enable it to compute the submatrices $\mathbf{K}_{12}$ and $\mathbf{K}_{21}$ independently.

In the first step of the distributed optimization procedure, the coupling variables $\mathbf{d}_{21}$ and $\mathbf{d}_{12}$ are either initialized using the results of system analysis of a baseline design, or randomly. Similarly, the design variable vector $\mathbf{x}$ common to both substructures can also be initialized. Analysis of the first subsystem involves the

solution of equation (6) for the substructure displacement vector $\mathbf{d}_1$. The analysis results can then be used to compute the objective and constraint functions of interest within substructure 1. Similarly, the second subsystem analysis involves the solution of equation (7) for the displacement vector $\mathbf{d}_2$. These results are used in conjunction with the current value of $\mathbf{d}_{12}$ to compute the objective and constraint functions for substructure 2.

In the subsequent ecosystem generations, each species posts the values of the coupling variables corresponding to its representative individual on a common blackboard. The values of $\mathbf{x}^1$ and $\mathbf{x}^2$ controlled by the two species independently are also posted on the blackboard. This facilitates timely exchange of data between the two species as they coevolve to optimum values of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}$. As shown in the subsequent section, this data transfer enables both the species to arrive at a solution which satisfies the coupling compatibility constraints, even though the coupling variables are not explicitly represented as design variables. In summary, the application of the CMDO architecture in conjunction with the substructuring strategy ultimately leads to the optimal values of the design variables, as well as the values of the substructure displacement vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ which satisfies the physics of the coupling.

This decomposition based coevolutionary optimization strategy leads to significant savings in the computational cost, particularly for the optimum design of large-scale structural systems. The procedure used here could of course be readily extended to scenarios with more than two substructures. It is also of interest to note that this procedure could be used for structures subjected to harmonic excitation. The only difference in the formulation would be to replace $\mathbf{K}_{ij}$ with $[\mathbf{K}_{ij} - \omega^2\mathbf{M}_{ij} + \mathrm{j}\omega\mathbf{C}_{ij}]$, where $\mathbf{M}_{ij}$ and $\mathbf{C}_{ij}$ are partitions of the mass and damping matrix, respectively, ω is the frequency of excitation, and $\mathrm{j} = \sqrt{-1}$.

As an aside, the formulation presented here for distributed structural optimization can also be applied to more general problems involving optimization of systems governed by partial differential equations (PDEs). Such an approach brings together techniques for optimization of coupled systems with domain decomposition schemes[27] for numerical solution of PDEs. It is of interest to note that such an approach has been studied in the context of parameter estimation problems by Dennis and Lewis.[28] Further, the approach used here can be employed to construct test problems with varying levels of difficulty for testing MDO formulations by varying the coupling bandwidth (i.e., the number of dof in the interface region) and the number of design variables. In other words, domain decomposition appears to be a natural approach for constructing MDO test problems.

5. DEMONSTRATION EXAMPLES, RESULTS AND DISCUSSION

This section presents the results of experimental studies conducted to generate computational data on the performance of the proposed CMDO architecture. The objective is to gain insights into its convergence characteristics and to make comparison studies with a system optimization approach. The test problems considered here are constructed using the domain decomposition-based structural optimization approach discussed in the previous section.

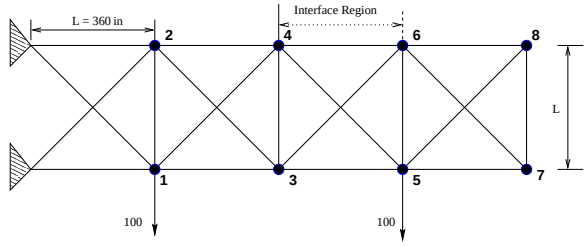

FIG. 6. 20 Bar Planar Truss Structure

For all the experiments conducted, a standard elitist GA with binary tournament selection, uniform crossover and bit mutation was used. The probability of crossover and bit mutation were kept constant at 0.5 and 0.01, respectively. A binary string of 10 bits is used to represent each of the continuous variables for the examples considered. The stress constraints are incorporated into the fitness functions using a penalty function formulation. The best design vector of a species is chosen as its representative individual. The results obtained using the CMDO architecture are compared to the optimal solution obtained using a standard GA applied to the complete coupled/original structural model, which is referred to as the system optimization approach. Convergence studies are presented for the objective functions, constraints, and the coupling compatibility constraint violation corresponding to the fittest design. Note that that the coupling compatibility constraints are not incorporated into the fitness function. They are computed here in the post-processing stage only for the purpose of tracking the multidisciplinary feasibility of the optimal solution. Ten runs were carried out for each case to compute the averaged convergence trends and other statistics. The computations were carried out using a SGI Origin 2000 computer with R10000 processors.

5.1. Problem 1

The first design problem considered is a 20 bar planar truss structure with 4 bays (see Figure 6), parameterized in terms of the sizing variables. The cross-sectional areas of the truss members are bounded between 0.1 and 15.0 in^2. The structural members have Young's modulus $E = 30000$ psi/in^2, mass density $\rho = 0.1$ lb/in^3, and yield stress $\sigma_{max} = 25$ ksi. The design objective is to minimize the weight subject to stress constraints. This leads to a total of 20 design variables and 20 constraints for this problem.

The structure is decomposed into two coupled substructures as shown by the solid line in Figure 6, i.e., the original analysis equations of size 16×16 are decomposed into two sets of coupled equations of size 8×8. Two species which collaboratively coevolve the variables corresponding to each substructure are set up to solve this problem. The first species controls the cross-sectional areas of the ten truss members in the first two bays. The analysis equation $\mathbf{A}1$ for this species (see equation

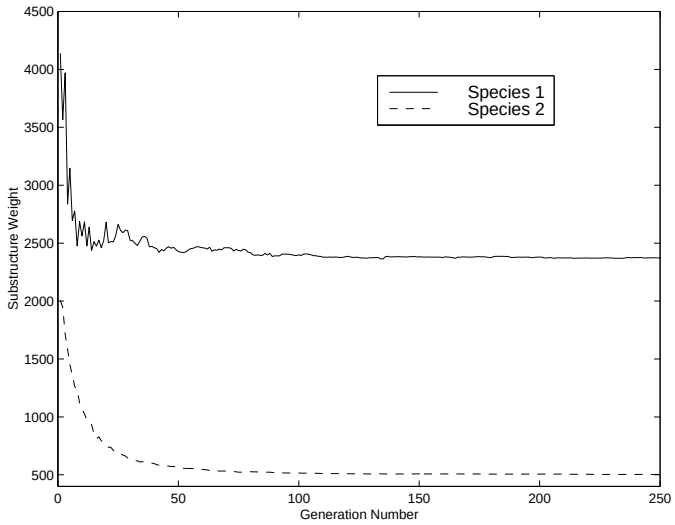

FIG. 7. Averaged Convergence Characteristics of Substructure Weights for Problem 1

(6)) involves the dof corresponding to nodes 1-4. Similarly, the second species controls the cross-sectional areas of the ten truss members in the next two bays, and the analysis equation **A**2 (see equation (7)) involves the dof corresponding to nodes 5-8. The sets of design variables corresponding to each substructure are defined as the chromosomes of the two species. Each species is entrusted with the task of minimizing the substructure weight subject to constraints on the maximum permissible stresses in the substructure elements, i.e., 10 constraints each.

A population size of 50 was used for each species and the termination criteria was kept fixed at 250 ecosystem generations. The initial data for the coupling variables were generated randomly in each optimization run. In the subsequent ecosystem generations, the coupling variables were updated using the values posted by the species representatives on the blackboard (see Figure 5). The results are compared to a system optimization approach which uses a population size of 100, and a termination criteria of 250 generations.

Figure 7 compares the averaged convergence characteristics (of the substructure weights) of the two species which coevolve to the optima of this problem. It can be seen from the figure that both species have converged quite rapidly to the optimal solution. The statistics of the optimal solution obtained using the CMDO architecture are compared with results obtained using the system optimization approach in Table 1. It can be clearly seen that the CMDO architecture gives a better solution on average, as compared to the system optimization approach. Note also that, in spite of the small size of the analysis problem (16 dof), the decomposition-based approach is nearly 1.5 times faster than the system optimization approach.

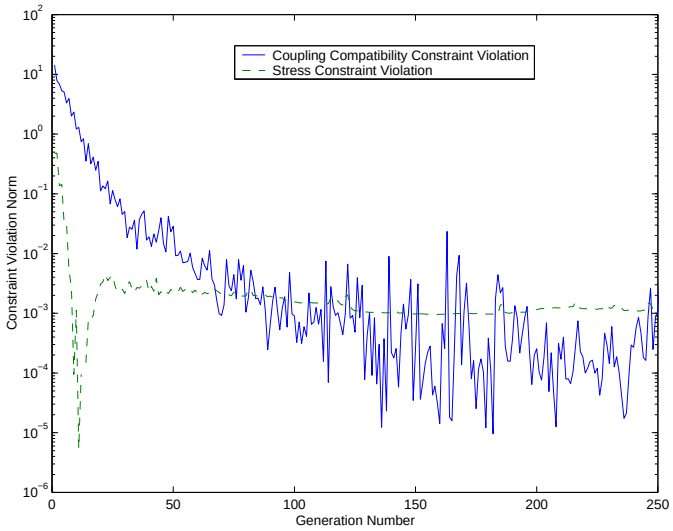

FIG. 8. Averaged Convergence Characteristics of the Coupling Compatibility Constraint Violation and Average Percentage Stress Constraint Violation for Problem 1.

TABLE 1

Comparison of Results for Problem 1

Approach	System Optimization	CMDO Architecture
Average Weight	2970	2876
Standard Deviation	107	64
Minimum Weight	2778	2776
CPU Time(sec)	27	19

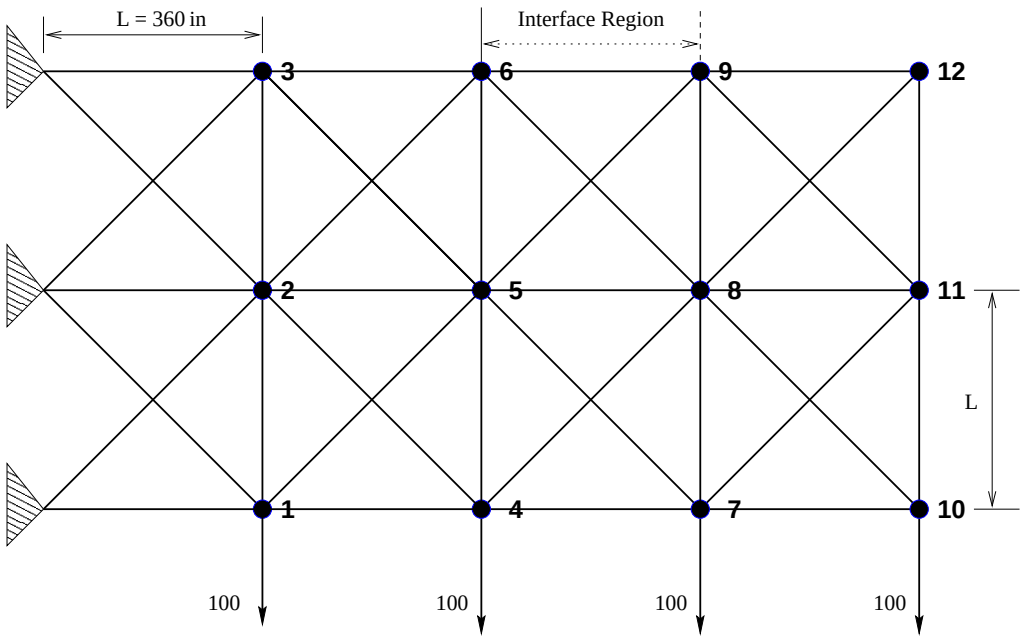

FIG. 9. 36 Bar Planar Truss Structure.

Figure 8 shows the averaged convergence trends of the coupling compatibility constraint violation and the average percentage violation of stress constraints. It can be seen from the figure that, even though the coupling compatibility constraints are not explicitly enforced in the CMDO architecture, they rapidly converge close to zero. These results also indicate that the proposed optimization architecture is relatively insensitive to the initialization values used for the coupling variables.

5.2. Problem 2

The second problem involves weight minimization of a 4 bay 36 bar truss structure (see Figure 9) again subject to stress constraints. The material properties of the structural members are similar to those used for problem 1. The cross sectional areas of the 36 truss members are considered as discrete design variables chosen from the set $0.1i, i = 1, \ldots, 128$ in^2. The coordinates of the joints are considered as continuous design variables varying between ± 90in from the baseline values. This yields a total of 60 design variables and 36 constraints for this problem.

As shown in Figure 9, the structure is decomposed into two coupled systems using the domain decomposition approach discussed earlier, i.e., the original analysis equations of size 24×24 are decomposed into two sets of coupled equations of size 12×12. The first substructure involves the dof corresponding to nodes 1-6, and the second substructure involves the dof corresponding to nodes 7-12. Since the interface region between the two substructures is connected via 6 dof on each side, the coupling bandwidth is higher than in problem 1.

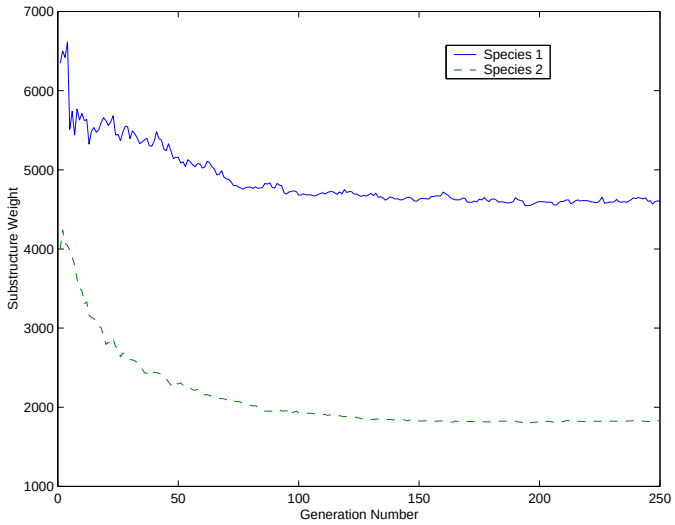

FIG. 10. Averaged Convergence Characteristics of Substructure Weights for Problem 2.

A CMDO architecture comprising of two species, each controlling the 30 design variables corresponding to its substructure is set up to solve this problem. The first species controls the cross-sectional areas of the 18 truss members in bays 1-2 and the coordinates of nodes 1-6. The second species controls the areas of the 18 truss members in bays 3-4 and the coordinates of nodes 7-12. Each species is entrusted with the task of minimizing the substructure weight subject to stress constraints for the substructure elements.

A population size of 100 was used for each species, and the termination criteria was kept fixed at 250 ecosystem generations, The coupling variables were initialized randomly for each optimization run. In the subsequent ecosystem generations, the values posted by the species on the blackboard were used. The performance of the CMDO architecture is compared to a system optimization approach which uses a population size of 200, and a termination criteria of 250 generations.

The convergence trends of the substructure weights averaged over 10 runs are shown in Figure 10. It can be seen from the figure that, species 1 which handles the design variables corresponding to the first substructure shows slightly oscillatory behavior. This is primarily due to the fact that the fitness function of species 1 depends on 19 parameters (displacements and geometry of nodes 7-9, and areas of members connecting nodes 4-6 to 7-9), which are controlled by species 2. In comparison, the fitness of species 2 depends on only 12 parameters (displacements and geometry of nodes 4-6) in species 1.

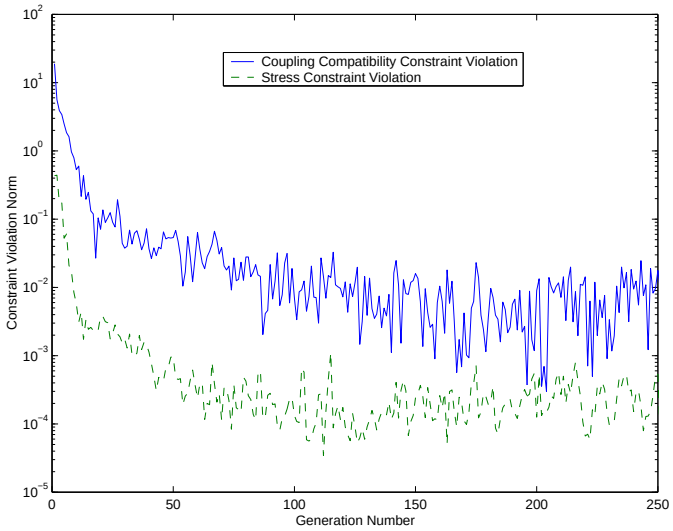

FIG. 11. Averaged Convergence Characteristics of the Coupling Compatibility Constraint Violation and Average Percentage Stress Constraint Violation for Problem 2.

Figure 11 shows the convergence trends of the coupling compatibility constraint violation and the average percentage violation of stress constraints, averaged over 10 runs. It can be seen that the coupling compatibility constraint violation reduces rapidly during the coevolution process. As mentioned earlier, the coupling compatibility constraints are computed here only for the purpose of tracking the multidisciplinary feasibility of the optimal solution.

The statistics of the optimal solution obtained using the CMDO architecture are compared with the system optimization approach in Table 2. It can be observed that the CMDO architecture gives a much better solution as compared to the system optimization approach. This can be attributed to the dynamic nature of the search space when a distributed optimization approach is used. This could be a key factor which enables the CMDO architecture to circumvent the problem of entrapment in a suboptimal solution as compared to the system optimization approach. The results for this problem indicate that the performance of the proposed optimization architecture show no degradation with increase in the coupling bandwidth, even though the convergence trends show slightly more oscillatory behavior. However, it can be noted that the coupling compatibility constraint violation is only reduced to the order of 10^{-2} (on an average) due to the high degree of coupling.

It is also of interest to note that, in spite of the small size of the analysis problem (a total of 24 dof), the substructuring strategy combined with the CMDO architecture is nearly twice as fast as compared to the system optimization approach, while giving results that are 17% better on average.

TABLE 2
Comparison of Results for Problem 2

Approach	System Optimization	CMDO Architecture
Average Weight	7769	6432
Standard Deviation	185	182
Minimum Weight	7501	6218
CPU Time(sec)	160	85

6. CONCLUDING REMARKS

This paper introduces a distributed MDO architecture inspired by the phenomena of coevolutionary genetic adaptation in ecological systems. The advantages offered by the proposed coevolutionary MDO (CMDO) architecture include retainment of disciplinary autonomy, massive parallelism, reduced software integration and interdisciplinary communication overheads, and accommodation of design parameterization with a mix of discrete and continuous variables. The most important advantage offered by the approach is its applicability to systems with arbitrary coupling bandwidth without an attendant increase in the problem size. It is suggested that the architecture may be a useful tool for large-scale nonconvex MDO problems with a mix of discrete and continuous variables.

A formulation is presented for distributed optimization of structural systems via domain decomposition to construct MDO test problems. Numerical studies conducted to examine the performance of the CMDO architecture for the two example problems considered are very encouraging. These results also give some insights into the convergence characteristics of the CMDO architecture. They indicate that the performance of the method does not deteriorate with increase in the coupling bandwidth. The formulation used for the example problems presented also suggests a computationally efficient way to solve large-scale structural optimization problems via domain decomposition, since the dimension of the substructure analysis equations is much lower than those of the original model. The computational cost savings are expected to be particularly significant for optimization of large-scale structural systems. Note that this approach could also be applied to optimization of structures subjected to dynamic loading, and the design of nonlinear structural systems.

It is to be noted that in this research the multidisciplinary design variables were decomposed into completely disjoint sets. In contrast, the issue of developing coevolutionary search frameworks for scenarios where the disciplines are allowed to share design variables remains an open research area. Some ideas for achieving this have

been discussed in reference.[29] However, the ability of such strategies to aid the disciplinary species in arriving at a consensus on the multidisciplinary design variables remains to be seen. Results from the area of computational immunology[30] may be useful in this context. It is also of interest to examine the possibility of incorporating multi-objective decision making capability within the CMDO framework, based on the wide body of research work in this area; see, for example, reference.[31]

The CMDO architecture proposed here implicitly handles the coupling compatibility constraints via randomly restarted generalized Jacobi iteration. Hence, for some problems, difficulties may arise in converging to a solution which satisfies the physics of the coupling. Clearly, for such problems, the present approach can only give a rough indication of the optimal solution. However, such a solution could be used as an initial guess for a distributed or a system-level gradient-based optimization scheme. Alternatively, more robust iterative analysis schemes[25] or evolutionary successive over relaxation techniques[32] could be incorporated in the final stages of the coevolutionary search. Finally, even though promising results were obtained for the examples under consideration, more detailed numerical studies are required to study the robustness of the CMDO architecture for a wider class of design problems.

ACKNOWLEDGMENTS

This research was supported by a grant from the Faculty of Engineering and Applied Sciences at the University of Southampton.

REFERENCES

[1]Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, 1997, pp. 1-23.

[2]Korte, J.J., Salas, A.O., Dunn, H.J., Alexandrov, N.M., Follett, W.W., Orient, G.E. and Hadid, A.H., "Multidisciplinary Approach to Aerospike Nozzle Design," *NASA TM 110326*, February 1997.

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

[4]Wujek, B., Renaud, J. E., Batill. S. M., Brockman, J. B., "Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment," *Concurrent Engineering: Research and Applications*, Vol. 4, No. 4, 1996, pp. 361-378,

[5]Braun, R.D., "Collaborative Optimization: An Architecture For Large-Scale Distributed Design," *Ph.D. Thesis*, Stanford University, 1996.

[6]Braun, R. D., and Kroo, I. M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State of the Art*, SIAM, 1997, pp. 98-116.

[7]Alexandrov, N.M., Lewis, R.M., "Analytical and Computational Aspects of Collaborative Optimization," NASA TM 2000-210104, April 2000.

[8]Balling, R., and Sobieszczanski-Sobieski, J,, "Optimization of Coupled Systems: A Critical Review of Approaches," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 6-17.

[9]Alexandrov, N. M., and Kodiyalam, S., "Initial Results of an MDO Method Evaluation Study," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, Missouri, AIAA Paper 98-4884, 1998.

[10]Hajela, P., "Nongradient Methods in Multidisciplinary Design Optimization- Status and Potential," *Journal of Aircraft*, Vol. 36, No.1, 1999, pp.255-265.

[11]Lee, J., and Hajela, P., "Parallel Genetic Algorithm Implementation for Multidisciplinary Rotor Blade Design," *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 962-969.

[12]Keane, A. J., "Experiences with Optimizers in Structural Design," *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control*, Plymouth, 1994, pp. 14-27.

[13]Nair, P. B., and Keane, A. J., "Passive Vibration Suppression of Flexible Space Structures Via Optimal Geometric Redesign," *AIAA Journal*, Vol. 39, No. 7, 2001, pp. 1338-1346.

[14]El-Beltagy, M. A., Nair, P. B., and Keane, A. J., "Metamodeling Techniques for Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations," *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, FL, Morgan Kaufmann Publishers, 1999, pp. 196-203.

[15]Venter, G., and Haftka, R. T., "A Two Species Genetic Algorithm for Designing Composite Laminates Subjected to Uncertainty," AIAA Paper 96-1535, April 1996.

[16]Potter, M. A., "The Design and Analysis of a Computational Model of Cooperative Coevolution," *Ph.D. Thesis*, George Mason University, Fairfax, VA, 1997.

[17]Potter, M. A., and De Jong, K. A., "A Cooperative Coevolutionary Approach to Function Optimization," *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, Springer-Verlag, 1994, pp. 249-257.

[18]Hillis, W. D., "Coevolving Parasites Improve Simulated Optimization as an Optimization Procedures," *Physica D*, Vol. 42, 1990, pp. 228-234.

[19]Husbands, P., "Distributed Coevolutionary Genetic Algorithms for Multi-criteria and Multi-constraint Optimization," *Evolutionary Computing, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 865, 1994, pp. 150-165.

[20]Potter, M. A., and De Jong, K. A., "Evolving Neural Networks with Collaborative Species," *Proceedings of the Summer Computer Simulation Conference*, 1995, pp. 340-345.

[21]Arian, E., "Convergence Estimates for Multidisciplinary Analysis and Optimization," *NASA/CR-201752*, October 1997.

[22]Bull, L., "Evolutionary Computing in Multi-Agent Environments: Partners," *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann, 1997, pp. 370-377.

[23]Hajela, P., and Yoo, J., "Constraint Handling in Genetic Search Using Expression Strategies," *AIAA Journal*, Vol. 34, No. 11, 1996, pp. 2414-2420.

[24]Koziel, S and Michalewicz, Z., "Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization," *Evolutionary Computation*, Vol. 7, No.1, 1999, pp. 19-44.

[25]Kelley, C. T., *Iterative Methods for Linear and Nonlinear Equations*, SIAM Frontiers in Applied Mathematics, No. 16, SIAM, Philadelphia, 1995.

[26]Keane, A. J., and Nair, P. B., "Problem Solving Environments in Aerospace Design," *Advances in Engineering Software*, Vol. 32, 2001, pp. 477-487.

[27]Smith, B. F., Bjϕrstad, P. E., and Gropp, W. D., *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[28]Dennis, J. E., and Lewis, R. M., "A Comparison of Nonlinear Programming Approaches to an Elliptic Inverse Problem and a New Domain Decomposition Approach," *CRPC-TR94468*, Center for Research on Parallel Computation, Rice University, 1994.

[29]Nair, P. B., and Keane, A. J., "Coevolutionary Genetic Adaptation: A New Paradigm for Distributed Multidisciplinary Design Optimization," AIAA Paper 99-1428, April 1999.

[30]Smith, R. E., Forrest, S., and Perelson, A. S., "Searching for Diverse, Cooperative Populations with Genetic Algorithms," *Evolutionary Computation*, Vol. 1, No. 3, 1993, pp. 191-211.

[31]Coello Coello, C. A., "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, Vol. 1, No. 3, 1999, pp. 269-308.

[32]He, J., Xu, J., and Yao, X., "Solving Equations by Hybrid Evolutionary Computation Techniques," *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 3, 2000, pp. 295-304.