AIAA-2000-4720

AN ANALYSIS OF COLLABORATIVE OPTIMIZATION METHODS*

Angel-Victor DeMiguel† and Walter Murray‡
Department of Management Science and Engineering
Stanford University, Stanford, CA

Abstract

A feature common to many optimization problems is a weak connectivity between component parts. An attempt to exploit this feature has led to decomposition algorithms, which break the problem into that of solving a set of smaller independent problems. MDO fall into this category of problems. Such problems may need to be solved by some form of decomposition since the problem may not reside on a single machine. Benders decomposition algorithm is widely used to solve linear problems. Despite the fact that Benders decomposition has been generalized to the convex nonlinear case its use as such is rare. We show how generalized Benders decomposition can be applied to the convex MDO problem. Braun proposed a method he termed collaborative optimization (CO) to solve nonconvex MDO problems. Despite its intuitive appeal this and other decomposition algorithms have to surmount some serious technical challenges. We illustrate the weaknesses of Braun's approach and suggest an alternative that circumvents some of the difficulties.

1 Introduction

Many engineering design problems involve the optimization of an objective function subject to a set of constraints belonging to N different design disciplines. For instance, we might want to optimize an airplane design subject to constraints belonging to disciplines such as aerodynamics, structures and control. These problems are known in the literature as Multidisciplinary Design Optimization problems (MDO). MDO problems arise in the design of aircraft, spacecraft, automobiles, engines, and ships.[1, 15, 25]

*Work for this paper was supported by Office of Naval Research grant N000149610274.

†Graduate student (avmiguel@stanford.edu)
‡Professor (walter@stanford.edu)

Several computational and organizational aspects characteristic of MDO problems suggest the use of decomposition algorithms to solve them. Among the computational reasons is the fact that most optimization variables are needed for the evaluation of the constraints corresponding to only one of the design disciplines. Also, constraints belonging to each design discipline present a different structure and sparsity pattern. However, there is no doubt that the organizational aspects of MDO are the main reason for the use of decomposition algorithms. Often constraints belonging to each design discipline are evaluated through the use of a different piece of software running in separate locations and managed by different teams of engineers. Porting all these codes to the same hardware platform is usually judged impractical if not impossible.

Decomposition algorithms break the MDO problem into N independent subproblems. These N subproblems are solved at each iteration of an algorithm solving the so-called master problem. Information gathered at the solution of the subproblems is used to solve the master problem. Constraints and local variables corresponding to the i-th discipline are used only within the ith subproblem.

In this paper we show how Generalized Benders decomposition can be used to solve the convex MDO problem. Collaborative Optimization (CO) is a promising decomposition algorithm introduced by Braun[13] and further developed by Sobieski[36] to solve the general (non-convex) MDO problem. Unfortunately, no proof of convergence was given for the method and computational experience has shown difficulty in its application (see Alexandrov and Kodiyalam[2]). Moreover, CO presents some inherent analytical and numerical weaknesses and the specific formulation proposed by Braun is not ideal (see section 5 and Alexandrov and Lewis[4]). In this paper, we introduce a Modified Collaborative Optimization algorithm (MCO) that partially circumvents the difficulties associated with the original CO approach.

The remainder of this paper is organized as follows. In section 2, we state the MDO problem and discuss the need for a solution in a distributed environment. In section 3, we describe some of the classical decomposition algorithms available for large-scale optimization. In section 4, we show how to use Generalized Benders decomposition to solve the convex MDO problem. In section 5, we describe the algorithm proposed by Braun for general (nonconvex) MDO problems and study the difficulties associated with it. In section 6, we propose a Modified Collaborative Optimization algorithm.

2 The MDO problem

In this section we state the MDO problem and discuss the need for a solution in a distributed environment. Here we adhere to the spirit of Braun's work.[13] Alternative formulations of the MDO problem (see Cramer et al.[15], Balling and Sobieszczanski-Sobieski[7] and Alexandrov and Lewis[3]) can always be expressed in terms of our formulation at the expense of introducing some extra variables and constraints. In section 2.1 we introduce the *coupled MDO* problem, which is the form in which the MDO problem is encountered in most applications. The main characteristic of the coupled MDO problem is the strong linking among disciplines. As a consequence of this strong coupling, iteration is needed just to evaluate the constraints. In section 2.2, we present the *standard MDO* problem, which can be obtained from the coupled MDO problem by introducing additional variables and constraints. The advantage in the standard MDO problem is that iteration is no longer needed to evaluate the constraints. In section 2.3 we introduce a new instance of a so-called *Individual Discipline Feasible* (IDF) MDO formulation. In an IDF formulation, coupling among disciplines is mild and interdisciplinary feasibility is achieved through the use of penalty functions. Finally, in section 2.4, we discuss computational and organizational aspects of MDO problems that suggest the use of decomposition algorithms to solve them.

2.1 The coupled MDO problem

The *coupled MDO* problem is

$$\begin{array}{lll} \min\limits_{x,\bar{x}_i} & F(x) & \\ s.t. & c_i(x,\bar{x}_i,y_i) \ \geq \ 0, & i=1:N, \end{array} \qquad (1)$$

where

- $x \in \mathcal{R}^{\hat{n}}$ are *interdisciplinary* or *global* variables needed for the evaluation of constraints corresponding to several design disciplines.
- $\bar{x}_i \in \mathcal{R}^{n_i}$ are *disciplinary* or *local* variables needed only for the evaluation of the constraints corresponding to the ith design discipline.
- $y_i = [y_{i1}^T, \cdots, y_{i,i-1}^T, y_{i,i+1}^T, \cdots, y_{i,N}^T]^T \in \mathcal{R}^{\hat{n}_i}$ are the interdisciplinary inputs to the ith design discipline constraints. In order to evaluate the coupled MDO problem constraints $c_i(x,\bar{x}_i,y_i)$ at a given point $(x^T,\bar{x}_1^T,\cdots,\bar{x}_N^T)^T$, we need to compute the interdisciplinary inputs y_i through iteration as the solution to the following system of so-called *interdisciplinary constraints*:

$$y_{ij} = y_{ij}(x,\bar{x}_j,y_j), \quad i=1:N, \quad j \neq i. \qquad (2)$$

  The need for an iterative procedure is obvious from the coupling among the constraints in (2). In particular, y_{12} might be one of the inputs necessary to compute y_{21}, which might also be needed to compute y_{12}.
- $F(x) : \mathcal{R}^{\hat{n}} \to \mathcal{R}$ is the objective function for the overall design process.
- $c_i(x,\bar{x}_i,y_i) : \mathcal{R}^{\hat{n}+n_i+\hat{n}_i} \to \mathcal{R}^{\hat{m}_i}$ are the constraints corresponding to the i-th discipline.

Note that in order to solve the *coupled MDO* problem, we eliminate the interdisciplinary inputs y_i from the optimization problem by solving the set of interdisciplinary constraints defined in (2). By eliminating the interdisciplinary inputs we reduce the dimension of the coupled MDO problem. Unfortunately, the iterative procedure necessary to compute the interdisciplinary inputs greatly slows the optimization process. Moreover, elimination of variables may lead to instability when we use gradient-type algorithms to solve large-scale nonlinear problems (see Barclay et al.[9]).

2.2 The standard MDO problem

The difficulties associated with the iterative procedure needed to compute y_i can be avoided if we keep the interdisciplinary inputs, y_i, as optimization variables and we explicitly include the interdisciplinary constraints as constraints in our optimization problem. The result is the following problem:

$$\begin{array}{lll} \min\limits_{x,\bar{x}_i,y_i} & F(x) & \\ s.t. & c_i(x,\bar{x}_i,y_i) \geq 0, & i=1:N, \\ & y_{ij} = y_{ij}(x,\bar{x}_j,y_j), & i=1:N, \quad j \neq i. \end{array} \qquad (3)$$

The number of variables and constraints in problem (3) is larger than in problem (1) but the number of degrees of freedom is the same because the number of additional variables and constraints introduced are the same. Moreover, when we solve the MDO problem formulated as in (3), iteration is no longer needed to evaluate the constraints and the sparsity pattern in the constraints may be exploited. From a computational point of view, problem (3) is a more convenient formulation than (1).

Notice that in problem (3) the interdisciplinary constraints $y_{ij} = y_{ij}(x, \bar{x}_j, y_j)$ can be considered as part of $c_j(x, \bar{x}_j, y_j)$ because both sets of constraints depend on the same arguments. Also, the interdisciplinary inputs y_i can be considered as part of the interdisciplinary variables vector x, as they are needed in the evaluation of constraints belonging to all disciplines. In that case, the result is the *standard MDO* problem:

$$\begin{array}{ll} \min\limits_{x,\bar{x}_i} & F(x) \\ s.t. & c_i(x,\bar{x}_i) \;\; \geq \;\; 0, \quad i = 1:N, \end{array} \tag{4}$$

where $x \in \mathcal{R}^n$, with $n = \hat{n} + \sum_{i=1}^{N} \hat{n}_i$, $\bar{x}_i \in \mathcal{R}^{n_i}$, $F(x) : \mathcal{R}^n \to \mathcal{R}$, $c_i(x, \bar{x}_i) : \mathcal{R}^{n+n_i} \to \mathcal{R}^{m_i}$ and m_i is the total number of constraints (including the interdisciplinary constraints) corresponding to the ith discipline.

2.3 Individual Discipline Feasible MDO

The standard MDO problem is *interdisciplinary feasible* in the sense that at each iteration of the optimization process the constraints belonging to all design disciplines are evaluated at the same value for the interdisciplinary variables x. In many engineering design problems, constraints must be evaluated at a feasible point (a structural analysis for an airplane wing might completely fail if the set of aerodynamic loads is such that the wing would break). This might be a concern when trying to solve the standard MDO problem since during the optimization process we might arrive at a value for the interdisciplinary variables x such that no feasible $\bar{x}_i$ exists for the ith discipline.

This difficulty is usually overcome by the introduction of additional variables and constraints, leading to what is known as an Individual Discipline Feasible (IDF) formulation. In an IDF formulation a feasible point can always be found for each discipline at every iteration of the optimization process. Here, we propose a new IDF formulation that makes use of an l_1 penalty function:

$$\begin{array}{llcll} \min\limits_{z,x_i,\bar{x}_i,s_i,t_i,\bar{s}_i} & F(z) + \gamma \sum_{i=1}^{N} e^T(s_i + t_i) & & & \\ s.t. & c_i(x_i,\bar{x}_i) - \bar{s}_i & = & 0, & i = 1:N, \\ & x_i + t_i - s_i & = & z, & i = 1:N, \\ & s_i, t_i, \bar{s}_i & \geq & 0, & i = 1:N, \end{array} \tag{5}$$

where

- F, c_i and $\bar{x}_i$ are as defined for the standard MDO problem.
- $x_i \in \mathcal{R}^n$ are *interdisciplinary* or *global* variables that should converge to a single vector z but may take different values within each discipline during the solution process.
- $z \in \mathcal{R}^n$ is the target variable vector such that $x_i = z$ implies interdisciplinary feasibility.
- $s_i, t_i, \bar{s}_i$ are elastic variables (see Gill et al.[23] and Boman[12]).
- γ is the penalty parameter.
- $e \in \mathcal{R}^n$ is a vector of ones.

The term Individual Discipline Feasible refers to the fact that, if the original MDO problem is feasible, for any value of the target variables z, there exist $x_i, \bar{x}_i, s_i, t_i, \bar{s}_i$ feasible with respect to the IDF constraints. In other words, disciplinary feasibility can be achieved for any value of the target variables z. Interdisciplinary feasibility is achieved asymptotically by means of the use of l_1 penalty functions computed as $\|x_i - z\|_1 = e^T(s_i + t_i)$.

The IDF problem also presents the advantage that its objective and constraint functions are separable in the target variables z. The equivalence, in terms of KKT conditions, between the standard MDO problem and the IDF problem is easy to show following arguments by Boman[12] and will be addressed by DeMiguel.[17] Finally, the IDF problem will be the basis for the Modified Collaborative Optimization algorithm we propose in section 6.

2.4 Solution in a Distributed Environment

Despite the fact that there are strong computational advantages in the use of decomposition algorithms for the solution of MDO problems, there is no doubt that the organizational aspects of MDO problems are the main motivation for the use of decomposition.[29] For example, in the aircraft industry one division designs the wings, another the engine nacelles, another the fuselage, and so on. Each group must rely on complex legacy codes whose method of use is subject to constant modification. Porting all the code to a specific machine is judged to be impractical (sometimes the source code is no longer available). Also it would raise the issue of how local modifications to the use of such codes would be incorporated into the master problem.

Decomposition algorithms allow MDO problems to be solved in a distributed environment in the manner described above. The key point in the design of a decomposition algorithm in this environment is that only a limited communication between the subproblems and the master problem should be required. The aim is that different engineering teams should solve different subproblems and only a small amount of information need be sent to the coordinator solving the master problem. The approaches described in sections 4-6 are naturally suited to multiprocessor and/or distributed computing environments.

3 Decomposition in Optimization

In this section, we review the main ideas underlying classical decomposition algorithms for mathematical programming. The structure of some large-scale mathematical programs suggests they can be broken into a set of N independent subproblems or slave problems. The advantage is that usually the subproblems are structured and smaller than the original mathematical program. These N subproblems are solved at each iteration of an algorithm that solves the so-called master problem. Information gathered at the subproblem solutions is used to solve the master problem. Mathematical programs involving a master problem and subproblems are referred to in the literature as *bilevel programs.*[10, 19, 35] Unfortunately, bilevel programs are difficult to solve. In the sixties, Dantzig and Wolfe[16] and Benders[11] developed efficient ways to deal with the bilevel programming problems resulting from decomposition when problem functions are linear. Geoffrion[22] extended the work of Benders to the more general convex case.

Large-scale mathematical programs whose structure is suitable for decomposition are usually classified as block angular or dual angular depending on the structure of their constraints. In the remainder of this section we describe block-angular and dual-angular mathematical programs and discuss how they can be decomposed.

3.1 Block-Angular Mathematical Programs

The main characteristic of block-angular mathematical programs is that the variables are naturally classified as belonging to N different disciplines. A few of the constraints (known as interdisciplinary constraints) involve all of the variables, whereas the rest of the constraints (known as disciplinary constraints) only involve variables belonging to one of the disciplines. An example of block-angular mathematical programs is the resource allocation problem. If, in a block-angular mathematical program, we ignore the coupling constraints the mathematical program decomposes into N independent subproblems, one per discipline. A master problem is used to account for the coupling constraints while at each iteration of the master problem N subproblems corresponding to the N disciplines are solved. Dantzig and Wolfe[16] proposed an efficient decomposition algorithm for the case when the disciplinary constraints are linear.

3.2 Dual-Angular Mathematical Programs

In a dual-angular mathematical program, constraints are naturally classified into N different disciplines. Then some of the variables (known as global or interdisciplinary variables) are needed to evaluate all of the constraints, whereas the rest of variables (known as local or disciplinary variables) are needed only in the evaluation of the constraints belonging to one of the disciplines. Examples of dual-angular mathematical programs are the two-stage stochastic programming problem[27] and both the standard MDO problem and the IDF problem defined above.

If, in a dual-angular mathematical program, we set the interdisciplinary variables to a fixed value, the problem breaks into N independent subproblems (this process is known as projection). The optimal objective function to the ith subproblem as a function of the interdisciplinary variables is known as the optimal-value function. A master problem can be constructed using the optimal-value function for the N subproblems. Unfortunately, the resulting bilevel program is difficult to solve because of the nonsmoothness of the optimal-value function.

Generalized Benders decomposition[22] solves dual-angular problems whose objective and constraint functions are convex in the disciplinary variables. When using Generalized Benders decomposition we deal with the nonsmoothness of the subproblem optimal-value function by introducing constraints in the master problem, known as cuts, which define an outer linearization of the subproblem optimal-value function. The main limitation of Generalized

Benders decomposition is the fact that outer linearization can only be performed when the MDO constraint functions c_i are convex.

In section 4, we describe how Generalized Benders decomposition can be applied to solve the convex MDO problem. In sections 5 and 6, we describe decomposition algorithms that deal with the general (nonconvex) MDO problem.

4 Benders Decomposition

Most decomposition algorithms transform the original MDO problem into a bilevel programming problem in which the master problem involves the optimal value-function for the subproblems. Unfortunately, even when problem functions are smooth, the optimal-value function for the subproblems is, in general, non-smooth. Balling and Sobieszczanski-Sobieski[6] proposed the use of Kelley's Cutting Plane algorithm[28] to solve the resulting nonsmooth master problem. Computational experience demonstrated that cutting-plane algorithms are efficient to solve the convex MDO problem.[8] Generalized Benders decomposition is also based on a cutting-plane strategy. In this section, we describe how Generalized Benders decomposition can be used to solve the convex MDO problem in its IDF formulation (5). We also prove convergence to an approximate solution in a finite number of steps.

The remainder of this section is organized as follows. In section 4.1, we show how to obtain a master problem equivalent to the MDO problem in its IDF formulation using convex duality theory. In section 4.2, we describe an algorithm to find an ϵ-solution to the equivalent master problem. Finally, in section 4.3 we prove the algorithm converges in a finite number of iterations. For a more detailed study of Generalized Benders decomposition see Geoffrion.[22]

4.1 Master Problem

The desired master problem can be obtained from the IDF problem by a sequence of two manipulations: (i) projection, and (ii) dualization. If we project the IDF problem onto the target variables space we get the following master problem:

$$\min_{z \in V} \quad F(z) + \gamma \sum_{i=1}^{N} g_i^*(z), \tag{6}$$

where $g_i^*(z)$ is the optimal-value function for the ith subproblem,

$$\begin{array}{rrcl} \min\limits_{x_i, \bar{x}_i, s_i, t_i, \bar{s}_i} & \sum_{i=1}^N e^T(t_i + s_i) & & \\ s.t. & c_i(x_i, \bar{x}_i) - \bar{s}_i & = & 0, \\ & x_i + t_i - s_i & = & z, \\ & s_i, t_i, \bar{s}_i & \geq & 0. \end{array} \tag{7}$$

The equivalence between the master problem (6) and the IDF problem in terms of global minima follows from Theorem 2.1 in Geoffrion.[22] The main benefit obtained from projection is that now we deal with a master problem that only depends on the target variables z. Variables and constraints belonging to the ith discipline are needed only within the i-th subproblem.

The second transformation applied in the MDO problem on our way to the equivalent master problem makes use of convex duality theory. Given that the original MDO problem is feasible, it is easy to show that the ith subproblem defined in (7) is feasible for all z. Also, the ith subproblem is bounded below because the objective function is by definition nonnegative. It is well-known that a bounded and feasible convex nonlinear program has a convex set of local minimizers. Moreover, in a convex program every local minimizer is also a global minimizer. Therefore, we know from convex duality theory (see Theorem 2.3 in Geoffrion[22]) that the optimal-value functions, g_i^*, can be computed as the solution to the dual of the ith subproblem:

$$g_i^*(z) = \sup_{u_i \in \mathcal{U}_i} \{-u_{i2}^T z + \theta_i(u_i)\}, \tag{8}$$

where

- $u_i = (u_{i1}^T, u_{i2}^T, u_{i3}^T, u_{i4}^T, u_{i5}^T)^T$,
- $u_i \in \mathcal{U}_i$ iff $u_{i3}, u_{i4}, u_{i5} \leq 0$,
- $C_i(x_i, \bar{x}_i, s_i, t_i, \bar{s}_i) = ((c_i(x_i, \bar{x}_i) - \bar{s}_i)^T, (x_i + t_i - s_i)^T, s_i^T, t_i^T, \bar{s}_i^T)^T$,
- $\theta_i(u_i) = \inf\limits_{x_i, \bar{x}_i, s_i, t_i, \bar{s}_i} \{e^T(t_i + s_i) + u_i^T C_i(x_i, \bar{x}_i, s_i, t_i, \bar{s}_i)\}$.

The dual subproblems (8) provide an effective way to approximate the subproblem optimal-value functions. In particular, from (8) we know $g_i^*(z)$ is the supremum of an infinite set of linear functions of z. This is a crucial property in order to define a computational procedure to solve the master problem (6).

Using (8) and the definition of the supremum as the largest lower bound we can transform (6) into the following master problem:

$$\begin{array}{ll} \min\limits_{z\in V, y_i} & F(z) + \gamma \sum_{i=1}^{N} y_i \\ s.t. & y_i \geq -u_{i2}^T z + \theta_i(u_i), \quad i = 1:N, \quad \forall u_i \in \mathcal{U}_i, \end{array} \tag{9}$$

where the inequality constraints in (9) are known as optimality cuts. When all problem functions are linear, Benders[11] showed that only a finite number of feasibility cuts suffices to approximate (or outer linearize) $g_i^*(z)$. For general convex problems an infinite number of optimality cuts may be needed to represent $g_i^*(z)$ accurately.

4.2 Computational Procedure

In this section, we outline an algorithm leading to an approximate solution to the master problem (9) in a finite number of iterations. The main difficulty in solving (9) is the fact that it involves an infinite number of constraints. A natural strategy to deal with a large number of constraints is relaxation. First a relaxed master problem involving only a few of the constraints is defined and solved. If the solution to the relaxed master problem is not feasible with respect to all constraints then some of the violated constraints are added to the relaxed master problem and a new solution is obtained. We repeat this process until a feasible solution to all constraints is obtained. We may also delete constraints (cuts) in the updated master problem that are not active. However, care needs to be taken to avoid cycling. For example, we could remove an inactive cut only once.

In practice, the N subproblems defined in (7) are solved at the solution to the k-th relaxed master problem, z_k. As a result, new Lagrange multipliers, u_i, and parameters, $\theta_i(u_i)$, are generated and used to define new optimality cuts. The minimum to the relaxed master problem is a lower bound to the original MDO problem because we take only a few of the constraints into account. On the other hand, the optimal objective we obtain solving the subproblems is an upper bound because we do consider all of the constraints but we set the target variables, z, to a fixed value. The procedure is terminated when these lower and upper bounds are sufficiently close.

The approximate solution can be found in a three-step procedure:

- **Step 1.** Given a value of the target variables z, solve the N subproblems defined by (7) and generate optimal multiplier vectors u_{i2} and scalars $\theta_i(u_i)$ corresponding to each subproblem. Set $p = 1$, $u_{i2}^1 = u_{i2}$ $(i = 1:N)$, and set the objective upper bound $U_b = F(z) + \gamma \sum_{i=1}^{N} g_i^*(z)$. Select the convergence tolerance ϵ.
- **Step 2.** Solve the current relaxed master problem,

$$\begin{array}{lll} \min\limits_{z\in V, y_i} & F(z) + \gamma \sum_{i=1}^{N} y_i & \\ s.t. & y_i \geq -(u_{i2}^j)^T z + \theta_i(u_i^j), & i = 1:N, \\ & & j = 1:p. \end{array} \tag{10}$$

  Let $(\hat{z}, \hat{y}_i)$ be an optimal solution. Then $F(\hat{z}) + \sum_{i=1}^{N} \hat{y}_i$ is a lower bound on the optimal value of the IDF problem. If $U_b \leq F(\hat{z}) + \sum_{i=1}^{N} \hat{y}_i + \epsilon$, terminate.
- **Step 3.** Solve the N subproblems at $\hat{z}$. If $F(\hat{z}) + \gamma \sum_{i=1}^{N} g_i^*(\hat{z}) \leq F(\hat{z}) + \gamma \sum_{i=1}^{N} \hat{y}_i + \epsilon$, terminate. Otherwise, generate optimal multiplier vectors $\hat{u}_i$ and scalars $\theta_i(\hat{u}_i)$ corresponding to each subproblem. Increase p by 1 and set $u_{i2}^p = \hat{u}_{i2}$ $(i = 1:N)$. If $F(\hat{z}) + \gamma \sum_{i=1}^{N} g_i^*(\hat{z}) \leq U_b$ set the objective upper bound $U_b = F(\hat{z}) + \gamma \sum_{i=1}^{N} g_i^*(\hat{z})$. Return to step 2.

Remark 4.1 *Note that u_{i2} and θ_i can be computed by solving the primal subproblems defined in (7). In particular, u_{i2} are the Lagrange multipliers for the primal subproblems corresponding to the constraints $x_i + t_i - s_i = z$, and $\theta_i(u_i) = g_i^*(z) + u_{i2}^T z$.*

4.3 Finite Convergence

The following Theorem is a consequence of Theorem 2.5 in Geoffrion[22] and shows the algorithm defined in section 4.2 converges to an ϵ-solution in a finite number of steps.

Theorem 4.2 (Finite Convergence) *Assume V is a nonempty compact set, the feasible region for the ith subproblem for all z is a nonempty compact set, the MDO constraint functions c_i are convex and continuous, and the set of optimal multiplier vectors for the ith subproblem is nonempty for any z and uniformly bounded in some neighborhood of each such z. Then, for any given $\epsilon > 0$, the generalized Benders decomposition procedure terminates in a finite number of steps.*

Among the assumptions used in the above theorem, the most restrictive for real MDO problems is the one regarding convexity of the MDO problem functions. In the remainder of this paper we focus on the general (nonconvex) MDO problem.

5 Collaborative Optimization

In section 4, we used projection and dualization to obtain a master problem equivalent to the convex MDO problem. When we face the more general (nonconvex) MDO problem, only projection can be used because we lack a duality theory leading to efficient decomposition algorithms (see Flippo and Rinnooy Kan[21] for a discussion in nonconvex duality theory and decomposition). Collaborative Optimization (CO) is a decomposition algorithm proposed by Braun[13] and further developed by Sobieski[36] and Manning[30] that uses projection to transform the standard MDO problem into a bilevel programming problem consisting of a master problem (also known as system-level problem) and N subproblems, one per design discipline. Since each of the design disciplines is confined to one of the independent subproblems, CO allows a high level of modularity in the solution process. Unfortunately no convergence proof is known for CO and difficulty has been encountered in its application.[2] In this section we study some analytical and computational difficulties inherent to the CO approach proposed by Braun that explain its irregular behavior.

In section 5.1, we introduce Braun's formulation and discuss its main features. In section 5.2, we describe the analytical and numerical difficulties inherent in Braun's formulation. Finally, in section 5.3, we discuss the pros and cons of an alternative CO formulation proposed by Braun.[13]

5.1 Formulation

Braun proposed solving the following master problem:

$$\begin{array}{lll} \min\limits_{z} & F(z) & \\ s.t. & g_i^*(z) = 0, & i = 1:N, \end{array} \qquad (11)$$

where the optimal value-function $g_i^*(z)$ is implicitly defined as the optimal objective value for the ith subproblem,

$$\begin{array}{ll} \min\limits_{x_i,\bar{x}_i} & \frac{1}{2}(x_i - z)^T(x_i - z) \\ s.t. & c_i(x_i,\bar{x}_i) \geq 0, \end{array} \qquad (12)$$

where x_i, $\bar{x}_i$, F and c_i are as defined for the IDF problem (5).

Remark 5.1 *The master problem defined above only depends on the target variables, z. Variables and constraints belonging to the ith discipline are kept within the ith subproblem. Different subproblems (disciplines) are linked only through the target variables vector, z, which is considered as a parameter to each of the N subproblems.*

Remark 5.2 *Interdisciplinary feasibility (i.e. $x_i = z$) is achieved asymptotically through the use of quadratic penalty functions in the subproblems and by constraints in the master problem (requiring each penalty function to be zero). We may think of the interdisciplinary variables, x_i, as being an approximation to the target variables, z, in each subsystem.*

Remark 5.3 *Braun in his numerical experiments used a slightly more elaborate form than that given above but we retain this form for expository purposes.*

5.2 Analysis of CO

Two different aspects have to be considered to analyze a decomposition algorithm for the MDO problem.[15] First, one must prove that the set of solutions to the standard MDO problem is equivalent to the set of solutions to the master problem proposed. Braun[13] showed this is the case for the CO approach defined in (11). Second, convergence to a solution has to be proved for the optimization algorithms used for both the master problem and the subproblems. Unfortunately, convergence analysis of algorithms such as Sequential Quadratic Programming[32] (SQP) are based on assumptions that are not satisfied by either the master problem or the subproblems defined in (11) and (12).

In order to prove convergence to a solution for an optimization algorithm, two different issues have to be addressed. First, we have to ensure global convergence, which means we will converge to a solution from any starting point, possibly far from any solution. Second, we have to ensure fast local convergence. Fast local convergence means we must converge to the solution at a high rate (superlinear or quadratic) once we are sufficiently close to a minimizer.

In the remainder of this section, we discuss the properties associated with Braun's CO formulation that violate the hypothesis of available convergence theory for conventional optimization algorithms. These properties explain the irregular behavior of CO when applied to real problems. Below we catalog some of the properties of Braun's formulation.

5.2.1 Nonsmoothness

Unfortunately, the optimal value-function, $g_i^*(z)$, is not smooth in general. Fiacco and McCormick[20] showed that the optimal value-function to a parametric nonlinear program is smooth at a point z if the solution at z satisfies certain so-called nondegeneracy conditions (in particular, smoothness is ensured if the solution satisfies Second Order Sufficient Conditions (SOSC), Linear Independence Constraint Qualification (LICQ) and Strict Complementarity Slackness (SCS)). Falk and Liu[19] argue that it is not reasonable to assume that the solution to a parametric nonlinear program satisfies all these three conditions at all points. As a consequence, the optimal value-functions, g_i^*, are not differentiable in general. The nondifferentiability of the optimal-value function hinders local as well as global convergence proofs for most optimization algorithms applied to the master problem.

5.2.2 Singular Jacobian

Another concern when we try to prove local convergence for the optimization algorithm applied to the master problem is the fact that, even if we assumed $g_i^*(z)$ are smooth, the Jacobian for the master problem constraints is singular at the solution. Assuming the optimal-value functions, $g_i^*(z)$, are smooth, Braun[13] shows that the gradients for the master problem constraints can be computed analytically as

$$\nabla g_i^*(z) = -(x_i^* - z).$$

Clearly, even when each $g_i^*(z)$ is smooth the Jacobian for the master problem constraints is singular at the solution (indeed it becomes the zero matrix since at the solution $x_i = z$). Even if an algorithm does converge for the master problem, the singularity of the Jacobian at the solution is bound to impact the rate of convergence adversely.

5.2.3 Multiple Subproblem Solutions

Since we are dealing with nonconvex MDO problems, there might exist different local minimizers for each subproblem at each z. Therefore, the optimal-value function $g_i^*(z)$ is in fact an optimal set-valued function (see Aubin and Frankowska[5] for a reference on set-valued analysis). No global convergence proof is known for optimization algorithms involving set-valued constraint functions.

Fiacco and McCormick[20] showed that if at a point z all local solutions to a subproblem are nondegenerate (i.e., SOSC, SCS, LICQ hold) then there exist smooth disjoint trajectories of local solutions in a neighborhood of z. In that case, the set-valued function $g_i^*(z)$ can be considered as composed of several functions corresponding to each isolated trajectory. Unfortunately, most optimization algorithms for the master problem would fail to converge if when solving the subproblems at different points we obtain local solutions belonging to different trajectories. In particular, a sufficient decrease might not be achieved if when performing a line-search we jump from one trajectory to another. Likewise, trust-region algorithms might fail if when checking the accuracy of the local model we evaluate g_i^* at a different trajectory than the one we used to derive the local quadratic approximation to the problem.

5.2.4 Subproblem Lagrange multipliers

Also troublesome is the fact that the Lagrange multipliers of the subproblems are either zero or converge to zero as z converges to z^*. (This follows immediately from the fact that $\nabla g_i^*(z^*) = 0$.) Again a typical assumption to prove convergence of an algorithm is that strict complementarity slackness (SCS) holds at the solution. Difficulty can therefore be expected in solving the subproblems.

5.2.5 Master Problem Active Set

A somewhat more subtle problem is the difficulty in identifying the active set of the master problem. It may seem odd to suggest difficulty in identifying the active set when only equality constraints are present. However, a worrying feature of the subproblems is that they fail to distinguish the case when $x_i^* = z$ is only just feasible, from the case when any change in z of sufficiently small magnitude would still result in $x_i^* = z$. In other words, this formulation is not able to identify those constraints in the master problem that are truly constraining the solution from those that are not.

5.3 An Alternative CO Formulation

In an attempt to avoid the singularity of the Jacobian, Braun proposed an alternative CO formulation. The idea is to solve the following master problem:

$$\begin{array}{lll} \min\limits_{z} & F(z) & \\ s.t. & x_i^*(z) - z = 0, & i = 1:N, \end{array} \tag{13}$$

where the subproblem optimizer $x_i^*(z)$ is implicitly defined as the optimal value for the interdisciplinary variables for the i-th subproblem,

$$\begin{array}{ll} \min\limits_{x_i, \bar{x}_i} & \frac{1}{2}(x_i - z)^T (x_i - z) \\ s.t. & c_i(x_i, \bar{x}_i) \geq 0. \end{array} \tag{14}$$

The only difference is in the constraints to the master problem. Here, nonlinear constraints are used in the master problem to force the target variables z to converge to the optimal interdisciplinary variables $x_i^*(z)$ for the subproblems, whereas in (11) nonlinear constraints are used to drive the optimal-value function to zero.

The Jacobian for the master problem constraints is

$$\begin{pmatrix} \frac{dx_1^*}{dz} - I \\ \frac{dx_2^*}{dz} - I \\ \vdots \\ \frac{dx_N^*}{dz} - I \end{pmatrix}. \tag{15}$$

An immediate consequence of this formulation is that some SQP methods would have difficulty solving the master problem because their QP subproblems would in general have no feasible solution. Moreover, QP subproblems are likely to have many degenerate vertices and few QP solvers cope with degeneracy. To appreciate why these difficulties arise, note that the number of constraints in the master problem will in general be much larger than the number of variables. The QP subproblems will have the same number of variables, and the number of linear equality constraints is the same as the number of nonlinear constraints. Ironically, Braun's use of SQP in his original formulation did not suffer from this difficulty because the number of linear constraints in the QP subproblems is small compared to the number of variables in the problems addressed. Had that not been the case, difficulties would have arisen.

Another concern is the fact that in order to compute the master problem Jacobian we need to solve the following system for $\frac{dx_i^*}{dz}$:

$$\begin{pmatrix} \nabla_{xx}^2 L_i^* & -(A_i^*)^T \\ A_i^* & 0 \end{pmatrix} \begin{pmatrix} \frac{dx_i^*}{dz} \\ \frac{d\lambda_i^*}{dz} \end{pmatrix} = \begin{pmatrix} -\nabla_z(\nabla_x L_i^*) \\ 0 \end{pmatrix}, \tag{16}$$

where L_i is the Lagrangian function for the ith subproblem, A_i is the Jacobian for the ith subproblem constraints, λ_i are the multipliers for the subproblem constraints and the asterisk (*) means evaluated at the optimum. Most conventional optimization algorithms, for instance SQP algorithms, require accurate first-order derivatives to ensure fast convergence. Therefore when solving (16) we need an accurate of $\nabla_{xx}^2 L_i^*$. Such an accurate estimate is not always available when we deal with large-scale problems. Therefore finite differences may be necessary to compute $\frac{dx_i^*}{dz}$. This may not be a great computational effort because the number of target variables z is usually small compared to the number of disciplinary variables, $\overline{x}_i$. Moreover, a warm start can accelerate the finite-differencing process because we would just need to repeatedly solve the N subproblems for close values of z. However, we would still need to deal with the rest of the difficulties discussed before (nonsmoothness, multiple local solutions to the subproblems, etc.).

6 Modified CO

As we discussed in section 5, the original CO approach proposed by Braun has several properties that explain the poor behavior observed on some test-problems[2] and prevents the development of a suitable convergence theory. Nonetheless, among all known decomposition algorithms for the general (nonconvex) MDO problem, CO is the one allowing the greatest level of modularity in the solution. Moreover, CO has been successfully applied to the solution of some real problems.[30, 36] In this section we propose a new CO formulation, we called Modified Collaborative Optimization (MCO), that overcomes some of the difficulties associated with previous approaches while keeping the desired level of modularity provided by CO. Firstly, we propose the use of an l_1 exact penalty function[33] instead of the quadratic penalty function used by Braun. As a result, we avoid the singularity of the master problem Jacobian, and the subproblem Lagrange multipliers are no longer zero at the solution. Then, in order to deal with the nonsmooth optimal-value functions, we propose solving a sequence of perturbed MCO problems that have better smoothness properties than the original MCO problem.

In section 6.1, we introduce the Modified Collaborative Optimization formulation. In section 6.2, we analyze the characteristics of the MCO formulation. In section 6.3, we discuss how bundle methods can be used to solve the nonsmooth MCO problem. Finally, in section 6.4, we propose an algorithm based on solving a sequence of perturbed MCO subproblems presenting better smoothness properties than the original MCO problem.

6.1 An Exact Penalty formulation

The quadratic penalty function was first described by Courant[14] and is widely used in engineering applications because of its intuitive appeal. Unfortunately, quadratic penalty functions have been associated in the specialized literature with numerical instability (see Gill et al.[24]). Two of the difficulties associated with CO are directly related to the use of quadratic penalty functions as the objective function for the subproblems. In particular, both the singularity of the master problem Jacobian and the fact that all Lagrange multipliers for the subproblems are zero at the solution are a consequence of the use of quadratic penalty functions. In this section we propose the use of an l_1 exact penalty function instead of the troublesome quadratic penalty function. We call this form the Modified Collaborative Optimization problem (MCO). Namely, solve the following master problem:

$$\min_{z} \quad F(z) + \gamma \sum_{i=1}^{N} g_i^*(z) \ ,$$

where $g_i^*(z)$ is the optimal-value function to the i-th subproblem,

$$\begin{array}{lrcl} \min\limits_{x_i,\bar{x}_i,s_i,t_i,\bar{s}_i} & e^T(s_i+t_i) & & \\ s.t. & c_i(x_i,\bar{x}_i)-\bar{s}_i & = & 0, \\ & x_i+s_i-t_i & = & z, \\ & s_i,t_i,\bar{s}_i & \geq & 0. \end{array}$$

Remark 6.1 *The l_1 exact penalty function is defined as the 1-norm of the vector $(x_i - z)$, i.e. $\|x_i - z\|_1 = \sum_{j=1}^{n} |x_{ij} - z_j|$. However, in order to avoid discontinuities in the subproblem objective function derivatives rather than using the l_1 exact penalty function explicitly, elastic variables, t_i and s_i, are introduced. Then, it can be shown that the l_1 exact penalty function can be computed as $\|x_i - z\|_1 = e^T(s_i + t_i)$.*

Remark 6.2 *In MCO we use the subproblem optimal-value functions as penalty terms for the master problem objective function rather than imposing constraints in the master problem enforcing the subproblem optimal-value function to be zero. As a result we deal with an unconstrained master problem.*

Remark 6.3 *Notice the close relationship between the MCO problem defined and the IDF formulation we introduced in section 2. In particular, one may think of the MCO problem as the projection of the IDF problem onto the target variables space.*

6.2 Analysis of MCO

6.2.1 Subproblem Multipliers

At the solution to the MCO subproblems the Lagrange multipliers are no longer necessarily zero. This is obvious from the fact that the subproblem objective function gradient is no longer zero at the solution (in fact, the subproblem objective gradient is constant).

6.2.2 Subproblem Sensitivity

At points z such that the solution to the ith subproblem is nondegenerate, the gradient of the subproblem optimal-value functions can be computed from the Lagrange multipliers of the subproblems. In particular, $\frac{\partial g_i}{\partial z_k}$ is equal to the multiplier corresponding to the subproblem constraint,

$$x_{ik} + t_{ik} - s_{ik} = z_k.$$

This is of great importance for the definition of the master problem optimization algorithm because most optimization algorithms applied to the subproblems will generate accurate estimates of the subproblem Lagrange multipliers. Therefore at the solution of the subproblem we automatically get an estimate of the master problem first-order derivatives.

6.2.3 Master Problem Singularity

When the subproblem solution is nondegenerate, the gradient of the subproblem optimal-value functions no longer converges to zero as the iterates converge to the solution. This follows immediately from the fact that the subproblem Lagrange multipliers are not zero in general and the optimal-value function gradients are obtained from the subproblem Lagrange multipliers.

6.2.4 Nonsmoothness

In the previous sections we discussed how MCO overcomes several of the difficulties associated with CO. Unfortunately, we still need to deal with the nonsmoothness of the subproblem optimal-value functions. Moreover, multiple local solutions might exist to the MCO subproblems when the problem functions are nonconvex.

6.3 Bundle Methods

Bundle methods can be used to compute the solution to the nonsmooth MCO master problem defined above. Bundle methods are optimization methods suitable for the solution of optimization problems where the objective and constraint functions are not differentiable but are Lipschitz continuous.[26, 34] When the global minima to the subproblems are computed the MCO optimal-value functions are Lipschitz continuous.[17] Even if only local minima are available, it can be shown that under mild nondegeneracy conditions the optimal-value function for the MCO subproblems is Lipschitz continuous.[35] Therefore bundle methods can be applied to solve the nonsmooth MCO master problem. This approach has been applied before to bilevel programming problems.[18, 19, 35]

6.4 Smoothing Algorithms

Unfortunately, bundle methods are known to behave poorly in practice. As an alternative, we propose solving a sequence of perturbed MCO problems that have better smoothness properties than the original MCO problem and for which algorithms for smooth optimization can be applied.

6.4.1 Perturbed Problem

The idea is to solve the following perturbed problem for a decreasing sequence of barrier parameters $\{\mu_k\}$ such that $\lim_{k\to\infty}\mu_k = 0$:

$$\min_{z} \quad F(z) + \gamma \sum_{i=1}^{N} g_i^*(\mu, z)$$

where $g_i^*(\mu, z)$ is implicitly defined as the the optimal value-function for the i-th subproblem,

$$\begin{array}{lrcl} \min\limits_{x_i,\bar{x}_i,s_i,t_i,\bar{s}_i} & e^T(s_i+t_i) - \mu\phi(s_i,t_i,\bar{s}_i) & & \\ s.t. & c_i(x_i,\bar{x}_i) - \bar{s}_i & = & 0, \\ & x_i + s_i - t_i & = & z, \end{array}$$

where the barrier function, ϕ, is defined as

$$\phi(s_i,t_i,\bar{s}_i) = \sum_{j=1}^{n}(\log s_{ij} + \log t_{ij}) + \sum_{j=1}^{m_i}\log \bar{s}_{ij}.$$

The perturbed MCO problem is obtained from the MCO problem by introducing barrier terms into the subproblem objective functions to make sure the non-negativity constraints never become active. Consequently, the discontinuities that arise in derivatives of $g_i^*(z)$ due to changes in the active set of the subproblem do not occur in this formulation.

6.4.2 Analysis

Since the perturbed MCO subproblems only have equality constraints, the Strict Complementarity Slackness (SCS) conditions are automatically satisfied at any solution. Moreover, the LICQ is obviously satisfied because the Jacobian for the MCO subproblems clearly has full rank. Finally, the Second Order Sufficient Conditions for optimality (SOSC) are likely to hold at the solution to the subproblems because in an interior-point formulation, usually isolated minima are attained (we can always force the reduced Lagrangian Hessian to be positive definite by adding the barrier terms $\log(u_{ij} - x_{ij})$). As a result, we can expect the subproblem solutions to be nondegenerate (i.e. satisfy LICQ, SCS, SOSC) and therefore we can expect the perturbed optimal-value functions have the smoothness properties required to show convergence for most optimization algorithms.

The difficulty is now that we need to deal with the ill-conditioning inherently associated with the use of barrier functions. In particular, both the master problem Hessian and the subproblem Lagrangian Hessian can be shown to be ill-conditioned as the barrier parameter, μ, converges to zero (see Murray[31] and Wright[37]). Nevertheless, we believe the behavior of commercial SQP software such as SNOPT[23] on ill-conditioned problems is better than the behavior of bundle methods on nonsmooth problems. In addition we can pose well-conditioned QP subproblems using the primal-dual form of the optimality conditions.

Other concerns are the fact that there might still exist multiple solutions to the subproblems, and the need for an adequate update rule for the barrier parameter μ.

7 Conclusions

We show how to use Generalized Benders decomposition to solve the convex MDO problem. For the nonconvex case, we propose a Modified Collaborative Optimization architecture with inherent properties that are necessary if convergence is to be proven. Future work will address a computational comparison between CO and MCO and a rigorous theoretical treatment of issues such as the smoothness of the perturbed master problem objective function and global convergence of the master problem algorithm.

8 References

[1] N.M. Alexandrov and M.Y. Hussaini, editors. *Multidisciplinary Optimization: State of the Art*, Philadelphia, 1997. SIAM.

[2] N.M. Alexandrov and S. Kodiyalam. Initial results of an MDO method evaluation study. In *Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998. AIAA Paper 98-4884.

[3] N.M. Alexandrov and R.M. Lewis. Comparative properties of collaborative optimization and other approaches to MDO. In *Proc., First ASMO UK/ISSMO conference on Engineering Design Optimization*. MCB Press, jul 1999.

[4] N.M. Alexandrov and R.M. Lewis. Analytical and computational aspects of collaborative optimization. Technical Report TM-2000-210104, NASA, 2000.

[5] J.P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhauser, 1990.

[6] R.J. Balling and J. Sobieszczanski-Sobieski. An algorithm for solving the system-level problem in multilevel optimization. *Structural Optimization*, 9:168–177, 1995.

[7] R.J. Balling and J. Sobieszczanski-Sobieski. Optimization of coupled systems: a critical overview of approaches. *AIAA Journal*, 34(1):6–17, 1996.

[8] R.J. Balling and C.A. Wilkinson. Execution of multidisciplinary design optimization approaches on common test problems. *AIAA Journal*, 35:178–186, 1997.

[9] A. Barclay, P.E. Gill, and J.B. Rosen. SQP methods and their application to numerical optimal control. Technical Report Report NA 97-3, Department of Mathematics, University of California, San Diego, 1998.

[10] J.F. Bard. *Practical bilevel optimization: Algorithms and applications*. Kluwer academic publishers, 1998.

[11] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[12] E.G. Boman. *Infeasibility and negative curvature in optimization*. PhD thesis, Stanford University, 1999.

[13] R.D. Braun. *Collaborative Optimization: An architecture for large-scale distributed design*. PhD thesis, Stanford University, 1996.

[14] R. Courant. Variational methods for the solution of problems with equilibrium and vibration. *Bull. Amer. Math. Soc.*, 49:1–23, 1943.

[15] E.J. Cramer, J.E. Dennis, P.D. Frank, R.M. Lewis, and G.R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM J. Optimization*, 4(4):754–776, November 1994.

[16] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[17] A.V. DeMiguel. *Collaborative Optimization: A decomposition algorithm for dual-angular nonlinear problems*. PhD thesis, Stanford University, In preparation.

[18] S. Dempe. A bundle algorithm applied to bilevel programming problems with non-unique lower level solutions. *Computational Optimization and Applications*, 15(2):145–166, 2000.

[19] J.E. Falk and J. Liu. On bilevel programming, part I: general nonlinear cases. *Mathematical Programming*, 70:47–72, 1995.

[20] A.V. Fiacco and G.P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. John Wiley and Sons, 1968.

[21] O.E. Flippo and A.H.G. Rinnooy Kan. Decomposition in general mathematical programming. *Mathematical Programming*, 60:361–382, 1993.

[22] A.M. Geoffrion. Generalized Benders decomposition. *JOTA*, 10(4), 1972.

[23] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report, Dept. of Mathematics, University of California, 1997. To appear in *SIAM J. Optimization*.

[24] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.

[25] R.T. Haftka and J. Sobieszczanski-Sobieski. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, 14:1–23, 1997.

[26] J. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and minimization algorithms*. Springer-Verlag, 1993.

[27] G. Infanger. *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. Boyd and Fraser, 1994.

[28] J.E. Kelley. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[29] I.M. Kroo. Decomposition and collaborative optimization for large-scale aerospace design programs. In N.M. Alexandrov and M.Y. Hussaini,

editors, *Multidisciplinary Design Optimization: State of the art*, 1997.

[30] V.M. Manning. *Large-scale design of supersonic aircraft via collaborative optimization.* PhD thesis, Stanford University, 1999.

[31] W. Murray. Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. *JOTA*, 7:189–196, 1971.

[32] W. Murray and F.J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. Optimization*, 1995.

[33] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer-Verlag, 1999.

[34] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optimization*, 2:121–152, 1992.

[35] K. Shimizu, Y. Ishizuka, and J.F. Bard. *Nondifferentiable and two-level mathematical programming.* Kluwer academic publishers, 1997.

[36] I.P. Sobiesky. *Multidisciplinary design using collaborative optimization.* PhD thesis, Stanford University, 1998.

[37] M.H. Wright. Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM J. Optimization*, 9:84–111, 1999.