The state of the art in Collaborative Design

Mohamed Masmoudi [†] Didier Auroux [‡] Yogesh Parte [*]

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 the 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 Multi-Disciplinary Optimization (MDO).

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 from 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 common point between the different publications is to consider two types of design parameters, public parameters (shared by disciplines) and private parameters (specific to the given discipline). 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 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 parametric definition.

Shape parameters are quite difficult to handle. Engineers usually 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 change in topology of the mesh is too high compared to the information to be calculated (the difference between very close responses).

Multidisciplinary design is generally performed using a platform allowing the interaction between analysis softwares and optimization toolboxes. The widely known softwares are iSight, Dakota, Optimus (from LMS) and Boss Quatro [64, 65, 58, 2]. These environments also support “black box” applications. The language PEARL for file parsing is commonly used.

Considering these points, the paper is organized as follows. In section 2, we discuss the parameter definition problem. Most classical MDO approaches are presented in section 3. The application of game theory is a recent contribution of J. Périaux et al. to MDO. It will be presented in section 4. In section 5 we give a brief survey of very recent developments and alternative approaches for MDO.

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 or choice of parameters is itself a design step, and the final result strongly depends on it.

In our opinion, the definition of parameters is a major issue in the collaborative design. Despite technological developments, this problem remains difficult. It cannot be defined in a design department (e.g. CAD department). It is the affair of engineering department; and generally the first simulations carried out on the product allow parameter 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 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 are 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 uninteresting, 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

†Laboratoire MIP, Université Paul Sabatier Toulouse 3, 31062 Toulouse Cedex 9, France - masmoudi@mip.ups-tlse.fr
‡Laboratoire MIP, Université Paul Sabatier Toulouse 3, 31062 Toulouse Cedex 9, France - auroux@mip.ups-tlse.fr
∗ Laboratoire MIP, Université Paul Sabatier Toulouse 3, 31062 Toulouse Cedex 9, France - parte@mip.ups-tlse.fr

SAROD-2005
Published in 2005 by Tata McGraw-Hill

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 [72] 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 [16, 50, 32]. Automatic differentiation methods could be used for adjoint code generation [24, 23, 22, 50, 55, 56, 63, 70, 71, 78].

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 aero-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 softwares proposed facilities allowing the parameterization of shapes under some constraints. These constraints can directly appear as mathematical equations connecting one set of parameters to others (imposed surface or volume). They can also be of geometrical nature: orthogonality, parallelism, etc.

These new tools are still not used, inspite of incentive policy of many companies for the use of CAD parameterization facilities. The reason being

- 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, y, z) of the control volume (see Fig. 1), where

$$\begin{pmatrix} x(u,v,w) \\ y(u,v,w) \\ z(u,v,w) \end{pmatrix} = \sum P_i(u,v,w) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad (1)$$

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

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

The coordinates of this point after perturbation are given by

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

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 analysis, only the structure's surface is meshed. 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 to 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 known for all points x of the edge. The displacement of an internal point is weighted by the inverse of its distance from the surface:

$$F(x) = \left(\int_{\partial\Omega_{r_0}} \frac{F(x')}{\|x - x'\|} dx' \right) / \left(\int_{\partial\Omega_{r_0}} \frac{1}{\|x - x'\|} dx' \right). \quad (4)$$

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 between these two edges, we have to force the control points of $B2$ to remain on the edge.

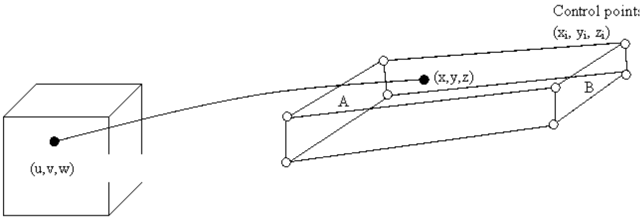

Figure 1: Reference cube (left) and initial hexahedral element with its 8 control points (right).

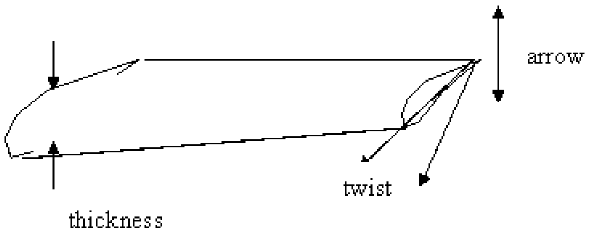

Figure 2: Wing parameters.

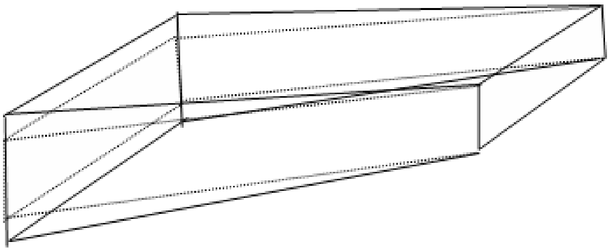

Figure 3: Modification of the wing thickness using a finite element of degree one in each direction.

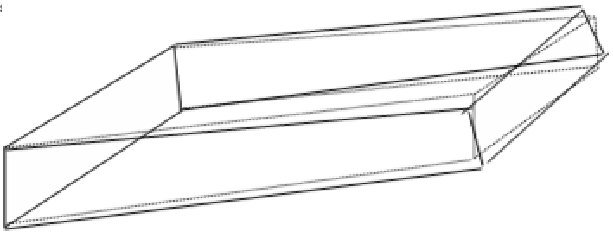

Figure 4: Modification of the aerofoil twist using a finite element of degree one in each direction.

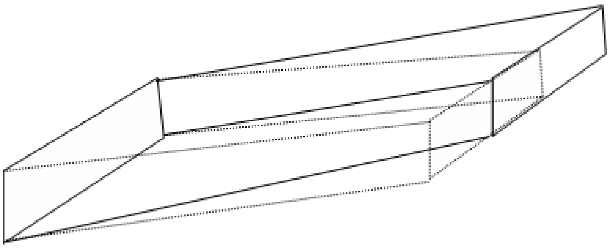

Figure 5: Modification of the arrow using a finite element of degree one in each direction.

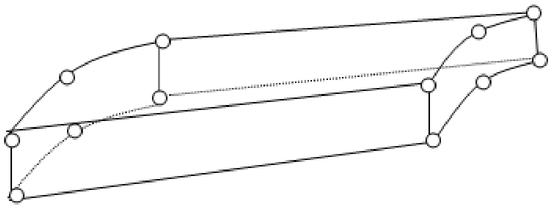

Figure 6: Modification of the bend using a finite element of second order in one direction and of degree one in the two other directions.

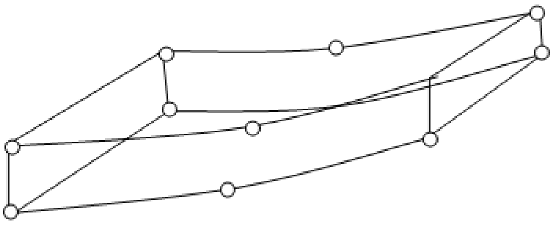

Figure 7: Modification of the inflection using a finite element of second order in one direction and of degree one in the two other directions.

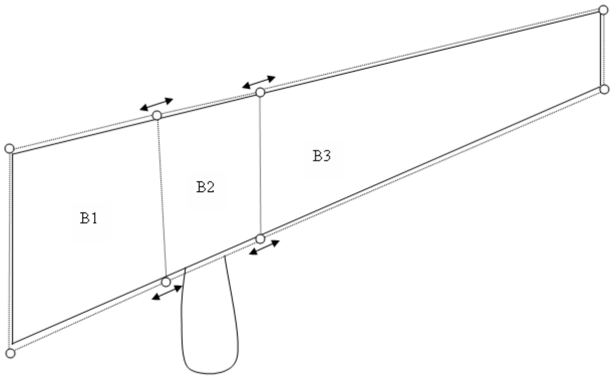

Figure 8: Engine displacement. The arrows correspond to the degrees of freedom of the different points, the other points are fixed.

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

$$a_1 = A_1(x, a_2) \tag{5}$$

$$a_2 = A_2(x, a_1) \tag{6}$$

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

$$\begin{array}{c} \min\limits_{x} \quad f(x, a_1(x), a_2(x)). \\ g(x, a_1(x), a_2(x)) \leq 0 \end{array} \tag{7}$$

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 (<u>M</u>ulti-<u>D</u>isciplinary <u>A</u>nalysis) 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 (<u>A</u>ll-<u>A</u>t-<u>O</u>nce) method

The AAO method is also the so-called “one shot”, SAND or SAD (<u>S</u>imultaneous <u>A</u>nalysis and <u>D</u>esign) method [15, 4, 30]. 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, i.e. 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:

$$\begin{array}{l} \min\limits_x \quad f_{DAO}(x, b_1, b_2) \\ g(z_0, b_1, b_2) \leq 0 \\ b_1 = a_1(z_1, b_2), \\ b_2 = a_2(z_1, b_1), \\ \quad z_0 = x \\ \quad z_1 = x \\ \quad z_2 = x \end{array} \tag{8}$$

3.4 The CO (Collaborative Optimization) method

In the CO method [15, 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:

$$\left\{ \begin{array}{l} \min\limits_x f(x, b_1, b_2) \\ \left[\|x - x_1\|^2 + \|x - x_2\|^2 + \|a_1 - b_1\|^2 + \|a_2 - b_2\|^2 \right] = 0 \\ g(z_0, b_1, b_2) \leq 0 \end{array} \right. \tag{9}$$

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

$$\min_{x_1} \frac{1}{2} \left[\|x_1 - x\|^2 + \|b_1 - a_1(x_1, b_2)\|^2 \right] \tag{10}$$

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

The optimization problem for discipline 2 is

$$\min_{x_2} \frac{1}{2} \left[\|x_2 - x\|^2 + \|b_2 - a_2(x_2, b_1)\|^2 \right] \tag{11}$$

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 [68] 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 multisciplinary 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 (denoted by $\tilde{A}_2$) in the equation corresponding to discipline 1, and reciprocally, we use an approximation of discipline 1 (denoted by $\tilde{A}_1$) in the equation corresponding to discipline 2.

Let us remind the initial problem

$$a_1 = A_1(x, a_2) \quad (12)$$
$$a_2 = A_2(x, a_1) \quad (13)$$

$$\min_x f(x, a_1(x), a_2(x))$$
$$g(x, a_1(x), a_2(x)) \leq 0 \quad (14)$$

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

$$a_1 = A_1(x, \tilde{a}_2) \quad (15)$$
$$\tilde{a}_2 = \tilde{A}_2(x, a_1) \quad (16)$$

$$\min_{x_1} f(x_1, x_2, a_1(x), \tilde{a}_2(x)) \quad (17)$$
$$g(x, a_1(x), \tilde{a}_2(x)) \leq 0$$

$$(1 - \Delta)x_{1\,cur} \leq x_1 \leq (1 + \Delta)x_{1\,cur}. \quad (18)$$

The last constraint is introduced to set the validity domain of the approximation of a_2 around the current point $x_{1\,cur}$. The approximate response $\tilde{a}_2$ and the coefficient Δ are provided by discipline 2.

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

$$\tilde{a}_1 = \tilde{A}_1(x, a_2) \quad (19)$$
$$a_2 = A_2(x, \tilde{a}_1) \quad (20)$$

$$\min_{x_2} f(x_1, x_2, \tilde{a}_1(x), a_2(x)) \quad (21)$$
$$g(x, \tilde{a}_1(x), a_2(x)) \leq 0$$

$$(1 - \Delta)x_{2\,cur} \leq x_2 \leq (1 + \Delta)x_{2\,cur} \quad (22)$$

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 [26, 27, 28, 29, 45, 41, 57, 62, 66, 67, 69].

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

$$\begin{aligned}\tilde{a}_1 &= \partial_x A_1(x_{cur}, a_2(x_{cur}))(x - x_{cur}) \\ &+ \partial_{a_2} A_1(x_{cur}, a_2(x_{cur}))(a_2(x) - a_2(x_{cur}))\end{aligned} \quad (23)$$

and

$$\begin{aligned}\tilde{a}_2 &= \partial_x A_2(x_{cur}, a_1(x_{cur}))(x - x_{cur}) \\ &+ \partial_{a_1} A_2(x_{cur}, a_1(x_{cur}))(a_1(x) - a_1(x_{cur}))\end{aligned} \quad (24)$$

The adjoint method (reverse mode in automatic differentiation) could be considered for large scale problems: x_i are of large dimension [24, 23, 22, 50, 55, 56, 63, 70, 71, 78].

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:

$$a_1 = A_1(x) \tag{25}$$
$$a_2 = A_2(x, a_1). \tag{26}$$

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 (DOE) 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 [18, 75, 60] 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 a 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) [26, 27, 28, 29, 45, 41, 57, 62, 66, 67, 69].

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 [46, 59], and other global optimization algorithms [34, 35, 54].

4 Game theory

Game theory has been used for multicriteria optimization [61]. Then, it has been enhanced by J. Périaux et al. [59, 46, 77] 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 $A \times B$:

$$\begin{array}{rrcl} f_A : & A \times B & \rightarrow & \mathbb{R} \\ & (x,y) & \mapsto & f_A(x,y) \end{array} \tag{27}$$

and

$$\begin{array}{rrcl} f_B : & A \times B & \rightarrow & \mathbb{R} \\ & (x,y) & \mapsto & f_B(x,y) \end{array} \tag{28}$$

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

$$\left\{ \begin{array}{l} f_A(x,y) \leq f_A(x^*,y^*) \\ f_B(x,y) < f_B(x^*,y^*), \end{array} \right.$$

or

$$\left\{ \begin{array}{l} f_A(x,y) < f_A(x^*,y^*) \\ f_B(x,y) \leq f_B(x^*,y^*). \end{array} \right.$$

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 $0 < \lambda < 1$. For all λ, 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

$$\left\{ \begin{array}{l} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (x-3)^2 + (x-y)^2. \end{array} \right. \tag{29}$$

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

$$f_\lambda = \lambda f_A + (1-\lambda) f_B, \tag{30}$$

and minimizing f_λ for all $0 < \lambda < 1$. In this case, we have

$$f_\lambda = \lambda(x-1)^2 + (1-\lambda)(y-3)^2 + (x-y)^2 \tag{31}$$

and then

$$\begin{array}{l} \partial_x f_\lambda = 2\left[\lambda(x-1) + (x-y)\right] = 0 \\ \partial_y f_\lambda = 2\left[(1-\lambda)(y-3) - (x-y)\right] = 0. \end{array} \tag{32}$$

We finally obtain

$$\left\{ \begin{array}{l} (1+\lambda)x - y = \lambda \\ -x + (2-\lambda)y = 3(1-\lambda). \end{array} \right. \tag{33}$$

4.2 The Nash equilibrium

4.2.1 Definition

The point (x^*, y^*) is a Nash equilibrium point if

$$\left\{ \begin{array}{l} f_A(x^*, y^*) = \inf\limits_x f_A(x, y^*) \\ f_B(x^*, y^*) = \inf\limits_y f_B(x^*, y). \end{array} \right. \tag{34}$$

It is straightforward to see that this equilibrium point must satisfiy the optimality system

$$\left\{ \begin{array}{l} \nabla_x f_A(x^*, y^*) = 0 \\ \nabla_y f_B(x^*, y^*) = 0 \end{array} \right. \tag{35}$$

4.2.2 Example

We consider again example (29)

$$\left\{ \begin{array}{l} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (y-3)^2 + (x-y)^2, \end{array} \right. \tag{36}$$

and the corresponding Nash equilibrium is given by

$$\begin{array}{l} \partial_x f_A = 0 \\ \partial_y f_B = 0 \end{array} \iff \begin{array}{l} y = 2x - 1 \\ x = 2y - 3 \end{array} \iff \begin{array}{l} x = \dfrac{5}{3} \\ y = \dfrac{7}{3} \end{array} \tag{37}$$

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:

$$\begin{array}{c} \min\limits_x f_A(x, y_x) \\ \text{where } y_x = \arg\min\limits_y f_B(x, y). \end{array} \tag{38}$$

This is an elimination method for unknown variables. Then, we only have to minimize

$$j(x) := f_A(x, y_x). \tag{39}$$

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, etc.). Each evaluation of f_B requires the resolution of a structural mechanics optimization problem.

4.3.2 Example

We still consider the previous example

$$\left\{\begin{array}{l} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (y-3)^2 + (x-y)^2. \end{array}\right. \tag{40}$$

The Stackelberg equilibrium is exactly the Nash equilibrium:

$$y_x = \frac{(x+3)}{2},$$

$$j(x) = (x-1)^2 + \left(\frac{x-3}{2}\right)^2.$$

The minimization of $j(x)$ gives

$$x = \frac{5}{3} \quad \text{and} \quad y_x = \frac{7}{3}$$

.

4.4 Application of the game theory to the multidisciplinary design

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

$$a_1 = A_1(x, a_2) \tag{41}$$

$$a_2 = A_2(x, a_1) \tag{42}$$

$$\min_x f(x, a_1(x), a_2(x)) \tag{43}$$

$$g(x, a_1(x), a_2(x)) \leq 0$$

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

$$a_1 = A_1(x, \tilde{a}_2) \tag{44}$$

$$\tilde{a}_2 = \tilde{A}_2(x, a_1) \tag{45}$$

$$\min_{x_1} f_1(x_1, x_2, a_1(x), \tilde{a}_2(x)) \tag{46}$$

$$g(x, a_1(x), \tilde{a}_2(x)) \leq 0$$

$$(1-\Delta)x_{1\,cur} \leq x_1 \leq (1+\Delta)x_{1\,cur}, \tag{47}$$

and discipline 2 solves the following problem

$$\tilde{a}_1 = \tilde{A}_1(x, a_2) \tag{48}$$

$$a_2 = A_2(x, \tilde{a}_1) \tag{49}$$

$$\min_{x_2} f_2(x_1, x_2, \tilde{a}_1(x), a_2(x)) \tag{50}$$

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

$$(1-\Delta)x_{2\,cur} \leq x_2 \leq (1+\Delta)x_{2\,cur}. \tag{51}$$

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 aesthetics, 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 Other approaches for collaborative design

In section 3 we discussed MDO framework in which a complex system is decomposed into subsystems based on the aspect of the involved analysis (such as discipline specific analysis and expertise). These frameworks are motivated primarily by aerospace industry in which the design task is based on disciplinary analysis, for example structure, fluid dynamics, control etc. In these frameworks all subsystems are at equal level and in principle there is no restriction on communications between subsystems. However, design cycles in industries like automotive sector are not discipline specific but are product object driven and maintain multi-hierarchical structure with restriction on communication paths amongst different subsystems and levels [52]. The Use of MDO approaches discussed in section 3 would require complete restructuring of the organization and hence, this sector follows a multi-level design and optimization framework using product design tools like analytical target cascading (ATC). We refer to [39, 52, 51] for further details on ATC and its applications. Comparative properties of ATC and CO and other approaches are discussed in [8]. A new framework based on nested ATC and CO methodology for multi-level systems is described in [9]

Another approach for collaborative design is based on the theory of multi-agent game and collectives [79, 38]. Unlike the hierarchical structure considered in earlier frameworks here the subsytems are considered as players with each player having strategies or moves for optimizing its objectives while interacting with other players who are simultaneously selecting there moves for optimizing their objective function. However, while doing so each player also tries to minimize the system objective function and receives incentives for moves that minimize the system objective function. Based on these incentives, each player decides his future moves. An equilibrium is reached when no player can improve its objective function further and thus, also results in optimized system objective. These techniques were first applied to variety of distributed optimization problems including network routing, computing resource allocation, and data collection by autonomous rovers [80, 81, 74] however, only recently were applied to aeronautical and other problems [13, 14].

6 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 aesthetics, 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 modelling.

References

[1] Y. Achdou, Y. A. Kuznetsov, O. Pironneau, Substructuring preconditioners for the Q_1 mortar element method, Numer.Math., 71, pp. 419-449, 1995.

[2] N. Alexandrov and S. Kodiyalam, Initial results of an MDO method evaluation study, *Technical Report AIAA-98-4884*, 1998.

[3] N. Alexandrov, Comparative Properties of Collaborative Optimization and Other Approaches to MDO, *Technical Report ICASE-99-209354*, 1999.

[4] N. M. Alexandrov and R. M. Lewis, Algorithmic perspectives on problem formulations in MDO, *Technical report, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary analysis and optimization*, AIAA 4719. 2000.

[5] N. M. Alexandrov and R. M. Lewis, Analytical and Computational Properties of Distributed Appraoches to MDO, *Technical Report AIAA-2000-4718*, 2000.

[6] N. Alexandrov, Analytical and Computational Aspects of Collaborative Optimization, *Technical Report NASA-2000-210104*, 2000.

[7] N. Alexandrov, Optimization of Engineering Systems Governed by Differential Equations: Editor's Introduction, *SIAM Activity Group on Optimization*, **11(2)**, 2000.

[8] J. T. Allison, Complex system optimization: A review of analytical target cascading, collaborative optimization, and other formulations, *Master's thesis, Department of Mechanical Engineering, University of Michigan*, 2004.

[9] J. Allison, M. Kokkolaras, M. Zawislak, P. Papalambros, On the use of anlaytical target cascading and collaborative optimization for complex system design, *6th World congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May-03 June, Brazil*,2005

[10] J. Allison, D. Walsh, M. Kokkolaras, P. Papalambros, M. Cartmell, Analytical target cascading in aircraft design, *AIAA paper*, 2005

[11] A. Aussem, Théorie et application des réseaux de neurones récurrents et dynamiques à la prédiction, à la modélisation et au contrôle adaptatif des processus dynamiques, *PhD Thesis, Université René Descartes Paris V*, 1995.

[12] W. Baur and V. Strassen, The Complexity of Partial Derivatives, *Theoretical Computer Science*, **22**, pp. 317–330 (1983).

[13] S. Bieniawski, D.H. Wolpert, I. Kroo, Discrete, Continuous, and Constrained Optimization Using Collectives,*AIAA Paper 2004-4580, Presented at the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY*, August 30-September 1, 2004.

[14] S. F. Bieniawaski, Distributed optimization and flight control using collectives, *PhD thesis, University of Stanford, U.S.A.*, October 2005.

[15] R. Braun, P. Gage, I. Kroo, and I. Sobieski, Implementation and performance issues in collaborative optimization, *Technical report, AIAA*, 1996.

[16] J. Cea, Conception optimale ou identification de forme, calcul rapide de la dérivée directionnelle de la fonction coût, *M. A. A. N.*, **20(3)**, pp. 371–402 (1986).

[17] M. Cormery, De l'optimisation aérodynamique vers l'optimisation multidisciplinaire dans un contexte industriel, *PhD Thesis, Université Paul Sabatier*, 2000.

[18] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines. (and other kernel-based learning methods), Cambridge University Press, 2000.

[19] M. Daumas, C. Mazenc, and J. C. Muller, User Transparent Interval Arithmetic, *Technical Report LIP-ENS Lyon*, 1994.

[20] J. C. Gilbert, G. Le Vey, and J. Masse, La différentiation automatique de fonctions représentées par des programmes, *Technical Report INRIA-1557*, 1991.

[21] J. C. Gilbert, Automatic Differentiation and Iterative Processes, *Optimization Methods and software*, **1**, pp. 13–21 (1992).

[22] A. Griewank, On Automatic Differentiation, *Mathematical Programming: Recent Developments and Applications, M. Iri and K. Tanabe Eds, Kluwer Academic Publishers, Dordrecht*, 1989.

[23] A. Griewank, D. Juedes, J. Srinivasan, and C. Tyner, ADOL-C, a Package for the Automatic Differentiation of Algorithms Written in C/C++, *ACM Trans. Math. Software*, (1990).

[24] Andreas Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, SIAM, 2000

[25] S. Grihon and M. Cormery, Multisciplinary Optimization Applied to a 500 Seat Aircraft, *Proceedings 4th ECCOMAS Conference, K. D. Papailiou et al. Eds, Athens, Greece, Wiley & Sons Ltd*, 1998.

[26] Ph. Guillaume and M. Masmoudi, Dérivées d'ordre supérieur en optimisation de domaines, *C. R. Acad. Sci. Paris, Série I*, **315**, pp. 859–862 (1992).

[27] Ph. Guillaume and M. Masmoudi, Calcul numérique des dérivées d'ordre supérieur en conception optimale de forme, *C. R. Acad. Sci. Paris, Série I*, pp. 1091-1096 (1993).

[28] Ph. Guillaume and M. Masmoudi, Computation of High Order Derivatives in Optimal Shape Design, *Numerische Mathematik*, **67**, pp. 231–250 (1994).

[29] Ph. Guillaume and M. Masmoudi, Solution to the Time-Harmonic Maxwell-s Equations in a Waveguide, Use of High Order Derivatives for Solving the Discrete Problem, *SIAM Journal on Numerical Analysis*, (1995).

[30] C. R. Gumbert, J. W. Hou, and P. A. Newman, Simultaneous aerodynamic and structural design optimization (sasdo) for a 2-d wing, *Technical report, AIAA-2001-2527*, 2001.

[31] Hillstrom, User Guide for Jakef, *Technical Report ANL/MCS TM-16, Argone National Laboratory, Argone, Il 60439*, 1985.

[32] Control Theory Applications for Optimum Design of Aerodynamic Shapes, Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, Dec. 1990, pp. 176-179.

[33] D. Kalman and R. Lindell, Automatic Differentiation in Astrodynamical Modelling, *Automatic Differentiation of Algorithms, A. Griewank, G. F. Corliss Eds, SIAM*, 1991.

[34] C. T. Kelley, Iterative methods for optimization, *Frontiers in Applied Mathematics, SIAM*, 1999.

[35] C. T. Kelley, Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition, *SIAM Journal of Optimization*, **10(1)**, pp. 43–55 (1999).

[36] I. Kroo and V. Manning, Collaborative optimisation status and directions, *Technical Report AIAA-2000-4721*, 2000.

[37] I. Kroo, Distributed mulitdisciplinary design and collaborative optimization,*VKI lecture series on Optimization Methods and Tools for Multicriteria/Multidisciplinary Design*, November 15-19, 2004

[38] I. Kroo, Collectives and complex design, *VK-I lecture series on Optimization Methods and Tools for Multicriteria/Multidisciplinary Design*, November 15-19, 2004.

[39] H. M. Kim, Target cascading in optimal system design,*PhD thesis, University of Michigan, U.S.A*, 2001.

[40] K. Kubota, A preprocessor for Fast Automatic Differentiation - Applications and Difficulties on Practical Problems, *in RIMS Kokyuroku 648, Fundamental Numerical Algorithms and their Software*, 1988.

[41] S. Lawrence, A. C. Tsoi, and A. D. Back, Function approximation with neural networks and local methods: bias, variance and smoothness, *Australian Conference on Neural Networks, Australian National Univ.*, pp. 16–21 (1996).

[42] S. Lawrence, C. L. Giles, and A. C. Tsoi, Lessons in neural network training: overfitting may be harder than expected, *14th Conference on Articial Intelligence, AIAA Press, Menlo Park, California*, pp. 540–545 (1997).

[43] J. D. Layne, Applying Automatic Differentiation and Self-Validation Numerical Methods in Satellite Simulations, *Automatic Differentiation of Algorithms, A. Griewank, G. F. Corliss eds, SIAM*, 1991.

[44] R. M. Lewis, V. Torczon, and M. W. Trosset, Why pattern search works, *Technical Report NASA/CR-1998-208966*, 1998.

[45] Y. Maday, A. Patera, G. Turinici, A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. J. Sci. Comput. 17, No.1-4, 437-446 (2002). MSC 2000: *65-99.

[46] R. Mäkinen, P. Neittaanmäki, J. Périaux, and J. Toivanen, A genetic algorithm for multiobjective design optimization in aerodynamics and electromagnetics, *Computation Fluid Dynamics '98, Proceedings of the ECCOMAS 98 Conference, K. D. Papailiou et al. (eds), Wiley*, **2**, pp. 418–422, Athens, Greece, 1998.

[47] J. Markish and K. Willcox, Multidisciplinary techniques for commercial aircraft system design, *Technical Report AIAA-2002-5612*, 2002.

[48] T. Masanao and N. Fujii, Second Order Necessary Conditions for Domain Optimization Problems in Elastic Structures, *Journal of Optimization Theory and Applications*, **72**, (1992).

[49] K. I. M. McKinnon, Convergence of the Nelder-Mead algorithm to a non stationary point, *SIAM Journal on Optimization*, **9(1)**, pp. 148–158 (1998).

[50] B. Mohammadi, O. Pironneau. Applied Shape Optimization for Fluids. Oxford University Press 2001.

[51] N. Michelena, H. M. Kim, P.Y. Pampalambros, A system partitioning and optimization approach to target cascading,*In Proceedings of the 12th International Conference on Engineering Design, Munich, Germany*, 1999.

[52] N. Michelena, P.Y. Pampalambros, Trends and challenges in system Design optimization, *Proceedings of the International Workshop on Multidisciplinary Design Optimization, Pretoria, S. Africa*, August 7-10, 2000.

[53] D. W. Monell and W. M. Piland, Aerospace Systems Design in NASA's Collaborative Engineering, *50th International Astronautical Congress, Amsterdam, The Netherlands*, **11(2)**, (1999).

[54] M. Mongeau, H. Karsenty, V. Rouzé, and J.-B. Hiriart-Urruty, Comparison of public-domain software for black box global optimization, *Optimization Methods and Software*, 1999.

[55] J. Morgenstern, How to compute fast a function and all its derivatives, a variation on the theorem of Baur-Strassen, *SIGACT News*, 1985.

[56] G. M. Ostrovskii, J. M. Volin, and W. W. Borisov, Uber die Berechnung von Ableitungen, *Wissenschaftliche Zeitschrift der Tecnischen Hochschule fur Chemie, Leuna Merseburg*, **13**, pp. 382–384 (1971).

[57] P.A. LeGresley and J.J. Alonso. Improving the Performance of Design Decomposition Methods with POD. 10th AIAA/ISSMO Multidisciplinary Analysis. AIAA 2004-4465, 2004.

[58] S. L. Padula, J. J. Korte, H. J. Dunn, and A. O. Salas, Multidisciplinary Optimization Branch Experience Using iSight Software, *Technical Report NASA-99-209714*, 1999.

[59] J. Périaux, Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications, *John Wiley & Son Ldt*, 1998.

[60] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, in Advances in Kernel Methods - Support Vector Learning, (Eds) B. Scholkopf, C. Burges, and A. J. Smola, MIT Press, Cambridge, Massachusetts, chapter 12, pp 185-208, 1999.

[61] A. M. Ramos, R. Glowinski, and J. Périaux, Nash equilibria for the multiobjective control of linear partial differential equations, *Journal of Optimization Theory and Applications*, **112(3)**, pp. 457–498 (2002).

[62] J. F. Rodriguez, J. E. Renaud, B. A. Wujek, and R. V. Trappeta, Trust Region Model Management In Multidisciplinary Design Optimization, *Journal of Computational and Applied Mathematics*, **124(1-2)**, pp. 139–154 (2000).

[63] N. Rostaing and S. Dalmas, Automatic Differentiation Analysis and Transformation of Fortran Program Using a Typed Functional Language, *International Conference on Computing Methods in Applied Sciences and Engineering, Paris*, 1992.

[64] A. O. Salas and J. L. Rogers, A Web-Based System for Monitoring and Controlling Multidisciplinary Design, *Technical Report NASA 97-206287*, 1997.

[65] A. O. Salas and J. C. Townsend, Framework Requirement for MDO Application Development, *Technical Report AIAA-98-4740*, 1998.

[66] D. Sandy, D. Balkin, and K. Lin, A neural approach to response surface methodology, *Communications in Statistics, Theory and Methods*, **29(9-10)**, pp. 2215-2227 (2000).

[67] D. B. Serafini, A framework for managing models in nonlinear optimization of computationally expensive functions, *PhD Thesis, Rice University*, 1998.

[68] J. Shankar, C. Ribbens, R. Haftka, and L. Watson, Computational study of nonhierarchical decomposition algorithm, *Computational Optimization and Applications*, 1993.

[69] I. P. Sobieski and I. M. Kroo, Collaborative Optimization Using Response Surface Estimation, *AIAA Journal*, **38(10)**, pp. 1931–1938 (2000).

[70] B. Speelpenning, Computing Fast Partial Derivatives of Functions Given by Algorithms, *PhD Thesis, University of Illinois*, 1980.

[71] V. Strassen, Algebraic Complexity Theory, *in J. van Leeuwen Ed., Handbook of Theoretical Computer Science, Elsevier, Amsterdam*, 1990.

[72] Z. L. Tang, J.A. Désidéri Towards Self-Adaptive Parameterization of Bezier Curves for Airfoil Aerodynamic Design. INRIA Technical Report RR-4572. October 2002. URL:http://www-sop.inria.fr/rapports/sophia/RR-4572.html

[73] V. Torczon and M. W. Trosset, Using approximations to accelerate engineering design optimization, *Technical Report NASA-ICASE*, 1998.

[74] K. Tumer, A. Agogino, D. H. Wolpert, Learning sequences of actions in collectives of autonomous

agents, *In Proceedings of the First International Joint Conference on Autonomous and Multi-Agent Systems, Bologna, Italy*, July 2002.

[75] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[76] J. L. Walsh, J. C. Townsend, A. O. Salas, J. A. Samareh, V. Mukhopadhyay, and J. F. Barthelemy, Multidisciplinary High-Fidelity Analysis and Optimization of Aerospace Vehicles, *Technical Report AIAA-2000-0418*, 2000.

[77] J. F. Wang, Optimisation distribuée multicritère par algorithmes génétiques et théorie des jeux - Application à la simulation numérique de problèmes d'hypersustentation en aérodynamique, *PhD thesis, Université Paris VI*, 1992.

[78] R. E. Wengert, A Simple Automatic Derivation Evaluation Program, *Comm. ACM*, **7**, pp. 463–464 (1964).

[79] D. H. Wolpert, K. Tumer, An Introduction to Collective Intelligence, *Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center*, 1999. URL:http://ic.arc.nasa.gov/ic/projects/coin pubs.html.

[80] D. H. Wolpert, K. Tumer, Collective Intelligence, Data Routing, and Braess' Paradox, *Journal of Artificial Intelligence Research*, 2002.

[81] D. H. Wolpert, K. Wheeler, K. Tumer, "Collective Intelligence for Control of Distributed Dynamical Systems, *Europhysics Letters, vol. 49 issue 6*, 708-714, 2000.

[82] J. A. Young, R. D. Anderson, and R. N. Yurkovich, A description of the F/A-18E/F design and design process, *Technical Report AIAA-98-4701*, 1998.