Struct Multidisc Optim (2012) 45:619–642
DOI 10.1007/s00158-011-0701-4

REVIEW ARTICLE

A survey of multidisciplinary design optimization methods in launch vehicle design

Mathieu Balesdent · Nicolas Bérend · Philippe Dépincé · Abdelhamid Chriette

Received: 11 March 2010 / Revised: 21 July 2011 / Accepted: 29 July 2011 / Published online: 27 September 2011

Abstract Optimal design of launch vehicles is a complex problem which requires the use of specific techniques called Multidisciplinary Design Optimization (MDO) methods. MDO methodologies are applied in various domains and are an interesting strategy to solve such an optimization problem. This paper surveys the different MDO methods and their applications to launch vehicle design. The paper is focused on the analysis of the launch vehicle design problem and brings out the advantages and the drawbacks of the main MDO methods in this specific problem. Some characteristics such as the robustness, the calculation costs, the flexibility, the convergence speed or the implementation difficulty are considered in order to determine the methods which are the most appropriate in the launch vehicle design framework. From this analysis, several ways of improvement of the MDO methods are proposed to take into account the specificities of the launch vehicle design problem in order to improve the efficiency of the optimization process.

Keywords Multidisciplinary design optimization · Launch vehicle design · MDO · Multi-objective optimization · Distributed optimization · Space transport system design · Multi-criteria optimization · Optimal control

The work presented in this paper is part of a CNES/ONERA PhD thesis attached to CNES's HADES project (Help on Advanced launchers DESign).

M. Balesdent (✉) · N. Bérend
Onera – The French Aerospace Lab, Palaiseau, France
e-mail: mathieu.balesdent@onera.fr

N. Bérend
e-mail: nicolas.berend@onera.fr

M. Balesdent
CNES, Launchers Directorate, Evry, France

M. Balesdent · P. Dépincé · A. Chriette
Ecole Centrale de Nantes, IRCCyN, Nantes, France

P. Dépincé
e-mail: philippe.depince@irccyn.ec-nantes.fr

A. Chriette
e-mail: abdelhamid.chriette@irccyn.ec-nantes.fr

Nomenclature

AAO	All At Once
ATC	Analytical Target Cascading
BLISS	Bi-Level System Synthesis
CO	Collaborative Optimization
CSSO	Concurrent SubSpace Optimization
DIVE	Discipline Interaction Variable Elimination
DyLeaf	Dynamic Leader Follower
ELV	Expendable Launch Vehicle
FPI	Fixed Point Iteration
GA	Genetic Algorithm
GAGGS	Genetic Algorithm Guided Gradient Search
GLOW	Gross Lift-Off Weight
GSE	Global Sensitivity Equation
IDF	Individual Discipline Feasible
LDC	Local Distributed Criteria
MCO	Modified Collaborative Optimization
MDA	Muldisciplinary Design Analysis
MDF	Multi Discipline Feasible
MDO	Multidisciplinary Design Optimization
MOPCSSO	Multi-Objective Pareto CSSO
MSTO	Multi-Stage To Orbit

NAND	Nested Analysis and Design
OBD	Optimization Based Decomposition
RLV	Reusable Launch Vehicle
RSM	Response Surface Method
SAND	Simultaneous Analysis and Design
SQP	Sequential Quadratic Programming
SNN	Single NAND NAND
SSA	System Sensitivity Analysis
SSN	Single SAND NAND
SSS	Single SAND SAND
SSTO	Single Stage To Orbit

1 Introduction

Launch vehicle design is a complex process in which the search of the best performance at the less cost is essential. This process takes account of many disciplines which have to be well handled in order to ensure the optimality of the launch vehicle designed. These disciplines, which often have conflicting objectives, require adapted design tools which allow to integrate the constraints inherent to each discipline and to facilitate the compromise search.

The traditional way to solve the launch vehicle design problem is a sequential method (often a Fixed Point Iteration method (Vanderplaats 1999)) in which each discipline solves its own optimization problem and aims to be consistent with the other ones. This method is built on the designer expertise and does not necessarily ensure the global optimality.

The Multidisciplinary Design Optimization (MDO) is a set of engineering systems design methods which handle several disciplines. The MDO methods aim to take advantage of the couplings and the synergisms between the different disciplines in order to reach the global optimal design (Sobieski 1993). The main objectives of the MDO process are the quality of the found solution, the computation time and the robustness of the optimization process (i.e. the ability to converge to an optimum from large initialization domain).

The specificity (and also the difficulty) of launch vehicle design is to include, within the global design, a trajectory optimization by using an optimal control law calculation subject to equality constraints (specifications of the mission). The launch vehicle design includes the optimization of not only design variables but also a control law. This task is responsible for the performance estimation and presents a relatively important difficulty in the optimization process.

In literature, we can find many MDO methods which have been applied in a great number of examples. Because the study cases are different, it is difficult to compare these methods in order to determine which is the best. Some review articles (Agte et al. 2009; Alexandrov and Hussaini 1995; Balling and Sobieszczanski-Sobieski 1994; Sobieszczanski-Sobieski and Haftka 1997; Tosserams et al. 2009) provide a state-of-the-art of the different MDO methods. The aim of this paper is not to make an exhaustive list of the different MDO methods but at first to describe the main MDO methods with common standardized notations, and then to evaluate the applicability of the expressed methods in launch vehicle design and to bring out the advantages and drawbacks of these methods with regard to this specific problem. On the grounds that the discussion presented in this paper is quite general, a great number of considerations given in the paper may be applicable in other application fields than space vehicles design.

In order to answer these problematics, the paper is organized as follows: Section 2 will describe the classical MDO methods applicable to the launch vehicle design. The advantages and drawbacks of these methods will be mentioned and the trajectory optimization problem will be explained. Finally, different optimization algorithms such as metaheuristics and gradient-based methods will be described in order to present their applicability in launch vehicle design.

Section 3 aims to explain some examples of application of MDO methods in launch vehicle design. In this part, we will analyze different study cases of launch vehicle design problem and the optimization of a Single-Stage-To-Orbit (SSTO) vehicle will be described to compare the different MDO methods. Finally, a comparative synthesis of the different methods will be presented.

The last section will present some concluding remarks and ways of improvement about the application of MDO methods in launch vehicle design.

2 Generalities about multidisciplinary design optimization

2.1 General description of a MDO problem

In this section, we define the fundamental notions of a classical MDO process. Launch vehicle design problem can be decomposed in different ways (e.g. into the disciplines, the flight phases, the launch vehicle stages...). These decompositions imply different optimization architectures. Indeed, the different disciplines can be considered separately or can be optimized simultaneously inside a same system. Hereinafter, we will use, with the same signification, both terms *discipline* and *subsystem* in order to describe the different process at the subsystem level (Fig. 1).

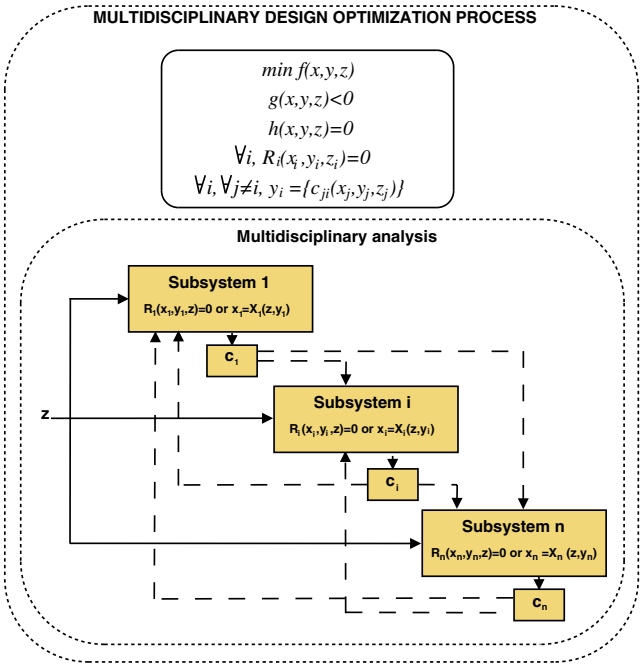

Fig. 1 Nomenclature used

2.1.1 Mathematical formulation

The general problematic of a MDO study can be formulated as follows:

$$\begin{aligned} &\text{minimize} && f(x, y, z) \\ &\text{with respect to} && z = \{z_{sh}, \bar{z}_k\} \\ & && g(x, y, z) \leq 0 && (1) \\ &\text{subject to} && h(x, y, z) = 0 && (2) \\ & && \forall i, \forall j \neq i,\ y_i = \{c_{ji}(x_j, y_j, z_j)\}_j && (3) \\ & && \forall i,\ R_i(x_i, y_i, z_i) = 0 && (4) \end{aligned}$$

2.1.2 Types of variables

We can differentiate three categories of variables in a general MDO problem:

- z: design variables. These variables evolve all along the optimization process in order to find the optimal design. They can be used in one or several subsystems $z = \{z_{sh}, \bar{z}_k\}$. The subscript $_{sh}$ symbolizes the variables which are shared between different subsystems (global variables) and $\bar{z}_k$ denotes the variables which are specific to one subsystem (local variables). Moreover, we use the notation z_k to describe the variables which refer to the kth subsystem.
- x: state (or disciplinary) variables. These variables (which are not design variables) can vary during the disciplinary analysis in order to find an equilibrium in the state equations (disciplinary equations). Unlike z, the state variables are not independent degrees of freedom but depend on the design variables z and the state equations. The cases in which x are given by explicit functions of z are uncommon in engineering applications. Indeed, these variables are most often defined by implicit functions, that generally requires specific optimization methods for solving complex industrial problems.
- y: coupling variables. These variables are used to link the different subsystems and to evaluate the consistency of the design with regard to the coupling.

2.1.3 Types of constraints

The constraints can be divided in two categories:

- g: inequality constraints,
- h: equality constraints.

2.1.4 Types of functions

The different functions used in a MDO problem are:

- $f(x, y, z)$: objective function,
- $c(x, y, z)$: coupling functions. These functions calculate the coupling variables which come out of a subsystem. We will note $c_{ij}(x_i, y_i, z_i)$ the coupling variables from the subsystem i to the subsystem j.
- $R_i(x_i, y_i, z_i)$: residual functions. The residuals quantify the satisfaction of the state equations.
- $X_i(y_i, z_i)$: state variables computation functions. These functions yield the roots x_i of the (4).

2.1.5 Disciplinary equations

The disciplinary (subsystem) equations can be formulated in two ways:

- Residual form (implicit):

$$R_i(x_i, y_i, z_i) = 0 \qquad (4b)$$

- Non-residual form (explicit):

$$x_i = X_i(y_i, z_i) \qquad (5)$$

We can distinguish two methods for handling the disciplinary equations:

Disciplinary analysis The disciplinary analysis consists in finding the state variables x_i which satisfy the state equations (4b). Generally, when the disciplinary equations are expressed in the residual form, an iterative loop (e.g. a Newton algorithm) is used in order to find the root of the equations (4b).

When the non residual-form is used, the iterative loop is not required in the subsystem because the state variables are directly expressed from the design and coupling variables by explicit relationships. In this case, the calculation scheme consists in sequentially evaluating the different relationships in order to compute the disciplinary outputs.

Disciplinary evaluation The disciplinary evaluation consists in calculating the value of the residual $R_i(x_i, y_i, z_i)$ from x_i, y_i and z_i. In this scheme, the equations (4b) are not solved. Furthermore, the state variables x are not handled in the subsystem but are considered as inputs in the same way as z and y (Fig. 2).

2.1.6 Coupling

From the parameters y_i and z_i coming in the subsystem i and the state variables x_i, the coupling variables which come out of the subsystem can be calculated in using the coupling function $c_i(x_i, y_i, z_i)$. The double indexation $c_{ij}(x_i, y_i, z_i)$ denotes that these coupling variables are transmitted to the jth subsystem. The coupling is consistent when the set of coupling variables y_i is equal to the one which contains the values returned by the different coupling functions (3).

2.1.7 Multidisciplinary analysis

The multidisciplinary analysis (MDA) is a process which aims to satisfy the coupling between the different subsystems. The MDA consists in finding for all the subsystems, the variables x_i such as the residuals are equal to zero (state equations are satisfied) and the coupling is consistent. In other words, the MDA consists in satisfying the system formed by the equations (3) and (4).

The MDA is developed in details in Keane and Nair (2005).

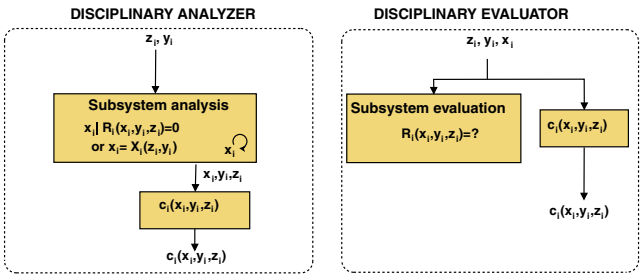

Fig. 2 Disciplinary analyzer and disciplinary evaluator

2.1.8 Some feasibility concepts in MDO

Individual disciplinary feasibility A problem is qualified as "individual disciplinary feasible" if at each iteration of the optimization process, the state equations are satisfied individually. In other words, we can always find the values of the state variables x which satisfy the equations (4) (but the consistency of the coupling is not guaranteed). Trivially, a process in which all the disciplinary equations are in the non-residual form (5) is intrinsically "individual disciplinary feasible".

Multi disciplinary feasibility If at each iteration of the optimization process, we can find the state variables x_i and the coupling variables y_i such as the "individual disciplinary feasibility" is realized and the coupling is consistent, the problem is qualified as "multi disciplinary feasible". A problem in which a MDA is performed is a "multi disciplinary feasible" problem. In this kind of problems, we can express the values of the state variables exclusively in function of the design variables: $x = X_c(z)$ (the subscript 'c' symbolizes the consistency of the coupling).

2.1.9 Specifications of the launch vehicle design problem

The launch vehicle design problem is generally decomposed into the different physical disciplines. These disciplines are commonly aerodynamics, propulsion, structure, costs, weights and sizing, and trajectory optimization. Classical objective functions (f) are the maximization of the payload mass, the minimization of the Gross Lift-Off Weight or the minimization of the launch vehicle cost. The design variables (z), such as the masses, the diameters, the propulsion variables (chamber pressures, mixture ratio...), the fairing shape, etc., are generally considered at the system-level. The trajectory variables (parameters of the control law if the trajectory optimization is computed by a direct method or adjoint vector if the optimization uses the Maximum of Pontryagin Principle) are usually considered as state variables (x). The optimality conditions associated with these variables can be satisfied at the subsystem level ($R = 0$) or at the system level ($R = 0$ are added to the system-level equality constraints). In this particular case, the control law variables are at the same level as the design variables z. Typical coupling variables (y) may be the dry mass, the specific impulse, the length to diameter ratio of the different stages, etc. Equality constraints (h) may be composed of the specifications of the mission (desired orbit, payload mass,

GLOW...) and inequality ones (g) may include for instance the maximum chamber pressure, the maximum load factor or the minimum nozzle exit pressure.

2.2 Classical MDO formulations

2.2.1 Single-level methods

Multi Discipline Feasible The Multi Discipline Feasible (MDF) method is the most usual method. It is also called "Nested Analysis and Design" (NAND), "Single NAND-NAND" (SNN) and "All-in-One". This method is explained in Allison (2004), Balling and Sobieszczanski-Sobieski (1994), Cramer et al. (1994), Gang et al. (2005) and Kodiyalam (1998). The architecture of the MDF method is similar to the one of a classical optimization problem which contains only one system. The main difference is that in MDF, the discipline of the classical optimization problem is replaced by a complete multidisciplinary analysis. This MDA is performed at each iteration. Thus, all the subsystems are coupled in an analysis module which checks the feasibility of the solution at each iteration (Fig. 3). In this formulation, the vector composed of the design variables is transmitted to the analysis module. This module executes, by a Fixed Point Iteration (FPI) or an optimization method, the multidisciplinary analysis of the system (3) and (4).

Once the MDA is performed, the analysis module output vector is used by the system-level optimizer to compute the objective function and the constraints. The procedure is repeated at each iteration. In this method, each set of found design variables is a consistent configuration. Furthermore, the disciplines are in charge of finding their local variables x to satisfy their own (4) (disciplinary analyzers are used).

In the MDF method, each found solution is a feasible one (i.e. individual disciplinary equations (4) as well as couplings equations (3) are satisfied). We can note that in this section, the term "feasible" does not imply the satisfaction of design constraints g and h but just the MDO ones.

Some systems do not have any feedback between the different subsystems (e.g. Duranté et al. 2004). In this case, a sequential analysis, without iteration, is possible and the MDF method is the most natural method to solve this kind of problems.

For large scale (industrial) application cases, the different subsystems can be composed of groups of specialists (each of them potentially located in a different place all over the world). In this case, the MDA becomes a very complex task and includes:

- transmission of information between the different groups of specialists (dotted lines in Fig. 3), and not only between computers or optimization programs,
- management requirements between and inside the different groups,
- definition of each group action domain and autonomy with respect to the community.

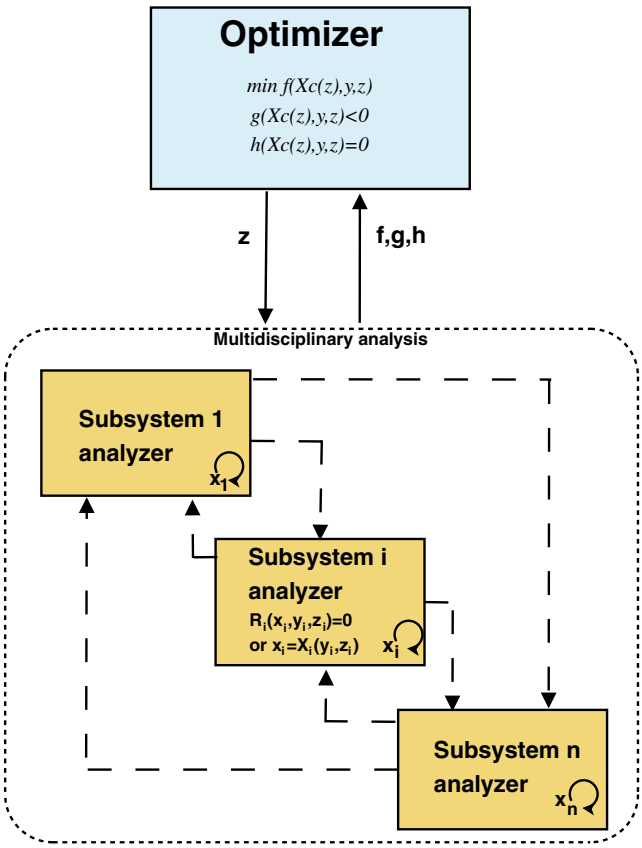

Fig. 3 MDF method

The optimization problem can be formulated as follows

$$\begin{aligned} &\text{minimize} && f(X_c(z), y, z) \\ &\text{with respect to} && z \\ &\text{subject to} && g(X_c(z), y, z) \leq 0 \quad (6) \\ & && h(X_c(z), y, z) = 0 \quad (7) \end{aligned}$$

The state variables do not intervene in the optimization problem formulation because they are totally handled by the MDA at the subsystem-level.

Advantages and drawbacks of MDF method The main advantage of the MDF method is its simplicity. Indeed, a limited number of design variables is used. The method implementation is relatively easy since the system decomposition is not required.

Moreover, if the optimization process is stopped, the found solution is consistent, even if it is not the optimal one.

The MDF method has important drawbacks. Indeed, the calculation cost is very important and the method does not take advantages of the coupling between the disciplines to improve the optimization process. The calculation cost is due to the MDA which has to be executed at each iteration. This analysis considers all the subsystems present in the optimization process (the MDF method modularity is very poor). Thus, when a parameter changes, all the other variables have to be recalculated. Furthermore, a gradient-based algorithm used as the optimizer will need a full MDA whenever the derivatives required to compute the gradient or the Hessian have to be calculated (Kodiyalam 1998).

Consequently, MDF is only applicable to the optimization problems in which the different subsystems can be quickly evaluated during the MDA or in which the MDA converges in a few iterations.

For large scale application problems (in which the subsystems are engineering teams), at each iteration, the MDA requires a lot a information transmissions and management tasks to define the autonomy and the action domain of each specialty group (possible conflict resolutions with engineers who want to decide design and to have as much autonomy as possible). Moreover, each group of specialists has to wait the previous one in order to perform its task (when FPI is used), which can be all the more time consuming as the different teams are dispersed all over the world.

Individual Discipline Feasible The Individual Discipline Feasible (IDF) formulation (Cramer et al. 1994; Martins and Marriage 2007), also called "Optimizer-Based-Decomposition (OBD)" (Kroo 2004), "Single-SAND-NAND" (SSN) (Balling and Sobieszczanski-Sobieski 1994), allows to avoid a complete multidisciplinary analysis at each iteration of the design process. Like MDF, a single optimization at the system-level is used and analysis blocks are called in the different subsystems. The main difference between MDF and IDF is that in IDF, the optimizer is also responsible for the coordination between the different subsystems and uses additional variables (coupling variables y) to ensure it. At each iteration, the different subsystems are individually feasible but the coupling between them is not guaranteed.

This method allows to break up the main problem into several subsystems. The coupling variables (and the associate coupling equations $y_i - c_{ji} = 0$) are introduced to preserve the consistency of the results found by the different analysis boxes. In IDF, the consistency of the solution (multi disciplinary feasibility (3)) is not guaranteed at each iteration but only at the convergence. Consequently, the IDF process should not be stopped before the convergence is reached.

This decomposition method considerably increases the number of variables but allows to improve the optimization efficiency (parallelization is possible). Therefore, unlike the MDF method, a single analysis is performed at each iteration in the different subsystems. The centralization degree of the IDF method is more important than the MDF one. There again, the subsystems determine their state variables (subsystem analyzers are used).

The problem may be summarized as follows:

$$\begin{aligned} &\text{minimize} && f(X(y,z),y,z) \\ &\text{with respect to} && y,z \\ & && g(X(y,z),y,z) \le 0 && (8) \\ &\text{subject to} && h(X(y,z),y,z) = 0 && (9) \\ & && \forall i, j \neq i, y_i = \{c_{ji}(X_j(y_j,z_j,),y_j,z_j)\}_j && (10) \end{aligned}$$

Advantages and drawbacks of IDF method If the number of coupling variables is relatively small, the IDF method is applicable and provides good results. Since the principle of this method is to add coupling variables and constraints, the number of iterations at the system-level is more important than in MDF. Indeed, the optimizer has to dialog with each disciplinary block and transmits to each of them its own coupling variables and constraints, in order to handle the optimization process.

In return, the IDF method presents the advantage of being implementable in a network. This particularity can considerably improve the efficiency of the method (calculation time is reduced). Furthermore, since the MDA process is removed, the internal analysis loop is broken up. In this way, if the system-level optimizer requires sensitivities calculations of the ith subsystem, the other subsystems do not intervene and no computation time is wasted . Therefore, if an important centralization of the optimization process

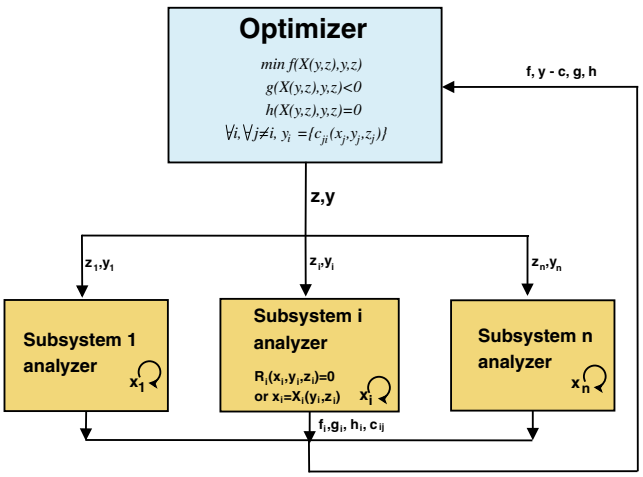

Fig. 4 IDF method

is permitted, the IDF method can be employed to solve effectively a MDO problem (Fig. 4).

For large scale applications (where subsystems are composed of engineers), the management tasks are less important than in MDF because each group just converses with the coordinator one. Indeed, since the MDA is broken, the different engineering teams have not to wait the results of other ones, that allows to parallelize the human tasks. However, the autonomy of the different teams is limited because the multi disciplinary feasibility is not ensured at subsystem-level but at only at the convergence of the system-level.

All At Once The All-At-Once method (AAO), also called "Single-SAND-SAND" (SSS) (Balling and Sobieszczanski-Sobieski 1994; Balling and Wilkinson 1997) solves simultaneously the optimization problem and the equations of the different subsystems (Allison 2004). These equations (13) and (14) are not satisfied at each iteration but they have to be at the convergence (Cramer et al. 1994) (design configuration is consistent only at the convergence). In this method, the subsystem equations (residuals) are regarded as equality constraints $R = 0$.

AAO is the most elementary MDO method. The control of the process is assigned to a system-level optimizer which aims to optimize a global objective and calls subsystem evaluations. The optimizer handles the design variables z, the coupling variables y and the state variables x. At the subsystem-level, the disciplinary analyzers are replaced by disciplinary evaluators. The design, the evaluations at the subsystem and system levels are performed at the same time. Therefore, the centralization of the problem is more important than in IDF and MDF.

The global state vector is divided in subvectors distributed to each subsystem and the residuals R are transmitted to the global optimizer simultaneously with the other variables. Since the residuals are equal to zero only at the optimum, the multidisciplinary and the individual disciplinary feasibilities are not ensured at the intermediary points.

The optimization problem can be summarized as follows:

$$\begin{aligned} \text{minimize} \quad & f(x, y, z) \\ \text{with respect to} \quad & x, y, z \\ & g(x, y, z) \leq 0 && (11) \\ \text{subject to} \quad & h(x, y, z) = 0 && (12) \\ & \forall i, R_i(x_i, y_i, z_i) = 0 && (13) \\ & \forall i, \forall j \neq i, y_i = \{c_{ji}(x_j, y_j, z_j)\}_j && (14) \end{aligned}$$

Advantages and drawbacks of AAO method The AAO method is the easiest method to solve MDO problems, but it presents drawbacks which make it inapplicable to large scale launch vehicle design problems. Indeed, in the case of problems which consider several complex subsystems, the number of variables handled by the optimizer soars and the AAO is not applicable (Fig. 5).

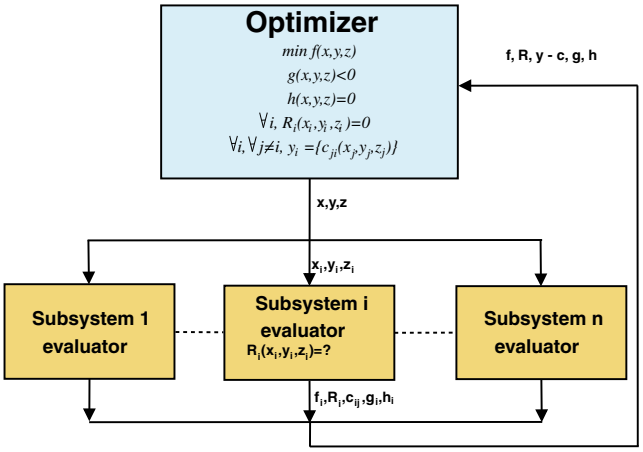

Fig. 5 AAO method

In the situations for which the convergence is not achieved, the AAO method (like IDF) does not provide a valid configuration. Moreover, AAO is difficult to implement because highly centralized. This method is not robust because the formulation results in a very constrained MDO problem.

In relatively small problems, the AAO method, like the IDF one, is applicable and allows to parallelize calculations. Thus, computation time may be considerably reduced.

For large scale applications, the different groups of engineers do not perform the individual disciplinary analysis of the different subsystems but are just reduced to make simple calculations with respect to the instructions given by the system-level. For this reason, the AAO method is just applicable to small problems where all the calculations can be performed by a small group of generalist engineers.

2.2.2 *Qualitative comparison of single-level methods*

Cramer et al. (1994) have performed a qualitative comparison of the MDF, IDF and AAO methods. Table 1 resumes this comparison and adds some considerations concerning the large scale applications.

2.2.3 *Multi-level methods*

Collaborative Optimization Collaborative Optimization (CO) method, initially developed by Braun and Kroo (1995), is a two-level optimization method. This method has been created to give more autonomy to the different subsystems in order to satisfy the compatibility constraints.

Table 1 Qualitative synthesis of the single level methods

	AAO	IDF	MDF
Use of traditional analysis tools	No	Yes	Yes, coupling is required
Ind. discip. feas. of the solution	Only at convergence	At each iteration	At each iteration
Multi discip. feas. of the solution	Only at convergence	Only at convergence	At each iteration
Decision variables of the optimizer	$\{z, x, y\}$	$\{z, y\}$	$\{z\}$
Convergence speed expected	Fast	Medium	Slow
Large scale applications			
Decomposition into concurrent data process	Yes	Yes	No
Decomposition into concurrent human tasks	Yes, but not judicious	Yes	No
Autonomy of engineering teams	None	Small	Moderate
	No decision	No decision	No decision
	No analysis	Ind. discip. analysis	Multi discip. analysis

The optimization problem is subdivided into disciplinary subproblems. In each subproblem, the local optimizer:

- controls its local design variables,
- is responsible for the satisfaction of its local constraints,
- does not have knowledge of the variables and the constraints of the other subsystems.

Each disciplinary optimizer modifies its variables in order to find an agreement with the other subsystems about the coupling variables. At the system-level, an optimizer is in charge of the coordination of the whole process and optimizes a global objective function. The main idea of this method is that the disciplinary experts can intervene in the subsystems without being constrained by the other subsystems.

The optimization problem may be formulated as follows:

First formulation (Braun and Kroo 1995)

At the system-level:

$$\begin{aligned} &\text{minimize} && f(y, z) \\ &\text{with respect to} && y, z \\ &\text{subject to} && \forall i,\ J_i^*(z^*, z, y_i, c_i(y_i, z_i^*)) = 0 \end{aligned} \tag{15}$$

where J_i^* is the optimized objective function of the ith subsystem and z^* the local copies of z.

For the ith subsystem, we have the following subproblem:

$$\begin{aligned} &\text{minimize} && J_i(z_i, z_i^*, y_i, c_i(y_i, z_i^*)) = \|z_i^* - z_i\|_2^2 \\ &&& \quad + \|y_{ij} - c_{ij}(y_i, z_i^*)\|_2^2 \\ &\text{with respect to} && z_i^* \\ &\text{subject to} && g_i(y_i, z_i^*) \leq 0 \qquad (16) \\ &&& h_i(y_i, z_i^*) = 0 \qquad (17) \end{aligned}$$

where y_{ij} are the coupling variables from the ith subsystem to the jth one.

Second formulation:

The second formulation (Alexandrov and Lewis 2000) consists in replacing the system-level constraint vector $J_i^* = 0$ by the following ones:

$$\forall i,\ J_{i1}^*(z_i, z_i^*) = z_i^* - z_i = 0 \tag{18}$$

$$\forall i, \forall j \neq i,\ J_{i2j}^*(y_{ji}, c_{ji}(y_i, z_i^*)) = y_{ji} - c_{ji}(y_i, z_i^*) = 0 \tag{19}$$

The subsystems can use either subsystem analyzers or subsystem evaluators to compute their local equations. In the second case, the equations of the subsystems become:

$$\begin{aligned} &\text{minimize} && J_i(z_i, z_i^*, y_i, c_i(y_i, z_i^*)) = \|z_i^* - z_i\|_2^2 \\ &&& \quad + \|y_{ij} - c_{ij}(y_i, z_i^*)\|_2^2 \\ &\text{with respect to} && z_i^* \\ &\text{subject to} && g_i(y_i, z_i^*) \leq 0 \qquad (20) \\ &&& h_i(y_i, z_i^*) = 0 \qquad (21) \end{aligned}$$

Advantages and drawbacks of CO method The collaborative optimization presents significant advantages with regard to the single-level optimization methods. Indeed, it is not necessary to modify the disciplinary codes in order to integrate a discipline into the CO optimization scheme (the modularity of the method is improved). Furthermore, CO allows to use optimization methods that are the most adapted to each subproblem, with possible actions of disciplinary experts. This method also allows to add or to modify some subsystems without changing the whole design process (Fig. 6).

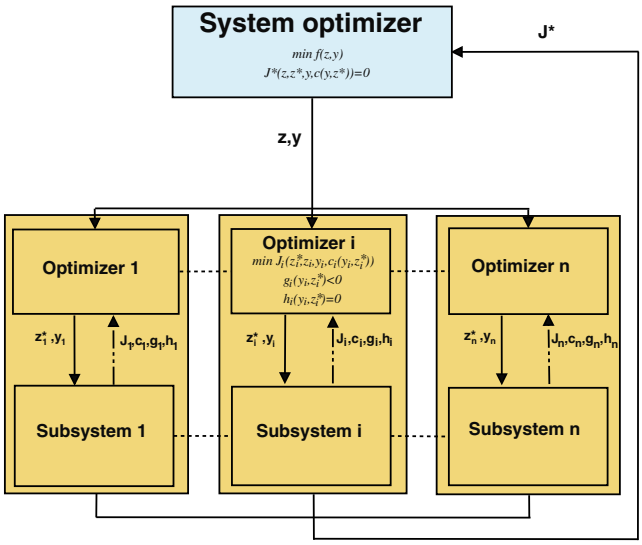

Fig. 6 CO method

Finally, CO is very flexible and the subsystems calculation efficiency can be improved by using response surfaces in addition to the local optimizers (Sobieski and Kroo 2000).

Unfortunately, CO seems not to be robust because it generates instabilities at convergence. Moreover, as the number of coupling variables increases, the CO method efficiency decreases. In other words, if the original MDO problem has a great number of coupling variables, CO formulation can be inefficient.

Concurrent SubSpace Optimization The Concurrent SubSpace Optimization (CSSO) method was formulated by Sobieszczanski-Sobieski (1988). This iterative method is also based on a system decomposition strategy which allows the subsystems to contribute independently to the optimization process. The global problem is solved by a system-level optimizer which ensures the coordination of the different subsystems and aims to find a compromise between the different solutions proposed at the subsystem level.

Approximations of the coupling variables are used in the different subsystems in order to determine their influence on the objective f and the constraints g and h. In this method, when performing the subsystems optimizations, the effects of a variable variation in one subsystem to the constraints of the other subsystems can be determined. This method introduces a concept of responsibility for constraint violation and uses of cumulative constraints (Sobieszczanski-Sobieski 1988), which consists in considering the partial satisfaction of a constraint in one discipline by the influences of the other disciplines. This concept is made possible by adding supplementary variables (coordinator ones) to the initial problem.

The approximations of the coupling variables can be performed by using neural networks (Sellar and Batill 1996), response surfaces (Renaud and Gabriele 1993, 1994; Sellar et al. 1996; Wujek et al. 1996, 1997) or other metamodels. Other expansions of the CSSO method using approximate models (Rodriguez et al. 1998, 2001; Perez et al. 2002) can be found in literature but are not developed in this paper. The CSSO method uses a multidisciplinary analysis to solve the optimization problem and often performs sensitivities analysis by using the Global Sensitivity Equation (GSE). The resolution of the GSE (Sobieszczanski-Sobieski 1990) allows to quickly obtain the total influence of the differents variables to the objective function.

The system-level optimizer solves the following problem:

$$\begin{aligned} \text{minimize} \quad & f(\tilde{y}, z) \\ \text{with respect to} \quad & z \\ \text{subject to} \quad & g(\tilde{y}, z) \leq 0 \qquad (22) \\ & h(\tilde{y}, z) = 0 \qquad (23) \end{aligned}$$

where $\tilde{y}$ represent the approximations of the coupling variables.

The optimization problem of the ith discipline can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f(\tilde{y}_{ji}, z_{sh}, \bar{z}_i), \ j \neq i \\ \text{with respect to} \quad & \bar{z}_i \\ \text{subject to} \quad & g_i(\tilde{y}_{ji}, z_{sh}, \bar{z}_i) \leq 0 \qquad (24) \\ & h_i(\tilde{y}_{ji}, z_{sh}, \bar{z}_i) = 0 \qquad (25) \end{aligned}$$

Unlike the CO method, the shared design variables are considered as constants during the concurrent optimizations at the subsystem-level. Huang and Bloebaum (2004) present a CSSO-based method: the Multi-Objective Pareto Concurent SubSpace Optimization (MOPCSSO). MOPCSSO has been developed to solve multi-objective problem with a CSSO architecture and integrates the concept of Pareto Optimality (Pareto 1971). This method allows to solve multi-objective large scale problems with a CSSO-based method.

Advantages and drawbacks of CSSO method The main characteristic of the CSS method is the use of approximate disciplinary models to evaluate the variables of the other disciplines and to solve the problem as a decoupled one. These approximate linearized models create a database which is used by the local optimizers in order to optimize the objectives and to satisfy the constraints. In that way, CSSO can reduce the calculation time of the optimization process. In brief, if the system is relatively small and the models approximations are easy to formulate, CSSO can be very efficient and gives solutions in a minimum time (Fig. 7).

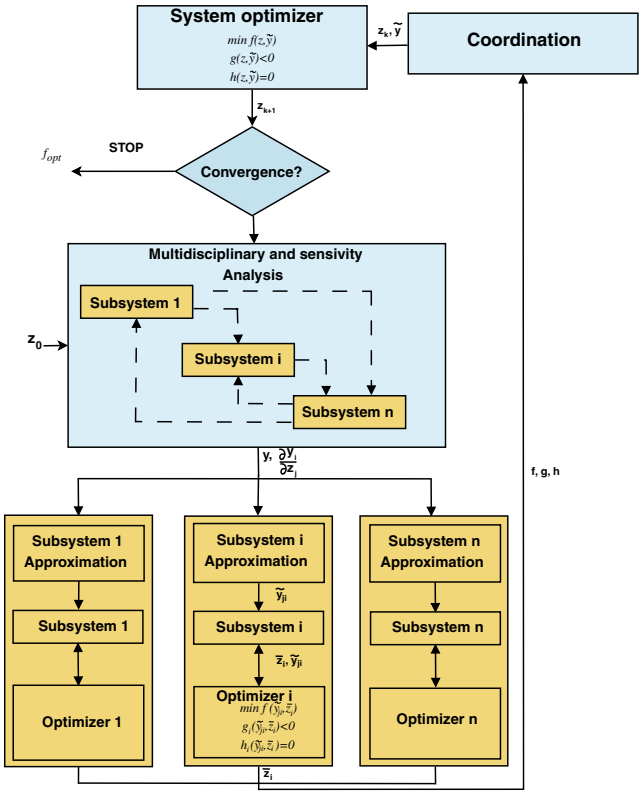

Fig. 7 CSSO method

Unfortunately, the efficiency of the CSSO method highly depends on the approximate models of the coupling variables.

Bi-Level Integrated System Synthesis The Bi-Level Integrated System Synthesis (BLISS) method was initially proposed by Sobieszczanski-Sobieski et al. (1998, 2000). In this paper, BLISS and BLISS-2000 (Sobieszczanski-Sobieski et al. 2003) are described. BLISS 2000 aims to improve the efficiency of the classical BLISS method by using response surfaces and other approximate models.

BLISS (Fig. 8) is an iterative multi-level method which is organized into a global optimizer and a set of optimizers at the subsystem level. This method resorts to a system analysis and subsystem analyses to improve the shared and individual design variables $z_{sh}, \bar{z}_i$. The method is based on a gradient approach and optimizes successively the contributions of the shared and individual design variables to the objective function.

At the kth iteration, the system-level solves the following problem:

$$\begin{aligned} \text{given} \quad & f_k, z_{sh_k} \\ \text{minimize} \quad & f_k + \frac{\partial f}{\partial z_{sh}} \Delta z_{sh} \\ \text{with respect to} \quad & \Delta z_{sh} \end{aligned}$$

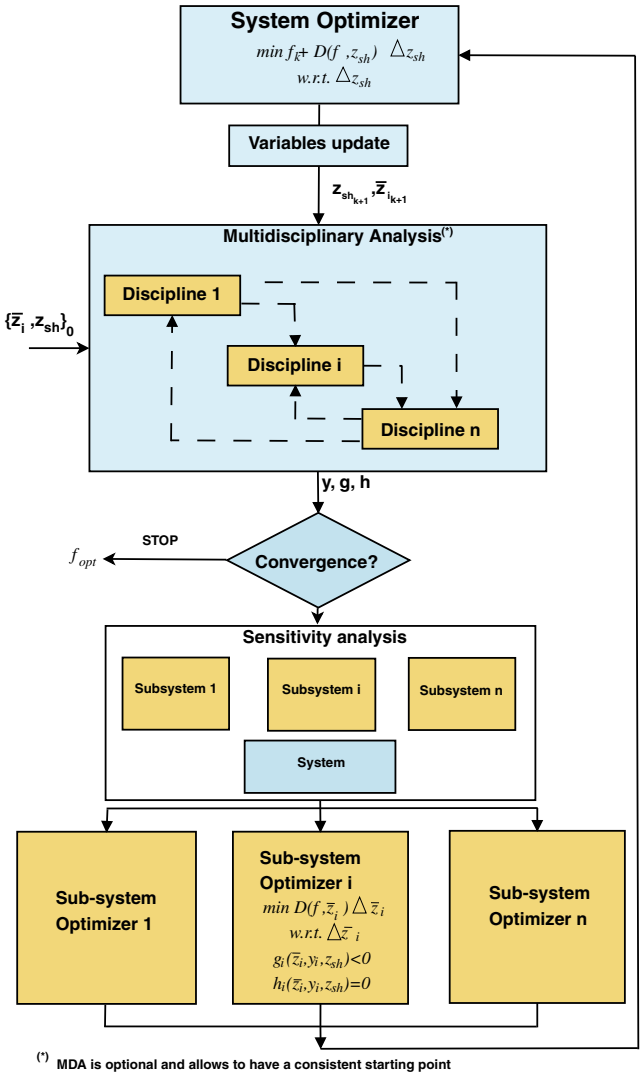

Fig. 8 BLISS method

The ith subsystem solves the problem:

$$\begin{aligned} \text{given} \quad & z_{sh}, \bar{z}_i, y_i \\ \text{minimize} \quad & \frac{\partial f}{\partial \bar{z}_i} \Delta \bar{z}_i \\ \text{with respect to} \quad & \Delta \bar{z}_i \\ \text{subject to} \quad & g(\bar{z}_i, z_{zh}, y_i) \leq 0 \qquad (26) \\ & h(\bar{z}_i, z_{sh}, y_i) = 0 \qquad (27) \end{aligned}$$

In BLISS2000, weighting factors w are added to structure the set of disciplinary outputs $\{o_i\}$. We denote by output o all variables which come from a disciplinary box. The set of output variables contains the coupling variables c and the state variables x which result of the subsystems optimizations. The objective of each subsystem is to minimize a weighted sum of their outputs, considering the global variables, the coupling variables and the weights as constants.

In BLISS 2000, the local optimizers are replaced by sets of response surfaces which approximate the optimized outputs c from the inputs given by the global optimizer. All along the optimization process, the response surfaces can be improved by adding and discarding some points.

The problem at the system-level can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f(z, y, w) = \tilde{o}_j \\ \text{with respect to} \quad & z, y, w \\ \text{subject to} \quad & y_i = \{\tilde{c}_{ji}(y_i, z_i)\}_j \quad \forall i, \forall j \neq i, \end{aligned} \tag{28}$$

For the ith local optimizer, the optimization problem is the following:

$$\begin{aligned} \text{minimize} \quad & \sum_k w_{ik}.o_{ik} \\ \text{with respect to} \quad & \bar{z}_i \\ \text{subject to} \quad & g_i(y_i, \bar{z}_i) \leq 0 \qquad (29) \\ & h_i(y_i, \bar{z}_i) = 0 \qquad (30) \end{aligned}$$

The parameters $\{z_{sh}, y, w\}$ are considered as constants during the subsystem optimization.

Advantages and drawbacks of BLISS method The main advantage of the BLISS method is to separate the system-level optimization and the ones in the different subsystems. Thus, the use of specific optimization tools for each subsystem is possible, because the disciplines are considered as black boxes by the system-level optimizer, which has the knowledge of only the outputs of the different subsystems.

The BLISS and BLISS-2000 methods, like all the gradient-based iterative algorithms, need move limits on the optimization variables. These methods will not be able to converge if the search space is too large or poorly defined. Like all the process using approximate models, the efficiency of BLISS-2000 method is highly dependent of the quality of the response surfaces which approximate the solutions of the optimization subproblems (Fig. 9). Nevertheless, these models can be generated off-line by specialists which make them more stable to design parameters variations. BLISS method prefers to handle a small number of global design variables and give good results for well decomposed problems.

Modified Collaborative Optimization The Modified Collaborative Optimization (MCO) method, described in DeMiguel and Murray (2000); DeMiguel (2001), presents the same architecture as CO but tries to correct some drawbacks of the CO method (e.g. non differentiability problems).

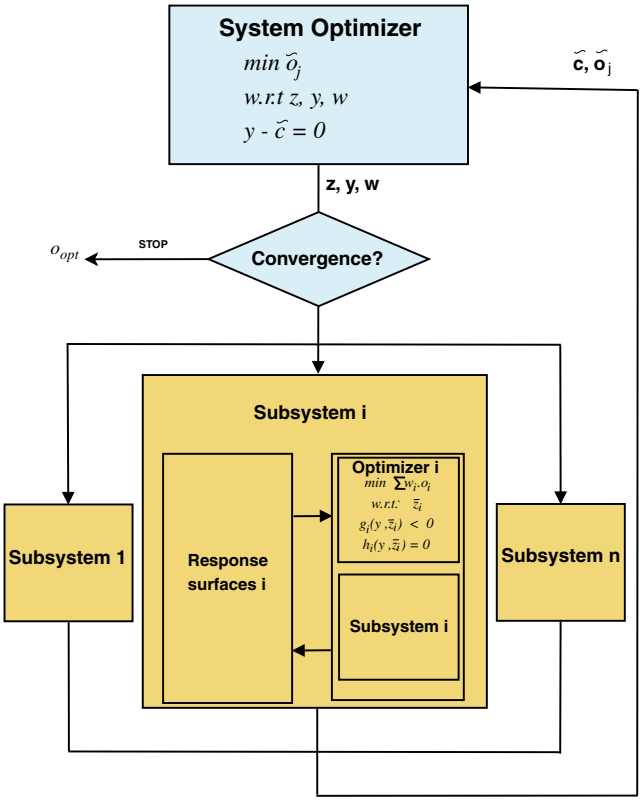

Fig. 9 BLISS2000 method

MCO replaces the quadratic penalty function used in CO by an exact form. Indeed, in order to avoid the instabilities induced by the quadratic penalty function (Gill et al. 1981), MCO replaces the quadratic form of local objective by an IDF form, in adding elastic variables (s_i and t_i). Furthermore, the equality constraints at the system-level are replaced by a penalty function. The formulation of the optimization problem at the system-level is:

$$\begin{aligned} \text{minimize} \quad & f(z, y) + w \sum_{i=1}^{N} J_i^*(z, y) \\ \text{with respect to} \quad & z, y \end{aligned}$$

The problem formulation of the ith subsystem is:

$$\begin{aligned} \text{minimize} \quad & J_i = \sum (s_i + t_i) \\ & \equiv \|z_i^* - z_i\|_1 + \|y_{ij} - c_{ij}\|_1 \\ \text{with respect to} \quad & z_i^*, s_i = \{z_{si}, c_{si}\}, t_i = \{z_{ti}, c_{ti}\}, r_i \\ & h_i(z_i^*) = 0 \qquad (31) \\ & g_i(z_i^*) + r_i = 0 \qquad (32) \\ \text{subject to} \quad & z_i^* + z_{si} - z_{ti} = z_i \qquad (33) \\ & c_{ij} + c_{si} - c_{ti} = y_i \qquad (34) \\ & s_i, t_i, r_i \geq 0 \qquad (35) \end{aligned}$$

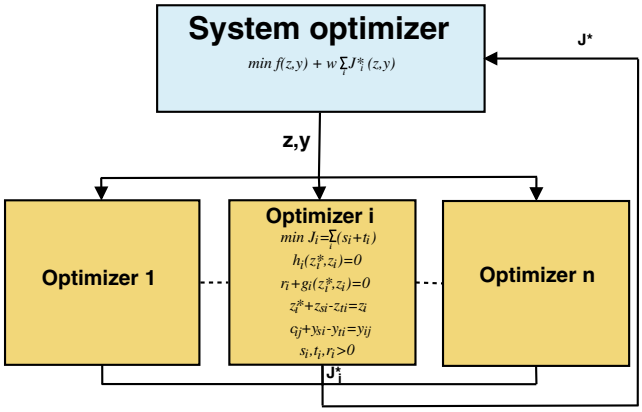

Fig. 10 MCO method

where:

- r_i slack variables added to transform the inequality constraint $g \leq 0$ in an equality one.
- s_i, t_i elastic variables (Boman 1999; Gill et al. 1981) ($s_i, t_i = 0$ at convergence)
- w penalty coefficient (weight)

Advantages and drawbacks of MCO method MCO presents theorically some advantages in comparison with CO method. Indeed, MCO avoids instabilities of CO method and aims to improve convergence properties (Fig. 10).

Analytical Target Cascading The Analytical Target Cascading (ATC), described in Michelena et al. (1999, 2003); Kim (2001), has been initially developed to formalize industrial product development processes. This formulation is very suitable to solve problems with a hierarchical structure. ATC is a multi-level MDO method (possibly involving more than two levels) which hierarchically propagates system and subsystem level targets through the different subsystem levels. In this formulation, the initial problem is subdivided into a set of subproblems. The specified design targets are "cascaded" from the system level to the lower levels and are also rebalanced to the higher levels after being optimized at the lower levels. At each level of the design process, a specific optimization problem is formulated to minimize the errors between the level outputs and the propagated objectives, and thus to ensure the consistency about the couplings between the upper and lower optimization levels. For some problems, the mathematical formulation of ATC can be similar to the CO one (Allison et al. 2005).

Let S_{ij} be the jth subsystem of the ith optimization level, the optimization problem to solve for this subsystem is the following:

$$\begin{aligned} \text{minimize} \quad & f_{ij} = \|C_{ij} - y_{(i-1)j}\| \\ & + \|z_{sh_{(i-1)j}} - z^*_{sh_{(i-1)j}}\| \\ & + \epsilon_{C_{ij}} + \epsilon_{Z_{ij}} \end{aligned}$$

$$\begin{aligned} \text{with respect to} \quad & \bar{z}_{ij}, z_{sh_{ij}}, z^*_{sh_{(i-1)j}}, y_{ij}, \epsilon_{C_{ij}}, \epsilon_{Z_{ij}} \\ \text{where} \quad & C_{ij} = c_{ij}(y_{ij}, \bar{z}_{ij}, z^*_{sh_{(i-1)j}}) \end{aligned}$$

$$\sum_{k \in Child_{ij}} \|y_{ij_k} - C_{(i+1)j_k}\| \leq \epsilon_{C_{ij}} \tag{36}$$

$$\text{subject to} \quad \sum_{k \in Child_{ij}} \|z_{sh_{ij_k}} - z^*_{sh_{ij_k}}\| \leq \epsilon_{Z_{ij}} \tag{37}$$

$$g_{ij}(c_{ij}, \bar{z}_{ij}, z^*_{sh_{(i-1)j}}) \leq 0 \tag{38}$$

$$h_{ij}(c_{ij}, \bar{z}_{ij}, z^*_{sh_{(i-1)j}}) = 0 \tag{39}$$

with $\bar{z}_{ij}$ the design variables of S_{ij}, $z_{sh_{ij}}$ the shared design variables of S_{ij}, $z^*_{sh_{(i-1)j}}$ the local copies of the S_{ij}'s parent shared design variables, C_{ij} the responses of S_{ij}, $C_{(i+1)j_k}$ the responses of the kth S_{ij}'s child, y_{ij} the coupling variables of S_{ij}, $y_{(i-1)j}$ the S_{ij}'s parent coupling variables, $\epsilon_{C_{ij}}$ and $\epsilon_{Z_{ij}}$ the relative tolerances on the satisfaction of the inequality constraints (38) and (39). $Child_{ij}$ stands for the childhood of S_{ij}.

For the top level ($i = 1$), the objective function does not involve the satisfaction of the coupling constraints and the term $y_{(i-1)j}$ is replaced by the real target to reach (the variables $z^*_{sh_{(i-1)j}}$ are not necessary). In the same way, at the bottom level, the equations (36) and (37) are not necessary and the bottom level subsystems optimizers only involve the variables $\bar{z}_{ij}, z^*_{sh_{(i-1)j}}$.

Advantages and drawbacks of the ATC method ATC is a generic formulation adapted to large scale problems which can be solved with a multilevel structure (Fig. 11).

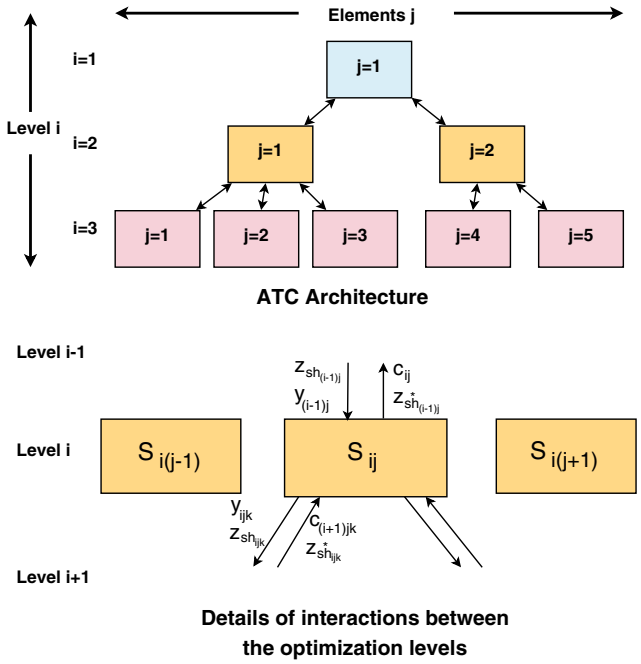

Fig. 11 ATC method

By partitioning the MDO process into a series of levels, ATC allows to distribute the complexity of the MDO problem into the different subsystems present in the different optimization levels. For that purpose, ATC is adapted for the MDO problems which can be decomposed into many small subproblems. ATC has been improved using Lagrangian coordination (Kim et al. 2006) and has been recently adapted to non-hierarchical formulation (Tosserams et al. 2010). Convergence proof and parallelization procedures of ATC have been proposed (Michelena et al. 2003; Han and Papalambros 2010).

Discipline Interaction Variable Elimination The Discipline Interaction Variable Elimination (DIVE) method, described in Auroux et al. (2009), Clément (2009), Masmoudi and Parte (2006) and Masmoudi et al. (2008), is derived form of BLISS-2000 (Agte 2005) and aims to simplify the optimization problem by solving the global, local and coupling variables. A MDA is used to handle the coupling between the different subsystems. The main characteristic of this method is to use meta-models to simplify the optimization problem.

Firstly, the local variables z_i are solved. Each discipline performs its own optimization problem by optimizing its own local variables $\bar{z}_i$, and is subject to its own local constraints g_i and h_i.

$$\begin{aligned} \text{minimize} \quad & f_i(y_i, \bar{z}_i, z_{sh}) \\ \text{with respect to} \quad & \bar{z}_i \\ \text{subject to} \quad & g_i(y_i, z_{sh}, \bar{z}_i) \leq 0 \qquad (40) \\ & h_i(y_i, z_{sh}, \bar{z}_i) = 0 \qquad (41) \end{aligned}$$

These local variables are calculated by local optimizers and are not used in the main optimization process.
Secondly, the coupling variables y are solved. This solving is performed by minimizing the following function:

$$\min_{y_i} \| c_i(y_i, z_{sh}) - y_i \|^2$$

Because the coupling problem is written in a quadratic form, good properties of convergence and effective coupling handling can be expected.

Finally, the global variables z_{sh} are calculated by the system-level optimizer, subject to the global constraints g and h. The subsystems use meta-models to compute their local functions and return a term a_i, which quantifies the fiability of the solution provided by the meta-models. This term is returned to the global optimizer and the validity of the meta-model at the point i is defined by the relationship $a_i < 0$. If the relationship is not valid, the point must be reevaluated.

The system-level problem can be summarized as follows:

$$\begin{aligned} \text{minimize} \quad & f(z_{sh}) \\ \text{with respect to} \quad & z_{sh} \\ & g(z_{sh}) \leq 0 \qquad (42) \\ \text{subject to} \quad & h(z_{sh}) = 0 \qquad (43) \\ & a(z_{sh}) \leq 0 \qquad (44) \end{aligned}$$

An optimal solution at the system-level z_{opt} must have the following properties:

- $\forall i \; a_i(z_{opt}) < 0$,
- the accuracy of the local equations using the meta-models have to be sufficient,
- if $a_i(z_{opt}) = 0$, the meta-model must be reevaluated.

Advantages and drawbacks of DIVE method Since the viability of meta-models and the accuracy of the local equations satisfaction are evaluated at z_{opt}, the insensitivity of the method to models uncertainties is improved. Moreover, the disciplinary experts can take part in the subsystem meta-models definition. In this way, adaptive meta-models can be created to improve the optimization efficiency throughout the design process (Fig. 12).

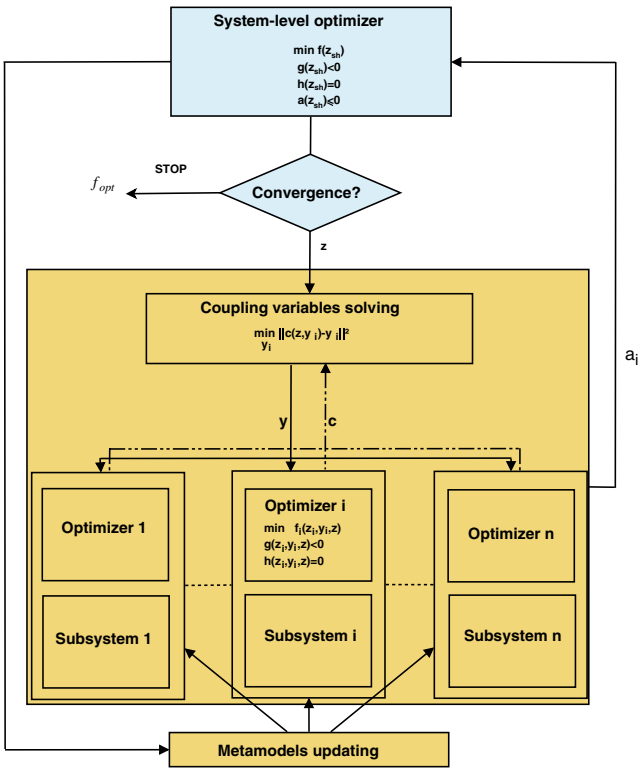

Fig. 12 DIVE method (Clément 2009)

A drawback of the method is its dependence on the generation of the meta-models which have to be accurate and to provide a good estimation of the solution. Indeed, a particularly great effort has to be realized on the generation of these meta-models in order to make them accurate and able to provide solutions in a small calculation time.

Dynamic Leader Follower The Dynamic Leader Follower (DyLeaf) method, proposed by Tava and Suzuki (2003) allows to split up the optimization problem without resorting to a system-level optimizer in order to coordinate the optimization process (Fig. 13). In this method, derived from the leader-follower game, each of the k optimizers takes successively the status of leader, optimizes its disciplinary criterion f_j and changes its local variables $\bar{z}_j$ and the global variables z_{sh}. The other $k-1$ optimizers (followers), optimize their objective f_i, modify their own design variables $\bar{z}_i$ and consider the global variables z_{sh} as constants. During the optimization, the status of the leader changes as many times as necessary. The leader optimizer is the one which corresponds to subsystem j, such as:

$$j = \underset{i}{\text{argmax}}(f_c(z_{sh}, \bar{z}_i)) \tag{45}$$

where f_c is the choice function of the leader. The status of the leader and the follower, depending on the relative satisfaction of each criterion, can change during the optimization process.

The formulation of the optimization problem is:

$$\begin{aligned}
&\text{for the leader} && (j\text{th subsystem})\\
&\text{minimize} && f_j(z_{sh}, \bar{z}_j)\\
&\text{with respect to} && z_{sh}, \bar{z}_j\\
&\text{subject to} && g_j(z_{sh}, \bar{z}_j) \leq 0 && (46)\\
& && h_j(z_{sh}, \bar{z}_j) = 0 && (47)
\end{aligned}$$

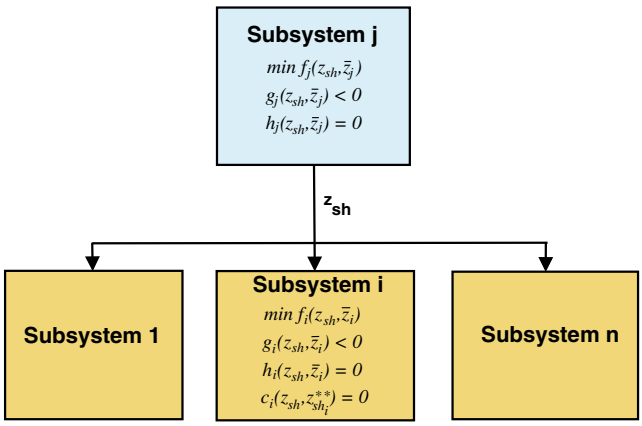

Fig. 13 Dyleaf method

$$\begin{aligned}
&\text{for the followers} && \forall i \neq j\\
&\text{minimize} && f_i(z_{sh}, \bar{z}_i)\\
&\text{with respect to} && \bar{z}_i, z^*_{sh_i}\\
& && g_i(z_{sh}, \bar{z}_i) \leq 0 && (48)\\
&\text{subject to} && h_i(z_{sh}, \bar{z}_i) = 0 && (49)\\
& && c_i(z_{sh}, z^*_{sh_i}) = 0 && (50)
\end{aligned}$$

where $z^*_{sh_i}$ are local copies of z_{sh} which intervene in the ith subsystem.

Advantages and drawbacks of DyLeaf method The DyLeaf method presents some advantages in comparison with the other methods. The priorities of the different disciplines can be finely adjusted with tuning on the selection function f_c. This method allows to treat multi-criteria problems, since all the objective functions are always optimized. Moreover, this method decomposes the large-scale problem into small ones. Therefore, improvements in robustness and computational time can be expected.

The main drawback of this method is that it needs a suitable decomposition of the problem to be efficient. Indeed, an effort has to be made at the decomposition step to correctly manage the variables between the different optimizers. Thus, the method can be not efficient for problems with a large number of interdisciplinary coupling variables. The paper Tava and Suzuki (2003) does not address convergence properties of the method and its behaviour when the objectives of the different disciplines are conflicting.

3 Optimization algorithms and trajectory handling

3.1 Trajectory optimization problem in launch vehicle optimization

The trajectory optimization is one of the specificity of launch vehicle design with regard to other MDO problems. This problem is frequently presented in literature and is usually performed by finding an optimal control law. Betts (1998) has summarized different numerical methods which can be used in launch vehicle trajectory optimization. Two types of methods are most widely used: direct and indirect methods.

Direct method In the direct method, a parametrical control law is established to optimize the trajectory. The parameters of the control law are calculated to optimize an objective and to satisfy equality and inequality constraints. We can consider single or multiple sections in the trajectory and use multiple shooting and collocation methods (transcription methods (Hargraves and Paris 1987)). The direct

method approach is suboptimal because the control law is expressed with a finite number of parameters.

The multiple shooting methods consist in using the direct methods in a subdivided problem. Indeed, the main problem is discretized into several sections, in order to improve the flexibility of the resolution. Conditions of continuity at the bounds of the different sections are added in order to ensure the consistency of the found trajectory.

The collocation methods discretize the state and the control law, and approximate the system equations by polynoms in order to optimize the trajectory. Indeed, the solution is a piecewise continuous polynomial that collocates (that satisfies the flight dynamics equations). The state and control equations are replaced by polynoms which are optimized in order to satisfy the system dynamics and to minimize an objective.

Indirect method The indirect shooting method is based of the Principle of Maximum of Pontryagin et al. (1962). This method is studied in details by Bryson and Ho (1975) and Bryson (1999). The principle of this method is to add an adjoint vector to the state vector owing to the expression of the Hamiltonian of the system. From the expression of the optimality and transversality conditions, the control law may be calculated. Like the direct shooting method, multiple sections can be considered in the trajectory in order to make the optimization process more flexible, and transcription methods can be used to efficiently perform the trajectory optimization.

3.2 Optimization algorithms used in MDO

3.2.1 Gradient-based algorithms

Gradient-based methods consist in differentiating the objective function and the constraints in order to evolve the variables. These methods are popular and well described (Fletcher 1987; Nocedal and Wright 2006). In literature, we can find several gradient-based algorithms (such as Post-Optimality (Braun et al. 1993), System Sensitivity Analysis (Olds 1994), Local Distributed Criteria (Filatyev et al. 2009)) which present some interest in a MDO framework.

3.2.2 Metaheuristic algorithms

Many metaheuristic algorithms can be found in literature. In this paragraph, we are interested in only three of them: the genetic algorithm, the ant colony algorithm and the simulated annealing algorithm.

One of the most classical metaheuristic process is the genetic algorithm (Holland 1975; Bäck 1996). This algorithm is based on the Evolution Theory in order to solve complex problems. It is inspired by biological evolution phenomena: mutation, reproduction, recombination and selection. This algorithm aims to bring a certain population to optimal values, by using stochastic process.

The Ant Colony Optimization algorithm (Dorigo and Süzle 2004) models the optimization problem as a minimum cost path problem in a graph. Artificial ants are used to solve this problem and to search good paths. These ants use pheromone trails to find the optimal path (stigmergy principle).

Simulated Annealing (Kirkpatrick et al. 1983) is an exploratory metaheuristic inspired of a process used in metallurgical industry. In this method, each point is considered as a state of some physical system, and the objective function to minimize is the internal energy of the system in that state. The goal is to bring the system from an initial state to a minimum energy state. This method uses random process in order to escape from local minima (by allowing occasional increases in the objective function) and reach the global optimum. This method is based on the Metropolis–Hastings algorithm (Metropolis et al. 1953).

Metaheuristics algorithms are adapted to solve multi-objective problems (e.g. Multi-Objective Simulated Annealing (Serafini 1992; Ulungu et al. 1999), Multi-Objective Genetic Algorithms (Coello 1998, 1999; Fonseca and Fleming 1993; Zitzler 1999), Ant Colony Optimization (Alaya et al. 2007; Doerner et al. 2001)).

3.2.3 Hybrid methods

We denote by hybrid methods the algorithms which combine metaheuristics and gradient-based methods. These methods aim to benefit from the efficiency of the gradient-based methods and the evolutionary algorithms ability to explore large space in order to find an initialization in the convergence domain of the local method and then to converge quickly toward the optimal solution (e.g. GAGGS method (Geethaikrishnan et al. 2008)).

4 Analysis of MDO methods application in launch vehicle design

In literature, we can find many papers treating the application of the MDO methods in launch vehicle design. The aims of this section is to synthesize and to compare (when it is possible) the different methods.

4.1 MDO methods and optimization algorithms

The Multi-Discipline Feasible method is the most employed to optimize the design of launch vehicle. Indeed, more than 80% of the surveyed papers for this study use a MDF

method to optimize the launch vehicle configuration. Nevertheless, we can find several articles which apply other methods like AAO, IDF, CO, BLISS or MCO (Table 2). Iterative methods such as gradient-based methods (including Local Distributed Criteria (LDC), Post-Optimality, SQP, GSE...) are preferred to evolutionary ones (more often Genetic Algorithms). We can also find several articles which propose hybrid methods (Akhtar and Linshu 2005, 2006; Briggs et al. 2007a; Geethaikrishnan et al. 2008; Rafique et al. 2009a) and combine genetic algorithms and gradient-based methods in order to benefit from the advantages of these two kinds of methods.

4.2 Optimization criteria and different study cases

The different study cases are resumed in Table 3. These cases indicate the different trends in launch vehicle design.

4.3 Optimal control handling

In the vast majority of the registered papers, the trajectory optimization is handled using a direct method. The trajectory optimization is often considered as a black box and a parametrical control law is found in order to reach the orbit. The parameters (of the command law and the state when collocation method is used (Tava and Suzuki 2002)) are considered as state variables x and are optimized by disciplinary analyzer (state equation solving: $R(x, y, z) = 0$). In most of the cases, the whole trajectory is optimized in one run.

Nevertheless, we can find several papers in which the trajectory is treated in a different way. In a few cases (more often MDF methods: Castellini et al. 2008; Delattre and Mongrard 2007; Duranté et al. 2004; Tsuchiya and Mori 2004), the control law parameters and the design variables z are treated at the same level and the residuals $R(x, y, z)$ are added to the set of equality constraints (2) to satisfy at the system-level.

Indirect methods (Pontryagin Maximum Principle) are used in Perrot (2003) and in the LDC method (Filatyev and Golikov 2008; Filatyev et al. 2009) to solve the trajectory optimization problem. In indirect methods, an adjoint vector x is used to indirectly optimize the control law. In this case, the residuals R are composed of the optimality and transversality conditions. The LDC method consists of

Table 2 Different MDO methods applied to launch vehicles design

	Evolutionary methods	Iterative methods
MDF	Akhtar and Linshu (2005, 2006)	Akhtar and Linshu (2005, 2006)
	Balesdent et al. (2010a, b)	Balesdent et al. (2010a, b)
	Bayley and Hartfield (2007) and Bayley et al. (2008)	Braun et al. (1993) and Braun and Kroo (1995)
	Briggs et al. (2007a, b)	Briggs et al. (2007a, b)
	Castellini et al. (2008) and Delattre and Mongrard (2007)	Filatyev and Golikov (2008) and Filatyev et al. (2009)
	Duranté et al. (2004) and Gang et al. (2005)	Geethaikrishnan et al. (2008) and Jodei et al. (2006)
	Lee et al. (2005b) and Rafique et al. (2009a)	Jodei et al. (2009) and Kalden (2007)
	Rafique et al. (2009b)	Kuratani et al. (2005) and Lee et al. (2005a)
		Lee et al. (2005b) and Moore et al. (1995)
		Olds (1992, 1994)
		Perrot (2003) and Qu et al. (2004)
		Rafique et al. (2009a) and Tava and Suzuki (2002)
		Tsuchiya and Mori (2002, 2004)
		Yokoyama et al. (2005, 2007)
IDF	/	Braun and Kroo (1995)
AAO	/	Braun et al. (1996) and Brown and Olds (2005)
		Brown and Olds (2006) and Tava and Suzuki (2003)
CO	Balesdent et al. (2010a, b)	Braun et al. (1996)
	Gang et al. (2005)	Brown and Olds (2005, 2006)
		Cormier et al. (2000) and Gang et al. (2005)
		Perrot (2003)
BLISS	/	Brown and Olds (2005, 2006)
MCO	/	Brown and Olds (2005, 2006)

Table 3 Different study cases of the MDO methods

Criterion	RLV-SSTO	RLV-MSTO	ELV-MSTO
Payload mass	Kalden (2007)	/	Castellini et al. (2008)
	Yokoyama et al. (2005)		Luo et al. (2004)
	Yokoyama et al. (2007)		
Cost	Braun et al. (1996)	Kuratani et al. (2005)	Bayley and Hartfield (2007)
	Kuratani et al. (2005)	Villeneuve and Mavris (2005)	Bayley et al. (2008)
	Moore et al. (1995)		Delattre and Mongrard (2007)
	Villeneuve and Mavris (2005)		Duranté et al. (2004)
Dry weight	Braun et al. (1993)	/	/
	Braun and Kroo (1995)		
	Brown and Olds (2005)		
	Brown and Olds (2006)		
	Cormier et al. (2000)		
	Olds (1992)		
	Olds and Walberg (1993)		
	Olds (1994)		
GLOW	/	Perrot (2003)	Balesdent et al. (2010a)
		Tsuchiya and Mori (2002)	Balesdent et al. (2010b)
		Tsuchiya and Mori (2004)	Akhtar and Linshu (2005)
			Akhtar and Linshu (2006)
			Briggs et al. (2007a)
			Briggs et al. (2007b)
			Jodei et al. (2006)
			Jodei et al. (2009)
			Liu et al. (1997)
			Luo et al. (2004)

an iterative method in which the design parameters optimization is guided by sensitivity calculations realized by an indirect approach.

Dynamic programming (Laurent-Varin et al. 2009; Liu et al. 1997) (Hamilton Jacobi Bellman approach) is more sporadicaly used in launch vehicle design because the implementation of this method is not easily applicable to a complete trajectory optimization problem.

4.4 Survey of papers with a common launch vehicle design case

Presentation of the optimization problem The problem considered in this section is the optimization of a reusable Single-Stage-To-Orbit (RLV SSTO) able to deliver 11 t of payload to the International Station (supposed to have a circular orbit of an altitude of 254 km and an inclination of 51.6°). The base vehicle configuration is optimized using three disciplines: Performance (trajectory optimization), Propulsion, and Weights and Sizing. The objective of the design process is to minimize the vehicle dry weight. This study case is used in many papers to apply the proposed MDO methods.

Comparison of AAO, MDF and CO methods In Braun et al. (1996), CO, AAO and MDF are compared in the case of the SSTO launch vehicle optimization. Moreover, these methods can be compared with SSA method used in Olds (1992, 1994) to solve an analog optimization problem.

In AAO (Braun et al. 1996), the optimizer has to handle 40 variables (36 parameters and four compatibility constraints). Hence, for 70 iterations, 2,870 analysis calls are required (more than 10,000 calls are required by using MDF method). Both AAO and MDF reach the same solution (within 0.05% in the dry weight).

In comparison with MDF, AAO allows to considerably reduce the computation time. This advantage is due to the fact that a consistent vehicle model is only required at the solution. Since disciplinary convergence is not required at each design point, the computation time is considerably reduced. Moreover, AAO can obtain accurate derivatives

without requiring strict convergence of the disciplinary models, that improves the process flexibility.

In the CO formulation, there are 23 design variables at the system-level and 72 at the subsystem one. At the system-level, the Jacobian is obtained from the use of Post-Optimality information (Braun et al. 1993; Braun 1996). A SQP algorithm NPSOL (Gill et al. 1986) is used to perform the system-level optimization. The objective function of the obtained solution has the same order of magnitude as the one obtained by MDF (Moore et al. 1995). In Braun and Kroo (1995), results obtained by CO are compared with the ones obtained AAO and MDF in Braun et al. (1995).

From the results of this study, we can bring out two main advantages of the CO method. Firstly, CO is a modular method. Indeed, we can add a discipline without spending much time to modify the optimization process. Therefore, CO is more flexible than MDF and AAO. The second main advantage of CO is that the communication requirements are reduced. This is due principally to the fact that in CO, a part of the optimization process is delegated to the different subspaces.

Comparison of FPI, AAO, BLISS-2000, CO and MCO methods FPI, AAO, BLISS-2000, CO and MCO methods have been compared in Brown and Olds (2006) in the same realistic SSTO optimization problem. Conventional disciplines and legacy codes are taken into account in order to be representative of classical RLV-SSTO early design studies.

Using the FPI method, each discipline optimizes their own variables, is subject to its own constraints and minimizes its own objectives. FPI is not a MDO method but is just a way to solve the Multidisciplinary Analysis problem and gives a reference to compare the methods.

In the AAO formulation, the optimizer is subject to two inequality constraints, seven equality constraints and changes 13 parameters. The results of this study show that AAO produces a 3.94% reduction in the dry mass versus the standard FPI approach.

The BLISS method studied in Brown and Olds (2006) is BLISS-2000 (Sobieszczanski-Sobieski et al. 2003). BLISS-2000 differs from BLISS-RSM (Altus 2002; Kodiyalam 2000) in the manner to apply the response surfaces.

The results of this study show than BLISS-2000 is able to find an optimum which presents a improvement of 3.76% in comparison with the traditionnal FPI method.

In the CO formulation, the objective of each discipline optimizer is to minimize a quadratic sum of coupling variable differences. The different disciplines handle the local copies of coupling variables and the objective of the global optimizer is to minimize the dry weight, subject to the coupling consistency.

Results show that CO is able to perform 4.39% reduction in the objective versus the FPI method, but provides a large convergence error.

In the MCO formulation, the objective at the system-level is to optimize the weighted sum of the dry mass and the three local objectives of the different disciplines. The system-level optimizer handles nine design parameters.

The study shows that in this test case, MCO led to inconsistent results. Indeed, the resulting solution varied greatly with the value of the penalty parameters at the system-level. Because of this phenomenon, it is impossible to conclude from this study about the efficiency of the MCO method.

Unfortunately, this study does not allow to conclude about which of the AAO, CO, BLISS-2000 and MCO methods is the best in the test case, because the results found by these different methods are very close. Nevertheless, the conclusions of this paper show that employing MDO methods can improve the design process (with respect to FPI). For the considered test case (which is not sufficient to provide general conclusions about the methods), the study shows that AAO is very difficult to implement and is not robust (because of resulting in a very large and highly constrained problem) but converges quickly with a good convergence error. CO is also difficult to implement, provides a better optimum but with a large convergence error. BLISS-2000 is the method which presents the best performance in the considered study case because it is not very difficult to implement, converges more quickly and has a better convergence error than CO.

In order to extend this comparison, we can compare the different methods through different qualitative criteria. This purpose is the object of the following section.

4.5 Comparative synthesis

4.5.1 Calculation costs and convergence speed

The methods which contain a multidisciplinary analysis are the ones that have the largest computation time. Indeed, the multidisciplinary analysis is a very expensive process, especially when it is coupled with a gradient-based method at the system-level.

Decoupling the MDA considerably reduces the computation time but introduces more variables which have to be handled by the optimizer. If the problem contains few disciplines, and therefore few coupling variables, IDF and AAO are the ones which converge fastest (Martins and Marriage 2007). As soon as the optimization problem becomes more complex, AAO is inapplicable because it is very difficult to implement.

Another way to improve the process efficiency is to implement a bi-level method. In that way, the main problem

can be subdivided into smaller ones and local optimizers share the coupling variables. The question of coupling variables can be handled as quadratic sums of differences (CO method), approximation by using response surfaces (CSSO) or by an IDF formulation (MCO).

CO can considerably improve the optimization process but generates instabilities (DeMiguel and Murray 2000) and an important convergence error (Brown and Olds 2006).

The performance of CSSO in complex problems is relatively poor. Indeed, the computation time used by the MDA and the one used to create the approximation models can exceed the calculation time saved by using them.

BLISS seems to have good convergence properties (Brown and Olds 2006) and the calculation costs can be reduced by using approximations models (BLISS-2000, BLISS-RSM). But there again, the question of using approximate models has to be studied individually, in order to determine if it allows or not to economize calculation time.

4.5.2 *Considerations about optimality conditions*

Among all the methods presented in this paper, AAO is the standard one because at the convergence (satisfaction of the Karush–Kuhn–Tucker conditions), this method ensures to find a local minimum. In principle, one should demonstrate that all the other methods are equivalent in the satisfaction of the same KKT conditions as AAO (apart from the errors introduced by using approximate models). This proof is given in Sobieszczanski-Sobieski et al. (2003) for the BLISS method. In other words, BLISS does not introduce any other errors than the ones of the approximate models and of the extrapolations between the different levels (no theoretical errors). Nevertheless, the BLISS-2000 method is subjected to convexity and response surfaces errors. Using the implicit function theorem (Kudryavtsev 2001), the satisfaction of the KKT conditions by using the single-level methods such as MDF and IDF can also be proved.

All the methods including an iterative scheme in the optimization process (BLISS, CSSO, etc.) and requiring intrinsically sensitivities calculations require convexity properties of the MDO problem to solve. This characteristic is a key difference with respect to other MDO methods (e.g. MDF, IDF, CO etc.) which allow to employ heuristics such as GA and consequently do not require strict convexity of the problem.

4.5.3 *Methods deftness*

Robustness In terms of robustness, defined here as the ability to converge to an optimum from large initialization domain, AAO seems to be the worst method. Indeed, since all the variables and all the constraints are handled by a single optimizer, AAO needs to be initialized in the vicinity of the optimum to converge.

Generally, in case of formulations which use approximate models, the robustness of the methods is correlated with the quality of the models and a particular effort has to be made on the generation of these models.

Applicability to large scale systems The heavily centralized methods are difficult to apply with large scale methods. Indeed, because the optimizer handles a lot of coupling variables, these methods are difficult to implement in case of large scale systems and have bad convergence properties. In that way, AAO appears as not applicable in large scale problems.

CO seems to perform well for only low number of coupling variables.

BLISS considering disciplines as black boxes, is applicable to large scale systems without difficulty, even if systems analyses are required because they can be performed in a network. BLISS presents some interest when both computer codes and engineering groups work concurrently, and gives some autonomy to the engineering groups in the different subsystems.

If the disciplinary models are not too complex, CSSO is also applicable in large scale systems, but its efficiency is dependent on the generated approximate models.

Concerning the applications of the MDO methods in an industrial context (when engineering teams intervene), the single-level methods are difficult to use because they are not adapted to the organization of the industrial structures. Moreover, these methods give very small autonomy to engineering teams which have no decision power on the design and just provide disciplinary analyses.

Contrariwise, the multi-level methods (such as CO, CSSO, BLISS ...) can distribute the disciplinary optimizations to specialty engineering teams and furthermore are more adapted to industrial problems. However, the efficiency of CSSO is to be put in perspective because in this formulation the different subsystems require the approximations of the other subsystems coupling influences.

Adaptability and implementation difficulty Since a lot of complex disciplines are taken into account in the launch vehicle optimization process, the modification time is an important criterion to compare the different MDO methods.

The bi-level methods (BLISS, CO, CSSO, MCO) seem to be more adaptable to a variation of the number of disciplines. For example, comparative studies in launch vehicle design (Braun et al. 1996) show that the modification time of CO is 75% less than the standard approach one and 66% less than the AAO method one. In BLISS since the disciplines are considered as black boxes, the global optimizer

does not intervene in the black boxes and adding a discipline does not change the process architecture.

However, the implementation difficulty of the CSSO method is heavily dependent on the disciplinary models. There again, since approximate models have to be generated in CSSO a very complex discipline can be approximated by non smooth models which provide instabilities and consume a lot of computation time.

Another important criterion which has to be taken into account is the different methods adaptability when the disciplinary models or the application case change (e.g. moving from a Single-Stage-Or-Orbit launch vehicle to a Two-Stage-To-Orbit one, minimizing the dry mass instead of maximizing the payload mass, etc.). MDF requires a global modification every time the design process changes, because all the disciplines are interlinked with the MDA. IDF and AAO are more flexible because each discipline evolves independently from the others. However, these methods are specific to an application case and when this case changes, all the process can be reorganized.

There again, the bi-level methods present some advantages with respect to the single-level ones. Indeed, these methods (especially CO and BLISS) are generic at the subsystem-level (black boxes for the system-level) and all the design process has not to be revised when a disciplinary model changes. This point is not true for CSSO because for a given subsystem, the influences of the other subsystem comportments have to be approximated. For CO and MCO, because the different subsystems optimize a quadratic sum or relative errors, a change in the system-level objective does not imply any modification at subsystem-level. For BLISS-2000, the adaptability of the method to a modification of the global objective function is ensured by the weighting factors which can evolve all along the optimization process.

5 Concluding remarks and ways of improvement

In the light of this survey, we can recommend several ways of improvement relating to the use of MDO methods in the specific launch vehicle design problem.

5.1 Requirement of a global comparison of all the methods

It would be interesting to compare all the methods described in the Section 2 (including the methods already compared in Brown and Olds (2006) and other methods such as DIVE, ATC, DyLeaf, etc.) in a common framework. Such a comparison has been performed in case of analytical problems (Tedford and Martins 2010) and would be valuable in a Launch Vehicle Design study case in order to identify which MDO method is the best for this application.

5.2 Inclusion of design stability aspects in the optimization process

The design stability aspect, defined here as insensitivity to uncertainties associated with both design parameters and design models, is a key factor in launch vehicle design. This particularity especially takes sense in early phase studies, where simplified models are employed (e.g. in structure discipline, finite element codes are often replaced by simplified relationships).

From this point of view, it appears essential that this aspect has to figure as an objective of the optimization process (Du and Chen 2002). Indeed, to make sure that the process goes smoothly, it would be interesting to insert a stability indicator, in order to help the designer to evaluate the quality of the solution found by the optimization process. In case of the use of gradient-based algorithms, the design stability with respect to the constraints can be evaluated by the Lagrange multipliers which can be used to quantify the variation of the objective function with respect to a violation of saturated constraints.

5.3 Evolutionary and gradient-based algorithms coupling

In order to have a robust (to starting points), efficient and global optimization process, coupling evolutionary and gradient-based algorithms at all optimization levels may be an interesting way of improvement. Indeed, a lot of synergisms could emanate from the coupling of these two types of methods in the MDO process, like in the GAGGS method (Geethaikrishnan et al. 2008). With the evolutionary algorithms, the process could explore the entire design space, and therefore avoid the local optima phenomena. By using gradient-method algorithms, the efficiency of the method could be considerably improved, and sensitivity information could be exploited by the evolutionary algorithms in order to bring the population to optimal solutions.

5.4 Trajectory optimization in the design process

The launch vehicle design optimization is heavily coupled to the optimal trajectory calculation. Indeed, the entire process is dependent of the results of the trajectory optimization, in which the specifications of the mission intervene.

Almost all examples in literature use a direct shooting method to optimize the launch vehicle trajectory. This choice is principally due to the fact that in the direct method, the initialization of parameters is easier to find than in the indirect method. Nevertheless, the direct method only gives suboptimal solutions.

The real optimal solution is given by the indirect shooting method (Bryson and Ho 1975; Bryson 1999). This

method is more complex to implement but gives precious information about the convergence, i.e. about the ability of the proposed launch vehicle configuration to reach the specifications of the mission. These informations (adjoint vector and sensibility of the objective with regard to the parameters) could be exploited in the global optimization process (the optimality conditions would be integrated to the system-level equality constraints $h = 0$), in order to change the design parameters z and the state variables x. These considerations could improve the efficiency of the optimization process (like in the LDC method (Filatyev and Golikov 2008; Filatyev et al. 2009)).

5.5 Another subdivision of the optimization process

All examples in literature consider a subdivision of the problem into disciplines such as propulsion, structure, performance estimating, etc. The trajectory optimization is considered as a standard discipline and the trajectory parameters are optimized either at the same level with design parameters or independently.

In order to improve the adaptability and the flexibility of the optimization process, a subdivision into a stage-wise decomposition (Balesdent et al. 2010a, b) (instead of the one into the different disciplines) could replace the trajectory optimization at the root of the optimization process. This kind of decomposition has been already tested in a single discipline (trajectory) problem (Ledsinger and Olds 2002; Rahn and Schöttle 1994) and can provide good results in a MDO context. Indeed, the optimization process would be subdivided in as many subsystems as the number of stages. Each subsystem would regroup the different disciplines which intervene in the stage design and these subsystems would be piloted by the trajectory optimization (optimal control, maybe by indirect shooting method). Such a decomposition (Fig. 14) could be handled by a multi-level process (e.g. a collaborative method).

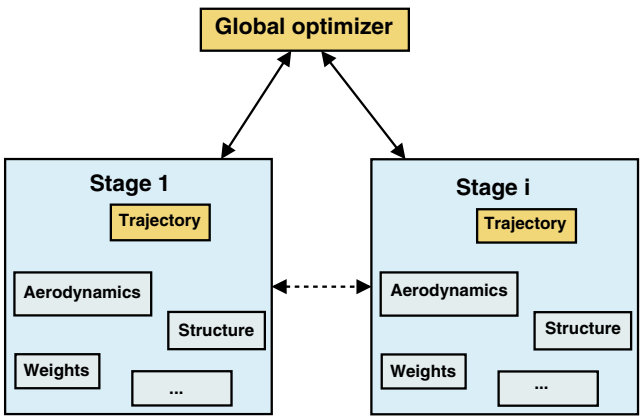

Fig. 14 Example of another process decomposition

References

Agte J (2005) A tool for application of bi-level integrated system synthesis to multidisciplinary design optimization problems. In: DGLR-2005-241, German air and space congress, Friedrichshafen, Germany

Agte J, de Weck O, Sobieszczanski-Sobieski J, Arendsen P, Morris A, Spieck M (2009) MDO: assessment and direction for advancement—an opinion of one international group. Struct Multidisc Optim. doi:10.1007/s00158-009-0381-5

Akhtar S, Linshu H (2005) Simulation-based optimization strategy for liquid fueled multi-stage space launch vehicle. In: 6th international conference on parallel and distributed computing, applications and technologies (PDCAT'05). Dalian, China

Akhtar S, Linshu H (2006) Support vector regression-driven multidisciplinary design optimization for multi-stage launch vehicle considering throttling effect. In: 44th AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Alaya I, Solnon C, Ghedira K (2007) Ant colony optimization for multi-objective optimization problems. In: 19th IEEE international conference on tools with artificial intelligence. Patras, Greece

Alexandrov N, Hussaini M (1995) Multidisciplinary design optimization: state of the art. In: Proceedings of the ICASE/NASA Langley workshop on multidisciplinary design optimization, SIAM proceedings series

Alexandrov N, Lewis R (2000) Algorithmic perspectives on problem formulations in MDO. In: 8th AIAA/UASAF/NASA/ISSMO symposium on multidisciplinary analysis & optimization. Long Beach, CA, USA

Allison J (2004) Complex system optimization: a comparison of analytical target cascading, collaborative optimization and other formulations. Ph.D. thesis, University of Michigan, USA

Allison J, Kokkolaras M, Zawislak M, Papalambros P (2005) On the use of analytical target cascading and collaborative optimization for complex system design. In: 6th world congress on structural and multidisciplinary optimization, 30 May–3 June, Rio de Janeiro, Brazil

Altus T (2002) A response surface methodology for Bi-Level Integrated System Synthesis (BLISS). NASA Technical Report n° NASA/CR-2002-211652. NASA Langley Research Center, USA

Auroux D, Clément J, Hermetz J, Masmoudi M, Parte Y (2009) Etat de l'art et nouvelles tendances en conception collaborative. Optimisation multidisciplinaire en mécanique. Hermes Science, ISTE Wiley, London

Bäck T (1996) Evolutionary algorithms in theory and practice. Oxford University Press, Oxford

Balesdent M, Bérend N, Dépincé P (2010a) Optimal design of expendable lauch vehicles using stage-wise MDO formulation. In: 13th AIAA/ISSMO multidisciplinary analysis and optimization conference. Fort Worth, Texas, USA. AIAA-2010-9324

Balesdent M, Bérend N, Dépincé P, Chriette A (2010b) Multidisciplinary design optimization of a multi-stage launch vehicle using flight phases decomposition. In: 3rd international conference on multidisciplinary design optimization and applications. Paris, France

Balling R, Sobieszczanski-Sobieski J (1994) Optimization of coupled systems: a critical overview of approaches. NASA/ICASE Report n° 94-100

Balling R, Wilkinson C (1997) Execution of multidisciplinary design optimization approaches on common test problems. AIAA J 35(1):178–186

Bayley D, Hartfield R (2007) Design optimization of space launch vehicles for minimum cost using a genetic algorithm. In: 43rd AIAA/ASME/SAE/ASEE joint propulsion conference and exhibit. Cincinnati, OH, USA. AIAA-2007-5852

Bayley D, Hartfield R, Burkhalter J, Jenkins R (2008) Design optimization of a space launch vehicle using a genetic algorithm. J Spacecr Rockets 45(4):733–740

Betts J (1998) Survey of numerical methods for trajectory optimization. J Guid Control Dyn 21(2):193–207

Boman E (1999). Infeasibility and negative curvature in optimization. Ph.D. thesis, Stanford University, USA

Braun R (1996) Collaborative optimization: an architecture for large-scale distributed design. Ph.D. thesis, Stanford University, USA

Braun R, Kroo I (1995) Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In: Multidisciplinary design optimization: state of the art. SIAM, Philadelphia, pp 98–116

Braun R, Kroo I, Gage P (1993) Post-optimality analysis in aerospace vehicle design. In: AIAA aircraft design, systems and operations meeting. Monterey, CA, USA

Braun R, Kroo I, Lepsch R, Powell R, Stanley D (1995) Comparison of two multidisciplinary optimization strategies for launch vehicle design. J Spacecr Rockets 32(3):404–410

Braun R, Moore A, Kroo I (1996) Use of the collaborative optimization architecture for launch vehicle design. In: 6th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. Bellevue, WA, USA

Briggs G, Ray T, Milthorpe J (2007a). Evolutionary algorithm use in optimisation of a launch vehicle stack model. In: 45th AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA. AIAA-2007-364

Briggs G, Ray T, Milthorpe J (2007b) Optimal design of an Australian medium launch vehicle. Innov Syst Softw Eng 2(3):105–116

Brown N, Olds J (2005) Evaluation of Multidisciplinary Optimization (MDO) techniques applied to a reusable launch vehicle. In: 43rd AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Brown N, Olds R (2006) Evaluation of multidisciplinary optimization techniques applied to a reusable launch vehicle. J Spacecr Rockets 43:1289–1300

Bryson A (1999) Dynamic optimization. Addison-Wesley, Menlo-Park

Bryson A, Ho Y-C (1975) Applied optimal control: optimization, estimation and control, revised printing. Taylor & Francis, Levittown

Castellini F, Lavagna M, Erf S (2008) Concurrent optimization of launcher architectures and ascent trajectories with global optimization algorithms. In: 59th international astronautical congress (IAC). Glasgow, Scotland

Clément J (2009) Optimisation multi-disciplinaire: étude théorique et application à la conception des avions en phase d'avant projet. Ph.D. thesis, Institut Supérieur de l'Aéronautique et de l'Espace. Toulouse, France

Coello C (1998) An updated survey of GA-based multi-objective optimization techniques. Technical Report RD-98-08. Laboratorio Nacional de Informatica Avanzada (LNIA). Xalapa, Veracruz, Mexico

Coello C (1999) A comprehensive survey of evolutionary-based multiobjective optimization. Knowl Inf Syst 1(3):129–137

Cormier T, Scott A, Ledsinger L, McCormick D, Way D, Olds R (2000) Comparison of collaborative optimization to conventional design techniques for a conceptual RLV. In: 8th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. Long Beach, CA, USA

Cramer E, Dennis J, Frank P, Lewis R, Shubin G (1994) Problem formulation for multidisciplinary optimization. SIAM J Optim 4:754–776

Delattre N, Mongrard O (2007) The use of multidisciplinary optimization for the design of expendable launchers. In: 2nd European conference for aerospace sciences (EUCASS). Brussels, Belgium

DeMiguel A (2001) Two decomposition algorithms for nonconvex optimization problems with global variables. Ph.D. thesis, Stanford University, USA

DeMiguel A, Murray W (2000) An analysis of collaborative optimization methods. In: 8th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. Long Beach, CA, USA

Doerner K, Gutjahr W, Hartl W, Strauss R, Stummer C (2001) Ant colony optimization in multiobjective porfolio selection. In: 4th metaheuristics international conference. Porto, Portugal

Dorigo M, Süzle T (2004) Ant colony optimization. MIT Press, Cambridge

Du X, Chen W (2002) Efficient uncertainty analysis methods for multidisciplinary robust design. AIAA J 40(3):545–552

Duranté N, Dufour A, Pain V (2004) Multidisciplinary analysis and optimisation approach for the design of expendable launchers. In: 10th AIAA/ISSMO multidisciplinary analysis and optimization conference. Albany, New-York, USA

Filatyev A, Golikov A (2008) The complex approach to optimization of STS parameters on the maximum principle basis. In: 59th international astronautical congress (IAC). Glasgow, Scotland

Filatyev A, Golikov A, Shanygin A, Voityshen V (2009) The local distributed criteria method for multidisciplinary optimization of launchers parameters. Acta Astron 65:584–590

Fletcher R (1987) Practical methods of optimization, 2nd edn. Wiley, New York

Fonseca C, Fleming P (1993) Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In: Forrest S (ed) 5th international conference on genetic algorithms. San Mateo, CA, USA

Gang C, Min X, Zi-ming W, Si-Lu C (2005) Multidisciplinary design optimization of RLV reentry trajectory. In: 13th AIAA/CIRA international space planes and hypersonices systems and technologies. Capua, Italy

Geethaikrishnan C, Mujumbar P, Sudhakar K, Adimurthy VV (2008) Genetic algorithm guided gradient search for launch vehicle trajectory optimization. In: International conference on aerospace science and technology. Bangalore, India

Gill P, Murray W, Wright M (1981) Practical optimization. Academic, New York

Gill P, Murray W, Sainders M, Wright M (1986) Users guide for NPSOL (version 4.0): a Fortran package for nonlinear programming. Technical Report SOL 86-2. Stanford University, USA

Han J, Papalambros P (2010) A note on the convergence of analytical target cascading with infinite norms. J Mech Des 132:034502

Hargraves C, Paris S (1987) Direct trajectory optimization using nonlinear programming and collocation. J Guid Control Dyn 10(4):338–342

Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Huang C, Bloebaum C (2004) Multi-objective Pareto concurrent subspace optimization for multidisciplinary design. In: 42nd AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Jodei J, Ebrahimi M, Roshanian J (2006) An automated approach to multidisciplinary system design optimization of small solid propellant launch vehicles. In: 1st ISSCAA international symposium on systems and control in aerospace and astronautics. Harbin, China

Jodei J, Ebrahimi M, Roshanian J (2009) Multidisciplinary design optimization of a small solid propellant launch vehicle using system sensitivity analysis. Struct Multidisc Optim 38:93–100

Kalden O (2007) Multidisciplinary design and trajectory optimization of the reusable launch vehicle concept Hopper. In: 3rd international conference on recent advances in space technologies. Istanbul, Turkey

Keane A, Nair P (2005) Computational approaches for aerospace design: the pursuit of excellence. Wiley, New York

Kim H (2001) Target cascading in optimal system design. Ph.D. thesis, University of Michigan

Kim H, Chen W, Wiecek M (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. AIAA J 44(10):2197–2207

Kirkpatrick S, Gelatt C, Vecchi J (1983) Optimization by simulated annealing. Science, New Series 220(4598):671–680

Kodiyalam S (1998) Evaluation of methods for Multidisciplinary Design Optimization (MDO), phase 1. NASA/CR-1998-20716

Kodiyalam S (2000) Evaluation of methods for Multidisciplinary Design optimization (MDO), phase 2. NASA/CR-2000-210313

Kroo I (2004) Distributed multidisciplinary design and collaborative optimization. In: VKI lecture series on optimization methods and tools for multicriteria/multidisciplinary design

Kudryavtsev L (2001) Implicit function. In: Hazewinkel M ed.) Encyclopaedia of mathematics. Springer, New York

Kuratani N, Suzuki H, Goeklich RA (2005) Multidisciplinary optimization of space transportation systems. In: 56th international astronautical congress of the international astronautical federation. Fukuoka, Japan

Laurent-Varin J, Bolanowski O, Cristiani E, Zidani H (2009) Trajectory optimization of launcher with Hamilton–Jacobi–Bellman approach. In: 3rd European conference for aerospace sciences. Versailles, France

Ledsinger L, Olds J (2002) Optimized solutions for Kistler K-1 Branching Trajectories using Multidisciplinary Design Optimization techniques. J Spacecr Rockets 39(3):420–429

Lee J-W, Choi Y-C, Byun Y-H (2005a). Optimal supersonic air-launching rocket design using multi-disciplinary system optimization approach. J Korean Soc Aeronaut Space Sci 33:12

Lee J-W, Jeon K-S, Byun Y-H, Kim S-J (2005b) Optimal space launcher design using a refined response surface method. In: Lecture notes in computer science, vol 3613/2005. Springer, Berlin

Liu X, Xiang A, Gang T (1997) A new efficient method of multi-stage solid rocket optimization. In: 33rd AIAA/ASME/SAE/ASEE joint propulsion conference and exhibit. Seattle, WA, USA

Luo Y, Liang Y, Zhou L (2004) Multi-level, multi-objective and multidisciplinary optimization design of a series of launch vehicles. In: 55th international astronautical congress. Vancouver, Canada

Martins J, Marriage C (2007) An object-oriented framework for Multidisciplinary Design Optimization. In: 3rd AIAA multidisciplinary design optimization specialist conference. Waikiki, Hawaii, USA

Masmoudi M, Parte Y (2006) Discipline Interaction Variable Elimination (DIVE) approach for MDO. In: European conference on computational fluid dynamics (ECCOMAS CFD). Egmond aan Zee, The Netherlands

Masmoudi M, Parte Y, Hermetz J, Clement J (2008) DIVE: a new approach for multidisciplinary optimization. In: 5th European congress on computational methods in applied sciences (ECCOMAS). Venice, Italy

Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equations of state calculations by fast computing machines. J Chem Phys 21:1087–1092

Michelena N, Kim H, Papalambros P (1999) A system partitioning and optimization approach to target cascading. In: Proceedings of the 12th international conference on engineering design. Munich, Germany

Michelena N, Park H, Papalambros P (2003) Convergence properties of analytical target cascading. AIAA J 41(5):897–905

Moore A, Braun R, Powell R (1995) Optimal technology investment strategies for a reusable launch vehicle. In: AIAA space programs and technologies conference. Huntsville, AL, USA

Nocedal J, Wright S (2006) Numerical optimization, 2nd edn. Springer, Berlin

Olds J (1992) The suitability of selected multidisciplinary design and optimization techniques to conceptual aerospace vehicle design. In: 4th AIAA/USAF/NASA/OAI symposium on multidisciplinary analysis and optimization. Cleveland, OH, USA

Olds J (1994) System sensitivity analysis applied to the conceptual design of a dual-fuel rocket SSTO. In: 5th AIAA/NASA/USAF/ISSMO symposium on multidisciplinary analysis and optimization. Panama City Beach, FL, USA

Olds J, Walberg G (1993) Multidisciplinary design of a rocket-based combined cycle SSTO launch vehicle using Taguchi methods. In: AIAA/AHS/ASEE aerospace design conference. Irvine, CA, USA

Pareto V (1971) Manual of political economy. A.M. Kelley, New York

Perez V, Renaud J, Watson L (2002) Adaptive experimental design for construction of response surface approximations. AIAA J 40(12):2495–2503

Perrot L (2003) Optimisation of reusable launchers by decomposition and coordination of trajectory optimisation problems. In: 5th international conference on launcher technology, space launchers: missions, control and avionics. Madrid , Spain

Pontryagin L, Boltyansky V, Gamkrelidze R, Mischenko E (1962) The mathematical theory of optimal processes. Interscience, New York

Qu M, Gage P, Sohn J, Bradford J, Starr B (2004) Generalized vehicle performance closure model for two-stage-to-orbit launch vehicles. In: 42nd AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Rafique A, LinShu H, Zeeshan Q, Nisar K, Xiaowei W (2009a) Hybrid optimization method for multidisciplinary design of air launched satellite launch vehicle. In: 45th AIAA/ASME/SAE/ASEE joint propulsion conference & exhibit. Denver, Colorado, USA

Rafique A, LinShu H, Zeeshan Q, Nisar K, Xiaowei W (2009b) Integrated system design of air launched small space launch vehicle using genetic algorithm. In: 45th AIAA/ASME/SAE/ASEE joint propulsion conference & exhibit. Denver, Colorado, USA

Rahn M, Schöttle U (1994) Decomposition algorithm for flight performance and system optimization of an airbreathing launch vehicle. In: 45th congress of International Astronautical Federation. Jerusalem, Israel. IAF-94.A.2.018

Renaud J, Gabriele G (1993) Improved coordination in nonhierarchic system optimization. AIAA J 31(12):2367–2373

Renaud J, Gabriele G (1994) Approximation in nonhierarchic system optimization. AIAA J 32(1):198–205

Rodriguez J, Renaud J, Watson L (1998) Trust region augmented Lagrangian methods for sequential response surface approximation and optimization. J Mech Des 120:58–66

Rodriguez J, Perez V, Padmanabhan D, Renaud J (2001) Sequential approximate optimization using variable fidelity response surface approximations. Struct Multidisc Optim 22(1):24–34

Sellar R, Batill S (1996) Concurrent subspace optimization using gradient-enhanced neural network approximations. In: 6th AIAA/NASA/USAF multidisciplinary analysis and optimization symposium. Bellevue, WA, USA

Sellar R, Batill S, Renaud J (1996) Response surface based, concurrent subspace optimization for multidisciplinary system design. In: 34th AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Serafini P (1992) Simulated annealing for multiple objective optimization problems. In: 10th international conference on multiple criteria decision making. Taipei, Taiwan

Sobieski I (1993) Multidisciplinary design optimization: attempt at definition. In: Industry university workshop on multidisciplinary aircraft design, pp 23–48

Sobieski I, Kroo I (2000) Collaborative optimization using response surface estimation. AIAA J 38(10):1931–1938

Sobieszczanski-Sobieski J (1988) Optimization by decomposition: a step from hierarchic to non-hierarchic systems. NASA Technical Report CP-3031

Sobieszczanski-Sobieski J (1990) Sensitivity of complex internally coupled systems. AIAA J 28(1):153–160

Sobieszczanski-Sobieski J, Haftka R (1997) Multidisciplinary aerospace design optimization: survey of recent developments. Struct Optim (14):1–23

Sobieszczanski-Sobieski J, Agte J, Sandusky R (1998) Bi-Level Integrated System Synthesis (BLISS). NASA/TM-1998-208715

Sobieszczanski-Sobieski J, Emiley M, Agte J, Sandusky R (2000) Advancement of Bi-Level Integrated System Synthesis (BLISS). NASA/TM-2000-210305

Sobieszczanski-Sobieski J, Altus T, Phillips M, Sandusky R (2003) Bi-level Integrated System Synthesis for concurrent and distributed processing. AIAA J 41(10):1996–2003

Tava M, Suzuki S (2002) Multidisciplinary Design Optimization of the shape and trajectory of a reentry vehicle. Trans Jpn Soc Aeronaut Space Sci 45(147):10–19

Tava M, Suzuki S (2003) Integrated multidisciplinary and multicriteria optimization of a space transportation system and its trajectory. In: 54th international astronautical congress of the International Astronautical Federation. Bremen, Germany

Tedford N, Martins J (2010) Benchmarking multidisciplinary design optimization algorithms. Optim Eng 11:159–183

Tosserams S, Etman L, Rooda J (2009) A classification of methods for distributed system optimization based on formulation structure. Struct Multidisc Optim

Tosserams S, Kokkolaras M, Etman L, Rooda J (2010) A nonhierarchical formulation of analytical target cascading. J Mech Des 23:051002

Tsuchiya T, Mori T (2002) Multidisciplinary Design Optimization to future space transportation vehicles. In: AIAA-2002-5171

Tsuchiya T, Mori T (2004) Optimal conceptual design of two-stage reusable rocket vehicles including trajectory optimization. J Spacecr Rockets 41(5):770–778

Ulungu E, Teghem J, Fortemps P, Tuyttens D (1999) MOSA method: a tool for solving multi-objective combinatorial optimization problems. J Multi-Crit Decis Anal 8:221–2396

Vanderplaats G (1999) Numerical optimization techniques for engineering and design, 3rd edn. Vanderplaats Research & Development, Inc., Colorado Springs

Villeneuve F, Mavris D (2005) A new method of architecture selection for launch vehicles. In: 13th AIAA/CIRA international space planes and hypersonics systems and technologies. Capua, Italy

Wujek B, Renaud J, Batill S (1997) A concurrent engineering approach for multidisciplinary design in a distributed computing environment. In: Alexandrov N, Hussaini MY (eds) Multidisciplinary Design Optimization: state-of-the-art. SIAM series: Proceedings in applied mathematics, vol 80, pp 189–208

Wujek B, Renaud J, Batill S, Brockman J (1996) Concurrent subspace optimization using design variable sharing in a distributed computing environment. Concurr Eng 4(4):361–377

Yokoyama N, Suzuki S, Tsuchiya T, Taguchi H, Kanda T (2005) Multidisciplinary Design Optimization of SSTO space plane considering rigid body characteristics. In: 42nd AIAA aerospace sciences meeting and exhibit. Reno, Nevada, USA

Yokoyama N, Suzuki S, Tsuchiya T, Taguchi H, Kanda T (2007) Multidisciplinary Design Optimization of SSTO space plane considering rigid body characteristics. J Spacecr Rockets 44(1):121–131

Zitzler E (1999) Evolutionary algorithms for multi-objective optimization: methods and applications. Ph.D. thesis, Department of Computer Science, Swiss Federal Institute of Technology (ETH), Zurich