A Regularized Inexact Penalty Decomposition Algorithm for Multidisciplinary Design Optimization Problems With Complementarity Constraints

Shen Lu
e-mail: shenlu2@illinois.edu

Harrison M. Kim
e-mail: hmkim@illinois.edu

Department of Industrial and Enterprise Systems
Engineering,
University of Illinois,
Urbana, IL 61801

Economic and physical considerations often lead to equilibrium problems in multidisciplinary design optimization (MDO), which can be captured by MDO problems with complementarity constraints (MDO-CC)—a newly emerging class of problem. Due to the ill-posedness associated with the complementarity constraints, many existing MDO methods may have numerical difficulties solving this class of problem. In this paper, we propose a new decomposition algorithm for the MDO-CC based on the regularization technique and inexact penalty decomposition. The algorithm is presented such that existing proofs can be extended, under certain assumptions, to show that it converges to stationary points of the original problem and that it converges locally at a superlinear rate. Numerical computation with an engineering design example and several analytical example problems shows promising results with convergence to the all-in-one solution.
[DOI: 10.1115/1.4001206]

1 Introduction

The research area of multidisciplinary design optimization (MDO) has been intensively investigated during the last several decades. Particularly, rich contribution has been made concerning two fundamental issues in this research area: (1) the *decomposition* modeling of the multidisciplinary system and (2) the *coordination* among individual subsystems to efficiently achieve the overall optimum. Previous research has addressed the core components of static, single multidisciplinary system design with models in systems engineering, nonlinear programming, and multistage programming. Yet, the ever increasing intricacy and complexity of engineering design often requires generalization to more complex models to facilitate the notion of *dynamic* design and operations, such as system-of-systems as well as competitive agents.

Specifically, this paper considers a generalization of multilevel MDO in which its lower-level constraints represent the *solution sets* of optimization or game-theoretic problems. Such settings include, for example, (1) iterative and interactive design decision making processes captured by hierarchical frameworks or game protocols [1–4]; (2) multimode system design where a system may undergo multiple operating conditions represented as lower-level constraints, but not necessarily satisfy all of them, i.e., switch among different modes; and (3) biddings among multiple teams or suppliers given upper-level system requirements [5,6]. These problems would fall under the setting of multidisciplinary design optimization problem with complementarity constraints (MDO-CC) [7]—a newly emerging class of problem, which relates the MDO with the areas of mathematical program with complementarity constraints (MPCC) [8].

1.1 The Relevance of Complementarity Constraints to Mechanical and Multidisciplinary Design Optimization.

Although few works are reported introducing complementarity constraints (CCs) into mechanical and multidisciplinary design optimization, CC can actually be related to mechanical and multidisciplinary design optimization in many scenarios. The most straightforward scenario addresses complementarities in physical feasibility. For example, complementarity can be used to "switch" a system from one possible working mode, e.g., weather condition, to another; it also models the cases in structural design, where Hooke's law is turned on/off beyond control of designer [9]. While these scenarios could as well be handled by introducing discrete variables, CC provides a useful alternative with potentially less computation under continuous setting.

A potentially more significant application of CC in mechanical and multiple disciplinary design optimizations involves the economic aspects of design decision making. For instance, in many settings, a designer may need to consider the competition among a collection of convex agents (e.g., market players). This can be modeled by adding the sufficient first-order optimality conditions, in the format of CCs, as design constraints, leading to an MPCC or MDO-CC. Such a framework ensures that strategic concerns can be better captured within the design phase and further enables the promising research on integrating enterprise decision and engineering design [5,10–13].

The need to consider CC in mechanical and multidisciplinary design is further motivated by the procedural or philosophical aspects of design. Design decision making, especially product design in a complex enterprise context, may involve interaction among different parties, e.g., collaborative design teams, manufacturing teams and maintenance teams. In some settings, these parties are often competing for some resources, implying that the problem could be modeled as a noncooperative game [2,14,15]. In some settings, there is a definite notion of a leader and a follower; in particular, a designer makes a decision subject to subsystems competing for resources contingent on such a decision—a single-leader multifollower game, referred to as a Stackelberg game [3,1,16,17]. The equilibrium conditions of many games such as Nash game, Stackelberg game, etc., can be reformulated into

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received April 14, 2009; final manuscript received January 29, 2010; published online April 13, 2010. Assoc. Editor: Zissimos P. Mourelatos. Paper presented at the ASME 2009 Design Automation Conference (DAC 2009, DETC2009-87278), San Diego, CA, Aug. 3–Sept. 2.

MPCCs. For example, the equilibrium conditions of a Nash game, if exists, can be captured by an aggregation of the optimality conditions of all the competing agents, which can be represented as a set of CCs. Due to this reason, MPCC has been traditionally used as a numerical solver of these games [8]. With the aid of CC, these procedural aspects of design processes can be captured and included in mechanical and multidisciplinary design optimization.

1.2 Literature Review. MDO problems can be solved directly with so-called all-in-one (AIO) approaches, which handle all the variables in a single optimization problem. The implementation of the AIO approach is straightforward; however, it may become impractical, undesirable, or even impossible, as the complexity of the problem increases. As an alternative to the AIO approaches, decomposition-based MDO methods have been investigated extensively over the past few decades—for example, monotonicity-based decomposition method (MBDM) [18], concurrent subspace optimization [19], bilevel integrated system synthesis [20], collaborative optimization (CO) [21,22], the constraint margin approach [23], analytical target cascading (ATC) [24–26], penalty decomposition (PD) [27], and augmented Lagrangian decomposition (ALD) [28]. Solving MDO problems with decomposition approaches could be advantageous for many reasons: computationally, it breaks the AIO problems into smaller subproblems usually easier to solve; it also allows specialized algorithms to be applied to each subproblems. Organizationally, it keeps the individual disciplinary design optimizations as independent as possible with minimum amount of communication, making it possible to integrate existing disciplinary analysis codes at small expense.

Among all the variants of MDO, the quasi-separable MDO problem has gained particular attention during recent years. Many of the above mentioned MDO methods (e.g., MBDM, CO, constraint margin approach, PD, and ALD) can be considered as the quasi-separable MDO problem. In addition to these, Lu et al. [29] proposed an ATC variant with local objectives under the context of multimode design optimization. Among these approaches, PD, ATC, and ALD have been shown to have formulations whose solutions satisfy the Karush–Kuhn–Tucker (KKT) conditions of the original problems under certain assumptions. Additionally, it is also shown that the PD algorithms (inexact PD and exact PD) converge locally at a superlinear rate.

MPCCs represent an active research area, which is not well connected to the MDO. An example of the complementarity constraint is given as follows:

$$\mathbf{0} \le \mathbf{F}(\mathbf{x}) \perp \mathbf{G}(\mathbf{x}) \ge \mathbf{0} \tag{1}$$

where $\mathbf{x}$ represents the variables and $\mathbf{F}$ and $\mathbf{G}$ are multifunctions in $\mathbf{R}^p$. Particularly, the symbol $\perp$ indicates the following inequalities:

$$\mathbf{F} \ge \mathbf{0}, \quad \mathbf{G} \ge \mathbf{0}, \quad \mathbf{G}(\mathbf{x},\mathbf{y}) \circ \mathbf{F}(\mathbf{x},\mathbf{y}) \le \mathbf{0} \tag{2}$$

where the symbol $\circ$ represents the Hadamard product, i.e., the term-by-term product operation between two vectors: $\mathbf{a}\circ\mathbf{b} = [a_1,\ldots,a_n]^T \circ [b_1,\ldots,b_n]^T = [a_1b_1,\ldots,a_nb_n]^T$.

In order to solve an MPCC, one intuitive approach is to reformulate it into a nonlinear programming problem through replacing the CCs (Eq. (1)) with its equivalent inequality constraints (Eq. (2)). However, the resulting nonlinear program (NLP) usually fails to satisfy the linear independence constraint qualification (LICQ) [30] and the weaker Mangasarian–Fromovitz constraint qualification (MFCQ) [30] at every feasible point. The failure of these constraint qualifications may have important negative numerical implications: the multiplier set may be unbounded; the active constraint normals may be linearly dependent; and a linear relaxation of the reformulated nonlinear programming problem can become inconsistent arbitrarily close to a solution to the MPCC [31]. As a consequence, existing nonlinear programming techniques may have difficulties solving this type of problem.

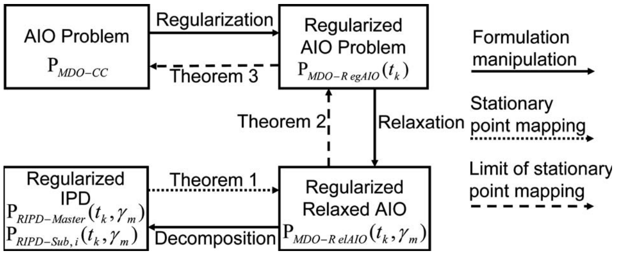


Fig. 1 The flow of formulation manipulation and stationary point mapping

Significant efforts have been made to investigate MPCC solution algorithm over the past few years. Fletcher et al. [31] followed the reformulation approach and report promising results using sequential quadratic programming (SQP) methods. Anitescu [32] provided global convergence theory for SQP methods. Some other methods solve a sequence of nonlinear programs with penalized complementarity constraints [33,34]. An important class of methods, known as regularization methods, requires the solution of a sequence of regularized problems involving the relaxed constraints $\mathbf{G}(\mathbf{x},\mathbf{y})\circ\mathbf{F}(\mathbf{x},\mathbf{y}) \le \mathbf{t}_k$ with $\mathbf{t}_k \to 0$. These regularized problems may be solved by interior methods or by SQP methods. Along this line of research, Luo et al. [8], Liu and Sun [35], and Raghunathan and Biegler [36] presented interior methods under various assumptions; Shanbhag [37] proposed an interior point method that converges to a second-order KKT point and extended this method to stochastic MPCC using scenario-based decomposition; and DeMiguel et al. [38] discussed a two-sided relaxation scheme and provided local convergence theory for an interior method coupled with such a relaxation scheme.

MDO-CCs are not frequently addressed in existing literature. Lu et al. [7] recently presented an augmented Lagrangian decomposition formulation for this problem to show the equivalence between the AIO formulation and the decomposed formulation. Additionally, Shanbhag [37] proposed a scenario-based decomposition formulation for stochastic MPCCs, and then solved it as an AIO problem with a parallel algorithm. In this paper, we propose a new decomposition approach for the MDO-CC based on the regularization technique and inexact penalty decomposition (IPD). While the IPD is established in existing MDO literature, this is the first time it is adapted to successfully solve the MDO-CC. The proposed algorithm is derived such that existing convergence proofs of the IPD method can be extended, under certain assumptions, to show convergence to strongly-stationary points of the original AIO problem. Additionally, superlinear convergence rate can be derived from standard results associated with the master and subproblem solvers. The proposed method is applied to two classes of examples and preliminary numerical results are encouraging.

The remainder of this paper is organized as follows. In Sec. 2, the formulation of the MDO-CC is stated, followed by its regularized inexact penalty decomposition (RIPD) formulation. Additionally, the connection between stationary solutions of the two formulations is established. A solution algorithm based on the RIPD formulation is presented in Sec. 3 and numerical results of the proposed algorithm are presented and discussed in Sec. 4.

2 Formulation

In this section, the formulations of the MDO-CC are presented. We follow the flow of formulation manipulation shown in Fig. 1: the MDO-CC is first stated in its AIO formulation; then its CCs are regularized, resulting in a regularized AIO formulation; after that, we relax the regularized AIO formulation by introducing local copies of the linking variables and penalizing the inconsistency among these copies, deriving a regularized, relaxed AIO formulation; finally, this formulation is decomposed into a bilevel

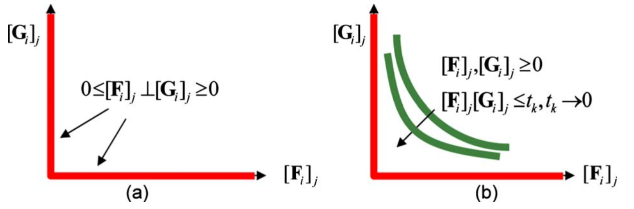

$[\mathbf{G}_i]_j$
$0\le[\mathbf{F}_i]_j \perp [\mathbf{G}_i]_j \ge 0$
$[\mathbf{F}_i]_j$
(a)
$[\mathbf{G}_i]_j$
$[\mathbf{F}_i]_j,[\mathbf{G}_i]_j \ge 0$
$[\mathbf{F}_i]_j[\mathbf{G}_i]_j \le t_k, t_k \to 0$
$[\mathbf{F}_i]_j$
(b)

Fig. 2 The feasible space of $0\le[\mathbf{G}_i]_j \perp [\mathbf{F}_i]_j \ge 0$ and its regularization [7] (*a* and *b*)

regularized IPD formulation. In addition to the formulation manipulation, the connection among solutions of these formulations is also presented.

2.1 AIO Formulation. We consider a general quasi-separable MDO-CC with n subsystems, whose AIO formulation is given as follows:

$$
\begin{aligned}
\mathrm{P}_{\text{MDO-CC-AIO}}: \min_{\mathbf{y},\mathbf{x}_1,\ldots,\mathbf{x}_n} & \sum_{i=1}^{n} f_i(\mathbf{x}_i,\mathbf{y}) \\
\text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}) \le \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{h}_i(\mathbf{x}_i,\mathbf{y}) = \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{0} \le \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}) \perp \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n
\end{aligned}
\tag{3}
$$

where $\mathbf{y}$ represents a vector of linking variables shared by all the n subsystems; and $\mathbf{x}_i$ represents the vector of local variables only relevant to subsystem i $(i=1,\ldots,n)$. The problem is quasi-separable in that the system objective is the summation of the subsystem objectives f_i, and that the subsystem objectives and constraints $\mathbf{g}_i$, $\mathbf{h}_i$, $\mathbf{G}_i$, and $\mathbf{F}_i$ are associated exclusively with subsystem i, depending only on the linking variables and the subsystem's local variables. The local complementarity constraint ($\perp$ symbol) indicates that $\mathbf{G}_i$ and $\mathbf{F}_i$ are non-negative and that $[\mathbf{G}_i]_j[\mathbf{F}_i]_j=0$ (shown in Fig. 2(*a*)) for $j=1,\ldots,p$, where $\mathbf{G}_i,\mathbf{F}_i$ are multifunctions in $\mathbb{R}^p$. We assume throughout this paper that the objective and constraint functions are three times continuously-differentiable. This assumption is required to derive the proposed RIPD algorithm (refer to Theorem 1).

In order to avoid the numerical difficulties associated with the CCs, we follow the regularization methods [39], which replace a CC with non-negative constraints on both of the two components and an inequality constraint on the product of the two. Additionally, the constraint on component product is further relaxed by a positive scalar so that the relaxed problem satisfies the LICQ. The regularization scheme is illustrated in Fig. 2(*a*) shows the feasible space of the original CC, which coincides with the two non-negative axes; such a feasible space fail to satisfy the MFCQ in that none of its points satisfies inequalities (Eq. (2)) strictly [40]. Figure 2(*b*) shows the feasible space of the regularized constraints, which resides between the two non-negative axes and the solid curve; its strict feasible region is nonempty for any positive scalar t_k. The regularized AIO formulation is as follows:

$$
\begin{aligned}
\mathrm{P}_{\text{MDO-RegAIO}}(t_k): \min_{\mathbf{y},\mathbf{x}_1,\ldots,\mathbf{x}_n} & \sum_{i=1}^{n} f_i(\mathbf{x}_i,\mathbf{y}) \\
\text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}) \le \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{h}_i(\mathbf{x}_i,\mathbf{y}) = \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}) \circ \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}) \le t_k\mathbf{e}, \quad \forall\, i=1,\ldots,n
\end{aligned}
\tag{4}
$$

where the symbol $\circ$ represents the Hadamard product, i.e., the term-by-term product operation between two vectors: $\mathbf{a}\circ\mathbf{b} = [a_1,\ldots,a_n]^T \circ [b_1,\ldots,b_n]^T = [a_1b_1,\ldots,a_nb_n]^T$. Additionally, $\{t_k\}$ is a positive descent sequence that converges to zero, and $\mathbf{e}$ is a vector of unit elements. Note that as $\{t_k\}$ approaches zero, the feasible space of $\mathrm{P}_{\text{MDO-RegAIO}}(t_k)$ generally converges to that of the original AIO problem in Eq. (3).

2.2 Bilevel Decomposed Formulation. Implementation of the AIO problem is straightforward in general, but obtaining its solution is usually computationally difficult due to the problem size. An alternative to the AIO approach is the decomposition-based approaches [41,42], where the original AIO problem is decomposed into a set of interrelated subproblems and solved through an iterative process of subproblem optimization and coordination among them. Using decomposition-based approach can be advantageous, as it reduces the size of individual problems by decomposition while limiting the communication among subproblems only to where necessary via linking variables.

As an initial step to decomposing (Eq. (4)), we introduce duplicated copies of the linking variables $\mathbf{y}_i$ in each subsystem i to separate local constraints. Also, additional constraints are introduced to ensure consistency among $\mathbf{y}_i$'s. The resulted regularized, modified AIO problem is given as follows:

$$
\begin{aligned}
\mathrm{P}_{\text{MDO-ModAIO}}(t_k): \min_{\mathbf{y},\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{x}_1,\ldots,\mathbf{x}_n} & \sum_{i=1}^{n} f_i(\mathbf{x}_i,\mathbf{y}_i) \\
\text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}_i) \le \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{h}_i(\mathbf{x}_i,\mathbf{y}_i) = \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \circ \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \le t_k\mathbf{e}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{c}_i = \mathbf{y} - \mathbf{y}_i = \mathbf{0}, \quad \forall\, i=1,\ldots,n
\end{aligned}
\tag{5}
$$

On the basis of Eq. (5), a second manipulation is applied so that the consistency constraints given as $\mathbf{c}_i=\mathbf{0}$ are relaxed, and the corresponding violations are penalized in the format of a quadratic penalty function. As a result of these manipulations, the regularized, relaxed AIO formulation (or simply relaxed AIO for convenience) is given as follows:

$$
\begin{aligned}
\mathrm{P}_{\text{MDO-RelAIO}}(t_k,\gamma_m): \min_{\mathbf{y},\mathbf{y}_1,\ldots,\mathbf{y}_n,\mathbf{x}_1,\ldots,\mathbf{x}_n} & \sum_{i=1}^{n} [f_i(\mathbf{x}_i,\mathbf{y}_i) + \gamma_m\|\mathbf{y}-\mathbf{y}_i\|_2^2] \\
\text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}_i) \le \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{h}_i(\mathbf{x}_i,\mathbf{y}_i) = \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0}, \quad \forall\, i=1,\ldots,n \\
& \mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \circ \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \le t_k\mathbf{e}, \quad \forall\, i=1,\ldots,n
\end{aligned}
\tag{6}
$$

where γ_m denotes the penalty parameter satisfying $\{\gamma\}\to\infty$.

We note that Eq. (6) consists of n subsystems coupled through the linking variables $\mathbf{y}$ only. Therefore, it can be decomposed into a bilevel program through holding $\mathbf{y}$ constant at each subsystem. The resulted subsystems are presented in Eq. (8). In addition, the optimal subsystem objective values under given $\mathbf{y}$ setting are used to define a master problem, as shown in Eq. (7)

$$\mathrm{P}_{\text{RIPD-Master}}(t_k,\gamma_m):\min_{\mathbf{y}} F^*(\mathbf{y}) = \sum_{i=1}^{n} F_i^*(\mathbf{y}) \tag{7}$$

where $F_i^*(\mathbf{y})$ is the optimal subsystem objective value given $\mathbf{y}$

$$\begin{aligned}
\mathrm{P}_{\text{RIPD-Sub},i}(t_k,\gamma_m):F_i^*(\mathbf{y}) = \min_{\mathbf{y}_i,\mathbf{x}_i} f_i(\mathbf{x}_i,\mathbf{y}_i) + \gamma_m\|\mathbf{y}-\mathbf{y}_i\|_2^2 \\
\text{subject to} \quad \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}_i) \le \mathbf{0} \\
\mathbf{h}_i(\mathbf{x}_i,\mathbf{y}_i) = \mathbf{0} \\
\mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0} \\
\mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \ge \mathbf{0} \\
\mathbf{G}_i(\mathbf{x}_i,\mathbf{y}_i) \circ \mathbf{F}_i(\mathbf{x}_i,\mathbf{y}_i) \le t_k\mathbf{e}
\end{aligned} \tag{8}$$

We call the above formulation (Eqs. (7) and (8)) the RIPD formulation to be differentiated from the IPD formulation [27].

2.3 Connection Among the Proposed Formulations.

This subsection is devoted to mapping the stationarity conditions of the RIPD formulation (Eqs. (7) and (8)) to those of the AIO formulation (Eq. (3)). We employ the regularized, relaxed AIO problem (Eq. (6)) and the regularized AIO problem (Eq. (4)) as intermediate steps to facilitate this mapping. The flow of stationary point mapping is indicated by the dotted and dashed arrows in Fig. 1: an accumulative stationary point of Eqs. (7) and (8) is first mapped to a stationary point of Eq. (6) through Theorem 1; then a limit of these stationary points is mapped to a stationary point of Eq. (4) following Theorem 2; finally a limit of solutions to Eq. (4) is mapped to a strongly-stationary point of Eq. (3) through Theorem 3. While these theorems are already well-established in their respective areas of research, our contribution is to show that they can be combined in the context of a new type of problem, namely, the MDO-CC, to effectively derive convergence results. Before proceeding, we provide definitions for several concepts related to MPCC.

Definition 1 (strong-stationarity conditions) for an MPCC

$$\begin{aligned}
\mathrm{P}_{\text{MPCC}}:\min_{\mathbf{x}} f(\mathbf{x}) \\
\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\
\mathbf{h}(\mathbf{x}) = \mathbf{0} \\
\mathbf{F}(\mathbf{x}) - \mathbf{s} = \mathbf{0} \\
\mathbf{G}(\mathbf{x}) - \mathbf{t} = \mathbf{0} \\
\mathbf{0} \le \mathbf{s} \perp \mathbf{t} \ge \mathbf{0}
\end{aligned} \tag{9}$$

a point $\mathbf{z} \equiv (\mathbf{x},\mathbf{s},\mathbf{t})$ is *strongly-stationary* if and only if there exist multipliers $(\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\nu}_1,\boldsymbol{\nu}_2)$ satisfying

$$\begin{aligned}
\begin{pmatrix} \nabla\mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \nabla\mathbf{g} & \mathbf{0} & \mathbf{0} \\ \nabla\mathbf{h} & \mathbf{0} & \mathbf{0} \\ \nabla\mathbf{F} & -\boldsymbol{I} & \mathbf{0} \\ \nabla\mathbf{G} & -\boldsymbol{I} & \mathbf{0} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\lambda} \\ \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{pmatrix} - \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{pmatrix} = \mathbf{0} \\
\mathbf{0} \le \boldsymbol{\mu} \perp -\mathbf{g}(\mathbf{x}) \ge \mathbf{0} \\
\mathbf{h}(\mathbf{x}) = \mathbf{0} \\
\mathbf{F}(\mathbf{x}) - \mathbf{s} = \mathbf{0} \\
\mathbf{G}(\mathbf{x}) - \mathbf{t} = \mathbf{0} \\
\mathbf{0} \le \mathbf{s} \perp \mathbf{t} \ge \mathbf{0} \\
[\boldsymbol{\nu}_1]_j[\mathbf{s}_i]_j = 0, \quad \forall j \\
[\boldsymbol{\nu}_2]_j[\mathbf{t}_i]_j = 0, \quad \forall j \\
\text{if} \quad [\mathbf{s}]_j = [\mathbf{t}]_j = 0, \quad \text{then} \quad [\nu_1]_j \ge 0 \quad \text{and} \quad [\nu_2]_j \ge 0, \quad \forall j
\end{aligned} \tag{10}$$

where ∇ denotes the Jacobian for a vector function.

Let $\mathcal{A}_1,\mathcal{A}_2 \subseteq \{1,\ldots,m\}$ be the sets of indices corresponding to s and t respectively, which satisfy $\mathcal{A}_1 \cup \mathcal{A}_2 = \{1,\ldots,m\}$. These sets can be employed to construct a *relaxed nonlinear program* (relaxed NLP)

$$\begin{aligned}
\mathrm{P}_{\text{MPCC-RNLP}}:\min_{\mathbf{x}} f(\mathbf{x}) \\
\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\
\mathbf{h}(\mathbf{x}) = \mathbf{0} \\
\mathbf{G}(\mathbf{x}) - \mathbf{s} = \mathbf{0} \\
\mathbf{F}(\mathbf{x}) - \mathbf{t} = \mathbf{0} \\
[\mathbf{s}]_j = \mathbf{0}, \quad \forall j \in \mathcal{A}_2^{\perp} \\
[\mathbf{t}]_j = \mathbf{0}, \quad \forall j \in \mathcal{A}_1^{\perp} \\
[\mathbf{s}]_j \ge \mathbf{0}, \quad \forall j \in \mathcal{A}_1 \\
[\mathbf{t}]_j \ge \mathbf{0}, \quad \forall j \in \mathcal{A}_2
\end{aligned} \tag{11}$$

The notion of strong-stationarity is intimately related to the relaxed NLP in that a point is a strongly-stationary solution of Eq. (9) if and only if it is a stationary point of Eq. (11) (see Proposition 4.1 of Ref. [31]).

In addition to the stationarity conditions, we also provide the definition of the following constraint qualification.

Definition 2 (MPCC-LICQ). *The MPCC (Eq. (9)) satisfies MPCC linear independence constraint qualification if the relaxed NLP (Eq. (11)) satisfies LICQ.*

The stationarity conditions of the RIPD formulation (Eqs. (7) and (8)) are mapped to those of the regularized, relaxed AIO formulation (Eq. (6)) following Theorems 4.9 and 4.8 of Ref. [27]. We provide these theorems in the context of the MDO-CC.

Theorem 1. *Assume that f_i, g_i, h_i, G_i, and F_i are three times continuous differentiable. For any t_k, γ_m, let $(\mathbf{y}(t_k,\gamma_m),\mathbf{y}_1(t_k,\gamma_m),\ldots,\mathbf{y}_n(t_k,\gamma_m),\mathbf{x}_1(t_k,\gamma_m),\ldots,\mathbf{x}_n(t_k,\gamma_m))$ be a local minimizer of Eqs. (7) and (8), which satisfies LICQ, strict complementarity slackness (SCS) and second-order sufficient conditions (SOSCs) for Eq. (8). Then $(\mathbf{y}(t_k,\gamma_m)\mathbf{y}_1(t_k,\gamma_m),\ldots,\mathbf{y}_n(t_k,\gamma_m),\mathbf{x}_1(t_k,\gamma_m),\ldots,\mathbf{x}_n(t_k,\gamma_m))$ is a first-order KKT point of Eq. (6) satisfying LICQ.*

If, in addition, SCS and SOSC hold at $(\mathbf{y}(t_k,\gamma_m),\mathbf{y}_1(t_k,\gamma_m),\ldots,\mathbf{y}_n(t_k,\gamma_m),\mathbf{x}_1(t_k,\gamma_m),\ldots,\mathbf{x}_n(t_k,\gamma_m))$ for Eq. (6), then the objective of Eq. (7) $F^*(\mathbf{y})$ is locally twice continuously-differentiable with respect to $\mathbf{y}$ in a neighborhood of $\mathbf{y}(t_k,\gamma_m)$. Also, $\mathbf{y}(t_k,\gamma_m)$ is a minimizer of $F^*(\mathbf{y})$ satisfying SOSC.

The next theorem maps the limit of stationary solutions of the relaxed AIO formulation (Eq. (6)) to a stationary solution of the regularized AIO formulation (Eq. (4)). It follows from well-established results of penalty methods, for example, Theorem 17.1 in Ref. [30] and Theorem 17.2 in Ref. [43].

Theorem 2. *For any t_k, let $\{\mathbf{y}(t_k,\gamma),\mathbf{y}_1(t_k,\gamma),\ldots,\mathbf{y}_n(t_k,\gamma),\mathbf{x}_1,(t_k,\gamma),\ldots,\mathbf{x}_n(t_k,\gamma)_m\}$ be a sequence of KKT points of Eq. (6) corresponding to a sequence of penalty parameters $\{\gamma\}_m$ with $\gamma_m \to \infty$. If LICQ holds for each KKT point in the sequence, then any of its limit point $(\mathbf{y}(t_k),\mathbf{y}_1(t_k),\ldots,\mathbf{y}_n(t_k),\mathbf{x}_1(t_k),\ldots,\mathbf{x}_n(t_k))$, at which LICQ holds for the equality constraints $\mathbf{y}(t_k)=\mathbf{y}_i(t_k)$ and all* *the active local constraints is a KKT point of Eq. (4) satisfying LICQ.*

The following theorem maps the limit of stationary solutions of the regularized AIO formulation (Eq. (4)) to a strongly-stationary solution of the AIO formulation (Eq. (3)). It follows Theorem 3.1 of Ref. [39]. The original theorem derived so-called B-stationarity[1] for Eq. (3), but it also made clear that the B-stationarity is equivalent to strong-stationarity under MPCC-LICQ.

Theorem 3. *Let* $\{\mathbf{y}(t),\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t)\}_k$ *be a sequence of KKT points of Eq. (4) corresponding to a sequence of regularization parameters* $\{t\}_k$ *with* $t_k \rightarrow 0$. *Assume that the sequence converges to* $(\overline{\mathbf{y}},\overline{\mathbf{x}_1},\ldots,\overline{\mathbf{x}_n})$, *which satisfies MPCC-LICQ. Let*

$$I_i^0 = \{j|[\mathbf{G}_i(\mathbf{x}_i(t_k),\mathbf{y}(t_k))]_j[\mathbf{F}_i(\mathbf{x}_i(t_k),\mathbf{y}(t_k))]_j = t_k \quad \textit{for infinitely many}\}$$

$$I_i^G = \{j|[\mathbf{G}_i(\overline{\mathbf{x}_i},\overline{\mathbf{y}})]_j = 0\}$$

$$I_i^F = \{j|[\mathbf{F}_i(\overline{\mathbf{x}_i},\overline{\mathbf{y}})]_j = 0\}$$

Then $(\overline{\mathbf{y}},\overline{\mathbf{x}_1},\ldots,\overline{\mathbf{x}_n})$ *is a strongly-stationary point of Eq. (3) if and only if for each subsystem i, the limits of the Lagrange multipliers corresponding to* $[\mathbf{G}_i]_j$ *and* $[\mathbf{F}_i]_j$, $[\nu_{i,1}]_j$ *and* $[\nu_{i,2}]_j$ *are zero for j such that* $j \in I_i^G \cap I_i^F \cap I_i^0$.

As a summary of the results presented in this subsection, we present the following proposition, which establishes the connection between stationarity conditions of the RIPD problems (Eqs. (5) and (6)) to those of the AIO problem (Eq. (1)).

Proposition 1. *Assume that* f_i, g_i, h_i, G_i, *and* F_i *are three times continuous differentiable. For any* t_k, *let*

(i) $\{\mathbf{y}(t_k,\gamma),\mathbf{y}_1(t_k,\gamma),\ldots,\mathbf{y}_n(t_k,\gamma),\mathbf{x}_1(t_k,\gamma),\ldots,\mathbf{x}_n(t_k,\gamma)_m\}$ *be a sequence of local minimizers of Eqs. (7) and (8) corresponding to a sequence of penalty parameters* $\{\gamma\}_m$ *with* $\gamma_m \rightarrow \infty$ *such that LICQ, SCS, and SOSC are satisfied for Eq. (8) at any point of the sequence.*

(ii) $(\mathbf{y}(t_k),\mathbf{y}_1(t_k),\ldots,\mathbf{y}_n(t_k),\mathbf{x}_1(t_k),\ldots,\mathbf{x}_n(t_k))$ *be a limit point of* $\{\mathbf{y}(t_k,\gamma)\mathbf{y}_1(t_k,\gamma),\ldots,\mathbf{y}_n(t_k,\gamma),\mathbf{x}_1(t_k,\gamma),\ldots,\mathbf{x}_n(t_k,\gamma)\}_m$ *at which LICQ holds for the equality constraints* $\mathbf{y}(t_k,\gamma) = \mathbf{y}_i(t_k,\gamma)$ *and all the active local constraints.*

If the sequence of $(\mathbf{y}(t_k),\mathbf{x}_1(t_k),\ldots,\mathbf{x}_n(t_k))$ *corresponding to a sequence of regularization parameters* $\{t\}_k$ *with* $t_k \rightarrow 0$ *converges to* $(\overline{\mathbf{y}},\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_n)$, *which satisfies the assumptions of Theorem 3, then* $(\overline{\mathbf{y}},\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_n)$ *is a strongly-stationary point of Eq. (3).*

Proof. For any t_k, Theorem 1 indicates that any point in $\{\mathbf{y}(t_k,\gamma)\mathbf{y}_1(t_k,\gamma),\ldots,\mathbf{y}_n(t_k,\gamma),\mathbf{x}_1(t_k,\gamma),\ldots,\mathbf{x}_n(t_k,\gamma)\}_m$ is a KKT point of Eq. (6) satisfying LICQ. Therefore, Theorem 2 guarantees that $(\mathbf{y}(t_k),\mathbf{y}_1(t_k),\ldots,\mathbf{y}_n(t_k),\mathbf{x}_1(t_k),\ldots,\mathbf{x}_n(t_k))$ is a KKT point of Eq. (4) under regularization parameter t_k. Following this result, $(\overline{\mathbf{y}},\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_n)$ is a strongly-stationary point of Eq. (3) due to Theorem 3.

3 Solution Algorithm

In this section, we present a solution algorithm for the MDO-CC following the regularization approach and the IPD method.

The framework of the presented bilevel decomposed algorithm is given as a nested loop shown in Fig. 3: in the outer loop, the regularization approach is followed so that the regularization parameter t_k is gradually reduced; in the inner loop, the IPD algorithm is applied to solve the regularized problem under each t_k setting specified by the outer loop. Specifically, the IPD algorithm increases the penalty parameter γ_m for the deviation terms at the beginning of each inner loop iteration, and solves the master problem (Eq. (7)) under the specified γ_m value. This procedure directly follows Proposition 1. Therefore, it converges to the solution if the assumptions of Proposition 1 are satisfied.

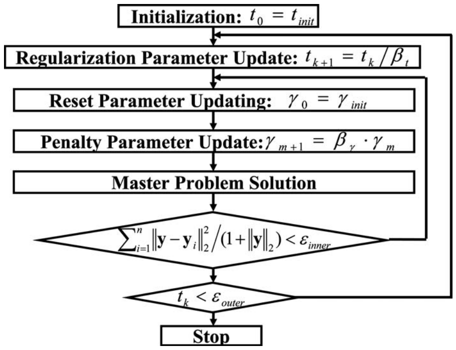

Initialization: $t_0 = t_{init}$
Regularization Parameter Update: $t_{k+1} = t_k / \beta_t$
Reset Parameter Updating: $\gamma_0 = \gamma_{init}$
Penalty Parameter Update: $\gamma_{m+1} = \beta_\gamma \cdot \gamma_m$
Master Problem Solution
$\sum_{i=1}^n \|\mathbf{y}-\mathbf{y}_i\|_2^2 / (1+\|\mathbf{y}\|_2) < \varepsilon_{inner}$
$t_k < \varepsilon_{outer}$
Stop

Fig. 3 The nested loop framework of the regularized IPD algorithm

While supported theoretically by Proposition 1, the nested loop framework may incur intensive computation due to costly inner loop iterations. In order to address this, an alternative strategy could be to employ looser inner loop termination criteria. Several approaches has been proposed to reduce the number of inner loop iterations in the context of MDO: Tosserams et al. [44] proposed to follow alternate direction methods of multipliers to reduce the number of inner loop iteration in ATC approach. Additionally, Li et al. [45] presented a truncated diagonal quadratic approximation of the ATC, which is closely related to Tosserams' method. Note that the nested loop computation framework in these papers refers to the decomposition algorithm for MDO problems, which is actually the inner loop of our presented nested loop framework. As an extreme case of reducing the inner loop iterations, we present a single loop procedure in which the inner loop is terminated after just a single RIPD problem solution. In other words, the regularization parameter and the penalty parameter are updated together. The analytical convergence property of the presented single loop procedure is not yet established. However, numerical computation shows promising results with convergence to AIO solutions. The procedure of the single loop framework is presented in Fig. 4; and the results are presented in Sec. 4.

Following the RIPD formulation presented in Sec. 2, the master problem is given as an unconstrained optimization problem. In general, the master objective of a nested bilevel program is nonsmooth, while in the RIPD formulation, Theorem 1 shows that the master objective function $F^*(\mathbf{y})$ is locally twice continuously-differentiable with respect to $\mathbf{y}$. Therefore, the master problem can be solved by algorithms utilizing derivatives. In practice, we solve the master problem following a modified Broyden-Fletcher-

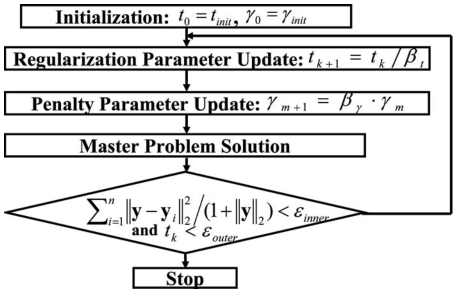

Initialization: $t_0 = t_{init}$, $\gamma_0 = \gamma_{init}$
Regularization Parameter Update: $t_{k+1} = t_k / \beta_t$
Penalty Parameter Update: $\gamma_{m+1} = \beta_\gamma \cdot \gamma_m$
Master Problem Solution
$\sum_{i=1}^n \|\mathbf{y}-\mathbf{y}_i\|_2^2 / (1+\|\mathbf{y}\|_2) < \varepsilon_{inner}$ and $t_k < \varepsilon_{outer}$
Stop

Fig. 4 The single loop framework of the regularized IPD algorithm

[1] A feasible point $\mathbf{x}$ of an MPCC is Bouligard stationary (B-stationary) if it is a local minimizer of the linearized MPCC, which is obtained by linearizing all data functions at point $\mathbf{x}$ [40].

```
Modified BFGS algorithm
  BEGIN
    Step 0: Given $\mathbf{x}_0$, convergence tolerance $\varepsilon_{grad}$ and initial Hessian
            approximation $B_k$ ; $k = 0$
    Step1: REPEAT
            Step 1.1: Evaluate function $F^*(\mathbf{y})$ and gradient$\nabla F^*(\mathbf{y})$
            Step 1.2: Calculate search direction: solve $B_k \Delta \mathbf{y}_k = -\nabla F^*(\mathbf{y}_k)$
            Step 1.3: Back tracking line search: $\alpha_k = 1$
                WHILE $F^*(\mathbf{y}_k + \alpha_k \Delta \mathbf{y}_k) - F^*(\mathbf{y}_k) > \sigma \nabla F^*(\mathbf{y}_k)^T \Delta \mathbf{y}_k$
                    $\alpha_k = \alpha_k / 2$ ; Calculate $F^*(\mathbf{y}_k + \alpha_k \Delta \mathbf{y}_k)$
                ENDWHILE
            Step 1.4: $\mathbf{s}_k = \alpha_k \Delta \mathbf{y}_k$, $\phi_k = \nabla F^*(\mathbf{y}_{k+1}) - \nabla F^*(\mathbf{y}_k)$,
                      $\mathbf{r}_k = \phi_k + \lambda_k \|\nabla F^*(\mathbf{y}_k)\|_2 \mathbf{s}_k$, $\mathbf{y}_{k+1} = \mathbf{y}_k + \alpha_k \Delta \mathbf{y}_k$,
                      $\lambda_k = 1 + \max\left\{ -\frac{\phi_k^T s_k}{\|\nabla F^*(\mathbf{y}_k)\|_2 \|s_k\|_2^2}, 0 \right\}$,
            Step 1.5: BFGS Hessian approximation update:
                      $B_{k+1} = B_k - \frac{B_k \mathbf{s}_k \mathbf{s}_k^T B_k^T}{\mathbf{s}_k^T B_k \mathbf{s}_k} + \frac{\mathbf{r}_k \mathbf{r}_k^T}{\mathbf{r}_k^T \mathbf{s}_k}$
                      $k = k + 1$
    UNTIL $\|\nabla F(\mathbf{y}_k)\| / |1 + F(\mathbf{y}_k)| < \varepsilon_{grad}$
  END
```

Fig. 5 The BFGS procedure

Goldfarb-Shanno (BFGS) method with backtracking line search [46], which builds an approximation of $F^*(\mathbf{y})$'s second derivative using its first gradient. A pseudocode for the algorithm is provided for the reader in Fig. 5; a numerical example is also included in the Appendix to illustrate the approach.

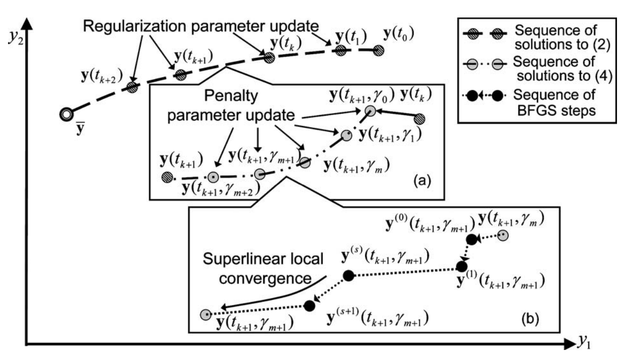
Note that the gradient of $F^*(\mathbf{y})$ is calculated with [27]

$$\nabla F^*(\mathbf{y}) = 2\gamma_m \sum_{i=1}^{n} (\mathbf{y} - \mathbf{y}_i) \tag{12}$$

Although the gradient in Eq. (12) looks as if it is derived through directly differentiating $F^*(\mathbf{y})$ with respect to $\mathbf{y}$, it actually considers the implicit functions such as $\mathbf{y}_i(\mathbf{y})$ and $\mathbf{x}_i(\mathbf{y})$, etc. The reader is referred to Ref. [27] for its detailed derivation.

3.1 The Convergence Procedure and Local Convergence Analysis. The convergence procedure of the nested loop framework is illustrated in Figs. 6(*a*) and 6(*b*). For demonstration purposes, we consider a simple example with two linking variables $\mathbf{y}=[y_1, y_2]^T$ and present the convergence procedure of the linking variables only.

In Figs. 6(*a*) and 6(*b*), the shaded dots, denoted as $\{\mathbf{y}(t_0), \mathbf{y}(t_1), \ldots, \mathbf{y}(t_k), \ldots\}$, represent a sequence of stationary solutions to Eq. (4), corresponding to $\{t\}_k$. For each adjacent pair of points in this sequence $\mathbf{y}(t_k)$ and $\mathbf{y}(t_{k+1})$, the inner loop takes a sequence of stationary solutions to Eq. (6), corresponding to $\{\gamma\}_m$, to move from $\mathbf{y}(t_k)$ to $\mathbf{y}(t_{k+1})$. This sequence, denoted as $\{\mathbf{y}(t_{k+1}, \gamma_0), \mathbf{y}(t_{k+1}, \gamma_1), \ldots, \mathbf{y}(t_{k+1}, \gamma_m), \ldots\}$, is illustrated by the gray dots in Fig. 6(*a*).



y_2
Regularization parameter update
$\mathbf{y}(t_{k+2})$
$\mathbf{y}(t_{k+1})$
$\mathbf{y}(t_k)$
$\mathbf{y}(t_1)$
$\mathbf{y}(t_0)$
$\bar{\mathbf{y}}$
Sequence of solutions to (2)
Sequence of solutions to (4)
Sequence of BFGS steps
Penalty parameter update
$\mathbf{y}(t_{k+1}, \gamma_0)$ $\mathbf{y}(t_k)$
$\mathbf{y}(t_{k+1}, \gamma_1)$
$\mathbf{y}(t_{k+1})$
$\mathbf{y}(t_{k+1}, \gamma_{m+1})$
$\mathbf{y}(t_{k+1}, \gamma_m)$
$\mathbf{y}(t_{k+1}, \gamma_{m+2})$
(a)
$\mathbf{y}^{(0)}(t_{k+1}, \gamma_{m+1})$ $\mathbf{y}(t_{k+1}, \gamma_m)$
Superlinear local convergence
$\mathbf{y}^{(s)}(t_{k+1}, \gamma_{m+1})$
$\mathbf{y}^{(1)}(t_{k+1}, \gamma_{m+1})$
$\mathbf{y}(t_{k+1}, \gamma_{m+1})$
$\mathbf{y}^{(s+1)}(t_{k+1}, \gamma_{m+1})$
(b)
y_1

Fig. 6 An illustration of the convergence procedure (*a* and *b*)

Furthermore, for each adjacent pair of points in the sequence of stationary solutions to Eq. (6) $\mathbf{y}(t_{k+1}, \gamma_m)$ and $\mathbf{y}(t_{k+1}, \gamma_{m+1})$, the BFGS algorithm takes a sequence of steps to converge from $\mathbf{y}(t_{k+1}, \gamma_m)$ to $\mathbf{y}(t_{k+1}, \gamma_{m+1})$. This sequence, denoted as $\{\mathbf{y}^{(0)} \times (t_{k+1}, \gamma_{m+1}), \mathbf{y}^{(1)}(t_{k+1}, \gamma_{m+1}), \ldots, \mathbf{y}^{(s)}(t_{k+1}, \gamma_{m+1}), \ldots\}$, is illustrated by the solid dots in Fig. 6(*b*).

As discussed previously, we employ the BFGS method and the SQP as the master and subproblem optimizer respectively. Following this setting, standard local convergence property can be expected. For any t_k, we note that given certain assumptions, the implicit function theorem and Theorem 6 of Ref. [43] indicate that there exists a locally unique twice continuously-differentiable trajectory of stationary solutions to Eq. (6) $\mathbf{y}(t_k, \gamma)$, which converges to $\mathbf{y}(t_k)$. The following theorem shows that for each parameter setting (regularization and penalty parameter combined), the BFGS master problem solver converges locally at a superlinear rate to a stationary solution to Eq. (6). Also presented is a restatement of the superlinear local convergence of the SQP solver for each subproblem.

Theorem 4. *Assume that the assumptions of Proposition 1 hold. If for any* t_k, $(\mathbf{y}(t_k), \mathbf{x}_1(t_k), \ldots, \mathbf{x}_n(t_k))$ *satisfies LICQ, SCS, and SOSC, then there exists* $\bar{\gamma}(t_k) > 0$ *satisfying for each* $\gamma > \bar{\gamma}(t_k)$

1. *There exists* $\varepsilon_1(t_k) > 0$ *such that the BFGS iterates,* $\{\mathbf{y}^{(0)} \times (t_k, \gamma), \mathbf{y}^{(1)}(t_k, \gamma), \ldots, \mathbf{y}^{(s)}(t_k, \gamma), \ldots\}$, *converge locally and superlinearly to* $\mathbf{y}(t_k, \gamma)$, *if* $\|\mathbf{y}^{(0)}(t_k, \gamma) - \mathbf{y}(t_k, \gamma)\| \leq \varepsilon_1(t_k)$.
2. *Assume that* $(\mathbf{x}_i(\mathbf{y}^{(s)}(t_k, \gamma)), \mathbf{y}_i(\mathbf{y}^{(s)}(t_k, \gamma))$ *is a KKT solution satisfying the LICQ and SOSC corresponding to* $\mathbf{y}^{(s)}(t_k, \gamma)$, *there exists* $\varepsilon_2(t_k) > 0$ *such that the SQP iterates converge locally and superlinearly to* $(\mathbf{x}_i(\mathbf{y}^{(s)}(t_k, \gamma)), \mathbf{y}_i(\mathbf{y}^{(s)}(t_k, \gamma))$ *, if the SQP starts sufficiently close to* $(\mathbf{x}_i(\mathbf{y}^{(s)}(t_k, \gamma)), \mathbf{y}_i(\mathbf{y}^{(s)} \times (t_k, \gamma))$ *, and* $\|\mathbf{y}^{(s)}(t_k, \gamma) - \mathbf{y}(t_k, \gamma)\| \leq \varepsilon_2(t_k,)$.

4 Numerical Results

In this section, we present a numerical study to illustrate the proposed regularized inexact penalty decomposition algorithm. Two classes of problems are tested: a variant of the Golinski's problem, and SQPECgen, a problem set of quasi-separable quadratic programs with complementarity constraints (QPCCs). In addition, another simple numerical example is presented in the Appendix to provide step-by-step demonstration of the proposed approach.

4.1 A Variant of Golinski's Problem. Golinski's speed reducer design problem [47] is originally presented as a regular MDO problem. Its AIO formulation is given as follows:

$$\begin{aligned}
\text{Gol}_{\text{AIO}}\text{: } \min_{x_1, \ldots, x_7} \; f(x_1, \ldots, x_7) = {} & 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 \\
& - 43.0934) - 1.5079 x_1 (x_6^2 + x_7^2) \\
& + 7.477 (x_6^3 + x_7^3) + 0.7854 (x_4 x_6^2 + x_5 x_7^2)
\end{aligned}$$

$$\begin{aligned}
\text{subject to} \quad & g_1 = \frac{1}{110 x_6^3} \sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 1.69 \cdot 10^7} - 1 \leq 0 \\
& g_2 = \frac{1}{87 x_7^3} \sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 1.575 \cdot 10^8} 1 \leq 0 \\
& g_3 = \frac{1.5 x_6 + 1.9}{x_4} - 1 \leq 0, \quad g_4 = \frac{1.1 x_7 + 1.9}{x_5} - 1 \leq 0, \quad g_5 \\
& = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0 \\
& g_6 = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0, \quad g_7 = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \leq 0, \quad g_8 = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \leq 0
\end{aligned} \tag{13}$$

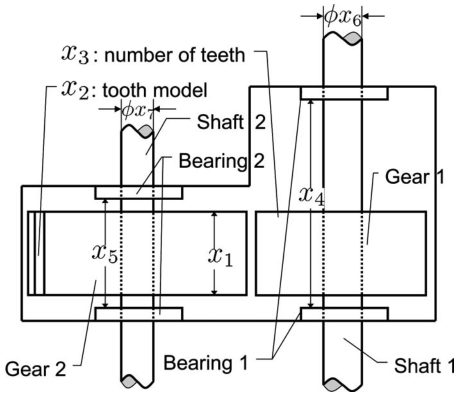


Fig. 7 A schematic of the Golinski's speed reducer

$$g_9 = \frac{x_2 x_3}{40} - 1 \le 0, \quad g_{10} = \frac{5x_2}{x_1} - 1 \le 0, \quad g_{11} = \frac{x_1}{12x_2} - 1 \le 0$$

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le x_2 \le 0.8, \quad 17 \le x_3 \le 28, \quad 7.3 \le x_4 \le 8.3$$

$$7.3 \le x_5 \le 8.3, \quad 2.9 \le x_6 \le 3.9, \quad 5.0 \le x_7 \le 5.5$$

The objective of this problem is to minimize the volume of a speed reducer subjected to stress, deflection, and geometric constraints. The design variables are the dimensions of the gear itself (x_1, x_2, x_3) and both the shafts (x_4, x_6, and x_5, x_7). Note that x_3 is originally presented as an integer variable. However, it is relaxed as a continuous variable here for demonstration purpose. A schematic of the speed reducer is presented in Fig. 7 with its design variables labeled.

While the Golinski's problem was not originally presented with complementarity constraints, one of its variant is employed here for demonstration purpose. We note that g_1 and g_2 in Eq. (13) specifies the maximum torsional stresses in both shafts. In this study, two additional constraints g_{12} and g_{13} are included to represent the maximum acceptable strain energies in the shafts. The two constraints are given as follows:

$$g_{12} = \frac{x_4}{7.93128 \times 10^5 x_6^4}\left(\left(745\frac{x_4}{x_2 x_3}\right)^2 + 1.69 \times 10^7\right) - 1 \le 0 \tag{14}$$

$$g_{13} = \frac{x_5}{1.91724 \times 10^6 x_7^4}\left(\left(745\frac{x_5}{x_2 x_3}\right)^2 + 1.69 \times 10^7\right) - 1 \le 0 \tag{15}$$

We set the maximum stress constraints and the maximum strain energy constraints complementary to each other for each shaft (as shown in Eq. (16)), thereby introducing complementarity

$$\begin{aligned} 0 \le -g_1 \perp -g_{12} \ge 0 \\ 0 \le -g_2 \perp -g_{13} \ge 0 \end{aligned} \tag{16}$$

By introducing local copies of the linking variables (x_1, x_2, x_3), which are labeled with superscripts (1), (2), and (3) in the three subproblems, respectively, the original AIO problem can be decomposed into the RIPD formulation consisting the following master (Eq. (17)) and subproblems (Eqs. (18)–(20)):

$$\text{Gol}_{\text{master}}: \min_{x_1^U, x_2^U, x_3^U} F^*(x_1^U, x_2^U, x_3^U) = \sum_{i=1}^{n} F_i^*(x_1^U, x_2^U, x_3^U) \tag{17}$$

$$\begin{aligned} F_1^*(x_1^U, x_2^U, x_3^U) = \min_{x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, x_4, x_6} & -1.5079 x_1^{(1)} x_6^2 + 7.477 x_6^3 \\ & + 0.7854 x_4 x_6^2 + \gamma_m \sum_{i=1}^{3} (x_1^U - x_i^{(1)})^2 \\ \text{subject to} \quad & [g_1, g_3, g_7, g_{12}]^T \le 0, \quad g_1 g_{12} \le t_k \end{aligned} \tag{18}$$

$$\begin{aligned} F_2^*(x_1^U, x_2^U, x_3^U) = \min_{x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, x_5, x_7} & -1.5079 x_1^{(2)} x_7^2 + 7.477 x_7^3 \\ & + 0.7854 x_5 x_7^2 + \gamma_m \sum_{i=1}^{3} (x_i^U - x_i^{(2)})^2 \\ \text{subject to} \quad & [g_2, g_4, g_8, g_{13}]^T \le 0, \quad g_2 g_{13} \le t_k \end{aligned} \tag{19}$$

$$\begin{aligned} F_3^*(x_1^U, x_2^U, x_3^U) = \min_{x_1^{(3)}, x_2^{(3)}, x_3^{(3)}} & 0.7854 x_1^{(3)} (x_2^{(3)})^2 (3.3333 (x_3^{(3)})^2 \\ & + 14.9334 x_3^{(3)} - 43.0934) + \gamma_m \sum_{i=1}^{3} (x_i^U - x_i^{(3)})^2 \\ \text{subject to} \quad & [g_5, g_6, g_9, g_{10}, g_{11}]^T \le 0 \end{aligned} \tag{20}$$

We applied the proposed RIPD algorithm to the MDO-CC variant of Golinski's problem. Both the nested loop and the single loop framework are tested. The numerical results are presented in Table 1 with a comparison between the RIPD solutions and the numerical AIO solution. It can be noted from Table 1 that both the nested loop RIPD approach and the single loop RIPD approach obtained solutions identical to the numerical AIO solution to MDO-CC variant of the Golinski's problem. All of the three solutions are generated with KNITRO® solver in the MATLAB® environment. The corresponding parameter settings are $t_0 = 0.05$, $\beta_t = 8$, $\gamma_0 = 1$, $\beta_\gamma = 2$, $\varepsilon_{\text{inner}} = 1 \times 10^{-6}$, $\varepsilon_{\text{outer}} = 1 \times 10^{-6}$, and $\varepsilon_{\text{grad}} = 5 \times 10^{-4}$. The active constraints at the solution are g_2, g_4, g_{10}, and g_{12}.

4.2 SQPECgen: A Test Problem Set for Quasi-Separable Quadratic Programs With Complementarity Constraints.

SQPECgen [37] is a quasi-separable QPCC test problem set based on a QPCC generator proposed by Jiang and Ralph [48]. It features quasi-separable structure, which differentiates linking variables from local variables. Specifically, the problems it generates possess a quadratic objective function with polyhedral *first level* (relevant to linking variables only) constraints and complementarity *second level* (relevant to both linking and local variables) constraints, as shown below

Table 1 Numerical results of the decomposition approach for the MDO-CC: Golinski's

Approach	Initial solution	Final solution
AIO	$[3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5]^T$	$[3.500000, 0.700000, 17.000000, 7.300000, 7.670397, 3.542421, 5.245814]^T$
Nested loop RIPD	$[3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5]^T$	$[3.500013, 0.700004, 17.000000, 7.300000, 7.670396, 3.542421, 5.245814]^T$
Single loop RIPD	$[3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5]^T$	$[3.500029, 0.700005, 17.000000, 7.300000, 7.670396, 3.542421, 5.245814]^T$

Table 2 Numerical results of the nested loop RIPD approach: SQPECgen with five subsystems

Case No.	Number of function evaluations
1	1.72×10^6
2	2.33×10^6
3	1.12×10^6
4	3.60×10^6
5	4.26×10^6

$$\text{SQPECgen:}\quad \min_{\mathbf{z}=[\mathbf{y}^T,\mathbf{x}_1^T,\ldots,\mathbf{x}_n^T]^T} \frac{1}{2}\mathbf{z}^T\begin{pmatrix} P_{\mathbf{yy}} & P_{\mathbf{yx}_1} & \cdots & P_{\mathbf{yx}_n} \\ P_{\mathbf{yx}_1}^T & P_{\mathbf{x}_1\mathbf{x}_1} & \cdots & \\ \vdots & & \ddots & \\ P_{\mathbf{yx}_n}^T & & & P_{\mathbf{x}_n\mathbf{x}_n}\end{pmatrix}\mathbf{z} + \begin{pmatrix}\mathbf{c}\\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_n\end{pmatrix}^T\mathbf{z}$$

$$\text{subject to}\quad G\mathbf{y}\le\mathbf{a} \tag{21}$$

$$\mathbf{0}\le\mathbf{x}_i\perp N_i\mathbf{y}+M_i\mathbf{x}_i+\mathbf{q}_i\ge\mathbf{0},\qquad\forall\, i=1,\ldots,n$$

Following the technique presented in Sec. 2, the decomposition formulations of the above QPCC are:

$$\text{SQPECgen}_{\text{RIPD}}\ \min_{\mathbf{y}}\quad F^*(\mathbf{y})=\sum_{i=1}^{n+1}F_i^*(\mathbf{y}) \tag{22}$$

$$F_i^*(\mathbf{y})=\min_{\mathbf{z}_i=[(\mathbf{y}_i)^T,\mathbf{x}_i^T]^T}\frac{1}{2}\mathbf{z}_i^T\begin{pmatrix}\mathbf{0} & P_{\mathbf{yx}_i}\\ P_{\mathbf{yx}_i}^T & P_{\mathbf{x}_i\mathbf{x}_i}\end{pmatrix}\mathbf{z}_i+\begin{pmatrix}\mathbf{0}\\ \mathbf{d}_i\end{pmatrix}^T\mathbf{z}_i+\gamma_m\|\mathbf{y}-\mathbf{y}_i\|_2^2$$

$$\text{subject to}\quad \mathbf{x}_i\ge\mathbf{0},\quad N_i\mathbf{y}_i^L+M_i\mathbf{x}_i+\mathbf{q}_i\ge\mathbf{0} \tag{23}$$

$$[\mathbf{x}_i]_j[N_i\mathbf{y}_i^L+M_i\mathbf{x}_i+\mathbf{q}_i]_j\le t_k$$

$$F_{n+1}^*(\mathbf{y})=\min_{\mathbf{y}_{n+1}}\frac{1}{2}(\mathbf{y}_{n+1})^TP_{\mathbf{yy}}\mathbf{y}_{n+1}+\mathbf{c}^T\mathbf{y}_{n+1}+\gamma_m\|\mathbf{y}-\mathbf{y}_i\|_2^2 \tag{24}$$

$$\text{subject to}\quad G\mathbf{y}_{n+1}\le\mathbf{a}$$

4.2.1 The Nested Loop Framework. In order to demonstrate the effectiveness of the proposed nested loop framework (Fig. 3), we have tested several QPCC problems from the SQPECgen problem set (Eq. (21)). In this study, the nested loop framework is followed to solve five test problems, each of which has five linking variables and five subsystems. Additionally, each of the subsystem has five local variables.

Table 2 presents the numerical results obtained through the RIPD algorithms. Each number in the table is the average number of function evaluations needed to converge from five random initial solutions. We note that for each initial solution the nested loop framework converged to a solution identical to a numerical AIO solution for all the problems tested. All of these solutions are generated with KNITRO® 5.0 solver in MATLAB® 7.2. The corresponding parameter settings are $t_0=0.05$, $\beta_t=4$, $\gamma_0=1$, $\beta_\gamma=2$, $\varepsilon_{\text{inner}}=1\times10^{-6}$, $\varepsilon_{\text{outer}}=5\times10^{-5}$, and $\varepsilon_{\text{grad}}=5\times10^{-3}$.

4.2.2 The Single Loop Framework. Although our numerical results show that the nested loop framework converges to AIO solutions, it usually incurs intensive computation due to excessive inner loop iterations, as shown in Table 2. In this subsection, we demonstrate the performance of the more practical single loop framework (Fig. 4) with a set of quasi-separable QPCC problems with varying number of subsystems. For each of the test problems, the single loop RIPD converged to a solution identical to a numerical AIO solution from five randomly generated initial solutions. The corresponding number of function evaluations (the average of two test problems with the same number of subsystems, five runs for each test problem) is plotted in Fig. 8. Additionally, the same metric corresponding to the ALD method [7], an existing approach, is also presented for a comparison purpose. Both algorithms converge to the same solution for each quasi-separable QPCC in the test problem set.

In Fig. 8, each data dot represents the average number of function evaluations for two test problems under the same subsystem settings. Here, each test problem has five linking variables, and each subsystem has five local variables. As shown in Fig. 8, the single loop RIPD approach takes approximately 14 times as many function evaluations to solve the 40-subsystem case as it takes for the five-subsystem case. The increase in computation cost results from both the increase in subsystem number and the increase in major iterations needed to generate a consistent solution. This ratio is approximately 75 for the case of ALD algorithm.

Both algorithms in this study are implemented with KNITRO® solver in the MATLAB® environment. The corresponding parameter for the five- and ten-subsystem cases are $t_0=0.05$, $\beta_t=4$, $\gamma_0=1$, $\beta_\gamma=2$, $\varepsilon_{\text{inner}}=1\times10^{-6}$, $\varepsilon_{\text{outer}}=1\times10^{-5}$, and $\varepsilon_{\text{grad}}=1\times10^{-3}$; for the 20–40-subsystem case $t_0=0.05$, $\beta_t=3$, $\gamma_0=1$, $\beta_\gamma=1.5$, $\varepsilon_{\text{inner}}=1\times10^{-6}$, $\varepsilon_{\text{outer}}=1\times10^{-5}$, and $\varepsilon_{\text{grad}}=1\times10^{-3}$. Additionally, the deviation tolerance of the ALD is set as 1×10^{-6}, the same value as $\varepsilon_{\text{inner}}$ in RIPD.

In addition to SQPECgen problems with varying number of subsystem, we have also tested the single loop framework on a set of SQPECgen problems with fixed number of linking variables and varying number local variables. Specifically, each problem tested has two linking variables and ten subsystems with the number of local variables per subsystem ranging from 30 to 50. The

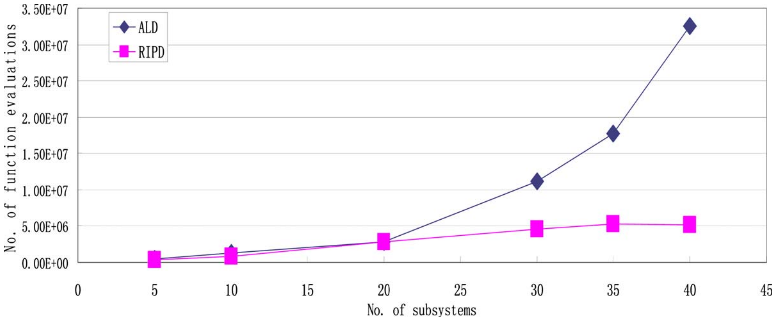


Fig. 8 Numerical behavior of the RIPD approach compared with the augmented Lagrangian decomposition [7]

Table 3 Numerical results of the single loop RIPD approach: SQPECgen with varying number of local variables per subsystem

Number of local variables per subsystem	30	40	50
Number of function evaluations: RIPD	6.28×10^5	7.05×10^5	8.44×10^5
Computation time: RIPD(s)	3.02×10^3	4.36×10^3	5.14×10^3
Number of function evaluations: regularized AIO	0.2×10^5	2.50×10^5	5.57×10^5
Computation time: regularized AIO(s)	0.83×10^3	9.25×10^3	17.77×10^3

numerical results are presented such that the computational cost of the single loop RIPD is compared with that of the regularized AIO approach to show the effect of decomposition.

Table 3 shows the numerical results obtained through both the RIPD and the regularized AIO approach. The numbers presented are the average numbers of function evaluations and computation times of five runs, for each of which the two algorithms obtained a same solution from a randomly generated initial solution. According to numerical results, the computation time of the RIPD algorithm is approximately four times as much as that of the regularized AIO approach for the 30-local variable case; while the former is approximately a third of the latter for the 50-local variable case. This indicates that the computational effect of decomposition is more remarkable when the coupling among the subsystems are looser. Additionally, we notice in Table 3 that function evaluations of the regularized AIO problem take more time than those of the RIPD subproblems due to the difference in dimensionality. Note that the RIPD computation time is measured under serial implementation, i.e., the computation time is the summation of all the subsystem computation time. If the RIPD is implemented in parallel, the computation time will be much shorter.

Both algorithms in this study are implemented with KNITRO® solver in the MATLAB® environment. The corresponding parameter setting is t_0=0.05, β_t=3, γ_0=1, β_γ=1.5, $\varepsilon_{\text{inner}}=1\times10^{-6}$, $\varepsilon_{\text{outer}}=1\times10^{-5}$, and $\varepsilon_{\text{grad}}=1\times10^{-3}$. The same t_0, β_t, and $\varepsilon_{\text{outer}}$ are applied to the regularized AIO approach.

4.2.3 Recommendation for Parameter Selection. Finally, we provide some recommendations for the selection of β_t and β_γ. According to our numerical experience, both the nested loop and the single loop framework converge with a $\beta_t \in (1,10]$. For the SQPECgen test problems, $2 \le \beta_t \le 4$ is recommended to speedup convergence.

Additionally, our numerical experience indicates that the nested loop framework converges with a $\beta_\gamma \in (1,4]$, while $2 \le \beta_\gamma \le 3$ is recommended to speedup convergence. On the other hand, the single loop framework appears to be more sensitive to β_γ setting. A $\beta_\gamma \in (1,2.5]$ worked for all the problems we tested; and $1 \le \beta_\gamma \le 2$ is recommended for convergence speed.

5 Conclusion and Future Work

The presence of complementarity constraints in a MDO problem poses a numerical challenge, which existing MDO approaches usually could not handle. This paper presents a nested decomposition formulation for the MDO-CC based on regularization and inexact penalty decomposition technique. As an important contribution, we showed that existing theories could be adapted to map a limit point of stationary solutions of the parameterized decomposition formulation to a strongly-stationary solution of the AIO formulation. Following this result, a solution algorithms for the MDO-CC were proposed with potential implementations of a nested loop framework and a single loop framework. Additionally, we showed that superlinear convergence rate could be expected for the proposed algorithm following the local convergence results of standard master and subproblem solvers. Numerical results showed that both the two presented algorithms converge to solutions identical to the AIO solutions.

Future work includes an in-depth study regarding the differences between nested loop versus single loop solution process. Currently, the numerical results show that the single loop cases converge to AIO solutions, which is identical to the nested loop case. An extensive study will be useful in terms of comparing efficiency and robustness (from ill-conditioning) of solution process, which can provide further guidelines for setting β coefficients.

Acknowledgment

This work is supported by the National Science Foundation under Grant No. 0726934 and by Caterpillar, Inc. under Project No. 2007-03101. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the National Science Foundation and Caterpillar, Inc. The authors acknowledge Uday Shanbhag for the suggestions in the initial stage.

Appendix: Illustrative Example

A simple numerical example is provided in this Appendix, for illustrative purpose. The example problem is a quasi-separable QPCC with two linking variables and two subsystems with two local variables each. The formulation of the problem is given as follows:

$$\min_{\mathbf{z}=[\mathbf{y}^T,\mathbf{x}_1^T,\mathbf{x}_2^T]^T} \frac{1}{2}\mathbf{z}^T \begin{pmatrix} 2.0009 & -0.3070 & -0.0916 & 0.6246 & -0.0916 & 0.6246 \\ -0.3070 & 0.8408 & -0.7418 & -0.7411 & -0.7418 & -0.7411 \\ -0.0916 & -0.7418 & 2.6815 & -0.1375 & 0 & 0 \\ 0.6246 & -0.7411 & -0.1375 & 2.6477 & 0 & 0 \\ -0.0916 & -0.7418 & 0 & 0 & 2.6815 & -0.1375 \\ 0.6246 & -0.7411 & 0 & 0 & -0.1375 & 2.6477 \end{pmatrix} \mathbf{z} + \begin{pmatrix} -1.4486 \\ 2.9529 \\ 1.4343 \\ -0.8599 \\ -2.2608 \\ -3.5206 \end{pmatrix}^T \mathbf{z}$$

$$\text{subject to}\quad (0.1243-0.1101)\mathbf{y}\le 0.2481 \tag{A1}$$

$$\mathbf{0}\le \mathbf{x}_1 \perp \begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y}+\begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_1 + \begin{pmatrix}-5.9478\\ -3.3817\end{pmatrix}\ge \mathbf{0}$$

$$\mathbf{0}\le \mathbf{x}_2 \perp \begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y}+\begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_2 + \begin{pmatrix}-2.8053\\ -1.6865\end{pmatrix}\ge \mathbf{0}$$

Following the technique presented in Sec. 2, the decomposition formulations of the above QPCC are:

$$\min_{\mathbf{y}} F^*(\mathbf{y})=\sum_{i=1}^{n+1}F_i^*(\mathbf{y}) \tag{A2}$$

$$F_1^*(\mathbf{y}) = \min_{\mathbf{z}_1=[\mathbf{y}_1^T,\mathbf{x}_1^T]^T} \tfrac{1}{2}\mathbf{z}_1^T\begin{pmatrix}0 & 0 & -0.0916 & 0.6246\\ 0 & 0 & -0.7418 & -0.7411\\ -0.0916 & -0.7418 & 2.6815 & -0.1375\\ 0.6246 & -0.7411 & -0.1375 & 2.6477\end{pmatrix}\mathbf{z}_1 + \begin{pmatrix}0\\ 0\\ 1.4343\\ -0.8599\end{pmatrix}^T\mathbf{z}_1+\gamma_m\|\mathbf{y}-\mathbf{y}_1\|_2^2$$

$$\text{subject to}\quad \mathbf{x}_1\ge\mathbf{0},\begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y} + \begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_1+\begin{pmatrix}-5.9478\\ -3.3817\end{pmatrix}\ge\mathbf{0} \tag{A3}$$

$$\operatorname{diag}(\mathbf{x}_1)\left(\begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y}+\begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_1 + \begin{pmatrix}-5.9478\\ -3.3817\end{pmatrix}\right)\le\begin{pmatrix}t_k\\ t_k\end{pmatrix}$$

$$F_2^*(\mathbf{y}) = \min_{\mathbf{z}_1=[\mathbf{y}_2^T,\mathbf{x}_2^T]^T} \tfrac{1}{2}\mathbf{z}_2^T\begin{pmatrix}0 & 0 & -0.0916 & 0.6246\\ 0 & 0 & -0.7418 & -0.7411\\ -0.0916 & -0.7418 & 2.6815 & -0.1375\\ 0.6246 & -0.7411 & -0.1375 & 2.6477\end{pmatrix}\mathbf{z}_1 + \begin{pmatrix}0\\ 0\\ -2.2608\\ -3.5206\end{pmatrix}^T\mathbf{z}_2+\gamma_m\|\mathbf{y}-\mathbf{y}_2\|_2^2$$

$$\text{subject to}\quad \mathbf{x}_2\ge\mathbf{0},\begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y} + \begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_2+\begin{pmatrix}-5.9478\\ -3.3817\end{pmatrix}\ge\mathbf{0} \tag{A4}$$

$$\operatorname{diag}(\mathbf{x}_2)\left(\begin{pmatrix}-0.0136 & -0.0067\\ -0.5144 & -0.6580\end{pmatrix}\mathbf{y}+\begin{pmatrix}4.2209 & 2.2445\\ 2.2445 & 1.5515\end{pmatrix}\mathbf{x}_2 + \begin{pmatrix}-2.8053\\ -1.6865\end{pmatrix}\right)\le\begin{pmatrix}t_k\\ t_k\end{pmatrix}$$

$$F_3^*(\mathbf{y})=\min_{\mathbf{y}_3}\tfrac{1}{2}(\mathbf{y}_3)^T\begin{pmatrix}2.0009 & -0.3070\\ -0.3070 & 0.8408\end{pmatrix}\mathbf{y}_3+\begin{pmatrix}-1.4486\\ 2.9529\end{pmatrix}\mathbf{y}_3 + \gamma_m\|\mathbf{y}-\mathbf{y}_3\|_2^2$$

$$\text{subject to}\quad \begin{pmatrix}0.1243 & -0.1101\end{pmatrix}\mathbf{y}_3\le 0.2481 \tag{A5}$$

We take an initial solution of $\mathbf{z}^T=(-2,2,0,0,0,0)$, with an initial t_1 of 0.05 and an initial γ_1 of 1. Following step 1.1 of Fig. 5, we start with solving the three subsystems (Eqs. (A3)–(A5)), respectively, which yields:

$$\begin{pmatrix}\mathbf{y}_1^{(0)}(t_1,\gamma_1)\\ \mathbf{x}_1^{(0)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-2.6223\\ 2.7791\\ 0.4257\\ 1.8731\end{pmatrix},\quad \begin{pmatrix}\mathbf{y}_2^{(0)}(t_1,\gamma_1)\\ \mathbf{x}_2^{(0)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-2.0992\\ 3.2007\\ 0.0414\\ 1.7076\end{pmatrix},\quad \mathbf{y}_3^{(0)}(t_1,\gamma_1)=\begin{pmatrix}-0.6145\\ 0.3021\end{pmatrix}$$

Following Eqs. (A2) and (12), the master objective and its gradient are:

$$F^*(\mathbf{y}^{(0)}(t_1,\gamma_1))=-3.0511,\nabla F^*(\mathbf{y}^{(0)}(t_1,\gamma_1))=\begin{pmatrix}-1.3280\\ -0.5640\end{pmatrix}$$

With the initial Hessian approximation B_0 as the identity matrix, this generates a descent search direction (step 1.2 of Fig. 5)

$$\Delta\mathbf{y}^{(0)}(t_1,\gamma_1)=\begin{pmatrix}1.3280\\ 0.5640\end{pmatrix}$$

Following step 1.3 of Fig. 5, let $\tilde{\mathbf{y}}=\mathbf{y}^{(0)}(t_1,\gamma_1)+\alpha_0\Delta\mathbf{y}^{(0)}(t_1,\gamma_1)=(-0.6720,2.5640)^T$ be a tentative solution (where $\alpha_0=1$), the corresponding subsystem solutions and master objective are:

$$\begin{pmatrix}\tilde{\mathbf{y}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-2.0427\\ 2.5272\\ 0.3225\\ 2.1077\end{pmatrix},\quad \begin{pmatrix}\tilde{\mathbf{y}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-1.1900\\ 3.6050\\ 0.0224\\ 2.2036\end{pmatrix},\quad \tilde{\mathbf{y}}_3(\tilde{\mathbf{y}},t_1,\gamma_1)=\begin{pmatrix}0.0856\\ 0.7749\end{pmatrix}$$

$$F^*(\tilde{\mathbf{y}})=-2.1618$$

Assume that $\sigma=0.05$, since $F^*(\tilde{\mathbf{y}})-F^*(\mathbf{y}^{(0)}(t_1,\gamma_1))>\sigma$, $\nabla F^*(\mathbf{y}^{(0)}(t_1,\gamma_1))^T\Delta\mathbf{y}^{(0)}(t_1,\gamma_1)$ is not satisfied, the tentative solution $\tilde{\mathbf{y}}$ is rejected, and the step length α_0 is diminished by a half, i.e., $\alpha_0=0.5$. With this updated step length, a new tentative solution is calculated, given as $\tilde{\mathbf{y}}=(-1.3360,2.2820)^T$. The corresponding subsystem solutions and master objective are calculated as follows:

$$\begin{pmatrix}\tilde{\mathbf{y}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-2.3260\\ 2.6545\\ 0.3854\\ 1.9768\end{pmatrix},\quad \begin{pmatrix}\tilde{\mathbf{y}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-1.6475\\ 3.3991\\ 0.0293\\ 1.9566\end{pmatrix},\quad \tilde{\mathbf{y}}_3(\tilde{\mathbf{y}},t_1,\gamma_1)=\begin{pmatrix}-0.2645\\ 0.5385\end{pmatrix}$$

$$F^*(\tilde{\mathbf{y}})=-2.1618$$

Because $F^*(\tilde{\mathbf{y}})-F^*(\mathbf{y}^{(1)}(t_1,\gamma_1))>\sigma\nabla F^*(\mathbf{y}^{(1)}(t_1,\gamma_1))^T\Delta\mathbf{y}^{(1)}$ $\times(t_1,\gamma_1)$, the tentative solution $\tilde{\mathbf{y}}$ is accepted

$$\mathbf{y}^{(1)}(t_1,\gamma_1)=\tilde{\mathbf{y}}$$

$$\begin{pmatrix}\mathbf{y}_1^{(1)}(t_1,\gamma_1)\\ \mathbf{x}_1^{(1)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}\tilde{\mathbf{y}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix},\quad \begin{pmatrix}\mathbf{y}_2^{(1)}(t_1,\gamma_1)\\ \mathbf{x}_2^{(1)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}\tilde{\mathbf{y}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix},\quad \mathbf{y}_3^{(1)}(t_1,\gamma_1)=\tilde{\mathbf{y}}_3(\tilde{\mathbf{y}},t_1,\gamma_1)$$

Additionally, the Hessian approximation is updated following steps 1.4 and 1.5 of Fig. 5.

(End of first iteration)

$$\nabla F^*(\mathbf{y}^{(1)}(t_1,\gamma_1))=\begin{pmatrix}0.4599\\ 0.5075\end{pmatrix},\quad \mathbf{s}_0=\begin{pmatrix}0.6640\\ 0.2820\end{pmatrix},\quad \mathbf{r}_0=\begin{pmatrix}2.7458\\ 1.4783\end{pmatrix},\quad H_1=\begin{pmatrix}3.5186 & 1.4523\\ 1.4523 & 1.8228\end{pmatrix}$$

This finishes the first iteration of the BFGS algorithm. Since its termination criterion is not met, the BFGS continues into the next iteration.

The second iteration follows a similar fashion: a descent search direction is first calculated with the master gradient and the Hessian approximation

$$\Delta\mathbf{y}^{(1)}(t_1,\gamma_1)=\begin{pmatrix}-0.0235\\ -0.2597\end{pmatrix}$$

Let $\tilde{\mathbf{y}}=\mathbf{y}^{(1)}(t_1,\gamma_1)+\alpha_0\Delta\mathbf{y}^{(1)}(t_1,\gamma_1)=(-1.3595,2.0223)^T$ be a tentative solution (where $\alpha_0=1$), the corresponding subsystem solutions and master objective are

$$\begin{pmatrix}\tilde{\mathbf{y}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-2.2173\\ 2.5082\\ 0.4316\\ 1.8839\end{pmatrix},\quad \begin{pmatrix}\tilde{\mathbf{y}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}-1.6214\\ 3.1129\\ 0.0341\\ 1.8379\end{pmatrix},\quad \tilde{\mathbf{y}}_3(\tilde{\mathbf{y}},t_1,\gamma_1)=\begin{pmatrix}-0.2905\\ 0.3529\end{pmatrix}$$

$$F^*(\tilde{\mathbf{y}})=-3.4574$$

Since $F^*(\tilde{\mathbf{y}})-F^*(\mathbf{y}^{(1)}(t_1,\gamma_1))>\sigma\nabla F^*(\mathbf{y}^{(1)}(t_1,\gamma_1))^T\Delta\mathbf{y}^{(1)}(t_1,\gamma_1)$, the tentative solution $\tilde{\mathbf{y}}$ is accepted

$$\mathbf{y}^{(2)}(t_1,\gamma_1)=\tilde{\mathbf{y}}$$

$$\begin{pmatrix}\mathbf{y}_1^{(2)}(t_1,\gamma_1)\\ \mathbf{x}_1^{(2)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}\tilde{\mathbf{y}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_1(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix},\quad \begin{pmatrix}\mathbf{y}_2^{(2)}(t_1,\gamma_1)\\ \mathbf{x}_2^{(2)}(t_1,\gamma_1)\end{pmatrix}=\begin{pmatrix}\tilde{\mathbf{y}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\\ \tilde{\mathbf{x}}_2(\tilde{\mathbf{y}},t_1,\gamma_1)\end{pmatrix},\quad \mathbf{y}_3^{(2)}(t_1,\gamma_1)=\tilde{\mathbf{y}}_3(\tilde{\mathbf{y}},t_1,\gamma_1)$$

Additionally, the Hessian approximation is updated.

(End of second iteration)

$$\nabla F^*(\mathbf{y}^{(2)}(t_1,\gamma_1))=\begin{pmatrix}0.1012\\ 0.1856\end{pmatrix},\quad \mathbf{s}_1=\begin{pmatrix}-0.0235\\ -0.2597\end{pmatrix},\quad \mathbf{r}_1=\begin{pmatrix}-0.3748\\ -0.4998\end{pmatrix},\quad H_2=\begin{pmatrix}3.0492 & 1.1673\\ 1.1673 & 1.8189\end{pmatrix}$$

The algorithm will keep iterating until the BFGS termination criterion is met. If, for example, $\varepsilon_{\mathrm{grad}}=1\times10^{-3}$, it terminates at $(-1.2737,1.7736)^T$ after the sixth iterations. At this point, the penalty parameter γ will be updated if the nested framework is followed, while if the single loop framework is followed, not only the penalty parameter γ but also the regularization parameter t will be updated. After this update, the algorithm will start a new set of BFGS iterations in a similar fashion as presented above.

References

[1] Chidambaram, B., 1992, "Stackelberg Games in Optimal Design," MS thesis, Department of Mechanical Engineering, University of Houston, Houston, TX.

[2] Lewis, K., and Mistree, F., 1997, "Modeling Interactions in Multidisciplinary Design: A Game Theoretic Approach," AIAA J., **35**(8), pp. 1387–1392.

[3] Pakala, R. R., 1994, "A Study on Applications of Stackelberg Game Strategies in Concurrent Design Models," MS thesis, Department of Mechanical Engineering, University of Houston, Houston, TX.

[4] Rao, J. R. J., Badhrinath, K., Pakala, R., and Mistree, F., 1997, "A Study of Optimal Design Under Conflict Using Models of Multi-Player Games," Eng. Optimiz., **28**, pp. 63–94.

[5] Choi, S. C., Desarbo, W. S., and Harker, P. T., 1990, "Product Positioning Under Price Competition," Manage. Sci., **36**, pp. 175–199.

[6] Shiau, C. S., and Michalek, J. J., 2009, "Optimal Product Design Under Price Competition," ASME J. Mech. Des., **131**(7), p. 071003.

[7] Lu, S., Shanbhag, U. V., and Kim, H. M., 2008, "Multidisciplinary and Multilevel Design Optimization Problems With Complementarity Constraints," Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.

[8] Luo, Z.-Q., Pang, J.-S., and Ralph, D., 1996, *Mathematical Programs With Equilibrium Constraints*, Cambridge University Press, Cambridge, UK.

[9] Wallace, J., Philpott, A., O'Sullivan, M., and Ferris, M., 2006, "Optimal Rig Design Using Mathematical Programming," Proceedings of the Second High Performance Yacht Design Conference.

[10] Gu, X., Renaud, J., Ashe, L., Batill, S., Budhiraja, A., and Krajewski, L., 2002, "Decision-Based Collaborative Optimization," ASME J. Mech. Des., **124**(1), pp. 1–13.

[11] Michalek, J., Feinberg, F., and Papalambros, P., 2005, "Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading," J. Prod. Innovation Manage., **22**(1), pp. 42–62.

[12] Wassenaar, H. J., Chen, W., Cheng, J., and Sudjianto, A., 2005, "Enhancing Discrete Choice Demand Modeling for Decision-Based Design," ASME J. Mech. Des., **127**(4), pp. 514–523.

[13] Georgiopoulos, P., Jonsson, M., and Papalambros, P. Y., 2005, "Linking Optimal Design Decisions to the Theory of the Firm: The Case of Resource Allocation," ASME J. Mech. Des., **127**(3), pp. 358–366.

[14] Rao, S. S., and Freiheit, T. I., 1991, "A Modified Game Theory Approach to Multiobjective Optimization," ASME J. Mech. Des., **113**(3), pp. 286–291.

[15] Lewis, K., and Mistree, F., 1998, "Collaborative, Sequential, and Isolated Decisions in Design," ASME J. Mech. Des., **120**(4), pp. 643–652.

[16] Hernandez, G., Seepersad, C. C., and Mistree, F., 2002, "Designing for Maintenance: A Game Theoretic Approach," Eng. Optimiz., **34**(6), pp. 561–577.

[17] Hernandez, G., and Mistree, F., 2000, "Integrating Product Design and Manufacturing: A Game Theoretic Approach," Eng. Optimiz., **32**(6), pp. 749–775.

[18] Azarm, S., and Li, W., 1988, "Optimal Design Using a Two-Level Monotonicity-Based Decomposition," *Advances in Design Automation*, S. Rao, ed., Vol. 10, ASME, New York, pp. 41–48.

[19] Sobieszczanski-Sobieski, J., 1988, "Optimization by Decomposition: A Step From Hierarchic to Non-Hierarchic Systems," Second NASA Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Paper No. NASA-CP 3031.

[20] Sobieszczanski-Sobieski, J., Agte, J., and Sandusky, R., Jr., 2000, "Bilevel Integrated System Synthesis," AIAA J., **38**(1), pp. 164–172.

[21] Braun, R., 1996, "Collaborative Optimization: An Architecture for Large-Scale Distributed Design," Ph.D. thesis, Stanford University, Stanford, CA.

[22] Balling, R. J., and Gale, D. L., 1998, "Collaborative Optimization of Systems Involving Discrete Design at the Discipline Level," ASME J. Mech. Des., **120**(1), pp. 32–39.

[23] Haftka, R., and Watson, L., 2005, "Multidisciplinary Design Optimization With Quasiseparable Subsystems," Optim. Eng., **6**(1), pp. 9–20.

[24] Kim, H. M., 2001, "Target Cascading in Optimal System Design," Ph.D. thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI.

[25] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L., 2003, "Ana-

lytical Target Cascading in Automotive Vehicle Design," ASME J. Mech. Des., **125**(3), pp. 481–489.

[26] Kim, H. M., Michelena, N. F., Papalambros, P. Y., and Jiang, T., 2003, "Target Cascading in Optimal System Design," ASME J. Mech. Des., **125**(3), pp. 474–480.

[27] DeMiguel, V., and Murray, W., 2006, "A Local Convergence Analysis of Bilevel Programming Decomposition Algorithms," Optim. Eng., **7**(2), pp. 99–133.

[28] Tosserams, S., Etman, L. F. P., and Rooda, J. E., 2007, "An Augmented Lagrangian Decomposition Method for Quasiseparable Problems in MDO," Struct. Multidiscip. Optim., **34**(3), pp. 211–227.

[29] Lu, S., Kim, H. M., Norato, J., and Ha, C., 2008, "Analytical Target Cascading for Multi-Mode Design Optimization: An Engine Case Study," Proceedings of the Fourth AIAA Multidisciplinary Design Optimization Specialist Conference.

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

[31] Fletcher, R., Leyffer, S., Ralph, D., and Scholtes, S., 2006, "Local Convergence of SQP Methods for Mathematical Programs With Equilibrium Constraints," SIAM J. Optim., **17**(1), pp. 259–286.

[32] Anitescu, M., 2005, "On Solving Mathematical Programs With Complementarity Constraints as Nonlinear Programs," SIAM J. Optim., **15**(4), pp. 1203–1236.

[33] Hu, X. M., and Ralph, D., 2004, "Convergence of a Penalty Method for Mathematical Programming With Complementarity Constraints," J. Optim. Theory Appl., **123**(2), pp. 365–390.

[34] Leyffer, S., Lopez-Calva, G., and Nocedal, J., 2006, "Interior Methods for Mathematical Programs With Complementarity Constraints," SIAM J. Optim., **17**(1), pp. 52–77.

[35] Liu, X., and Sun, J., 2004, "A New Decomposition Technique in Solving Multistage Stochastic Linear Programs by Infeasible Interior Point Methods," J. Global Optim., **28**(4), pp. 197–215.

[36] Raghunathan, A. U., and Biegler, L. T., 2005, "An Interior Point Method for Mathematical Programs With Complementarity Constraints (MPCCs)," SIAM J. Optim., **15**(3), pp. 720–750.

[37] Shanbhag, U. V., 2006, "Decomposition and Sampling Methods for Stochastic Equilibrium Problems," Ph.D. thesis, Department of Management Science and Engineering (Operations Research), Stanford University, Stanford, CA.

[38] DeMiguel, V., Friedlander, M. P., Nogales, F. J., and Scholtes, S., 2005, "A Two-Sided Relaxation Scheme for Mathematical Programs With Equilibrium Constraints," SIAM J. Optim., **16**(2), pp. 587–609.

[39] Scholtes, S., 2001, "Convergence Properties of a Regularization Scheme for Mathematical Programs With Complementarity Constraints," SIAM J. Optim., **11**(4), pp. 918–936.

[40] Scheel, H., and Scholtes, S., 2000, "Mathematical Programs With Complementarity Constraints: Stationarity, Optimality and Sensitivity," Math. Methods Oper. Res., **25**, pp. 1–22.

[41] Alexandrov, N. M., 1997, "Multilevel Methods for Multidisciplinary Design Optimization," *Multidisciplinary Design Optimization*, N. M. Alexandrov and M. Y. Hussaini, eds., SIAM, Philadelphia, PA, pp. 79–89.

[42] Balling, R. J., and Sobieszczanski-Sobieski, J., 1996, "Optimization of Coupled Systems: A Critical Overview of Approaches," AIAA J., **34**(1), pp. 6–17.

[43] Fiacco, A. V., and McCormick, G. P., 1968, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York.

[44] Tosserams, S., Etman, L. F. P., Papalambros, P. Y., and Rooda, J. E., 2006, "An Augmented Lagrangian Relaxation for Analytical Target Cascading Using the Alternating Direction Method of Multipliers," Struct. Multidiscip. Optim., **31**(3), pp. 176–189.

[45] Li, Y., Lu, Z., and Michalek, J., 2008, "Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading," ASME J. Mech. Des., **130**(5), pp. 1–11.

[46] Li, D. H., and Fukushima, M., 2001, "A Modified BFGS Method and Its Global Convergence in Nonconvex Minimization," J. Comput. Appl. Math., **12**, pp. 15–35.

[47] Golinski, J., 1970, "Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods," J. Mech., **5**, pp. 287–309.

[48] Jiang, H., and Ralph, D., 1999, "QPECGEN, A MATLAB Generator for Mathematical Programs With Quadratic Objectives and Affine Variational Inequality Constraints, and Its Application to Algorithms," Comput. Optim. Appl., **13**, pp. 25–29.