

Chapter 1

Collaborative Optimization

1.1. Introduction

In this chapter we 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 optimization. Indeed, the optimization tool is only one part which can not be separated from the total design process. The goal of collaborative optimization is not to create an automatic design process based on optimization algorithms but to allow easy interaction amongst teams from different disciplines.

Different publications related to MDO 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 our discussion. 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. The choice of these parameters is often dictated by the ability and the experience of the design engineers. The optimal design strongly depends on the set of selected public parameters. For example, for the design of an aircraft wing, the structural analysis team usually tries to increase the wing thickness and the aim of aerodynamicists is to decrease it. Parameterization is a way to help teams to find a compromise.

Chapter written by Yogesh PARTE and Didier AUROUX and Joël CLÉMENT and Mohamed MASMOUDI and Jean HERMETZ .

2 Collaborative Optimization

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

Now a days standard Computer Aided Design (CAD) codes provide parameterization facilities taking into account different constraints such as parameter bounds, volumes, density, etc. Shape definition by CAD is an excellent way to facilitate interaction between design teams. However, use of CAD for shape parameterization is not prevalent among engineering teams as 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, for example, the difference between very close responses. Therefore, for design involving shape optimization, mesh morphing which is a mean to achieve mesh parameterization, is recommended.

Multidisciplinary design is generally performed using a platform allowing interaction between analysis software and optimization toolboxes. The widely known software are iSIGHT, Dakota, Optimus (from LMS) and Boss Quatro [ALE 98, PAD 99, SAL 97, SAL 98]. These environments also support “black box” applications. The data and information exchange across different disciplines is permitted using XML or Python language based file parsing.

Considering these points, the discussions are organized as follows. In section 1.2, we discuss the parameter definition problem. Most classical MDO approaches are presented in section 1.3. Application of these MDO methods are illustrated in section 1.4. Application of game theory is a recent contribution of J. Périaux et al. to MDO. It will be presented in section 1.5. In section 1.6 we give a brief survey of very recent developments and alternative approaches for MDO.

1.2. Definition of parameters

1.2.1. *Public parameters*

In a multidisciplinary context, public parameters are design parameters shared by more than one discipline. The definition or choice of public parameters is itself a design step, and the final result strongly depends on it. The choice of the public parameters determine the interactions between different disciplines. We refer to the chapter “Multi parameter Shape Optimization” in this book for detailed discussion on parameterization.

In our opinion, the definition of parameters is a major issue in the collaborative design. Despite technological developments, this problem remains difficult. The choice of public parameters is to be made by discussion amongst all disciplines involved in design and analysis. Generally, first few simulations carried out on the product allow parameter definition in a relevant way.

1.2.2. *Private parameters*

Private parameters are design parameters for specific discipline and are not shared by other disciplines. For example, if we consider an aircraft wing, 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.

1.2.3. *Definition of public parameters*

If one considers the case of shape optimization, number of parameters are infinite. In practice, one can work with a large number of parameters; for example all the nodes of the mesh can be 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 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, 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.

1.2.3.1. *Engineering knowledge*

The knowledge base of engineers in terms of the experience they have gained in design and analysis process plays vital role, especially in the selecting shape parameters and their bounds. For example, in aerodynamics, the allowable range for the

4 Collaborative Optimization

leading edge sweep angle of the wing and the wing curvature for the operational characteristics of the aircraft are decided by the engineer based on the experience gained from wind tunnel testings and operational characteristics of the existing aircrafts.

1.2.3.2. *Adjoint methods*

Adjoint method is an excellent way to define parameters. Using adjoint methods, different criteria of different 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. For the definition of adjoint, we refer to [CEA 86, JAM 90, MOH 01]. Automatic differentiation methods could be used for adjoint code generation [GRI 89, GRI 90, GRI 00, MOH 01, MOR 85, OST 71, ROS 92, SPE 80, STR 90, WEN 64].

These two approaches (adjoint methods and engineering knowledge) for the definition of the parameters are complementary. In [TAN 02] the authors determine most significant design parameters by considering an hierarchical approach. This approach is also discussed in the chapter "Multiparameter Shape Optimization" of this book. We refer readers to this chapter for further details. The method based on control volumes is presented in the next section.

1.2.4. *Creation of parameters*

In the 90's, commercial CAD software introduced utilities for shape parameterization under constraints. These constraints can directly appear as mathematical equations connecting one set of parameters to others (for example, constrained surface or volume). They can also be of geometrical nature: orthogonality, parallelism, etc.

These new tools are still not used, in spite of incentive policy of many companies for the use of CAD parameterization facilities. The reasons 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 known a priori and CAD parameters may not be suitable.

1.2.4.1. Control volumes

In this section we propose an idea that could fulfill the needs of engineers. In what follows, we consider dead or fixed meshes. The method is based on the volume transformations. The central theme is to associate a parameterized volume to the mesh. The mesh nodes undergo the same perturbations as the points of the control volume.

The bivariate Non Uniform Rational B-Splines (NURBS) 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.

Major advantage of this method is to create a regular mesh perturbation inside each volume.

1.2.4.1.1. 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. 1.1), where

$$\begin{pmatrix} x(u, v, w) \\ y(u, v, w) \\ z(u, v, w) \end{pmatrix} = \sum P_i(u, v, w) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad (1.1)$$

and where (x_i, y_i, z_i) are the control points of the control volume. How to build a

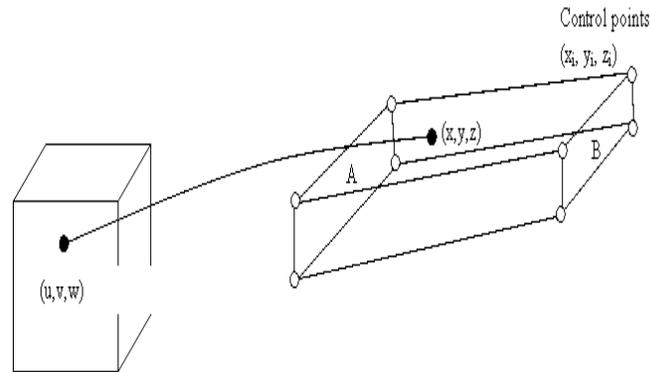


Figure 1.1. Reference cube (left) and initial hexahedral element with its 8 control points (right).

perturbation of the mesh? Let us denote by (x_i^0, y_i^0, z_i^0) the coordinates of the control points before 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

$$\begin{pmatrix} x^0 \\ y^0 \\ z^0 \end{pmatrix} = \sum P_i(u^0, v^0, w^0) \begin{pmatrix} x_i^0 \\ y_i^0 \\ z_i^0 \end{pmatrix}. \quad (1.2)$$

The coordinates of this point after perturbation are obtained using equation (1.3).

$$\begin{pmatrix} x^1 \\ y^1 \\ z^1 \end{pmatrix} = \sum P_i(u^0, v^0, w^0) \begin{pmatrix} x_i^1 \\ y_i^1 \\ z_i^1 \end{pmatrix} \quad (1.3)$$

One should solve the system of three equations (1.2) and find the three unknowns at each node of the initial mesh. This operation should be done only once for all perturbations.

1.2.4.1.2. Perturbation of the external mesh

In acoustic or electromagnetic field analysis, only the surface of the structure is meshed. For these applications, it is possible to perturb the surface nodes using the process described in the previous paragraph because by construction these nodes are inside the control volume. In aerodynamic analysis often the domain of analysis is outside the structure. In this case, the large part of the mesh is outside the control volume. We propose to use the control volume technique to compute surface mesh perturbations, and to use traditional methods of mesh propagation for the computation of the external mesh perturbations. Some of these mesh perturbation methods are now discussed.

1.2.4.2. *Mesh propagation*

In the case of an external mesh, from a surface perturbation, one needs to propagate the modification to the neighboring mesh.

1.2.4.2.1. Elasticity methods

The surface 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. Initial mesh is used as a support for the computation [BAT 91].

1.2.4.2.2. Integral methods

Let us denote by $F(x)$ an image of a point x after perturbation. We assume that $F(x)$ is known for all points x of the surface. The displacement of an internal point is weighted by the inverse of its distance from the surface:

$$F(x) = \left(\int_{\partial\Omega_{r_0}} \frac{F(x')}{\|x - x'\|} dx' \right) / \left(\int_{\partial\Omega_{r_0}} \frac{1}{\|x - x'\|} dx' \right). \quad (1.4)$$

This integral may be computed by using the perturbed mesh.

1.2.5. Application to a wing

We now consider a wing whose design variables are thickness, twist, sweep, airfoil and dihedral angle (see Fig. 1.2). We show the natural parameterization of these variables using the method of control volumes discussed earlier.

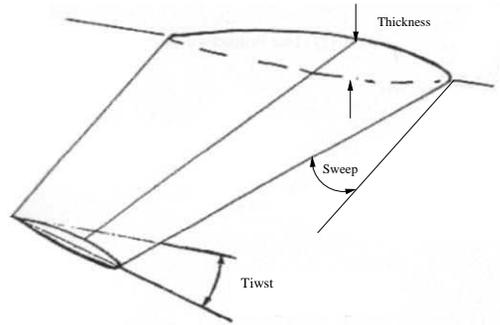


Figure 1.2. Wing parameters.

A thickness modification is achieved simply by a vertical perturbation of the control points of face A (see Fig. 1.1 and 1.3). The twist parameterization is achieved by a rigid rotation of face B (see Fig. 1.1 and 1.4). The modification in wing sweep angle consists in a rigid translation of face B (see Fig. 1.1 and 1.5).

One can imagine a finite element of degree two if we have to modify the curvature of the airfoil or the dihedral of the wing. One can see in figure 1.6 a finite element of degree two in the plane containing the airfoil, and in figure 1.7 another finite element of degree two in the direction corresponding to the dihedral. One can see on figure

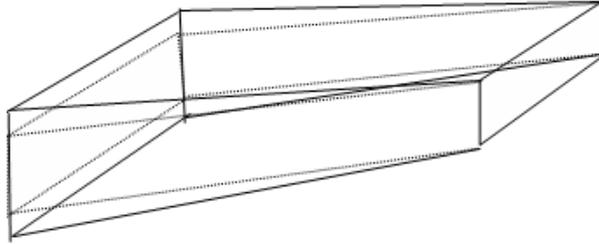


Figure 1.3. *Modification of the wing thickness using a finite element of degree one in each direction.*

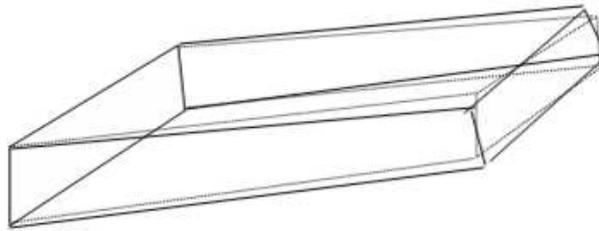


Figure 1.4. *Modification of the wing twist using a finite element of degree one in each direction.*

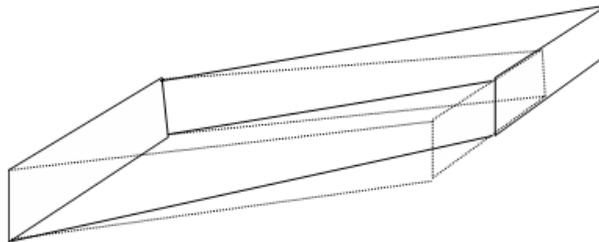


Figure 1.5. *Modification of the wing sweep using a finite element of degree one in each direction.*

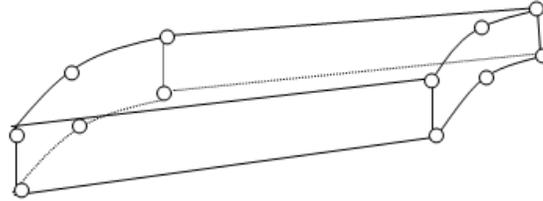


Figure 1.6. Modification of the wing profile using a finite element of degree two in one direction and of degree one in the two other directions.

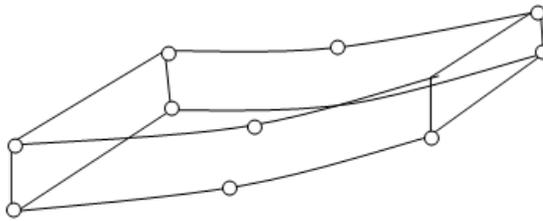


Figure 1.7. Modification of the dihedral using a finite element of degree two in one direction and of degree one in two other directions.

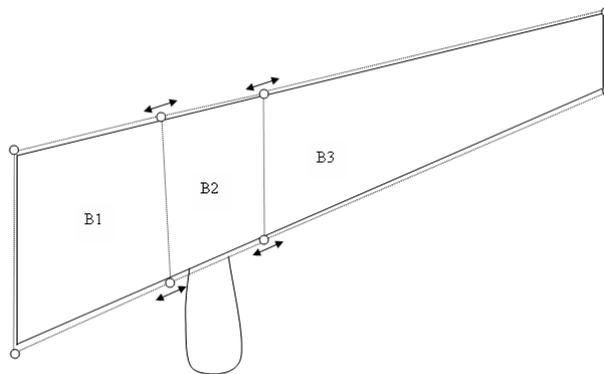


Figure 1.8. Engine displacement. The arrows correspond to the degrees of freedom of different points, other points are fixed.

1.8, 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 trailing 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.

1.3. Different frameworks for collaborative optimization

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

$$a_1 = A_1(x, a_2) \quad (1.5)$$

$$a_2 = A_2(x, a_1) \quad (1.6)$$

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 equation 1.5, a_2 corresponds to the contribution of discipline 2 in the computation of a_1 and in equation 1.6, a_1 represents the output of discipline 1 in the computation of a_2 . Thus a_1 and a_2 also represent the coupling between two disciplines.

We want to solve the following optimization problem

$$\begin{aligned} \min_x \quad & f(x, a_1(x), a_2(x)) \\ \text{Subjected to:} \quad & \\ & g(x, a_1(x), a_2(x)) \leq 0 \end{aligned} \quad (1.7)$$

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

1.3.1. *Multi-Disciplinary Feasible (MDF) method*

In this formulation, all design variables x are provided to the coupled system of disciplinary analysis. For given design variables x , a complete multidisciplinary analysis (MDA) is performed via a fixed-point iteration to obtain output variables a_1, a_2 . These outputs are then used in evaluating the objective $f(x)$ and the constraints $g(x, a_1, a_2)$. The optimization problem solved at system level is:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{Subjected to:} \quad & \\ & g(x) \leq 0 \end{aligned} \quad (1.8)$$

If a gradient-based method is used to solve the above problem, then a complete MDA is necessary not just at each iteration, but at every point where the derivatives are to be evaluated. Thus, attaining multidisciplinary feasibility can be prohibitively expensive in realistic application [KOD 98].

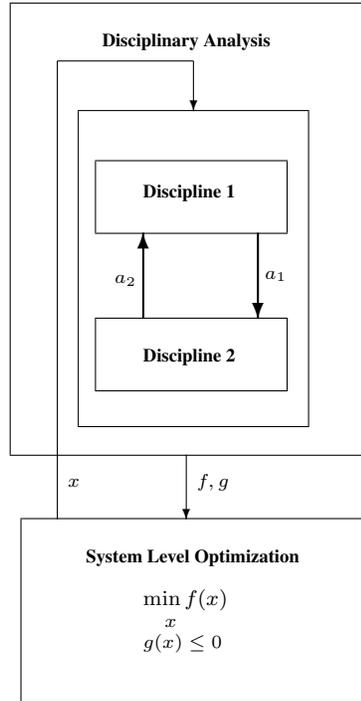


Figure 1.9. MDF framework for MDO

1.3.2. Individual Discipline Feasible (IDF) method

As the name suggest, the IDF formulation provides a way to avoid a complete MDA at optimization. IDF maintains individual discipline feasibility, while allowing the optimizer to drive individual disciplines to multidisciplinary feasibility and optimality by controlling interaction variables. The specific analysis variables that represent communication, or coupling, between disciplines are treated as optimization variables. The IDF formulation is:

$$\begin{aligned} \min_{x, b_1, b_2} \quad & f(x, b_1, b_2). \\ & g(x, b_1, b_2) \leq 0 \\ & b_1 = a_1(x, b_2) \\ & b_2 = a_2(x, b_1) \end{aligned} \tag{1.9}$$

$a_1(x, b_2)$ and $a_2(x, b_1)$ are the respective solutions of the state equations 1.5 and 1.6.

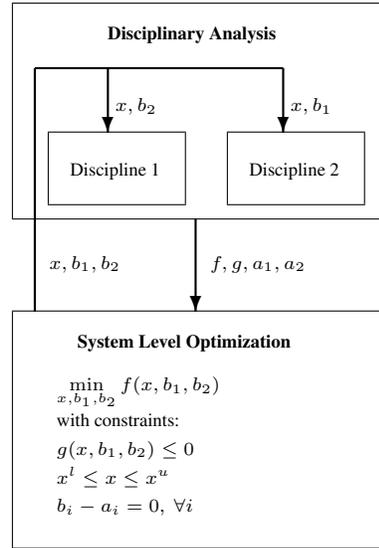


Figure 1.10. IDF framework for MDO

1.3.3. All-At-Once (AAO) method

The AAO method is also the so-called “one shot”, SAND or SAD (Simultaneous Analysis and Design) method [ALE 00b, BRA 96, GUM 01]. Simultaneously one solves state equations and the optimization problem. Contrary to the MDF 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. State equations are indeed considered as equality constraints. The optimization problem defined by this method is:

$$\begin{aligned}
 & \min_{x, b_1, b_2} f(x, b_1, b_2). \\
 & g(x, b_1, b_2) \leq 0 \\
 & b_1 = t_1 \\
 & b_2 = t_2 \\
 & t_1 = A_1(x, b_2) \\
 & t_2 = A_2(x, b_1)
 \end{aligned} \tag{1.10}$$

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 mono-disciplinary context, the convergence is not always achieved.

1.3.4. Bi-Level Integrated System Synthesis (BLISS)

This method is described in [SOB 98]. At every stage, equations of state given by equation (1.5) and (1.6) are solved and corresponding gradients are computed using an associate method. The gradient information is used to construct an approximation of the global cost function and the constraints. The approximation of the cost function includes a term associated with Lagrange multipliers, λ , coming from the local optimizations. The region of confidence to ascertain reliability of these approximations is also computed. The optimization problem in terms of point x_{cur} is as follows:

$$\begin{aligned}
 \partial_x f_0 &= \partial_x f(x_{cur}, a_1(x_{cur}), a_2(x_{cur})) \\
 \partial_x g_0 &= \partial_x g(x_{cur}, a_1(x_{cur}), a_2(x_{cur})) \\
 g_0 &= g(x_{cur}, a_1(x_{cur}), a_2(x_{cur})) \\
 \min_x & (\partial_x f_0 + \lambda \cdot \partial_x g_0)(x - x_{cur}) \\
 g_0 + \partial_x g_0(x - x_{cur}) &\leq 0 \\
 (1 - \Delta)x_{cur} &\leq x \leq (1 + \Delta)x_{cur}
 \end{aligned} \tag{1.11}$$

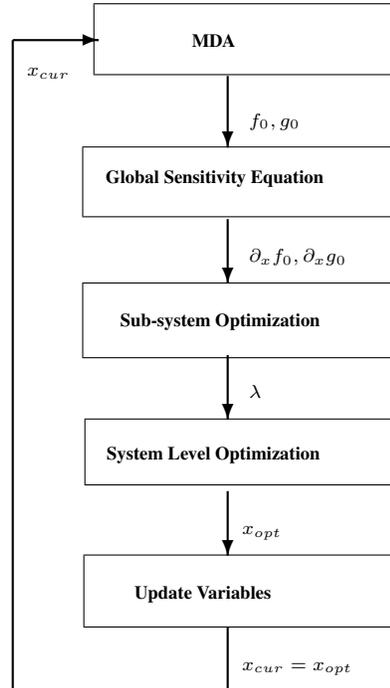


Figure 1.11. Diagram of the BLISS formulation

where, Δ denotes the region of confidence.

Depending on the method used to generate approximations of the cost function and constraints, different variant of BLISS are cited in literature. The formulation BLISS-RSM [ALT 02] uses polynomial approximations as disciplinary meta-models. The formulation BLISS-2000 [AGT 00] generalizes the BLISS by using disciplinary meta-models.

1.3.5. *Disciplinary Analysis Optimization (DAO) method*

In DAO method [ALE 98, ALE 00b], the optimization is no more performed by 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. Private parameters are generally used to satisfy these constraints, but there may not exist any vector x , representing the public variables, that is solution of the problem. This is why one usually relaxes x . Each discipline chooses its own set of public parameters z_i in order to satisfy its private constraints. When the convergence is achieved, all z_i are equal. Convergence of all z_i is ensured by added equality constraint of the form $z_i = x$ in the optimization problem. The optimization problem solved by DAO method is:

$$\begin{aligned} \min_{x, b_1, b_2} \quad & f_{DAO}(x, b_1, b_2). \\ & g(z_0, b_1, b_2) \leq 0 \\ & b_1 = a_1(z_1, b_2) \\ & b_2 = a_2(z_2, b_1) \\ & z_0 = x \\ & z_1 = x \\ & z_2 = x \end{aligned} \tag{1.12}$$

1.3.6. *Collaborative Optimization (CO) method*

In the CO method [ALE 98, ALE 00b, ALE 00c, ALE 00a, BRA 96], all disciplines contribute to the improvement of the results. We use same notations used to present DAO method. Collaborative Optimization is a bi-level optimization approach. The optimization problem is solved at the system level and the disciplinary level.

At system level we solve the following problem:

$$\left\{ \begin{array}{l} \min_x f(x, b_1, b_2) \\ \|x - z_1\|^2 + \|x - z_2\|^2 + \\ \|a_1 - b_1\|^2 + \|a_2 - b_2\|^2 = 0 \\ g(z_0, b_1, b_2) \leq 0 \end{array} \right. \tag{1.13}$$

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. For discipline i , output target is b_i and the disciplinary solution z_i should be close to x .

The disciplinary optimization problem for discipline 1 is then

$$\min_{z_1} \frac{1}{2} [\|z_1 - x\|^2 + \|b_1 - a_1(z_1, b_2)\|^2] \quad (1.14)$$

where a_1 is the solution of the state equation (1.5).

The optimization problem for discipline 2 is

$$\min_{z_2} \frac{1}{2} [\|z_2 - x\|^2 + \|b_2 - a_2(z_2, b_1)\|^2] \quad (1.15)$$

where a_2 is the solution of the state equation (1.6).

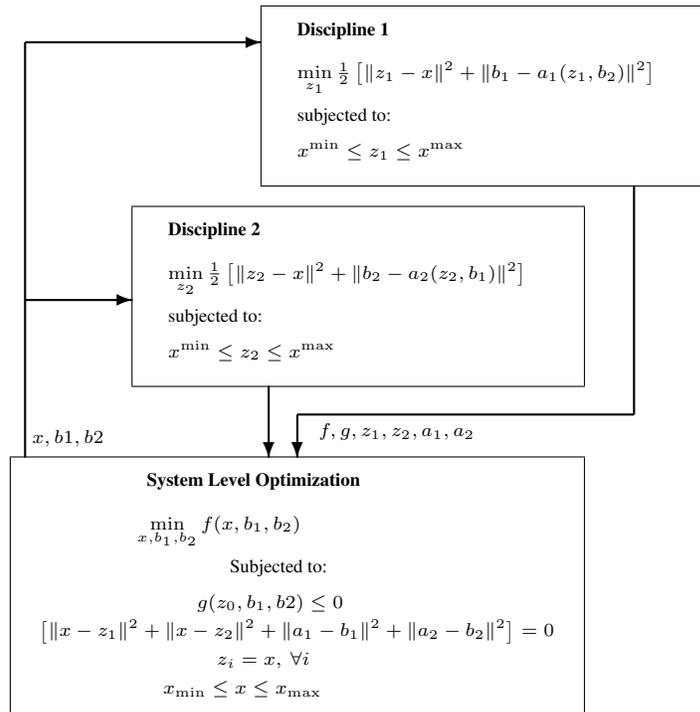


Figure 1.12. Diagram of the CO formulation

1.3.7. *Collaborative Sub-Space Optimization (CSSO) method*

We will give a quick overview of the CSSO method, one can see [SHA 93] for a more detailed presentation. The design parameters are distributed between the different disciplines. Each discipline has to optimize certain 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 by definition public parameters are to be shared between different disciplines.

We will use an approximation or meta-model 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 recall the initial problem

$$a_1 = A_1(x, a_2), \quad (1.16)$$

$$a_2 = A_2(x, a_1), \quad (1.17)$$

$$\min_x f(x, a_1(x), a_2(x)).$$

$$g(x, a_1(x), a_2(x)) \leq 0 \quad (1.18)$$

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 output of a_i . Further details about these approximations are referred to section 1.3.9.

The optimization problem for discipline 1 is:

$$a_1 = A_1(x, \tilde{a}_2), \quad (1.19)$$

$$\tilde{a}_2 = \tilde{A}_2(x, a_1), \quad (1.20)$$

$$\min_{x_1} f(x_1, x_2, a_1(x), \tilde{a}_2(x)) \quad (1.21)$$

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

$$(1 - \Delta_2)x_{1\ cur} \leq x_1 \leq (1 + \Delta_2)x_{1\ cur} \quad (1.22)$$

The constraint given by equation 1.22 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_2 are provided by discipline 2.

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

$$\tilde{a}_1 = \tilde{A}_1(x, a_2), \quad (1.23)$$

$$a_2 = A_2(x, \tilde{a}_1), \quad (1.24)$$

$$\min_{x_2} f(x_1, x_2, \tilde{a}_1(x), a_2(x)). \quad (1.25)$$

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

$$(1 - \Delta_1)x_{2\ cur} \leq x_2 \leq (1 + \Delta_1)x_{2\ cur} \quad (1.26)$$

where constraint given by equation 1.26 is introduced to set the validity domain of the approximation of a_1 around the current point $x_{2\ cur}$. The approximate response \tilde{a}_1 and the coefficient Δ_1 are provided by discipline 1.

Many results have been published about parameterization methods with the aim of increasing the autonomy of each discipline and enlarging the trust region [GUI 92, GUI 93, GUI 94, GUI 95, MAD 00, LAW 96, LEG 04, ROD 00, SAN 00, SER 98, SOB 00].

In most domain decomposition methods [ACH 95], there is a natural way to associate unknowns to equations, and there are as many equations as unknowns. This association makes the finding the solution much easier. In collaborative design, it is not so natural to associate a sub-space to a discipline. The chapter *Concurrent Optimization* proposes strategy for sharing different design parameters.

1.3.8. *Disciplinary Interaction Variable Elimination (DIVE) method*

This MDO formulation is described in [MAS 06]. It is primarily inspired by the BLISS2000 method [AGT 00]. The core idea in this formulation is to reduce the complexity of the optimization problem by a progressive elimination of different types of variables and to generalization of the trust region method.

In DIVE method each discipline is responsible for construction of its meta-model. The output of the meta-model depends only on the public parameters. For a given public parameters x , the response of each meta-model is optimal with respect to its private parameters. This constitutes disciplinary level. In this way it eliminates private variables. Along with the output \tilde{A}_i each discipline also provides the reliability index α_i of the output. When $\alpha_i < 0$, the meta model is valid at current design point. If $\alpha_i \geq 0$ it is necessary to re-construct the meta-model about current design point. Meta-models can be constructed using the neural network techniques, Kriging, etc. When the meta-model is a linear or quadratic approximation, the reliability constraint

$\alpha_i < 0$ defines what we call the trust region. On the other hand, if $\alpha_i \leq 0$ in the complete design space, DIVE method is similar to other meta-model based MDO methods. For this reason DIVE method is considered as a generalization of the trust region method.

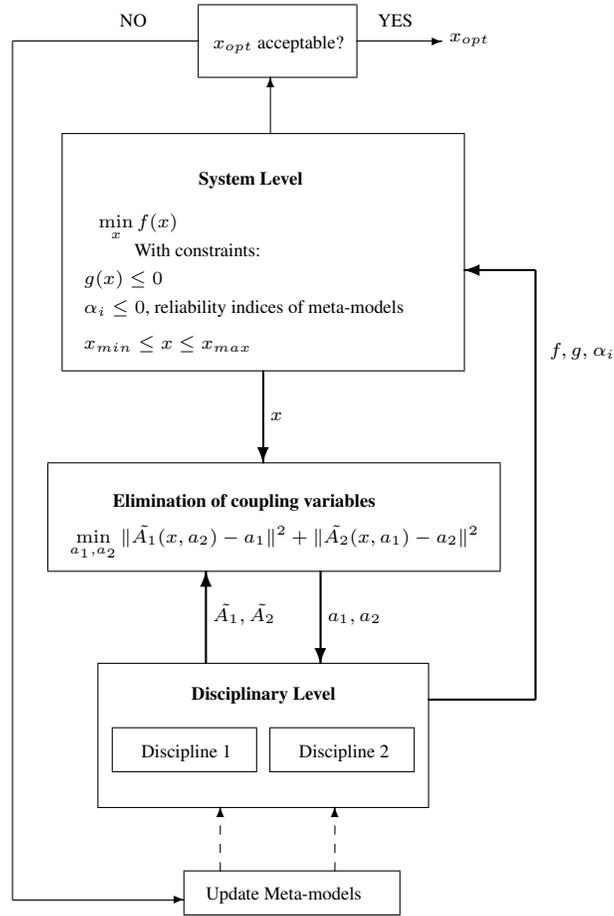


Figure 1.13. Diagram for DIVE formulation

Once meta-models are constructed, coupling variables a_1 and a_2 are eliminated by minimizing the residue of the state equations in a least squares sense:

$$\min_{a_1, a_2} \|\tilde{A}_1(x, a_2) - a_1\|^2 + \|\tilde{A}_2(x, a_1) - a_2\|^2. \quad (1.27)$$

This specific approach in DIVE to find the solution of state equations using minimization problem defer from other formulations where state equations are solved either using fixed point method (as in MDF, BLISS) or by adding equality constraints (as in AAO, IDF, CO). This characteristic makes DIVE a more robust method and allows use of efficient minimization techniques such as the method of Levenberg-Marquardt or the adjoint method.

After elimination of coupling variables, shared variables are treated at system level optimization. The optimization problem solved at system level is:

$$\begin{aligned} \min_x f(x) \\ g(x) \leq 0 \\ \alpha_i \leq 0 \end{aligned} \quad (1.28)$$

The solution of above optimization problem is checked for following points

- If the design point obtained is on the boundary of the validity domain of any of the meta model (that is $\alpha_i = 0$ for any i), a new meta-model is constructed around the design point and optimization procedure is repeated.
- If state equations are not satisfied with desired accuracy, meta-models are updated.

In this way, DIVE formulation also provides the quality index for the obtained optimal solution. Distinguishing features of DIVE formulation are:

- In DIVE method, priority is given to accurate solution of state equations of the system.
- Peculiar construction of disciplinary meta-model provides reliability index for the disciplinary output variables and thus DIVE becomes generalization of trust region method.

1.3.9. *Reduced order models and approximations*

In this section we take a brief review of different meta-modeling techniques. For detailed discussion on this topic we refer to chapters, *Response Surface Methodology (RSM) and Reduced Order Models (ROM)*, *Meta-modeling using Principal Component Analysis* and *Model Reduction for Coupled Problems* in this book. Meta models also known as surrogate models provide a convenient way to reduce MDO complexity. Any approximation has a local validity in an appropriate trust region, for example,

- 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.

1.3.9.1. *First order approximation*

The most simple way to build a local approximation is to differentiate (1.5) and (1.6), but its trust region is relatively small. We obtain

$$\begin{aligned} \tilde{a}_1 &= \partial_x A_1(x_{cur}, a_2(x_{cur}))(x - x_{cur}) \\ &+ \partial_{a_2} A_1(x_{cur}, a_2(x_{cur}))(a_2(x) - a_2(x_{cur})), \end{aligned} \quad (1.29)$$

and

$$\begin{aligned} \tilde{a}_2 &= \partial_x A_2(x_{cur}, a_1(x_{cur}))(x - x_{cur}) \\ &+ \partial_{a_1} A_2(x_{cur}, a_1(x_{cur}))(a_1(x) - a_1(x_{cur})). \end{aligned} \quad (1.30)$$

The adjoint method (reverse mode in automatic differentiation) could be considered for large scale problems [GRI 89, GRI 90, GRI 00, MOH 01, MOR 85, OST 71, ROS 92, SPE 80, STR 90, WEN 64].

1.3.9.2. *Approximation using parameterization methods*

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

$$a_1 = A_1(x) \quad (1.31)$$

$$a_2 = A_2(x, a_1). \quad (1.32)$$

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 (1.5) and (1.6) for several vectors x . Generally, we consider Design Of Experiments (DOE) methods to select appropriate vectors x^k , $k = 1, \dots, n$ where n is small.

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

1) Approximation of objective function and constraints using surface response methods [CRI 00, PLA 99, VAP 95] like polynomial approximation, neural networks, Support Vector Machine (SVM), Radial Basis Functions (RBF), etc. ;

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 Proper Orthogonal Decomposition (POD) [GUI 92, GUI 93, GUI 94, GUI 95, MAD 00, LAW 96, LEG 04, ROD 00, SAN 00, SER 98, SOB 00].

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 [MÄK 98, PÉR 98], and other global optimization algorithms [KEL 99b, KEL 99a, MON 99].

1.4. Application of MDO to conceptual design of super sonic business jet (SSBJ)

The test case of conceptual design of super sonic business jet (SSBJ) is proposed by DASSAULT aviation as a part of the OMD-RNTL project [OMD06] to compare performance of different MDO frameworks described in previous sections. We briefly discuss main features of this test case. Detailed description of this test case is referred to [RAV 07].

Three different disciplines, namely, structures, aerodynamics and propulsion constitute three coupled subsystems or computational modules in the design cycle of SSBJ.

- Structures: Computes mass of different parts of an aircraft such as wing, fuselage, tail, engine, etc., and the take-off weight as function of geometrical parameters of an aircraft and engine mass.

- Aerodynamics: Computes the lift and drag during cruise using aircraft mass, geometrical details and engine size as input parameters. The drag estimates from aerodynamics module are used by propulsion module for engine sizing operations.

- Propulsion: Performs engine sizing and estimates the engine thrust using drag estimates generated using aerodynamic analysis. The engine mass computed by propulsion module acts as an input to the structures module.

Output from these subsystems, namely the take-off weight and engine thrust constitute input for the performance module. The performance module computes the range, take-off distance and landing speed of an aircraft. Figure 1.14 shows the coupling between different disciplines under SSBJ test case and Table 1.1 enlists different design variables, their units and symbols used for their representation.

Catagory	Variable	Unit	Description
Cruise parameters	Z	m	Altitude
	X_{mach}		Mach number
	W_{fuel}	kg	Fuel mass
Wing geometry	S	m^2	Reference wing surface area
	ϕ_0^w	deg	Wing leading edge angle
	ϕ_{100}^w	deg	Wing trailing edge angle
	Xl^w		Taper ratio for wing
	t_c^w		Thickness to chord ratio for wing
Tail geometry	ϕ_0^t	deg	Tail leading edge angle
	ϕ_{100}^t	deg	Tail trailing edge angle
	Xl^t		Taper ratio for tail
	t_c^t		Thickness to chord ratio for tail
Fuselage geometry	D_{fus}	m	Fuselage diameter
Misc	α	deg	Angle of attack
	X_{fac}		Ratio of landing mass to take-off mass

Table 1.1. Different design variables in SSBJ test case

Catagory	Variable	Unit	Description
Interaction	TOW	kg	Take-off mass
	$F0$	N	Thrust

Table 1.2. Interaction variables in SSBJ test case

The design cycle consists of iterative solution procedure to find converged values of take-off weight and engine thrust under different constraints. Other design parameters such as lift, drag and engine size are functions of take-off weight and engine thrust. Take-off weight and engine thrust thus constitute shared variables. Their converged values represents the equilibrium amongst above three disciplines. Using take-off weight and engine thrust one aircraft range is computed using Breguet equation in performance module. Landing speed and take-off distance is computed using empirical relations [RAV 07]. The aim of this test case is to design SSBJ which has

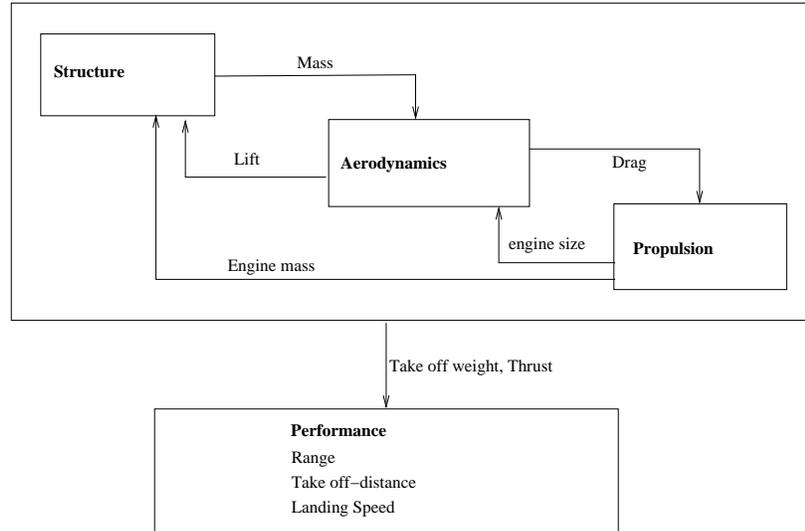


Figure 1.14. Coupling between different disciplines under DASSAULT SSBJ test case

minimum take off weight (TOW) subjected to following constraints on aircraft range, take-off distance ($D_{\text{take-off}}$) and landing speed (V_{approach}). These constraints are:

- The aircraft range, R , should be greater than or equal to 6500 km
- The take-off distance, $D_{\text{take-off}}$, should be less than or equal to 1828 m
- The landing speed, V_{approach} , should be less than or equal to 70 m/s

1.4.1. Results obtained using different frameworks

The SSBJ test case is formulated as a MDO problem using DIVE, MDF, IDF and BLISS formulations discussed in section 1.3. For DIVE method, meta-models are linear approximations constructed using gradient of disciplinary output variables. Table 1.3 to 1.5 shows results obtained using different MDO framework for SSBJ test case. Table 1.3 shows optimal values of design variables obtained using different MDO frameworks. Considerable differences exist in the optimal value of design variables pertaining to the geometry of the wing and tail computed by different disciplines.

– The trailing edge sweep angle, Φ_{100}^w , for the wing as computed by DIVE, MDF and IDF is about 5.5 deg whereas by BLISS method this value is -5.3 deg.

– The taper ratio Xl^w value for wing is less than 0.05 by BLISS, IDF and MDF method. Taper ratio for wing as computed by DIVE is 0.11.

Variables		DIVE	MDF	IDF	BLISS	min	max
Z	(m)	14585	14985	14845	14895	8000	18500
X_{mach}		1.6	1.6	1.6	1.6	1.6	2
S	(m^2)	100	100	100	100	100	200
ϕ_0^w	(deg)	46.1	51.8	51.9	46.8	40	70
ϕ_{100}^w	(deg)	5.5	5.6	5.4	-5.3	-10	20
Xl^w		0.11	0.05	0.05	0.05	0.05	0.5
t_c^w		0.04	0.046	0.0463	0.04	0.04	0.08
ϕ_0^t	(deg)	56	56	57	70	40	70
ϕ_{100}^t	(deg)	5	5	5	0	0	10
Xl^t		0.274	0.274	0.274	0.500	0.05	0.5
t_c^t		0.063	0.060	0.059	0.05	0.05	0.08
D_{fus}	(m)	2	2	2	2	2	2.5
W_{fuel}	(kg)	15000	15000	15000	15000	15000	40000
α	(deg)	15	15	15	15	10	15
X_{fac}		0.85	0.85	0.85	0.85	0.85	0.95
TOW	(kg)	33066	33207	32862	33065	1	100000
$F0$	(N)	62221	62704	61963	62220	1	300000

Table 1.3. Optimal design variables obtained using different MDO frameworks

– Leading edge sweep angle for the tail, Φ_0^t attains its maximum value whereas the trailing edge sweep angle for the tail, Φ_{100}^w attains its minimum value using BLISS method. Using DIVE, MDF and IDF, the leading edge sweep angle of the tail is 56 deg whereas the trailing edge sweep angle for tail is 5 deg.

– Taper ratio for the tail has optimal value of 0.5, which is the upper limit on the permissible values under BLISS framework and has a value of 0.274 under DIVE, MDF and IDF framework.

Rest of the design parameters have more or less similar values under different MDO frameworks.

Table 1.4 shows optimal values of disciplinary outputs under different MDO frameworks. The take-off weight as computed by BLISS and DIVE is 33066 kg and 33065 kg respectively. The take-off weight computed using IDF is 32862 kg and that with MDF is 33207 kg . It is to be observed that though take-off weight computed by IDF is minimum it is not the feasible solution as it violates constraints of landing speed.

This is due to the fact that in IDF priority is given to individual disciplinary feasibility and thus state equations are not solved accurately. Even when multi-disciplinary analysis is coupled with IDF it doesn't provide feasible solution as evident from the value of range, take-off distance and landing speed recorded under column IDF+MDA in Table 1.4. This underlines the fact that in MDO framework it is utmost important to accurately solve the state equations.

Output	DIVE	MDF	IDF	IDF +MDA	BLISS	min	max
TOW (kg)	33066	33207	32862	33141	33065		50000
$F0$ (N)	62221	62704	61963	62356	62220		
R (m)	6500	6501	6510	6443	6500	6500	
$D_{\text{take-off}}$ (N)	1828.0	1827.3	1826.4	1842.0	1828.0		1828
V_{approach} ($m.s^{-1}$)	70.0	70.0	69.9	70.2	70.0		70

Table 1.4. Optimal value of output variables obtained using different MDO frameworks

Table 1.5 shows number of calls made to disciplinary analysis under different MDO frameworks. It also includes calls to disciplinary analysis to compute gradient of disciplinary output variables and those required for the construction of meta-models. As evident, DIVE method requires minimum number of disciplinary analy-

Discipline	DIVE	MDF	IDF	BLISS
Structure	630	1459	1144	966
Aerodynamics	630	1459	1144	966
Propulsion	636	1490	1144	992
Performance	606	1552	1144	1044

Table 1.5. Number of calls for disciplinary analysis by different MDO frameworks

ses (about 630 for each disciplines) followed by BLISS (about 1000) and IDF. MDF formulation requires maximum number of disciplinary analyses, about 1450 for each discipline. These results also show the advantage of using meta-models at disciplinary level, as in DIVE method, rather than using system level meta models as utilized in

BLISS method. One must realize that for this test case there are only two shared or coupling variables and construction of disciplinary meta-models is easy. The final configuration obtained using different MDO methods is more or less similar. Therefore, performance index of different MDO methods is judged on the basis of computational cost measured in terms of number of calls to disciplinary analysis. With this performance index DIVE method outperforms other method. However, in more complicated cases, construction of disciplinary meta-model itself can be computationally expensive step. None-the-less it can be concluded that in meta-model based MDO, DIVE method outperforms other methods.

1.5. Game theory

Game theory has been used for multi-criteria optimization [RAM 02]. Then, it has been enhanced by J. Périaux et al. [MÄK 98, PÉR 98, WAN 92] 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 of totally different nature. The main advantage of game theory is to work in a multi-criteria 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 $A \times B$:

$$\begin{aligned} f_A : A \times B &\rightarrow \mathbb{R} \\ (x, y) &\mapsto f_A(x, y) \end{aligned} \quad (1.33)$$

and

$$\begin{aligned} f_B : A \times B &\rightarrow \mathbb{R} \\ (x, y) &\mapsto f_B(x, y) \end{aligned} \quad (1.34)$$

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 .

1.5.1. The Pareto equilibrium

1.5.1.1. 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

$$\begin{cases} f_A(x, y) \leq f_A(x^*, y^*) \\ f_B(x, y) < f_B(x^*, y^*), \end{cases}$$

or

$$\begin{cases} f_A(x, y) < f_A(x^*, y^*) \\ f_B(x, y) \leq f_B(x^*, y^*). \end{cases}$$

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.

1.5.1.2. Example

Let us consider a very simple example

$$\begin{cases} f_A = (x - 1)^2 + (x - y)^2 \\ f_B = (y - 3)^2 + (x - y)^2. \end{cases} \quad (1.35)$$

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

$$f_\lambda = \lambda f_A + (1 - \lambda)f_B, \quad (1.36)$$

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

$$f_\lambda = \lambda(x - 1)^2 + (1 - \lambda)(y - 3)^2 + (x - y)^2 \quad (1.37)$$

and then

$$\begin{aligned} \partial_x f_\lambda &= 2[\lambda(x - 1) + (x - y)] = 0 \\ \partial_y f_\lambda &= 2[(1 - \lambda)(y - 3) - (x - y)] = 0. \end{aligned} \quad (1.38)$$

We finally obtain

$$\begin{cases} (1 + \lambda)x - y = \lambda \\ -x + (2 - \lambda)y = 3(1 - \lambda). \end{cases} \quad (1.39)$$

1.5.2. The Nash equilibrium

1.5.2.1. Definition

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

$$\begin{cases} f_A(x^*, y^*) = \inf_x f_A(x, y^*) \\ f_B(x^*, y^*) = \inf_y f_B(x^*, y). \end{cases} \quad (1.40)$$

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

$$\begin{cases} \nabla_x f_A(x^*, y^*) = 0 \\ \nabla_y f_B(x^*, y^*) = 0 \end{cases} \quad (1.41)$$

1.5.2.2. Example

We consider again example (1.35)

$$\begin{cases} f_A = (x - 1)^2 + (x - y)^2 \\ f_B = (y - 3)^2 + (x - y)^2, \end{cases} \quad (1.42)$$

and the corresponding Nash equilibrium is given by

$$\begin{aligned} \begin{cases} \partial_x f_A = 0 \\ \partial_y f_B = 0 \end{cases} &\iff \begin{cases} y = 2x - 1 \\ x = 2y - 3 \end{cases} &\iff \begin{cases} x = \frac{5}{3} \\ y = \frac{7}{3} \end{cases} \end{aligned} \quad (1.43)$$

One may remark that the solution of (1.43) is exactly the same as in (1.39) 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.

1.5.3. The Stackelberg equilibrium

1.5.3.1. 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:

$$\min_x f_A(x, y_x) \quad (1.44)$$

$$\text{where } y_x = \arg \min_y f_B(x, y).$$

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

$$j(x) := f_A(x, y_x). \quad (1.45)$$

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.

1.5.3.2. Example

We still consider the previous example

$$\begin{cases} f_A = (x - 1)^2 + (x - y)^2 \\ f_B = (y - 3)^2 + (x - y)^2. \end{cases} \quad (1.46)$$

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

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

1.5.4. Application of the game theory to the multidisciplinary design

We simply have to consider equations (1.5), (1.6) and cost function (1.7)

$$a_1 = A_1(x, a_2) \quad (1.47)$$

$$a_2 = A_2(x, a_1) \quad (1.48)$$

$$\min_x f(x, a_1(x), a_2(x)) \quad (1.49)$$

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

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 1.3.7). But, here the system level cost function is replaced by disciplinary

cost functions. Discipline 1 solves the following problem

$$a_1 = A_1(x, \tilde{a}_2), \quad (1.50)$$

$$\tilde{a}_2 = \tilde{A}_2(x, a_1), \quad (1.51)$$

$$\min_{x_1} f_1(x_1, x_2, a_1(x), \tilde{a}_2(x)) \quad (1.52)$$

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

$$(1 - \Delta_2)x_{1\ cur} \leq x_1 \leq (1 + \Delta_2)x_{1\ cur}, \quad (1.53)$$

and discipline 2 solves the following problem

$$\tilde{a}_1 = \tilde{A}_1(x, a_2), \quad (1.54)$$

$$a_2 = A_2(x, \tilde{a}_1), \quad (1.55)$$

$$\min_{x_2} f_2(x_1, x_2, \tilde{a}_1(x), a_2(x)) \quad (1.56)$$

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

$$(1 - \Delta_1)x_{2\ cur} \leq x_2 \leq (1 + \Delta_1)x_{2\ cur}. \quad (1.57)$$

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

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.

1.6. Other approaches in collaborative design

In section 1.3 we discussed different MDO frameworks 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. All subsystems are considered at equal level and in principle there is no restriction on communications between subsystems. However, in certain industries like those in automotive sector, design cycles are not discipline specific but are product object driven and maintain multi-hierarchical structure with restriction on communication paths between different subsystems and levels [MIC 00]. Use of MDO approaches discussed in section 1.3 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 [KIM 01, MIC 99, MIC 00] for further details on ATC and its applications. Comparative properties of ATC and collaborative optimization and other approaches are discussed in [ALL 04]. A new framework based on nested ATC and collaborative optimization methodology for multi-level systems is described in [ALL 05].

Another approach for collaborative design is based on the theory of multi-agent game and collectives [KRO 04, WOL 02]. Unlike, hierarchical structure considered in earlier frameworks here subsystems are considered as players. Each player has strategies or moves for optimizing his objectives while interacting with other players who are simultaneously selecting their moves for optimizing their objective function. While doing so each player also tries to minimize the system objective function and receives incentives for his 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 his 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 [TUM 02, WOL 02, WOL 00], however, only recently were applied to aeronautical and other problems [BIE 04, BIE 05].

1.7. Conclusions

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 can not 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 public parameters, which are shared by all disciplines.
- Creation of disciplinary reduced order 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 and subjective criteria. The subjective criteria (such as aesthetics, manufacturability, etc.) 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.

1.8. Bibliography

- [ACH 95] ACHDOU Y., KUZNETSOV Y. A., PIRONNEAU O., “Substructuring preconditioners for the Q_1 mortar element method”, *Numer.Math.*, vol. 71, p. 419-449, 1995.
- [AGT 00] AGTE J. S., A tool for application of bi-level integrated system synthesis (BLISS) to multi-disciplinary optimization problem, 2000.
- [ALE 98] ALEXANDROV N. M., KODIYALAM S., Initial results of an MDO method evaluation study, Report num. AIAA-98-4884, NASA Langley, 1998.
- [ALE 00a] ALEXANDROV N. M., Analytical and Computational Aspects of Collaborative Optimization, Report num. NASA-2000-210104, NASA Langley, 2000.
- [ALE 00b] ALEXANDROV N. M., LEWIS R., “Algorithmic perspectives on problem formulations in MDO”, *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000, AIAA Paper 2000-4719.
- [ALE 00c] ALEXANDROV N. M., LEWIS R. M., Analytical and Computational Properties of Distributed Approaches to MDO, Report num. AIAA-2000-4718, NASA Langley, 2000.
- [ALL 04] ALLISON J. T., 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.
- [ALL 05] ALLISON J. T., KOKKOLARAS M., ZAWISLAK M., PAPALAMBROS P., “On the use of analytical target cascading and collaborative optimization for complex system design”, *6th World congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, 30 May-03 June 2005.
- [ALT 02] ALTUS T. D., A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS), Report num. NASA/CR-2002-211652, NASA Langley Research Center, 2002.
- [BAT 91] BATINA J. T., “Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis”, *AIAA Journal*, vol. 29(3), p. 327-333, 1991.

- [BIE 04] BIENIAWSKI S. F., WOLPERT D., KROO I., “Discrete, Continuous, and Constrained Optimization Using Collectives”, *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, August 30-September 1 2004.
- [BIE 05] BIENIAWSKI S. F., Distributed optimization and flight control using collectives, PhD thesis, University of Stanford, U.S.A., October 2005.
- [BRA 96] BRAUN R., GAGE P., KROO I., SOBIESKI I., Implementation and performance issues in collaborative optimization, Report , AIAA, 1996.
- [CEA 86] CEA J., “Conception optimale ou identification de forme, calcul rapide de la dérivée directionnelle de la fonction coût”, *M. A. A. N.*, vol. 20(3), p. 371–402, 1986.
- [CRI 00] CRISTIANINI N., SHAWE-TAYLOR J., *An introduction to support vector machines*, Cambridge University Press, 2000.
- [GRI 89] GRIEWANK A., “*Mathematical Programming: Recent Developments and Applications*”, Chapter On Automatic Differentiation, Kluwer Academic Publishers, 1989.
- [GRI 90] GRIEWANK A., JUEDES D., SRINIVASAN J., TYNER C., “ADOL-C, a Package for the Automatic Differentiation of Algorithms Written in C/C++”, *ACM Trans. Math. Software*, 1990.
- [GRI 00] GRIEWANK A., “Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation”, *SIAM*, 2000.
- [GUI 92] GUILLAUME P., MASMOUDI M., “Dérivées d’ordre supérieur en optimisation de domaines”, *C. R. Acad. Sci. Paris, Série I*, vol. 315, p. 859–862, 1992.
- [GUI 93] GUILLAUME P., MASMOUDI M., “Calcul numérique des dérivées d’ordre supérieur en conception optimale de forme”, *C. R. Acad. Sci. Paris, Série I*, p. 1091-1096, 1993.
- [GUI 94] GUILLAUME P., MASMOUDI M., “Computation of High Order Derivatives in Optimal Shape Design”, *Numerische Mathematik*, vol. 67, p. 231–250, 1994.
- [GUI 95] GUILLAUME P., MASMOUDI M., “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.
- [GUM 01] GUMBERT C. R., HOU J. W., NEWMAN P. A., Simultaneous aerodynamic and structural design optimization (SASDO) for a 2-d wing, Report num. AIAA-2001-2527, AIAA, 2001.
- [JAM 90] JAMESON A., “Control Theory Applications for Optimum Design of Aerodynamic Shapes”, *29th IEEE Conference on Decision and Control*, Honolulu, p. 176-179, Dec. 1990.
- [KEL 99a] KELLEY C. T., “Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition”, *SIAM Journal of Optimization*, vol. 10, num. 1, p. 43–55, 1999.
- [KEL 99b] KELLEY C. T., “Iterative methods for optimization”, *Frontiers in Applied Mathematics*, *SIAM*, 1999.
- [KIM 01] KIM H. M., Target cascading in optimal system design, PhD thesis, University of Michigan, U.S.A, 2001.

- [KOD 98] KODIYALAM S., YUAN C., Evaluation of methods for multidisciplinary design optimization, Phase I, Report , National Aeronautics and Space Administration, 1998.
- [KRO 04] KROO I., “Collectives and complex design”, *VKI lecture series on Optimization Methods and Tools for Multicriteria/Multidisciplinary Design*, November 15-19 2004.
- [LAW 96] LAWRENCE S., TSOI A. C., BACK A. D., “Function approximation with neural networks and local methods: bias, variance and smoothness”, *Australian Conference on Neural Networks*, Australian National Univ., p. 16–21, 1996.
- [LEG 04] LEGRESLEY P. A., ALONSO J. J., “Improving the Performance of Design Decomposition Methods with POD”, *10th AIAA/ISSMO Multidisciplinary Analysis*, AIAA, 2004, AIAA 2004-4465.
- [MAD 00] MADAY Y., PATERA A., TURINICI G., “A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations”, *J. Sci. Comput.*, vol. 17, num. 1-4, p. 437-446, 2000.
- [MÄK 98] MÄKINEN R., NEITTAANMÄKI P., PÉRIAUX J., “A genetic algorithm for multiobjective design optimization in aerodynamics and electromagnetics”, ET AL. P., Ed., *Computation Fluid Dynamics 98, Proceedings of the ECCOMAS 98 Conference*, vol. 2, Athens, Greece, Wiley, p. 418–422, 1998.
- [MAS 06] MASMOUDI M., PARTE Y., “Disciplinary Interaction Variable Elimination (DIVE) Approach for MDO”, *ECCOMAS CFD 2006*, 2006.
- [MIC 99] MICHELENA N., KIM H. M., PAMPALAMBROS P., “A system partitioning and optimization approach to target cascading”, *12th International Conference on Engineering Design*, Munich, Germany, 1999.
- [MIC 00] MICHELENA N., PAMPALAMBROS P., “Trends and challenges in system Design optimization”, *International Workshop on Multidisciplinary Design Optimization*, Pretoria, S. Africa, August 7-10 2000.
- [MOH 01] MOHAMMADI B., PIRONNEAU O., *Applied Shape Optimization for Fluids*, Oxford University Press, 2001.
- [MON 99] MONGEAU M., KARSENTY H., ROUZÉ V., HIRIART-URRUTY J.-B., “Comparison of public-domain software for black box global optimization”, *Optimization Methods and Software*, 1999.
- [MOR 85] MORGENSTERN J., “How to compute fast a function and all its derivatives, a variation on the theorem of Baur-Strassen”, *SIGACT News*, 1985.
- [OMD06] “<http://omd.lri.fr>”, 2006.
- [OST 71] OSTROVSKII G. M., VOLIN J. M., BORISOV W. W., “Über die Berechnung von Ableitungen”, *Wissenschaftliche Zeitschrift der Technischen Hochschule für Chemie, Leuna Merseburg*, vol. 13, p. 382–384, 1971.
- [PAD 99] PADULA S. L., KORTE J. J., DUNN H. J., SALAS A. O., Multidisciplinary Optimization Branch Experience Using iSight Software, Report num. NASA-99-209714, NASA, 1999.

- [PÉR 98] PÉRIAUX J., *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, John Wiley & Son Ltd, 1998.
- [PLA 99] PLATT J. C., “*Advances in Kernel Methods - Support Vector Learning*”, Chapter Fast training of support vector machines using sequential minimal optimization, p. 185-208, Num. 12, MIT Press, Cambridge, Massachusetts, 1999.
- [RAM 02] RAMOS A. M., GLOWINSKI R., PÉRIAUX J., “Nash equilibria for the multiobjective control of linear partial differential equations”, *Journal of Optimization Theory and Applications*, vol. 112, num. 3, p. 457–498, 2002.
- [RAV 07] RAVACHOL M., Rapport d’avancement. "Optimisation Multidisciplinaire", Report , DASSAULT AVIATION / OMD-RNTL, 2007.
- [ROD 00] RODRIGUEZ J. F., RENAUD J. E., WUJEK B. A., TRAPPETA R. V., “Trust Region Model Management In Multidisciplinary Design Optimization”, *Journal of Computational and Applied Mathematics*, vol. 124, num. 1-2, p. 139–154, 2000.
- [ROS 92] ROSTAING N., DALMAS S., “Automatic Differentiation Analysis and Transformation of Fortran Program Using a Typed Functional Language”, *International Conference on Computing Methods in Applied Sciences and Engineering*, 1992.
- [SAL 97] SALAS A. O., ROGERS J. L., A Web-Based System for Monitoring and Controlling Multidisciplinary Design, Report num. NASA 97-206287, NASA, 1997.
- [SAL 98] SALAS A. O., TOWNSEND J. C., Framework Requirement for MDO Application Development, Report num. AIAA-98-4740, AIAA, 1998.
- [SAN 00] SANDY D., BALKIN D., LIN K., “A neural approach to response surface methodology”, *Communications in Statistics, Theory and Methods*, vol. 29, num. 9-10, p. 2215-2227, 2000.
- [SER 98] SERAFINI D. B., A framework for managing models in nonlinear optimization of computationally expensive functions, PhD thesis, Rice University, 1998.
- [SHA 93] SHANKAR J., RIBBENS C., HAFTKA R., WATSON L., “Computational study of nonhierarchical decomposition algorithm”, *Computational Optimization and Applications*, 1993.
- [SOB 98] SOBIESKI J., AGTE J. S., SANDUSKY R. R., “Bi-Level Integrated System Synthesis (BLISS)”, , num. AIAA 98-4916, 1998.
- [SOB 00] SOBIESKI I. P., KROO I. M., “Collaborative Optimization Using Response Surface Estimation”, *AIAA Journal*, vol. 38, num. 10, p. 1931–1938, 2000.
- [SPE 80] SPEELPENNING B., Computing Fast Partial Derivatives of Functions Given by Algorithms, PhD thesis, University of Illinois, 1980.
- [STR 90] STRASSEN V., “*Handbook of Theoretical Computer Science*”, Chapter Algebraic Complexity Theory, Elsevier, Amsterdam, 1990.
- [TAN 02] TANG Z. L., DÉSIDÉRI J. A., Towards Self-Adaptive Parameterization of Bezier Curves for Airfoil Aerodynamic Design, Report num. RR-4572, INRIA, October 2002, URL:<http://www-sop.inria.fr/rapports/sophia/RR-4572.html>.

- [TUM 02] TUMER K., AGOGINO A., WOLPERT D. H., “Learning sequences of actions in collectives of autonomous agents”, *First International Joint Conference on Autonomous and Multi-Agent Systems*, Bologna, Italy, July 2002.
- [VAP 95] VAPNIK V. N., *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.
- [WAN 92] WANG J. F., 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.
- [WEN 64] WENGERT R. E., “A Simple Automatic Derivation Evaluation Program”, *Comm. ACM*, vol. 7, p. 463–464, 1964.
- [WOL 00] WOLPERT D. H., WHEELER K., TUMER K., “Collective Intelligence for Control of Distributed Dynamical Systems”, *Europhysics Letters*, vol. 49, num. 6, p. 708-714, 2000.
- [WOL 02] WOLPERT D. H., TUMER K., “Collective Intelligence, Data Routing, and Braess’ Paradox”, *Journal of Artificial Intelligence Research*, 2002.