

Coupling Functions Treatment in a Bi-Level Optimization Process

Benoît Guédas* - Philippe Dépincé

Ecole Centrale de Nantes, Institut de Recherche en Communication et Cybernétique de Nantes, France

The optimization of complex engineering systems is often a mix of multi-objective optimization process - each service, discipline has to fulfill several objectives - and multidisciplinary optimization process - several disciplines are required. The disciplines are bound to each other: outputs of one discipline are used as input of other disciplines (the coupled variables) and a discipline can not have a direct access and knowledge to the whole set of variables. An approximation of the coupled variables is thus needed. The Collaborative Optimization Strategy for Multi-Objective Systems (COSMOS) has been developed at IRCCyN to perform simultaneously multi-objective and multidisciplinary optimization while discipline autonomy is guaranteed. It uses a simple method for the approximation of coupled variables and assumes that the quality of the approximation will increase as the algorithm converges to optimum solutions. In this paper, experiments are made to verify whether this assumption is true. We show that satisfying results are found on some test problems but limits of the methods are pointed out.

© 2008 Journal of Mechanical Engineering. All rights reserved.

Keywords: multidisciplinary optimization, multi-objective optimization, genetic algorithms, coupled problems

1 INTRODUCTION

Nowadays, the designer has to face the continuous growing complexity of engineering problems, but also, the increasing economic competition that have led to a specialization and distribution of knowledge, expertise, tools and work sites. Consequently, multi-objective optimization (MOO) and multidisciplinary design optimization (MDO) are more and more used to provide one solution or an optimal set of solutions.

While single-discipline optimization is mature, the design and optimization of complex systems (more than one discipline) is still quite young. Since the white papers provided in 1991 and 1998 by the AIAA [18] and [12], lot of research has been done in the multidisciplinary optimization domain: at the beginning centered on the aerospace industries, they are now used in different kind of enterprises (automotive, ship building, etc.) which expect from such a tool a way to improve their products, their organizations. Alexandrov and Lewis [1] defined MDO as a "systematic approach to optimization of complex coupled engineering systems where "multidisciplinary" refers to the different aspects that must be included in the design problem".

A classical way to describe a multidisciplinary problem is presented in Figure 1. In a multidisciplinary problem, each sub-system (discipline) has its own design variables, objective and constraint functions. Some design variables, common to at least two sub-systems, are called common variables. Disciplinary outputs from one discipline can be needed to evaluate another sub-system. In this case there is a coupling between two disciplines and these variables are called coupling variables [8] and [5]. The third variable type, state variables, are internal variables particular to one discipline: they represent conditions that have to be satisfied within the discipline. In each discipline an evaluation/analysis is conducted that allows to compute the outputs: functions, constraints and coupling variables if needed.

Frequently, complex systems are non-hierarchical implying that there is no reason to process the optimization of one sub-system before another [5]. In the optimization process of such systems, the presence of coupling functions and their recognition constitutes a real challenge for researchers.

Several methods have been designed to deal with coupling problems (MDF, IDF, AAO, CO, CSSO, BLISS, etc. - section 2), but they are not

*Corr. Author's Address: Ecole Centrale de Nantes, Institut de Recherche en Communication et Cybernétique de Nantes, 1, rue de la Noë - BP 92 101, 44321 Nantes CEDEX 3, France, benoit.guedas@irccyn.ec-nantes.fr

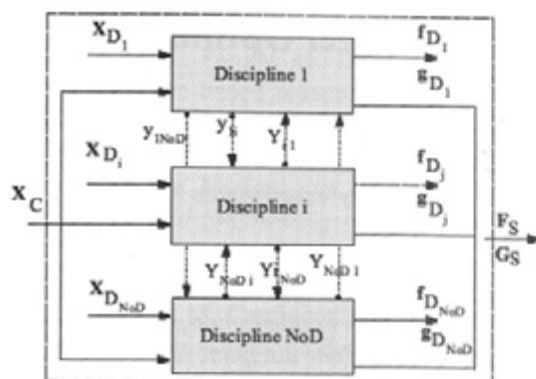


Fig. 1. A fully coupled disciplines system

suitable for the extended enterprise context where disciplines and tools are distributed on multiple sites. The main drawbacks of these approaches are *i)* the unique solution given to the designer and *ii)* their mathematical formulation that is not always adapted to the industrial context: most of these methods centralize the optimization at the system level while it should be handled by sub-system levels. Collaborative Optimization Strategy for Multi-Objective Systems (COSMOS) [19] is a method aiming to fill the gap between classical MDO methods and industrial needs by: *i)* taking advantages of the multi-objective genetic algorithms and provides the designer with a set of optimum solutions and *ii)*, giving more autonomy to the disciplines.

The next part of this paper will present the MDO methods: the classical ones based on exact mathematical approaches and some that try to simulate an engineering process and are based on multi-objective genetic algorithm. The third part introduces the problems caused by the autonomy of disciplines in the resolution process. Parts 4 and 5 describe the test examples and the result obtained by COSMOS method on coupled problems. Finally, some conclusions and perspective are given.

2 COUPLING FUNCTIONS IN MULTIDISCIPLINARY OPTIMIZATION METHODS

The ideal optimization process to solve a multidisciplinary optimization problem consists in separating the analysis phase from the optimization phase: the *multidisciplinary analysis* (MDA) computes the set of feasible solutions then the optimization selects the optimum from the previous set. This approach is not possible in practice

because of the high computational cost required to determine the whole set of feasible solution. Moreover in most problems the disciplines cannot easily exchange data between each others.

2.1 A Multi-Objective Coupled Problem

Let's consider a simplified model of two disciplines D_1 and D_2 (it can be generalized to n disciplines). Each discipline D_i has a state equation $E_i(x_c, x_i, y_i, u_i) = 0$. We will consider that there is an implicit function $e_i: X_c \times X_i \times Y_i \rightarrow U_i$, where $x_c \in X_c$ is the design variables that are shared among disciplines (common variables), $x_i \in X_i$ is the vector of disciplinary variables, $y_i \in Y_i$ is a parameter given by the discipline D_j , and $u_i \in U_i$ the vector of state variables given by the state equation e_i . This results in the following coupled problem:

$$D_1 \begin{cases} u_1 = e_1(x_c, x_1, y_2) \\ y_1 = l_1(x_c, x_1, u_1) \end{cases} D_2 \begin{cases} u_2 = e_2(x_c, x_2, y_1) \\ y_2 = l_2(x_c, x_2, u_1) \end{cases} \quad (1)$$

Finding the feasible solutions to this problem is called multidisciplinary analysis (MDA). We call A the set of feasible solutions of the problem (the multidisciplinary feasible solutions):

$$A = \{(x_c, x_1, x_2, u_1, u_2) \in X_c \times X_1 \times X_2 \times U_1 \times U_2 \mid u_1 = e_1(x_c, x_1, l_2(x_c, x_2, u_2)) \wedge u_2 = e_2(x_c, x_2, l_1(x_c, x_1, u_1))\} \quad (2)$$

Multidisciplinary optimization consists in adding an optimization problem to the satisfaction of the coupled problem. We will focus on the case where each discipline has its own optimization problem which is to minimize an objective function $f_i: X_c \times X_i \times U_i \rightarrow \mathbb{R}^{p_i}$ with p_i the number of objectives. A standard formulation of a two disciplines multidisciplinary optimization problem is then:

$$D_1 \begin{cases} \min_{x_c, x_1} f_1(x_c, x_1, u_1) \\ u_1 = e_1(x_c, x_1, y_2) \\ y_1 = l_1(x_c, x_1, u_1) \end{cases} D_2 \begin{cases} \min_{x_c, x_2} f_2(x_c, x_2, u_2) \\ u_2 = e_2(x_c, x_2, y_1) \\ y_2 = l_2(x_c, x_2, u_1) \end{cases} \quad (3)$$

The set S of solutions to the problem is then:

$$S = \{(x_c^*, x_1^*, x_2^*) \mid \exists (u_1^*, u_2^*) \in U_1 \times U_2 \text{ s.t. } (x_c, x_1, x_2, u_1, u_2) \in A \wedge \nexists (x_c, x_1, x_2, u_1, u_2) \in A \text{ s.t. } f(x_c, x_1, x_2, u_1, u_2) < f(x_c^*, x_1^*, x_2^*, u_1^*, u_2^*)\} \quad (4)$$

With $<$ the order relation of Pareto dominance: given $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$, $a < b \Leftrightarrow \forall i \in \{1, \dots, n\}, a_i \leq b_i \wedge \exists j \in \{1, \dots, n\}, a_j < b_j$.

Unfortunately, such a set of optimum solutions is intractable under the hypothesis of partitions of the variables in each discipline. Indeed, a discipline i cannot access another discipline j variables and does not know its coupling function.

Most of the MDO methods reported in the literature are developed specifically for single-objective problems with continuous variables and differentiable objective. These MDO methods are classified in two groups: mono-level and bi-level. The single-level (mono-level) group implies optimization at the supervisor level only. The bi-level group allows each discipline to manage its own optimization regarding its design variables. Multidisciplinary problems are often written in a simpler form, where the state variables are directly given to the other discipline, so they are also coupling variables:

$$D_1 \left\{ \begin{array}{l} \min_{x_c, x_1} f_1(x_c, x_1, y_1) \\ y_1 = e_1(x_c, x_1, y_2) \end{array} \right. D_2 \left\{ \begin{array}{l} \min_{x_c, x_2} f_2(x_c, x_2, y_2) \\ y_2 = e_2(x_c, x_2, y_1) \end{array} \right. \quad (5)$$

This notation will be used in the next section to present some multidisciplinary optimization methods.

2.2 Mono-Level Approaches

The mono-level family contains three multidisciplinary methods: Multidisciplinary Feasible (MDF), All-At-Once (AAO) and Individual Discipline Feasible (IDF) [16], [1], [14] and [6]. All the given formulations have different ways to handle the dependency of coupling functions. Dennis *et al.* [9] proposed an extension of all the above methodologies to the optimization of system of systems.

2.2.1 Multidisciplinary Feasible (MDF)

MDF is the most used approach to solve a MDO problem. A complete multidisciplinary analysis is performed for each choice of the design variables by the optimizer. This is conceptually very simple, and once all disciplines are coupled to form one single multidisciplinary analysis module, one can use the same techniques used in single discipline optimization.

In this formulation the optimization variables are the design variables, the optimization is global and each iteration gives a feasible solution. Moreover the evaluation within the

disciplines are independent. Drawbacks are the computational effort and the lack of guaranty for the coupling variables to converge to a feasible solution.

$$\left\{ \begin{array}{l} \min_{(x_c, x_1)} f_1(x_c, x_1, y_1) \\ \min_{(x_c, x_2)} f_2(x_c, x_2, y_2) \\ \text{s.t. } y_1 = e_1(x_c, x_1, y_2) \\ y_2 = e_2(x_c, x_2, y_1) \end{array} \right. \quad (6)$$

The optimization variables are x_c, x_1 and x_2 .

At each optimization step, the set of feasible solutions - described as A in (2) - is computed. The system of coupling equations must be solved. A fixed point iteration (FPI) algorithm, often used in this case, may not converge if the functions are not convex and may avoid hidden solutions [3].

2.2.2 All at Once (AAO)

All the variables (design, coupling, state) are considered as design variables and the analysis system equations becomes constraint. Hence, CPU consuming iterative analysis of sub-system are skipped but it increases the dimension of the design space. The problem formulation can be expressed by:

$$\left\{ \begin{array}{l} \min_{(x_c, x_1, y_2)} f_1(x_c, x_1, y_1) \\ \min_{(x_c, x_2, y_1)} f_2(x_c, x_2, y_2) \\ \text{w.r.t. } y_1 - e_1(x_c, x_1, y_2) = 0 \\ y_2 - e_2(x_c, x_2, y_1) = 0 \end{array} \right. \quad (7)$$

The optimization variables are x_c, x_1, x_2, y_1 and y_2 .

2.2.3 Individual Feasible (IDF)

IDF is a compromise between AAO and MDF. At each point, each discipline is feasible but the whole system will only be feasible at the end. In this methodology, coupling variables are added to design variables and some auxiliary variables, z_j , are introduced to allow decoupling the disciplines. Some equality constraints are added that allow compatibility between coupling and auxiliary variables. This substitution relaxes the coupling between disciplines: for some iterations, a point cannot fulfill all the coupling.

$$\begin{cases} \min_{(x_c, x_1, x_2)} f_1(x_c, x_1, y_1) \\ \min_{(x_c, x_1, x_2)} f_2(x_c, x_2, y_2) \\ \text{s.t. } y_1 = e_1(x_c, x_1, z_2) \\ y_2 = e_2(x_c, x_2, z_1) \\ \text{w.r.t. } y_1 - z_1 = 0 \\ y_2 - z_2 = 0 \end{cases} \quad (8).$$

The optimization variables are x_c, x_1, x_2, z_1 and z_2 .

In mono-level approaches, the optimization problem is seen as a single global problem and all the variables are accessible. Multi-level approaches give more autonomy to the disciplines by allowing them to solve their own optimization problem locally.

2.3 Multi-Level Approaches

In the case of bi-level optimization method, the original optimization problem is divided into optimization at both system and sub-system levels. Coordination between sub-systems is managed by an optimizer in charge of solving inconsistencies between the disciplines. Several strategies have been developed and the most discussed are Collaborative Optimization (CO) [7], and Concurrent SubSpace Optimization (CSSO) [20]. Other methods like Bi-Level Integrated System Synthesis (BLISS) [15], Analytical Target Cascading (ATC) [2] or Physical Programming (PP) [17] have been developed but will not be detailed in this paper. The two firsts are part of the Discipline Feasible Constraint (DFC) group. The primary features of each of these architectures include : *i*) the use of heterogeneous hardware or software, specific to the domain, to solve the subspace optimization problems, *ii*) the decomposition keeps domain-specific constraint information in the sub-problem, *iii*) the system leaves most of the design decisions (selection of local variables) to the disciplinary groups that understand the local formulation.

2.3.1 Collaborative Optimization (CO)

In CO subspace optimizers are integrated with each subsystem. Through sub-system optimization each discipline can control its own set of local design variables and is in charge of satisfying its own domain specific constraints. Explicit

knowledge of the other groups constraints or design variables is not required. The objective of each subsystem optimizer is to agree upon the values of the interdisciplinary variables with the other groups. A system level optimizer is employed to coordinate this process while minimizing the overall objective. It promotes disciplinary autonomy while achieving interdisciplinary compatibility.

The subsystem optimizer does not allow discipline optimization but only tries to reach consistency upon the common and coupled variables. The optimization process remains global at the system level.

All the methods described above are not designed for multi-objectives problems and give only a single solution to the designer.

2.4 Multi-Objective Multi-Level Approaches

As far as we are concerned, main advantage of MDO methods should focus on their ability to decompose a multidisciplinary problem into several sub-problems of manageable size that can be solved simultaneously. According to the current complexity and antagonist objectives to achieve, it should also be able to provide a set of solutions (not only a single one that relies on an *a priori* choice of the designers) and finally MDO should be adapted to the structure of the enterprise and the way design of systems including several disciplines is conducted.

The three methods presented thereafter are a first answer to such specifications. They can solve MDO problems that are decomposed into a hierarchy of several subsystem-level problems each of which has multiple objectives and constraints. Among different optimization algorithms that can be used for solving the subsystem problems, genetic algorithms (GAs) are used in the three methodologies. Using a population based optimization approach at both levels (i.e., system and subsystem levels) implies that a compromise as to be found at the system level to map fitness of solutions from multiple Pareto sets to a single system level candidate solution.

2.4.1 Multidisciplinary Optimization and Robust Design Approaches Applied to Concurrent Engineering (MORDACE)

The MORDACE method [11] is based on a robust design approach: finding solutions that are

robust with respect to changes in variable values due to discipline interactions. The MORDACE approach allows to independently perform the different discipline optimizations.

Each discipline aims at finding optimum solution with respect to its own design variables thank to a Multi-Objective Genetic Algorithm (MOGA) in order to obtain for each discipline the Pareto frontier as the set of best solution design. When the independent optimization processes are finished, the designer has to find a compromise on common variable values. Changes in common variable values in order to find the best trade-off may worsen performance levels. This difficulty is solved by adding to the set of design objectives f_p , a function that minimizes the effect of values variation of common variables. As disciplines simultaneously minimize objective functions f_i and sensitive function, they are always multi-objectives. Among available designs, the procedure chooses Pareto designs plus all individuals that dominate the original one with regard to different disciplines. Then, it defines all possible couples made up of solutions proposed by discipline 1 and 2, respectively. At this stage, the calculation of a distance parameter allows efficient solutions to be sorted out from the very large set of all possible couples. Thus, a limited number of couples are automatically chosen. Those solutions show small difference between discipline 1 and 2 common variable values, and they are robust with regard to changes in those values. Then, performances and coupling functions of the compromise designs defined by the new vectors of variables have to be verified.

Within the MORDACE method the designer needs to use a compromise method limited by the number of evaluation of potential solutions the designer allows.

MORDACE uses approximations (response surfaces) of the coupling functions in each discipline.

2.4.2 Entropy-Based Multi-Level Multi-Objective Genetic Algorithm (E-MMGA)

This method relies on a decomposition of the optimization at the disciplinary level. A first proposal was given in [13], but do not take into account the coupling functions and was limited to hierarchical system.

Each multi-objective GA at the sub-problems operates on its own population of (x_c, x_i) . The population size, n , for each sub-problem is kept the same. In addition, E-MMGA maintains two populations external to the sub-problems: the grand population and the grand pool. Both are populations of complete design variable vector: $(x_c, x_1, \dots, x_d)$. The grand population is an estimate of the solution set to the overall optimization problem. The grand pool is an archive of the union of solutions generated by the sub-problems. The size of the grand population is the same as the sub-problem population size, n . The size of the grand pool is d times the size of the subsystem's population (d is the number of sub-problems or disciplines).

The population of the grand population set is used as the initial population for each sub-problem. Since the sub-problem multi-objective GAs operates on its own variables (x_c, x_i) , only the chromosomes corresponding to x_c and x_i are used in the i^{th} sub-problem. After each run of sub-problem multi-objective GA there will be d populations having n individuals each. As each of the d populations contains only the chromosomes of only (x_c, x_i) , $i = 1, \dots, d$, they are completed using the rest of the chromosome sequence $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$ from the grand pool. After the chromosomes in all d populations are reconstituted to form the complete design variable vector, they are added to the grand pool. Then based on an entropy index that preserves the diversification of the solutions set, n individuals are chosen within the grand pool and replace the n individuals from the grand population.

An important drawback is that the size of the grand pool increases very quickly with the number of disciplines and individuals. A variant has been proposed in [4]: in each subsystem only one solution is selected on the Pareto frontier and its objectives and constraints values are used to assign a fitness value for the system level individual. The so-called best solution for each disciplines is chosen by an algorithm thank to the designer or decision maker preferences. The coupling functions are taken into account thanks to supplementary constraints -added both at system and subsystem levels- and auxiliary variables. Note that the shared variables can be treated as parameters in the subsystems and it reduces the dimensionality of the subsystem level optimization problems. In this last case, the coupling variable values are not passed to the system level.

2.4.3 Collaborative Optimization Strategy for Multi-Objective Systems (COSMOS)

Two variants (COSMOS-G and COSMOS-L [19]) have been proposed and the fundamental difference between them resides in a different treatment of common design variables. In the following, COSMOS refers to COSMOS-G.

Lets n the size of the population and d the number of disciplines. During the initialization the supervisor creates a population of common design variables X_c , with $x_{c,k}$ the k^{th} element of the n -tuple $X_c = (x_{c,1}, \dots, x_{c,n})$. Each discipline i also creates a population of disciplinary variables $X_i = (x_{i,1}, \dots, x_{i,n_i})$. In order to get a fully determined population, the supervisor sends the n -tuple of coupling design variables X_c to each discipline. Each discipline i builds a disciplinary population $Pop_i = \{(x_{c,k}, x_{i,k}) | 1 \leq k \leq n\}$ for which it can evaluate objective and constraint functions. An initial population can be created by the aggregation of common and disciplinary design variables and saved in $Pop_{\text{memorized}}$.

Optimization at sub-system level: The supervisor provides a set of common design variables X_c to the disciplines. Each discipline i optimizes the design variables of a population of individuals $(x_{c,k}, x_{i,k})$ where $1 \leq k \leq n$. The n -tuple X_c is fixed in order to keep the disciplinary population coherent with the other disciplinary populations. At the end of the disciplinary optimizations, each discipline sends a vector of disciplinary design variables optimized X_i^* to the supervisor. Since the vector of common design variables has not been modified in the discipline, a global population can be built and is naturally coherent: $Pop_{\text{current}} = \{(x_{c,k}, x_{i,k}^*) | 1 \leq k \leq n\}$.

Optimization at system level: The goal is to propose new and better common design variables (in order to improve the population). So, the current population, Pop_{current} , is assembled with the memorized population, $Pop_{\text{memorized}}$, in order to provide a double-sized population: Pop_{double} . This population is ranked by the Fonseca and Fleming's criterion (notion of Pareto domination) according to all the objective functions of the problem. The best individuals are selected to build a normal-sized population, Pop_{current} . This population will be sent to the disciplines. In parallel, cross-over and mutations are made on the common design variables of the population. This new population

is saved: $Pop_{\text{memorized}}$. It will be evaluated once by the disciplines in order to determine its objective and constraint functions.

Coupling function treatment: All the values of the coupling variables y_i computed in the discipline i at the end of its sub-system optimization are stored in an array. This array contains all the couples $(y_{i,k}, x_{c,k})$ where $x_{c,k}$ is the vector of coupled variables of the k^{th} individual of the population, and $y_{i,k}$ is the coupled variables computed by the coupling function l_i in the discipline i with the common $x_{c,k}$ and disciplinary variables $X_{i,k}^*$ obtained at the end of the sub-system optimization. When another discipline j needs the value of y_j for all its $x_{c,k}$ it will search the corresponding $y_{i,k}$ - when it exists - in the array. If the array does not contain the desired $x_{c,k}$, the closest is picked. In other words, for each discipline i , there is a table T_i of couples (x_c, y_i) such as $T_i \subseteq X_c \times Y_i$. Thus, T_i is the graph of the relation $\mathcal{T}_i = (X_c, Y_i, T_i)$ which is given to the discipline j .

Our main objective, in this paper, is to study the behavior of COSMOS on coupled problems and more precisely the evolution of the error of approximation of the coupled variables introduced by using the relation $\mathcal{T}_i$ instead of the function l_i . We will first introduce the problems caused by the separation of the design problem into more autonomous sub-problems.

3 PROBLEMS CAUSED BY DISCIPLINE AUTONOMY

In order to study the treatment of coupled functions, we need to point out the types of difficulties that coupled functions and division of work in subsystems may introduce in multidisciplinary problems.

From a global point of view, a multi-objective multidisciplinary problem is composed of two steps: analysis and optimization. Within the natural organization - set A in Figure 2 - analysis could be performed on the whole system aside from the optimization but distribution of work in multiple disciplines implies to split the problem in a way that suit to industrial context - set B in Figure 2. This raises new difficulties in problem solving because optimization and analysis are no more fully solved at each step. Indeed, they are split in two sub-problems in which optimization and analysis are partially performed for each discipline.

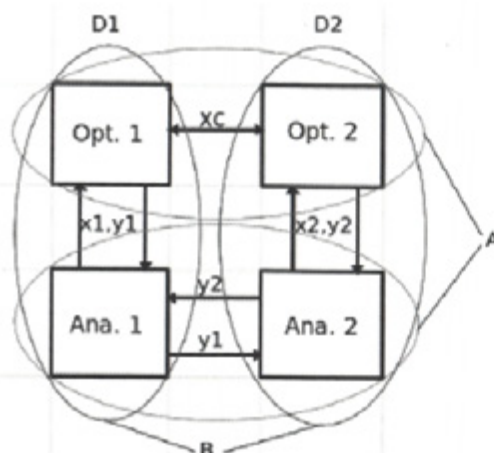


Fig. 2. Decomposition of the problem

3.1 Dependency Between Optimization and Analysis

The mix of optimization and analysis in each discipline can lead to two problems:

1. optimization disturbs analysis: optimization could tend towards solutions that may not be feasible
2. analysis disturbs optimization: analysis tend towards non-optimum solutions

1) Optimization leads analysis

Optimization is treated first, and the results are given to the coupling functions. The values computed by the analysis do not influence the objective values, but just ensure that the design variables allow to find a feasible solution or not. The design variables are seen by the coupling functions as simple parameters so it cannot influence on their choice. The multidisciplinary feasibility becomes more difficult to reach.

2) Analysis leads optimization

The analyzers give solutions to the optimizers which select the optimum. The analyzer may give solutions which are not optimum.

COSMOS uses the first solution: the first goal is to reach the disciplines objectives, and the coupled variables are just results computed at the end of the disciplinary optimization process that are needed in other disciplines. This can be illustrated by using another notation derived from the standard two discipline multidisciplinary optimization problem (1) where the objective is to minimize the state variables:

$$D_1 \begin{cases} \min_{x_1, y_1} u_1 := e_1(x_c, x_1, y_2) \\ y_1 = l_1(x_c, x_1, u_1) \end{cases} \quad D_2 \begin{cases} \min_{x_2, y_2} u_2 := e_2(x_c, x_2, y_1) \\ y_2 = l_2(x_c, x_2, u_1) \end{cases} \quad (9).$$

In the notation described in (5), coupling functions and state equations are confused, and the evaluation of the objective function comes after the analysis. However, in the latter notation (9), objective and state function are confused, and evaluation of the coupling functions only comes after optimization.

From this point of view, disciplinary optimization is more important than multidisciplinary consistency. This means that COSMOS is more suited to problems where disciplinary optimization is difficult but coupling between disciplines is weak.

Moreover, if we remove u_i in the right side of the coupling functions, the resolution with classical mono-level methods is obvious because the multidisciplinary analysis consists in function evaluations. This sort of coupling is only difficult when optimizers and analyzers cannot access all the disciplinary variables at the same time. These organizational constraints cannot be handled by mono-level methods, only by multi-level ones as presented previously. An approximation of the coupling function has to be found.

3.2 Approximation of Coupling Variables

A discipline D_i needs a coupling variable y_j from discipline D_j to solve its problem. Giving the value of the coupling variable y_j at a time t is not useful without knowing the parameters x_c, x_j, y_i used to find it. One way to do it, is to let the system level in charge of proposing a set of variables to the disciplines and ensure that it is consistent with the values computed in the disciplines (as in CO). Another one, is to use an approximation of the coupling function l_j in the discipline D_i (CSSO, MORDACE). COSMOS has a different approach using an array at each system iteration, the x_c are consistent, the x_i are the ones obtained at the end of optimization of each discipline and the y_j are approximated using the table T_i .

If we call $\tilde{y}_i$ the approximation of y_i found in T_i , it seems that as the population converge to the optimum solutions, the distance between $\tilde{y}_i$ and y_i tends toward zero. The next section describes the test problems designed to verify this hypothesis.

4 TEST PROBLEMS

To test the behaviour of COSMOS, we have implemented several examples. Problems #1, #2 and #3 are built to show normal conditions. #4 and #5 were especially designed to exhibit specific behaviours in limit conditions.

All the problems given are multi-objectives per sub-systems. They remain simple because our goal is to understand the behaviour of the approximation method on coupled problems. Thus the results should not be hidden by the complexity of the other parts of the problem.

4.1 Problem #1

$$x_c \in [0; 10], x_1, x_2, x_3, x_4 \in [0; 20]$$

Sub-system 1

$$\begin{cases} \min_{x_c, x_1} f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 \\ \min_{x_c, x_2} f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 9)^2 \\ y = \frac{x_1}{f_2(x_c, x_2) + 0.1} \end{cases} \quad (10)$$

Sub-system 2

$$\begin{cases} \min_{x_c, x_3} f_3(x_c, x_3) = (x_3 - 4)^2 + (x_c - 5)^2 \\ \min_{x_c, x_4} f_4(x_c, x_4) = (y - 5)^2 + (x_4 - 5)^2 \\ \quad + (x_c - 9)^2 \end{cases} \quad (11)$$

4.2 Problem #2

$$x_c, x_1, x_2, x_3, x_4, x_5, x_6 \in [-20; 20]$$

Sub-system 1

$$\begin{cases} \min_{x_c, x_1, x_2} f_1(x_c, x_1, x_2) = (x_1 - 1)^2 + x_2^2 \\ \min_{x_c, x_1, x_2} f_2(x_c, x_1, x_2) = x_c \times (x_1 - 3)^2 \times (x_2 - 7)^2 \\ \min_{x_c, x_1, x_2} f_3(x_c, x_1, x_2) = x_1 + x_2 + 1 \\ y_{12} = y_{13} = 10 \times x_1 + x_c \end{cases} \quad (12)$$

Sub-system 2

$$\begin{cases} \min_{x_c, x_3, x_4} f_4(x_c, x_3, x_4) = x_3^2 + (x_4 - 3)^2 \\ \min_{x_c, x_3, x_4} f_5(x_c, x_3, x_4) = y_{12} + (x_3 + x_4)^2 + (x_c - 4)^2 \\ \min_{x_c, x_3, x_4} f_6(x_c, x_3, x_4) = y_{32} \times x_3 \times x_4 \times x_c \end{cases} \quad (13)$$

Sub-system 3

$$\begin{cases} \min_{x_c, x_5, x_6} f_7(x_c, x_5, x_6) = 7 \times y_{13}^2 / x_c^2 + 0.1 \\ \min_{x_c, x_5, x_6} f_8(x_c, x_5, x_6) = (x_5 - 1)^2 + (x_6 - 9)^2 \\ y_{32} = 10 \times x_c + x_5 \end{cases} \quad (14)$$

4.3 Problem #3

$$x_c, x_1, x_2, x_3, x_4 \in [0; 20]$$

Sub-system 1

$$\begin{cases} \min_{x_c, x_1} f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 + (y_3 - 5)^2 \\ \min_{x_c, x_2} f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 7)^2 + (y_4 - 7)^2 \\ y_1 = \frac{x_2}{\sqrt{x_c + x_1 + 0.1}} \\ y_2 = \frac{x_1}{\sqrt{x_c + x_2 + 0.1}} \end{cases} \quad (15)$$

Sub-system 2

$$\begin{cases} \min_{x_c, x_3} f_3(x_c, x_3) = (x_3 - 3)^2 + (x_c - 5)^2 + (y_1 - 3)^2 \\ \min_{x_c, x_4} f_4(x_c, x_4) = (x_4 - 5)^2 + (x_c - 9)^2 + (y_2 - 9)^2 \\ y_3 = \frac{x_3}{\sqrt{x_c + x_4 + 0.1}} \\ y_4 = \frac{x_4}{\sqrt{x_c + x_3 + 0.1}} \end{cases} \quad (16)$$

4.4 Problem #4

$$x_c \in [-50; 50], x_1, x_2, x_3, x_4 \in [-10; 10]$$

Sub-system 1

$$\begin{cases} \min_{x_c, x_1} f_1(x_c, x_1) = x_1 \times \frac{(x_1 - x_c + 0.1)^2}{(x_1 - y_2 + 0.1)^2} \\ \min_{x_c, x_1} f_2(x_c, x_2) = x_2 \times \frac{(x_2 - x_c + 0.1)^2}{(x_2 - y_2 + 0.1)^2} \\ y_1 = x_c \end{cases} \quad (17)$$

Sub-system 2

$$\begin{cases} \min_{x_c, x_3} f_3(x_c, x_3) = -x_3 \times \frac{(x_3 - x_c + 0.1)^2}{(x_3 - y_1 + 0.1)^2} \\ \min_{x_c, x_4} f_4(x_c, x_4) = -x_4 \times \frac{(x_4 - x_c + 0.1)^2}{(x_4 - y_1 + 0.1)^2} \\ y_2 = x_c \end{cases} \quad (18)$$

4.5 Problem #5

$$x_c, x_1, x_2, x_3, x_4 \in [-50; 50]$$

Sub-system 1

$$\begin{cases} \min_{x_c, x_1} f_1(x_c, x_1) = (x_1 - 1)^2 + (x_c - 3)^2 + (y_3 - 5)^2 \\ \min_{x_c, x_2} f_2(x_c, x_2) = (x_2 - 3)^2 + (x_c - 7)^2 + (y_4 - 7)^2 \\ y_1 = (y_3 + 2)^2 - 22 \\ y_2 = (y_4 + 2)^2 - 22 \end{cases} \quad (19)$$

Sub-system 2

$$\begin{cases} \min_{x_c, x_3} f_3(x_c, x_3) = (x_3 - 3)^2 + (x_c - 5)^2 + (y_1 - 3)^2 \\ \min_{x_c, x_4} f_4(x_c, x_4) = (x_4 - 5)^2 + (x_c - 9)^2 + (y_2 - 9)^2 \\ y_3 = (y_1 + 3)^2 - 33 \\ y_4 = (y_2 + 3)^2 - 33 \end{cases} \quad (20)$$

5 RESULTS

COSMOS has been run on each test problem with two sets of parameters: the first with a population of 20 individuals, 20 system iterations and 20 subsystem iterations (light); the second one with 100 individuals, 50 system and subsystem iterations (heavy).

Two criteria are verified for the validation of results. First, the convergence of the approximate coupled variable to its real value, and then the quality of the solutions obtained.

5.1 Convergence of Coupled Variables

During subsystem optimization, each discipline only knows the coupled variable computed in the other subsystem at the previous system iteration and uses it to find an approximation of its real value.

We only evaluate the evolution of the error of the approximation of the coupling functions along time (i.e., at each iteration) and not the evolution of coupling variables to their optimum value at the end of the optimization process. We

also verify if the y_i which is put in the array T_i is the one that would give the best approximation among all the y_i which have been computed during the sub-system optimization.

Table 1 presents the results computed on the problems #1, #2, #3 #4 and #5 with the *light* and *heavy* sets of parameters. The results are given in % of the error relative to the size of the domain of the coupled variables.

We observe that the error between the approximation and the real value is quite small except for y_{12} and y_{13} in example #2 as shown in Figures 4 and 5. The fact that a coupling function is used in two different disciplines at the same time seems to disturb its approximation. We also notice that the mean value of the error is close to the minimum value except for this coupling variable that seems to indicate that there are as many good values as bad ones. Two phenomena could explain these errors:

- The values of x_c^* do not cover its domain: the optimum x_c for disciplines 1, 2 and 3 are respectively in $\{-20\}$, $[-20,4]$ and $\{20\}$ so $x_c^* \in [-20,4] \in \{20\}$. For values of $x_c \in]4;20[$ the approximation will not be good.

Table 1. Minimum, mean and maximum value of the error of approximation (%): 1 minimum, mean and maximum value of the error of approximation (%)

opt.	test	var.	min	mean	max
Light	#1	y_1	0.0007	0.0157	0.1038
	#2	y_{1x}	0.0192	9.2963	38.2864
		y_{32}	0.0041	1.4345	21.3760
	#3	y_1	0.0047	0.1540	0.4441
		y_2	0.0047	0.2923	0.9366
		y_3	0.0051	0.1391	0.9548
		y_4	0.0972	0.4443	0.8602
	#4	y_1	0.0495	2.4839	15.0109
		y_2	0.0495	2.4839	15.0109
Heavy	#1	y_1	0.0001	0.0030	0.0129
	#2	y_{1x}	0.1293	15.5426	41.0447
		y_{32}	0.0002	0.1891	2.4605
	#3	y_1	0.0041	0.2539	1.3165
		y_2	0.0035	0.2215	1.0609
		y_3	0.0022	0.2020	0.8487
		y_4	0.0008	0.3336	1.3102
	#4	y_1	0.0054	0.4863	2.3182
		y_2	0.0054	0.4863	2.3182
	#5	y_1	failed	failed	failed
		y_2	failed	failed	failed
		y_3	failed	failed	failed
		y_4	failed	failed	failed

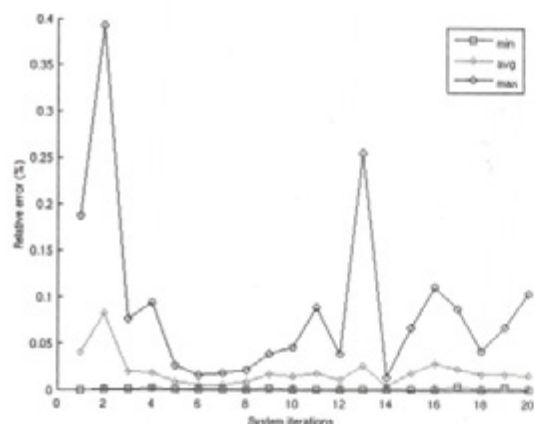


Fig. 3. Relative error of y on #1 (light)

- The optimum value of x_i is not unique but in $[-20;3]$. When this value is unique, the other discipline only has one choice for the corresponding x_c . Here the discipline increases its probability to pick an x_c corresponding to the x_i that does not give the optimum value of $y_i x$ for its own discipline.

There is no strict decrease of the error along time. But we notice a global decrease around the five first iterations which is stabilized then (Figs. 3 and 6).

The error of the approximation, in the general case, does not decrease but stays quite small along time. Moreover, the tests performed with the heavy set of parameters are less chaotic (e.g., the differences between Figures 4 and 5). We explain this by the spread of the values of v in the coupling table. Indeed, the larger the population is, the more chance we have to find a x_c in the table which is close to the value we need. This error should also tend toward zero if the population tends to a unique x_c .

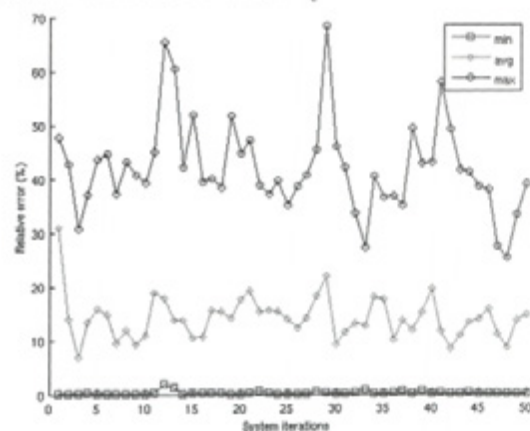


Fig. 5. Relative error of $y_{12} = y_{13}$ on #2 (heavy)

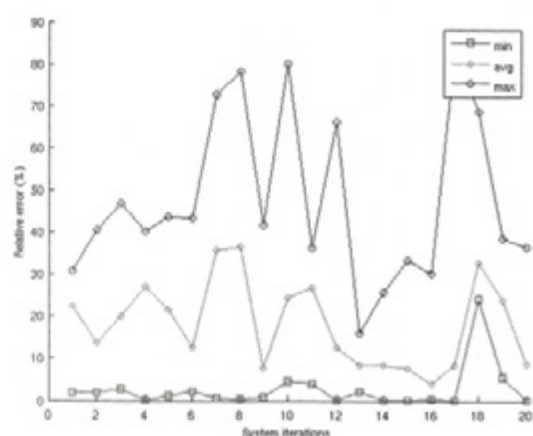


Fig. 4. Relative error of $y_{12} = y_{13}$ on #2 (light)

To summarize, on most cases we distinguish two phases in the evolution of the error of approximation of the coupled variables illustrated by Figure 7. The first phase is a global decrease of the error due to the convergence of the individuals (disciplinary variables). The second phase is a state where the error is stabilized and does not show noticeable changes. The evolution is chaotic and its amplitude decreases when we add more individuals in the population. The distance d decreases by increasing the population size.

This description seems to be relevant on examples where the objective functions are convex. Under this assumption, to each x_c , there is a single and unique x_i^* that minimize f_i . Thus, there is a function $s_i : X_c \rightarrow X_i \times U_i$ that allow to replace the

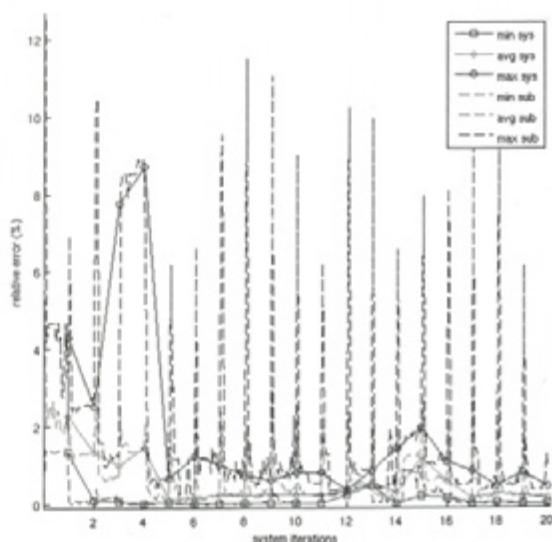


Fig. 6. Relative error of y_1 on #3 (light)

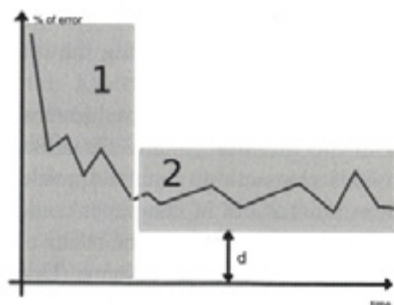


Fig. 7. Scheme of the evolution of the error along time

function $I_i = (X_c \times X_i \times U_i, Y_i, G_i)$ by a simplified one $t_i = (X_c, Y_i, G_i)$ with $G_i = \{(x_c, y_i) | (x_c, s(x_c)), y_i \in G_i\}$, where G_i is the graph of the function $f: G_i = \{(x, y) | y = f(x)\}$. We have $T_i t_i$, thus if the objective functions are convex, T_i is a good candidate for the approximation of I_i .

Moreover, - as shown on Figure 6 - the approximation at the system level is not always the best computed at subsystem level. This is true in two conditions:

- the optimum of the x_i are unique and the system did not yet converge to these solutions.
- the optimum of the x_i are not unique and as explained above, the x_i value picked by the first discipline does not correspond to the x_i that gives the best (x_c, x_i) that can lead to the optimum solution of the coupled variable for the other discipline.

Example #5 does not fit to the same category of problems tested. The coupling functions have been chosen as a system of equation that do not

converge with a Fixed Point Iteration algorithm. On this problem, COSMOS does not converge either. The error of approximation increases during time.

5.2 Quality of Solutions

The solutions found for problems #1 and #3 are optimum. They correspond to the solutions that are Pareto-optimum in the both disciplines at the same time. Thus, we are sure that they are on the global Pareto front. Indeed, for the first test problem, COSMOS globally converged to the solutions $x^* = (a, 1, 3, 3, 5)$ with $a \in [3; 7]$ which are optimum solutions of the problem. Results of the third test problem are satisfying too (Figs. 8 and 9). The thin lines at the bottom of the figures are the Pareto frontiers of the two sub-problems solved independently. The larger line corresponds to the Pareto front for fixed values of the other sub-problems at their optimum (i.e. $x_3 = 3$ and $x_4 = 5$ in the first discipline, and $x_1 = 1$ and $x_2 = 3$ for the second one). The solutions found by COSMOS are close to this second Pareto front.

On example #4 the coupled variable converged but COSMOS did not find the right solution. Indeed, the optimum solution is unique and $(f_1^*, f_2^*, f_3^*, f_4^*) = (-10, -10, -10, -10)$ but COSMOS converged to a unique point $(f_1, f_2, f_3, f_4) = (0, 0, 0, 0)$. We explain this behavior by the fact that the population is used at the same time to fit the objectives and to fit the approximation of the coupling functions. Here, this interference is

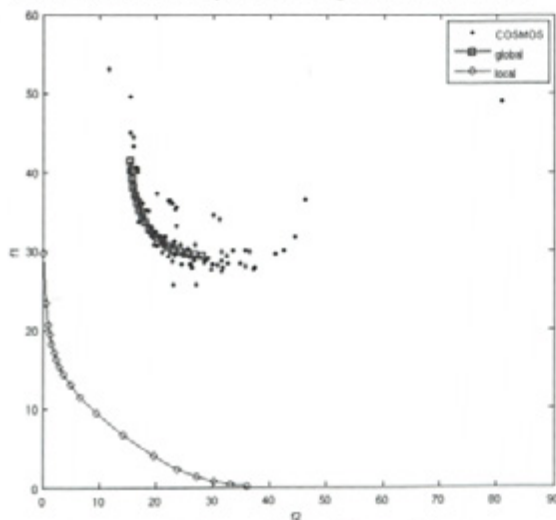


Fig. 8. Pareto front f_1 - f_2 on #3

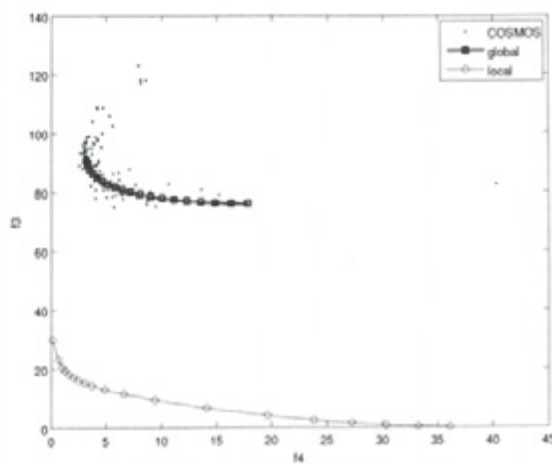


Fig. 9. Pareto front f_3 - f_4 on #3

explicit because the common variables are also the coupling variables.

If we call P , the set of solutions found by COSMOS, and S_i the set of theoretical solutions found for the global optimization problem, the P solution are all in the Pareto set of the global problem S_i (i.e., $P \subseteq S_i$) but they are not spread on the whole Pareto front (i.e., $P \cap S_i \neq 0$). This comes from the decomposition of a multi-objective problem in multiple multi-objective sub-problems. One way to obtain lost solutions is described in [10].

6 CONCLUSION

Many methods have been proposed to solve multidisciplinary problems. Mono-level ones (MDF, IDF, AAO), are global methods and thus are not adapted to the context of extended enterprise where each discipline has to solve its own optimization problem without direct access to the others variables. Multi-level methods give more autonomy to the disciplines but most of them are not specifically designed for multi-objective problems and need adaptations to propose more than one solution to the designer. COSMOS has been developed to fulfil these problems.

In this paper, we pointed out problems that can arise in the treatment of coupling variables in multi-objective multidisciplinary optimization. COSMOS, as a first step, has only be tested on simple example problems and it would be interesting to perform more tests and particularly real industrial problems.

Nevertheless, it appears that in this case and under the assumption of "to a given common variable x_c , it exists one and only one value x_i^* of the disciplinary variable x_i that minimizes the objective function f_i ", the array that uses COSMOS to approximate the coupling functions is a good approximation of the coupling functions. This assumption comes true if the objective function is convex. However, in general case our tests indicate that if this condition is not verified, the method is not capable of determining the correct value for approximation. Moreover, we showed that the values that are kept for approximation - the ones found at the end of subsystem optimization - are not always the values which give the best results. The system tends to solutions that are optimum for the objective functions but not enough to solutions that respect the coupling functions.

Taking into account these results, future research will focus on improving the method of approximation of coupling functions.

All the above methods have been intensively studied in the literature, many classification of methods exists but nothing about the problems. The difficulties pointed out in this paper could be use as a basis of a classification of multi-objective multidisciplinary design problems. This would permit to run tests on specific class in order to test the methods.

7 REFERENCES

- [1] Alexandrov, N.M., Lewis, R.M. *Comparative properties of collaborative optimization and other approaches to mdo*. Technical report, ICASE-99-24; NAS 1.26209354; NASA CR-1999-209354, 1999.
- [2] Allison, J.T., Kokkalaras, M., Papalambros, P.Y. On the use of analytical target cascading and collaborative optimization for complex system design. *6th World Conference on Structural and Multidisciplinary Optimization*, 2005.
- [3] Allison, T.J., Kokkalaras, M., Papalambros, Y.P. On selecting single-level formulations for complex system design optimization. *Journal of Mechanical Design*, 129(9):898-906, September 2007.
- [4] Aute, V., Azarm, S. A genetic algorithms based approach for multidisciplinary multiobjective collaborative optimization. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.
- [5] Balling, R.J., Sobieszczanski-Sobieski, J. Optimization of coupled systems: A critical overview of approaches. *AIAA Journal*, vol. 34(1):6-17, 1996.
- [6] Behdinan, K., Perez, R.E., Liu, H.T. Multidisciplinary design optimization of aerospace systems. *Proceedings of the 2nd International Design Conference CDEN*, 2005.
- [7] Braun, R., Cage, P., Kroo, I., Sobieski, I. Implementation and performance issues in collaborative optimization. *6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, vol. 1, p. 295-305, 1996.
- [8] Cramer, E.J., Dennis, J.E., Frank, P.D., Lewis, R.M., Shubin, G.R. Problem formulation for

- multidisciplinary optimization. *SIAM Journal of Optimization*, 4:754-776, 1994.
- [9] Dennis, J.E., Arroyo, S.F., Cramer, E.J., Frank P.D. Problem formulations for systems of systems. *IEEE International Conference on Systems, Man and Cybernetics*, 2005.
- [10] Engau, A., Wiecek, M.M. 2D decision-making for multicriteria design optimization. *Structural and Multidisciplinary Optimization*, 34(4):301-315, October 2006.
- [11] Giassi, A., Bennis, F., Maisonneuve, J.-J. Multidisciplinary design optimization and robust design approaches applied to concurrent design. *Structural and Multidisciplinary Optimization*, 27:1-16, 2004.
- [12] Giesing, P.J., Barthelemy, M.J.-F. A summary of industry mdo applications and needs. *7-th AIAA/USAF/NASA/ISSMO*, 1998.
- [13] Gunawan, S., Farhang-Mehr, A., Azarm, S. Multi-level multi-objective genetic algorithm using entropy to preserve diversity. *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, vol. 2632, p. 148-161, Faro, Portugal, April 8-11 2003.
- [14] Hulme, K.F., Bloebaum, C.L. A simulation-based comparison of multidisciplinary design optimization solution strategies using cascade. *Structural and Multidisciplinary Optimization*, 19(1):17-35, 2000.
- [15] Sobieszczanski-Sobieski, J., Agte, J., Sandusky Jr., R. *Bi-level integrated system synthesis (bliss)*. Technical report, NASA/TM-1998-208715, 1998.
- [16] Kodiyalam, S. *Evaluation of methods for multidisciplinary design optimization (m.d.o.), phase I*. Technical report, CR-1998-20871-NASA, 1998.
- [17] McAllister, C.D., Simpson, T.W., Lewis, K., Messac, A. Robust multiobjective optimization through collaborative optimization and linear programming. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.
- [18] MDO-Technical-Committee. *Current state of the art in multidisciplinary design optimization*. Technical report, AIAA Techreport, 1991.
- [19] Rabeau, S., Dépincé, Ph., Bennis, F., Janiaut, R. Comparison of global and local treatment for coupling variables into multidisciplinary problems. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.
- [20] Wujek, B., Renaud, J., Batill, S. A concurrent engineering approach for multidisciplinary design in a distributed computing environment. *Multidisciplinary Design Optimization: State of the Art*. SIAM, 1997.