Problem Formulations and Other Optimization Issues in Multidisciplinary Optimization

J.E. Dennis, Jr.
Robert Michael Lewis

CRPC-TR94469
April, 1994

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Problem Formulations and Other Optimization Issues in Muldisciplinary Optimization

J. E. Dennis, Jr and Robert Michael Lewis. *

Abstract

This paper is about multidisciplinary (design) optimization, or MDO, the coupling of two or more analysis disciplines with numerical optimization. The "individual discipline feasible" (IDF) approaches introduced here make use of existing specialized analysis codes, and they introduce significant opportunities for coarse-grained computational parallelism particularly well-suited to heterogeneous computing environments.

Keywords: Constrained optimization, multidisciplinary design optimization, optimal design, computational engineering, parallel computation.

1 Introduction

Because of the complexity of most MDO problems, we need to understand the formulations of MDO problems and their *interdependence* with optimization algorithms. To this end we sketch here a set of tools consisting of problem formulations and optimization algorithms.

In part, the genesis of the ideas given here was an attempt to find ways for nonlinear programming methods to exploit both parallel computation and decomposition methods for simulation. We decided on problem reformulation as an approach to pursue because the design of parallel algorithms for general nonlinear programming (NLP) has not been successful, and we hoped to mesh the simulation with the optimization in such a way as to lift into the optimization routine the parallelism in the simulation. We were able to do this, and we have extended these ideas to MDO in joint work with Evin Cramer, Paul Frank, and Greg Shubin of Boeing Computer Services. A detailed version can be found in [9]. In this paper we will present a sketch of that work together with a brief discussion of some optimization algorithms likely to be useful for MDO. We recommend Greg Shubin's computational study [21] of these formulations applied to a model problem in aircraft design.

The IDF formulation given in section 2 has a simplicity of structure that allows easy incorporation of existing disciplinary analysis codes without the need for a multidisciplinary analysis procedure. However, this is not the only motivation for this formulation. From the point of view of nonlinear programming, the IDF approach may prove to be a more efficient approach for solving MDO problems than conventional approaches. This has certainly been our experience in parameter estimation for ordinary differential equations (ODE).

Having formulated MDO as a large-scale constrained optimization problem, we can apply algorithms developed in the past decade of research in large-scale nonlinear programming. In Section 3, we discuss some optimization algorithms we have developed at Rice because we believe that these trust-region methods for constrained optimization are especially likely to be useful. For our purposes here, the reader can think of trust-region methods as NLP methods that adaptively set their own move limits as the iteration proceeds. Thus, there must be a choice only of the first move limits to try for the initial step, and from that point on, the algorithm sets the move limits.

There is certainly other applicable work on NLP algorithms. The Stanford Optimization Laboratory has produced the program NPSOL [20] which has

*Department of Computational and Applied Mathematics and Center for Research in Parallel Computation, Rice University, P.O. Box 1892, Houston, TX, 77251-1892. This work was supported by the State of Texas under contract #1059, the Air Force Office of Scientific Research under grants F49629-92-J-0203 and F49629-9310212, the Department of Energy under grant DE–FG005–86ER25017, and the National Science Foundation under cooperative agreement CCR-9120008. Copyright c 1994 by the American Institute of Aeronautics and Astronautics, Inc. All right reserved.

been used successfully in aerospace applications at Boeing. It uses a more conventional line search approach.

In order to be computationally effective, optimization algorithms must exploit the pronounced block structure of MDO problems. Our large-scale optimization algorithms are designed to do this. These algorithms were developed in our work on parameter estimation for systems governed by ODE and PDE (partial differential equations) to solve problems that are structurally very similar to MDO problems. In that work, the "disciplines" corresponded to the subdomains produced by applying a domain decomposition method to the governing differential equations. We are also very hopeful of the success of a nonlinear multilevel approach being developed by Natalia Alexandrov of ICASE [2, 3]. The decomposition of MDO problems along the lines of the constituent disciplines also leads one to a natural coarse-grained computational parallelism, a point worth noting but one we will not discuss here in any detail.

A major obstacle to making MDO truly practical is the need for sensitivities of the disciplinary analysis codes with respect to the parameters controlled by the optimizer. It was our interest in sensitivities for use in nonlinear programming that we helped to start the development of ADIFOR, an automatic differentiation tool for Fortran programs, and we continue to influence the work of the ADIFOR group. Specifically, we are discussing with them tools to aid the developer of analysis codes to produce sensitivities as well.

2 Formulations of the MDO problem

The importance of problem formulation is a major topic in the engineering literature on MDO. Sobieszczanski–Sobieski argued for its significance at least as far back as 1982 [22]. The status of this research area is well summarized in [1].

The observation that lies at the root of the formulations we propose is that the disciplinary analyses of MDO—the simulations of the response of the physical system in the MDO problem to those parameters we can control—constitute *equality constraints* in the optimization problem. Thus, we view the MDO problem as a nonlinear programming problem in which the standard notion of MDO constraints, such as performance goals or design parameter bounds, are dominated by constraints from the disciplinary analyses which are strongly partitioned into a relatively small number of large blocks. These blocks represent the contributions of the various disciplines to the model of the entire system. Our view is that this block structure and the nature of the constraints representing the interdisciplinary coupling must also figure in the design of optimization algorithms for the solution of the MDO problem.

We have identified three broad classes of formulations of MDO problems:

- The multidisciplinary feasible (MDF) formulation,
- The individual disciplinary feasible (IDF) formulations, and
- The "all-at-once" (AAO) formulation.

Our major contribution is the large and flexible class of *individual discipline feasible* (IDF) formulations of MDO. One may view the IDF formulations as reformulations of the system-level optimization problem so that a nonhierarchical problem can be written as a hierarchical problem. We accomplish this by introducing new optimization variables and new optimization constraints. The key is not to introduce so many new variables that the problem grows too large.

We will give an abbreviated presentation of our taxonomy of the formulations of MDO, using notation that differs slightly from that used in the AIAA paper [8] or the more developed paper [9]. In the latter paper, we give a complete framework for describing MDO problems. For purposes of the present exposition, the example we will discuss in this section will be an MDO problem with two disciplines. For simplicity, we will ignore side constraints not associated with any particular discipline, such as constraints on the design variables x. However, extending the formulations given here to larger numbers of disciplines or admitting design constraints in the original MDO problem presents no difficulty. It is useful to point out that whether or not the user chooses to have the optimization algorithm maintain feasibility with respect to design constraints is not an issue in any of these formulations. The formulations are distinguished only by their treatment of the analysis constraints with respect to the optimization variables step computation.

2.1 The multidisciplinary feasible (MDF) approach

The conventional formulation of our example MDO problem is:

$$\text{minimize } f(x,\, y_1(x),\, y_2(x)),$$

where $y_1(x)$ and $y_2(x)$ are defined via the implicit relations

$$h(x,\, y(x)) = \left\{ \begin{array}{lcl} h_1(x,\, y_1(x),\, y_{12}(y_2(x))) & = & 0 \\ h_2(x,\, y_{21}(y_1(x)),\, y_2(x)) & = & 0. \end{array} \right. \tag{1}$$

The relations h_i represent individual discipline analyses, say, structural response and aerodynamics in static aeroelasticity. The design variables are x and might be spline coefficients describing the shape of a wing. The dependent variable y_i represents the analysis outputs computed by the analysis h_i. For instance, y_1 might represent structural deflections or stresses and y_2 might represent pressures computed by the aerodynamics analysis.

The quantities y_{ij} are very important to our approach; they represent the interdisciplinary transfer of information. For instance, if h_1 represents structural response and $y_2(x)$ the pressure field, $y_{12}(y_2(x))$ might represent the pressures on the wing, a subset of the information contained in the entire pressure field. The amount of information passed between the disciplines, i.e., the size of the y_{ij}, we call the *interdisciplinary bandwidth*.

In the conventional approach to this problem, we would treat the design variables x as the only truly free variables. At each optimization iteration we would choose x and then solve for the analysis outputs $y(x)$ via the relation $h(x, y(x)) = 0$. We call the solution of this coupled system of equations, for a fixed x, a *multidisciplinary analysis*, or MDA. An MDA corresponds to computing a consistent multidisciplinary simulation.

We call the conventional approach the *multidisciplinary feasible* (MDF) approach. The optimizer controls the design variables and any design constraints. At every optimization iteration we determine the state variables of the coupled aerodynamical-structural multidisciplinary analysis by doing a full MDA each time a value is needed for y. This approach treats the state variables y as dependent on x via (1) and eliminates y as a variable in the optimization problem. However, we must compute sensitivities of $y(x)$ with respect to x, and these sensitivities must be computed at a full MDA solution. The MDF approach has some attractive features:

- MDF leads to the smallest optimization problems of the three formulations.
- Existing disciplinary solver codes can be used as black boxes. We do not need explicit access to the internal operation of the solver codes or to residuals of the equations they solve.
- Each MDF iterate is a feasible design insofar as it satisfies the coupled system of disciplinary solvers, i.e., the MDA.
- It is a traditional approach well-understood by users.
- MDF is the MDO version of the generalized reduced gradient (GRG) algorithm. For general nonlinear programming, at least, GRG can be effective if carefully implemented.

However, MDF has some significant disadvantages which we believe make it impractical when the MDA is expensive:

- Each function value in MDF requires a complete multidisciplinary analysis.
- Computing sensitivities of the state variables with respect to the design variables require full MDA solutions.
- If we compute sensitivities by finite-differences, we may encounter serious problems with the computational cost and accuracy because of the solution operators inside the multidisciplinary analysis. These typically involve the solution of PDE and can be very difficult to accurately differentiate using finite-differences.
- The time it takes to solve the optimization problem via the MDF formulation is closely linked to the efficiency with which we can solve the MDA.

Thus, in general this approach will be very costly since at every value of x considered for any reason by the optimization iteration the MDF formulation requires that we solve for the multidisciplinary state vector y that simultaneously satisfies *all* of the individual discipline analyses. We are required to devote computational effort running the simulation codes to perform a multidisciplinary analysis when we would rather be making progress on the optimal design problem. A variation on this straightforward

MDF approach that might reduce the computational expense would be to mimic for MDO what has been done in single discipline structural optimization by using so-called approximation concepts [4] for each analysis code. We plan to investigate such short-cuts for the alternative problem formulations presented in the following sections.

2.2 The All-at-once (AAO) approach

If we treat feasibility of the multidisciplinary analysis entirely as a constraint, then we obtain what we call the *all-at-once* (AAO) formulation of the MDO problem:

$$\begin{array}{ll} \text{minimize} & f(x,\, y_1,\, y_2) \\ \text{subject to} & h_1(x,\, y_1,\, y_{12}(y_2)) = 0 \\ & h_2(x,\, y_{21}(y_1),\, y_2) = 0. \end{array}$$

Now y_1 and y_2 are independent variables, and the relations between them and x are viewed as equality constraints. We no longer need to perform an MDA to obtain y as a function of x. Sensitivities are also much less expensive than in the MDF approach; we do not need to compute derivatives of $y(x)$ with respect to x at a full MDA solution $y(x)$. Instead, one computes the partials of the residual h with respect to x and y.

The sensitivities of f and h that must be computed are the same in all three of the approaches. The big difference between these sensitivity computations for f and h in the three approaches is the value of y at which these sensitivities are to be evaluated. In MDF, $y(x)$ is the result of a full MDA. In IDF, it is the less expensive $y_1(x, y_{12})$ and $y_2(x, y_{21})$ with y_{ij} treated as independent variables. In AAO, y is an independent variable.

The AAO approach has been investigated in a number of engineering fields, for instance, aerodynamics [15], structural optimization [19], where it is called SAD or SAND, for simultaneous analysis and design, and chemical engineering [24], where it is called the open equations or nonlinear programming approach. We have applied this approach to parameter estimation for flow in porous media and for chemical kinetics problems.

Why might it be a good idea to treat disciplinary residuals as explicit equality constraints? After all, in moving from the MDF formulation to the AAO formulation we have greatly increased the number of optimization variables and introduced a large number of equality constraints for the NLP algorithm to manage. However, the reformulation of the MDO problem as a larger constrained problem enables us to obtain a solution more efficiently, because we do not spend time computing a full multidisciplinary analysis at each optimization iteration to get $(x,\, y(x))$ such that $h(x, y(x)) = 0$ and additional multidisciplinary analyses if we are computing sensitivities via finite-differences.

In fact, optimization algorithms that allow for iterates infeasible with respect to nonlinear constraints generally are more efficient than algorithms that enforce feasibility at every iteration. This is usually explained by saying that the additional degrees of freedom may enable the optimization algorithm to move much more quickly to a solution by cutting "cross-country" toward optimality *and* feasibility rather than following a feasible but winding path.

In the AAO formulation, if we allow infeasible optimization iterates, then the additional freedom we gain by expanding the parameter space from x to $(x,\, y_1,\, y_2)$ should enable us to move more quickly towards an optimal solution, but at the expense of solving a larger nonlinear program.

In the AAO approach, treating the state relations inside the analyses as constraints and allowing infeasibility with respect to these constraints amounts to a relaxation of the amount of work required for a multidisciplinary analysis. The amount of work required can be determined *automatically* by the optimization algorithm during the progress of the optimization—a detailed multidisciplinary analysis is forced only as we approach optimality.

For our model problem, this would afford a completely rigorous way to use approximate CFD and structural analyses at the beginning of the optimization; the optimization would determine how accurate a multidisciplinary analysis we would need. The optimization algorithm would then guide the refinement of the accuracy of the analyses as the optimization progresses towards an optimal and feasible solution. For the supporting theory, see [12]. The advantages of the AAO formulation follow:

- AAO should be more efficient than MDF and IDF, insofar as AAO will generally require fewer optimization iterations than MDF and IDF to find a solution.
- AAO does not require even single discipline solutions until optimality.
- The required sensitivities are relatively inexpen-

sive, since they need not be computed at even single discipline solutions, as in IDF, much less at full MDA solutions, as in MDF.

However, the AAO formulation is not without disadvantages:

- The AAO leads to a *much* larger optimization problem than MDF or IDF. For discretized time-varying problems, one would need to store the state values for each point in each spatial grid for each point in time.
- AAO will not easily incorporate analysis algorithms that adaptively change the number of state variables, such as adaptive grid methods, since the state variables appear as optimization variables.
- The optimization algorithm we use in an AAO formulation must have explicit access to residuals treated by single discipline solvers.
- The optimization algorithm we use in an AAO formulation must have built into it any special methods used to achieve single discipline solutions, since we have given the optimizer the task of achieving individual disciplinary and multidisciplinary feasibility—which is essentially equivalent to computing an MDA as we converge to optimality.
- Experience in aeronautical applications shows that this approach can experience difficulty when shocks are present. More generally, the optimization algorithms applied to the AAO formulation can experience difficulty converging to feasibility if started far from feasibility [15].

2.3 The Individual Feasible (IDF) approach

The AAO approach shifts the difficulty of multidisciplinary feasibility from the level of the analysis codes up to the level of the optimization algorithm. This may be inconvenient if it requires extensive rearrangement of the analysis software. Moreover, if one of the analysis codes, aerodynamics, for instance, involves highly specialized solution techniques, then the optimization software may have difficulties achieving feasibility unless these solution techniques are somehow integrated into the optimization. The AAO approach may also lead to a prohibitively large optimization problem.

As a compromise between the MDF and AAO approaches we have devised a family of completely new MDO formulations. We call these the *individual discipline feasible* (IDF) formulations. Conceptually and practically, the IDF formulations span the gulf between the MDF and AAO approaches. The IDF approaches lead to equality constrained optimization problems, so we can apply optimization methods that allow infeasible iterates to solve the resulting optimization problems more efficiently than the MDF formulation. At the same time we avoid the size of the optimization problem that results from the AAO formulation. Moreover, the IDF approaches allow us to use existing analysis software, as in the MDF approach.

In the IDF formulations the analysis outputs corresponding to a given discipline are thought of as dependent within that discipline's analysis and as independent in the other analyses. For our example problem, at each optimization iteration we have a "correct" aerodynamic analysis and a "correct" structural analysis, except that these analyses use the *optimizer's* current estimate of each discipline's inputs, rather than the one corresponding to the output of the other discipline. A representative IDF formulation is then

$$\begin{array}{ll} \text{minimize} & f(x,\, y_1(x, x_{12}),\, y_2(x, x_{21})) \\ \text{subject to} & x_{12} - y_{12}(y_2(x, x_{21})) = 0 \\ & x_{21} - y_{21}(y_1(x, x_{12})) = 0, \end{array}$$

where $y_1(x, x_{12})$ and $y_2(x, x_{21})$ are defined by solving the individual discipline analyses

$$\begin{array}{rcl} h_1(x,\, y_1(x, x_{12}),\, x_{12}) & = & 0 \\ h_2(x,\, x_{21},\, y_2(x, x_{21})) & = & 0. \end{array} \tag{2}$$

We have expanded the space of optimization variables from x to $(x,\, x_{12},\, x_{21})$, and expressed the coupling of the individual disciplines in terms of explicit equality constraints.

The IDF formulation has the attractive feature of not requiring a multidisciplinary analysis solver, unlike the MDF formulation. Instead, we only require the *individual discipline feasibility* expressed in (2). The IDF formulation yokes together the individual discipline analyses at the level of the optimization and does not demand that we assemble a separate MDA solver, itself a significant undertaking. On the other hand, if a multidisciplinary analysis solver is available, we can use it in the subtasks of the optimization algorithm.

Equally important is the observation that the IDF approach leads to optimization problems with more variables (but also more freedom for an infeasible iterates optimization algorithm to exploit) than the MDF formulation, but, in general, far fewer optimization variables than the AAO approach. For instance, in our example, x_{12} plays the role of the pressures or loads only on the wing, which can be described with far fewer parameters than the entire pressure field y_2, which we would need to keep around in the AAO approach. We say that we have decomposed the problem at a low bandwidth interdisciplinary interface.

Additional IDF formulations are described in [9], including the sequenced IDF formulations, which were anticipated in the literature by equation (12) in [18]. The more detailed presentation given in [9] discusses further how we might use the IDF approach to obtain constrained MDO formulations that are much smaller than in the AAO approach.

The common feature of all the IDF approaches is that we maintain feasibility separately with respect to the individual discipline analyses, and we use them to provide an implicit relation to remove some state variables from the optimization problem. Note that not every optimization procedure that maintains feasibility of the individual disciplines is an IDF procedure. The key issue is whether the state variables are eliminated from the optimization iteration, as in the IDF and MDF formulations, or whether they are set between optimization iterations by a restoration step, as might be the case in a restoration algorithm applied to an AAO formulation.

The IDF approach allows us to incorporate, at the level of the optimization problem, certain types of parallelism inside a single discipline analysis. If the parallelism inside a discipline is implemented via some sort of global consistency constraint, as is the case for many domain decomposition methods for partial differential equations, then we can extract this consistency constraint and include it as a coupling constraint in the formulation of the optimization problem. This approach treats the domain decomposition subproblems as individual analyses coupled by an equality constraint. We have had success with IDF formulations of ODE inverse problems, and we are currently investigating this approach in our work on parameter estimation for flow in porous media.

The following are among the advantages of the IDF formulations:

- IDF allows the use of existing disciplinary solvers, which can be run concurrently on appropriate machines.
- The optimization problems resulting from the IDF formulation are nearer in size to those of the MDF than the AAO.
- Sensitivities are less expensive than those of the MDF formulation.

There are also some relative disadvantages.

- We must compute sensitivities of the state variables produced by the single discipline solvers with respect to the inputs to each discipline. This makes the sensitivities more expensive than those of the AAO formulation.
- Any special methods required to handle cross disciplinary consistency must be built into the optimizer.

Shubin [21] contains a useful practical discussion of the issues involved in choosing a formulation for a particular problem.

3 Optimization

A major thrust of our approach is our work on optimization tools and algorithms for MDO. The optimization research that has been conducted at Rice over the course of the past decade can make significant contributions to MDO:

- Exploratory optimization tools for investigating the design space while producing a nominal best design.
- An approach to objective synthesis for resolving competing objectives.
- Parallel direct search methods for derivative-free optimization.
- Parallel large-scale nonlinear programming algorithms that exploit the block structure of the MDO formulations presented in Section 2.
- Techniques for addressing the problem of obtaining sensitivities.

Space permits us to give little detail concerning these topics. We recommend [14] for an informative study on the use of optimization methods in MDO.

Exploratory optimization and objective synthesis are interactive procedures involving the designer.

These procedures should prove very useful in the preliminary stages of the MDO solution process. First we need to decide what we want to design for—the design objective we wish to optimize—and where in the design space we should not look—the design constraints. This first stage could be viewed as an exploratory stage in which single discipline analysis codes, possibly of less physical fidelity, are used in conjunction with an optimization method to explore the design space and at the same time to resolve multiple objectives into a single objective function.

Having produced a provisional single objective to optimize and having identified the region of the design space in which we wish to look for a solution, we can then pass to the use of analysis codes of high physical fidelity (and greater computational cost) possibly applied to more sophisticated designs involving more design variables, coupled with nonlinear programming algorithms appropriate to the nature of the optimization problem. In this second stage, we optimize the design objective in the region of the design space selected during the exploratory stage. The designer can then return to the exploratory optimization phase, if desired, in order to investigate the design space around the design produced.

The point of developing such a suite of optimization tools is to provide designers with the power and sophistication of state-of-the-art nonlinear programming algorithms, but at the same to keep the designer involved in the design optimization process. This is motivated, frankly, by our suspicion of the notion of "push-button" design, especially for problems as complex as MDO.

3.1 NLP algorithms

While there are many computational optimization approaches that present themselves as candidates to solve MDO problems, the most promising involve using calculus-based quasi-Newton methods for numerical optimization [14].

We will discuss several optimization algorithms that we believe will prove effective in solving the different formulations of the MDO problem. The first of these is parallel direct search, which is a parallel optimization algorithm that has the attractive features of not requiring derivatives and being robust and simple to apply. It is most suitable for the MDF formulation of MDO problems involving relatively few (≤ 50) optimization variables.

For the IDF and AAO equality constrained formulations of MDO in Section 2, we need more sophisticated, derivative-based optimization methods. Moreover, if we are really to take advantage of the alternative AAO and IDF formulations of the MDO problem, these optimization algorithms should allow optimization iterates that may be infeasible with respect to the nonlinear constraints. Some such algorithms are of a class known as trust region methods, which are based on local quadratic programming models that incorporate both the objective and the constraints.

Trust region algorithms are robust, efficient, and less dependent than line-search methods on accurate information about derivatives and merit functions. The reader familiar with the notion of move limits in structural optimization methods can think of trust region methods as algorithms that adaptively set move limits at each iteration by comparing actual versus predicted reduction in a merit function, such as the augmented Lagrangian.

We will discuss two related classes of trust region algorithms for the equality constrained MDO formulations. The first of these is a class of large-scale constrained optimization algorithms that are descendents of the Celis-Dennis-Tapia (CDT) algorithm [7]. The other class of trust region methods are generalizations to optimization of the methods of Brown and Brent for nonlinear equations.

Both the parallel direct search and CDT algorithms have already proven their effectiveness in our work on parameter estimation for ODE and for geophysical systems governed by partial differential equations. The multilevel trust region generalizations of the methods of Brown and Brent to optimization are completely new, having been introduced in the thesis [2]. They are motivated by the potential expense of computing derivatives in MDO, and by the possible convenience of being able to process each discipline, or each block of cross disciplinary constraints, independently in the optimization. For this reason we have placed the discussion of these methods in Section 3.2, where we discuss the issue of sensitivities in MDO.

3.1.1 Parallel direct search methods

We can apply to the MDF formulation the parallel direct search methods developed by Dennis and Torczon [11]. Direct search methods are optimization methods that neither require nor estimate derivatives, so they avoid one of the major difficulties in the MDF approach to MDO. Instead of trying to compute gradients, direct search methods move towards optimality by working directly with values of the objective function.

While they are not as efficient as methods which use derivatives, the direct search methods have attractive qualities of their own, such as ease of use. They are also robust in the presence of noise and well-defined even when the objective function is neither continuous nor differentiable. Moreover, when the objective function is differentiable, the direct search methods are supported by a convincing convergence theory [26, 28], making them an attractive alternative to more ad hoc methods such as genetic algorithms or neural networks.

Dennis and Torczon [11] have developed direct search methods specifically for parallel computers, and the parallelism is of an uncommon type insofar as in principle, there is no upper limit to the number of processors that can be used effectively in the algorithm. A code developed by Torczon [27] has solved a variety of problem, including a velocity estimation problem in oil exploration, a problem in tumor modeling, and a parameter estimation problem in modeling hearing loss. Recently it was inserted as a crucial component of a Boeing parts nesting code developed by Eldersveld and Grandine [13, 17] to plan for minimizing waste while cutting aircraft parts from sheet metal.

3.1.2 The CDT algorithm for large-scale equality constrained optimization

We have implemented an algorithm that can handle very large equality constrained optimization problems and successfully used it in the parallel solution of parameter estimation for flow in porous media. These flow in porous media problems are quite large, involving tens of thousands of design variables and equality constraints. The code, developed by Michael Lewis, solved the problem in MDF, IDF, and AAO formulations.

The problem with allowing infeasible iterates in constrained optimization is that one is pulled in two different directions. On the one hand, we want to achieve feasibility as we arrive at constrained optimality. At the same time, attempting to optimize the objective without proper consideration to the constraints will tend to draw us away from feasibility.

Our algorithm is based on the following idea of Celis, Dennis, and Tapia for solving the equality constrained problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0. \end{array}$$

The CDT idea is to predict at each iteration a certain rigorously justified amount of improvement in feasibility while optimizing a local quadratic model of the Lagrangian function. The CDT algorithm accomplishes this by the approximate solution of the following subproblem. If x_c is our current estimate of the optimal design, we obtain our optimization step s by computing s that approximately solves

$$\begin{array}{ll} \text{minimize} & q_c(s) \\ \text{subject to} & \|s\| \leq \delta_c \\ & \|\nabla h_c^T s + h_c\| \leq \theta_c. \end{array} \tag{3}$$

In this problem, $q_c(s)$ is a quadratic model of the Lagrangian and ∇h_c^T is the Jacobian of the constraints at x_c. The quantity δ_c is the trust radius, which serves to limit the size of the step we take and which is updated at the end of each optimization iteration based on the how well the model predicted the actual behavior of the function and constraints. The quantity θ_c is chosen to ensure convergence to feasibility; in theory we need only require a fraction of Cauchy decrease on the sum of squares of the linearized constraints [10].

Having approximately solved this subproblem, we apply an acceptance test to the new iterate. This acceptance test currently uses the augmented Lagrangian as its merit function. The augmented Lagrangian combines the objective and the constraints to measure progress towards the competing goals of optimality and feasibility. The choice of the penalty parameter in the augmented Lagrangian follows that in [12]. In contrast to augmented Lagrangian methods, the penalty weight in the augmented Lagrangian does not directly enter into the computation of the optimization step. Instead, it only figures in the step acceptance test.

Because we make an effort to improve feasibility of the constraints at each iteration, the algorithm is robust—barring some rare pathologies, it will converge to a solution even when starting far from optimality and feasibility. On the other hand, the algorithm is designed so that we do not move too quickly to feasibility. If we arrive at feasibility too quickly then we can become mired while moving around the feasible set. This can deprive us of the advantages of infeasible iterates and the additional degrees of freedom we enjoy in the equality constrained formulations of the MDO problem.

These algorithms can be implemented to take advantage of either sparse direct methods or preconditioned conjugate direction methods in the solution of the linear systems and local minimization of quadratic forms that arise in the optimization iteration. This decision is of importance for the parallel solution of large-scale optimization problems.

Solving the subtasks of the optimization iteration by use of iterative methods such as conjugate gradients is particularly efficient. This choice allows us to use the ideas of Steihaug and Toint [23, 25] for truncated conjugate direction iterations to reduce the amount of work in an optimization iteration when we are far from a solution.

Because we are using conjugate directions methods, we only require the action of Jacobian and Hessian matrices on vectors, and not the actual computation and assembly of these matrices. This greatly reduces the expense of the sensitivity information required. Moreover, iterative methods for linear algebra reduce the expense and increase the parallelism of the computation. Finally, the structure of the IDF formulations should assist in using various components from the analysis codes to construct effective block preconditioners for use in iterative solvers.

The major issue in large scale optimization is how best to deal with inequality constraints. Any active set method [16] is compatible with our CDT algorithm, but no one has much experience with the use of active set methods on problems as large as MDO. We intend to investigate a simple active set method recently proposed by John Dennis, as well as interior point methods such as the classical barrier methods and Polyak's modified barrier methods.

3.2 The problem of sensitivities

Quasi-Newton optimization methods require that we compute partial derivatives to form gradients and Jacobians (or Jacobians and Hessians times vectors) to use in the algorithm. The relations we must differentiate may be very complex; consequently, this information about derivatives may be very difficult to extract from the analyses since these derivatives will involve the sensitivity of the analysis outputs with respect to the design variables and other analysis inputs.

There are several approaches to the problem of computing derivatives in MDO. One new and radically different approach involves multilevel optimization algorithms. There is also the possibility of using the current version of the ADIFOR automatic differentiation tool. Another approach we plan to pursue is to help extend ADIFOR to assist in semi-automatic implicit differentiation.

3.3 Multilevel algorithms

The difficulty of obtaining derivatives and the block structured analysis constraints in MDO motivate a class of optimization methods, known as multilevel algorithms, based on Brown-Brent methods for nonlinear equations. The multilevel algorithms can be applied to all classes of MDO formulations.

Brown [6] suggested a nonlinear generalization of Gaussian elimination for solving square systems of nonlinear equations. Brown's method is locally q-quadratically convergent, the same rapid rate of convergence as Newton's method, and its variant that uses finite-difference derivatives can be implemented to require the calculation of fewer—roughly half as many—sensitivities than we would need to calculate for Newton's method. Brent [5] and other authors have suggested different perspectives and variants, but no one had previously globalized these methods.

The multilevel methods being developed at ICASE and Rice, see [2, 3], can be viewed as globalizations of the Brown-Brent methods for nonlinear equations and their extension to constrained optimization. The basic idea amounts to successive minimization of progressively smaller-dimensional (reduced) models of the arbitrarily partitioned constraint blocks and finally the reduced model of the objective function. The method avoids the cost of simultaneous sensitivity analyses. In addition, this approach enables us to determine optimization steps while computing fewer sensitivities than in Newton's method.

Another important application for the multilevel algorithms is MDA. As mentioned previously and discussed in detail in [9], MDA, or the procedure of bringing all the disciplines into equilibrium, can be extremely expensive because of the need to execute the analysis codes and to compute the associated sensitivities repeatedly.

We believe that for a typical problem, finite-difference sensitivities are most viable for AAO and

IDF formulations, and less viable for the MDF formulation because of the difficulties associated with computing accurate finite-differences of the MDA. In any event, for any one of these formulations, finite-difference sensitivities are, in general, the least desirable in terms of both cost and accuracy.

4 Widening the interest in MDO

The modeling system we have developed with our collaborators at Boeing seems to be powerful enough to represent MDO problems in a diverse set of fields. We published a detailed version of that system in the optimization literature in order to reach out to the optimization community. We plan to extend this work by adding examples from chemical process control, environmental process control, nondestructive testing, and aquifer characterization and showing how to use our system to model those problems. We hope in this way to help make MDO a powerful conceptual model for computational engineering.

References

[1] AIAA Technical Committee on Multidisciplinary Design Optimization (MDO). White paper on current state of the art. American Institute of Aeronautics and Astronautics, 1991.

[2] Natalia Alexandrov. Multilevel algorithms for nonlinear equations and equality constrained optimization. Technical Report TR93–20, Department of Computational and Applied Mathematics, May 1993.

[3] Natalia Alexandrov and J. E. Dennis, Jr. Multilevel trust-region algorithms for nonlinear equations and equality constrained optimization. Technical report, ICASE, 1994. in progress.

[4] J.-F. M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design. *Structural Optimization*, 5:129–144, 1993.

[5] Richard P. Brent. Some efficient algorithms for solving systems of nonlinear equations. *SIAM Journal on Numerical Analysis*, 10(2):327–344, 1973.

[6] Kenneth M. Brown. A quadratically convergent Newton-like method based upon Gaussian elimination. *SIAM Journal on Numerical Analysis*, 6(4):560–569, December 1969.

[7] Maria R. Celis. *A Trust Region Strategy for Nonlinear Equality Constrained Optimization.* PhD thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1985; also available as TR85–4, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251–1892.

[8] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. On alternative problem formulations for multidisciplinary design optimization. Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992. Cleveland, Ohio.

[9] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary design optimization. *SIAM Journal on Optimization*, to appear 1994.

[10] J. E. Dennis, Jr., Mahmoud El-Alem, and Maria Cristina Maciel. A global convergence theory for general trust-region-based algorithms for equality constrained optimization. Technical Report TR92–28, Rice University, Department of Computational and Applied Mathematics, Houston, Texas, 1992. Revised April 1994.

[11] J. E. Dennis, Jr. and Virginia Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, November 1991.

[12] Mahmoud M. El-Alem. A global convergence theory for the Celis–Dennis–Tapia trust region algorithm for constrained optimization. *SIAM Journal on Numerical Analysis*, 28(1):266–290, 1991.

[13] Samuel K. Eldersveld. Automated part nesting for just-in-time manufacturing. *TechNet: Briefs on Computing Technology*, 7(3):7–8, June 1993.

[14] P. D. Frank, A. J. Booker, T. P. Caudell, and M. J. Healy. Optimization and search methods for multidisciplinary design. Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992. Cleveland, Ohio.

[15] P. D. Frank and G. R. Shubin. A comparison of optimization-based approaches for a model computational aerodynamics design problem. Technical Report Engineering Computing and Analysis Division ECA–TR–136–R1, Boeing Computer Services, October 1990.

[16] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[17] Thomas A. Grandine. Private communication, November 1992.

[18] R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula. On options for interdisciplinary analysis and design optimization. *Structural Optimization*, 4(2):65–74, 1992.

[19] Raphael T. Haftka, Zafer Gürdal, and Manohar P. Kamat. *Elements of Structural Optimization*. Kluwer Academic Publishers, 2nd edition, 1990.

[20] W. P. E. Gill, Murray, M. A. Saunders, and M. H. Wright. User's guide for npsol (version 4.0): a fortran package for nonlinear programming. Technical Report SOL 86-2, Department of Operations Research, Stanford University, 1986.

[21] G. R. Shubin. Application of alternate multidisciplinary optimization formulations to a model problem for static aeroelasticity. Technical Report BCSTECH-93-022, Boeing Computer Services, December 1993.

[22] J. Sobieszczanski-Sobieski. A linear decomposition method for large optimization problems—Blueprint for development. Technical Report TM 83248, NASA, February 1982.

[23] Trond Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[24] Iauw-Bhieng Tjoa and Lorenz T. Biegler. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial and Engineering Chemistry Research*, 30(2):376–385, 1991.

[25] Phillipe L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In Iain S. Duff, editor, *Sparse matrices and their uses*. Academic Press, 1981.

[26] Virginia Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, February 1991.

[27] Virginia Torczon. PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines. Technical Report TR92–09, Rice University, Department of Computational and Applied Mathematics, Houston, TX 77251–1892, 1992.

[28] Virginia Torczon. On the convergence of pattern search algorithms. Technical Report TR93–10, Rice University, Department of Computational and Applied Mathematics, Houston, Texas, 1993.