Struct Multidisc Optim (2009) 39:503–517
DOI 10.1007/s00158-008-0347-z

REVIEW ARTICLE

A classification of methods for distributed system optimization based on formulation structure

S. Tosserams · L. F. P. Etman · J. E. Rooda

Received: 5 June 2008 / Revised: 5 November 2008 / Accepted: 17 November 2008 / Published online: 8 January 2009

Abstract This paper presents a classification of formulations for distributed system optimization based on formulation structure. Two main classes are identified: nested formulations and alternating formulations. Nested formulations are bilevel programming problems where optimization subproblems are nested in the functions of a coordinating master problem. Alternating formulations iterate between solving a master problem and disciplinary subproblems in a sequential scheme. Methods included in the former class are collaborative optimization and BLISS2000. The latter class includes concurrent subspace optimization, analytical target cascading, and augmented Lagrangian coordination. Although the distinction between nested and alternating formulations has not been made in earlier comparisons, it plays a crucial role in the theoretical and computational properties of distributed optimization methods. The most prominent general characteristics for each class are discussed in more detail, providing valuable insights for the theoretical analysis and further development of distributed optimization methods.

Keywords Distributed optimization · Classification · Nested optimization · Alternating optimization · Multidisciplinary design optimization · Multi-level optimization · Bilevel programming

This work is funded by MicroNed, grant number 10005898.

S. Tosserams (✉) · L. F. P. Etman · J. E. Rooda
Department of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
e-mail: s.tosserams@tue.nl

1 Introduction

Distributed optimization is a technique to partition a single, typically large system optimization problem into a number of smaller optimization subproblems. A coordination algorithm is used to drive the subproblem designs towards a solution that is optimal for the original problem.

Distributed optimization approaches are attractive for addressing the challenges that arise in the optimal design of advanced engineering systems (see, e.g., Sobieszczanski-Sobieski and Haftka 1997). The main motivation for the use of distributed optimization is the organization of the design process itself. Since a single designer is not able to oversee each relevant aspect, the design process is commonly distributed over a number of design teams. Each team is responsible for a part of the system, and typically uses specialized analysis and design tools to solve its design subproblems. Distributed optimization methods apply naturally to such organizations since they provide a degree of decision autonomy to the different design teams (Alexandrov 2005). Full disciplinary autonomy can rarely be obtained completely. Instead, the disciplinary design subproblems typically involve some quantities from other disciplines related to the interdisciplinary interaction.

Computational savings may be a second motivation for distributed optimization. Although benefits for local optimization are commonly perceived to be small if present at all (see, e.g., Alexandrov and Lewis 1999), global optimization may benefit substantially from distributed optimization. Computational costs for global optimization algorithms often increase rapidly with the number of design variables. Therefore, solving a number of smaller subproblems is expected to be preferable over solving a single large problem if a good degree of separability can be achieved, provided that the coordination overhead introduced by distributed optimization is (relatively) small (Haftka et al. 1992; Haftka and Watson 2005). Additional benefits are expected when these subproblems can be solved in a parallel computing environment.

Examples of advanced engineering systems can be found in the aerospace and automotive industry, and an emerging field is microelectromechanical systems (MEMS). Typically, such systems are partitioned along disciplinary lines, or along the lines of systems, subsystems, and components. For instance, an aircraft may be partitioned with respect to the various physics involved (mechanics, aerodynamics, control, etc.), or with respect to its structural components (fuselage, wing, tail, panels, spars, ribs, etc.). The former aspect-based partition is considered in the traditional multidisciplinary design optimization (MDO) approaches, while the second object-based partition is often found in multi-level product design formulations. Throughout this article, the term *discipline* is used to refer to a single decision-making element in a partition, which may be a discipline in the classical MDO sense, or a component in an object-based partition.

A large number of distributed optimization methods for engineering design has been proposed during the past three decades. Typically, a central master optimization problem is introduced to coordinate the interactions between the disciplinary subproblems. In the field of MDO, these methods are referred to as multi-level methods. Its counterpart, single-level methods, have centralized decision-making and do not allow design decisions to be made at the disciplinary level. Due to their relative simplicity, single-level methods are well understood; a review can be found in Cramer et al. (1994). Multi-level methods—i.e. distributed optimization methods—offer greater freedom in defining coordination approaches, and therefore this field is far less transparent.

We will refer to the methods with subproblem decision freedom as "distributed optimization methods" since the phrase "multi-level" in the linear and nonlinear programming community strictly refers to nested formulations, see Section 4.5. Similarly, "parallel optimization" may refer to both single-level and multi-level methods. The interested reader is referred to Lasdon (1970), Lootsma and Ragsdell (1988), Bertsekas and Tsitsiklis (1989), Wagner and Papalambros (1993), Wagner (1993), Censor and Zenios (1997) for overviews of parallel optimization methods from the nonlinear programming community.

This article presents a classification of distributed optimization methods from the engineering literature, based on the distinction between nested and alternating formulations. *Nested* methods have a central coordination master problem in which the solution of all disciplinary subproblems is required for an evaluation of the master problem functions. *Alternating* approaches iterate between solving a master coordinating problem and the disciplinary subproblems. Following Alexandrov and Lewis (1999), a further division within each class is made based on maintaining feasibility with respect to design constraints and consistency constraints. The main focus of this article is to give an overview of the major concepts used in distributed optimization approaches, and the role of the coordination approach (nested vs. alternating). In our discussion, we will therefore not focus on individual approaches from the literature, but on the most common ingredients to these coordination methods. This focus on general characteristics clearly reveals similarities and differences between various existing methods.

The distinction between nested and alternating methods has been mentioned before (see, e.g., Balling 2002), but its importance has not been emphasized sufficiently. We put the distinction central in our discussion since it has a large influence on important considerations such as solution equivalence, well-posedness of optimization problems, and convergence behavior. These properties are crucial for analytical and practical evaluation of coordination methods (Alexandrov and Lewis 1999). However, a detailed theoretical or numerical convergence analysis of existing methods is beyond the scope of this paper.

This article is organized as follows. First, the classification criteria are discussed in more detail in Section 2, and the general system optimization problem is introduced in Section 3. The nested formulations are treated in their general form in Section 4, together with the categorization of several existing formulations in this class. The section is concluded with a discussion on the general properties and considerations of nested formulations, and references to other fields in which nested formulations have appeared. These references are far from exhaustive and should not be seen as a review, but are intended to allow researchers to connect to decomposition theory from other fields. Alternating methods are discussed from a similar perspective in Section 5. Finally, we conclude with some summarizing remarks.

2 Classification

The classification of coordination methods presented in this chapter is based on two characteristics:

1. Formulation structure (nested or alternating)
2. Constraint relaxation (none, design and/or consistency)

The first characteristic relates to the structure of the problem formulation that is either nested or alternating. In a nested formulation, evaluation of the master problem objective and constraint functions requires the optimization of all disciplinary subproblems. An alternating formulation iterates between solving a master problem and disciplinary subproblems in a sequential scheme.

The second characteristic indicates which constraints are relaxed during the coordination phase (Alexandrov and Lewis 1999). Distributed optimization methods typically relax certain constraints during coordination, and only enforce feasibility with respect to these constraints at convergence. Constraints that are relaxed are referred to as *open*, and constraints for which feasibility is enforced at every iteration are called *closed*.

We differentiate between the relaxation of design constraints and consistency constraints. *Design* constraints are constraints related to the design problem of a discipline. *Consistency* constraints make sure that variables that appear in more than one subproblem take equal values. A relaxation of the consistency constraints implies that the same variable may take different values in different subproblems. Similarly, relaxation of the design constraints means that some of these constraints can be violated at subproblem solutions. The coordination algorithm is then responsible for driving the violations of the relaxed constraints to zero, thereby obtaining a consistent and feasible design.

Whether constraints are open or closed can be observed from the formulations of the disciplinary subproblems in the following sections. When design constraints are closed, they are included as explicit constraints in the subproblems. Open design constraints are typically relaxed with a penalty function, denoted by ϕ, and appear in the objectives of subproblems. When the consistency constraints are closed, subproblems only have freedom in optimizing their local variables while the variables of the other disciplines are fixed. When the consistency constraints are relaxed (are open), each subproblem is given freedom to deviate from the variable values used in other disciplines through the use of variable copies. A penalty function is then added to a subproblem objective to penalize large differences between the variable copies.

The classification presented in this article differs from the earlier classifications presented by Balling and Sobieszczanski-Sobieski (1996) and Alexandrov and Lewis (1999). The main difference is that we differentiate between formulation structure (nested vs alternating), whereas the aforementioned classifications do not. In the notation of Balling and Sobieszczanski-Sobieski (1996), nested formulations can be represented by $SO[O_1\| \dots \| O_M]$, where SO is the coordinating master problem (system optimization problem), and O_j is the optimization subproblem for discipline j. In their notation, the symbol $\|$ represents parallel execution, and [] represents nested execution. Alternating execution can be represented by the $\Leftrightarrow$ symbol introduced by Balling (2002). For example, a sequence that iterates between a system optimization followed by the parallel solution of the disciplinary subproblems can be formulated as $SO \Leftrightarrow O_1\| \dots \| O_M$.

Finally, we assume that the disciplinary subproblems are responsible for satisfying their own analysis equations.

It is possible to extend the proposed classification by including a third characteristic based on the type of information linking subproblems and master problems. One can think of optimal solutions (i.e. zeroth order information), optimal solutions including sensitivity information (i.e. local approximations), or more global models of the subproblems solutions such as response surfaces, neural networks, or Kriging models (i.e. mid-range or global approximations). Instead of including this characteristic as a discriminating factor, we choose to view approximation concepts as possible enhancements of the basic formulations that exchange only optimal solutions.

3 System optimization problem statement

The general form of a system optimization problem is given by

$$\begin{aligned}
\min_{\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M} \quad & f(\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M) \\
\text{s.t.} \quad & \mathbf{g}_j(\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M) \leq \mathbf{0} \qquad j=0,\dots,M \\
& \mathbf{h}_j(\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M) = \mathbf{0} \qquad j=0,\dots,M
\end{aligned} \tag{1}$$

where $\mathbf{y}$ is the vector of linking variables, and $\mathbf{x}_j$ is the vector of local variables associated with discipline

j. The overall objective is denoted by f, constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ are system-level constraints, and $\mathbf{g}_j$ and $\mathbf{h}_j$, $j = 1, \dots, M$ are local to subsystem j. M denotes the number of subsystems, and we assume that a feasible solution to (1) exists.

Many methods have been proposed for a subclass of Problem (1) given by

$$\begin{aligned} \min_{\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M} \quad & f_0(\mathbf{y}, \mathbf{r}_1(\mathbf{y}, \mathbf{x}_1), \dots, \mathbf{r}_M(\mathbf{y}, \mathbf{x}_M)) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\ & \mathbf{g}_0^c(\mathbf{y}, \mathbf{r}_1(\mathbf{y}, \mathbf{x}_1), \dots, \mathbf{r}_M(\mathbf{y}, \mathbf{x}_M)) \leq \mathbf{0} \\ & \mathbf{h}_0^c(\mathbf{y}, \mathbf{r}_1(\mathbf{y}, \mathbf{x}_1), \dots, \mathbf{r}_M(\mathbf{y}, \mathbf{x}_M)) = \mathbf{0} \\ & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \qquad j = 1, \dots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \qquad j = 1, \dots, M \end{aligned} \tag{2}$$

where the system-level constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ only depend on the linking variables $\mathbf{y}$. System-level objective f_0 and coupling constraints $\mathbf{g}_0^c$ and $\mathbf{h}_0^c$ may depend on the linking variables $\mathbf{y}$ and a number of disciplinary responses $\mathbf{r}_j$ that depend only on the linking variables $\mathbf{y}$ and the disciplinary variables $\mathbf{x}_j$ of the associated discipline. The disciplinary responses typically represent generalized properties of disciplines (e.g. total mass or power consumption), and their number is therefore assumed to be much smaller than the number of local variables $\mathbf{x}_j$.

The above formulation can be reformulated to a more convenient form by introducing auxiliary variables for the disciplinary responses, together with a number of equality constraints relating these auxiliary variables to the original functions. For each $\mathbf{r}_j(\mathbf{y}, \mathbf{x}_j)$, introduce variables $\mathbf{t}_j$ and equality constraints $\mathbf{h}_j^r = \mathbf{t}_j - \mathbf{r}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0}$. With $\mathbf{t} = [\mathbf{t}_1, \dots, \mathbf{t}_M]$, Problem (2) can then be written as

$$\begin{aligned} \min_{\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M,\mathbf{t}} \quad & f_0(\mathbf{y}, \mathbf{t}) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\ & \mathbf{g}_0^c(\mathbf{y}, \mathbf{t}) \leq \mathbf{0} \\ & \mathbf{h}_0^c(\mathbf{y}, \mathbf{t}) = \mathbf{0} \\ & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \qquad j = 1, \dots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \qquad j = 1, \dots, M \\ & \mathbf{h}_j^r = \mathbf{t}_j - \mathbf{r}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \qquad j = 1, \dots, M \end{aligned} \tag{3}$$

If we now redefine the linking variables as $\mathbf{y} = [\mathbf{y}, \mathbf{t}]$, the system-wide constraints $\mathbf{g}_0(\mathbf{y}) = [\mathbf{g}_0(\mathbf{y}), \mathbf{g}_0^c(\mathbf{y}, \mathbf{t})]$ and $\mathbf{h}_0(\mathbf{y}) = [\mathbf{h}_0(\mathbf{y}), \mathbf{h}_0^c(\mathbf{y}, \mathbf{t}))]$, and the local constraints $\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = [\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j), \mathbf{h}_0^r(\mathbf{y}, \mathbf{x}_j, \mathbf{t}_j)$, then Problem (3) reduces to the convenient form

$$\begin{aligned} \min_{\mathbf{y},\mathbf{x}_1,\dots,\mathbf{x}_M} \quad & f_0(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\ & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \qquad j = 1, \dots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \qquad j = 1, \dots, M \end{aligned} \tag{4}$$

Since this problem covers only a subclass of the problems of (1), formulations can also be classified according to which type of problem they can be applied. Although we do not explicitly include this in our classification, we will refer to these problem classes where appropriate.

4 Nested formulations

Nested methods typically apply to problems of the form (4), and reformulate it into a bilevel optimization problem. This bilevel problem consists of a top-level master problem in $\mathbf{y}$, and M lower-level disciplinary subproblems in $\mathbf{x}_j$ that are solved for fixed $\mathbf{y}$. Effectively, the local variables $\mathbf{x}_j$ are eliminated from the top-level problem. Since fixing $\mathbf{y}$ in the subproblems separates the local constraint sets, all lower-level problems can be solved in parallel. The bilevel formulation is called nested since at each iteration of the master problem, all lower-level disciplinary design problems have to be solved.

4.1 Closed design, closed consistency

The first category in this class does not relax any of the constraints in (4). The resulting bilevel optimization problem is given by

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\ & \mathbf{y} \in \mathscr{D}_j \qquad j = 1, \dots, M \end{aligned}$$

$$\text{where } \mathscr{D}_j = \left\{ \mathbf{y} | \exists \mathbf{x}_j \text{ s.t. } \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0},\ \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \right\} \tag{5}$$

The master problem tries to find a $\mathbf{y}$ that minimizes the system objective function subject to the system constraints, while $\mathbf{y}$ must be in the sets $\mathscr{D}_j$, $j = 1, \dots, M$. Set $\mathscr{D}_j$ is defined as those values of $\mathbf{y}$ for which discipline j can find a feasible solution in $\mathbf{x}_j$. Determining whether such a $\mathbf{x}_j$ exists (i.e. executing the "where" statement), requires the solution of a disciplinary optimization problem that tries to find a $\mathbf{x}_j$ that satisfies $\mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0}$ and $\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0}$.

The closed design/closed consistency classification is reflected in the above formulation. All design constraints are included as explicit constraints in the subproblems, and therefore have to be satisfied for any subproblem solution. Since the linking variable $\mathbf{y}$ set by the master problem appears as a fixed parameter in the subproblems, consistency between variable values is enforced.

Benders' decomposition method (Benders 1962) and the method presented in Lootsma and Ragsdell (1988) are examples of closed design and closed consistency formulations. The authors are not aware of other, more recent distributed optimization formulations in this category. Likely reasons for the lack of formulations is the difficulty associated with those values of $\mathbf{y}$ for which no feasible subproblem solution exists.

4.2 Closed design, open consistency

To assure that subproblems always have a feasible solution, consistency between disciplines can be relaxed by defining the modified bilevel problem

$$\begin{aligned} \min_{\mathbf{y}} \ & f(\mathbf{y}) \\ \text{s.t. } & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\ & \mathbf{c}_j(\mathbf{y} - \mathbf{y}_j^*(\mathbf{y})) = 0, \qquad j = 1, \ldots, M \end{aligned}$$

$$\begin{aligned} \text{where } \mathbf{y}_j^*(\mathbf{y}), \mathbf{x}_j^*(\mathbf{y}) = \ & \underset{\mathbf{y}_j, \mathbf{x}_j}{\operatorname{argmin}} \ \phi_j(\mathbf{y} - \mathbf{y}_j) \\ & \text{s.t. } \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \\ & \qquad \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \end{aligned} \tag{6}$$

In this formulation, local copies $\mathbf{y}_j$ of the linking variables $\mathbf{y}$ are introduced at each disciplinary subproblem. These copies are added to the set of lower-level variables to assure that subproblems always have a feasible solution. The objective of the subproblems is then to minimize the function $\phi_j(\mathbf{y} - \mathbf{y}_j)$ that penalizes inconsistencies between the master-level $\mathbf{y}$ and the lower-level $\mathbf{y}_j$. Such inconsistencies arise for values of $\mathbf{y}$ for which no feasible $\mathbf{x}_j$ exists. At the master problem, consistency constraints $\mathbf{c}_j(\mathbf{y} - \mathbf{y}_j^*(\mathbf{y})) = \mathbf{0}$ are included to assure consistency between the linking variable copies $\mathbf{y}_j^*$ computed at the subproblems and the original linking variables $\mathbf{y}$. Since these consistency constraints may only be satisfied at convergence of the master problem, the formulation is open with respect to the consistency constraints. The formulation is closed with respect to the design constraints since these have to be satisfied at each subproblem.

Collaborative optimization (Braun 1996; Braun et al. 1997) is an example of the above formulation, and uses $\mathbf{c}_j = \phi_j = \|(\mathbf{y} - \mathbf{y}_j^*)\|_2^2$. Another example is the penalty decomposition formulation of DeMiguel and Murray (2006) that uses a penalty approach to handle the consistency constraints $\mathbf{c}_j$ of the master problem. DeMiguel and Nogales (2008) present an interior point variant of this method that allows inexact solution of the subproblems.

The BLISS2000 formulation of Sobieszczanski-Sobieski et al. (2003) is partly closed and partly open with respect to the interdisciplinary consistency constraints, since local copies are only introduced for a subset of the linking variables $\mathbf{y}$. In their formulation, the set of linking variables $\mathbf{y}$ is divided into a set of "global" variables $\mathbf{z}$ and a set of analysis input-output variables $\mathbf{y}_{\text{io}}$. Local copies $\mathbf{y}_j$ are only introduced for $\mathbf{y}_{\text{io}}$, while global variables $\mathbf{z}$ are fixed at the subproblems. BLISS2000 assumes that the freedom introduced by the copies of $\mathbf{y}_{\text{io}}$ is sufficient to guarantee that a feasible subproblem solution always exists; no system-level constraints are included in the master problem for $\mathbf{z}$. Furthermore, the BLISS2000 formulation includes a set of weights variables $\mathbf{w}$ in the master problem variables, and takes $\mathbf{c}_j = \mathbf{y}_{\text{io}} - \mathbf{y}_j^*(\mathbf{y}_{\text{io}}, \mathbf{z}, \mathbf{w})$, and $\phi_j = \mathbf{w}^T(\mathbf{y}_j - \mathbf{y}_{\text{io}})$. The weights have a fixed value when solving the subproblems.

4.3 Open design, closed consistency

An alternative approach for achieving subproblem feasibility is to relax the design constraints, which gives

$$\begin{aligned} \min_{\mathbf{y}} \ & f(\mathbf{y}) \\ \text{s.t. } & \mathbf{g}_0(\mathbf{y}) \leq 0 \\ & \mathbf{h}_0(\mathbf{y}) = 0 \\ & \phi_j^*(\mathbf{y}) \leq 0, \qquad j = 1, \ldots, M \end{aligned}$$

$$\text{where } \phi_j^*(\mathbf{y}) = \min_{\mathbf{x}_j} \{\phi_j(\mathbf{g}_j(\mathbf{y}, \mathbf{x}_j), \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j))\} \tag{7}$$

In this formulation, the design constraints of the disciplinary subproblems are relaxed through a penalty function ϕ_j, where ϕ_j is chosen such that $\phi_j \leq 0$ implies feasibility with respect to the local constraints $\mathbf{g}_j$ and $\mathbf{h}_j$, and $\phi_j > 0$ implies infeasibility. In the master problem, constraints $\phi_j^*(\mathbf{y}) \leq 0$ are introduced to assure subproblem feasibility. Constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ that do not depend on $\mathbf{y}$ need not be relaxed. Since the subproblems do not have any freedom in changing the value of the linking variables $\mathbf{y}$, the formulation is closed with respect to the consistency constraints. The design constraints are not explicitly enforced but appear in the subproblem objectives through the penalty function ϕ. Hence, the design constraints may be violated for some values of $\mathbf{y}$, and are therefore open.

The quasiseparable decomposition (QSD) approach of Haftka and Watson (2005) follows the above approach by taking $\phi_j = \max(\mathbf{g}_j(\mathbf{y}, \mathbf{x}_j), |\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j)|)$. The linear decomposition method (OLD) of Sobieszczanski-Sobieski et al. (1985) uses $\phi_j = \|\mathbf{g}_j^+(\mathbf{y}, \mathbf{x}_j)\|_2^2 + \|\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j)\|_2^2$, with $\mathbf{g}_j^+$ being the component-wise maximum of $\mathbf{g}_j$ and $\mathbf{0}$: $\mathbf{g}_j^+ := \max(\mathbf{0}, \mathbf{g}_j)$.

4.4 Open design, open consistency

The remaining approach in this class relaxes both disciplinary design constraints and interdisciplinary consistency constraints

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq 0 \\ & \mathbf{h}_0(\mathbf{y}) = 0 \\ & \phi_j^*(\mathbf{y}) \leq 0, \qquad j = 1, \ldots, M \end{aligned}$$

$$\text{where } \phi_j^*(\mathbf{y}) = \min_{\mathbf{y}_j, \mathbf{x}_j} \{\phi_j(\mathbf{y} - \mathbf{y}_j, \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j), \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j))\} \tag{8}$$

where $\phi_j \leq 0$ if subproblem j satisfies its design constraints as well as its consistency constraints. Since the subproblems have the freedom in the linking variables $\mathbf{y}_j$ and the design constraints are relaxed, the formulation is open with respect to both the consistency and the design constraints. Examples of this approach are given by Balling and Sobieszczanski-Sobieski (1995) and Balling and Sobieszczanski-Sobieski (1996) that investigate the use of norms for ϕ_j.

4.5 General characteristics of nested formulations

An advantage of nested formulations is that, under suitable assumptions, a local solution to the master problem can be proven to be a local solution to the original Problem (4) without convexity assumptions. Examples of analyses of solution equivalence are given by Haftka and Watson (2005) and Haftka and Watson (2006) for their QSD approach, and DeMiguel and Murray (2006) for their penalty decomposition approaches.

Local convergence to these optimal solutions may however be complicated by the nondifferentiability or ill-posedness of the master problem constraints. The main cause for nondifferentiability is the typically non-smooth dependence of the subproblem solutions $\mathbf{y}_j^*$ or ϕ_j^* on the master problem variables $\mathbf{y}$ due to constraint activity changes at the disciplinary subproblems (see Vanderplaats and Yoshida 1985; Lootsma and Ragsdell 1988). Since optimal solutions often lie on constraint boundaries, these non-smooth transitions typically occur at optimal solutions to the master problem. Solutions located at such a point of nondifferentiability violate the regularity condition, and therefore the Karush-Kuhn-Tucker (KKT) conditions for optimality do not hold. In general, the KKT conditions do not hold at master problem solutions if the master problem constraints depend linearly on optimal subproblem solutions $\mathbf{y}_j^*$ or ϕ_j^*, or include them in l_1 or l_∞ (i.e. maximum) norms. Examples of methods in this category are collaborative optimization (more specifically, the CO_1 formulation that uses l_1 norms, Braun 1996), quasiseparable decomposition (l_∞, Haftka and Watson 2005), and BLISS2000 (linear dependence on $\mathbf{y}_j^*$, Sobieszczanski-Sobieski et al. 2003).

Alternatively, any approach that uses l_2 norms produces differentiable constraints, but a master problem whose constraint gradients vanish at feasible designs. Hence, the master problem is ill-posed and, again, the KKT conditions do not hold at its solutions. Existing methods in this category are collaborative optimization (the CO_2 formulation, Braun 1996), and optimization by linear decomposition (Sobieszczanski-Sobieski et al. 1985).

For an extensive treatment of the collaborative optimization formulation in this context, the reader is referred to the studies of DeMiguel and Murray (2000), Alexandrov and Lewis (2002), and Lin (2004). A discussion on the use of different norms in the master problem constraints and their properties is given by Balling and Sobieszczanski-Sobieski (1995).

The penalty decomposition approach of DeMiguel and Murray (2006) is, to the best of our knowledge, the only existing formulation that explicitly tackles the non-differentiability and ill-posedness of the master problem. In their approach, the consistency constraints of the master problem are relaxed with a penalty function, and fast local convergence to master problem solutions has been proven.

A second source of difficulties may occur when the number of auxiliary equality constraints $\mathbf{c}$ introduced at the master problem is larger than the number of master problem variables. Especially algorithms that rely on sequential approximation of the master problem constraints may fail to find a feasible solution to the approximated problem since the number of equality constraints is larger than the number of variables.

Convergence speed of all nested formulations depends on two factors: the cost of solving the disciplinary subproblems, and the cost for restoring coupling by solving the master problem. In general, distributed optimization problems are intended for problems with a narrow coupling bandwidth, i.e. the number of linking variables $\mathbf{y}$ is small compared to the number of local variables $\mathbf{x}_j$. For these problems, the cost of restoring coupling is expected to be small, and the gains via parallelization of subproblem solutions are expected to pay off (see, e.g., Sobieszczanski-Sobieski and Haftka 1997).

The use of efficient gradient-based algorithms for solving the master problem may be complicated by the aforementioned complications associated with nondifferentiability and ill-posedness of the master problem. To overcome these difficulties, less efficient algorithms that do not require (constraint) gradients have to be used. These typically inefficient algorithms may require many subproblem optimizations to obtain accurate solutions.

To reduce the number of computationally intensive subproblem optimizations, approximation concepts have been introduced for various formulations. The general idea is to have the master problem solution algorithms operate on approximations of the subproblem solutions, rather than explicitly solving the subproblems each time their solution is required. The search at the master problem is conducted at the approximate level, which is expected to reduce the number of detailed subproblem optimizations. Another motivation of using approximations is that optimal disciplinary designs can be computed beforehand, after which the approximations can be used in different master problems without having to repeat the disciplinary optimizations, as demonstrated in Kaufman et al. (1996) and Liu et al. (2004).

Two types of approximations have appeared in combination with distributed optimization formulations: single-point and multi-point approximations. Single-point methods use post-optimal sensitivity information to construct linear approximations of subproblem optimal solutions. It should be noted that the underlying gradients may still be non-smooth, and local approximation techniques may experience difficulties accordingly. Examples of approaches that use single-point approximations are OLD (Sobieszczanski-Sobieski et al. 1985) and the sequential linearization approach of Vanderplaats et al. (1990).

Multi-point methods typically construct smooth approximations of optimal subproblem solutions by fitting a model to solutions computed for a range of master problem variable values. These models can for example be regression models, neural networks, or Kriging models. The smoothness of these approximations allows efficient gradient-based algorithms to be used for solving the master problem. However, the approximations introduce inconsistencies between the actual subproblem solutions and their models used by the master problem that has to be controlled. What is more, constructing an accurate, high-dimensional approximation to a subproblem optimal solution is highly non-trivial for most engineering problems. Multi-point approximations have been proposed for CO by Sobieski and Kroo (2000) and Zadeh et al. (2008), for BLISS2000 by Sobieszczanski-Sobieski et al. (2003), and for QSD by Haftka and Watson (2005) and Liu et al. (2004).

4.6 Nested formulations in other application fields

Nested formulations for distributed optimization have appeared in other application fields. In general, nested formulations are referred to as bilevel or multilevel programming problems. In operations research (OR), examples are problems dealing with toll setting and congestion management in traffic networks, ticket pricing and seat allocation in the airline industry, and game theory. Other bilevel programming applications are chemical or physical problems involving equilibria conditions. Relatively recent examples of the use of bilevel programming theory in distributed engineering design are given by Lewis and Mistree (1998), Chanron et al. (2005). Examples of overviews of bilevel programming theory and methods from the nonlinear programming community are given by Vicente and Calamai (1994), Migdalas et al. (1997), Colson et al. (2007). In general, bilevel programming problems arise from opposing objectives between the master problem (maximize profit, often the objective of suppliers) and the subproblems (minimize cost, local objective of consumers).

5 Alternating formulations

In this section, we turn our attention to the second class of formulations: alternating methods. Where nested formulations have subproblem optimizations within the constraints of the master problem, alternating formulations iterate between solving the solution master problem and the solution of the subproblems.

5.1 Closed design, closed consistency

This first alternating approach defines a master problem that solves Problem (1) for the linking variables $\mathbf{y}$, and M subproblems that each solve for one set of disciplinary variables $\mathbf{x}_j$. The master problem in $\mathbf{y}$ is given by

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) \\ \text{s.t.} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) \leq \mathbf{0} \qquad j = 0, \ldots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) = \mathbf{0} \qquad j = 0, \ldots, M \end{aligned} \tag{9}$$

The remaining variables $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are fixed during optimization of the master problem. Since the objective and constraint functions all depend on $\mathbf{y}$, they have to be included in the master problem.

Disciplinary subproblem j aims to find $\mathbf{x}_j$ that minimizes f while satisfying all disciplinary constraints $\mathbf{g}_i$ and $\mathbf{h}_i$, $i = 0, \ldots, M$ and fixing $\mathbf{y}$ and the $\mathbf{x}_i | i \neq j$ of the remaining subproblems

$$\begin{aligned} \min_{\mathbf{x}_j} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) \\ \text{s.t.} \quad & \mathbf{g}_i(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) \leq \mathbf{0} \qquad i = 0, \ldots, M \\ & \mathbf{h}_i(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) = \mathbf{0} \qquad i = 0, \ldots, M \end{aligned} \tag{10}$$

Again, all functions that depend on $\mathbf{x}_j$ and have to be included in the subproblem.

The above formulation has closed design constraints, since all design constraints are explicitly included in the subproblems. Subproblems do not have freedom to search in the space of the design variables of other subproblems, and therefore the formulation is closed with respect to the consistency constraints.

A drawback of the above formulation is that, although subproblems are formulated in disciplinary subsets of variables, the constraint models of every other discipline have to be considered when solving the disciplinary subproblems. Hence, the above formulation does not provide the degree of disciplinary autonomy desired for engineering design. A similar difficulty occurs even for problems of the form (4) in which constraints only depend on $\mathbf{y}$ and a single subset $\mathbf{x}_j$. The disciplinary subproblems then only include their disciplinary constraints. However, the master problem still has to include all disciplinary constraints since they all depend on $\mathbf{y}$. This unattractive feature may explain why multi-level formulations in this category are rare.

An additional difficulty of the above formulation is the non-separability of the constraints sets; constraints of each subproblem may depend on variables of other subproblems as well. Local convergence proofs for alternating optimization approaches as found in Bertsekas and Tsitsiklis (1989), Grippo and Sciandrone (2000), and Bezdek and Hathaway (2002) assume that the constraints of a subproblem are separable such that they do not depend on the variables of other subproblems. In other words, theoretical convergence proofs require that subproblem constraint sets are separable with respect to the subproblems. Note that the objective function need not be separable. The results of Pan and Diaz (1990) confirm these findings by showing that the above formulation gets stuck in non-optimal points even for a two-dimensional linear programming problem.

An approach that has been proven to be convergent for convex problems is hierarchical overlapping coordination (HOC, Park et al. 2001). The HOC convergence analysis assumes that (at least) two problem partitions with disjoint linking variable sets $\mathbf{y}$ are available. In other words, a linking variable in one partition cannot be a linking variable in the other. This condition appears impractical in engineering design where a partition is typically static, and linking variables emerge naturally when partitioning the system.

5.2 Closed design, open consistency

Penalty relaxation of non-separable constraints is an often used technique to arrive at subproblems with separable constraints. Approaches that relax interdisciplinary consistency through penalty functions typically apply to a subclass of Problem (1) in which the disciplinary constraints $\mathbf{g}_j$ and $\mathbf{h}_j$, $j = 1, \ldots, M$ depend only on the linking variables $\mathbf{y}$, and the design variables $\mathbf{x}_j$ of one discipline, and system-level constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ depend only on the linking variables $\mathbf{y}$. For these problems, constraints are linked through the linking variables $\mathbf{y}$. These problems are similar to Problem (4), but allow the objective function to also depend on the local variables.

To remove the coupling of constraints, local copies $\mathbf{y}_j$ of the linking variables $\mathbf{y}$ are introduced at each subproblem. Non-separable consistency constraints $\mathbf{c}_j = \mathbf{y} - \mathbf{y}_j = \mathbf{0}$, $j = 1, \ldots, M$ are included to force the copies to be equal to the originals. These consistency constraints are subsequently relaxed using a penalty function ϕ_j resulting in a master problem given by

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \sum_{j=1}^{M} \phi_j(\mathbf{y} - \mathbf{y}_j) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \end{aligned} \tag{11}$$

and M disciplinary subproblems given by

$$\begin{aligned} \min_{\mathbf{y}_j, \mathbf{x}_j} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \phi_j(\mathbf{y} - \mathbf{y}_j) \\ \text{s.t.} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \end{aligned} \tag{12}$$

The above formulation is open with respect to the consistency constraints since $\mathbf{y}$ and $\mathbf{y}_j$ may take different values during coordination. Disciplinary constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ are explicitly enforced, and the formulation is therefore closed with respect to the design constraints.

An example of the above formulation is the augmented Lagrangian decomposition method for quasiseparable problems presented by the authors (Tosserams et al. 2007). This formulation uses an augmented Lagrangian penalty function for ϕ_j, and assumes a separable objective function that can be written as $f = \sum_{j=1}^{M} f_j(\mathbf{y}_j, \mathbf{x}_j)$ such that each subproblem minimizes its own component f_j. Since the formulation also assumes that $\mathbf{g}_0$ and $\mathbf{h}_0$ are not present, the master problem reduces to the minimization of the convex quadratic augmented Lagrangian penalty function to which an analytical solution is available. Tosserams et al. (2008b) show that a similar approach also applies to problems with block-separable system constraints $\mathbf{g}_0$ and $\mathbf{h}_0$; the master problem becomes a quadratic programming problem.

Enhanced Collaborative Optimization (ECO, Roth and Kroo 2008) is a formulation similar to the augmented Lagrangian method for quasiseparable problems of Tosserams et al. (2007), except that the subproblem for discipline j includes linearized constraint models of the remaining disciplines $i \neq j$. Furthermore, ECO uses a quadratic penalty function for ϕ_j.

The analytical target cascading (ATC) formulation is another example in this category, but now formulated for an arbitrary number of levels (Michelena et al. 1999; Kim 2001; Kokkolaras et al. 2002; Kim et al. 2003; Michelena et al. 2003). Various penalty functions ϕ_j have been proposed for ATC (see Michelena et al. 2003; Lassiter et al. 2005; Kim et al. 2006; Tosserams et al. 2006), and a comparison can be found in Li et al. (2008).

5.3 Open design, closed consistency

An alternative formulation with separable subproblem constraint sets does not relax consistency, but relaxes the design constraints. This formulation can be applied to the general system optimization problem (1). The master problem is given by

$$\min_{\mathbf{y}} \; f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \sum_{j=0}^{M} \phi_j(\mathbf{g}_j(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M), \mathbf{h}_j(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M)) \quad (13)$$

and the M disciplinary subproblems are given by

$$\min_{\mathbf{x}_j} \; f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \sum_{i=0}^{M} \phi_i(\mathbf{g}_i(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M), \mathbf{h}_i(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M)) \quad (14)$$

In this formulation, all design constraints may be violated, and are therefore open. Consistency is closed since a subproblem does not have freedom to search in the variable space of other subproblems.

An unattractive feature of the above formulation is that any convergence proof requires that each subproblem has to include the constraint functions of all other subproblems as well. This conflicts with the main goal of distributed optimization methods that aim to establish a degree of disciplinary autonomy between the subproblems that does not require the inclusion of models from other subproblems. Note that even if we restrict ourselves to problems of the form (4), then the master problem still includes the constraint models of all subproblems. Each disciplinary subproblem however only includes its own constraint functions.

A popular approach to avoid including all constraint models in each subproblem is to relax some constraints completely. This way, some constraints can be removed from the master problem or the subproblem. For example, setting $\phi_j = 0$ for $j = 1, \ldots, M$ in master problem (13) removes the penalties on subproblem constraints from its formulation. Similarly, subproblems (14) include only their disciplinary constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ by setting the penalties $\phi_i, i \neq j$ for the remaining constraints set equal to zero. Moreover, the disciplinary constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ are included as explicit constraints in subproblem j, thus without any relaxation. This combination results in a mixed formulation with subproblems that have some design constraints relaxed (completely removed, actually), and some enforced explicitly.

Multidisciplinary design optimization with independent subspaces (MDOIS) of Shin and Park (2005) is an example of the above mixed formulation. The master problem in their formulation only performs a so-called system analysis: it is responsible for finding the values of the interdisciplinary coupling variables $\mathbf{y}$ such that consistency between the analysis input and output variables of all disciplines is enforced. A master problem that only performs a system analysis is obtained when f does not depend on $\mathbf{y}$ and thus drops out of (9), $\mathbf{g}_0$ is not present, and $\mathbf{h}_0$ only includes the interdisciplinary analysis coupling constraints.

Bilevel integrated system synthesis (BLISS, Sobieszczanski-Sobieski et al. 2000) is another example. Their approach assumes that the linking variables $\mathbf{y}$ include a set of analysis input-output variables $\mathbf{y}_{\text{io}}$, and a set of global design variables $\mathbf{z}$. System constraints $\mathbf{g}_0$ are not present, and the constraints $\mathbf{h}_0$ only include interdisciplinary analysis coupling constraints, similar to MDOIS. The master problem in $\mathbf{y}$ is split into two separate problems. First, a system analysis problem searches for values of $\mathbf{y}_{\text{io}}$ that satisfy the system analysis equations $\mathbf{h}_0 = \mathbf{0}$. Second, a system optimization problem is defined in $\mathbf{z}$ to minimize the objective f while fixing all local variables $\mathbf{x}_j$. No constraints are included in this system optimization problem. Instead of using the actual objective function f, BLISS includes a linear approximation $f = \hat{f}(\mathbf{z}, \mathbf{x}_1, \ldots, \mathbf{x}_M)$ of how the objective f depends on the remaining variables $\mathbf{z}, \mathbf{x}_1, \ldots, \mathbf{x}_M$ in (13) and (14). The linear approximation is constructed using the global sensitivity equations (GSE, Sobieszczanski-Sobieski 1990), and post-optimal sensitivities of disciplinary subproblem solutions (Vanderplaats and Yoshida 1985).

A third example is the concurrent subspace optimization (CSSO) formulation of Sobieszczanski-Sobieski (1988) that relaxes only a subset of design constraints. The formulation combines design constraint relaxation at the system level with enforcing (an approximation of) the design constraints at the disciplinary level. Their approach also uses the master problem for performing a system analysis to determine the values of the linking variables $\mathbf{y}$, similar to MDOIS and BLISS. Subproblem j however does not relax the design constraints $\mathbf{g}_i$ and $\mathbf{h}_i$ of other subproblems $i \neq j$. Since inclusion of detailed models of all disciplines in a subproblem is undesirable, CSSO includes linear approximations of a cumulative constraint violation measure for subproblems $i \neq j$ in the formulation of subproblem j instead. A variant of CSSO that does not use a system analysis master problem but considers only disciplinary subproblems has appeared in Shankar et al. (1993). Another modification to CSSO can be found in Bloebaum (1992).

5.4 Open design, open consistency

The final category relaxes both design and consistency constraints. Similar to the closed design, open consistency formulation, copies $\mathbf{y}_j$ of the linking variables are introduced at each subproblem. Interdisciplinary consistency constraints $\mathbf{c}_j = \mathbf{y} - \mathbf{y}_j$ are relaxed using a penalty function ϕ_j. The design constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ are relaxed with a penalty function θ_j. The resulting problem formulation is given by a master problem in $\mathbf{y}$

$$\begin{aligned}\min_{\mathbf{y}} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \sum_{j=1}^{M} \phi_j(\mathbf{y} - \mathbf{y}_j) \\ & + \theta_0(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M), \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M)) \end{aligned} \tag{15}$$

and the M disciplinary subproblems are given by

$$\begin{aligned}\min_{\mathbf{y}_j, \mathbf{x}_j} \quad & f(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M) + \phi_j(\mathbf{y} - \mathbf{y}_j) \\ & + \theta_0(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M), \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_M)) \\ & + \sum_{i=1}^{M} \theta_i(\mathbf{g}_i(\mathbf{y}_j, \mathbf{x}_1, \ldots, \mathbf{x}_M), \mathbf{h}_i(\mathbf{y}_j, \mathbf{x}_1, \ldots, \mathbf{x}_M)) \end{aligned} \tag{16}$$

The formulation is open both for the design constraints and the consistency constraints since all constraints are relaxed, and each subproblem includes a separate copy $\mathbf{y}_j$ for the linking variables.

For the general problem of (1), the above formulation does not provide disciplinary autonomy since the local constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ appear in all subproblems. For partially separable problems of the form (4) however, the formulation does yield a degree of design autonomy at the master problem and subproblems. For these problems, the penalty θ_j of subproblem j does not appear in any other subproblem. Although such an approach resembles the open design, closed consistency formulation of Section 5.3, the open design, open consistency master problem does not include the constraint functions of the disciplinary subproblems. The formulation of Blouin et al. (2005) for such partially separable problems uses the explicit relaxation of design constraints to improve the convexity properties of the subproblems. Their formulation uses an ordinary Lagrangian function for ϕ_j to formulate subproblems that can be solved independently. To improve convergence characteristics for non-convex problems, an augmented Lagrangian penalty is used for θ_j to relax the design constraints.

Another example of the above formulation is the augmented Lagrangian coordination method presented by Tosserams et al. (2008a), which relaxes only a subset of the design constraints. Their formulation applies to problems that have system-level constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ that may depend on all design variables, and disciplinary constraints $\mathbf{g}_j$ and $\mathbf{h}_j$, $j = 1, \ldots, M$ that depend on the linking variables $\mathbf{y}_j$ and one set of local variables $\mathbf{x}_j$. Consistency constraints $\mathbf{c}_j$ are relaxed using an augmented Lagrangian penalty function. The system-level design constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ are relaxed using an augmented Lagrangian penalty as well, resulting in an approach that is open with respect to constraints $\mathbf{g}_0$ and $\mathbf{h}_0$, and closed with respect to $\mathbf{g}_j$ and $\mathbf{h}_j$, $j = 1, \ldots, M$.

5.5 General characteristics of alternating formulations

The main advantage of alternating formulation is that the associated optimization problems are typically smooth and well-posed. Where nested formulations explicitly include the optimal value functions in the constraints, alternating formulations include the optimal solutions as *fixed parameters* in the subproblems. Hence, the smoothness properties of subproblems are similar to those of the original non-decomposed problem. Unlike many nested methods, alternating functions do not introduce any additional sources of nonsmoothness.

The major drawback of alternating methods is that proving convergence to a local system solution requires stringent assumptions on the problem properties. Existing convergence proofs for alternating formulations are typically limited to continuous problems with a smooth objective, and convex constraint sets that are separable with respect to the disciplines (see, e.g., Michelena et al. 2003; Tosserams et al. 2008a; Li et al. 2008). Although separability can often be achieved by requiring that problems are of the form (4), many practical problems have non-convex constraint sets by nature.

These strict assumptions are also observed from the nonlinear programming literature. In this field, the alternating nature of the optimization process is also referred to as block coordinate descent (Bertsekas 2003), alternating minimization (Grippo and Sciandrone 2000; Bezdek and Hathaway 2002), or block-nonlinear Gauss-Seidel or Jacobi methods (Bertsekas and Tsitsiklis 1989). Most of these methods solve subproblems sequentially in a Gauss-Seidel type scheme that updates the optimal solutions as soon as they become available. The block-nonlinear Jacobi implementation solves all subproblems in parallel after which all optimal solutions are exchanged. Schemes that combine sequential and parallel solutions of subproblems are also possible. Figure 1 illustrates these three approaches. The assumptions for local convergence of parallel schemes are typically more strict than those for sequential schemes (Bertsekas and Tsitsiklis 1989), but in both cases problems are required to have convex and separable constraint sets.

Notable exceptions to the strict requirements are the formulation for non-convex problems with separable constraint sets proposed by Blouin et al. (2005), and the parallel variable distribution techniques (PVD) for non-separable convex, and separable non-convex problems proposed in, e.g., Ferris and Mangasarian (1994), Sagastizábal and Solodov (2002). The method of Blouin et al. (2005) uses the separable Lagrangian function to relax the consistency constraints, and therewith arrives at uncoupled subproblems that can be solved in parallel. The disciplinary constraints are then relaxed with an augmented Lagrangian penalty to close the duality gap that may exist for non-convex problems. Since no iterations of the Gauss-Seidel or Jacobi type are required, this approach does not require the convexity assumptions of traditional alternating schemes from the nonlinear programming literature. An external mechanism is however necessary to select the penalty parameters of the relaxation functions.

The key ingredient of parallel variable distribution (PVD) methods is that each of the subproblems is given freedom to change the variables of the remaining subproblems along a certain direction (e.g., a feasible descent or Newton direction), thereby effectively relaxing the consistency between subproblems. After the parallel solution of all subproblems, consistency is restored in a so-called synchronization phase which requires the integration of all constraint models. Note that if the PVD directions are set equal to the all-zero vector, the problems (9) and (10) are obtained, and the synchronization step can be omitted. For this all-zero direction however, convergence can only be guaranteed when constraint sets are convex and separable with

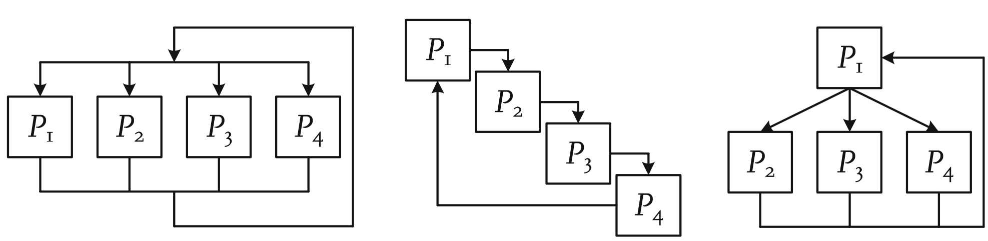

Fig. 1 Alternating optimization schemes: parallel Jacobi that exchanges subproblem solutions at the end of an iteration (*left*), sequential Gauss-Seidel that exchanges solutions as soon as they become available (*center*), and hybrid (*right*)

respect to the subproblems (Sagastizábal and Solodov 2002), which corroborates the stringent assumptions for convergence given earlier in this section.

Computational efficiency of alternating methods is determined by the convergence speed of the alternating minimization schemes (typically in an inner loop), together with the efficiency of any penalty update scheme performed in an additional outer loop. For the inner loop, a sequential scheme is usually more efficient that a parallel scheme since subproblem solutions are exchanged as soon as they become available. Furthermore, sequential schemes are known to be more stable than parallel approaches (Bertsekas and Tsitsiklis 1989). An advantage that can improve the efficiency of parallel schemes is that subproblems can be solved in parallel.

Similar to nested formulations, approximation concepts can be used to speed up the inner loop. A first option is to include gradient information on how the optimal design of one subproblem depends on the design of the other subproblems. With this additional information, a Newton-like fixed point algorithm can be formulated. Gradient information may be obtained through techniques such as post-optimal sensitivities, the global sensitivity equations, or from the iteration history. In the context of multidisciplinary design optimization, gradient information has been used for reducing the cost of performing a system analysis (see, e.g., Haftka et al. 1992). A potential difficulty for a Newton-like approach for alternating formulations is that subproblem solution gradients can be discontinuous due to constraint activity changes and multi-modality. Methods that incorporate gradient information include MDOIS (Shin and Park 2005), BLISS (Sobieszczanski-Sobieski et al. 2000), and CSSO (Sobieszczanski-Sobieski 1988).

A second approach is to construct multi-point approximations using techniques such as linear regression, neural networks, or Kriging. The two drawbacks mentioned at the end of Section 4.5 for the nested approaches also hold for alternating formulations. First, multi-point approximations may introduce inconsistencies between the actual subproblem solutions and their models used by the master problem that has to be controlled. Second, constructing an accurate, high-dimensional approximation to a subproblem optimal solution is non-trivial for many engineering problems. Example of the use of multi-point approximations in alternating methods are given by Kodiyalam and Sobieszczanski-Sobieski (2000) for the BLISS method, and for CSSO by Renaud and Gabrielle (1994).

The second factor in efficiency are the outer loop updates required for the penalty relaxation approach. Although exact penalty methods that do not require updating exist (see, e.g., Bertsekas 2003), they are typically non-smooth, which violates the smoothness assumptions for the alternating inner loop, or require specific gradient information that is difficult to compute for distributed engineering problems. Efficiency improvements may be gained by using penalty approaches that allow the inner loop to be terminated early, as found in e.g. Bertsekas and Tsitsiklis (1989), Bertsekas (2003). Examples of their application in alternating formulations can be found in Tosserams et al. (2006), Li et al. (2008) for ATC, and in Tosserams et al. (2007, 2008a) for augmented Lagrangian coordination.

5.6 Alternating formulations in other application fields

Alternating formulations are commonplace techniques in optimization, and example applications can be found in pattern recognition problems (Bezdek and Hathaway 2002), power unit commitment problems (Beltran and Heredia 2002), and multi-stage stochastic programming (Ruszczynski 1995). The theoretical properties of alternating optimization methods are well understood, and analyses from the nonlinear programming community can be found in (Bertsekas and Tsitsiklis 1989), Bezdek and Hathaway (2002), Grippo and Sciandrone (2000), and Bertsekas (2003). The progressive hedging algorithm of Mulvey and Vladimirou (1991) is an approach similar to the closed design, open consistency alternating formulation presented by Tosserams et al. (2007), and an overview of decomposition methods using alternating optimization in convex stochastic programming can be found in Ruszczynski (1997). In general, alternating methods are employed when the individual subproblems are much easier to solve than the integrated problem that considers all variables simultaneously, a motivation that we recognize from the engineering literature.

6 Summarizing remarks

A classification based on formulation structure of distributed optimization formulations for engineering design is presented. Two classes are identified: nested formulations and alternating formulations. Nested formulations are bilevel programming problems where subproblem solutions are nested in the functions of a coordinating master problem. Alternating formulations iterate between solving a master problem and disciplinary subproblems in a Gauss-Seidel or Jacobi type scheme. A subdivision in each class is made based on

Table 1 Classification of several existing coordination methods

Constraint relaxation		Formulation structure	
Design	Consistency	Nested	Alternating
Closed	Closed	Benders' decomposition (Benders 1962) Lootsma and Ragsdell (1988)	Hierarchical overlapping coordination (Park et al. 2001)
Closed	Open	Collaborative optimization (Braun et al. 1997) BLISS2000 (Sobieszczanski-Sobieski et al. 2003)	Analytical target cascading (Kim 2001) Augmented Lagrangian decomposition (Tosserams et al. 2007) Enhanced collaborative optimization (Roth and Kroo 2008)
Open	Closed	Optimization by linear decomposition (Sobieszczanski-Sobieski et al. 1985) Quasiseparable decomposition (Haftka and Watson 2005)	CSSO (Sobieszczanski-Sobieski 1988) BLISS (Sobieszczanski-Sobieski et al. 2000) MDOIS (Shin and Park 2005)
Open	Open	Balling and Sobieszczanski-Sobieski (1995) Balling and Sobieszczanski-Sobieski (1996)	Blouin et al. (2005) Augmented Lagrangian coordination (Tosserams et al. 2008a)

the relaxation of disciplinary constraints and consistency constraints. Many existing distributed optimization formulations for engineering design are classified according to the criteria, a summary of which is given in Table 1.

For each formulation structure, common characteristics are identified that play a key role in the convergence properties. For example, the KKT conditions do not hold at master problem solutions for many nested formulations. Consequently, gradient-based algorithms may experience difficulties when solving these master problems. For alternating formulations, existing local convergence proofs are available, but require strict assumptions such as convexity and separability of subproblem constraint sets. These common characteristics can be used as a starting point for the detailed theoretical analysis of existing formulations, or as inspiration for developing new ones.

The classification assumes that optimization subproblems are solved to sufficient accuracy. Recent efforts have also considered inexact subproblem solutions. For example, DeMiguel and Nogales (2008) perform only a single Newton step at each subproblem, and Sagastizábal and Solodov (2002) solve quadratic approximations to the subproblems. Although these methods are developed from a nested or alternating formulation as a starting point, their implementations may become very alike, or even identical.

Our classification focusses on the common theoretical properties of formulations. In practice, computational efficiency and robustness are other important factors in determining the applicability of a formulation (Alexandrov and Lewis 1999). The classification may be used as a starting point for a numerical comparison study of the various formulations. For (discussions on) numerical comparisons, the reader is referred to Balling and Sobieszczanski-Sobieski (1995, 1996), Balling and Wilkinson (1997), Alexandrov and Lewis (1999), Perez et al. (2004), Yi et al. (2008), and the references therein. Since specifying various problem decompositions for a number of coordination approaches is tedious and error prone, a flexible and user-friendly approach is desired for specification of problem partitions, and implementation of coordination methods. The works of Alexandrov and Lewis (2004a, b), Etman et al. (2005), Tedford and Martins (2006), de Wit and van Keulen (2008) are efforts in this direction.

Acknowledgements The authors would like to thank the reviewers, Prof. Haftka of the University of Florida, and Dr. Kokkolaras of the University of Michigan for their valuable comments that helped to improve the paper.

References

Alexandrov NM (2005) Editorial—multidisciplinary design optimization. Optim Eng 6(1):5–7

Alexandrov NM, Lewis RM (1999) Comparative properties of collaborative optimization and other approaches to MDO. In: ASMO UK/ISSMO conference on engineering design optimization. MCB University Press, Bradford, pp 39–46

Alexandrov NM, Lewis RM (2002) Analytical and computational aspects of collaborative optimization for multidisciplinary design. AIAA J 40(2):301–309

Alexandrov NM, Lewis RM (2004a) Reconfigurability in MDO problem synthesis, part 1. In: Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization

conference, AIAA paper 2004-4307, Albany, 30 August–1 September 2004
Alexandrov NM, Lewis RM (2004b) Reconfigurability in MDO problem synthesis, part 2. In: Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper 2004-4308, Albany, 30 August–1 September 2004
Balling RJ (2002) Decomposition methods in structural optimization. In: Burns SA (ed) Recent advances in optimal structural design. ASCE, Reston
Balling RJ, Sobieszczanski-Sobieski J (1995) An algorithm for solving the system-level problem in multilevel optimization. Struct Optim 9(3–4):168–177
Balling RJ, Sobieszczanski-Sobieski J (1996) Optimization of coupled systems: a critical overview of approaches. AIAA J 34(1):6–17
Balling RJ, Wilkinson CA (1997) Execution of multidisciplinary design optimization approaches on common test problems. AIAA J 35(1):178–186
Beltran C, Heredia FJ (2002) Unit commitment by augmented lagrangian relaxation: testing two decomposition approaches. J Optim Theory Appl 112(2):295–314
Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numer Math 4:238–252
Bertsekas DP (2003) Nonlinear programming, 2nd edn, 2nd printing. Athena Scientific, Belmont
Bertsekas DP, Tsitsiklis JN (1989) Parallel and distributed computation. Prentice-Hall, Englewood Cliffs
Bezdek JC, Hathaway R (2002) Some notes on alternating optimization. Lect Notes Comput Sci 2275:288–300
Bloebaum CL (1992) An intelligent decomposition approach for coupled engineering systems. In: Proceedings of the 4th AIAA/NASA/USAF/OAI symposium on multidisciplinary analysis and optimization, AIAA paper 1992-4821, Cleveland, 21–23 September 1992
Blouin VY, Lassiter JB, Wiecek MM, Fadel GM (2005) Augmented Lagrangian coordination for decomposed design problems. In: Proceedings of the 6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, 30 May–3 June 2005
Braun RD (1996) Collaborative optimization: an architecture for large-scale distributed design. PhD Thesis, Stanford University
Braun RD, Moore AA, Kroo IM (1997) Collaborative approach to launch vehicle design. J Spacecr Rockets 34(4):478–486
Censor Y, Zenios SA (1997) Parallel optimization: theory, algorithms, and applications. Oxford University Press, New York
Chanron V, Singh T, Lewis K (2005) Equilibrium stability in decentralized design systems. Int J Syst Sci 36(10):651–662
Colson B, Marcotte P, Savard G (2007) An overview of bilevel programming. Ann Oper Res 153(1):235–256
Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR (1994) Problem formulation for multidisciplinary optimization. SIAM J Optim 4(4):754–776
DeMiguel AV, Murray W (2000) An analysis of collaborative optimization methods. In: Proceedings of the 8th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, AIAA paper 2000-4720, Long Beach, 6–8 September 2000
DeMiguel AV, Murray W (2006) A local convergence analysis of bilevel decomposition algorithms. Optim Eng 7(2):99–133
DeMiguel AV, Nogales FJ (2008) On decomposition methods for a class of partially separable nonlinear programs. Math Oper Res 33(1):119–139
Etman LFP, Kokkolaras M, Hofkamp AT, Papalambros PY, Rooda JE (2005) Coordination specification in distributed optimal design of multilevel systems using the χ language. Struct Multidisc Optim 29(3):198–212
Ferris MC, Mangasarian OL (1994) Parallel variable distribution. SIAM J Optim 4(4):815–832
Grippo L, Sciandrone M (2000) On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Oper Res Lett 26(3):127–136
Haftka RT, Watson LT (2005) Multidisciplinary design optimization with quasiseparable subsystems. Optim Eng 6(1): 9–20
Haftka RT, Watson LT (2006) Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems. Optim Eng 7(2):135–149
Haftka RT, Sobieski J, Padula SL (1992) On options for interdisciplinary analysis and design. Struct Optim 4(2):65–74
Kaufman M, Balabanov V, Burgee SL, Giunta AA, Grossman B, Haftka RT, Mason WH, Watson LT (1996) Variable-complexity response surface approximations for wing structural weight in hsct design. Comput Mech 18(2):112–126
Kim HM (2001) Target cascading in optimal system design. PhD Thesis. University of Michigan
Kim HM, Michelena NF, Papalambros PY, Jiang T (2003) Target cascading in optimal system design. ASME J Mech Des 125(3):474–480
Kim HM, Chen W, Wiecek MM (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. AIAA J 44(10):2197–2207
Kodiyalam S, Sobieszczanski-Sobieski J (2000) Bilevel integrated system synthesis with response surfaces. AIAA J 38(8): 1479–1485
Kokkolaras M, Fellini R, Kim HM, Michelena NF, Papalambros PY (2002) Extension of the target cascading formulation to the design of product families. Struct Multidisc Optim 24(4):293–301
Lasdon LS (1970) Optimization theory for large systems. Macmillan, New York
Lassiter JB, Wiecek MM, Andrighetti KR (2005) Lagrangian coordination and analytical target cascading: solving atc-decomposed problems with lagrangian duality. Eng Optim 6(3):361–381
Lewis K, Mistree F (1998) Collaborative, sequential, and isolated decisions in design. ASME J Mech Des 120(4):643–652
Li Y, Lu Z, Michalek JJ (2008) Diagonal quadratic approximation for parallelization of analytical target cascading. ASME J Mech Des 31(5):1–11
Lin J (2004) Analysis and enhancement of collaborative optimization for multidisciplinary design. AIAA J 42(2):348–360
Liu B, Haftka RT, Watson LT (2004) Global-local structural optimization using response surfaces of local optimization margins. Struct Multidisc Optim 27(5):352–359
Lootsma FA, Ragsdell KM (1988) State-of-the-art in parallel nonlinear programming. Parallel Comput 6(2):133–155
Michelena N, Kim HM, Papalambros PY (1999) A system partitioning and optimization approach to target cascading. In: Proceedings of the 12th international conference on engineering design, Munich, 24–26 August 1999
Michelena NF, Park H, Papalambros PY (2003) Convergence properties of analytical target cascading. AIAA J 41(5): 897–905
Migdalas A, Pardalos PM, Värbrand P (1997) Multilevel optimization: algorithms and applications. Springer, Heidelberg
Mulvey JM, Vladimirou H (1991) Applying the progressive hedging algorithm to stochastic generalized networks. Ann Oper Res 31(1):399–424
Pan J, Diaz AR (1990) Some results in optimization of nonhierarchic systems. ASME J Mech Des 112(3):399–405

Park H, Michelena NF, Kulkarni D, Papalambros PY (2001) Convergence criteria for hierarchical overlapping coordination of linearly constrained convex design problems. Comput Optim Appl 18(3):273–293

Perez RE, Liu HHT, Behdinan K (2004) Evaluation of multidisciplinary optimization approaches for aircraft conceptual design. In: Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper 2004-4537, Albany, 30 August–1 September 2004

Renaud JE, Gabrielle GA (1994) Approximation in nonhierarchic system optimization. AIAA J 32(1):198–205

Roth BD, Kroo IM (2008) Enhanced collaborative optimization: a decomposition-based method for multidisciplinary design. In: Proceedings of the ASME design engineering technical conferences, Brooklyn, New York, 3–6 August 2008

Ruszczynski A (1995) On convergence of an augmented Lagrangian decomposition for sparse convex optimization. Math Oper Res 20(3):634–656

Ruszczynski A (1997) Decomposition methods in stochastic programming. Math Program 79(1–3):333–353

Sagastizábal CA, Solodov MV (2002) Parallel variable distribution for constrained optimization. Comput Optim Appl 22(1):111–131

Shankar J, Ribbens CJ, Haftka RT, Watson LT (1993) Computational study of a nonhierarchical decomposition algorithm. Comput Optim Appl 2(3):273–293

Shin MK, Park GJ (2005) Multidisciplinary design optimization based on independent subspaces. Int J Numer Methods Eng 64(5):599–617

Sobieski IP, Kroo I (2000) Collaborative optimization using response surface estimation. AIAA J 38(10):1931–1938

Sobieszczanski-Sobieski J (1988) Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In: Proceedings of the 2nd NASA air force symposium on advances in multidisciplinary analysis and optimization, NASA CP-3031, Hampton, 28–30 September 1988

Sobieszczanski-Sobieski J (1990) Sensitivity of complex, internally coupled systems. AIAA J 28(1):153–160

Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: Survey of recent developments. Struct Optim 14(1):1–23

Sobieszczanski-Sobieski J, James BB, Dovi AR (1985) Structural and multidisciplinary optimization by multilevel decomposition. AIAA J 23(11):1775–1782

Sobieszczanski-Sobieski J, Agte JS, Sandusky Jr RR (2000) Bilevel integrated system synthesis. AIAA J 38(1):164–172

Sobieszczanski-Sobieski J, Altus TD, Phillips M, Sandusky Jr RR (2003) Bilevel integrated system synthesis for concurrent and distributed processing. AIAA J 41(10):1996–2003

Tedford NP, Martins JRRA (2006) On the common structure of MDO problems: a comparison of architectures. In: Proceedings of the 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA paper 2006-7080, Portsmouth, 6–8 Sepember 2006

Tosserams S, Etman LFP, Papalambros PY, Rooda JE (2006) An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. Struct Multidisc Optim 31(3):176–189

Tosserams S, Etman LFP, Rooda JE (2007) An augmented Lagrangian decomposition method for quasi-separable problems in mdo. Struct Multidisc Optim 34(3):211–227

Tosserams S, Etman LFP, Rooda JE (2008a) Augmented Lagrangian coordination for distributed optimal design in mdo. Int J Numer Methods Eng 73(13):1885–1910

Tosserams S, Etman LFP, Rooda JE (2008b) Block-separable linking constraints in augmented Lagrangian coordination. Struct Multidisc Optim. doi:10.1007/s00158-008-0244-5

Vanderplaats GN, Yoshida N (1985) Efficient calculation of optimum design sensitivity. AIAA J 23(11):1798–1803

Vanderplaats GN, Yang YJ, Kim DS (1990) Sequential linearization method for multilevel optimization. AIAA J 28(2):290–295

Vicente LN, Calamai P (1994) Bilevel and multilevel programming: a bibliography review. J Glob Optim 5(3):291–306

Wagner TC (1993) A general decomposition methodology for optimal system design. PhD Thesis. University of Michigan, Ann Arbor

Wagner TC, Papalambros PY (1993) General framework for decomposition analysis in optimal design. In: Gilmore B, Hoeltzel D, Azarm S, Eschenauer H (eds) Advances in design automation. American Society of Mechanical Engineers, Albuquerque, pp 315–325

de Wit AJ, van Keulen F (2008) Framework for multilevel optimization. In: Proceedings of the 5th China-Japan-Korea joint symposium on optimization of structural and mechanical systems, Seoul, June 2008

Yi SI, Shin JK, Park GJ (2008) Comparison of MDO methods with mathematical examples. Struct Multidisc Optim 35(5):391–402

Zadeh PM, Toropov VV, Wood AS (2008) Metamodel-based collaborative optimization framework. Struct Multidisc Optim. doi:10.1007/s00158-008-0286-8