

An augmented Lagrangian coordination method for distributed optimal design in MDO: Part I formulation and algorithms

Citation for published version (APA):

Tosserams, S., Etman, L. F. P., & Rooda, J. E. (2007). *An augmented Lagrangian coordination method for distributed optimal design in MDO: Part I formulation and algorithms*. (SE report; Vol. 2007-04). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2007

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Systems Engineering Group
Department of Mechanical Engineering
Eindhoven University of Technology
PO Box 513
5600 MB Eindhoven
The Netherlands
<http://se.wtb.tue.nl/>

SE Report: Nr. 2007-04

An augmented Lagrangian
coordination method for
distributed optimal design in
MDO: Part I formulation and
algorithms

S. Tosserams, L.F.P. Etman, and J.E. Rooda

ISSN: 1872-1567

SE Report: Nr. 2007-04
Eindhoven, February 2007
SE Reports are available via <http://se.wtb.tue.nl/sereports>

Abstract

Quite a number of coordination methods have been proposed for the distributed optimal design of large-scale systems consisting of a number of interacting subsystems. Several coordination methods are known to have numerical convergence difficulties that can be explained theoretically. The methods for which convergence proofs are available have mostly been developed for so called quasi-separable problems (i.e. problems with individual subsystems coupled only through a set of shared variables, not through constraints and/or objectives). In this paper we present a new coordination method for MDO problems with coupling variables as well as coupling objectives and constraints. Our approach employs an augmented Lagrangian penalty relaxation in combination with a block coordinate descent method. The coordination method can be shown to converge to KKT points of the original problem by using existing convergence results. Two formulation variants are presented offering a large degree of freedom in tailoring the coordination algorithm to the design problem at hand. The first centralized variant introduces a master problem to coordinate coupling of the subsystems. The second distributed variant coordinates coupling directly between subsystems. In a sequel paper we demonstrate the flexibility of the formulations, and investigate the numerical behavior of the proposed method.

1 Introduction

Multidisciplinary design optimization (MDO) problems are encountered in the design of large-scale engineering systems that consist of a number of interacting subsystems. The design of systems is a complicated task since size and the required level of expertise often prohibit the design problem to be solved as a whole. Consequently, the problem is decomposed into smaller, more manageable parts. Each of these parts, or design subproblems, is tackled by different design teams or disciplines, which act rather autonomously and use their own (legacy) design tools. As a result, one design group does not know how its decisions affect the other disciplines. To deal with multiple disciplines, a systematical coordination approach to systems design is desired. Within the field of MDO, such coordination methods with local decision autonomy are also referred to as multilevel methods (see, e.g., [1]).

Many coordination methods have been proposed for the distributed optimal design of MDO problems. Each method differs in the way the interaction is coordinated, and what type of problem structures are allowed. In general, the coordination principles can be divided into three main categories: interaction approximation methods, nested bi-level programming methods, and penalty relaxation methods.

Well-known *interaction approximation methods* include Concurrent SubSpace Optimization (CSSO) [2], the original Bi-Level Integrated System Synthesis method (BLISS) [3], and Multidisciplinary Design Optimization with Independent Subspaces (MDOIS) [4]. These interaction approximation methods define optimization subproblems for each subsystem, in which some approximation of the contributions of the remaining subsystems is included. Interaction approximation methods typically do not pose restrictions on the structure of the MDO problem. Interaction approximation methods are applicable to the most general class of MDO problems.

Numerical performance of interaction approximation methods typically depends on the quality of the approximations, and whether or not a system analysis is required. Some methods may show poor performance as a result of possibly infeasible subproblems [4, 5], and constraint activity changes [3].

Bi-level programming methods include Collaborative Optimization (CO) [7, 8], the Bi-Level Integrated System Synthesis variant BLISS2000 [9], the constraint margin approach of Haftka and Watson [10], and the penalty decomposition methods (IPD/EPD) of DeMiguel and Murray [11], amongst others. Bi-level programming methods introduce a coordinating master problem, which is superimposed over the optimization subproblems associated with the subsystems. These methods are referred to as bi-level programming methods because for a function evaluation of the master problem, all subproblems are optimized. Bi-level programming methods are typically restricted to so-called quasi-separable problems [10] in which subsystems are only coupled through a set of coupling variables. Bi-level programming methods are not applicable to problems with coupling constraints.

Many bi-level programming methods experience numerical difficulties when solving the master problem due to non-smoothness or failure to meet certain constraint qualifications [12, 13, 14]. Smoothness and constraint qualification are important requirements for the use of existing efficient gradient-based solution algorithms such as Sequential Quadratic Programming (SQP). Bi-level methods that do not satisfy these requirements have to use specialized, typically inefficient algorithms to solve the associated optimization problems. To reduce the computational costs associated with the use of inefficient solvers in combination with the bi-level formulation, the use of response surface modeling (RSM) techniques has been proposed [9, 15, 16]. Creating appropriate response surfaces is however not straightforward, and may become cumbersome for an increasing number of variables and non-smooth functions. Similar difficulties may arise for interaction approximation methods.

Penalty relaxation methods include Analytical Target Cascading (ATC) [17, 18], and the augmented Lagrangian methods of Blouin et al. [19], and Tosserams et al. [20]. The penalty relaxation methods relax the coupling constraints of MDO problems to arrive at subproblems with separable constraint sets. An iterative outer loop is typically introduced to restore feasibility with respect to the relaxed constraints. An advantage of penalty methods is that they can often be proven to converge to KKT points of the original problems [19, 20, 21]. Available penalty relaxation methods have however only been developed for quasi-separable problems coupled through a set of coupling variables; coupling objectives and constraints are not allowed.

In this paper, we propose a new penalty relaxation coordination method that can be used to solve MDO problems with coupling variables, a coupling objective, and coupling constraints. The proposed method is derived such that convergence to KKT points of the original non-decomposed problem can be shown under mild assumptions by combining existing results from nonlinear programming textbooks such as Refs. [22, 23]. Furthermore, efficient gradient-based solvers may be used to solve the associated subproblems, since the subproblem formulations are smooth and constraint qualifications hold. The method is based on augmented Lagrangian relaxation and block coordinate descent, two techniques used in the method of Ref. [20], and also recently linked to ATC [24].

Two variants of the method are presented. The first follows a centralized formulation where a single problem on the top level coordinates the coupling of the remaining subproblems on the lower level, similar to many bi-level MDO methods. The second follows a distributed formulation, similar to multilevel ATC, where coordination is performed directly between subproblems at different levels (which may be more than two). Hybrid versions of both formulations are also possible, providing the method a high degree of flexibility in formulating the decomposed problem to match existing organizational relations between subsystems.

Furthermore, we demonstrate that the multilevel hierarchical ATC formulation [17] and the bi-level augmented Lagrangian method of Tosserams et al. for quasi-separable MDO problems [20] are actually subclasses of the method proposed here.

In a sequel paper [25], we demonstrate the flexibility of the formulation, and investigate its numerical behavior on a number of example problems.

2 Decomposition of the original all-in-one problem

The multidisciplinary design optimization problem with M subsystems, here referred to as the *original all-in-one (AIO) problem*, is given by:

$$\begin{aligned} \min_{\mathbf{z}=[\mathbf{y}^T, \mathbf{x}_1^T, \dots, \mathbf{x}_M^T]^T} \quad & f_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) \leq \mathbf{0}, \\ & \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) = \mathbf{0}, \\ & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M, \end{aligned} \quad (1)$$

where the vector of design variables $\mathbf{z} = [\mathbf{y}^T, \mathbf{x}_1^T, \dots, \mathbf{x}_M^T]^T \in \mathbb{R}^n$ consists of a number of coupling variables $\mathbf{y} \in \mathbb{R}^{n^y}$, and a number of local variables $\mathbf{x}_j \in \mathbb{R}^{n_j^x}$ associated exclusively to subsystem j , and $n^y + \sum_{j=1}^M n_j^x = n$. The coupling variables may be common design variables shared by multiple subsystems, and input-output variables that link the analysis models of different subsystems. The coupling objective $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ and coupling constraints $\mathbf{g}_0 : \mathbb{R}^n \mapsto \mathbb{R}^{m_g^0}$ and $\mathbf{h}_0 : \mathbb{R}^n \mapsto \mathbb{R}^{m_h^0}$ are non-separable and may depend on all design variables \mathbf{z} . Local objectives $f_j : \mathbb{R}^{n_j} \mapsto \mathbb{R}$, and local

3 Decomposition of the original all-in-one problem

	\mathbf{y}	\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\mathbf{x}_4
$f_0 \mathbf{g}_0 \mathbf{h}_0$					
$f_1 \mathbf{g}_1 \mathbf{h}_1$					
$f_2 \mathbf{g}_2 \mathbf{h}_2$					
$f_3 \mathbf{g}_3 \mathbf{h}_3$					
$f_4 \mathbf{g}_4 \mathbf{h}_4$					

Figure 1: Functional dependence table for four-element example original MDO problem of the form (1)

constraints $\mathbf{g}_j : \mathbb{R}^{n_j} \mapsto \mathbb{R}^{m_j^g}$ and $\mathbf{h}_j : \mathbb{R}^{n_j} \mapsto \mathbb{R}^{m_j^h}$ are associated exclusively to subsystem j , and may depend on the coupling variables \mathbf{y} and the local variables \mathbf{x}_j of only a single subsystem j , such that $n_j = n^y + n_j^x$. Furthermore, $m_0^g + \sum_{j=1}^M m_j^g = m^g$ and $m_0^h + \sum_{j=1}^M m_j^h = m^h$. Unless indicated otherwise, all vectors in this paper are assumed to be column vectors.

Figure 1 illustrates the structure of the original AIO problem in the functional dependence table (FDT) for a four element example. Similar to Ref. [26], we shade the (i, j) -entry of the table if the function of row i depends on the variables of column j . Throughout this section, we use the functional dependence table to illustrate the effect of the proposed problem transformations on the problem structure. Observe the coupling of the problem through both the coupling variables \mathbf{y} , as well as the coupling functions f_0 , \mathbf{g}_0 , and \mathbf{h}_0 . Without these shared quantities, the problem would be block-diagonal, and fully decomposable into M smaller subproblems that can be solved independently. When no coupling functions f_0 , \mathbf{g}_0 , and \mathbf{h}_0 are present, (1) reduces to a quasi-separable MDO problem, as considered in, e.g., Ref. [10].

The AIO problem can be solved directly with so called single-level MDO methods such as the “individual discipline feasible” (IDF) approach, or “multidisciplinary feasible” (MDF) algorithms. These single-level MDO methods facilitate disciplinary *analysis* autonomy, rather than *decision* autonomy as obtained with multi-level decomposition methods. The reader is referred to Ref. [27] for an overview of single-level formulations.

The augmented Lagrangian coordination method presented in this paper solves the original MDO problem by the following four steps:

1. Introduction of auxiliary variables and consistency constraints
2. Relaxation of the coupling and consistency constraints
3. Formulation of the decomposed problem
4. Distributed solution of the decomposed problem

Here, Steps 1 through 3 are problem transformation steps, and Step 4 entails the actual solution algorithms.

Existing convergence proofs for the solution algorithms of Step 4 only apply to problems with fully separable constraint sets. The separability of constraint sets implies that a subproblem’s constraints may only depend on the variables associated with that subproblem, but not on variables of other subproblems. However, the local constraint sets of the original AIO problem (1) are not separable because of the coupling variables \mathbf{y} , and the coupling constraints \mathbf{g}_0 and \mathbf{h}_0 . To obtain a decomposed problem formulation with fully separable constraint sets, the first three problem transformation steps have to be taken first.

The structure of the decomposed formulation of Step 3 is determined by the definition of the consistency constraints in Step 1. A degree of flexibility exists in the definition of these constraints, and as a result, we are able to manipulate the structure of the decomposed problem. In the following two sections we demonstrate two alternative formulations, each resulting in a different decomposed problem. In the first alternative (Section 3), consistency constraints are defined such that the final decomposed problem has a bi-level structure, similar to existing bi-level MDO methods such as CO, and BLISS2000. In this “centralized” bi-level formulation, a central coordinating subproblem on the top level is superimposed over the M subproblems, associated with the M subsystems of the original problem. The second alternative (Section 4) gives a multilevel “distributed” formulation in which no artificial coordination problem is introduced, but coordination is handled directly between the subsystems (similar to ATC). Hybrid versions of both formulations are also possible, providing a large degree of freedom in formulating the decomposed problem such that the problem structure becomes compatible with the real-life organizational relations between the subsystems. The solution algorithms used in Step 4 are presented separately in Section 5.

3 Variant 1: Centralized coordination

This section presents the centralized decomposed formulation of the original AIO problem (1). In this centralized formulation, a top level central coordinating problem is positioned on top of the M subsystem design subproblems at the lower level (illustrated in Fig. 4). The solution algorithms of Step 4 alternate between solving the master problem and the subproblems, in contrast to the nested bi-level programming methods such as collaborative optimization and BLISS2000.

3.1 Step 1: introduction of auxiliary variables and consistency constraints

In the first transformation, auxiliary coupling variables $\mathbf{y}_j \in \mathbb{R}^{n^y}$ are introduced at each subsystem to separate the local constraint sets \mathbf{g}_j and \mathbf{h}_j , $j = 1, \dots, m$. To assure consistency amongst the auxiliary variables, *consistency constraints* are introduced that force $\mathbf{y}_1 = \mathbf{y}_2 = \dots = \mathbf{y}_M$. There are many alternatives for these consistency constraints, and each alternative gives a specific final structure of the decomposed problem.

In this section, we choose to link all auxiliary coupling variables \mathbf{y}_j to the original coupling variables \mathbf{y} , and therefore define the consistency constraints $\mathbf{c} : \mathbb{R}^{(M+1) \cdot n^y} \mapsto \mathbb{R}^{m^c}$ as $\mathbf{c}(\mathbf{y}, \mathbf{y}_1, \dots, \mathbf{y}_M) = [\mathbf{c}_1^T, \dots, \mathbf{c}_M^T]^T = \mathbf{0}$ with:

$$\mathbf{c}_j = \mathbf{y} - \mathbf{y}_j = \mathbf{0}, \quad (2)$$

where, $m^c = M \cdot n^y$, and $\mathbf{c}_j : \mathbb{R}^{2 \cdot n^y} \mapsto \mathbb{R}^{n^y}$ denotes the vector of inconsistencies between the original vector of coupling variables \mathbf{y} and the auxiliary coupling variables \mathbf{y}_j at subsystem j . The original vector \mathbf{y} can be seen here as a central ‘master copy’.

The *modified AIO problem* after introduction of the auxiliary variables and consistency constraints is given by:

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M} \quad & f_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) \leq \mathbf{0}, \\ & \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) = \mathbf{0}, \\ & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{c}(\mathbf{y}, \mathbf{y}_1, \dots, \mathbf{y}_M) = \mathbf{0}. \end{aligned} \quad (3)$$

Observe that the solutions to the modified AIO problem (3) and the original AIO problem (1) are

5 Variant 1: Centralized coordination

	y	x ₁	y ₁	x ₂	y ₂	x ₃	y ₃	x ₄	y ₄
$f_0 \mathbf{g}_0 \mathbf{h}_0$	■	■		■		■		■	
\mathbf{c}	■		■		■		■		■
$f_1 \mathbf{g}_1 \mathbf{h}_1$		■	■						
$f_2 \mathbf{g}_2 \mathbf{h}_2$				■	■				
$f_3 \mathbf{g}_3 \mathbf{h}_3$						■	■		
$f_4 \mathbf{g}_4 \mathbf{h}_4$								■	■

Figure 2: Functional dependence table for four-element example modified AIO problem (3) with only coupling constraints

equal because of the consistency constraints. In Theorem 4.1 of Ref. [11], solution equivalence is proven for quasi-separable problems, and we expect that the result can be extended to the modifications presented here.

The FDT of the modified AIO problem (3) is illustrated in Fig. 2, where separability of the local constraint sets can be observed, as well as non-separability of the introduced consistency constraints \mathbf{c} and the coupling constraints \mathbf{g}_0 , and \mathbf{h}_0 . By introducing the auxiliary variables and consistency constraints, we have only coupling constraints, instead of both coupling variables and constraints.

Other instances of the modified all-in-one problem (3) have appeared in the MDO literature. For example, Cramer et al. [27] have used the term “All-At-Once ” approach, and Alexandrov and Lewis [12] presented the modified AIO problem as “Distributed Analysis Optimization ”.

Although the local constraint sets \mathbf{g}_j and \mathbf{h}_j are now fully separable with respect to the subsystem design variables, the consistency constraints \mathbf{c} and the coupling constraints \mathbf{g}_0 and \mathbf{h}_0 are not, and prevent application of the distributed optimization techniques in Step 4.

3.2 Step 2: relaxation of the consistency and coupling constraints

The second transformation relaxes the consistency and coupling constraints to arrive at a problem with fully separable constraint sets.

First, the consistency constraints are relaxed using an augmented Lagrangian penalty function $\phi_{\mathbf{c}} : \mathbb{R}^{m^c} \mapsto \mathbb{R}$:

$$\phi_{\mathbf{c}}(\mathbf{c}) = \mathbf{v}_{\mathbf{c}}^T \mathbf{c} + \|\mathbf{w}_{\mathbf{c}} \circ \mathbf{c}\|_2^2 = \sum_{j=1}^M \phi_{\mathbf{c},j}(\mathbf{c}_j) = \sum_{j=1}^M \mathbf{v}_{\mathbf{c},j}^T \mathbf{c}_j + \sum_{j=1}^M \|\mathbf{w}_{\mathbf{c},j} \circ \mathbf{c}_j\|_2^2, \quad (4)$$

where $\phi_{\mathbf{c},j}(\mathbf{c}_j) : \mathbb{R}^{n^y} \mapsto \mathbb{R}$ is the penalty function on the consistency constraints \mathbf{c}_j for subsystem j , $\mathbf{v}_{\mathbf{c}} = [\mathbf{v}_{\mathbf{c},1}^T, \dots, \mathbf{v}_{\mathbf{c},M}^T]^T \in \mathbb{R}^{m^c}$ is the vector of Lagrange multiplier estimates for the consistency constraints, and $\mathbf{w}_{\mathbf{c}} = [\mathbf{w}_{\mathbf{c},1}^T, \dots, \mathbf{w}_{\mathbf{c},M}^T]^T \in \mathbb{R}^{m^c}$ is the vector of penalty weights, and $\mathbf{v}_{\mathbf{c},j} \in \mathbb{R}^{n^y}$, and $\mathbf{w}_{\mathbf{c},j} \in \mathbb{R}^{n^y}$. The symbol \circ represents the Hadamard product: an entry-wise multiplication of two vectors, such that $\mathbf{a} \circ \mathbf{b} = [a_1, \dots, a_n]^T \circ [b_1, \dots, b_n]^T = [a_1 b_1, \dots, a_n b_n]^T$.

Second, the coupling equality constraints \mathbf{h}_0 are relaxed using an augmented Lagrangian penalty function $\phi_{\mathbf{h}} : \mathbb{R}^{m^h} \mapsto \mathbb{R}$:

$$\phi_{\mathbf{h}}(\mathbf{h}_0) = \mathbf{v}_{\mathbf{h}}^T \mathbf{h}_0 + \|\mathbf{w}_{\mathbf{h}} \circ \mathbf{h}_0\|_2^2, \quad (5)$$

where $\mathbf{v}_{\mathbf{h}} \in \mathbb{R}^{m^h}$ is the vector of Lagrange multiplier estimates for the system-wide equality con-

	\mathbf{y}	\mathbf{x}_0	\mathbf{x}_1	\mathbf{y}_1	\mathbf{x}_2	\mathbf{y}_2	\mathbf{x}_3	\mathbf{y}_3	\mathbf{x}_4	\mathbf{y}_4
f_0	■		■		■		■		■	
ϕ_g, ϕ_h	■	■	■		■		■		■	
ϕ_c	■			■		■		■		■
$f_1, \mathbf{g}_1, \mathbf{h}_1$			■	■	■					
$f_2, \mathbf{g}_2, \mathbf{h}_2$					■	■				
$f_3, \mathbf{g}_3, \mathbf{h}_3$							■	■		
$f_4, \mathbf{g}_4, \mathbf{h}_4$									■	■

Figure 3: Functional dependence table for four-element example relaxed AIO problem (7) with block-diagonal constraint structure

straints, and $\mathbf{w}_h \in \mathbb{R}^{m_h}$ is the vector of penalty weights.

Third, the coupling inequality constraints \mathbf{g}_0 are relaxed using an augmented Lagrangian penalty function $\phi_g : \mathbb{R}^{2 \cdot m_g^s} \mapsto \mathbb{R}$:

$$\phi_g(\mathbf{g}_0, \mathbf{x}_0) = \mathbf{v}_g^T (\mathbf{g}_0 + \mathbf{x}_0^2) + \|\mathbf{w}_g \circ (\mathbf{g}_0 + \mathbf{x}_0^2)\|_2^2, \quad (6)$$

where $\mathbf{v}_g \in \mathbb{R}^{m_g^s}$ is the vector of Lagrange multiplier estimates for the coupling inequality constraints, $\mathbf{w}_g \in \mathbb{R}^{m_g^s}$ is the vector of penalty weights, $\mathbf{x}_0 \in \mathbb{R}^{m_g^s}$ are slack variables, and $\mathbf{x}_0^2 = \mathbf{x}_0 \circ \mathbf{x}_0$. Because augmented Lagrangian methods drive its argument ($\mathbf{g}_0 + \mathbf{x}_0^2$ in this case) to zero, the slack variables allow for non-positive values for the inequality coupling constraints after relaxation (see [22]). Without the slack variables, the coupling inequality constraint values would be driven to zero and therefore act as equality instead of inequality constraints.

The *relaxed AIO problem* is given by:

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{x}_0, \mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{y}_M, \mathbf{x}_M} \quad & f_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) + \phi_c(\mathbf{c}(\mathbf{y}, \mathbf{y}_1, \dots, \mathbf{y}_M)) \\ & + \phi_g(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M)) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M. \end{aligned} \quad (7)$$

The FDT of the relaxed AIO problem is illustrated in Fig. 3. The full separability of constraints can be seen in the block-diagonal structure. The figure also shows the non-separability of the penalty terms ϕ_c , ϕ_g , and ϕ_h . In step 4, this coupling is accounted for by the solution strategy.

The solution to the relaxed problem (7) is not equal to the original problem (1), because a *relaxation error* is introduced by relaxing the coupling constraints. By appropriate selection of the Lagrange multiplier estimates $\mathbf{v} = [\mathbf{v}_c^T, \mathbf{v}_g^T, \mathbf{v}_h^T]^T$, and penalty weights $\mathbf{w} = [\mathbf{w}_c^T, \mathbf{w}_g^T, \mathbf{w}_h^T]^T$, this relaxation error can be driven to zero. In fact, the algorithms we propose in Section 5 solve the decomposed problem for a sequence of penalty parameters.

Note that any penalty function can be used to relax the problem. Here we use the augmented Lagrangian function for a number of reasons. First, the augmented Lagrangian function is continuous and also has continuous first and second order derivatives. Second, it avoids the ill-conditioning of the relaxed problem, encountered for some classes of penalty functions. Third, it is additively separable with respect to the individual consistency constraints \mathbf{c}_j , which allows for a degree of parallelism during distributed optimization in Step 4. Finally, the augmented Lagrangian function has been extensively studied in the field of nonlinear programming, providing a large knowledge-base of theory and parameter update strategies (see, e.g., Refs. [22, 28, 29] for overviews).

7 Variant 1: Centralized coordination

After relaxation of the coupling and consistency constraints, the constraint sets are separable with respect to the subsystem variables, as illustrated in Fig. 3, and we are ready to formulate the decomposed problem.

3.3 Step 3: formulation of the decomposed problem

In the third transformation, we decompose the relaxed problem into a number of subproblems P_j , $j = 1, \dots, M$, each associated with a subsystem of the original problem, and a coordinating master problem P_0 . The block coordinate descent algorithm we propose for Step 4 iterates between solving the relaxed AIO problem (7) for a subset of variables, while holding the remaining variables fixed at their previous value. The master problem P_0 is equivalent to solving the relaxed problem for the master copy of coupling variables \mathbf{y} and the slack variables \mathbf{x}_0 , while fixing the remaining variables. Subproblems P_j , $j = 1, \dots, M$, are solved for the subsystem variables $(\mathbf{y}_j, \mathbf{x}_j)$, while fixing the remaining variables.

In the master problem P_0 only the penalty terms have to be included that depend on the master problem variables \mathbf{y} and \mathbf{x}_0 . The remaining functions are independent of \mathbf{y} and \mathbf{x}_0 and are therefore constant. The master problem P_0 is given by:

$$\min_{\bar{\mathbf{x}}_0 = [\mathbf{y}^T, \mathbf{x}_0^T]^T} f_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M \phi_{c,j}(\mathbf{c}_j(\mathbf{y}, \mathbf{y}_j)) + \phi_g(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M)), \quad (8)$$

Note that subproblem P_0 is an unconstrained minimization problem, however lower and upper bounds on the shared variables \mathbf{y} may be included in the problem.

The M subproblems P_j , $j = 1, \dots, M$ associated with the original subsystems are solved for $\bar{\mathbf{x}}_j$, $j = 1, \dots, M$, and are given by:

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j = [\mathbf{y}_j^T, \mathbf{x}_j^T]^T} & f_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M) + f_j(\mathbf{y}_j, \mathbf{x}_j) + \phi_{c,j}(\mathbf{c}_j(\mathbf{y}, \mathbf{y}_j)) \\ & + \phi_g(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M)) \\ \text{subject to} & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}. \end{aligned} \quad (9)$$

If the original AIO problem satisfies standard smoothness and constraint qualification assumptions, then these assumptions also hold for the master problem and the subproblems (this easily follows from the observation that the augmented Lagrangian function is smooth). Therefore, we are able to use existing efficient gradient-based algorithms such as SQP for their solution. Moreover, the Lagrange multipliers associated with the local constraints \mathbf{g}_j and \mathbf{h}_j at the solution to the decomposed problem are equal to those of the original AIO problem when the penalty parameters \mathbf{v} and \mathbf{w} are selected appropriately (Proposition 2.11 of Ref. [28]).

The structure of the decomposed problem (8)–(9) is depicted in Fig. 4. Dashed lines indicate coupling through either the system objective f_0 or the penalty functions ϕ_g and ϕ_h , and solid lines indicate coupling through $\phi_{c,j}$. These solid lines clearly reflect the centralized character of the consistency constraints defined by (2). As illustrated in this figure, the subproblems of the decomposed problems are still coupled. How to account for the coupling of subproblems is discussed in Section 5.

The decomposed problem (8)–(9) is highly coupled because of the system-wide objective and system-wide constraints. Any sparsity in the original problem (1) is however directly reflected in sparsity in the decomposed problem. When e.g. elements 2 and 3 are not linked through a coupling objective or coupling constraints in the original problem, the associated subproblems P_2

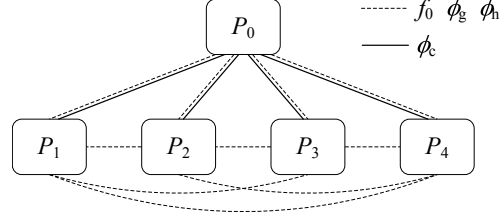


Figure 4: Bi-level decomposed problem structure for the centralized formulation

and P_3 will not be linked in the decomposed problem. Such problem sparsity gives opportunities for parallelization, and can be exploited in the solution strategies of Step 4.

3.4 Coupling variable sparsity

So far, we assumed that all subsystems depend on the complete set of coupling variables \mathbf{y} . In practice however, each subsystem may depend only on a subset of coupling variables. The theory presented in this section can be applied straightforwardly in such a case, however it would be unnatural to include coupling design variables in subproblems that do not depend on them.

To reflect such coupling variables sparsity in the formulation, a binary selection matrix \mathbf{S}_j of size $n_j^y \times n^y$ can be defined for each subsystem. The matrices \mathbf{S}_j are defined such that the matrix multiplication $\mathbf{S}_j \mathbf{y}$ collects only the n_j^y components of \mathbf{y} relevant to subsystem j , where $n_j^y \leq n^y$. With the selection matrices, centralized consistency constraints $\mathbf{c} = [\mathbf{c}_1^T, \dots, \mathbf{c}_M^T]^T = \mathbf{0}$ similar to (2) are defined, however with:

$$\mathbf{c}_j = \mathbf{S}_j \mathbf{y} - \mathbf{y}_j = \mathbf{0} \quad j = 1, \dots, M, \quad (10)$$

where the auxiliary variables $\mathbf{y}_j \in \mathbb{R}^{n_j^y}$ and inconsistencies $\mathbf{c}_j \in \mathbb{R}^{n_j^y}$ are only introduced for those n_j^y components relevant to subsystem j , and therefore $\mathbf{c} : \mathbb{R}^{n^y + \sum_{i=1}^M n_i^y} \mapsto \mathbb{R}^{m^c}$ with $m^c = \sum_{i=1}^M n_i^y \leq M \cdot n^y$. The definition of \mathbf{y}_j introduced here differs from the definition in (2), where copies of all of the coupling variables \mathbf{y} are included in \mathbf{y}_j . Here, only the n_j^y components relevant to subsystem j are included.

The inconsistency penalty terms $\phi_{c,j}$ of subproblem j in problem (8)–(9) are in the sparse case given by:

$$\phi_{c,j}(\mathbf{y}, \mathbf{y}_j) = \mathbf{v}_{c,j}^T (\mathbf{S}_j \mathbf{y} - \mathbf{y}_j) + \|\mathbf{w}_{c,j} \circ (\mathbf{S}_j \mathbf{y} - \mathbf{y}_j)\|_2^2. \quad (11)$$

where in this case $\phi_{c,j} : \mathbb{R}^{n^y + n_j^y} \mapsto \mathbb{R}$, and the penalty terms $\mathbf{v}_{c,j} \in \mathbb{R}^{n_j^y}$, and $\mathbf{w}_{c,j} \in \mathbb{R}^{n_j^y}$ are only introduced for the relevant coupling variable inconsistencies. The remainder of the decomposed problem is unchanged.

To illustrate the use of the selection matrices \mathbf{S}_j , consider a four-subsystem problem where subsystems 2, 3 and 4 are coupled through variable y_1 , y_2 couples subsystems 1 and 2, and y_3 couples subsystems 1 and 3. The assembled vector of coupling variables becomes $\mathbf{y} = [y_1, y_2, y_3]^T$, and $n^y = 3$. For the consistency constraints of (10), the selection matrices are $\mathbf{S}_1 = [0 \ 1 \ 0; 0 \ 0 \ 1]$, $\mathbf{S}_2 = [1 \ 0 \ 0; 0 \ 1 \ 0]$, $\mathbf{S}_3 = [1 \ 0 \ 0; 0 \ 0 \ 1]$, and $\mathbf{S}_4 = [1 \ 0 \ 0]$, which leaves $\mathbf{y}_1 = [y_2^{[1]}, y_3^{[1]}]^T$, $\mathbf{y}_2 = [y_1^{[2]}, y_2^{[2]}]^T$, $\mathbf{y}_3 = [y_1^{[3]}, y_3^{[3]}]^T$, and $\mathbf{y}_4 = [y_1^{[4]}]$, where the top-right index denotes the subproblem of computation. Furthermore, $n_1^y = 2$, $n_2^y = 2$, $n_3^y = 2$, and $n_4^y = 1$. Under these conventions, the

$m^c = \sum_{j=1}^M n_j^y = 7$ consistency constraints (10) are given by $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \mathbf{c}_3^T, \mathbf{c}_4^T]^T = \mathbf{0}$, where:

$$\begin{aligned}\mathbf{c}_1 &= \mathbf{S}_1 \mathbf{y} - \mathbf{y}_1 = [y_2^{[0]} - y_2^{[1]}, y_3^{[0]} - y_3^{[1]}]^T = \mathbf{0}, \\ \mathbf{c}_2 &= \mathbf{S}_2 \mathbf{y} - \mathbf{y}_2 = [y_1^{[0]} - y_1^{[2]}, y_2^{[0]} - y_2^{[2]}]^T = \mathbf{0}, \\ \mathbf{c}_3 &= \mathbf{S}_3 \mathbf{y} - \mathbf{y}_3 = [y_1^{[0]} - y_1^{[3]}, y_3^{[0]} - y_3^{[3]}]^T = \mathbf{0}, \\ \mathbf{c}_4 &= \mathbf{S}_4 \mathbf{y} - \mathbf{y}_4 = [y_1^{[0]} - y_1^{[4]}].\end{aligned}\tag{12}$$

4 Variant 2: Distributed coordination

In this section, we present a distributed decomposed formulation of the original AIO problem (1). In this formulation, the subsystem design subproblems can be positioned in a user-specified (possibly multilevel) structure. No coordinating master problem is introduced, and coordination of the coupling variables is handled directly between subproblems, similar to analytical target cascading. The distributed formulation is able to reflect a multilevel (organizational) structure present in the original AIO design problem.

Similar to the previous section, the non-sparse case is given first, after which an extension to coupling variables sparsity is presented.

4.1 Step 1: introduction of auxiliary variables and consistency constraints

For each subsystem we again introduce auxiliary coupling variables $\mathbf{y}_j \in \mathbb{R}^{n^y}$ to separate the local constraint sets. However, we omit the use of the original vector of coupling variables \mathbf{y} in the modified problem. Instead, we introduce consistency constraints that only depend on the auxiliary coupling variables.

As observed in the previous section, the definition of the consistency constraints results in a specific decomposed problem structure. Many choices for the consistency constraints \mathbf{c} are possible, and each set of constraints given as specific final problem structure. In the centralized approach, we link the auxiliary coupling variables (\mathbf{y}_j) to a central master copy of the coupling variables (\mathbf{y}), which resulted in a bi-level decomposed problem. A similar bi-level structure can be obtained by taking $\mathbf{c}(\mathbf{y}_1, \dots, \mathbf{y}_M) = [\mathbf{c}_{M1}^T, \dots, \mathbf{c}_{M(M-1)}^T]^T = \mathbf{0}$ where:

$$\mathbf{c}_{Mj} = \mathbf{y}_M - \mathbf{y}_j = \mathbf{0}.\tag{13}$$

For this choice, the consistency constraints $\mathbf{c}_{Mj} : \mathbb{R}^{2 \cdot n^y} \mapsto \mathbb{R}^{n^y}$ link the auxiliary shared variable copies of subsystems $j = 1, \dots, M-1$ to those of subsystem M . For this choice, the subproblem associated with subsystem M will assume the role of subproblem P_0 , as illustrated in Fig. 5(a). The use of the double index for the consistency constraints allows for a more general notation of the consistency constraints in this section.

By defining alternative consistency constraints, other problem structures can be obtained. For example if the consistency constraints are given by $\mathbf{c} = [\mathbf{c}_{12}^T, \dots, \mathbf{c}_{(M-1)M}^T]^T = \mathbf{0}$ with:

$$\mathbf{c}_{j(j+1)} = \mathbf{y}_j - \mathbf{y}_{j+1} = \mathbf{0},\tag{14}$$

inconsistencies are defined between coupling variables of subsystem j and its neighboring subsystem $j+1$, instead of subsystem M . Under these conventions narrow-tree structures as depicted in Fig. 5(b) are formed.

The multilevel structure of Fig. 5(c) is generated by defining the consistency constraints as $\mathbf{c} =$

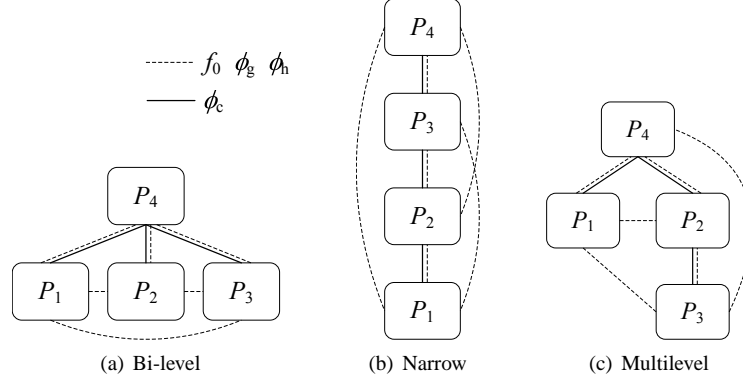


Figure 5: Non-hierarchical multilevel problem structures for alternative consistency constraints formulations

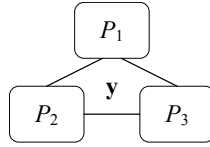


Figure 6: Three subsystem example problem structure

$[\mathbf{c}_{23}^T, \mathbf{c}_{41}^T, \mathbf{c}_{42}^T]^T = \mathbf{0}$ where:

$$\begin{aligned} \mathbf{c}_{4j} &= \mathbf{y}_4 - \mathbf{y}_j = \mathbf{0} & j = 1, 2, \\ \mathbf{c}_{23} &= \mathbf{y}_2 - \mathbf{y}_3 = \mathbf{0}. \end{aligned} \quad (15)$$

For a general problem structure, the consistency constraints \mathbf{c} must meet the following requirements:

1. The consistency constraints \mathbf{c} must force $\mathbf{y}_1 = \mathbf{y}_2 = \dots = \mathbf{y}_M$
2. The consistency constraints \mathbf{c} must be linearly independent

The first requirement makes sure that the auxiliary coupling variables all take the same values, as is obviously necessary for consistency. The second requirement assures that the Lagrange multipliers associated with the consistency constraints are unique, which is important for the solution algorithms of Step 4. As will be discussed in Section 5, the penalty parameters \mathbf{v} have to approach the optimal Lagrange multipliers in order to reduce the relaxation error. Therefore, non-uniqueness of these multipliers may complicate, or even prevent convergence to the optimal solution. Although some algorithms may not be hindered by this, we can simply prevent possible difficulties by making sure that consistency constraints are linearly independent.

The number of consistency constraints can easily be determined from the number of subsystems M , and the number of coupling variables n^y . Observe that in the original AIO problem, one has n^y degrees of freedom through the coupling variables \mathbf{y} . For each subsystem, auxiliary copies are introduced, resulting in a total of $M \cdot n^y$ degrees of freedom. The consistency constraints \mathbf{c} force $\mathbf{y}_1 = \mathbf{y}_2 = \dots = \mathbf{y}_M$ (requirement 1), reducing the degrees of freedom again to n^y . Hence, there must be exactly $M \cdot n^y - n^y = (M - 1) \cdot n^y$ linearly independent consistency constraints.

Although the number of consistency constraints may seem trivial, consider the case where three subsystems are linked through a vector of coupling variables \mathbf{y} , as illustrated in Fig. 6. For this system, it might seem natural to define the following consistency constraints: $\mathbf{c} = [\mathbf{c}_{12}^T, \mathbf{c}_{23}^T, \mathbf{c}_{13}^T]^T$ where

$$\begin{aligned}\mathbf{c}_{12} &= \mathbf{y}_1 - \mathbf{y}_2, \\ \mathbf{c}_{23} &= \mathbf{y}_2 - \mathbf{y}_3, \\ \mathbf{c}_{13} &= \mathbf{y}_1 - \mathbf{y}_3.\end{aligned}$$

The above $M \cdot n^y$ consistency constraints however are linearly dependent ($\mathbf{c}_{13} = \mathbf{c}_{12} + \mathbf{c}_{23}$). As discussed, linearly dependent consistency constraints have non-unique Lagrange multipliers, which may cause numerical difficulties in the solution algorithms of Step 4. By removing one of the dependent vectors of consistency constraints, $(M - 1) \cdot n^y$ linearly independent consistency constraints remain and force $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{y}_3$, which is required for the consistency constraints.

In general, the consistency constraints \mathbf{c} can be defined as a collection of consistency constraints \mathbf{c}_{jn} defined between subsystems $j = 1, \dots, M$ and their set of N_j neighbors:

$$\mathbf{c}_{jn} = \mathbf{y}_j - \mathbf{y}_n = \mathbf{0} \quad \{n \in \mathcal{N}_j | n > j\} \quad j = 1, \dots, M \quad (16)$$

that must force $\mathbf{y}_1 = \mathbf{y}_2 = \dots = \mathbf{y}_M$. The neighbors \mathcal{N}_j are defined as the subsystems to which subsystem j is coupled to through the *consistency constraints*. Coupling through the coupling objective or constraints is not considered. Furthermore, the condition $n > j$ makes sure that only one of the linearly dependent pair \mathbf{c}_{jn} and \mathbf{c}_{nj} is included in the consistency constraints (e.g., only $\mathbf{c}_{12} = \mathbf{y}_1 - \mathbf{y}_2$, and not also $\mathbf{c}_{21} = \mathbf{y}_2 - \mathbf{y}_1$).

Before defining the modified AIO problem, the functional dependency of the coupling objective and constraints have to be reallocated. In the original AIO problem (1), these functions depend on the coupling variables \mathbf{y} . For the distributed formulation of this section, the use of the master copy \mathbf{y} has been omitted. Instead, and without loss of generality, the functional dependencies of the coupling objective f_0 and coupling constraint penalties are modified to depend on the coupling variables of subsystem M , \mathbf{y}_M , instead of the master copy \mathbf{y} , which gives $f_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M)$, $\mathbf{g}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M)$, and $\mathbf{h}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M)$. In this case, subsystem M has been given the design freedom with respect to the coupling variables in the coupling functions. The components of \mathbf{y} can also be distributed over two or more subsystems by using selection matrices, as described for the sparse case in Section 4.4.

The modified AIO problem for the multilevel formulation is now given by:

$$\begin{aligned}\min_{\mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{y}_M, \mathbf{x}_M} \quad & f_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M) \leq \mathbf{0}, \\ & \mathbf{h}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M) = \mathbf{0}, \\ & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{c}_{jn} = \mathbf{y}_j - \mathbf{y}_n = \mathbf{0} \quad \{n \in \mathcal{N}_j | n > j\} \quad j = 1, \dots, M.\end{aligned} \quad (17)$$

Observe that the solutions to the modified AIO problem (17) and the original AIO problem (1) are equal because of the consistency constraints.

4.2 Step 2: relaxation of the consistency and coupling constraints

Again, the consistency and coupling constraints are non-separable. An augmented Lagrangian penalty function is used to relax the consistency constraints \mathbf{c}_{jn} of the modified AIO problem (17):

$$\phi_{\mathbf{c},jn} = \mathbf{v}_{jn}^T (\mathbf{c}_{jn}) + \|\mathbf{w}_{jn} \circ (\mathbf{c}_{jn})\|_2^2, \quad (18)$$

with $\phi_{c,jn} : \mathbb{R}^{n^y} \mapsto \mathbb{R}$, and where $\mathbf{v}_{jn} \in \mathbb{R}^{m^c}$, and $\mathbf{w}_{jn} \in \mathbb{R}^{m^c}$.

The coupling constraints are again relaxed using an augmented Lagrangian function, similar to the centralized approach of Section 3 (see Eqs. (5) and (6)).

The *relaxed AIO problem* for the multilevel formulation is given by:

$$\begin{aligned} \min_{\mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{y}_M, \mathbf{x}_M} \quad & f_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M) + \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) + \sum_{j=1}^M \sum_{\{n \in \mathcal{N}_j | n > j\}} \phi_{c,jn}(\mathbf{c}_{jn}(\mathbf{y}_j, \mathbf{y}_n)) \\ \text{subject to} \quad & + \phi_g(\mathbf{g}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M)) \\ & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M. \end{aligned} \quad (19)$$

4.3 Step 3: formulation of the decomposed problem

After relaxation, the constraint sets are fully separable with respect to the subsystem variables. For the distributed solution algorithms of Step 4 we define M variable subsets $\bar{\mathbf{x}}_j = [\mathbf{y}_j^T, \mathbf{x}_j^T]^T \in \mathbb{R}^{n_j}$, $j = 1, \dots, M$, where each subset is associated with one of the subsystem.

The slack variables \mathbf{x}_0 can no longer be appointed to the central master problem present in the centralized formulation. Instead, the slack variables are included in one of the subsystem variable subsets $\bar{\mathbf{x}}_j$. Without loss of generality, the slack variables are included in subsystem M such that $\bar{\mathbf{x}}_M = [\mathbf{y}_M^T, \mathbf{x}_M^T, \mathbf{x}_0^T]^T \in \mathbb{R}^{n^y + n_M^x + m_0^s}$.

The *general distributed subproblem* P_j only has to include the functions that depend on $\bar{\mathbf{x}}_j$, and is given by:

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j = [\mathbf{y}_j^T, \mathbf{x}_j^T, \mathbf{x}_0^T]^T} \quad & f_0(\mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M) + f_j(\mathbf{y}_j, \mathbf{x}_j) \\ & + \sum_{\{n \in \mathcal{N}_j | n > j\}} \phi_{c,jn}(\mathbf{c}_{jn}(\mathbf{y}_j, \mathbf{y}_n)) + \sum_{\{n \in \mathcal{N}_j | n < j\}} \phi_{c,nj}(\mathbf{c}_{nj}(\mathbf{y}_n, \mathbf{y}_j)) \\ \text{subject to} \quad & + \phi_g(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M)) \\ & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}, \end{aligned} \quad (20)$$

where the slack variables \mathbf{x}_0 are only included in subproblem P_M . The consistency constraint penalty of P_j only includes terms that depend on \mathbf{y}_j and hence consists of two parts. The first part is associated with the consistency constraints between subsystem j and its neighbors that have a higher subsystem index $n > j$. The second part accounts for the consistency constraints between subsystem j and its neighbors that have a lower index $n \in \mathcal{N}_j | n < j$.

4.4 Coupling variable sparsity

Coupling variable sparsity may also exist in distributed problems since a subproblem P_j may not depend on all coupling variables \mathbf{y} , but only on a selection. Similar to the centralized formulation, auxiliary coupling variables and consistency constraints are only introduced for the coupling variables relevant to subsystem j .

The reflection of this sparsity in the consistency constraints requires the definition of N_j selection matrices $\mathbf{S}_{jn} \in \mathbb{R}^{n_{jn}^y \times n_n^y}$ for each of the subsystem $j = 1, \dots, M$, where N_j denotes the number of neighbors of subsystem j , and n_{jn}^y is the number of variables shared by subsystems j and n , therefore $n_{jn}^y = n_{nj}^y$, $n_{jn}^y \leq n_j^y$ and $n_{jn}^y \leq n_n^y$. The selection matrix \mathbf{S}_{jn} collects the set of coupling variables from \mathbf{y}_j that are shared with neighbor subsystem n . With the selection matrices, the

consistency constraints between subsystems j and their neighbors \mathcal{N}_j are defined by:

$$\mathbf{c}_{jn} = \mathbf{S}_{jn}\mathbf{y}_j - \mathbf{S}_{nj}\mathbf{y}_n = \mathbf{0}, \quad \{n \in \mathcal{N}_j | n > j\} \quad j = 1, \dots, M \quad (21)$$

where typically $\mathbf{S}_{jn} \neq \mathbf{S}_{nj}$.

An additional difficulty arises for the coupling variables \mathbf{y} in the coupling objective and constraints. For shared variable sparsity, none of the subproblems may depend on all of these coupling variables, and therefore solving for \mathbf{y} in the coupling objective and constraints cannot be appointed to a single subproblem. Hence, the solution has to be distributed over more than one subproblem.

To achieve this distribution, the solution of every component of \mathbf{y} is assigned to a subproblem through the definition of selection matrices $\mathbf{T}_j \in \mathbb{R}^{n_j^x \times n_j^y}$, $j = 1, \dots, M$. These selection matrices collect the subset of n_j^x coupling variables from \mathbf{y}_j solved for in the coupling functions by subsystem j . Note that each component of the coupling variables \mathbf{y} in the original all-in-one problem (1) is only solved for at a single subproblem, and therefore the vector $\mathbf{y}' = [(\mathbf{T}_1\mathbf{y}_1)^T, \dots, (\mathbf{T}_M\mathbf{y}_M)^T]^T \in \mathbb{R}^{n^y}$ has the same components as \mathbf{y} but not necessarily in the same order, and $\sum_{j=1}^M n_j^x = n^y$.

The general distributed subproblem P_j for sparsity of coupling variables is defined by:

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j = [\mathbf{y}_j^T, \mathbf{x}_j^T, (\mathbf{x}_0^T)]^T} \quad & f_0(\mathbf{T}_1\mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{T}_M\mathbf{y}_M, \mathbf{x}_M) + f_j(\mathbf{y}_j, \mathbf{x}_j) \\ & + \sum_{\{n \in \mathcal{N}_j | n > j\}} \phi_{c,jn}(\mathbf{c}_{jn}(\mathbf{y}_j, \mathbf{y}_n)) + \sum_{\{n \in \mathcal{N}_j | n < j\}} \phi_{c,nj}(\mathbf{c}_{nj}(\mathbf{y}_n, \mathbf{y}_j)) \\ & + \phi_g(\mathbf{g}_0(\mathbf{T}_1\mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{T}_M\mathbf{y}_M, \mathbf{x}_M), \mathbf{x}_0) + \phi_h(\mathbf{h}_0(\mathbf{T}_1\mathbf{y}_1, \mathbf{x}_1, \dots, \mathbf{T}_M\mathbf{y}_M, \mathbf{x}_M)) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}, \end{aligned} \quad (22)$$

with the inconsistency penalty $\phi_{c,jn}$ given by (18) but with the consistency constraints defined by (21), and $\mathbf{v}_{jn} \in \mathbb{R}^{n_{jn}^y}$, $\mathbf{w}_{jn} \in \mathbb{R}^{n_{jn}^y}$. One may also define different sets of selection matrices \mathbf{T}_j for the coupling objective and each of the coupling constraints.

To illustrate both types of selection matrices, again take the four-subsystem example of the previous section. Recall that subsystems 2, 3, and 4 share y_1 , subsystems 1 and 2 share y_2 , and subsystems 1 and 3 share y_3 . The total vector of coupling variables is $\mathbf{y} = [y_1, y_2, y_3]^T$, and the coupling variable copies are $\mathbf{y}_1 = [y_2^{(1)}, y_3^{(1)}]^T$, $\mathbf{y}_2 = [y_1^{(2)}, y_2^{(2)}]^T$, $\mathbf{y}_3 = [y_1^{(3)}, y_3^{(3)}]^T$, $\mathbf{y}_4 = [y_1^{(4)}]$. The top-right index again refers to the subsystem of computation. In the problem structure of Fig. 7, the neighbors¹ for each subsystem are given by $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1, 3, 4\}$, $\mathcal{N}_3 = \{1, 2\}$, and $\mathcal{N}_4 = \{2\}$. The selection matrices are $\mathbf{S}_{12} = [1, 0]$, $\mathbf{S}_{13} = [0, 1]$, $\mathbf{S}_{21} = [0, 1]$, $\mathbf{S}_{23} = [1, 0]$, $\mathbf{S}_{24} = [1, 0]$, $\mathbf{S}_{31} = [0, 1]$, $\mathbf{S}_{32} = [1, 0]$, and $\mathbf{S}_{42} = [1]$, which gives the consistency constraints $\mathbf{c} = [\mathbf{c}_{12}^T, \mathbf{c}_{13}^T, \mathbf{c}_{23}^T, \mathbf{c}_{24}^T]^T$ as in (21) with:

$$\begin{aligned} \mathbf{c}_{12} &= y_2^{(1)} - y_2^{(2)} = 0 \\ \mathbf{c}_{13} &= y_3^{(1)} - y_3^{(3)} = 0 \\ \mathbf{c}_{23} &= y_1^{(2)} - y_1^{(3)} = 0 \\ \mathbf{c}_{24} &= y_1^{(2)} - y_1^{(4)} = 0 \end{aligned}$$

Furthermore assume that the system objective and coupling constraints depend on all three of the coupling variables. To solve for y_1 in subproblem P_4 , y_2 in P_1 , and y_3 in P_3 , define $\mathbf{T}_1 = [1, 0]$, $\mathbf{T}_2 = [1]$, $\mathbf{T}_3 = [0, 1]$, and $\mathbf{T}_4 = [1]$, which leaves $\mathbf{y}' = [(\mathbf{T}_1\mathbf{y}_1)^T, (\mathbf{T}_2\mathbf{y}_2)^T, (\mathbf{T}_3\mathbf{y}_3)^T, (\mathbf{T}_4\mathbf{y}_4)^T]^T = [y_2, [1], y_3, y_1]^T = [y_2, y_3, y_1]^T$.

¹Recall that the neighbors \mathcal{N}_j of subsystem j are defined as the subsystems to which subsystem j is coupled to through the inconsistency penalties (solid lines).

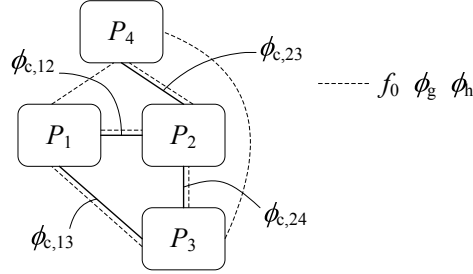


Figure 7: Example non-hierarchical multilevel problem structure for sparse shared variable coupling

5 Solution algorithms

Now that the decomposed problems are formulated, a solution strategy is required to solve them. Although the decomposed problems presented in the previous sections have different formulations, they are all obtained through the same principle, namely augmented Lagrangian relaxation. Therefore a generic solution strategy can be formulated that can easily be adapted to both formulations.

Solution strategies for solving generalized ATC problems have two tasks:

1. minimizing the relaxation error by appropriate penalty parameter setting, and
2. accounting for the coupling of subproblems through the non-separable objective.

Similar to ATC [30, 37] and the augmented Lagrangian approach of Ref. [20], we propose a strategy that consists of inner and outer loops. In the outer loops, penalty parameters \mathbf{v} and \mathbf{w} are updated, and in the inner loops, the coupled decomposed problem is solved for fixed penalty parameters. Such a nested solution strategy is of the form:

Step 0 Set initial guess and penalty parameters

Step 1 (Inner loop) Solve decomposed problem for fixed penalty parameters with block coordinate descent algorithm

Step 2 If converged stop, otherwise go to Step 3

Step 3 (Outer loop) Update penalty parameters with method of multipliers and return to Step 1

5.1 Outer loop: method of multipliers

In the outer loop, the penalty parameters $\mathbf{v} = [\mathbf{v}_c^T, \mathbf{v}_g^T, \mathbf{v}_h^T]^T$, and $\mathbf{w} = [\mathbf{w}_c^T, \mathbf{w}_g^T, \mathbf{w}_h^T]^T$ are updated to reduce the relaxation error. This error can be reduced by two mechanisms [22]:

1. select \mathbf{v} close to λ^*
2. increase \mathbf{w} to $+\infty$

where $\lambda^* = [\lambda_c^{*T}, \lambda_g^{*T}, \lambda_h^{*T}]^T$ is the vector of optimal Lagrange multipliers associated with the coupling constraints \mathbf{c} , \mathbf{g}_0 , and \mathbf{h}_0 at the solution to the modified problem (3). Furthermore $\lambda_c^* \in \mathbb{R}^{m^c}$, $\lambda_g^* \in \mathbb{R}^{m_0^g}$, $\lambda_h^* \in \mathbb{R}^{m_0^h}$, and $\lambda^* \in \mathbb{R}^{m^c + m_0^g + m_0^h}$.

Many convergent methods have been proposed to use either one or both of the error reduction mechanisms. From duality theory (see, e.g., [22, 28]), *dual methods* utilize the first mechanism but are not applicable to general non-convex problems. From nonlinear programming (see, e.g., [22]), *penalty methods* can be adapted to use the second mechanism. Such methods however have problems with ill-conditioning because the weights \mathbf{w} have to approach infinity for convergence.

Here, we use the well-known *method of multipliers* (see e.g. [22, 28, 29]), which is specifically designed for the augmented Lagrangian function, and uses both mechanisms simultaneously. The method of multipliers is often more efficient when compared to both dual methods and penalty approaches alone [28].

Let \mathbf{q} be the vector of linking constraints defined by the concatenation of the consistency and coupling constraints vectors:

$$\mathbf{q} = \begin{bmatrix} \mathbf{c} \\ \mathbf{g}_0 + \mathbf{x}_0^2 \\ \mathbf{h}_0 \end{bmatrix} \quad (23)$$

Then, the method of multipliers updates the estimates of the Lagrange multipliers for outer loop iteration $k + 1$ by:

$$\mathbf{v}^{k+1} = \mathbf{v}^k + 2\mathbf{w}^k \circ \mathbf{w}^k \circ \mathbf{q}^k \quad (24)$$

where \mathbf{w}^k are the penalty weights at iteration k , and the constraint values \mathbf{q} are evaluated at the inner loop solution of outer loop iteration k .

We increase the weights by a factor β only when the reduction in the linking constraint value is smaller than some fraction γ [22]. As a result, the penalty weights are only increased when the contribution of the Lagrange multiplier update (24) did not lead to a large enough reduction in the violation of the linking constraints.

For the i th linking constraint q_i , $i = 1, \dots, m^c + m_0^g + m_0^h$ of \mathbf{q} , the associated penalty weight w_i is updated as:

$$w_i^{k+1} = \begin{cases} w_i^k & \text{if } |q_i^k| \leq \gamma |q_i^{k-1}| \\ \beta w_i^k & \text{if } |q_i^k| > \gamma |q_i^{k-1}| \end{cases} \quad i = 1, \dots, m^c + m_0^g + m_0^h \quad (25)$$

where $\beta > 1$ and $0 < \gamma < 1$. Typically $\gamma = 0.25$ and $2 < \beta < 3$ are recommended to speed up convergence [22]. In combination with the block coordinate descent inner loop, we observe that $\beta = 2.2$ and $\gamma = 0.4$ perform well in general.

Convergence to local solutions of the modified AIO problem has been proven for the method of multipliers algorithm under mild assumptions: local solutions must satisfy second order sufficiency conditions, and \mathbf{w} sufficiently large (see, e.g., Proposition 2.4 in Ref. [28]). Under the more strict assumption of convexity, the method of multipliers can be shown to converge to the globally optimal solution of the original AIO problem for any positive penalty weight, as long as the sequence of weights is non-decreasing. The weight update scheme of (25) makes sure that weights eventually become large enough to assure convergence.

The solution procedure is terminated when two conditions are satisfied. First, the change in the maximal linking constraint value for two consecutive outer loop iterations must be smaller than

some user-defined termination tolerance $\varepsilon > 0$:

$$\|\mathbf{q}^k - \mathbf{q}^{k-1}\|_\infty < \varepsilon \quad (26)$$

Second, the maximal linking constraint violation must also be smaller than tolerance $\varepsilon > 0$:

$$\|\mathbf{q}^k\|_\infty < \varepsilon \quad (27)$$

For original AIO problems that do not have a feasible solution, the second criterion will never be satisfied, and the algorithm will not be terminated. In such cases, one may omit the second criterion, but at the risk of converging prematurely at a non-feasible solution because of a (locally) small reduction in \mathbf{q} . Another option is to monitor the value of the penalty term, which goes to zero for feasible solutions. For non-feasible solutions, the value of the penalty term will go to infinity. Therefore, if the second criterion is not satisfied, and the penalty term grows very large, then it is likely that the problem does not have a feasible solution.

5.2 Inner loop: block coordinate descent

The update algorithms of the outer loop require the solution to the relaxed AIO problem for fixed weights. To find the solution to the relaxed AIO problem (7) and (19) for fixed weights, we use the iterative block coordinate descent (BCD) algorithm [22]. Instead of solving the relaxed AIO problem as a whole, the block coordinate descent method sequentially solves the disciplinary subproblems P_1, \dots, P_M , and, for the centralized formulation, also the master problem P_0 . The BCD method is also known as the “nonlinear Gauss-Seidel” method [31], or “alternating optimization” [32, 36].

Convergence to KKT points of the relaxed AIO problem for fixed penalty parameters has been proven under mild conditions: global solutions to subproblems P_1, \dots, P_M are uniquely attained, and the objectives f_j , $j = 1, \dots, M$ of the relaxed AIO problem are continuously differentiable (Proposition 2.7.1 in Ref. [22]).

The inner loop BCD algorithm is terminated when the relative change in the objective function value of the relaxed AIO problem for two consecutive inner loop iterations is smaller than some user-defined termination tolerance $\varepsilon_{\text{inner}} > 0$. Let F denote the objective of the relaxed problem (7) and (19), then the inner loop is terminated when:

$$\frac{|F^\xi - F^{\xi-1}|}{1 + |F^\xi|} < \varepsilon_{\text{inner}}, \quad (28)$$

where ξ denotes the inner loop iteration number. The division by $1 + |F^\xi|$ is used for proper scaling of the criterion for very large as well as very small objectives [33]. The termination tolerance $\varepsilon_{\text{inner}}$ should be smaller than the outer loop termination tolerance ε to assure sufficient accuracy of the inner loop solution. We use $\varepsilon_{\text{inner}} = \varepsilon/100$.

An alternative termination strategy is to use looser tolerances when the penalty parameters are still far from their optimal values. The tolerances are tightened as k increases, and the penalty parameters approach their optimal values. With this strategy we do not waste costly inner loop iterations for finding a solution to the relaxed problem that is far from the optimal solution. More formally, such an inexact approach uses a different tolerance $\varepsilon_{\text{inner}}^k$ for each outer loop iteration. Providing that the sequence $\{\varepsilon_{\text{inner}}^k\}$ is non-increasing, and $\varepsilon_{\text{inner}}^k \rightarrow 0$, convergence of the inner loop solutions to a KKT point of the original problem has been proven (Proposition 2.14 in [28]).

In this inexact inner loop, more moderate values for β (smaller) and γ (larger) are advised for efficiency. Experiments indicate that $\beta = 2.0$ and $\gamma = 0.5$ give good results.

It would be possible to terminate the inner loop after a single iteration, similar to the alternating direction method of multipliers [31]. Such an approach proved to be very efficient for ATC [24] and the augmented Lagrangian approach of Ref. [20] for quasi-separable problems. An alternating direction variant of the solution algorithms presented here, though, would not be within the assumptions of the convergence proof as given in Ref. [31]. This proof is only valid for linear coupling constraints and an additively separable objective. The problems considered in this study, however, may have nonlinear coupling constraints and a non-separable objective. Although the proof does not apply to the problems considered in the current study, experiments on a number of example problems indicate that the performance of an alternating direction variant depends on the problem at hand. For some problems, the algorithm finds accurate solutions very efficiently. For other problems though, the algorithm terminates prematurely at non-optimal designs, which is clearly not desired.

Part of the block coordinate descent algorithm can be parallelized, since subproblems that are not directly coupled can be solved in parallel. The general convergence proof however is not valid for fully parallelized subproblem solutions, as in a Jacobi-like scheme (Ref. [31]). Although parallelization methods exist, the associated convergence proofs often require convexity of the original problem, and additive separability of the objective and/or coupling constraints (see, e.g., Refs. [34, 35]).

5.3 Initial weight selection

Although the above algorithms converge for any positive initial weight, the performance of the outer loop method of multipliers depends on the choice of the initial weight \mathbf{w} (see, e.g., [22, 28, 29]). To select the initial weights, we use an approach similar to the one suggested in Ref. [20]. With this approach, the weights are chosen such that the sum of the penalty terms is a fraction α of the objective function value: $\phi \approx \alpha|f|$, with $\alpha = 0.1$.

Similar to Ref. [20], we initially set $\mathbf{v} = \mathbf{0}$, and take all weights equal $\mathbf{w} = w$, such that $\phi = w^2 \mathbf{q}^T \mathbf{q}$ with \mathbf{q} defined by Eq. (23). The initial weights are then selected as:

$$w = \sqrt{\frac{\alpha|\hat{f}|}{\hat{\mathbf{q}}^T \hat{\mathbf{q}}}} \quad (29)$$

where \hat{f} and $\hat{\mathbf{q}}$ are estimates of typical objective function and the linking constraint values. For many engineering problems, a reasonable (order of magnitude) estimate of the objective function minimum in the optimum can often be given. The approach assumes that the estimate of the objective is non-zero, which is often the case in engineering design. However, if \hat{f} happens to be zero, a non-zero, conservative “typical value” should be taken for the objective function.

The estimates for the linking constraints $\hat{\mathbf{q}}$ are obtained by solving the decomposed problem for small weights \mathbf{w} , and zero Lagrange multipliers $\mathbf{v} = \mathbf{0}$. For these weights, the penalty term will be small when compared to the objective function value. As a consequence, the allowed linking constraint violations will be large, and the solution of the relaxed all-in-one problem will produce an estimate $\hat{\mathbf{q}}_j$ for the size of the linking constraint values.

6 Existing subclasses of the proposed method

In this section, we demonstrate that the hierarchical and multilevel analytical target cascading formulation is actually a subclass of the distributed formulation as presented in Section 4. Second, we show that the augmented Lagrangian method of Ref. [20] is a subclass of the centralized

approach of Section 3.

6.1 Analytical target cascading (ATC)

ATC subproblems [17, 24, 37] have more specific characteristics when compared to the subproblems (22) of Section 4. These characteristics and their effects on the subproblem formulation are:

1. A subproblem does not have system objectives or system constraints, therefore $f_0 = 0$, $\mathbf{g}_0 = []$, and $\mathbf{h}_0 = []$.
2. A subproblem is only connected to its single parent subsystem p_j and its set of c_j children $\mathcal{C}_j = \{k_1, \dots, k_{c_j}\}$, thus the set of neighbors is given by $\mathcal{N}_j = \{p_j, k_1, \dots, k_{c_j}\}$.
3. The shared variables for a subproblem are given by the responses to their parent and the targets sent to their children, which gives $\mathbf{y}_j = [\mathbf{r}_j^T, \mathbf{t}_{k_1}^T, \dots, \mathbf{t}_{k_{c_j}}^T]^T$.
4. The selection matrix \mathbf{S}_{jp_j} picks the responses $\mathbf{r}_j = \mathbf{S}_{jp_j} \mathbf{y}_j$ from the vector of shared variables, hence $\mathbf{S}_{jp_j} = [\mathbf{I}, \mathbf{0}, \dots, \mathbf{0}]$, where \mathbf{I} is the identity matrix.
5. The selection matrix \mathbf{S}_{jk} , $k \in \mathcal{C}_j$ picks the targets $\mathbf{t}_k = \mathbf{S}_{jk} \mathbf{y}_j$ for child k from the vector of shared variables: $\mathbf{S}_{jk} = [\mathbf{0}, \mathbf{0}, \dots, \mathbf{I}, \dots, \mathbf{0}]$, where the identity matrix \mathbf{I} is inserted at the appropriate position for child k .

Under these assumptions and after expansion of the inconsistency penalties of (21), the general distributed subproblem (22) for ATC is given by:

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j = [\mathbf{y}_j^T, \mathbf{x}_j^T]^T} \quad & f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{v}_{p_j j}^T (\mathbf{t}_j - \mathbf{r}_j) + \sum_{k \in \mathcal{C}_j} \mathbf{v}_{jk}^T (\mathbf{t}_k - \mathbf{r}_k) \\ & + \|\mathbf{w}_{p_j j} \circ (\mathbf{t}_j - \mathbf{r}_j)\|_2^2 + \sum_{k \in \mathcal{C}_j} \|\mathbf{w}_{jk} \circ (\mathbf{t}_k - \mathbf{r}_k)\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}, \\ \text{where} \quad & \mathbf{y}_j = [\mathbf{r}_j^T, \mathbf{t}_{k_1}^T, \dots, \mathbf{t}_{k_{c_j}}^T]^T. \end{aligned}$$

The terms $\mathbf{v}_{p_j j}^T \mathbf{t}_j$ and $-\mathbf{v}_{jk}^T \mathbf{r}_k$ drop out of the objective because they do not depend on any of the subproblem variables $\bar{\mathbf{x}}_j$. Furthermore, the first index of the penalty parameters $\mathbf{v}_{p_j j}$, $\mathbf{w}_{p_j j}$, \mathbf{v}_{jk} , \mathbf{w}_{jk} , can be dropped because it always refers to the unique parent of the element designated by the second index. The uniqueness is guaranteed by the pure hierarchical formulation of ATC. When finally all variables, functions, and parameters are augmented with a level index i denoting the level at which the associated subsystem is located, the ATC subproblem (30) in the formulation of Ref. [24] is obtained, which shows that the ATC formulation is a subclass of the distributed formulation presented in this paper.

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{ij}} \quad & f_{ij}(\bar{\mathbf{x}}_{ij}) - \mathbf{v}_{ij}^T \mathbf{r}_{ij} + \sum_{k \in \mathcal{C}_{ij}} \mathbf{v}_{(i+1)k}^T \mathbf{t}_{(i+1)k} \\ & + \|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2 \\ & + \sum_{k \in \mathcal{C}_{ij}} \|\mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where} \quad & \bar{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}^T, \mathbf{r}_{ij}^T, \mathbf{t}_{(i+1)k_1}^T, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}^T]^T. \end{aligned} \tag{30}$$

Note that ATC problems may also be coordinated using the centralized approach of Section 3. With this approach, all subproblems of all levels are allocated to the bottom level and can be

solved in parallel, and the master subproblem P_0 is superimposed to coordinate consistency amongst all subproblems.

6.2 Augmented Lagrangian method for quasi-separable problems of Ref. [20]

The augmented Lagrangian method of Tosserams et al. [20] has been proposed to solve quasi-separable MDO problems. Such problems have subsystems that are coupled through shared variables only, and not through a shared objective or constraints. To solve these problems, several other coordination methods have been developed such as Collaborative Optimization [7, 8], Bi-Level Integrated System Synthesis [9], the constraint margin approach of Ref. [10], and Multidisciplinary Design Optimization based on Independent Subspaces [4]. Note that ATC coordinates quasi-separable problems that have a hierarchical structure.

The general quasi-separable AIO problem is given by the general AIO problem (1), without coupling functions ($f_0 = 0$, and $\mathbf{g}_0 = 0$, \mathbf{h}_0):

$$\begin{aligned} \min_{\mathbf{z}=[\mathbf{y}^T, \mathbf{x}_1^T, \dots, \mathbf{x}_M^T]^T} \quad & \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \dots, M, \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \dots, M, \end{aligned} \quad (31)$$

The decomposed formulation presented in Ref. [20] is given by a master subproblem P_0 :

$$\min_{\bar{\mathbf{x}}_0=[\mathbf{y}]} \sum_{j=1}^M \phi_{c,j}(\mathbf{y}, \mathbf{y}_j), \quad (32)$$

and disciplinary subproblems P_j , $j = 1, \dots, M$:

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j=[\mathbf{y}_j^T, \mathbf{x}_j^T]^T} \quad & f_j(\mathbf{y}_j, \mathbf{x}_j) + \phi_{c,j}(\mathbf{y}, \mathbf{y}_j) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}, \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}. \end{aligned} \quad (33)$$

This formulation is equal to the centralized formulation of (8)–(9) when excluding the f_0 , ϕ_g and ϕ_h terms, illustrating that the method presented in this paper is a generalization of the augmented Lagrangian method of Ref. [20]. Note that a distributed formulation variant of the centralized formulation of [20] for quasi-separable problems can be formed by the techniques of Section 4.

7 Conclusions

This paper presents a new augmented Lagrangian coordination method for multilevel MDO problems with coupling variables as well as coupling objectives and constraints. The coordination method is derived from available techniques as found in the nonlinear programming literature. The main techniques used are augmented Lagrangian relaxation and block coordinate descent. Convergence to KKT points of the original problem can be shown under mild assumptions by combining existing results on convergence analysis.

When compared to existing coordination methods for MDO, the proposed method has several advantages. Besides being provably convergent, the proposed formulations have smooth master and subproblems that can be solved efficiently using gradient-based techniques. The solution algorithm solves the subproblems sequentially, with the freedom to choose the number of iterations before the penalty parameters are updated in the outer loop.

The proposed method offers a large degree of flexibility to the designer through the two formulation variants. With the centralized variant, the coupling between subsystems is coordinated through a central master problem, as is often seen in classic MDO coordination methods. The distributed formulation variant gives the designer the opportunity to coordinate subsystem coupling aligned with an existing (possibly multi-level) organizational structure of the design problem. Hybrid approaches that use a combination of both centralized and distributed coordination are also possible.

Another degree of flexibility is offered through the distinction between coupling variables and coupling functions. With coupling constraints, each subsystem optimizes only for its own set of design variables, while fixing the variables of the other subsystems. For coupling variables on the other hand, each subsystem is given additional design freedom by introducing local copies of the variables coupling the subsystems. Which situation is desired typically depends on the design problem at hand, and should not be prescribed by the coordination algorithm used.

Furthermore, we showed that the hierarchical Analytical Target Cascading method and the centralized augmented Lagrangian method of Ref. [20] are subclasses of our method proposed here. Hence the flexibility our method offers can be used, e.g., to centralize coordination of ATC, or employ distributed coordination in the method of Ref. [20].

In a sequel paper [25], we demonstrate our method on a number of example problems, and investigate its numerical behavior.

Bibliography

- [1] Balling RJ, Sobieszczanski-Sobieski J. Optimization of coupled systems: a critical overview of approaches. *AIAA Journal*, 1996; **24**(1):6–17.
- [2] Sobieszczanski-Sobieski J. Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In: *Recent Advances in Multidisciplinary Analysis and Optimization*, Barthelemy J-F (ed.). NASA-CP 3031, 1988.
- [3] Sobieszczanski-Sobieski J, Agte JS, Sandusky Jr. RR. Bilevel Integrated System Synthesis. *AIAA Journal*, 2000; **38**(1):164–172.
- [4] Shin MK, Park GJ. Multidisciplinary design optimization based on independent subspaces. *International Journal for Numerical Methods in Engineering*, 2005; **64**:599–617.
- [5] Shankar Y, Ribbens CJ, Haftka RT, Watson LT. Computational study of a nonhierarchical decomposition algorithm. *Computational Optimization and Applications*, 1993; **2**:273–293.
- [6] Alexandrov NM, Lewis RM. Comparative properties of collaborative optimization and other approaches to MDO. In: *1st ASMO UK/ISSMO Conference on Engineering Design Optimization*. MCB-press, 1999.
- [7] Braun RD. *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*. PhD dissertation, Stanford University, 1996.
- [8] Braun RD, Moore AA, Kroo IM. Collaborative approach to launch vehicle design. *SIAM Journal of Optimization*, 1997; **4**(4):754–776.
- [9] Sobieszczanski-Sobieski J, Altus TD, Phillips M, Sandusky Jr RR. Bilevel Integrated System Synthesis for concurrent and distributed processing. *AIAA Journal*, 2000; **41**(10):1996–2003.
- [10] Haftka RT, Watson LT. Multidisciplinary design optimization with quasiseparable subsystems. *Optimization and Engineering*, 2005; **6**:9–20.
- [11] DeMiguel AV, Murray W. A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering*, 2006; **7**:99–133.
- [12] Alexandrov NM, Lewis RM. Analytical and computational aspects of Collaborative Optimization for multidisciplinary design. *AIAA Journal*, 2002; **40**(2):301–309.
- [13] DeMiguel AV, Murray W. An analysis of collaborative optimization methods. In: *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000; AIAA Paper 00-4720.
- [14] DeMiguel AV, Murray W. Analysis and enhancement of collaborative optimization for multidisciplinary design. *AIAA Journal*, 2004; **42**(2):348–360.
- [15] Sobieski IP, Kroo I. Collaborative Optimization using response surface estimation. *AIAA Journal*, 2000; **38**(10):1931–1938.
- [16] Liu B, Haftka RT, Watson LT. Global-local structural optimization using response surfaces of local optimization margins. *Structural and Multidisciplinary Optimization*, 2004; **27**(5):352–359.
- [17] Kim HM. *Target Cascading in Optimal System Design*, PhD dissertation, University of Michigan, 2001.
- [18] Kim HM, Michelena NF, Papalambros PY, Jiang T. Target cascading in optimal system design. *ASME Journal of Mechanical Design*, 2003; **125**(3):475–480.

- [19] Blouin VY, Lassiter JB, Wiecek MM, Fadel GM. Augmented Lagrangian coordination for decomposed design problems. In *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 30 May - 3 June, 2005.
- [20] Tosserams S, Etman LFP, Rooda JE. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Structural and Multidisciplinary Optimization*, 2007; DOI 10.1007/s00158-006-0077-z. To appear.
- [21] Michelena NF, Park H, Papalambros PY. Convergence properties of analytical target cascading. *AIAA Journal*, 2003; **41**(5):897–905.
- [22] Bertsekas DP. *Nonlinear Programming* (2nd ed., 2nd printing). Athena Scientific, 2003.
- [23] Bazaraa MS, Sherali HD, Shetty CM. *Nonlinear Programming: Theory and Algorithms* (2nd ed.). John Wiley & Sons, 1993.
- [24] Tosserams S, Etman LFP, Papalambros PY, Rooda JE. An augmented lagrangian relaxation for Analytical Target Cascading using the alternating directions method of multipliers. *Structural and Multidisciplinary Optimization*, 2006; **31**(3):176–189.
- [25] Tosserams S, Etman LFP, Rooda JE. An augmented Lagrangian coordination method for distributed optimal design in MDO: Part II examples. *International Journal for Numerical Methods in Engineering*, 2006; submitted.
- [26] Wagner TC. *A general decomposition methodology for optimal system design*. PhD thesis, Univerisy of Michigan, Ann Arbor, 1993.
- [27] Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR. Problem formulation for multidisciplinary optimization. *SIAM Journal of Optimization*, 1994; **4**(4):754–776.
- [28] Bertsekas DP. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.
- [29] Arora JS, Chahande AI, Paeng JK .Multiplier methods for engineering optimization. *International Journal for Numerical Methods in Engineering*, 1991; **32**(7):1485–1525.
- [30] Michalek JJ, Papalambros PY. An efficient weighting update method to achieve acceptable inconsistency deviation in Analytical Target Cascading. *ASME Journal of Mechanical Design*, 2005 **127**(2): 206–214
- [31] Bertsekas DP, Tsitsiklis JN. *Parallel and Distributed Computation*. Prentice-Hall, 1989.
- [32] Fortin M, Glowinski R. *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems*. North-Holland, 1983.
- [33] Gill PE, Murray W, Wright MH. *Practical Optimization*. Academic Press, 1981.
- [34] Censor Y, Zenios SA. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1997.
- [35] Ruszczyński A. On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Research*, 1995 **20**: 634–656
- [36] Bezdek JC, Hathaway RJ. Some notes on alternating optimization. *Lecture Notes in Computer Science*, 2002; **2275**:288–300.
- [37] Kim HM, Chen W, Wiecek MM. Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA Journal*, 2003; **41**(5):897–905.