AIAA JOURNAL
Vol. 37, No. 7, July 1999

Hierarchical Overlapping Coordination for Large-Scale Optimization by Decomposition

Nestor Michelena* and Panos Papalambros†
University of Michigan, Ann Arbor, Michigan 48109-2125
and
Hyungju A. Park‡ and Devadatta Kulkarni§
Oakland University, Rochester, Michigan 48309-4485

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. We present the HOC algorithm and a sufficient condition for global convergence of the algorithm to the solution of a convex optimization problem. 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 problems of various sizes are also presented.

Nomenclature

$A := \begin{bmatrix} A^I \\ A^E \end{bmatrix}$ = matrix of liner inequality and equality constraints
A^E, A^I = constraint matrices corresponding to linear equalities, inequalities
$\bar{A}^I$ = submatrix of A^I consisting of active inequality constraints
A_{α_i} = block of functional dependence table associated with the ith subproblem of the α decomposition
$\hat{A}$ = full row rank submatrix of A
$C(A)$ = cone defined by the row vectors of A^E and $\bar{A}^I$
$\boldsymbol{c}^E, \boldsymbol{c}^I$ = constant vectors for linear equality, inequality constraints
$f(\boldsymbol{x})$ = objective function
f_{α_i} = summand in objective function involving $\boldsymbol{x}_{\alpha_i}$
f_{α_0} = summand in objective function involving only linking variables
$\boldsymbol{g}(\boldsymbol{x})$ = vector of inequality constraint functions
H_α = indicator matrix such that $H_\alpha \boldsymbol{x} = \boldsymbol{y}_\alpha$
$\boldsymbol{h}(\boldsymbol{x})$ = vector of equality constraint (affine) functions
$\boldsymbol{h}_{\alpha_i}, \boldsymbol{g}_{\alpha_i}$ = vector of constraint functions involving $\boldsymbol{x}_{\alpha_i}$
$\bar{J}^I(\boldsymbol{p})$ = submatrix of $J^I(\boldsymbol{p})$ consisting of the active inequality constraints at $\boldsymbol{p}$
$J(\boldsymbol{x}) := \begin{bmatrix} J^I(\boldsymbol{x}) \\ J^E(\boldsymbol{x}) \end{bmatrix}$ = Jacobian of inequality and equality constraint functions
$\hat{J}(\boldsymbol{x})$ = full row rank submatrix of $J(\boldsymbol{x})$
K_α = matrix obtained by augmenting A with H_α
$K_\alpha(\boldsymbol{x})$ = matrix obtained by augmenting $J(\boldsymbol{x})$ with H_α
$\hat{K}_{\alpha\beta}$ = matrix obtained by augmenting $\hat{A}$ with H_α and H_β
$\hat{K}_{\alpha\beta}(\boldsymbol{p})$ = matrix obtained by augmenting $\hat{J}(\boldsymbol{p})$ with H_α and H_β
n_α = number of linking variables for the α decomposition
p_α = number of subproblems in the α decomposition
$\mathbb{R}$ = set of real numbers
$\mathbb{R}^n$ = n-dimensional Euclidean space
$T_a(T_a(\boldsymbol{p}))$ = set of indices corresponding to active inequality constraints (at $\boldsymbol{p}$)
X = set constraint
$\boldsymbol{x}$ = vector of optimization variables
$\boldsymbol{x}_{\alpha_i}$ = vector of local variables associated with the ith subproblem of the α decomposition
$\boldsymbol{y}_\alpha$ = vector of linking variables for the α decomposition
$\boldsymbol{\lambda}^E, \boldsymbol{\lambda}^I$ = Lagrange multipliers for equality, inequality constraints

I. 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. The resulting optimization problem may be, in general, a mixed-discrete nonlinear problem whose model contains noncontinuous mathematical functions. In this paper, we address design problems that can be formulated (or reformulated) as smooth, nonlinear optimization problems and solved for a local solution using a gradient-based optimization algorithm. One expects that a local solution provides a reasonable improvement over a baseline design or is sufficiently close to the global solution because of engineering constraints on the feasible region. For the mathematical proofs we consider a convex optimization problem of the form

find $\boldsymbol{x} \in X$ such that

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0},\ \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \text{ and } f(\boldsymbol{x}) \text{ is minimized} \tag{1}$$

Received 23 September 1998; revision received 9 December 1998; accepted for publication 20 January 1999.

*Assistant Research Scientist, Department of Mechanical Engineering and Applied Mechanics; nestorm@umich.edu. Member AIAA.
†Professor, Department of Mechanical Engineering and Applied Mechanics; pyp@umich.edu. Member AIAA.
‡Assistant Professor, Department of Mathematics and Statistics; park@oakland.edu.
§Associate Professor, Department of Mathematics and Statistics; kulkarni @oakland.edu.

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. This problem is the foundation for most nonlinear programming solution algorithms.

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 involve 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 affect one or more explicit problem decompositions. Thus, the 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, matrix-type organizations structured according to product lines or subsystems and the disciplines involved in the design process. At the product level, for example, a powertrain system design model could be partitioned by subsystems (e.g., engine, torque converter, transmission, differential, wheel, and vehicle submodels) or by aspects (e.g., thermodynamics, combustion, heat transfer, multibody, and fluid dynamics submodels). Coordination of the design subproblems is achieved by the exchange of information between decompositions.

The mathematical formulation of HOC was first proposed by Macko and Haimes,[1] and several criteria for convergence of the coordination algorithm under linear equality and inequality constraints were developed in Refs. 1 and 2. Convergence criteria developed in those articles are computationally difficult to check and possibly incorrect (see Remark 4.4 in Ref. 3). To remedy the situation, new computationally efficient criteria for convergence of the HOC algorithm under linear constraints were developed in Ref. 3. In the present paper, we present an HOC algorithm and a condition that ensures the convergence of the algorithm for nonlinear convex problems. This condition also guarantees convergence to a stationary point in the case of more general smooth, nonlinear optimization problems.

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[4] and Sobieszczanski-Sobieski and Haftka.[5] Recently, Nelson and Papalambros[6] presented Sequentially Decomposed Programming as a globally convergent coordination scheme for hierarchic systems. Other promising coordination algorithms, including concurrent subspace optimization (CSSO)[7] and collaborative optimization (CO)[8] for nonhierarchic systems, require further in-depth study of their robustness and convergence properties.

II. 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 ith function depends on the jth variable, and zero otherwise.

Hypergraph-based model decomposition[9] can be applied to the constraint functions of Problem (1) to obtain two or more distinct (e.g., $\alpha, \beta, \gamma, \ldots$) decompositions. This involves generating a decomposition of the FDT by reordering the variables and constraints, as shown in Fig. 1.

In Fig. 1, $\boldsymbol{x}_{\alpha_i}$ is the vector of local variables associated with block A_{α_i}, i.e., with Subproblem α_i; $\boldsymbol{y}_\alpha$ is the vector of n_α linking variables for the α decomposition; and p_α denotes the number of subproblems in the α decomposition (diagonal blocks in Fig. 1).

Provided that the objective function f is α-additive separable (in general, HOC can be used if the objective function can be written as

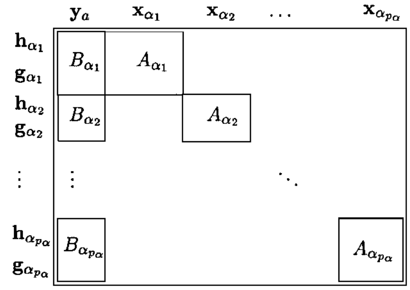

Fig. 1 Block α decomposition of functional dependence table.

monotonic functions of local objective functions derived from the α and β decompositions), Problem (1) takes the following form:

$$\underset{x}{\text{Min}}\ f_{\alpha_0}(\boldsymbol{y}_\alpha) + \sum_{i=1}^{p_\alpha} f_{\alpha_i}(\boldsymbol{y}_\alpha, \boldsymbol{x}_{\alpha_i}) \text{ subject to}$$

$$\boldsymbol{h}_{\alpha_i}(\boldsymbol{y}_\alpha, \boldsymbol{x}_{\alpha_i}) = \boldsymbol{0}, \qquad \boldsymbol{g}_{\alpha_i}(\boldsymbol{y}_\alpha, \boldsymbol{x}_{\alpha_i}) \le \boldsymbol{0}, \qquad i = 1, \ldots, p_\alpha \tag{2}$$

For a given vector $\boldsymbol{d}_\alpha \in \mathbb{R}^{n_\alpha}$, fixing the α-linking variables $\boldsymbol{y}_\alpha = \boldsymbol{d}_\alpha$ in Eq. (2) results in the following Problem α:

Problem α:

For each $i = 1, \ldots, p_\alpha$

$$\underset{x_{\alpha_i}}{\text{Min}}\ f_{\alpha_i}(\boldsymbol{d}_\alpha, \boldsymbol{x}_{\alpha_i}) \text{ subject to}$$

$$\boldsymbol{h}_{\alpha_i}(\boldsymbol{d}_\alpha, \boldsymbol{x}_{\alpha_i}) = \boldsymbol{0}, \qquad \boldsymbol{g}_{\alpha_i}(\boldsymbol{d}_\alpha, \boldsymbol{x}_{\alpha_i}) \le \boldsymbol{0} \tag{3}$$

Problem α can be solved by solving p_α 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).

A. Generic HOC Algorithm

Step 1. Fix linking variables $\boldsymbol{y}_\alpha$, and solve Problem α by solving the p_α independent subproblems given in Eq. (3).

Step 2. Fix linking variables $\boldsymbol{y}_\beta$ to their values determined in step 1, and solve Problem β by solving p_β 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.

Thus, in the HOC algorithm the linking variables for one of the decompositions are fixed at values that result from the solution of a number of independent subproblems associated with the previous decomposition.

B. Properties of HOC

The following properties of HOC were observed in Ref. 1 and can be proved in a straightforward manner:

1) If the HOC algorithm is started with a feasible point $\boldsymbol{x}_0$, then at each stage of the process Problem α and Problem β will have nonempty feasible domains.

2) If the sequences $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ and $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$ result from solving Problem α and Problem β, respectively, and $f^{\min} := \min\{f(\boldsymbol{x}) \mid \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}\}$, then

a) $f(\boldsymbol{x}_{\alpha_i}) \ge f(\boldsymbol{x}_{\beta_i}) \ge f(\boldsymbol{x}_{\alpha_{(i+1)}})$

b) $\lim_{i\to\infty} f(\boldsymbol{x}_{\alpha_i}) = \lim_{i\to\infty} f(\boldsymbol{x}_{\beta_i}) = f^* \ge f^{\min}$

3) Any accumulation point $\boldsymbol{x}^*$ of either $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ or $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$ solves both Problem α and Problem β.

III. Convergence Under Linear Constraints

In general, the accumulation point achieved in step 4 of the generic HOC algorithm in Sec. II.A is not necessarily an optimal solution of

Problem (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,[3] 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:

$$\operatorname*{Min}_{x} f(\boldsymbol{x}) \text{ subject to } A^I\boldsymbol{x} \le \boldsymbol{c}^I, \qquad A^E\boldsymbol{x} = \boldsymbol{c}^E \tag{4}$$

where $f:\mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, A^I (A^E, respectively) is an $m_I \times n$ ($m_E \times n$, respectively) constraint matrix with real entries, $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of optimization variables, and $\boldsymbol{c}^I \in \mathbb{R}^{m_I}$ ($\boldsymbol{c}^E \in \mathbb{R}^{m_E}$, respectively) is a constant vector. Let A be the matrix

$$\begin{bmatrix} A^I \\ A^E \end{bmatrix}$$

The problem is assumed to have a nonempty solution set.

The generic HOC algorithm described in Sec. II.A applied to Problem (4) results in two sequences: $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ and $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$. For an accumulation point $\boldsymbol{x}^*$ of $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ or $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$, define T_a to be the set of indices corresponding to active inequality constraints, i.e.,

$$T_a := \left\{i \mid \left(a_{i1}^I, \ldots, a_{\text{in}}^I\right)\boldsymbol{x}^* = c_i^I\right\}$$

where a_{ij}^I 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 $C(A)$ by

$$C(A) := \left\{\boldsymbol{x} \mid \boldsymbol{x} = \sum_{i \in T_a} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E, \quad a_i \ge 0\right\}$$

where $\boldsymbol{v}_i^I$ ($\boldsymbol{v}_i^E$, respectively) denotes the ith row vector of A^I (A^E, respectively). Let H_α (H_β, respectively) be the unique matrix such that $H_\alpha \boldsymbol{x} = \boldsymbol{y}_\alpha$ ($H_\beta \boldsymbol{x} = \boldsymbol{y}_\beta$, respectively). That is, H_α and H_β are the indicator matrices identifying the linking variables in the α and β decompositions, respectively. Also, define K_α, K_β, and $K_{\alpha\beta}$ as follows:

$$K_\alpha := \begin{bmatrix} A \\ H_\alpha \end{bmatrix}, \qquad K_\beta := \begin{bmatrix} A \\ H_\beta \end{bmatrix}, \qquad K_{\alpha\beta} := \begin{bmatrix} A \\ H_\alpha \\ H_\beta \end{bmatrix}$$

We now define the induced cones $C(K_\alpha)$ and $C(K_\beta)$ as follows:

$$C(K_\alpha) := \left\{\boldsymbol{x} \mid \boldsymbol{x} = \sum_{i \in T_a} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \boldsymbol{e}_{\alpha(i)}, \quad a_i \ge 0\right\}$$

$$C(K_\beta) := \left\{\boldsymbol{x} \mid \boldsymbol{x} = \sum_{i \in T_a} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\beta} t_i \boldsymbol{e}_{\beta(i)}, \quad a_i \ge 0\right\}$$

where $\boldsymbol{e}_i \in \mathbb{R}^n$ is the ith standard row vector and $\alpha(i)$ [$\beta(i)$, respectively] is the index for the ith α-linking (β-linking, respectively) variable.

The Lagrange multiplier theorem for linear constraints (Ref. 10, Proposition 3.4.1) states that $\boldsymbol{x}^* \in \mathbb{R}^n$ is a solution to Problem (4) if and only if there exists a nonnegative vector $\boldsymbol{\lambda}^I$ and a vector $\boldsymbol{\lambda}^E$ such that

$$\nabla f^t(\boldsymbol{x}^*) + \bar{A}^{I^t}\boldsymbol{\lambda}^I + A^{E^t}\boldsymbol{\lambda}^E = \boldsymbol{0} \tag{5}$$

As in the case of only equality constraints, this result is valid even when $\boldsymbol{x}^*$ is not a regular point (Ref. 10, p. 292).

Condition (5) is equivalent to

$$-\nabla f^t(\boldsymbol{x}^*) = \bar{A}^{I^t}\boldsymbol{\lambda}^I + A^{E^t}\boldsymbol{\lambda}^E, \qquad \boldsymbol{\lambda}^I \ge \boldsymbol{0} \tag{6}$$

which can be rephrased as

$$-\nabla f^t(\boldsymbol{x}^*) \text{ belongs to the cone } C(A)$$

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

$$\operatorname*{Min}_{x} f(\boldsymbol{x}) \text{ subject to}$$

$$A^I\boldsymbol{x} \le \boldsymbol{c}^I, \qquad A^E\boldsymbol{x} = \boldsymbol{c}^E, \qquad H_\alpha\boldsymbol{x} = \boldsymbol{d}_\alpha \tag{7}$$

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

$$-\nabla f^t\left(\boldsymbol{x}_\alpha^*\right) \text{ belongs to the cone } C(K_\alpha)$$

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

$$-\nabla f^t\left(\boldsymbol{x}_\beta^*\right) \text{ belongs to the cone } C(K_\beta)$$

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

$$C(A) = C(K_\alpha) \cap C(K_\beta)$$

then $\boldsymbol{x}^*$ is a solution to the optimization problem in Eq. (4).

Proof. As explained in Sec. II.B, $\boldsymbol{x}^*$ solves both Problem α and Problem β. Therefore,

$$-\nabla f^t(\boldsymbol{x}^*) \in C(K_\alpha), \qquad -\nabla f^t(\boldsymbol{x}^*) \in C(K_\beta)$$

Because $C(A) = C(K_\alpha) \cap C(K_\beta)$, one gets $-\nabla f^t(\boldsymbol{x}^*) \in C(A)$, which implies $\boldsymbol{x}^*$ is a solution to the original optimization problem in Eq. (4). □

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

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

Theorem 3.2. Let r be the rank of

$$A = \begin{bmatrix} A^I \\ A^E \end{bmatrix}$$

and $\hat{A}$ be an $r \times n$ submatrix of A with full row rank. If the matrix

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

has full row rank, then $C(A) = C(K_\alpha) \cap C(K_\beta)$.

Proof. Clearly, $C(A) \subset C(K_\alpha)$ and $C(A) \subset C(K_\beta)$. Therefore, $C(A) \subset C(K_\alpha) \cap C(K_\beta)$.

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

$$\boldsymbol{v} = \sum_{i \in T_a} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \boldsymbol{e}_{\alpha(i)}, \qquad a_i \ge 0$$

$$\boldsymbol{v} = \sum_{i \in T_a} d_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} e_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\beta} t_i \boldsymbol{e}_{\beta(i)}, \qquad d_i \ge 0$$

Therefore,

$$\sum_{i \in T_a} (a_i - d_i)\boldsymbol{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i)\boldsymbol{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \boldsymbol{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \boldsymbol{e}_{\beta(i)} = 0 \tag{8}$$

Because $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ form a basis for the row space of

$$\begin{bmatrix} A^I \\ A^E \end{bmatrix}$$

there exist $\gamma_1, \ldots, \gamma_r \in \mathbb{R}$ such that

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

Hence, Eq. (8) becomes

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

Because

$$\hat{K}_{\alpha\beta} = \left(\boldsymbol{v}_1,\dots,\boldsymbol{v}_r,\boldsymbol{e}_{\alpha(1)},\dots,\boldsymbol{e}_{\alpha(n_\alpha)},\boldsymbol{e}_{\beta(1)},\dots,\boldsymbol{e}_{\beta(n_\beta)}\right)^t$$

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

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

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

$$\boldsymbol{v} = \sum_{i\in T_a} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E, \qquad a_i \ge 0$$

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

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

Corollary 3.3. For Problem (4), suppose α and β decompositions are given. Let $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ and $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$ be two sequences obtained by applying HOC to these decompositions, and $\boldsymbol{x}^*$ be an accumulation point of $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ or $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$. If $\hat{K}_{\alpha\beta}$ has full row rank, then $\boldsymbol{x}^*$ is a solution to the optimization problem (4).

IV. Convergence Under Nonlinear Constraints

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

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ subject to } \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \qquad \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0} \tag{9}$$

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 (9) plays a role similar to that of the matrix

$$\begin{bmatrix} A^I \\ A^E \end{bmatrix}$$

in the linear problem (4). Let $J(\boldsymbol{x})$ be the matrix

$$\begin{bmatrix} J^I(\boldsymbol{x}) \\ J^E(\boldsymbol{x}) \end{bmatrix}$$

where $J^I(\boldsymbol{x})$ and $J^E(\boldsymbol{x})$ are the Jacobians of $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$, respectively. This matrix function $J(\boldsymbol{x})$ will be simply referred to as the Jacobian of Problem (9). Define the matrices $K_\alpha(\boldsymbol{x})$, $K_\beta(\boldsymbol{x})$, and $K_{\alpha\beta}(\boldsymbol{x})$ by

$$K_\alpha(\boldsymbol{x}) := \begin{bmatrix} J(\boldsymbol{x}) \\ H_\alpha \end{bmatrix}, \qquad K_\beta(\boldsymbol{x}) := \begin{bmatrix} J(\boldsymbol{x}) \\ H_\beta \end{bmatrix}$$

$$K_{\alpha\beta}(\boldsymbol{x}) := \begin{bmatrix} J(\boldsymbol{x}) \\ H_\alpha \\ H_\beta \end{bmatrix}$$

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

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

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

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

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

where $\boldsymbol{v}_i^I(\boldsymbol{p})$ [$\boldsymbol{v}_i^E(\boldsymbol{p})$, respectively] denotes the ith row vector of $J^I(\boldsymbol{p})$ [$J^E(\boldsymbol{p})$, respectively]. Also, define the induced cones $C(K_\alpha)(\boldsymbol{p})$ and $C(K_\beta)(\boldsymbol{p})$ as follows:

$$C(K_\alpha)(\boldsymbol{p}) := \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \sum_{i\in T_a(\boldsymbol{p})} a_i \boldsymbol{v}_i^I(\boldsymbol{p}) + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E(\boldsymbol{p}) + \sum_{i=1}^{n_\alpha} s_i e_{\alpha(i)}, \quad a_i \ge 0 \right\}$$

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

The generic HOC algorithm described in Sec. II.A applied to Problem (9) results in two sequences: $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^{\infty}$ and $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^{\infty}$. The Lagrange multiplier theorem for nonlinear constraints[10] states that a regular point $\boldsymbol{x}^* \in \mathbb{R}^n$ is a solution to Problem (9) if and only if there exists a nonnegative vector $\boldsymbol{\lambda}^I$ and a vector $\boldsymbol{\lambda}^E$ such that

$$\nabla f^t(\boldsymbol{x}^*) + \bar{J}^I(\boldsymbol{x}^*)^t\boldsymbol{\lambda}^I + J^E(\boldsymbol{x}^*)^t\boldsymbol{\lambda}^E = \boldsymbol{0} \tag{10}$$

Condition (10) is equivalent to

$$-\nabla f^t(\boldsymbol{x}^*) = \bar{J}^I(\boldsymbol{x}^*)^t\boldsymbol{\lambda}^I + J^E(\boldsymbol{x}^*)^t\boldsymbol{\lambda}^E, \qquad \lambda^I \ge \boldsymbol{0} \tag{11}$$

which can be rephrased as

$$-\nabla f^t(\boldsymbol{x}^*) \text{ belongs to the cone } C(J(\boldsymbol{x}^*))$$

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

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ subject to}$$
$$\boldsymbol{h}(\boldsymbol{x}) = 0, \qquad \boldsymbol{g}(\boldsymbol{x}) \le 0, \qquad H_\alpha \boldsymbol{x} = \boldsymbol{d}_\alpha \tag{12}$$

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

$$-\nabla f^t(\boldsymbol{x}_\alpha^*) \text{ belongs to the cone } C(K_\alpha)(\boldsymbol{x}_\alpha^*)$$

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

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

Now the following theorem is immediate.

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

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

then $\boldsymbol{x}^*$ is a solution to the optimization problem (9).

Proof. As explained in Sec. II.B, $\boldsymbol{x}^*$ solves both Problem α and Problem β. Therefore,

$$-\nabla f^t(\boldsymbol{x}^*) \in C(K_\alpha)(\boldsymbol{x}^*), \qquad -\nabla f^t(\boldsymbol{x}^*) \in C(K_\beta)(\boldsymbol{x}^*)$$

Because $C(J)(\boldsymbol{x}^*) = C(K_\alpha)(\boldsymbol{x}^*) \cap C(K_\beta)(\boldsymbol{x}^*)$, one gets $-\nabla f^t(\boldsymbol{x}^*) \in C(J)(\boldsymbol{x}^*)$, which implies $\boldsymbol{x}^*$ is a solution to the original optimization problem (9). □

Theorem 4.2. Let r be the rank of

$$J(\boldsymbol{x}^*) = \begin{bmatrix} J^I(\boldsymbol{x}^*) \\ J^E(\boldsymbol{x}^*) \end{bmatrix}$$

and $\hat{J}(\boldsymbol{x}^*)$ be an $r \times n$ submatrix of $J(\boldsymbol{x}^*)$ with full row rank. If the matrix

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

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

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

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

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

$$\boldsymbol{v} = \sum_{i \in T_a(\boldsymbol{x}^*)} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \boldsymbol{e}_{\alpha(i)}, \qquad a_i \geq 0$$

$$\boldsymbol{v} = \sum_{i \in T_a(\boldsymbol{x}^*)} d_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} e_i \boldsymbol{v}_i^E + \sum_{i=1}^{n_\beta} t_i \boldsymbol{e}_{\beta(i)}, \qquad d_i \geq 0$$

Therefore,

$$\sum_{i \in T_a(\boldsymbol{x}^*)} (a_i - d_i) \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} (b_i - e_i) \boldsymbol{v}_i^E + \sum_{i=1}^{n_\alpha} s_i \boldsymbol{e}_{\alpha(i)} - \sum_{i=1}^{n_\beta} t_i \boldsymbol{e}_{\beta(i)} = 0 \tag{13}$$

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

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

Hence, Eq. (13) becomes

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

Because the matrix

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

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

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

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

$$\boldsymbol{v} = \sum_{i \in T_a(\boldsymbol{x}^*)} a_i \boldsymbol{v}_i^I + \sum_{i=1}^{m_E} b_i \boldsymbol{v}_i^E, \qquad a_i \geq 0$$

This implies that $\boldsymbol{v} \in C(J)(\boldsymbol{x}^*)$. □

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

Corollary 4.3. For Problem (9), suppose α and β decompositions are given. Let $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^\infty$ and $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^\infty$ be two sequences obtained by applying HOC to these decompositions and $\boldsymbol{x}^*$ be an accumulation point of $\{\boldsymbol{x}_{\alpha_i}\}_{i=1}^\infty$ or $\{\boldsymbol{x}_{\beta_i}\}_{i=1}^\infty$. If $\boldsymbol{x}^*$ is a regular point and $\hat{K}_{\alpha\beta}(\boldsymbol{x}^*)$ has full row rank, then $\boldsymbol{x}^*$ is a solution to the optimization problem (9).

Remark 4.4. $\hat{K}_{\alpha\beta}(\boldsymbol{x}^*)$ has full row rank only if the sets of α- and β-linking variables are disjoint and only if the sum of the rank of $J(\boldsymbol{x}^*)$ plus the total number of linking variables is less than or equal to the total number of variables.

V. HOC Algorithm for Nonlinear Convex Problems

Step 1. Apply the generic HOC algorithm starting at a point $\boldsymbol{x}_0$ for α and β decompositions that make $\hat{K}_{\alpha\beta}(\boldsymbol{x}_0)$ full row rank. Let $\boldsymbol{x}^*$ be a resulting accumulation point.

Step 2. If $\hat{K}_{\alpha\beta}(\boldsymbol{x}^*)$ has full row rank, then (by Corollary 4.3) conclude that $\boldsymbol{x}^*$ is a solution to the original optimization problem (9) and exit.

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

This process is repeated until we reach a point $\boldsymbol{x}^\dagger$ such that $\hat{J}(\boldsymbol{x}^\dagger)$ has full row rank (or we reach a maximum number of iterations). Then this point $\boldsymbol{x}^\dagger$ is, by Corollary 4.3, a solution to the original optimization problem (9). Valid decompositions are generated using hypergraph-based model decomposition techniques[3,9] described in Sec. VI.A.

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

$$\hat{K}_{\alpha\beta}(\boldsymbol{x}) := \begin{bmatrix} \hat{J}(\boldsymbol{x}) \\ H_\alpha \\ H_\beta \end{bmatrix}$$

has full row rank for every $\boldsymbol{x}$ in the feasible domain, then $\hat{K}_{\alpha\beta}(\boldsymbol{x}^*)$ also has full rank, and HOC is convergent. The method used in the proof (Ref. 3, Theorem 3.5) can be used to show that $\hat{K}_{\alpha\beta}(\boldsymbol{x})$ has full row 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, we can check computationally whether $\hat{K}_{\alpha\beta}(\boldsymbol{x})$ has full row rank everywhere. Some symbolic computer algebra systems, e.g., Macauley[¶] and Singular,[11] are capable of computing the projection of an ideal onto a subring.

VI. Computational Results

A. Obtaining Two Distinct Decompositions

Recall that the 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 ith function depends on the jth 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 decomposition algorithm proposed in Michelena and Papalambros[9] uses a hypergraph representation of the design model that is then optimally partitioned into weakly connected subgraphs that can be identified with subproblems. Design variables are represented by the hypergraph edges, whereas design constraints interrelating these variables are represented by the nodes. These constraints may be given as algebraic equations, response surfaces, or look-up tables, or evaluated using simulation or analysis black boxes. The formulation can account for computational demands and resources by means of design constraint weighting coefficients and partitions sizes, respectively, and for the strength of interdependencies between the analysis modules contained in the model by means of design variable weighting coefficients.

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 $\boldsymbol{x}_0$ (Corollary 4.3) can be found by the following heuristic:

1. Apply the hypergraph-based model decomposition algorithm** to Problem P to obtain an α decomposition.

2. In the process of obtaining a β decomposition, penalize 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. (A variable

¶Bayer, D., and Stillman, M., *Macauley: A Computer Algebra System for Algebraic Geometry*, 1995, available by anonymous ftp from ftp.math.harvard.edu.

**An implementation of this decomposition algorithm can be found online at http://arc.engin.umich.edu/graph_part.html (cited December 7, 1998).

Table 1 Comparison of CPU run times for nonlinear optimization problems of various sizes (initial point $x_0 = -0.1$)

Problem	No. of variables	No. of constraints	No. of subproblems	AAO Objective	AAO Run time	HOC Objective	HOC Serial run time	HOC Parallel run time[a]	HOC No. of HOC 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	1,446	33.4092	30.18	2.835	3
P_8	375	315	30	50.1139	6,904	50.1139	45.22	2.832	3
P_9	500	420	40	66.8185	20,539	66.8185	60.77	2.890	3

[a]Run time is measured in CPU seconds on a Sun UltraSparc 1.

Table 2 Comparison of CPU run times for nonlinear optimization problems of various sizes (initial point $x_0 = 0$)

Problem	No. of variables	No. of constraints	No. of subproblems	AAO Objective	AAO Run time	HOC Objective	HOC Serial run time	HOC Parallel run time[a]	HOC No. of HOC 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	1,701	33.4092	19.32	1.885	1
P_8	375	315	30	50.1139	10,869	50.1139	28.85	1.877	1
P_9	500	420	40	66.8187	27,662	66.8185	38.99	1.910	1

[a]Run time is measured in CPU seconds on a Sun UltraSparc 1.

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 model decomposition algorithm described in Ref. 9). 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 whether 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.

B. Example Problems

The HOC algorithm developed in this paper 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 21×25 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 preceding decomposition heuristic to example problem P_1. Maple[12] was used to verify that these two decompositions do satisfy the convergence condition in Corollary 4.3 for the initial point $\boldsymbol{x}_0$. The α decomposition in Fig. 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). Whereas 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. A MATLAB™ (Ref. 13) implementation of the sequential quadratic programming (SQP) algorithm, constr, was used for this purpose. The HOC iteration process stops if the relative difference between the values of the objective function 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) approach (i.e., one not using decompositions), the problems were solved in both ways. Even though the original problem yields a sparse matrix, each of the subproblems in the HOC may not be really sparse. A sparse optimizer was not used with either approach to ensure fair comparison of performances. An AAO approach with sparse optimizers may turn out to be comparable in performance with HOC. However, HOC may increase computational efficiency of any general-purpose optimizer in the case of sparse problems.

The results for P_1 and the other problems of larger sizes are shown in Tables 1 and 2 for two different initial points. Run time was measured in CPU seconds on a Sun UltraSparc 1. Run times include only function calls to constr, excluding I/O and data transfer between α and β decompositions. Each run time represents the average run time for five separate runs of the algorithm; the times of the five runs were consistently close. Serial run time is measured for the HOC computation in which the subproblems are solved sequentially, whereas parallel run time is measured for the HOC computation in which the subproblems are simulated to be solved in parallel.

C. Discussion

HOC has shorter parallel and serial run times than AAO for all problems and initial points except for problem P_1 with initial point $\boldsymbol{x}_0 = -\mathbf{0.1} := -(0.1, 0.1, \ldots, 0.1)$. 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. AAO converges faster when it is started with $\boldsymbol{x}_0 = -\mathbf{0.1}$ as the initial point, whereas HOC converges faster when it is started with $\boldsymbol{x}_0 = \mathbf{0}$ as the initial point.

Note that HOC for P_9 has parallel and serial run times that are 7100–14500 and 340–710 times shorter than the AAO run times, respectively, depending on the initial point. Surprisingly, HOC terminates after three and one iterations regardless of problem size. HOC serial run times vary linearly with the size of the problem, whereas parallel run times remain about constant. This result is expected because subproblem sizes are similar for all nine problems. AAO

[††]These example problems are available online at http://arc.engin.umich.edu/HOC_examples (cited December 7, 1998).

```
             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      ..................@.....@
```

α **decomposition**

```
             DESIGN  VARIABLES

             0001111000000111111222222
             3960268124578134579012345


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

β **decomposition**

Fig. 2 Decompositions of example problem P_1.

run times seem to vary polynomially (third or fourth 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 example, for $x_0 = -\mathbf{0.1}$, P_9 was solved in less than 2 min by HOC, whereas its AAO solution took over 7.5 h on the same workstation. This is also true for the smaller problems, for example, HOC took 11.9 and 35.6 s to solve P_3 and P_6, respectively, whereas AAO took 17.3 and 363 s to solve the same problems, respectively. This result demonstrates the advantage of using HOC for large-scale problems with many loosely linked subproblems.

VII. Conclusions

Hierarchical overlapping coordination takes advantage of two or more decompositions of an optimization model to coordinate the solution of the underlying optimization problem. Each decomposition effects smaller subproblems whose separate solutions result in the solution of the original problem with possible run-time savings. Additional savings could be achieved by implementing the algorithm on a parallel computer. Convergence of the algorithm depends on the way the model decompositions interact with each other. In this paper we present a sufficient condition for convergence that can be computationally verified for decompositions of a nonlinear convex problem. This condition also guarantees convergence to a stationary point in the case of more general smooth, nonlinear optimization problems. Moreover, the condition together with model partitioning methods can help in generating appropriate problem decompositions.

Acknowledgments

This research was supported by the U.S. Army Automotive Research Center for Modeling and Simulation of Ground Vehicle under Contract DAAE07-94-C-R094 and by the Oakland University Foundation. This support is gratefully acknowledged.

References

[1]Macko, D., and Haimes, Y., "Overlapping Coordination of Hierarchical Structures," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 10, 1978, pp. 745–751.

[2]Shima, T., and Haimes, Y., "The Convergence Properties of Hierarchical Overlapping Coordination," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-14, No. 1, 1984, pp. 74–87.

[3]Park, H., Michelena, N., Kulkarni, D., and Papalambros, P., "Convergence Criteria for Hierarchical Overlapping Coordination of Linearly Constrained Convex Problems," *Computational Optimization and Applications* (submitted for publication).

[4]Wagner, T., and Papalambros, P., "A General Framework for Decomposition Analysis in Optimal Design," *Advances in Design Automation*, edited by B. Gilmore, Vol. 2, American Society of Mechanical Engineers, New York, 1993, pp. 315–322.

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

[6]Nelson, S. A., and Papalambros, P. Y., "Sequentially Decomposed Programming," *AIAA Journal*, Vol. 35, No. 7, 1997, pp. 1209–1216.

[7]Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, 1988, Pt. 1, pp. 51–78; also NASA TM-101494, 1988.

[8]Braun, R., and Kroo, I., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State of the Art*, edited by N. Alexandrov and M. Hussaini, Society of Industrial and Applied Mathematics, Philadelphia, 1997, pp. 98–116.

[9]Michelena, N., and Papalambros, P., "A Hypergraph Framework to Optimal Model-Based Decomposition of Design Problems," *Computational Optimization and Applications*, Vol. 8, No. 2, 1997, pp. 173–196.

[10]Bertsekas, D., *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995, p. 292.

[11]Greuel, G., Pfister, G., and Schönemann, H., "Singular Reference Manual," Centre for Computer Algebra, Repts. on Computer Algebra No. 12, Univ. of Kaiserslautern, Kaiserslautern, Germany, May 1997.

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

[13]*Using MATLAB*, Ver. 5.1, The MathWorks, Inc., Natick, MA, 1997.

A. D. Belegundu
Associate Editor