The state of the art in Collaborative Design

Mohamed Masmondi † Didier Auroux ‡

Abstract

The goal of this paper is to present and discuss main contributions in MultiDisciplinary Optimization (MDO) or collaborative design. The goal of MDO is to define a methodology for disciplines interaction with the aim of solving a global design problem. In MDO, most of authors consider two types of parameters, public (shared by disciplines) parameters, and private ones. In this paper, we first focus our interest on the definition, creation and use of public parameters. We give then an overview of most classical MDO approaches. Finally, we present another way to deal with MDO problems, based on game theory. Various examples are given in the following.

Key Words: Multidisciplinary optimization, collaborative design, game theory.

1 INTRODUCTION

The goal of this paper is to present and discuss most significant contributions in MultiDisciplinary Optimization (MDO).

The common point between the different publications is to consider two types of parameters, public parameters (shared by disciplines) and private ones. Unlike most contributions, we consider only public parameters in this paper. We assume that, for a given choice of public parameters, private parameters are fixed by each discipline at their optimal value.

The name multidisciplinary optimization is often taken literally. We prefer to call it collaborative design. Indeed, the optimization tool is only one part, which cannot be separated, of the total design process. Moreover, the goal is not to create an automatic design process, based on optimization algorithms, but to allow easy interaction between teams from different disciplines.

The definition of public parameters and their range of validity are big issues in MDO. Generally, the final design depends strongly on the set of valid parameters. For example, for the design of a wing, a structural analysis team usually hopes to increase the thickness and the hope of CFD (Computational Fluid Dynamics) team is to decrease it. Parameterization is a way to help teams to find a compromise. Only few papers point out the role of parameters definition.

Shape parameters are quite difficult to handle. Engineers used to create parameters on dead meshes. Unfortunately, each discipline has its own mesh: inside the structure for structural analysis, on its surface for acoustic, and outside it for CFD. For these reasons, we have to define discipline independent parameterized shapes.

Nowadays standard CAD (Computer Aided Design) codes provide parameterization facilities taking into account bounds and other types of constraints. Shape definition by CAD is an excellent way to facilitate interaction between design teams. Unfortunately, this is not the case for engineering teams. CAD codes do not offer mesh parameterization facilities. Using CAD environment, it is only possible to build a new mesh for a given choice of design parameters. This way of generating meshes is not suitable for optimal shape design. Generally, the numerical error due to the topology change of the mesh is too high in comparison with the information to be calculated (the difference between very close responses).

Multidisciplinary design is generally performed using a plateforme allowing the interaction between
analysis software and optimization toolboxes. The most known software are iSight, Dakota, Optimus (from LMS) and Boss Quatro [54, 55, 48, 2]. These environments also support “black box” applications. The language PEARL for file parsing is commonly used.

In section 2, we discuss the parameters definition problem. Most classical MDO approaches are presented in section 3. The application of game theory is a more recent contribution of J. Périaux et al. to MDO. It will be presented in section 4.

2 DEFINITION OF PARAMETERS

2.1 Public parameters

In a multidisciplinary context, the choice of the parameters can lead to many negotiations between teams. The definition of parameters is itself a design step, and the final result strongly depends on it.

In our opinion, the main issue of collaborative design is the definition of parameters step. Despite the technological development, this problem remains difficult. It cannot be defined in a Design Department (CAD Department), it is the affair of Engineering Department and generally the first simulations carried out on the product allow parameters definition in a relevant way.

2.2 Private parameters

If we consider a wing, the number of ribs, their positions and their sizes may only interest the engineer in charge of structural analysis. One calls them private parameters.

For a given choice of public parameters, the structure engineering department should carry out his computations with an optimal configuration of private parameters. This idea is considered in most papers.

2.3 Definition of public parameters

If one considers the significant case of shape optimization, the number of parameters is infinite. In practice, one can work with a large number of parameters: all the nodes of the mesh are design parameters. In this case, it is possible to find the optimal solution of the optimization problem, but not the optimal design. The design criteria like aesthetic and manufacturability are difficult to describe and they are not taken into account by the optimization problem. By reducing the number of design parameters we reduce the probability of unacceptable solutions.

However, an optimization process involving a large number of parameters can have at least two applications:

- the optimization process may compute an unexpected form, and one might realize afterward that the computed form is far from being interesting, despite some unsatisfied criteria,
- in the development step, the computation of the sensitivity with respect to all degrees of freedom of the problem (using adjoint methods) makes it possible to find the most sensitive regions and to create more appropriate parameters.

It is clear that in a multidisciplinary context, one can only work with a limited number of parameters.

2.3.1 Adjoint methods

Adjoint methods are an excellent way to define parameters. The criteria of the various disciplines can be combined in only one criterion in order to have
a common gradient. The distribution of the gradient on the surface of the initial design indicates more sensitive parts of the shape. One can also compute a gradient for each criterion. Each gradient, defined on the surface, can be considered as a basic perturbation function.

In [62] the authors determine the significant parameters by considering an hierarchical approach. A method based on control volumes is presented in the next section.

For the definition of adjoint, we refer to [11, 42, 27]. Automatic differentiation methods could be used for adjoint code generation [19, 18, 17, 42, 45, 46, 53, 60, 61, 67].

2.3.2 Engineering knowledge

The second approach is based on engineering knowledge about the dimensioning parameters of a shape. This is, for example, the case of the so-called aerodynamic functions like functions controlling the curvature of the leading edge of a wing.

These two approaches (adjoint methods and engineering knowledge) are complementary.

2.4 Creation of parameters

2.4.1 Overview

In the 90’s, commercial CAD software proposed facilities allowing the parameterization of shapes under some constraints. These constraints can directly appear as mathematical equations connecting some parameters to some others (imposed surface or volume). They can also be of geometrical nature: orthogonality, parallelism, ...

These new tools are still not used, despite of incentive policy of many companies for the use of CAD parameterization facilities:

- they are difficult to implement: it is not so easy to implement simultaneously an infinite number of shapes whereas it is already difficult to draw a fixed geometry;
- the technical communications often rely on CAD within a company. However, there is no standard format allowing parameterized geometry transfer between different software.

Other concerns appear in the particular context of collaborative design. The relevant parameters are not a priori known and CAD parameters may not be suitable.

Engineers generally don’t use the CAD facilities for the creation of parameters. They used to make some perturbations of the dead mesh.

2.4.2 Control volumes

We propose in this section an idea that meets the needs of engineers. In what follows, we will work on a dead mesh.

We present a method based on the volume transformations. It consists in associating a parameterized volume to the mesh. The mesh nodes undergo the same perturbations as the points of the volume (see paragraph Mesh perturbation).

The bivariate NURBS (Non Uniform Rational B-Splines) are a standard way for surface representation. One can imagine NURBS depending on three variables \((u, v, w)\) for the definition of volumes. The perturbation of the control points of the NURBS leads to a natural mesh perturbation.

One can use hexahedral finite elements as NURBS of degree 1 or 2. These elements are generally implemented in the computational part of most CAD software.

The major advantage of these methods is to create a regular mesh perturbation inside each volume.

Mesh perturbation The use of NURBS or finite elements both require an explicit function, which at each point \((u, v, w)\) of the reference cube \((0 \leq u \leq 1, 0 \leq v \leq 1, 0 \leq w \leq 1)\) associates a point \((x_i, y_i, z_i)\) of the control volume (see Fig. 1), where

\[
\begin{pmatrix}
x(u, v, w) \\
y(u, v, w) \\
z(u, v, w)
\end{pmatrix} = \sum P_i(u, v, w) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}
\]

(1)

and where \((x_i, y_i, z_i)\) are the control points of the control volume.
How to build a perturbation of the mesh? Let us denote by \((x_i^0, y_i^0, z_i^0)\) the coordinates of the control points before the perturbation, and \((x_i^1, y_i^1, z_i^1)\) the coordinates of these points after the perturbation. Let \((x^0, y^0, z^0)\) be a mesh node before perturbation. We first compute the triplet \((u^0, v^0, w^0)\) such that

\[
\begin{pmatrix}
x^0 \\
y^0 \\
z^0
\end{pmatrix} = \sum P_i(u^0, v^0, w^0) \begin{pmatrix}
x_i^0 \\
y_i^0 \\
z_i^0
\end{pmatrix}. \tag{2}
\]

The coordinates of this point after perturbation are given by

\[
\begin{pmatrix}
x^1 \\
y^1 \\
z^1
\end{pmatrix} = \sum P_i(u^0, v^0, w^0) \begin{pmatrix}
x_i^1 \\
y_i^1 \\
z_i^1
\end{pmatrix}. \tag{3}
\]

One should solve the system of the three equations (2) and find the three unknowns at each node of the initial mesh. This operation should be done only once for all perturbations.

**Perturbation of the external mesh** In the acoustic or electromagnetic field, one only has to mesh the structure’s surface. In this case, the process described in the previous paragraph makes it possible to perturb the surface nodes which are by definition inside the control volume.

Aerodynamic phenomena are often studied outside the structure. In this case, the greatest part of the mesh is outside the control volume.

We propose to use the control volume technique for the computation of the perturbations of the mesh surface, and to use traditional methods for the computation of the volume perturbations. There are several methods of mesh propagation.

### 2.4.3 Mesh propagation

In the case of an external mesh, if the parameter is defined by a surface perturbation, one needs to propagate the perturbation of the surface on the neighbouring mesh.

**Elasticity methods** The edge perturbation is considered as an imposed displacement. Inside the domain, one solves a linear elasticity problem. In general, stiffness is an increasing function of the element volume. We use the current mesh as a support for the computation.

**Integral methods** Let us denote by \(F(x)\) the image of a point \(x\). We assume that \(F(x)\) is well known for all points \(x\) of the edge. The displacement of an internal point is weighted by the inverse of its distance to the surface:

\[
F(x) = \left( \int_{\partial \Omega_v} \frac{F(x')}{||x-x'||} dx' \right) / \left( \int_{\partial \Omega_v} \frac{1}{||x-x'||} dx' \right). \tag{4}
\]

This integral may be computed by using the finite elements mesh.

### 2.5 Application to a wing

We now consider a wing whose parameters are the thickness, the twist, the arrow (see Fig. 2), the bend, and the inflection. (see Fig. 6 and 7).

In this case, we will naturally use the methods based on control volumes.

A thickness modification simply consists in a vertical perturbation of the control points of face \(A\) (see Fig. 1 and 3). The twist consists in a rigid rotation of face \(B\) (see Fig. 1 and 4). The arrow modification consists in a rigid translation of face \(B\) (see Fig. 1 and 5).

Naturally, one can imagine a finite element of degree 2 if we have to modify the bend of the aerofoil or the inflection of the wing. One can see in figure 6 a finite element of order two in the direction corresponding to the bend, and in figure 7 another finite element of order two in the direction corresponding to the inflection.

One can see on figure 8 the engine displacement along the wing. Blocks \(B1\) and \(B3\) follow the displacement induced by the control points of block \(B2\). If the leading edge and the tailing edge were parallel, one could only consider rigid displacements of this block. If we want to take into account the angle be-
3 SEVERAL COLLABORATIVE DESIGN METHODS

In order to simplify this presentation, we limit ourselves to two disciplines and the private parameters are hidden by each discipline. The state equations are given in an explicit form

\[
\begin{align*}
    a_1 &= A_1(x, a_2) \\
    a_2 &= A_2(x, a_1)
\end{align*}
\]

where \( x \) represents public design parameters. We assume that one forgets the state, and we only take care about the output. We denote by \( a_i \) the output of discipline \( i \). In the first equation, \( a_2 \) (resp. \( a_1 \)) corresponds to the contribution of discipline 2 (resp. 1) in the computation of \( a_1 \) (resp. \( a_2 \)).
We want to solve the following optimization problem
\[
\min_x f(x, a_1(x), a_2(x)), \\
g(x, a_1(x), a_2(x)) \leq 0
\] (7)

In this presentation we do not consider disciplinary internal constraints; they should be implicitly taken into account by the state equations.

### 3.1 The MDA (MultiDisciplinary Analysis) methods

In each iteration of the optimization algorithm, the state equations are assumed to be satisfied [3]. This is the most natural method in which each discipline provides some analysis services and possibly some gradient computations.

### 3.2 The AAO (All-At-Once) method

The AAO method is also the so-called “one shot”, SAND or SAD (Simultaneous Analysis and Design) method [10, 4, 25]. One solves simultaneously the state equations and the optimization problem. Contrary to the MDA method, the state equations are not supposed to be satisfied at each step of the optimization algorithm, but they are satisfied when the convergence is achieved. The state equations are indeed considered as equality constraints.

This method is really efficient, but it can be difficult to obtain its convergence when some of the state equations are strongly nonlinear. Even in a monodisciplinary context, the convergence is not always achieved.

### 3.3 The DAO (Disciplinary Analysis Optimization) method

In the DAO method [2, 4], the optimization is no more performed by the disciplines. However, each discipline may change the parameters in order to obtain an admissible point. We previously assumed that the private constraints are part of the state equations. The private parameters are generally used to satisfy these constraints, but there may not exist any vector $x$, representing the public variables, solution of the
problem. This is why one usually relaxes $x$. Each discipline chooses its own $z_i$ in order to satisfy its private constraints. When the convergence is achieved, the three $z_i$ are equal if we add three constraints in the optimization problem:

$$
\begin{align*}
\min_{x, b_1, b_2} f_{DAO}(x, b_1, b_2) \\
g(z_0, b_1, b_2) &\leq 0 \\
b_1 &= a_1(z_1, b_2), \\
b_2 &= a_2(z_1, b_1), \\
z_0 &= x \\
z_1 &= x \\
z_2 &= x
\end{align*}
$$  \hspace{1cm} (8)

### 3.4 The CO Method

In the CO method [10, 2, 6, 4, 5], all the disciplines contribute to the improvement of the results. We use the same notations as in the previous section presenting the DAO method.

Collaborative Optimization is a bilevel optimization approach: the system level and the disciplinary level.

At the system level we solve the following problem:

$$
\begin{align*}
\min_{x, b_1, b_2} f(x, b_1, b_2) \\
\left\{ \begin{array}{c}
\|x - x_1\|^2 + \|x - x_2\|^2 + \|a_1 - b_1\|^2 + \|a_2 - b_2\|^2 \\
g(z_0, z_1, z_2) \leq 0
\end{array} \right. \\
g(x, a_1(x), a_2(x)) \leq 0
\end{align*}
$$  \hspace{1cm} (9)

The main goal of this system level optimization problem is to impose to each discipline some objectives to be reached in a least square sense. The objective target is $b_i$ for discipline $i$ and the disciplinary solution $x_i$ should be close to $x$.

The disciplinary optimization problem for discipline 1 is then

$$
\begin{align*}
\min_{x_1} \frac{1}{2} \left[ \|x_1 - x\|^2 + \|b_1 - a_1(x_1, b_2)\|^2 \right]
\end{align*}
$$  \hspace{1cm} (10)

where $a_1$ is the solution of the state equation (5).

The optimization problem for discipline 2 is

$$
\begin{align*}
\min_{x_2} \frac{1}{2} \left[ \|x_2 - x\|^2 + \|b_2 - a_2(x_2, b_1)\|^2 \right]
\end{align*}
$$  \hspace{1cm} (11)

where $a_2$ is the solution of the state equation (6).

### 3.5 The CSSO (Collaborative Sub-Space Optimization) method

We will give a quick overview of the CSSO method, one can see [58] for a more detailed presentation. The design parameters are distributed between the different disciplines. Each discipline has to optimize a few parameters.

The CSSO is similar to the domain decomposition method and its application for solving state equations, except for one aspect: in domain decomposition, an unknown can easily be associated to one equation of the state. In multidisciplinary optimization, it seems hard to associate any public parameter to one particular discipline, because it has by definition to be shared between different disciplines.

We still use the same notations as in the previous sections. From now on, we will use an approximation of discipline 2 in the equation corresponding to discipline 1, and reciprocally, we use an approximation of discipline 1 in the equation corresponding to discipline 2.

Let us remind the initial problem

$$
\begin{align*}
a_1 &= A_1(x, a_2) \\
a_2 &= A_2(x, a_1)
\end{align*}
$$  \hspace{1cm} (12)

and we set $x = (x_1, x_2)$ where $x_i$ is the vector of the variables associated to discipline $i$. Then, we denote by $\tilde{a}_i$ the approximation of the analysis of $a_i$. We will further give more details about these approximations.

It is then natural to consider the following equations

$$
\begin{align*}
a_1 &= \tilde{A}_1(x, \tilde{a}_2) \\
\tilde{a}_2 &= \tilde{A}_2(x, a_1)
\end{align*}
$$  \hspace{1cm} (15)

\begin{align*}
\min_{x_1} f(x_1, a_1(x_1), \tilde{a}_2(x)) \\
g(x_1, a_1(x_1), \tilde{a}_2(x)) \leq 0
\end{align*}
$$  \hspace{1cm} (17)

(1 - \Delta)x_{1 \text{cur}} \leq x_1 \leq (1 + \Delta)x_{1 \text{cur}}.
$$  \hspace{1cm} (18)
The last constraint is introduced to set the validity domain of the approximation of $a_2$ around the current point $x_{i, cur}$. The approximated response $\tilde{a}_2$ and the coefficient $\Delta$ are provided by discipline 2.

In the same way, we can write the problem solved by discipline 2

$$\tilde{a}_1 = \tilde{A}_1(x, a_2)$$

$$a_2 = A_2(x, \tilde{a}_1)$$

$$\min_{x_2} f(x_1, x_2, \tilde{a}_1(x), a_2(x))$$

$$g(x, \tilde{a}_1(x), a_2(x)) \leq 0$$

$$(1 - \Delta)x_{2, cur} \leq x_2 \leq (1 + \Delta)x_{2, cur}$$

The process $i$ is supposed to provide to all other disciplines the state variable $x_{i, cur}$, the approximation $\tilde{a}_i$ of $a_i$ and its trust region.

Many results have been published about parameterization methods with the aim of increasing the autonomy of each discipline and enlarging the trust region [21, 22, 23, 24, 37, 33, 47, 52, 56, 57, 59].

In most domain decomposition methods [1], there is a natural way to associate unknowns to equations, and there are as many equations as unknowns. This association makes the resolution much easier. In collaborative design, it is not so natural to associate a sub-space to a discipline.

### 3.6 Surrogate models

Surrogate models provide a convenient way to reduce MDO complexity. An approximation has a local validity in an appropriate trust region:

- linearization is valid in a small region and for a large number of parameters if adjoint is considered,
- response surfaces are generally valid in a large region for only few parameters.

#### 3.6.1 First order approximation

The most simple way to build a local approximation is to differentiate (5) and (6), but its trust region is relatively small. We obtain

$$\tilde{a}_1 = \partial_x A_1(x_{cur}, a_2(x_{cur}))(x - x_{cur})$$

$$+ \partial_{a_2} A_1(x_{cur}, a_2(x_{cur}))(a_2(x) - a_2(x_{cur}))$$

and

$$\tilde{a}_2 = \partial_x A_2(x_{cur}, a_1(x_{cur}))(x - x_{cur})$$

$$+ \partial_{a_1} A_2(x_{cur}, a_1(x_{cur}))(a_1(x) - a_1(x_{cur}))$$

The adjoint method (reverse mode in automatic differentiation) could be considered for large scale problems: $x_i$ are of large dimension [19, 18, 17, 42, 45, 46, 53, 60, 61, 67].

#### 3.6.2 Approximation using parameterization methods

Parameterization methods are usually appropriate when the disciplines are weakly coupled. More precisely, we generally assume that one discipline is independent of the others:

$$a_1 = A_1(x)$$

$$a_2 = A_2(x, a_1).$$

In this example, discipline 1 does not require any information from discipline 2.

One can see this kind of approximations in the field of aeroelasticity. The output $a_1$ usually represents the structure response to mechanical excitation (a modal analysis is often necessary for determining the structure response). A good approximation $\tilde{a}_1$ of $a_1$ consists in using the same modal basis when the design parameters $x$ vary.

To build surrogate (or reduced) models, we proceed in the following way. First we solve (5) and (6) for several vectors $x$. Generally, we consider design of experiments methods to select appropriate vectors $x^k, k = 1, \cdots, n$ where $n$ is small.

The second step consists in building the surrogate model. There are two families of methods:

1. approximation of criteria and constraints using surface response methods [13, 64, 50] like polynomial approximation, neural networks, SVM (Support Vector Machine), RBF (Radial Basis Functions), ..
2. for each vector \( x^k \), we consider the corresponding solution \( u^k \) (the result of the full analysis). Then, for a given new vector \( x \), we look for the corresponding solution \( u \) as a linear combination of vectors \( u^k \). This problem is small and has only \( n \)-dimensional unknown vector. From the numerical point of view, it is necessary to build an orthonormal basis from vectors \( u^k \). For this reason, this method is called POD (Proper Orthogonal Decomposition) [21, 22, 23, 24, 37, 33, 47, 52, 56, 57, 59].

If we compare methods 1 and 2, the parameterization provided by POD takes into account the knowledge of the model.

With parameterization, the evaluation of the cost function is so cheap that it is possible to consider genetic algorithms [38, 49], and other global optimization algorithms [29, 30, 44].

4 GAME THEORY

Game theory has been used for multicriteria optimization [51]. Then, it has been enhanced by J. Périaux et al. [49, 38, 66] as an excellent way to deal with multidisciplinary optimization. Game theory seems to be more suitable for multidisciplinary design than most of classical methods. In particular, criteria like the maximum of the Von Mises stress, the drag and the lift, the maximum displacement, and the fuel consumption are criteria of totally different nature. In section 3, we simply weighted these criteria, but any weighting operation is arbitrary. The main advantage of game theory is to work in a multicriteria context, and each discipline is in charge of its own criterion.

The natural way to adapt game theory to multidisciplinary optimization is to consider each discipline as a player.

We consider two real cost functions defined on \( A \times B \):

\[
\begin{align*}
&f_A : A \times B \rightarrow \mathbb{R} \quad (x, y) \mapsto f_A(x, y) \quad (27) \\
&f_B : A \times B \rightarrow \mathbb{R} \quad (x, y) \mapsto f_B(x, y) \quad (28)
\end{align*}
\]

where \( A \) and \( B \) are two parts of \( \mathbb{R}^n \) and \( \mathbb{R}^m \) respectively. Player \( A \) minimizes the cost function \( f_A \) with respect to \( x \), while player \( B \) minimizes the cost function \( f_B \) with respect to \( y \).

4.1 The Pareto equilibrium

4.1.1 Definition

The point \((x^*, y^*)\) is Pareto optimal (or a Pareto equilibrium point) if there does not exist any \((x, y) \in A \times B\) such that:

\[
\begin{align*}
&f_A(x, y) \leq f_A(x^*, y^*) \\
&f_B(x, y) < f_B(x^*, y^*).
\end{align*}
\]

or

\[
\begin{align*}
&f_A(x, y) < f_A(x^*, y^*) \\
&f_B(x, y) \leq f_B(x^*, y^*).
\end{align*}
\]

This is equivalent to say that there is no way to improve \( f_A \) without a loss of performance of \( f_B \) and vice versa.

The set of all Pareto equilibrium points is called the Pareto front.

When the Pareto front is a convex set, one can find all its points while minimizing \( f_\lambda = \lambda f_A + (1 - \lambda) f_B \) for all \( 0 \leq \lambda < 1 \). For all \( \lambda \), there exists a Pareto equilibrium point.

The teams in charge of disciplines may use the set of the Pareto front to find a compromise.

4.1.2 Example

Let us consider a very simple example

\[
\begin{align*}
&f_A = (x - 1)^2 + (x - y)^2 \\
&f_B = (x - 3)^2 + (x - y)^2.
\end{align*}
\]

One can find the Pareto equilibrium points by considering a convex combination of the two criteria

\[
\begin{align*}
&f_\lambda = \lambda f_A + (1 - \lambda) f_B, \\
&\lambda = \lambda f_A + (1 - \lambda) f_B,
\end{align*}
\]

and minimizing \( f_\lambda \) for all \( 0 < \lambda < 1 \). In this case, we have

\[
\begin{align*}
&f_\lambda = \lambda(x - 1)^2 + (1 - \lambda)(y - 3)^2 + (x - y)^2
\end{align*}
\]
and then
\[
\begin{align*}
\partial_x f_\lambda &= 2 \left[ \lambda (x - 1) + (x - y) \right] = 0 \\
\partial_y f_\lambda &= 2 \left[ (1 - \lambda) (y - 3) - (x - y) \right] = 0.
\end{align*}
\] (32)
We finally obtain
\[
\begin{cases}
(1 + \lambda) x - y = \lambda \\
-x + (2 - \lambda) y = 3(1 - \lambda).
\end{cases}
\] (33)

4.2 The Nash equilibrium

4.2.1 Definition

The point \((x^*, y^*)\) is a Nash equilibrium point if
\[
\begin{cases}
f_A(x^*, y^*) = \inf_x f_A(x, y^*) \\
f_B(x^*, y^*) = \inf_y f_B(x^*, y).
\end{cases}
\] (34)

It is straightforward to see that this equilibrium point must satisfies the optimality system
\[
\begin{cases}
\nabla_x f_A(x^*, y^*) = 0 \\
\nabla_y f_B(x^*, y^*) = 0.
\end{cases}
\] (35)

4.2.2 Example

We consider again example (29)
\[
\begin{cases}
f_A = (x - 1)^2 + (x - y)^2 \\
f_B = (y - 3)^2 + (x - y)^2,
\end{cases}
\] (36)
and the corresponding Nash equilibrium is given by
\[
\begin{align*}
\partial_x f_A &= 0 \iff y = 2x - 1 \\
\partial_y f_B &= 0 \iff x = 2y - 3
\end{align*}
\] \Rightarrow \quad x = \frac{5}{3}, \quad y = \frac{7}{3}. 
\] (37)

One may remark that the solution of (37) is exactly the same as in (33) with \(\lambda = 1\) in the first equation and \(\lambda = 0\) in the second one.

The Nash equilibrium is an interesting way to avoid arbitrary weighting of cost functions. However, the compromise, between disciplines, is hidden by parameters definition. For example, the choice of the parameters subsets corresponding to discipline 1 and discipline 2 is a hidden compromise.

4.3 The Stackelberg equilibrium

4.3.1 Definition

We now assume that one of the two players is the leader of the game, for example player A. The multidisciplinary problem is formulated as follows:
\[
\min_x f_A(x, y_x)
\]
where \(y_x = \arg \min_y f_B(x, y)\). (38)

This is an elimination method for unknown variables. Then, we only have to minimize
\[
j(x) := f_A(x, y_x). \quad (39)
\]
If we assume that \(f_A\) is an aerodynamics cost function, for example the drag, and that \(f_B\) is the weight, then \(y\) represents the private parameters of the structural mechanics (number of spars, position, thickness, ...). Each evaluation of \(f_B\) requires the resolution of a structural mechanics optimization problem.

4.3.2 Example

We still consider the previous example
\[
\begin{cases}
f_A = (x - 1)^2 + (x - y)^2 \\
f_B = (y - 3)^2 + (x - y)^2.
\end{cases}
\] (40)
The Stackelberg equilibrium is exactly the Nash equilibrium:
\[
y_x = \frac{(x + 3)}{9},
\]
\[
j(x) = (x - 1)^2 + \left( \frac{x - 3}{2} \right)^2.
\]
The minimization of \(j(x)\) gives \(x = \frac{5}{3}\).
4.4 Application of the game theory to the multidisciplinary design

We simply have to consider equations (20) and cost function (7)

\[ a_1 = A_1(x, a_2) \]  \hspace{1cm} \text{(41)}
\[ a_2 = A_2(x, a_1) \]  \hspace{1cm} \text{(42)}
\[ \min f(x, a_1(x), a_2(x)) \]  \hspace{1cm} \text{(43)}
\[ g(x, a_1(x), a_2(x)) \leq 0 \]

where the weighted criterion \( f(x, a_1(x), a_2(x)) \) is replaced by \( f_1(x, a_1(x), a_2(x)) \) for discipline 1 and by \( f_2(x, a_1(x), a_2(x)) \) for discipline 2. For example, if discipline 1 is structural mechanics, \( f_1 \) may represent the weight of the structure. If discipline 2 is aerodynamics, then \( f_2 \) may be the lift (with a constraint on the drag).

Then, we consider the decomposition introduced in the frame of the CSSO method (see section 3.5). But, here the system level cost function is replaced by disciplinary cost functions. Discipline 1 solves the following problem

\[ a_1 = A_1(x, \tilde{a}_2) \]  \hspace{1cm} \text{(44)}
\[ \tilde{a}_2 = \tilde{A}_2(x, a_1) \]  \hspace{1cm} \text{(45)}
\[ \min f_1(x_1, x_2, a_1(x), \tilde{a}_2(x)) \]  \hspace{1cm} \text{(46)}
\[ g(x, a_1(x), \tilde{a}_2(x)) \leq 0 \]

\[ (1 - \Delta)x_{1, \text{cur}} \leq x_1 \leq (1 + \Delta)x_{1, \text{cur}} \]  \hspace{1cm} \text{(47)}

and the discipline 2 solves the following problem

\[ \tilde{a}_1 = \tilde{A}_1(x, a_2) \]  \hspace{1cm} \text{(48)}
\[ a_2 = A_2(x, \tilde{a}_1) \]  \hspace{1cm} \text{(49)}
\[ \min f_2(x_1, x_2, \tilde{a}_1(x), a_2(x)) \]  \hspace{1cm} \text{(50)}
\[ g(x, \tilde{a}_1(x), a_2(x)) \leq 0 \]

\[ (1 - \Delta)x_{2, \text{cur}} \leq x_2 \leq (1 + \Delta)x_{2, \text{cur}} \]  \hspace{1cm} \text{(51)}

One can add the constraints directly in the cost function in order to be exactly in the context of the game theory.

The Pareto front provides a fundamental answer to the multidisciplinary problems of optimization. Among all the solutions given by the Pareto front, one can choose the most relevant solution by taking into account implicit criteria, such as aesthetic, manufacturability, ... 

The Nash equilibrium makes it possible to avoid the main issue of compromises.

The Stackelberg equilibrium corresponds exactly to the optimization of internal (private) variables of the discipline.

Game theory is little used in the context of multidisciplinary optimization. However, at various levels, it can offer interesting answers:

- The Pareto equilibrium provides a complete range of relevant solutions. Among these solutions, one can choose the best design that satisfies subjective criteria
- The Nash equilibrium makes it possible to avoid compromises, even if a hidden compromise lies in the choice of the subspaces of discipline 1 and discipline 2.
- The Stackelberg equilibrium is particularly well adapted to optimization with respect to private variables.

5 CONCLUSION

Most contributions consider collaborative design from the optimization point of view. We pointed out the crucial role of parameters definition as a preliminary step for MDO. It is obvious that this step has a crucial impact on the final performances of each discipline. We also think that the choice of parameters cannot be done in an automatic way.

In our opinion, the collaborative design consists in providing an environment of aided design facilities
where engineers and designers still have the most important role.

The main steps of the collaborative design process are:

- joint definition of the public parameters, which are shared by all disciplines,
- creation of disciplinary reduced models allowing real-time response for a given value of the parameters,
- exploration of the design space (set of possible designs) to find the best design.

We also think that there are two main types of criteria or cost functions:

- objective criteria,
- subjective criteria.

The subjective criteria (such as aesthetic, manufacturability, ...) are difficult to handle by mathematical models, and should still be taken into account by engineers and designers. Only objective criteria are concerned with mathematical modeling.

References


[53] N. Rostaing and S. Dalmas, Automatic Differentiation Analysis and Transformation of Fortran


