Issues on Nonlinear Programming for Multidisciplinary Design Optimization (MDO) in Ship Design Framework

EMILIO F. CAMPANA[1], GIOVANNI FASANO[1,2] and DANIELE PERI[1]

(1) Istituto Nazionale per Studi ed Esperienze di Architettura Navale INSEAN,
via di Vallerano 139, 00128 Roma, ITALY
E-mail: (e.campana,g.fasano,d.peri)@insean.it.

(2) Istituto di Analisi dei Sistemi ed Informatica 'A. Ruberti' - IASI, CNR,
viale Manzoni 30, 00185 Roma, ITALY
E-mail: fasano@iasi.rm.cnr.it.

Abstract

This work is concerned with the study and the evaluation of formulations for Multidisciplinary Design Optimization (MDO) problems. The latter topic includes very difficult computational aspects, naturally arising from the simultaneous solution of several related optimization problems. In particular, in the naval applications deep interaction among different *disciplines* (i.e. optimization/feasibility problems) is claimed, in order to provide the solution. Each discipline concurs to give intermediate results to other disciplines, until the convergence of the overall framework is reached. The usual nonlinear programming techniques, including multiobjective optimization methods, seem still inadequate to cope with most of these challenging applications. In this work we first consider some relevant formulations proposed in the literature for MDO. Then, we consider the relationship between MDO and NonLinear Programming (NLP), with specific reference to feasibility and convergence analysis. Preliminary numerical results on a ship design application are included.

Keywords: Multidisciplinary Design Optimization, Nonlinear Programming, Formulation and Feasibility issues, Convergence analysis.

1 Introduction

In the last decades a large number of industrial applications claimed for complex optimization approaches, in order to obtain reliable and effective solutions. Several of these applications (e.g. aircraft and spacecraft engines design) provide challenging problems which have been formulated within MDO frameworks [9, 1]. The latter methodologies substantially include both parallelization and coupling of different optimization or feasibility schemes, i.e. the *disciplines*, involved in the application. This kind of approach is still missing in the framework of the marine design, while the design activities in this field involve naturally several different disciplines, with an high degree of coupling. Under this perspective, marine design would be taking an enormous advantage by the application of MDO.

Unfortunately, the interaction among the standard optimization techniques used within the respective disciplines is non-trivial. This requires a special care for the overall resulting formulation. Nevertheless, the formal and accurate coupling among disciplines is essential, in order to guarantee both the *convergence* and *satisfactory performance* of the overall framework [1]. On one hand, specific attention has to be paid in order to ensure the *correctness* of the MDO formulation in hand. Indeed, a single application often allows different

MDO formulations. However, some of them are *naive* or permit only a poor convergence analysis; hence they should be discarded if theoretical achievements are sought [9]. On the other hand, nonlinear programming often provides powerful and reliable techniques for the solution of standard optimization problems. This suggests that those MDO formulations, which rely on the abilities of optimization methods, may efficiently gain advantage from conventional optimization algorithms [3, 4]. Observe that most of the usual ingredients considered for nonlinear programming (e.g. feasibility, optimality conditions, sensitivity analysis, duality, etc.), are not immediately applicable and require a suitable adaptation to MDO frameworks.

On these guidelines, the first classifications proposed in the literature for MDO problems, often relied on the possibility of managing standard and simple nonlinear programming schemes. This was partially imposed by the huge computational burden usually involved in large scale design problems. In addition, whenever coarse solutions where allowed, for extremely tough practical problems, the coupling among disciplines was weakened. Thus, independent optimization methods could be autonomously applied, so that MDO frameworks reduced to a sequence of mere nonlinear programming schemes, adaptively coordinated by some rules [3]. Unfortunately, the resulting scenarios poorly described several non-convex real problems, which tried engineers' abilities and efforts [16].

In this work we review some issues related to the current formulation of MDO problems. Then we consider more recent papers, where novel MDO formulations are studied. In particular we focus on *bilevel programming* methods, where the original MDO formulation is decomposed into a *system level* problem and a set of *lower level* subproblems. The master (i.e. the system level) problem depends on the optimal solutions of subproblems; conversely each subproblem includes a set of unknowns provided by the master level. We specifically review some theoretical difficulties related to the convergence of MDO bilevel formulations.

Then, we apply our preliminary conclusions to the formulation of a small scale ship design problem, where PDE solvers are included. We finally give evidence that the solution of the resulting optimization problem requires efficient techniques from nonconvex Nonlinear Programming.

2 Preliminaries

In this paper[1] we indicate with

$$\min_{x \in \mathcal{A}} f(x) \tag{2.1}$$

the general mathematical programming problem, where $\mathcal{A} \subseteq \mathbb{R}^n$, $n \geq 1$, is the *feasible set* and $f : \mathbb{R}^n \to \mathbb{R}^q$ is the *objective function.* If $q > 1$ then (2.1) is a *multiobjective* problem, while $f(x) = const.$ for any $x \in \mathcal{A}$, transforms (2.1) into a *feasibility* problem.

A *local minimum* of $f(x)$ on the set $\mathcal{A}$ is a point $x^* \in \mathbb{R}^n$ such that:

$$f(x^*) \leq f(x), \qquad \forall x \in \mathcal{A} \cap B(x^*, \rho),$$

where $B(x^*, \rho) \subset \mathbb{R}^n$ is a ball with center in x^* and radius $\rho > 0$. Similarly, a *global minimum* of $f(x)$ on the set $\mathcal{A}$ is a point $x^* \in \mathbb{R}^n$ such that:

$$f(x^*) \leq f(x), \qquad \forall x \in \mathcal{A}.$$

We define a set of *optimality conditions* for (2.1) as a set of analytical relations satisfied by the so called *stationary points* of $f(x)$ in (2.1). The set of stationary points includes the minima of $f(x)$ on $\mathcal{A}$, but in general may also include other points like local maxima, inflection points, etc. Then, when we report that the point $x^* \in \mathcal{A}$ simply satisfies some optimality conditions, we mean that the latter conditions are only *necessary* for x^* to be a local minimum.

Then, we say that an algorithm for solving (2.1) is *globally convergent* when it generates the (possibly infinite) sequence $\{x_k\} \subset \mathcal{A}$, converging to a *stationary point* of (2.1), regardless of the choice for the initial point x_0. For instance, when $\mathcal{A} \equiv \mathbb{R}^n$ and $f(x)$ is continuously differentiable, the latter definition corresponds to an algorithm which generates the sequence $\{x_k\}$, such that $\liminf \|\nabla f(x_k)\| \to 0$. When further (stronger) assumptions are considered (e.g. the convexity of $f(x)$ in (2.1), the convexity of $\mathcal{A}$, etc.),

[1] The following notation is adopted: $\mathbb{R}^n$ is the real n-dimensional space. We indicate the Euclidean norm of vector x as $\|x\| \doteq < x, x >^{1/2}$, while $\|x\|_\alpha$ represents the general α norm, for any $\alpha > 0$.

it is possible to ensure also *sufficient* conditions for the convergence to a minimum point.
Observe that despite the misleading use of the term '*global*', the definition of global convergence and global minimum are independent, i.e. a globally convergent method could in general yield a local minimum [7].

Without loss of generality we consider in the paper only *minimization problems*. This is not a limitation; indeed, suppose for instance that we have to maximize the objective function $f : \mathbb{R}^n \to \mathbb{R}$ over the set $\mathcal{A}$. Then, by means of the identity $\max_{x\in\mathcal{A}} f(x) = -\min_{x\in\mathcal{A}}[-f(x)]$ we can immediately transform the maximization into an equivalent minimization. On this guideline, we remark that the optimality conditions can be intended for both minimization or maximization problems[2].

3 General aspects of an MDO formulation

As the Section 1 suggests, in general the MDO formulation of a real problem may be not unique. Indeed, an MDO formulation is characterized by the following ingredients:

- each discipline contributes to describe the overall problem;
- each discipline represents an independent problem with its own formulation. This formulation and its solution rely on theoretical results (e.g. optimality conditions, sensitivity analysis, convergence analysis), solution techniques (e.g. solution methods, heuristics, etc.) and possibly use software codes arising from within that discipline;
- the independent formulations of the different disciplines require a suitable unification, to form an overall MDO formulation, which claims for its own theory, solution methods and software packages.

From this scenario several theoretical difficulties naturally arise when dealing with MDO, not to mention severe *feasibility issues*, which arise in different forms for both the discipline and the MDO level (see Section 3.2). Let us describe now a formal representation of an MDO formulation.

Some variables of the overall formulation, which comprehends the formulations of the disciplines, are *design unknowns*, i.e. they have a physical meaning related to real parameters of the ship. On the other hand, in the formulation associated with each discipline, other variables are in general introduced, e.g. *state unknowns* of the discipline, auxiliary parameters, etc., and do not play a direct role in the design problem. More formally, consider the discipline D_i, $i = 1, \dots, p$, and let the pair of vectors $(x_i, s_i) \in \mathbb{R}^{n_i \times m_i}$ be associated with D_i. In particular, s_i is the subvector of unknowns representing the *state* of that discipline D_i (e.g. variables generated by the discretization of a PDE solver), while x_i is the subvector of design unknowns included in the formulation of D_i. With these positions, and including the subvector x_0 of design unknowns shared by the p disciplines, we want to address a formal definition for MDO formulations, by considering the vectors: $x^T = (x_0^T \; x_1^T \; \cdots \; x_p^T) \in \mathbb{R}^n$, $n = n_0 + n_1 + \cdots + n_p$ (design variables), and $s^T = (s_1^T \; \cdots \; s_p^T) \in \mathbb{R}^m$, $m = m_1 + \cdots + m_p$ (state vector).

Assumption 3.1 *Consider a real problem and suppose it involves the disciplines D_i, $i = 1, \dots, p$. Let $x^T = (x_0^T \; x_1^T \; \cdots \; x_p^T) \in \mathbb{R}^n$, $n = n_0 + n_1 + \cdots + n_p$, and $s^T = (s_1^T \; \cdots \; s_p^T) \in \mathbb{R}^m$, $m = m_1 + \cdots + m_p$, suppose that*

1. *the set $B_i \subseteq \mathbb{R}^{n_0 \times n_i \times m}$ exists for the discipline D_i, such that it is defined by the block of nonlinear constraints*
$$B_i = \{(x_0, x_i, s) \in \mathbb{R}^{n_0 \times n_i \times m} : \; g_i(x_0, x_i, s) \geq 0, \quad A_i(x_0, x_i, s) = 0\};$$
2. *the nonlinear function $f_i(x_0, x_i, s)$ exists, with $f_i : \mathbb{R}^{n_0 \times n_i \times m} \to \mathbb{R}^{q_i}$, such that it is always possible to associate the formulation*
$$\min_{(x_0, x_i, s) \in B_i} f_i(x_0, x_i, s_i)$$
to each discipline D_i, $i = 1, \dots, p$;

[2] Note that the Karush-Kuhn-Tacker (KKT) conditions are among the most adopted optimality conditions for nonlinear optimization. However, in the literature they are often referred to minimization problems, so that their application to a maximization problem requires some simple but *careful* changes.

3. *the functions $f(x,s) = \varphi[f_1(x_0,x_1,s),\dots,f_p(x_0,x_p,s)]$ and $g_0(x,s)$ exist, with $f : \mathbb{R}^{n\times m} \to \mathbb{R}^q$ and $g_0 : \mathbb{R}^{n\times m} \to \mathbb{R}$, such that if $B = \{(x,s) \in \mathbb{R}^{n\times m} : g_0(x,s) \geq 0,\ (x,s) \in B_1 \cap \cdots \cap B_p\}$ the real problem may be formulated as*

$$\min_{(x,s)\in B} f(x,s). \tag{3.2}$$

□

We are now ready to give the following general definition for MDO formulations, which will be adopted in this paper, unless differently specified.

Definition 1 *Consider a real problem and suppose the Assumption 3.1 holds; then, we say that (3.2) is a* `nonlinear MDO formulation` *for that problem.* □

Observe that the previous definition is necessary, in order to distinguish between *tractable* MDO problems (i.e. those problems whose formulation can take advantage from the nonlinear programming techniques), and *intractable* MDO problems, whose formulation is unclear or it is not a mathematical program.
If the formulation (3.2) were treated as a nonlinear program, the usual techniques from numerical optimization could be adopted for its solution. Unfortunately, the specific difficulty of (3.2) is in the equality constraints, the so called *MultiDisciplinary Analysis (MDA)* of the feasible set B, given by

$$MDA = \begin{cases} A_1(x_0,x_1,s) = 0 \\ \vdots \\ A_p(x_0,x_p,s) = 0. \end{cases}$$

Indeed, the i-th block of equalities $A_i(x_0,x_i,s) = 0$ may not correspond exactly to a set of nonlinear equations; though it may be a black-box, which only implicitly defines a nonlinear relation among the variables x_0, x_i and s. Moreover, the MDA often corresponds to the discretization of PDE systems, so that the *implicit function theorem* cannot be exploited to retrieve $s = s(x)$. Thus, (3.2) can be hardly reformulated as a nonlinear program uniquely dependent on the design vector x.

We say that an algorithm for solving the formulation (3.2) is *convergent* if it is globally convergent, so that a stationary point (x^*,s^*) is given for (3.2), which satisfies some optimality conditions. The *Karush-Kuhn-Tacker (KKT) conditions* are the most common optimality conditions adopted for the nonlinear program (3.2): they may involve the use of first and second order derivatives [14]. We avoid a detailed description of KKT conditions [6], nonetheless we remind that they often provide analytical conditions which can be fruitfully implemented within convergence frameworks of optimization algorithms. On the other hand, KKT conditions require some assumptions on both the objective function and the feasible set in (3.2). It is not difficult to find simple examples of MDO problems where the latter assumptions do not hold. Hence, this proves the intrinsic difficulty of providing both a complete convergence analysis and effective algorithms for the formulation (3.2).

Finally, we remark that most of the real problems implicitly require box constraints, at least for a subset of the design unknowns. The block of inequalities $g_0(x,s) \geq 0$ in (3.2) also includes the latter constraints.

3.1 MDO reformulations: issues on classification

In order to provide a general classification for nonlinear MDO formulations, a further set of constraints and unknowns has to be introduced in (3.2). In particular, suppose we modify (3.2) as [4]

$$\min_{(x,s,t)\in B'\cap\mathbb{R}^{n\times m\times m}} f'(x,s,t), \qquad B' = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3, \tag{3.3}$$

where the sets of constraints Γ_1, Γ_2, Γ_3 are defined by

$$\textit{Design Constraints:}\quad \Gamma_1 = \begin{cases} g_0(x,s) \geq 0 \\ g_1(x_0,x_1,s) \geq 0 \\ \quad\vdots \\ g_p(x_0,x_p,s) \geq 0 \end{cases}$$

$$\textit{Disciplinary Analysis Constraints (MDA)}\quad \Gamma_2 = \begin{cases} A_1(x_0,x_1,s_1,t_2,\dots,t_p) = 0 \\ \quad\vdots \\ A_p(x_0,x_p,s_p,t_1,\dots,t_{p-1}) = 0 \end{cases}$$

$$\textit{Interdisciplinary Consistency Constraints}\quad \Gamma_3 = \begin{cases} t_1 = \mathcal{C}_1(s_1) \\ \quad\vdots \\ t_p = \mathcal{C}_p(s_p). \end{cases}$$

The Interdisciplinary Consistency Constraints are assumed nonlinear and introduce with respect to (3.2) the new set of unknowns $t^T = (t_1^T \cdots t_p^T)$. For each block $A_i(x_0,x_i,s)$, $i = 1,\dots,p$ in (3.2), they weaken the dependency from the entire vector s of the state unknowns. The latter modification may be suitably appreciated by recalling that in this way, the general blocks $A_i(x_0,x_i,s) = 0$ and $A_j(x_0,x_j,s) = 0$, in the MDA, only share the subvector of unknowns x_0. This particular structure of the formulation could be used to apply optimization algorithms for the nonlinear program (3.3). In particular, either *decomposition techniques* or *multilevel methods* could be advisable in this case. According with a common terminology within the MDO literature, we can say that the set t of auxiliary variables is introduced in order to *decouple the interdisciplinarity* among disciplines.
As previously reported, the formulation (3.3) can be hardly solved with a direct application of standard mathematical programming techniques. Most of the times, optimality conditions, as KKT conditions, cannot be directly used and (3.3) must be reformulated into *more* tractable alternative problem(s), were nonlinear algorithms may be adopted.

Definition 1 *Let the set B' in (3.3) be nonempty, let Z^* be the solution set of the MDO formulation (3.3). We say that $\hat{\mathcal{F}}$ is a* reformulation *of (3.3) if a smooth nonlinear function $\varphi_{\hat{\mathcal{F}}}$ exists such that $\varphi_{\hat{\mathcal{F}}}(\hat{z}^*) \in Z^*$, for any $\hat{z}^* \in \hat{Z}^*$, where $\hat{Z}^*$ is the solution set of $\hat{\mathcal{F}}$.* □

Note that any MDO formulation is also an MDO reformulation with φ given by the identity. Furthermore, the solution(s) obtained for the reformulated problem *are not* in general optimal solutions of (3.3). This opens a serious discussion on all the possible and reliable reformulations for (3.3). Anyway, this is far beyond the purposes of the present paper; thus, we simply introduce the following definition of *equivalence among reformulations.*

Definition 2 *Let $\tilde{\mathcal{F}}$ and $\bar{\mathcal{F}}$ be reformulations of the MDO formulation (3.3). Let $\tilde{Z}^*$ and $\bar{Z}^*$ be the solutions sets of $\tilde{\mathcal{F}}$ and $\bar{\mathcal{F}}$. We say that $\tilde{\mathcal{F}}$ is* equivalent *to $\bar{\mathcal{F}}$ (i.e. $\tilde{\mathcal{F}} \sim \bar{\mathcal{F}}$) if the nonlinear functions $\varphi_{\tilde{\mathcal{F}}}$, $\varphi_{\bar{\mathcal{F}}}$ exist such that $\varphi_{\tilde{\mathcal{F}}}(\tilde{x}^*,\tilde{s}^*,\tilde{t}^*) \in \bar{Z}^*$ and $\varphi_{\bar{\mathcal{F}}}(\bar{x}^*,\bar{s}^*,\bar{t}^*) \in \tilde{Z}^*$, for any $(\tilde{x}^*,\tilde{s}^*,\tilde{t}^*) \in \tilde{Z}^*$ and $(\bar{x}^*,\bar{s}^*,\bar{t}^*) \in \bar{Z}^*$.* □

The latter definition[3] is evidently inoperative, due to the difficulty in computing all the solutions $(\tilde{x}^*,\tilde{s}^*,\tilde{t}^*)$ and $(\bar{x}^*,\bar{s}^*,\bar{t}^*)$ of the MDO reformulations $\tilde{\mathcal{F}}$, $\bar{\mathcal{F}}$. Then, a more qualitative (and realistic) classification for the reformulations of (3.3), may be given according with either of the following *general criteria* (see also [3]).

- *Structural (or Analytical) Perspective*: we give a reformulation of (3.3) which meets a suitable nice structure. On this guideline we usually consider a structure where each discipline may be treated approximately as independent with respect to the others, so that some known results from nonlinear programming may be usefully applied. Consequently, the resulting reformulation is partially decomposed with respect to the disciplines: this requires a *globalization method* for analytically coordinate the intermediate results from each discipline.

[3] It is easy to prove that the equivalence of reformulations introduced in Definition 2 satisfies the standard *Reflexive*, *Symmetric* and *Transitive* properties. Thus, the equivalence relation induces a suitable partition of the set of all possible reformulations for MDO formulation (3.3).

- Algorithmic Perspective: the reformulation of (3.3) is aimed at using as many results, algorithms and packages as possible, from nonlinear programming. For instance, convex or continuously differentiable reformulations would be in general preferable to respectively nonconvex or nonsmooth ones.

Another possible criterion, for classifying the reformulations of nonlinear MDO formulation (3.3), is hinted by the following definition (see [2]).

Definition 3 *Consider the reformulation $\hat{\mathcal{F}}$ of the nonlinear MDO formulation (3.3). We say that $\hat{\mathcal{F}}$ is* closed [open] *with respect to the block Γ_i, $i = 1, 2, 3$, of constraints, if the structure of $\hat{\mathcal{F}}$ assumes that the block Γ_i is* satisfied [not satisfied], *regardless of the optimization algorithm(s) used to solve $\hat{\mathcal{F}}$.* □

Note that in the previous definition, the classification does not rely on the optimization technique(s), which can be used to approach the set of solutions of the MDO reformulation. From Definition 3 we associate to the MDO reformulation $\hat{\mathcal{F}}$ the label

$$\hat{\alpha}\mathrm{D} \ / \ \hat{\beta}\mathrm{DA} \ / \ \hat{\gamma}\mathrm{IC}, \qquad \hat{\alpha}, \hat{\beta}, \hat{\gamma} \in \{\mathrm{O}, \mathrm{C}\}, \tag{3.4}$$

where the possible entries $\{\mathrm{O}, \mathrm{C}\}$ for $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$ stand respectively for '*OPEN*' and '*CLOSE*'. As an example, an MDO reformulation with the label OD/CDA/CIC is *OPEN* with respect to the Design Constraints and *CLOSED* with respect to both the Disciplinary Analysis and the Interdisciplinary Constraints.

3.2 Relevant MDO reformulations from the literature

This section reviews some specific MDO reformulations of (3.3), which are also MDO formulations and are widely adopted in the applications. We urge to remark that the structure of both the objective function and the constraints in (3.3) is strongly dependent on the application in hand. This suggests that a reformulation may be suitable for a specific real problem though, it might be completely inadequate for another application. The latter drawback is intuitively a consequence of the general complexity of nonlinear MDO formulations, where the interdisciplinarity represents both a theoretical and a computational challenge. Here we consider the following MDO reformulations of (3.3), according with the classification suggested by Definition 3.

- *MultiDisciplinary Feasible (MDF).* It is an MDO reformulation of (3.3) also known as *FIO* or *AIO* [9], which represents the most trivial approach to the solution. It consists of using the implicit function theorem to explicit the vectors $s = s(x)$ and $t = t(x)$ from the Disciplinary Analysis and the Interdisciplinary Constraints. Then, the resulting MDO reformulation is simply

$$\begin{array}{ll} \min\limits_{x} & f'(x, s(x), t(x)) \\ & g_0(x, s(x)) \geq 0 \\ & g_1(x_0, x_1, s(x)) \geq 0 \\ & \quad \vdots \\ & g_p(x_0, x_p, s(x)) \geq 0, \end{array} \tag{3.5}$$

  which may be treated as a nonlinear program in the unknown $x \in \mathbb{R}^n$. As previously said, the equality constraints in (3.3) often represent the state equations of the disciplines, hence they strongly affect the difficulty to get a solution. Therefore, the equality constraints in (3.3) can be hardly inverted to provide $s = s(x)$, so that the reformulation (3.5) is quite unlikely in general for the MDO formulation (3.3). According with the pattern (3.4), the MDF scheme is an OD/CDA/CIC reformulation.

- *Simultaneous Analysis and Design (SAD).* Also known with the acronyms *AAO* or *SAND* [11], may be considered as the counterpart of MDF. Indeed, here x, s and t are all variables of the reformulation, so that the overall optimization problem to be solved is

$$\begin{array}{ll} \min\limits_{x,s,t} & f'(x,s,t) \\ & g_0(x,s) \geq 0 \\ & g_1(x_0,x_1,s) \geq 0 \\ & \qquad \vdots \\ & g_p(x_0,x_p,s) \geq 0 \\ & A_1(x_0,x_1,s_1,t_2,\dots,t_p) = 0 \\ & \qquad \vdots \\ & A_p(x_0,x_p,s_p,t_1,\dots,t_{p-1}) = 0 \\ & t_1 = \mathcal{C}_1(s_1) \\ & \qquad \vdots \\ & t_p = \mathcal{C}_p(s_p). \end{array} \tag{3.6}$$

Observe that the number of unknowns for SAD may be relatively larger than in the case of MDF. However the reformulation (3.6) may be treated as a unique nonlinear program, implying the use of theory and algorithms from optimization. From (3.4) and the Definition 3 the SAD scheme is an OD/ODA/OIC reformulation.

- *Distribute Analysis Optimization (DAO).* This is an intermediate approach between the previous two (indeed it is often addressed as the *In Between* [9, 12] reformulation, or alternatively it is the *IDF* approach [9]). Here, a subset of the equality constraints is used to explicit a subvector of the unknowns in terms of the remaining variables. Considering the following partition of vectors $s^T = (\tilde{s}^T \;\; \hat{s}^T)$ and $t^T = (\tilde{t}^T \;\; \hat{t}^T)$, the resulting optimization problem becomes (for simplicity we have compounded the Disciplinary Analysis and the Interdisciplinary Consistency constraints)

$$\begin{array}{ll} \min\limits_{x,\tilde{s},\tilde{t}} & f'\left[x, (\tilde{s}^T \;\; \hat{s}^T(x,\tilde{s}))^T, (\tilde{t}^T \;\; \hat{t}^T(x,\tilde{s}))^T\right] \\ & g_0\left[x, (\tilde{s}^T \;\; \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\ & g_1\left[x_0, x_1, (\tilde{s}^T \;\; \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\ & \qquad \vdots \\ & g_p\left[x_0, x_p, (\tilde{s}^T \;\; \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\ & A\left[x, (\tilde{s}^T \;\; \hat{s}^T(x,\tilde{s}))^T, (\tilde{t}^T \;\; \hat{t}^T(x,\tilde{s}))^T\right] = 0 \\ & \tilde{t} = \tilde{\mathcal{C}}(\tilde{s}). \end{array} \tag{3.7}$$

Finally observe that the DAO scheme is an OD/CDA/OIC reformulation.

- *Optimization by Linear Decomposition (OLD).* Despite the previous MDO reformulations, this is a *bilevel reformulation* [10]. Here the first (*upper*) level of minimization has the role of coordinating the results coming from the second (*lower*) level of minimization, which is the disciplines level. The overall nonlinear program is

$$\begin{array}{ll} \min\limits_{x_0,t} & f'\left[x_0,x_1,\dots,x_p,s_1(x_0,x_1,t),\dots,s_p(x_0,x_p,t)\right] \\ & g_0\left[x_0,x_1,\dots,x_p,s_1(x_0,x_1,t),\dots,s_p(x_0,x_p,t)\right] \geq 0 \\ & m_i(x_0,x_i,t) \leq 0, \qquad i \leq p \\ & \min\limits_{x_i} \;\; m_i(x_0,x_i,t) \\ & \qquad t_i = C_i\left[s_i(x_0,x_i,t)\right], \qquad i \leq p \end{array} \tag{3.8}$$

where

$$m_i(x_0, x_i, t) = \left\| g_i^+ \left[x_0, x_i, s_i(x_0, x_i, t)\right] \right\|^2$$

$$g_i^+ \left[x_0, x_i, s_i(x_0, x_i, t)\right] = \min \left\{0, g_i \left[x_0, x_i, s_i(x_0, x_i, t)\right]\right\},$$

and the last equality is intended componentwise. Observe that here the subvector $s_i(x_0, x_i, t)$ is made explicit from the i-th block of constraints $A_i(x_0, x_i, t_1, \ldots, t_{i-1}, s_i, t_{i+1}, \ldots, t_p) = 0$. The function $m_i()$ (*discrepancy function* [4]) in both the upper and lower level of (3.8) substantially measures a penalization of the infeasibility. Moreover, the exponent 2 in its definition yields a continuously differentiable objective function for the lower level.

- *Collaborative Optimization (CO).* Likewise the previous case, CO provides a multilevel optimization reformulation [8], where it is more clear the different role played by the system level and the disciplines level. In the following scheme we allow the dependency of the Interdisciplinary Constraints from both the design variables and the state variables. The final nonlinear program is given by introducing a sequence $\{y_1, \ldots, y_p\}$ of the so called *surrogates* of vector x_0. For each discipline the latter unknowns have substantially a role similar to that of vector t, i.e. they are used to decouple the upper level and the lower level in the following MDO reformulation.

$$\begin{array}{ll} \min\limits_{x_0, t} & f'(x_0, x_1, \ldots, x_p, t) \\ & \\ & \left\| t_i - C_i\left(y_i - x_0, s_i(y_i, x_i, t)\right) \right\|_* = 0, \qquad i \le p \\ & \\ & \min\limits_{y_i, x_i} \quad \dfrac{1}{2} \left[\|y_i - x_0\|^2 + \|s_i(y_i, x_i, t) - t_i\|^2 \right] \\ & \\ & \qquad g_i\left(y_i, x_i, s_i(y_i, x_i, t)\right) \ge 0, \qquad i \le p \end{array} \tag{3.9}$$

where the explicit availability of the subvector $s_i = s_i(y_i, x_i, t)$, by the implicit function theorem applied to the i-th block of constraints $A_i(y_i, x_i, t_1, \ldots, t_{i-1}, s_i, t_{i+1}, \ldots, t_p) = 0$, is a strong prerequisite. Observe that also in this case the MDO reformulation is quite articulate, however the bilevel structure may be fruitfully exploited by suitable techniques [10] of nonlinear programming. The choice of the norm '$*$' is substantially arbitrary. However, common choices are '$*$'$= 2$ (which yields CO_2) and '$*$'$= 1$ (which yields CO_1).

The first one is appealing because it gives a smooth feasible region of the upper level in (3.9). Unfortunately, in case the feasible region of the upper level is open, the KKT optimality conditions may fail, since the Jacobian matrix (of the upper level constraints) vanishes in any feasible point. Thus, the Lagrange multiplier rule [7] may not be satisfied, unless the solution is also an unconstrained stationary point of the upper level objective function.
The choice '$*$'$= 1$ in (3.9) may be also troublesome, inasmuch as the constraints of the upper level are not differentiable. This again implies that the Lagrange multiplier rule may fail. Numerical results give evidence of the shortcomings described for the MDO reformulations CO_1 and CO_2 [17].

4 A case study: sail boat keel design

The theory described in Sections 2-3, though not immediately applicable, must be considered for any MDO problem. We describe here an application on ship design, where unfortunately the discussion above has not a straightforward application. The design optimization of a fin of a sailing yacht is here described. This particular device is often used in race yacht to sustain the bulb, a faired object whose weight is able to give stability to the yacht itself. The bulb sometimes represents a large portion of the ship's displacement, around 80% in the America's Cup sailing yachts. A "pure" fluid dynamic approach will simply consider the hull, the fin and the bulb as rigid, connected bodies. Unlikely, as a consequence of the weight of the bulb and of its position, the shape of the fin is largely modified by the bending moments and stresses arising from the different sailing positions plus the dynamic pressure field, and the final performances of the yacht are undoubtedly influenced by the structural behavior of the fin.

In the following an extremely simplified case will be considered, which however contains all the fundamental elements of a typical MDO problem: an immersed, isolated fin moving at constant speed at a yaw and heel angle. The objective function is to maximize the efficiency of the fin (i.e. the ratio between the horizontal lift and the drag) The fin is assumed to have a constant horizontal section. Only the hydrodynamic actions are take into account and will be responsible for the bending of the fin which is assumed to be fixed at a certain level. This simplified problem will be solved by using a limited number of design variables and a single constraint is imposed, that is, a prescribed volume must be contained into the fin body.

The multidisciplinary equilibrium can be obtained in an iterative way. A non-linear BEM solver is adopted for the determination of the hydrodynamic loads: once these are computed, the deformed shape is obtained by the FEM solver and then passed back to the BEM solver, and so on until convergence. Convergence check is performed by monitoring the difference of the objective function value of the fin between two successive iterations j and $j+1$. When the difference is less than a cut-off threshold the equilibrium is assumed to be reached. The optimizer has then the task of finding a better shape, which at the beginning will not satisfy the multidisciplinary equilibrium. This basic approach is however well suited to test different reformulations and degrees of coupling among the disciplines, the focus of the present paper, which may be enforced by simply changing the convergence cut-off parameter.

In this preliminary application, a MDF reformulation and a suite of different DAO reformulations are solved and compared. In the following, $\mathbf{H}, \mathbf{S}$ indicate the hydrodynamic and the structural simulations, respectively, u is the vector of the disciplinary variables, $\mathbf{x}$ is the vector of the design variables identifying the shape of the fin and f is the objective function (here we drop the $'$ for simplicity). The basic algorithm at the step k is the following:

$$\mathbf{H} : x^k, \tilde{u}_S^0 \Longrightarrow u_H^j, f^j$$

$$\mathbf{S} : x^k, u_H^j \Longrightarrow \tilde{u}_S^j$$

$$\mathbf{H} : x^k, \tilde{u}_S^j \Longrightarrow u_H^{j+1}, f^{j+1}$$

$$\mathbf{S} : x^k, u_H^{j+1} \Longrightarrow \tilde{u}_S^{j+1}$$

Then convergence is checked: if $|\frac{f^{j+1}-f^j}{f^j}| \leq \epsilon$ then $\begin{cases} \tilde{u}_S^{j+1} & \rightarrow u_S^k \\ f^{j+1} & \rightarrow f^k \end{cases}$

Once the new multidisciplinary equilibrium has been found, the optimizer play the role of finding the new shape $\mathbf{x}^{k+1}$ and the cycle is ready to continue. In a MDF-type reformulation, the initial guess for the disciplinary variables is $\tilde{u}_S^0 = 0$, whereas in a DAO reformulation, the initial guess is $\tilde{u}_S^0 = u_S^k$. The latter choice ensures a faster convergence of the disciplinary variables once we are in the nearby of the optimal solution, because the initial values are nearly the convergent ones.

As a preliminary example, the standard (single-discipline) non-MDO reformulation can be compared with one MDO result. In figure 1, the objective function and the second design variable values provided by the single discipline optimization problem (hydrodynamic only) and the MDF reformulation of the coupled hydroelastic problem are reported. The same optimization algorithm has been adopted in both the numerical experiments. The difference in the objective function values is evident: the objective function value provided by the non-MDO reformulation is better than the one of the MDO problem. However, we should keep in mind that the non-MDO reformulation consider the body as rigid, *which in this case is rather far from the reality.* The hydroelastic approach is the only way to correctly consider (and try to solve) the design problem at hand. Differences in the results are indirect indicators of the inaccuracy in the solution of the problem if the hydroelastic coupling is not taken into account via an MDO approach.

The second example of the paper is introduced in the attempt of detecting some of the pros and cons coming out from the application of different reformulations and optimization algorithms. In the following, two different reformulations (MFD and DAO) and two different optimization algorithms will be applied to the same optimization problem. The coupling between the disciplines is modulated by means of the differences in the objective function value between two successive iterations: in this particular application,

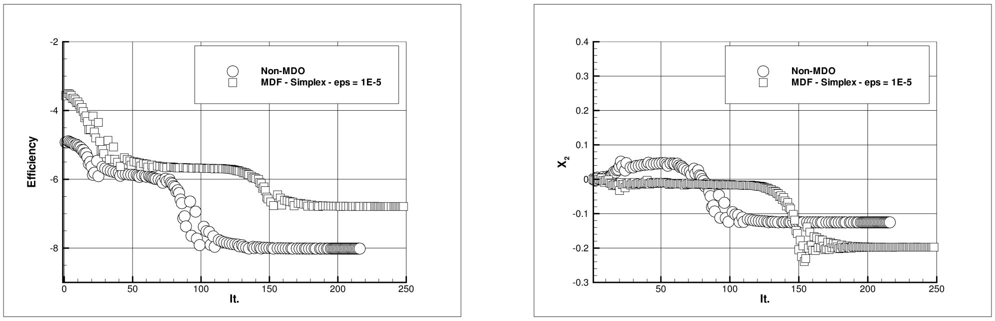

Figure 1: Comparison between the standard non-MDO reformulation and an MDO reformulation. The pure hydrodynamic non-MDO reformulation consider the body as rigid, whereas the more realistic hydroelastic approach take into account the deformation of the fin due to the hydrodynamics loads. **On left**: history of the objective function value during the iterations. **On right**: history of the second design variable value during the iterations. The other variables behave similarly.

the threshold value ϵ for the equilibrium constraints is set to 10^{-5} (see the above convergence condition).

Both the MDO reformulation have been tested with two different optimization algorithms: a standard Simplex Method [15] and a Derivative-Free Method [13]. In figure 2 the history of the design variables for the four different numerical experiments are reported.

A first comment about the influence of the MDO reformulation of the problem is nearly straightforward. In fact, no real differences are observed between the solutions provided by the Derivative-Free algorithm when changing the MDO problem reformulation, whereas only small differences are observed when using the Simplex algorithm. This behavior could be interpreted as a sign of the weak influence of the reformulation on the numerical solution of the MDO problem: the final solution does not change with the reformulation, which in turn means that the two reformulations can be considered as totally equivalent.

In table 1, the number of objective function evaluations and the number of inner iteration to achieve the convergence are also presented; in the last column, the average of the cost of the single objective function evaluation in terms of inner iterations is computed. It is evident how the DAO reformulation gives advantages in terms of the convergence of the disciplinary variables, decreasing the unit cost of the single multidisciplinary analysis.

On the other hand, different reformulations have a higher impact on the overall cost of the solution. In table 1, the number of objective function evaluations (i.e. the total number of CFD solution required which in more general terms are the number of disciplinary analyzes) and the number of inner iteration to achieve the convergence are presented; in the last column, the average cost of the single objective function evaluation in terms of inner iterations is computed. The DAO reformulation gives a slightly advantage in terms of convergence of the disciplinary variables, decreasing the unit cost of the single multidisciplinary analysis.

On the other hand, an increase in the number of iterations is observed. Moreover, the difference in the adopted algorithm results in different detected local minima. Both these features could be partly explained in a similar way. In fact, due to the adopted approach of considering the disciplines coupling convergence, we are introducing a numerical noise, i.e. a feasibility error when attempting at satisfying the equilibrium constraints. When the MFD reformulation is applied, the starting value for the disciplinary variables is set to zero and this is true for every single computation. On the contrary, for the DAO reformulation, the initial guess depends from the last computed configuration. Since the objective function $f' = f'(x, s(x), t(x))$ depends on the disciplinary variables $s(x)$, a different coupling parameter ϵ in the convergence check and/or a different initial guess for $s(x)$ could result in different final values of $s(x^*)$ and $f'(x^*, s(x^*), t(x^*))$. The coupling parameter ϵ was set quite low in this test and with an expected weak influence on the objective function value, but one must keep in mind that optimization algorithms are in general highly sensitive to this feature. As a consequence, the algorithm could perform poorly till convergence.

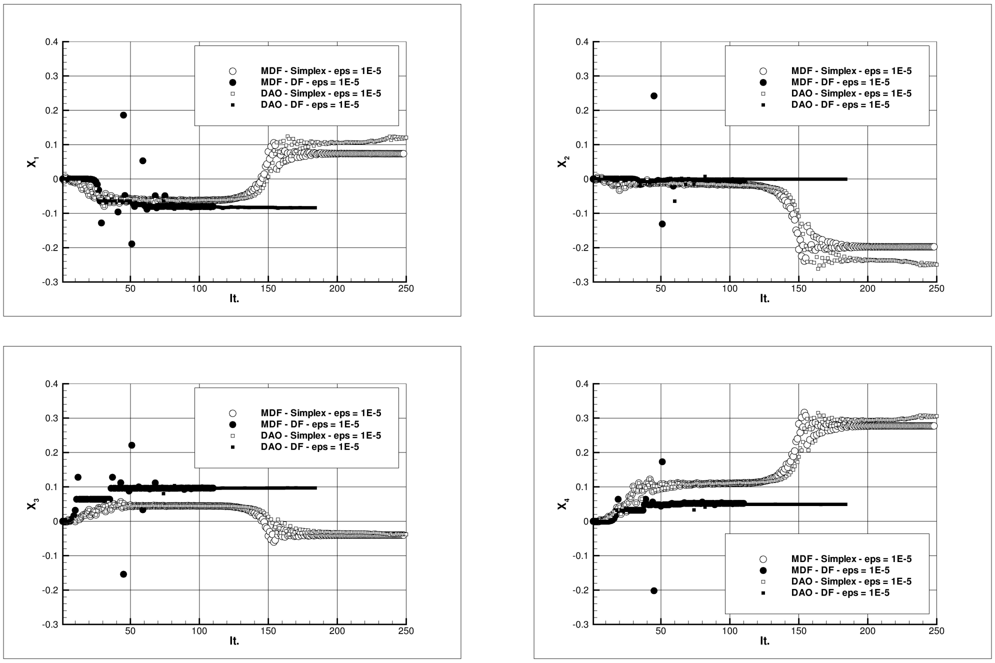

Figure 2: History of the design variables during the optimization process. Full symbols refer to the Derivative-Free algorithm, empty symbols refer to the Simplex algorithm. Circles indicate the MDF reformulation, squares the DAO. From top to bottom, left to right, variables 1, 2, 3 and 4.

A second simpler explanation for the different solutions detected is probably the fact that different algorithms have different selection strategy for the new iterate, and possibly this concurs in the selection of different local minima. This explains the difference in the optimal solution detected by the different algorithms.

It is finally fundamental here to stress that by changing the ϵ value in the convergence check, the advantage of the DAO w.r.t the MDF reformulation will probably increase greatly. In other terms, ϵ play the role of control over computational burden and by relaxing its value a substantial reduction in the total time is expected. The verification of this feature will represents the next step of this work.

5 Conclusions

This paper partially surveys some introductory aspects of renowned MDO formulations from the literature. Furthermore, specific emphasis is devoted to the relation between MDO problems and Nonlinear programming. A numerical experience is finally provided, in order to start the exploration of some of the main features of the problem reformulations. Numerical results demonstrate both the usefulness and the complexity of the problem.

Acknowledgments

This work has been supported by the *Ministero delle Infrastrutture e dei Trasporti* in the framework of the research plan “Programma di Ricerca sulla Sicurezza”, Decreto 17/04/2003 G.U. n. 123 del 29/05/2003.

Reformulation	Algorithm	Total calls	Iterations	Unit cost
MDF	Simplex	2597	248	10.5
MDF	Derivative Free	1303	110	11.9
DAO	Simplex	2832	290	9.8
DAO	Derivative Free	1710	184	9.3

Table 1: Cost of the optimization problem solution as a function of the adopted reformulation and algorithm. *Total calls* indicates the total number of disciplinary calls: total elapsed time is proportional to this value.*Iteration* represents the number of solutions required by the algorithm to achieve convergence. *Unit cost* is the ratio between the total calls and the iterations number and gives the average number of solver calls per each multidisciplinary analysis.

References

[1] Natalia M. Alaxandrov, M. Y. Hussaini (Eds.), 1997, "*Multidisciplinary Design Optimization - state of the art*", Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM Proceedings Series.

[2] Natalia M. Alaxandrov, Robert M. Lewis, 1999, "*Comparative properties of Collaborative Optimization and other approaches to MDO*", Proceedings of the First ASMO UK/ISSMO CONFERENCE on Engineering Design Optimization, July 8-9, 1999, MCB Press.

[3] Natalia M. Alexandrov, Robert M. Lewis, 2000, "*Algorithmic Perspectives on Problem Formulations in MDO*", AIAA 2000-4719.

[4] Natalia M. Alexandrov, Robert M. Lewis, 2000, "*Analytical and Computational Properties of Distributed Approaches to MDO*", NASA Langley Research Center, Hampton, Virginia, AIAA 2000-4718.

[5] Natalia M. Alexandrov, Robert M. Lewis, 2000, "*Analytical and Computational Aspects of Collaborative Optimization*", NASA Langley research Center, Hampton, Virginia, NASA/TM 2000-210104.

[6] Moktar S. Bazaraa, Hanif D. Sherali, C.M. Shetty, 1993 "*Nonlinear Programming: Theory and Algorithms - second edition*", John Wiley and Sons, Inc., Canada.

[7] Dimitri P. Bertsekas, 1995, "*Nonlinear Programming*", Athena Scientific, Belmont, Massachusetts.

[8] Robert D. Braun, 1996 "*An architecture for large-scale distributed design*". PhD thesis, Stanford University, May 1996. Department of Aeronautics and Astronautics.

[9] Evin J. Cramer, J.E. Dennis Jr., Paul D. Frank, Robert M. Lewis, Gregory R. Shubin, 1994 "*Problem Formulation for Multidisciplinary Optimization*", SIAM Journal on Optimization, Vol. 4., Issue 4, pp. 754-776.

[10] Stephan Dempe, 2002, "*Foundations of Bilevel Programming*", Kluwer Academic Publishers, Dordrecht, Boston, London, The Netherlands.

[11] Raphael T. Haftka, Zafer Gurdal, Manoharp P. Kamat, 1990 "*Elements of Structural Optimization*", Kluwer Academic Publishers, Dordrecht, The Netherland.

[12] Robert M. Lewis, 1997 "*Practical aspects of variable reduction formulations and reduced basis algorithms in Multidisciplinary Design Optimization*", in Natalia M. Alaxandrov, M. Y. Hussaini (Eds.), 1997, *Multidisciplinary Design Optimization - state of the art*, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM Proceedings Series.

[13] Stefano Lucidi, Marco Sciandrone, 2002 "*On the Global Convergence of Derivative Free Methods for Unconstrained Optimization*", SIAM Journal on Optimization, Vol. 13, issue 1, pp. 97-116.

[14] Olvi L. Mangasarian, 1994 "*Nonlinear Programming*", SIAM Classic in Applied Mathematics, Philadelphia, PA.

[15] J. L. Nelder, R. Mead, 1965 "*A Simplex Method for Function Minimization*", Computer Journal, Vol.7, pp. 308-313.

[16] Jaroslaw Sobieszczanski-Sobieski, Raphael T. Haftka, 1997 "*Multidisciplinary aerospace design optimization: Survey of recent developments*", Structural Optimization, Vol. 14, no. 1, pp. 1-23.

[17] Rajiv Thareja, Raphael T. Haftka, 1986, "*Numerical Difficulties associated with using equality constraints to achieve multi-level decomposition in structural optimization*", AIAA Paper 86-0854.