A98-39772

AIAA-98-4795

# HIERARCHICAL OVERLAPPING COORDINATION UNDER NONLINEAR CONSTRAINTS

Nestor Michelena* Hyungju A. Park†
Panos Papalambros * Devadatta Kulkarni †

## ABSTRACT

Decomposition of large engineering design problems into smaller design subproblems enhances robustness and speed of numerical solution algorithms. Design subproblems can be solved in parallel, using the optimization technique most suitable for the underlying subproblem. This also reflects the typical multidisciplinary nature of system design problems and allows better interpretation of results.

Hierarchical overlapping coordination (HOC) simultaneously uses two or more problem decompositions, each of them associated with different partitions of the design variables and constraints. Coordination is achieved by the exchange of information between decompositions.

This article presents the HOC algorithm and a sufficient condition for convergence of the algorithm to the optimum in the case of convex problems with nonlinear constraints. The convergence condition involves the rank of a matrix derived from the Jacobian of the constraints. Computational results obtained by applying the HOC algorithm to nonlinear convex programming problems of various sizes are also presented.

## 1 Introduction

A typical approach to engineering design consists of formulating an optimization problem using models to estimate design criteria and constraint functions, and applying formal optimization methods to search the design space for an optimum.

In this article, we consider design problems that can be formulated as convex optimization problems of the form

$$\begin{aligned}&\text{find } \mathbf{x} \in X \text{ such that}\\ &\mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0 \text{ and } f(\mathbf{x}) \text{ is minimized,} \quad (1.1)\end{aligned}$$

where $X \subset \mathbb{R}^n$ is a nonempty open convex set, $f : X \to \mathbb{R}$ and $g_i : X \to \mathbb{R}$ are convex and differentiable functions on $X$, and $h_i : X \to \mathbb{R}$ are affine functions on $X$.

Although optimization methods have been applied with practical success to individual system components, difficulties arise for system level design—a system being a collection of connected components or processes. Exploiting the structure of the design problem by decomposition of the problem into smaller subproblems may be necessary in the case of systems that involves hundreds of variables and constraints. The subproblems are then solved in parallel, using the optimization technique most suitable for the underlying submodel, gaining in robustness, speed and interpretation of results. Moreover, system design problems are typically multidisciplinary and/or involve subsystem design teams that effect one or more explicit problem decompositions. Thus, coordinated solution of design subproblems may be the only way to address the overall system problem in a practical and robust manner.

Hierarchical overlapping coordination (HOC) simultaneously uses two or more design problem decompositions, each of them associated with different partitions of the design variables and constraints. This kind of problem decomposition may reflect, for example,

*Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109. e-mail: {nestorm, pyp}@umich.edu

†Department of Mathematics and Statistics, Oakland University, Rochester, MI 48309. e-mail: {park, kulkarni}@oakland.edu.

matrix-type organizations structured according to product lines or subsystems and the disciplines involved in the design process. Coordination is achieved by the exchange of information between decompositions.

The mathematical formulation of HOC was first proposed in [6], and several criteria for convergence of the coordination algorithm under linear equality and inequality constraints were developed in [6] and [11]. Convergence criteria developed in those articles are computationally difficult to check and possibly incorrect (see Remark 4.4 in [10]). In order to remedy the situation, new computationally efficient criteria for convergence of the HOC algorithm under linear constraints were developed in [10]. In the present article, we present an HOC algorithm and a condition that ensure the convergence of the algorithm for nonlinear constraints.

Several researchers have proposed coordination strategies to exploit the structure of a problem associated with its decomposition. Reviews of optimization procedures that use decomposition are presented by Wagner and Papalambros [14] and Sobieszczanski-Sobieski and Haftka [13]. Recently, Nelson and Papalambros [5] presented Sequentially Decomposed Programming as a globally convergent coordination scheme for hierarchic systems. Other promising coordination algorithms, including concurrent subspace optimization (CSSO) [12] and collaborative optimization (CO) [3] for non-hierarchic systems, still require further in-depth study of their robustness and convergence properties.

## 2 Hierarchical Overlapping Coordination

Dependence of design functions on variables can be represented by a Boolean matrix termed the functional dependence table (FDT). The $(i,j)$-th entry of the FDT is one if the $i$-th function depends on the $j$-th variable, and zero otherwise.

Hypergraph-based model decomposition [8] can be applied to the constraint functions of problem (1.1) to obtain two or more distinct (say, α-, β-, γ-, ...) decompositions. This involves generating a decomposition of the functional dependence table (FDT) by reordering the variables and constraints, as shown in Figure 1.

In Figure 1, $\mathbf{x}_{\alpha_i}$ is the vector of local variables associated with block $A_{\alpha_i}$, i.e., with subproblem $\alpha_i$; $\mathbf{y}_\alpha$ is the vector of $n_\alpha$ linking variables for the α-decomposition; and $p_\alpha$ denotes the number of subproblems in the α-decomposition (diagonal blocks in the figure).

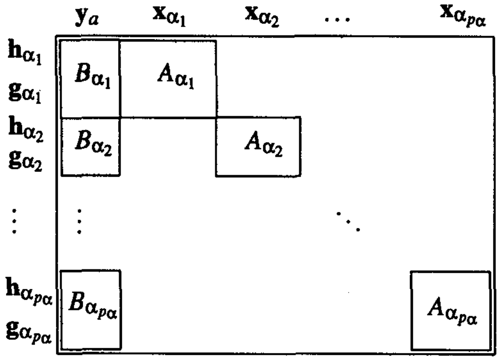

Figure 1. Block α-decomposition of functional dependence table

Provided that the objective function $f$ is α-additive separable[1], problem (1.1) takes the following form:

$$\text{Min}_{\mathbf{x}}\ f_{\alpha_0}(\mathbf{y}_\alpha)+\sum_{i=1}^{p_\alpha} f_{\alpha_i}(\mathbf{y}_\alpha,\mathbf{x}_{\alpha_i})\ \text{subject to} \quad (2.1)$$
$$\mathbf{h}_{\alpha_i}(\mathbf{y}_\alpha,\mathbf{x}_{\alpha_i})=\mathbf{0},\ \mathbf{g}_{\alpha_i}(\mathbf{y}_\alpha,\mathbf{x}_{\alpha_i})\le\mathbf{0},\ i=1,\dots,p_\alpha.$$

For a given vector $\mathbf{d}_\alpha\in\mathbb{R}^{n_\alpha}$, fixing the α-linking variables $\mathbf{y}_\alpha=\mathbf{d}_\alpha$ in (2.1) results in the following Problem α:

$$\begin{array}{l}\textbf{Problem } \alpha:\\ \quad\text{For each } i=1,\dots,p_\alpha,\\ \quad\text{Min}_{\mathbf{x}_{\alpha_i}}\ f_{\alpha_i}(\mathbf{d}_\alpha,\mathbf{x}_{\alpha_i})\ \text{subject to}\\ \quad\mathbf{h}_{\alpha_i}(\mathbf{d}_\alpha,\mathbf{x}_{\alpha_i})=\mathbf{0},\ \mathbf{g}_{\alpha_i}(\mathbf{d}_\alpha,\mathbf{x}_{\alpha_i})\le\mathbf{0}.\end{array} \quad (2.2)$$

Problem α can be solved by solving $p_\alpha$ independent uncoupled subproblems. Similarly, Problem β can be defined and solved for a β-decomposition after fixing the β-linking variables.

The hierarchical overlapping coordination algorithm can be described as follows, for the case of two decompositions (α and β):

**Generic HOC Algorithm**

Step 1. Fix linking variables $\mathbf{y}_\alpha$, and solve Problem α by solving the $p_\alpha$ independent subproblems given in (2.2).

Step 2. Fix linking variables $\mathbf{y}_\beta$ to their values determined in Step 1, and solve Problem β by solving $p_\beta$ independent subproblems.

Step 3. Go to Step 1 with the fixed values of α-linking variables determined in Step 2.

Step 4. Repeat these steps until convergence is achieved.

[1] In general, HOC can be used if the objective function can be written as monotonic functions of local objective functions derived from the α- and β-decompositions.

## 3 Convergence Under Linear Constraints

In general, the accumulation point achieved in Step 4 of the Generic HOC Algorithm in Section 2 is not necessarily an optimal solution of Problem 1.1. A condition that guarantees convergence of the HOC algorithm to an optimal solution will be referred to as an *HOC convergence condition.* For linearly constrained problems, several equivalent HOC convergence conditions were developed in an earlier work [10], one of them being notably efficient in a computational sense. This result for the case of linear constraints is reviewed in this section.

Consider the following optimization problem under linear equality and inequality constraints:

$$\mathrm{Min}_{\mathbf{x}}\ f(\mathbf{x}) \text{ subject to } A^I\mathbf{x} \leq \mathbf{c}^I \text{ and } A^E\mathbf{x} = \mathbf{c}^E \qquad (3.1)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, $A^I$ ($A^E$, resp.) is an $m_I \times n$ ($m_E \times n$, resp.) constraint matrix with real entries, $\mathbf{x} \in \mathbb{R}^n$ is the vector of optimization variables, and $\mathbf{c}^I \in \mathbb{R}^{m_I}$ ($\mathbf{c}^E \in \mathbb{R}^{m_E}$, resp.) is a constant vector. We assume that the problem has a nonempty solution set.

The Generic HOC Algorithm described in Section 2 applied to Problem 3.1 results in two sequences $\{\mathbf{x}_{\alpha_i}\}_{i=1}^{\infty}$ and $\{\mathbf{x}_{\beta_i}\}_{i=1}^{\infty}$. For an accumulation point $\mathbf{x}^*$ of $\{\mathbf{x}_{\alpha_i}\}_{i=1}^{\infty}$ or $\{\mathbf{x}_{\beta_i}\}_{i=1}^{\infty}$, define $T_a$ to be the set of the indices corresponding to the active inequality constraints, i.e.,

$$T_a := \{i \mid (a^I_{i1}, \ldots, a^I_{in})\mathbf{x}^* = c^I_i\},$$

where $a^I_{ij}$ denotes the $(i, j)$-entry of the matrix $A^I$. Let $\bar{A}^I$ be the submatrix of $A^I$ consisting of the active inequality constraints.

Define the cone $\mathrm{C}(A)$ by

$$\mathrm{C}(A) := \{\mathbf{x} \mid \mathbf{x} = \sum_{i \in T_a} a_i \mathbf{v}^I_i + \sum_{i=1}^{m_E} b_i \mathbf{v}^E_i,\ a_i \geq 0\},$$

where $\mathbf{v}^I_i$ ($\mathbf{v}^E_i$, resp.) denotes the $i$-th row vector of $A^I$ ($A^E$, resp.). Also, define the induced cones $\mathrm{C}(K_\alpha)$ and $\mathrm{C}(K_\beta)$ as follows:

$$\mathrm{C}(K_\alpha) := \{\mathbf{x} \mid \mathbf{x} = \sum_{i \in T_a} a_i \mathbf{v}^I_i + \sum_{i=1}^{m_E} b_i \mathbf{v}^E_i + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)}, a_i \geq 0\},$$

$$\mathrm{C}(K_\beta) := \{\mathbf{x} \mid \mathbf{x} = \sum_{i \in T_a} a_i \mathbf{v}^I_i + \sum_{i=1}^{m_E} b_i \mathbf{v}^E_i + \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)}, a_i \geq 0\},$$

where $\mathbf{e}_i \in \mathbb{R}^n$ is the $i$-th standard row vector, and $\alpha(i)$ ($\beta(i)$, resp.) is the index for the $i$-th $\alpha$-linking ($\beta$-linking, resp.) variable.

The Lagrange multiplier theorem for linear constraints [2, Proposition 3.4.1] states that $\mathbf{x}^* \in \mathbb{R}^n$ is a solution to Problem 3.1 if and only if there exists a nonnegative vector $\lambda^I$ and a vector $\lambda^E$ such that

$$\nabla f^t(\mathbf{x}^*) + \bar{A}^{I^t}\lambda^I + A^{E^t}\lambda^E = \mathbf{0}. \qquad (3.2)$$

As in the case of only equality constraints, this result is valid even when $\mathbf{x}^*$ is not a regular point [2, page 292].

Condition 3.2 is equivalent to

$$-\nabla f^t(\mathbf{x}^*) = \bar{A}^{I^t}\lambda^I + A^{E^t}\lambda^E, \quad \lambda^I \geq \mathbf{0}, \qquad (3.3)$$

which can be rephrased as

"$-\nabla f^t(\mathbf{x}^*)$ belongs to the cone $\mathrm{C}(A)$."

Let $H_\alpha$ be the unique matrix such that $H_\alpha\mathbf{x} = \mathbf{y}_\alpha$. Then, for fixed values of the $\alpha$-linking variables $\mathbf{y}_\alpha = \mathbf{d}_\alpha$, Problem $\alpha$ can be defined as

$$\begin{gathered}\mathrm{Min}_{\mathbf{x}}\ f(\mathbf{x}) \text{ subject to} \\ A^I\mathbf{x} \leq \mathbf{c}^I,\ A^E\mathbf{x} = \mathbf{c}^E \text{and } H_\alpha\mathbf{x} = \mathbf{d}_\alpha.\end{gathered} \qquad (3.4)$$

Based on the above reasoning, $\mathbf{x}^*_\alpha$ is a solution to Problem $\alpha$ if and only if

"$-\nabla f^t(\mathbf{x}^*_\alpha)$ belongs to the cone $\mathrm{C}(K_\alpha)$."

Analogously, $\mathbf{x}^*_\beta$ is a solution to Problem $\beta$ if and only if

$-\nabla f^t(\mathbf{x}^*_\beta)$ belongs to the cone $\mathrm{C}(K_\beta)$.

**Theorem 3.1** *Let* $\mathbf{x}^*$ *be an accumulation point of* $\{x_{\alpha_i}\}_{i=1}^{\infty}$ *or* $\{x_{\beta_i}\}_{i=1}^{\infty}$*. If*

$$\mathrm{C}(A) = \mathrm{C}(K_\alpha) \cap \mathrm{C}(K_\beta),$$

*then* $\mathbf{x}^*$ *is a solution to the optimization problem in (3.1).*

**Proof:** As explained in [10, Section 2.3], $\mathbf{x}^*$ solves both Problem $\alpha$ and Problem $\beta$. Therefore,

$$-\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(K_\alpha) \text{ and } -\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(K_\beta).$$

Since $\mathrm{C}(A) = \mathrm{C}(K_\alpha) \cap \mathrm{C}(K_\beta)$, one gets $-\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(A)$, which implies $\mathbf{x}^*$ is a solution to the original optimization problem in 3.1. □

The HOC convergence condition stated in Theorem 3.1 cannot be practically used because one has to know a priori the accumulation point $\mathbf{x}^*$ and the set $T_a$ of active constraints in order to compute the cones $\mathrm{C}(A)$, $\mathrm{C}(K_\alpha)$ and $\mathrm{C}(K_\beta)$.

Theorem 3.2 below fixes this problem and provides a new sufficient condition for the convergence of HOC. This condition does not rely on the accumulation point $\mathbf{x}^*$.

**Theorem 3.2** *Let $r$ be the rank of $\begin{pmatrix} A^I \\ A^E \end{pmatrix}$ and $\hat{A}$ be an $r \times n$ submatrix of $\begin{pmatrix} A^I \\ A^E \end{pmatrix}$ with full rank. If the matrix*

$$\hat{K}_{\alpha\beta} := \begin{pmatrix} \hat{A} \\ H_\alpha \\ H_\beta \end{pmatrix}$$

*has full rank, then $\mathrm{C}(A) = \mathrm{C}(K_\alpha) \cap \mathrm{C}(K_\beta)$.*

**Proof:** Clearly, $\mathrm{C}(A) \subset \mathrm{C}(K_\alpha)$ and $\mathrm{C}(A) \subset \mathrm{C}(K_\beta)$. Therefore, $\mathrm{C}(A) \subset \mathrm{C}(K_\alpha) \cap \mathrm{C}(K_\beta)$.

To show the reverse inclusion, choose an arbitrary $\mathbf{v} \in \mathrm{C}(K_\alpha) \cap \mathrm{C}(K_\beta)$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be the row vectors of $\hat{A}$, and $\mathbf{e}_i \in \mathbb{R}^n$ be the $i$-th standard row vector. Since $\mathbf{v} \in \mathrm{C}(K_\alpha)$ and $\mathbf{v} \in \mathrm{C}(K_\beta)$,

$$\begin{aligned} \mathbf{v} &= \sum_{i \in T_a} a_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)}, \quad a_i \geq 0, \\ &\text{and} \\ \mathbf{v} &= \sum_{i \in T_a} d_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} e_i \mathbf{v}_i^E + \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)}, \quad d_i \geq 0. \end{aligned}$$

Therefore,

$$\begin{aligned} &\sum_{i \in T_a} (a_i - d_i)\mathbf{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i)\mathbf{v}_i^E \\ &\qquad + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)} = \mathbf{0}. \qquad (3.5) \end{aligned}$$

Since $\mathbf{v}_1, \ldots, \mathbf{v}_r$ form a basis for the row space of $\begin{pmatrix} A^I \\ A^E \end{pmatrix}$, there exist $\gamma_1, \ldots, \gamma_r \in \mathbb{R}$ such that

$$\sum_{i \in T_a} (a_i - d_i)\mathbf{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i)\mathbf{v}_i^E = \sum_{i=1}^{r} \gamma_i \mathbf{v}_i.$$

Hence, (3.5) becomes

$$\sum_{i=1}^{r} \gamma_i \mathbf{v}_i + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)} = \mathbf{0}.$$

Since

$$\hat{K}_{\alpha\beta} = (\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{e}_{\alpha(1)}, \ldots, \mathbf{e}_{\alpha(n_\alpha)}, \mathbf{e}_{\beta(1)}, \ldots, \mathbf{e}_{\beta(n_\beta)})^t$$

has full rank, $\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{e}_{\alpha(1)}, \ldots, \mathbf{e}_{\alpha(n_\alpha)}, \mathbf{e}_{\beta(1)}, \ldots, \mathbf{e}_{\beta(n_\beta)}$ are linearly independent. Therefore,

$$\gamma_i = 0, \; s_j = 0, \; t_k = 0$$

for all $i = 1, \ldots, r$, $j = 1, \ldots, n_\alpha$, $k = 1, \ldots, n_\beta$, and thus

$$\mathbf{v} = \sum_{i \in T_a} a_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E, \quad a_i \geq 0.$$

This implies that $\mathbf{v} \in \mathrm{C}(A)$. □

Theorem 3.2 combined with Theorem 3.1 immediately implies the following Corollary.

**Corollary 3.3** *Same notations as in Theorem 3.2. For Problem 3.1, suppose $\alpha$- and $\beta$-decompositions are given. Let $\{\mathbf{x}_{\alpha_i}\}_{i=1}^\infty$ and $\{\mathbf{x}_{\beta_i}\}_{i=1}^\infty$ be two sequences obtained by applying HOC to these decompositions, and $\mathbf{x}^*$ be an accumulation point of $\{\mathbf{x}_{\alpha_i}\}_{i=1}^\infty$ or $\{\mathbf{x}_{\beta_i}\}_{i=1}^\infty$. If $\hat{K}_{\alpha\beta}$ has full rank, then $\mathbf{x}^*$ is a solution to the optimization problem in (3.1).*

## 4 Convergence Under Nonlinear Constraints

Consider a design problem that can be formulated as a convex optimization problem of the form

$$\mathrm{Min}_{\mathbf{x}}\, f(\mathbf{x}) \text{ subject to } \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0 \qquad (4.1)$$

where $X \subset \mathbb{R}^n$ is a nonempty open convex set, $f : X \to \mathbb{R}$ and $g_i : X \to \mathbb{R}$ are convex and differentiable functions on $X$, and $h_i : X \to \mathbb{R}$ are affine functions on $X$.

The Jacobian of the constraint functions in Problem 4.1 plays a role similar to that of the matrix $\begin{pmatrix} A^I \\ A^E \end{pmatrix}$ in the linear problem (3.1). Let $J(\mathbf{x})$ be the matrix $\begin{pmatrix} J^I(\mathbf{x}) \\ J^E(\mathbf{x}) \end{pmatrix}$ where $J^I(\mathbf{x})$ and $J^E(\mathbf{x})$ are the Jacobians of $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$, respectively. This matrix function $J(\mathbf{x})$ will be simply referred to as the Jacobian of the Problem 4.1. Define the matrices $K_\alpha(\mathbf{x})$, $K_\beta(\mathbf{x})$ and $K_{\alpha\beta}(\mathbf{x})$ by

$$K_\alpha(\mathbf{x}) := \begin{pmatrix} J(\mathbf{x}) \\ H_\alpha \end{pmatrix}, \; K_\beta(\mathbf{x}) := \begin{pmatrix} J(\mathbf{x}) \\ H_\beta \end{pmatrix}, \; K_{\alpha\beta}(\mathbf{x}) := \begin{pmatrix} J(\mathbf{x}) \\ H_\alpha \\ H_\beta \end{pmatrix}.$$

For a fixed point $\mathbf{p} \in \mathbb{R}^n$, define $T_a(\mathbf{p})$ to be the set of the indices corresponding to the active inequality constraints at $\mathbf{p}$, i.e.,

$$T_a(\mathbf{p}) := \{i \mid g_i(\mathbf{p}) = 0\},$$

where $g_i$ denotes the $i$-th inequality constraint. Let $\bar{J}^I(\mathbf{p})$ be the submatrix of $J^I(\mathbf{p})$ consisting of the active inequality constraints at $\mathbf{p}$.

The cones $\mathrm{C}(J)(\mathbf{p})$, $\mathrm{C}(K_\alpha)(\mathbf{p})$ and $\mathrm{C}(K_\beta)(\mathbf{p})$ can be defined analogously as in the linear case. Define the cone $\mathrm{C}(J)(\mathbf{p})$ by

$$\mathrm{C}(J)(\mathbf{p}) := \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \sum_{i \in T_a(\mathbf{p})} a_i \mathbf{v}_i^I(\mathbf{p}) + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E(\mathbf{p}), a_i \geq 0\},$$

where $\mathbf{v}_i^I(\mathbf{p})$ ($\mathbf{v}_i^E(\mathbf{p})$, resp.) denotes the $i$-th row vector of $J^I(\mathbf{p})$ ($J^E(\mathbf{p})$, resp.). Also, define the induced cones $\mathrm{C}(K_\alpha)(\mathbf{p})$ and $\mathrm{C}(K_\beta)(\mathbf{p})$ as follows:

$$\mathrm{C}(K_\alpha)(\mathbf{p}) := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \sum_{i \in T_a(\mathbf{p})} a_i \mathbf{v}_i^I(\mathbf{p}) + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E(\mathbf{p}) + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)}, a_i \geq 0\},$$

$$\mathrm{C}(K_\beta)(\mathbf{p}) := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \sum_{i \in T_a(\mathbf{p})} a_i \mathbf{v}_i^I(\mathbf{p}) + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E(\mathbf{p}) + \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)}, a_i \geq 0\}.$$

The Generic HOC Algorithm described in Section 2 applied to Problem 4.1 results in two sequences $\{\mathbf{x}_{\alpha_i}\}_{i=1}^\infty$ and $\{\mathbf{x}_{\beta_i}\}_{i=1}^\infty$. The Lagrange multiplier theorem for nonlinear constraints [2] states that a regular point $\mathbf{x}^* \in \mathbb{R}^n$ is a solution to Problem 4.1 if and only if there exists a nonnegative vector $\lambda^I$ and a vector $\lambda^E$ such that

$$\nabla f^t(\mathbf{x}^*) + \bar{J}^I(\mathbf{x}^*)^t \lambda^I + J^E(\mathbf{x}^*)^t \lambda^E = \mathbf{0}. \qquad (4.2)$$

Condition 4.2 is equivalent to

$$-\nabla f^t(\mathbf{x}^*) = \bar{J}^I(\mathbf{x}^*)^t \lambda^I + J^E(\mathbf{x}^*)^t \lambda^E, \; \lambda^I \geq \mathbf{0}, \quad (4.3)$$

which can be rephrased as

"$-\nabla f^t(\mathbf{x}^*)$ belongs to the cone $\mathrm{C}(J(\mathbf{x}^*))$."

For fixed values of the $\alpha$-linking variables $\mathbf{y}_\alpha = \mathbf{d}_\alpha$, Problem $\alpha$ can be defined as

$$\mathrm{Min}_{\mathbf{x}}\, f(\mathbf{x}) \text{ subject to}$$
$$\mathbf{h}(\mathbf{x}) = 0,\; \mathbf{g}(\mathbf{x}) \leq 0, \text{ and } H_\alpha \mathbf{x} = \mathbf{d}_\alpha. \qquad (4.4)$$

Based on the above reasoning, $\mathbf{x}_\alpha^*$ is a solution to Problem $\alpha$ if and only if

"$-\nabla f^t(\mathbf{x}_\alpha^*)$ belongs to the cone $\mathrm{C}(K_\alpha)(\mathbf{x}_\alpha^*)$."

Analogously, $\mathbf{x}_\beta^*$ is a solution to Problem $\beta$ if and only if

$-\nabla f^t(\mathbf{x}_\beta^*)$ belongs to the cone $\mathrm{C}(K_\beta)(\mathbf{x}_\beta^*)$.

Now the following theorem is immediate.

**Theorem 4.1** *Let $\mathbf{x}^*$ be an accumulation point of $\{x_{\alpha_i}\}_{i=1}^\infty$ or $\{x_{\beta_i}\}_{i=1}^\infty$. If $\mathbf{x}^*$ is a regular point and*

$$\mathrm{C}(J)(\mathbf{x}^*) = \mathrm{C}(K_\alpha)(\mathbf{x}^*) \cap \mathrm{C}(K_\beta)(\mathbf{x}^*),$$

*then $\mathbf{x}^*$ is a solution to the optimization problem in (4.1).*

**Proof:** As in the linear case, $\mathbf{x}^*$ solves both Problem $\alpha$ and Problem $\beta$. Therefore,

$$-\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(K_\alpha)(\mathbf{x}^*) \text{ and } -\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(K_\beta)(\mathbf{x}^*).$$

Since $\mathrm{C}(J)(\mathbf{x}^*) = \mathrm{C}(K_\alpha)(\mathbf{x}^*) \cap \mathrm{C}(K_\beta)(\mathbf{x}^*)$, one gets $-\nabla f^t(\mathbf{x}^*) \in \mathrm{C}(J)(\mathbf{x}^*)$, which implies $\mathbf{x}^*$ is a solution to the original optimization problem in (4.1). □

**Theorem 4.2** *Same notations as in Theorem 4.1. Let* $r$ *be the rank of* $J(\mathbf{x}^*) = \begin{pmatrix} J^I(\mathbf{x}^*) \\ J^E(\mathbf{x}^*) \end{pmatrix}$ *and* $\hat{J}(\mathbf{x}^*)$ *be an* $r \times n$ *submatrix of* $J(\mathbf{x}^*)$ *with full rank. If the matrix*

$$\hat{K}_{\alpha\beta}(\mathbf{x}^*) := \begin{pmatrix} \hat{J}(\mathbf{x}^*) \\ H_\alpha \\ H_\beta \end{pmatrix}$$

*has full rank, then* $\mathrm{C}(J)(\mathbf{x}^*) = \mathrm{C}(K_\alpha)(\mathbf{x}^*) \cap \mathrm{C}(K_\beta)(\mathbf{x}^*)$.

**Proof:** Clearly, $\mathrm{C}(J)(\mathbf{x}^*) \subset \mathrm{C}(K_\alpha)(\mathbf{x}^*)$ and $\mathrm{C}(J)(\mathbf{x}^*) \subset \mathrm{C}(K_\beta)(\mathbf{x}^*)$. Therefore,

$$\mathrm{C}(J)(\mathbf{x}^*) \subset \mathrm{C}(K_\alpha)(\mathbf{x}^*) \cap \mathrm{C}(K_\beta)(\mathbf{x}^*).$$

To show the reverse inclusion, choose an arbitrary $\mathbf{v} \in \mathrm{C}(K_\alpha)(\mathbf{x}^*) \cap \mathrm{C}(K_\beta)(\mathbf{x}^*)$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be the row vectors of $\hat{J}(\mathbf{x}^*)$, and $\mathbf{e}_i \in \mathbb{R}^n$ be the $i$-th standard row vector. Since $\mathbf{v} \in \mathrm{C}(K_\alpha)(\mathbf{x}^*)$ and $\mathbf{v} \in \mathrm{C}(K_\beta)(\mathbf{x}^*)$,

$$\begin{aligned} \mathbf{v} &= \sum_{i \in T_a} a_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)}, \quad a_i \geq 0, \\ &\text{and} \\ \mathbf{v} &= \sum_{i \in T_a} d_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} e_i \mathbf{v}_i^E + \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)}, \quad d_i \geq 0, \end{aligned}$$

Therefore,

$$\sum_{i \in T_a} (a_i - d_i)\mathbf{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i)\mathbf{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)} = \mathbf{0}. \qquad (4.5)$$

Since $\mathbf{v}_1, \ldots, \mathbf{v}_r$ form a basis for the row space of $\hat{J}(\mathbf{x}^*)$, there exist $\gamma_1, \ldots, \gamma_r \in \mathbb{R}$ such that

$$\sum_{i \in T_a} (a_i - d_i)\mathbf{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i)\mathbf{v}_i^E = \sum_{i=1}^{r} \gamma_i \mathbf{v}_i.$$

Hence, (4.5) becomes

$$\sum_{i=1}^{r} \gamma_i \mathbf{v}_i + \sum_{i=1}^{n_\alpha} s_i \mathbf{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \mathbf{e}_{\beta(i)} = \mathbf{0}.$$

Since the matrix

$$\hat{K}_{\alpha\beta}(\mathbf{x}^*) = (\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{e}_{\alpha(1)}, \ldots, \mathbf{e}_{\alpha(n_\alpha)}, \mathbf{e}_{\beta(1)}, \ldots, \mathbf{e}_{\beta(n_\beta)})^t$$

has full rank, its row vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$, $\mathbf{e}_{\alpha(1)}, \ldots, \mathbf{e}_{\alpha(n_\alpha)}$, and $\mathbf{e}_{\beta(1)}, \ldots, \mathbf{e}_{\beta(n_\beta)}$ are linearly independent. Therefore,

$$\gamma_i = 0, \; s_j = 0, \; t_k = 0$$

for all $i = 1, \ldots, r$, $j = 1, \ldots, n_\alpha$, $k = 1, \ldots, n_\beta$, and thus

$$\mathbf{v} = \sum_{i \in T_a} a_i \mathbf{v}_i^I + \sum_{i=1}^{m_E} b_i \mathbf{v}_i^E, \quad a_i \geq 0.$$

This implies that $\mathbf{v} \in \mathrm{C}(A)$. □

Theorem 4.2 combined with Theorem 4.1 immediately implies the following Corollary.

**Corollary 4.3** *Same notations as in Theorem 4.2. For Problem 4.1, suppose* $\alpha$*- and* $\beta$*-decompositions are given. Let* $\{\mathbf{x}_{\alpha_i}\}_{i=1}^{\infty}$ *and* $\{\mathbf{x}_{\beta_i}\}_{i=1}^{\infty}$ *be two sequences obtained by applying HOC to these decompositions, and* $\mathbf{x}^*$ *be an accumulation point of* $\{\mathbf{x}_{\alpha_i}\}_{i=1}^{\infty}$ *or* $\{\mathbf{x}_{\beta_i}\}_{i=1}^{\infty}$*. If* $\mathbf{x}^*$ *is a regular point and* $\hat{K}_{\alpha\beta}(\mathbf{x}^*)$ *has full rank, then* $\mathbf{x}^*$ *is a solution to the optimization problem in (4.1).*

**Remark 4.4** $\hat{K}_{\alpha\beta}(\mathbf{x}^*)$ has full rank only if the sets of $\alpha$- and $\beta$-linking variables are disjoint.

### HOC Algorithm for Nonlinear Convex Problems

Step 1. Apply the Generic HOC Algorithm starting at a point $\mathbf{x}_0$ for $\alpha$- and $\beta$-decompositions that makes $\hat{J}(\mathbf{x}_0)$ full rank. Let $\mathbf{x}^*$ be a resulting accumulation point.

Step 2. If $\hat{J}(\mathbf{x}^*)$ has full rank, then (by Corollary 4.3) conclude that $\mathbf{x}^*$ is a solution to the original optimization problem in (4.1) and exit.

Step 3. If $\hat{J}(\mathbf{x}^*)$ fails to have full rank, then find new $\alpha$- and $\beta$-decompositions that make $\hat{J}(\mathbf{x}^*)$ full rank, and go to Step 1 with $\mathbf{x}^*$ as new starting point. If it is not possible to find appropriate $\alpha$- and $\beta$-decompositions, then assume another starting point and go to Step 1 or exit.

This process is repeated until we reach a point $\mathbf{x}^\dagger$ such that $\hat{J}(\mathbf{x}^\dagger)$ has full rank (or a maximum number of iterations). Then this point $\mathbf{x}^\dagger$ is, by Corollary 4.3, a solution to the original optimization problem in (4.1). Valid decompositions are generated using hypergraph-based model decomposition techniques [8, 10] described in Section 5.1.

**Remark 4.5** The HOC convergence condition stated in Corollary 4.3 requires that one has to know the accumulation point $\mathbf{x}^*$ in order to compute $\hat{K}_{\alpha\beta}(\mathbf{x}^*)$. However, if the matrix function

$$\hat{K}_{\alpha\beta}(\mathbf{x}) = \begin{pmatrix} \hat{f}(\mathbf{x}) \\ H_\alpha \\ H_\beta \end{pmatrix}$$

has full rank for every $\mathbf{x}$ in the feasible domain, then $\hat{K}_{\alpha\beta}(\mathbf{x}^*)$ also has full rank, and HOC is convergent. The method used in the proof [10, Theorem 3.5] can be used to show that $\hat{K}_{\alpha\beta}(\mathbf{x})$ has full rank everywhere if and only if there is no functional dependency exclusively among the α- and β-linking variables. This situation can be described in terms of *Elimination Theory*: In the case of polynomial constraints, let $\mathbb{R}[x_1,\ldots,x_n]$ be the ring of polynomials in $n$ variables, and let $I$ be the ideal generated by the given constraints. Then the projection of $I$ onto the subring $\mathbb{R}[x_{\alpha(1)},\ldots,x_{\alpha(n_\alpha)},x_{\beta(1)},\ldots,x_{\beta(n_\beta)}]$ consisting of polynomials in α- and β-linking variables, should be empty. The same argument can be made for the case of analytic constraints.

In the case of polynomial constraints, it can be computationally checked whether $\hat{K}_{\alpha\beta}(\mathbf{x})$ has full rank everywhere. Some symbolic computer algebra systems, e.g. Macaulay [1] and Singular [4], are capable of computing the projection of an ideal onto a subring.

## 5 Computational Results

### 5.1 Obtaining Two Distinct Decompositions

Recall that the functional dependence table (FDT) is a Boolean matrix representing the dependence of design functions on variables. The $(i,j)$-th entry of the FDT is one if the $i$-th function depends on the $j$-th variable and zero otherwise. A decomposition of the given optimization problem can be achieved by reordering rows and columns of the FDT corresponding to the constraints and variables, respectively.

The sufficient condition for convergence of HOC cannot easily be (and does not need to be) enforced at every point of the feasible space. As explained before, the condition will be enforced at the initial point and verified at the accumulation point(s). To solve a nonlinear optimization problem $P$ by the HOC algorithm, two distinct (α-, β-) decompositions of $P$ satisfying the *sufficient condition for convergence of HOC at the initial* point $\mathbf{x}_0$ (Corollary 4.3) can be found by the following heuristic:

1. Apply the hypergraph-based model decomposition algorithm developed in [8][2] to problem $P$ to obtain an α-decomposition.
2. In the process of obtaining a β-decomposition, penalize[3] the α-linking variables so that the disjointness of the set of α-linking variables and the set of β-linking variables is accomplished, as required by the convergence condition in Corollary 4.3. If the two sets of linking variables are not disjoint, then go back to Step 1 and obtain a new α-decomposition after penalizing the common linking variables.
3. Check if the resulting α- and β-decompositions satisfy the convergence condition in Corollary 4.3. If the convergence condition is not satisfied, then go back to Step 1 and obtain a new α-decomposition after penalizing one of the interdependent linking variables.

### 5.2 Example Problems

The HOC algorithm developed in this article has been applied to a family of nonlinear optimization problems of various sizes. The smallest problem $P_1$ has 25 variables and 21 constraints (19 linear equalities and 2 nonlinear inequalities) with a strictly convex, additive separable objective function. (Thus, the FDT of problem $P_1$ constraints is a $21 \times 25$ table.) The largest problem $P_9$ has 500 variables and 420 constraints (380 linear equalities and 40 nonlinear inequalities).

Figure 2 shows the reordered FDTs for the α- and β-decompositions obtained by applying the above decomposition heuristic to example problem $P_1$. *Maple* [9] was used to verify that these two decompositions do satisfy the convergence condition in Corollary 4.3 for the initial point $\mathbf{x}_0$. The α-decomposition in Figure 2 has two subproblems and one linking variable ($x_{13}$), whereas the β-decomposition has two subproblems and two linking variables ($x_3$ and $x_9$). While each of the two decompositions for $P_1$ has two subproblems, each of the two decompositions for $P_9$ has 40 subproblems.

Once the α- and β-decompositions are determined, the subproblems have to be repeatedly solved. *constr*, the *MATLAB* [7] implementation of the Sequential Quadratic Programming algorithm (SQP), was used for this purpose. The HOC iteration process stops if the relative difference between the values of the objective function

[2] An implementation of this decomposition algorithm can be found at the URL: http://arc.engin.umich.edu/graph_part.html.

[3] A variable is penalized when it is not desirable to have the variable as a linking variable. This can be achieved by assigning a high weight to the corresponding hyperedge in the hypergraph-based model decomposition algorithm described in [8].

| | | | | All-At-Once | | Hierarchical Overlapping Coordination | | | |
|---|---|---|---|---|---|---|---|---|---|
| prob | no. | no. | no. | objective | runtime† | objective | serial- | parallel- | no. HOC |
| | var | constr | subpr | | | | runtime† | runtime† | iterations |
| $P_1$ | 25 | 21 | 2 | 3.34092 | 1.847 | 3.34092 | 3.045 | 2.820 | 3 |
| $P_2$ | 50 | 42 | 4 | 6.68185 | 6.067 | 6.68185 | 5.997 | 2.790 | 3 |
| $P_3$ | 75 | 63 | 6 | 10.0227 | 16.85 | 10.0227 | 9.032 | 2.800 | 3 |
| $P_4$ | 100 | 84 | 8 | 13.3637 | 29.59 | 13.3637 | 12.03 | 2.812 | 3 |
| $P_5$ | 125 | 105 | 10 | 16.7046 | 59.03 | 16.7046 | 15.07 | 2.827 | 3 |
| $P_6$ | 200 | 168 | 16 | 26.7274 | 333 | 26.7274 | 24.13 | 2.832 | 3 |
| $P_7$ | 250 | 210 | 20 | 33.4092 | 1446 | 33.4092 | 30.18 | 2.835 | 3 |
| $P_8$ | 375 | 315 | 30 | 50.1139 | 6904 | 50.1139 | 45.22 | 2.832 | 3 |
| $P_9$ | 500 | 420 | 40 | 66.8185 | 20539 | 66.8185 | 60.77 | 2.890 | 3 |

†Runtime is measured in CPU seconds on a Sun UltraSparc 1.

Table 1. Comparison of CPU-runtimes for nonlinear optimization problems of various sizes. Initial point $\mathbf{x}_0 = -\mathbf{0.1}$.

| | | | | All-At-Once | | Hierarchical Overlapping Coordination | | | |
|---|---|---|---|---|---|---|---|---|---|
| prob | no. | no. | no. | objective | runtime† | objective | serial- | parallel- | no. HOC |
| | var | constr | subpr | | | | runtime† | runtime† | iterations |
| $P_1$ | 25 | 21 | 2 | 3.34092 | 2.860 | 3.34092 | 1.942 | 1.867 | 1 |
| $P_2$ | 50 | 42 | 4 | 6.68185 | 11.41 | 6.68185 | 3.867 | 1.877 | 1 |
| $P_3$ | 75 | 63 | 6 | 10.0227 | 25.57 | 10.0227 | 5.782 | 1.870 | 1 |
| $P_4$ | 100 | 84 | 8 | 13.3637 | 52.46 | 13.3637 | 7.697 | 1.867 | 1 |
| $P_5$ | 125 | 105 | 10 | 16.7046 | 96.85 | 16.7046 | 9.662 | 1.872 | 1 |
| $P_6$ | 200 | 168 | 16 | 26.7274 | 818 | 26.7274 | 15.47 | 1.887 | 1 |
| $P_7$ | 250 | 210 | 20 | 33.4092 | 1701 | 33.4092 | 19.32 | 1.885 | 1 |
| $P_8$ | 375 | 315 | 30 | 50.1139 | 10869 | 50.1139 | 28.85 | 1.877 | 1 |
| $P_9$ | 500 | 420 | 40 | 66.8187 | 27662 | 66.8185 | 38.99 | 1.910 | 1 |

†Runtime is measured in CPU seconds on a Sun UltraSparc 1.

Table 2. Comparison of CPU-runtimes for nonlinear optimization problems of various sizes. Initial point $\mathbf{x}_0 = \mathbf{0}$.

for two consecutive iterations is less than a preset tolerance value. The tolerance value used for the computation was $10^{-5}$.

To compare the effectiveness of the HOC algorithm with the ordinary All-At-Once (AAO) algorithm (i.e. one not using decompositions), the problems were solved in both ways. The results for $P_1$ and the other problems of larger sizes are shown in Tables 1 and 2 for two different initial points.

Runtime was measured in CPU seconds on a Sun UltraSparc 1. Runtimes only include *constr* function calls, excluding I/O and data transfer between $\alpha$- and $\beta$-decompositions. Each runtime represents the average runtime for five separate runs of the algorithm; the times of the five runs were consistently close. Serial-runtime is measured for the HOC computation in which the subproblems are solved sequentially, whereas parallel-runtime is measured for the HOC computation in which the subproblems are simulated to be solved in parallel.

## 5.3 Discussion

HOC has shorter parallel and serial runtimes than AAO algorithm for all problems and initial points except for problem $P_1$ with initial point $\mathbf{x}_0 = -\mathbf{0.1} := -(0.1, 0.1, \ldots, 0.1)$. Moreover, for all problem sizes and initial points, the HOC algorithm did not require repartition of the model as dictated by Step 3 of the algorithm. That is, the condition for convergence was satisfied both at the initial points and at the first-generated accumulation points. The AAO algorithm converges faster when it is started with $\mathbf{x}_0 = -\mathbf{0.1}$ as initial point, whereas the HOC algorithm converges faster when it is started with $\mathbf{x}_0 = \mathbf{0}$ as initial point.

Note that HOC for $P_9$ has parallel- and serial-runtimes that are $7100 - 14500$ and $340 - 710$ times shorter than the AAO runtimes, respectively, depending

```
          DESIGN  VARIABLES

        1000000000111111111222222
        3123456789012456789012345

     1  .%%%%%...................
     2  ...%..%..................
D    3  .%.....%%................
E    4  ...%.....%%..............
S    5  .......%...%.............
I    6  ..........%.%............
G    7  *%%%.....................
N    8  *%.........%.%...........
     9  .......%......%..........
    10  ..........%....%.........
R   11  .%.....%......%.%........
E   12  ...%......%....%.%.......
L   20  .....%..%%...............
A   21  ...%........%............
T   13  *.................@......
I   14  ..................@@.....
O   15  *...................@....
N   16  ...................@@@...
S   17  ..................@...@..
    18  ..................@....@.
    19  ..................@.....@

               (a)
```

```
          DESIGN  VARIABLES

        0001111000000111111222222
        3960268124578134579012345

     2  *.%......................
     4  **.%.....................
     6  ...%%....................
    10  ...%.%...................
    12  *..%.%%..................
    21  *...%....................
     1  *......@@@@..............
     3  .......@...@@............
     5  ...........@.@...........
     7  *......@@.....@..........
     8  .......@.....@@@.........
     9  ...........@....@........
    11  .......@...@....@@.......
    13  ..............@...@......
    14  ..................@@.....
    15  ..............@.....@....
    16  ...................@@@...
    17  ..................@...@..
    18  ..................@....@.
    19  ..................@.....@
    20  .*........@.@............

               (b)
```

Figure 2. Decompositions of example Problem $P_1$: (a) α-decomposition and (b) β-decomposition

on the initial point. Surprisingly, HOC terminates after three and one iterations regardless of problem size. HOC serial-runtimes vary linearly with the size of the problem, whereas parallel-runtimes remain about constant. This is expected since subproblem sizes are similar for all nine problems. AAO runtimes seem to vary polynomially ($3^{rd}$ or $4^{th}$ order) with the size of the problem. HOC serial wall-clock times, which include data transfer between α- and β-decompositions, were significantly shorter than their AAO counterpart for the larger problems. For example, $P_9$ was solved in less than 2 minutes by HOC, while its AAO solution took over $7\frac{1}{2}$ hours on the same workstation. This clearly demonstrates the advantage of using HOC for large scale problems with many loosely-linked subproblems.

## Acknowledgment

This research was supported by the U.S. Army Automotive Research Center for Modeling and Simulation of Ground Vehicle under contract No. DAAE07-94-C-R094, and by the Oakland University Foundation. This support is gratefully acknowledged.

## References

[1] D. Bayer and M. Stillman. *Macaulay: A computer algebra system for algebraic geometry*, 1995. Available by anonymous ftp from ftp.math.harvard.edu.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.

[3] R. D. Braun and I. M. Kroo. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In N. Alexandrov and M. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, pages 98–116. SIAM, 1997.

[4] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular Reference Manual. In *Reports On Computer Algebra*, number 12. Centre for Computer Algebra, University of Kaiserslautern, May 1997. http://www.mathematik.uni-kl.de/~zca/Singular.

[5] S. A. Nelson II and P. Y. Papalambros. Sequentially decomposed programming. *AIAA Journal*, 35(7):1209–1216, July 1997.

[6] D. Macko and Y. Haimes. Overlapping coordination of hierarchical structures. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8:745–751, 1978.

[7] The Math Works, Inc., Natick. MA. *Using MATLAB*, version 5.1 edition, 1997.

[8] N. Michelena and P. Papalambros. A hypergraph framework to optimal model-based decomposition of design problems. *Computational Optimization and Applications*, 8(2):173–196, September 1997.

[9] R. Nicolaides and N. Walkington. *Maple: A Comprehensive Introduction*. Cambridge University Press, New York, 1996.

[10] H. Park, N. Michelena, D. Kulkarni, and P. Papalambros. Convergence criteria for hierarchical overlapping coordination under linear constraints. *Computational Optimization and Applications*, 1998. (submitted).

[11] T. Shima and Y. Haimes. The convergence properties of hierarchical overlapping coordination. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14:74–87, February 1984.

[12] J. Sobieszczanski-Sobieski. Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In *Presented at the Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA, September 1988. NASA CP-3031, Part 1. Also NASA TM-101494.

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

[14] T. Wagner and P. Papalambros. A general framework for decomposition analysis in optimal design. In B.J. Gilmore, editor, *Advances in Design Automation*, volume 2, pages 315–322, New York, 1993. ASME.