

Multidisciplinary design optimisation methods for automotive structures

R.D. Bäckryd¹, A.-B. Ryberg^{1,2*} and L. Nilsson¹

¹Division of Solid Mechanics, Linköping University, SE-581 83 Linköping, Sweden

²Combitech AB, Åkerssjövägen 8, SE-461 53 Trollhättan, Sweden

*E-mail: ann-britt.ryberg@combitech.se

ABSTRACT

Multidisciplinary design optimisation (MDO) can be used as an effective tool to improve the design of automotive structures. Large-scale MDO problems typically involve several groups who must work concurrently and autonomously in order to make the solution process efficient. In this article, the formulations of existing MDO methods are compared and their suitability is assessed in relation to the characteristics of automotive structural applications. Both multi-level and single-level optimisation methods are considered. Multi-level optimisation methods distribute the design process but are complex. When optimising automotive structures, metamodels are often required to relieve the computational burden of detailed simulation models. The metamodels can be created by individual groups prior to the optimisation process, and thus offer a way of distributing work. Therefore, it is concluded that a single-level method in combination with metamodels is the most straightforward way of implementing MDO into the development of automotive structures. If the benefits of multi-level optimisation methods, in a special case, are considered to compensate for their drawbacks, analytical target cascading has a number of advantages over collaborative optimisation, but both methods are possible choices.

Keywords: Multidisciplinary design optimisation, single-level optimisation methods, multi-level optimisation methods, automotive structures

INTRODUCTION

Automotive companies are exposed to tough competition and continuously strive to improve their products in order to maintain their position in the market. The aim of multidisciplinary design optimisation (MDO) is to find the best possible design taking into account several disciplines simultaneously [1]. Introducing MDO can thus aid in the search for improved products. The challenge is to find MDO methods suitable for automotive development that fit company organisations and that can be effectively integrated into product development processes [2].

Product development in the automotive industry is largely based on computer aided engineering (CAE). Simulation models are used to evaluate different aspects of the design proposals, and the number of hardware prototypes is kept to a minimum in order to reduce cost and development time. During the product development, several design groups work concurrently and autonomously on different aspects or parts of the car. The aspects or parts cannot be considered to be isolated entities as they mutually influence one another. Therefore, the groups must interact during the development. The term ‘group’ is used here to denote both the administrative unit and a team working with a specific task. Traditionally, the goal of the design process is to meet a certain number of requirements

by repeated parallel development phases with intermediate data synchronisations between the groups. Optimisation is occasionally used by individual groups. Using a traditional approach leads to a feasible design, but, in all probability, not to an optimal one. The goal of MDO is to find the optimal design taking into account two or more disciplines simultaneously using a formalised optimisation methodology. A discipline is an aspect of the product, and different disciplines are typically handled by different groups. Consequently, performing MDO generally involves several groups. Implementing MDO as a standard industrial activity requires problem decomposition so that individual groups can work on the problem concurrently and autonomously, which places restrictions on the choice of method.

Multidisciplinary design optimisation evolved as a new engineering discipline in the area of structural optimisation, mainly within the aerospace industry [3]. Single-level optimisation methods, in which a central optimiser makes all the design decisions, were the first to be developed. Through distributing the analyses, i.e. the evaluations of the simulation models, computational resources could be used in parallel as shown in Figure 1(a). In order to allow groups to work concurrently and be involved in the decision-making process, the optimisation process as such was distributed, giving rise to the multi-level optimisation methods which are illustrated in Figure 1(b).

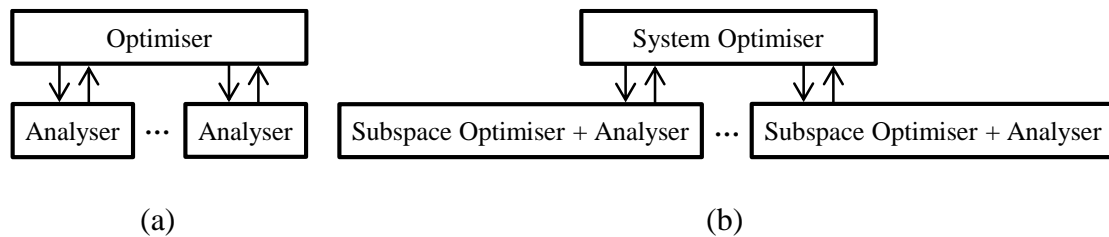


Figure 1. (a) Single-level optimisation method with distributed analyses and (b) Multi-level optimisation method.

Detailed simulation models used to evaluate automotive design proposals are often computationally expensive to run. A large number of evaluations are needed when performing optimisation studies. Direct optimisation, which implies evaluations of the detailed simulation models during the optimisation procedure, may require more computational resources than are available. Metamodel-based design optimisation generally requires fewer evaluations of the detailed simulation models and can therefore be an efficient alternative. Metamodels, which are simplified models of the detailed simulation models, can be evaluated quickly and used as approximations of the detailed models during optimisation studies. They are developed based on a series of evaluations of the detailed models and many different types have been proposed, of which the most common are polynomial regression, Kriging, and different neural network models [4, 5]. Since metamodels are approximations of the detailed simulation models, an extra source of error is introduced and the challenge is to keep this error on a sufficiently low level for the problem at hand.

Multidisciplinary design optimisation studies of automotive structures with full vehicle models are still rather rare. However, one type of such a study found in the literature involves minimising the mass of the vehicle body considering noise, vibration, and harshness (NVH) and crashworthiness. Examples are described by Craig et al. [6], Sobieszcanski-Sobieski et al. [7], and Kodiyalam et al. [8], which are all executed using

metamodels and single-level optimisation methods. There are automotive MDO studies that use direct optimisation [9] but metamodel-based design optimisation is the most common approach when computationally expensive models are involved, as described in the recent article by Rakowska et al. [10] since it is more computationally efficient [11]. Multi-level optimisation methods are rarely used for automotive structural applications although some examples can be found [12, 13].

The aim of this article is to provide a systematic review of existing MDO methods suitable for automotive structural applications. Automotive examples found in the literature can often be comprehended by a single engineer, while the methods looked for in this article must be suitable for large-scale problems involving several groups within a company. The methods also need to fit into existing product development processes. Many of the MDO methods intended for large-scale problems are developed in conjunction with the aerospace industry, and the question is then whether these methods are also suitable for automotive applications. There are studies that review MDO methods for aerospace applications [14] and launch vehicle applications [15]. However, there do not appear to be any studies which focus on suitable MDO methods for automotive applications. Thus, this article intends to fill this gap. Martins and Lambe [16] provide a comprehensive overview of various MDO methods but do not make any specific evaluation related to automotive structural applications. The approach employed in this article is to compare the formulations of different MDO methods and assess their suitability in relation to the specific characteristics of automotive structural MDO problems. The article is thereby intended to give an introduction to methods suitable for MDO of automotive structures and provide engineers with guidance on the selection of methods.

The article is organised as follows. First, an MDO problem is defined and the consequences of problem decomposition are covered. The nature of automotive problems is discussed and compared with the nature of aerospace problems. Next, requirements that need to be fulfilled for a method to be of interest for automotive structural applications are defined and a number of MDO formulations are described. Finally, the various MDO methods are discussed in terms of automotive prerequisites and recommendations are presented.

MATERIALS AND METHODS

Multidisciplinary Design Optimisation Problems

Multidisciplinary design optimisation is a formalised methodology used to perform optimisation of a product considering several disciplines simultaneously. Giesing and Barthelemy [1] provide the following definition of MDO: ‘A methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena.’ In general, a better design can be found when considering the interactions between different aspects of a product than when considering them as isolated entities; something which is taken advantage of when using MDO.

Problem Formulation

A general optimisation problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned} \quad (1)$$

The goal is to find the values of the design variables \mathbf{x} that minimise the objective function

f and fulfil the constraints \mathbf{g} . The problem becomes multidisciplinary if the design variables, objective function, and constraints affect different aspects of the design. To solve an MDO problem, the objective and constraint functions must be evaluated for a number of design variable configurations by an analyser. For automotive structures, the analyser is typically a finite element (FE) model that is computationally costly and time-consuming to evaluate.

Problem Decomposition

Using a single-level optimisation method, the MDO problem can be decomposed by distributing the analyses to subspace analysers. When a multi-level optimisation method is used, the optimisation process as such is distributed to subspace optimisers that communicate with a system optimiser. Even if the decomposition is different for single-level and multi-level optimisation methods, a unified terminology can be used.

Regardless of whether the analyses or the optimisation process are distributed, the connections between the resulting subspaces need to be handled. Each subspace has a number of variables, indicated by the vector \mathbf{x}_j for subspace j . The union of the variables in all subspaces is the original set of design variables \mathbf{x} . The variables in the different subspaces are in general not disjoint. Variables that are unique to a specific subspace are called local variables, denoted by the vector \mathbf{x}_{lj} for subspace j . There will also be a number of shared variables. \mathbf{x}_{sj} indicates the vector of shared variables in subspace j , where each component is present in at least one other subspace. The union of shared variables in all subspaces is denoted by \mathbf{x}_s . When a problem is decomposed, it is necessary to handle the couplings between the resulting subspaces. Coupling variables are defined as output from one subspace needed as input to another subspace. The notion of coupled subspaces indicates the presence of coupling variables. The vector \mathbf{y}_{ij} consists of output from subspace j that is input into subspace i and the collection of all coupling variables is indicated by the vector \mathbf{y} . Decomposition of a system into three subspaces is illustrated in Figure 2. The local, shared, and coupling variables are given as input to each subspace. The output from the subspaces is used by the optimiser or system optimiser to solve and coordinate the MDO problem.

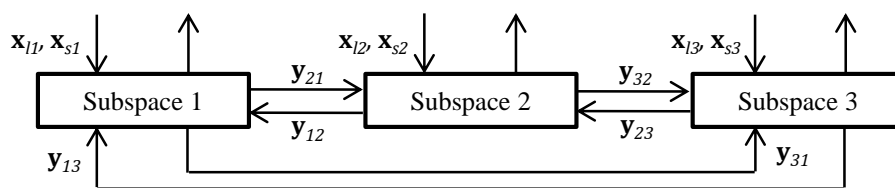


Figure 2. Decomposition of a system into three subspaces.

The presence of coupling variables complicates the problem considerably. Consistency of coupling variables means that the input \mathbf{y}_{ij} to subspace i is the same as the output \mathbf{y}_{ij} from subspace j , and it is obtained using an iterative approach. This is referred to as multidisciplinary feasibility by Cramer et al. [17]. However, since feasibility in an optimisation context refers to a solution that fulfils the constraints, the term multidisciplinary consistency is used here. The optimisation problem in Eq. (1) will now be reformulated using the terminology introduced. The objective function is assumed to be computed from a combination of data from the different subspaces, where f_j denotes the data from subspace j . Furthermore, the constraints are assumed to be separable, which means that each constraint belongs only to one subspace. These assumptions hold

throughout this article. The resulting optimisation formulation is then:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(f_1(\mathbf{x}_{l1}, \mathbf{x}_{s1}, \mathbf{y}_{1*}), f_2(\mathbf{x}_{l2}, \mathbf{x}_{s2}, \mathbf{y}_{2*}), \dots, f_n(\mathbf{x}_{ln}, \mathbf{x}_{sn}, \mathbf{y}_{n*})) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{x}_{lj}, \mathbf{x}_{sj}, \mathbf{y}_{j*}) \leq \mathbf{0}, j = 1, 2, \dots, n, \end{aligned} \quad (2)$$

where \mathbf{y}_{j*} denotes all coupling variables input to subspace j .

A system can be decomposed in different ways [18]. Aspect-based decomposition refers to dividing the system into different disciplines. The system will then naturally consist of two levels: one top level and one for all the disciplines. Object-based decomposition simply means dividing the entire system into its constituent subsystems, which can, in turn, be divided into smaller subsystems or components. A system decomposed by object can have an arbitrary number of levels. Examples of aspect-based and object-based decomposition for automotive problems will be given in the next section.

Automotive Problems

Automotive development is performed by several groups in a company. Some groups are responsible for designing different parts of the product, for example, the body, the interior, and the chassis system. In addition, other groups are responsible for different performance aspects, for example, safety, NVH, and aerodynamics. The natural way of decomposing the MDO problem is then both object-based and aspect-based as illustrated in Figure 3. A product development process typically contains parallel development phases with intermediate synchronisations between the groups. The synchronisations have traditionally been performed through interaction between the groups with the goal of fulfilling all requirements. When introducing MDO, the groups working with different parts or aspects of the product must retain their autonomy and be able to work in parallel in order to effectively use the resources available. Moreover, MDO must fit into the product development process, replacing the traditional interaction between groups with a formalised methodology in order to find the optimum design.

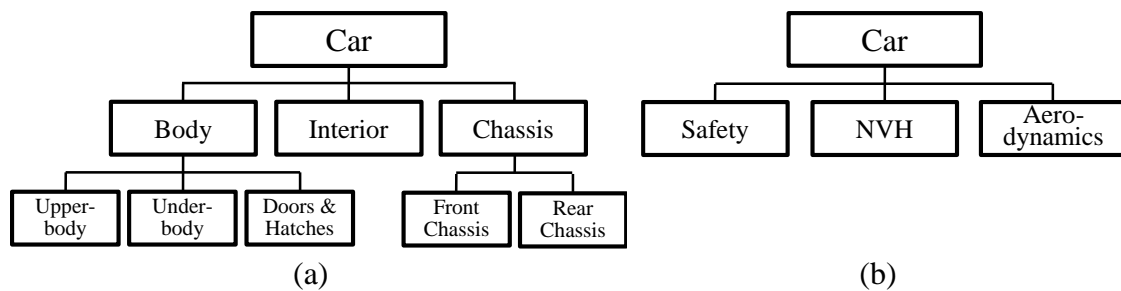


Figure 3. Examples of (a) object-based decomposition and (b) aspect-based decomposition for automotive MDO problems.

Simulation models are used to evaluate design proposals. In many cases, these models are detailed finite element models that are used to evaluate structural aspects, such as crashworthiness and stiffness, for the full vehicle or parts of the vehicle. The design of an automotive structure is to a large extent governed by crashworthiness requirements and these are therefore usually included in MDO studies. Crash scenarios are non-linear by nature and demand extensive computational resources. It is not unusual for a full vehicle crash model to consist of several million elements. Although computers have become much faster over the years, the level of detail of the models has also increased. A

crashworthiness simulation therefore still takes many hours to run, even on a high-performance computing cluster. When performing optimisation studies, the values of the objective and constraint functions need to be evaluated for a large number of design variable settings. Metamodel-based design optimisation can be an efficient approach since it generally requires fewer evaluations of the detailed simulation models than direct optimisation. Using metamodels is an inexpensive way to evaluate different design variable settings during the optimisation procedure. In addition to being computationally expensive, non-linear simulation models produce complex responses resulting in optimisation problems with several local optima. Moreover, gradient information may be unavailable or spurious. Stochastic optimisation algorithms, such as genetic algorithms and simulated annealing, are then suitable when searching for the global optimum. These algorithms do not use gradients but require many evaluations in order to find the optimum, and the need for metamodels therefore becomes even more obvious. A positive spin-off of using metamodels is that they can serve as a filter for the numerically noisy responses that are typical for crash scenarios.

The automotive subspaces are typically linked by shared variables but there are not necessarily coupling variables to be taken into account. Agte et al. [3] state that automotive designs are created in a multi-attribute environment rather than in a truly multidisciplinary environment, and that aspects, such as NVH and crashworthiness, are coupled only by shared variables. This observation is confirmed by the weight optimisation studies mentioned in the introduction. There are examples of coupled automotive disciplines but they do not govern the development of automotive structures. The absence of coupling variables simplifies the solution process of an MDO problem considerably. In addition, it is easier to incorporate metamodels in the optimisation procedure when coupling variables are not considered.

Comparison Between Automotive and Aerospace Problems

As mentioned before, MDO has its roots within the aerospace industry where several MDO methods have been developed for aerospace applications. In order to assess their suitability for automotive structural applications, this section contains a comparison between aerospace and automotive MDO problems. There are many similarities between the problem types. Both the automotive and aerospace industries design complex products, which require the joint effort of many skilled people with expert knowledge from many different areas of the organisation. The need for individual groups to work autonomously and concurrently is therefore obvious in both industries. However, there are also some key differences that become important when dealing with MDO.

Most aeroplane structures are dimensioned for the loads applied during everyday use multiplied by a safety factor, and there is a strong focus on fatigue. Although there is considerable movement of the wings during flight, the stresses are maintained within the elastic region. The structural analyses are therefore linear and associated with relatively low computational costs. The aerodynamic properties are fundamental when developing aeroplanes. Aerodynamic analyses are non-linear and computationally costly. However, in general, the analyses consider stationary states and the responses are usually smooth. Usable gradients can therefore be obtained without difficulty. The use of gradient-based optimisation algorithms, which typically require fewer evaluations than the stochastic optimisation algorithms, is therefore a natural approach. Even though aerodynamic analyses are expensive, direct optimisation may be affordable when using gradient-based algorithms. The need for metamodel-based design optimisation is therefore not as obvious for aerospace applications as it is for automotive applications.

Another difference between automotive and aerospace problems is the connection between the disciplines. Aerospace problems are in general linked by both shared and coupling variables. An example of aerospace coupling variables can be found when considering the structural and aerodynamic disciplines. The slender shapes of aeroplane wings result in structural deformations induced by the aerodynamic forces. These deformations in turn affect the aerodynamics of the structure and hence the aerodynamic forces. The structural and aerodynamic disciplines are thus coupled. Similar strong coupling is not present in the same automotive disciplines or in other disciplines typically involved when performing optimisation of automotive structures. It should be easier to incorporate MDO into automotive development than into aerospace development since less complex methods can be used when coupling variables are not present.

Multidisciplinary Design Optimisation Methods

The aim of this section is to define a number of requirements that must be fulfilled for an MDO method to be useful for automotive structures, and to describe a number of MDO formulations found in the literature. The focus will be on methods that fulfil the requirements defined, but a couple of other methods will also be covered briefly.

Requirements for Automotive Applications

The characteristics of automotive MDO problems have been discussed in a previous section and a list of requirements imposed on an MDO method for automotive structural applications will now be given. First of all, performing large-scale MDO of automotive structures involves several groups within a company. To solve the problem efficiently, the MDO method must allow individual groups to work on the problem concurrently and autonomously. Concurrency means that human and computational resources can be used in parallel, and autonomy means that groups can make design decisions and govern methods and tools. Another issue, also related to efficiency, is the handling of the computationally expensive simulation models that are frequently included in automotive MDO studies. In order to reduce the required computational effort, the MDO method must be able to integrate the use of metamodels. Furthermore, coupling variables are seldom present in MDO of automotive structures. This fact simplifies the solution process considerably and should be taken advantage of. The requirements imposed on an MDO method for automotive structural applications can hence be summarised as follows:

- i) It should allow groups to work on the problem concurrently and autonomously.
- ii) It should allow the use of metamodels.
- iii) It does not need to handle coupled disciplines, i.e. coupling variables.

Some of the more well-known MDO formulations will now be assessed in relation to these defined requirements in order to find the methods best suited for MDO of automotive structures.

Single-Level Optimisation Methods

Single-level optimisation methods has a central optimiser that makes all design decisions. If the analyses are distributed, computer resources can be used in parallel, and individual groups can govern methods and tools for performing the analyses. The most common and basic single-level approach is the multidisciplinary feasible (MDF) formulation [17]. In the MDF formulation, the optimiser requests the values of the objective and constraint functions for different sets of design variables from a system analyser. The system

analyser enforces multidisciplinary consistency from the subspace analysers, i.e. finds a consistent set of coupling variables for each set of design variables. This is typically done iteratively using fixed-point iteration or Newton’s method [19].

The individual discipline feasible (IDF) formulation is an alternative single-level approach proposed by Cramer et al. [17], where multidisciplinary consistency is only enforced at optimum. There are two copies of each coupling variable; one that is controlled by the optimiser and treated as a design variable, and one that is computed as output from the relevant subspace analyser. The optimiser sends the design variables, including coupling variables, to the subspace analysers. The subspace analysers return updated coupling variables and contributions to the global objective and constraint functions to the optimiser. An additional constraint is introduced for each coupling variable to drive the optimisation process towards multidisciplinary consistency at optimum. In this approach, the subspace analysers are decoupled and the iterative process needed to find a multidisciplinary consistent design at each call from the optimiser is avoided.

The MDF and the IDF methods differ in the handling of coupling variables, i.e. how multidisciplinary consistency is enforced. However, the two formulations coincide for MDO problems lacking coupling variables. Since one of the requirements imposed on an MDO method for automotive structural applications defined in the previous section was that it need not consider coupling variables, the resulting simplified formulation will be illustrated here. In Figure 4, the optimiser sends the values of the local and shared variables to the subspace analysers. The subspace analysers return contributions to the global objective function and values for the constraint functions to the optimiser. The resulting optimisation problem can thus be formulated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(f_1(\mathbf{x}_{l1}, \mathbf{x}_{s1}), f_2(\mathbf{x}_{l2}, \mathbf{x}_{s2}), \dots, f_n(\mathbf{x}_{ln}, \mathbf{x}_{sn})) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{x}_{lj}, \mathbf{x}_{sj}) \leq \mathbf{0}, j = 1, 2, \dots, n. \end{aligned} \tag{3}$$

This can be compared to the optimisation formulation in Eq. (2). Solving an MDO problem without coupling variables is straightforward when using this approach, and it can be done using any preferred optimisation algorithm.

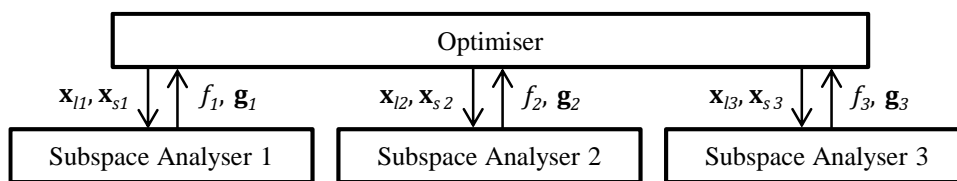


Figure 4. Illustration of a single-level MDO method without coupling variables.

When single-level optimisation methods are used in combination with metamodels, the groups can work in parallel to create the metamodels before the optimisation starts. The previously defined requirements for an MDO method are then fulfilled, with the exception that autonomy is only partly achieved for the groups. The groups can govern methods and tools used to perform the analyses but they have no direct control over the design decisions. It will later be discussed how this drawback can be relieved.

Multi-level Optimisation Methods

Single-level optimisation methods have a central optimiser making all design decisions. Distribution of the decision-making process is enabled using multi-level optimisation methods, where a system optimiser communicates with a number of subspace optimisers.

Individual groups can then work on the MDO problem concurrently and govern methods and tools for their analyses and their part of the optimisation process. Several multi-level optimisation formulations have been presented in the literature, and some of the most well-known ones are investigated here. Two of the methods fulfil the requirements defined previously and will be discussed in more detail, while two other methods will only be covered briefly. The focus is on the formulations and not on how to solve the resulting optimisation problems.

Collaborative optimisation

Collaborative optimisation (CO) is a bi-level method developed at Stanford University. An early description of CO was published by Kroo et al. [20] and further refined by Braun [21]. The method can handle coupling variables, but a simplified variant without coupling variables is presented here.

In CO, the system optimiser is in charge of target values of the shared variables. The subspaces have the freedom to change the local variables and local copies of the shared variables during the optimisation process. The target values of the shared variables are called \mathbf{x}_s^+ and the local copies are denoted \mathbf{x}_s . The variables corresponding to subspace j are called \mathbf{x}_{sj}^+ and \mathbf{x}_{sj} , respectively. A consistent design is enforced at optimum, i.e. the local copies converge towards the target values. An overview of the formulation is illustrated in Figure 5. The system optimiser minimises the global objective function subject to constraints that ensure a consistent design. The subspace optimisers minimise the deviation from consistency subject to local constraints.

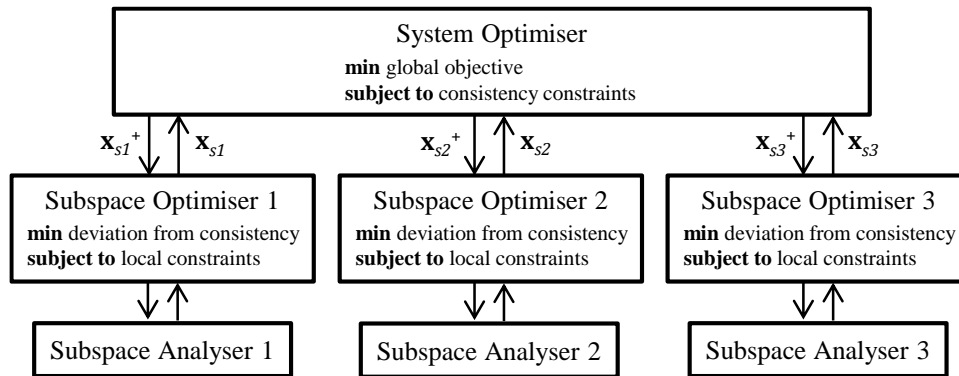


Figure 5. Overview of the collaborative optimisation formulation.

The system optimisation problem is formulated as:

$$\begin{aligned} \min_{\mathbf{x}_s^+} \quad & f(\mathbf{x}_s^+) \\ \text{subject to} \quad & J_j = \|\mathbf{x}_{sj}^+ - \mathbf{x}_{sj}\|_2^2 = 0, j = 1, 2, \dots, n, \end{aligned} \tag{4}$$

where n is the number of subspaces. The system optimiser minimises the global objective function f with respect to the target values of the shared variables \mathbf{x}_s^+ . There is one constraint for each subspace, ensuring that the local copies \mathbf{x}_{sj} match the target values \mathbf{x}_{sj}^+ . The j^{th} subspace optimisation problem is formulated as:

$$\begin{aligned} \min_{\mathbf{x}_{lj}, \mathbf{x}_{sj}} \quad & J_j = \|\mathbf{x}_{sj}^+ - \mathbf{x}_{sj}\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{x}_{lj}, \mathbf{x}_{sj}) \leq \mathbf{0}. \end{aligned} \quad (5)$$

The subspace optimisers minimise the deviations between the local copies \mathbf{x}_{sj} and the corresponding target values \mathbf{x}_{sj}^+ subject to local constraints. The optimisation is performed with respect to local variables and to local copies of the shared variables. In this formulation, there are no local variables in the system objective function. However, the system optimisation problem can include local variables by also making copies of these and enforce consistency by including them in J_j .

There are a number of numerical problems associated with CO when used in combination with gradient-based algorithms, mainly related to the formulation of the system constraint functions and the subspace objective functions [22, 23]. These issues hinder convergence proofs and have an unfavourable effect on the convergence rate, leading to difficulties for conventional non-linear programming algorithms. A number of attempts to modify the CO formulation in order to overcome these problems are documented in the literature. Braun et al. [24] suggested that the system constraints should be stated as inequalities, i.e. $J_j \leq 0$ for all j . This adjustment can improve convergence when a gradient-based solution algorithm that linearises constraints is used. Sobieski and Kroo [25] introduce the use of polynomial surrogate models to represent the subspace objective functions, which are also the system constraints. In modified collaborative optimisation (MCO), presented by DeMiguel and Murray [22], the L_1 -norm is indirectly used instead of the square of the L_2 -norm in the subspace objective functions, and the system problem is made unconstrained by adding these as penalty terms to the system objective function. Both collaborative optimisation with surrogate models and MCO solve some, but not all, of the numerical issues associated with the original formulation. However, these specific numerical problems only arise when gradient-based optimisation algorithms are used and, genetic algorithms have successfully been applied to solve the system problem of CO [26].

Enhanced collaborative optimisation (ECO) was developed by Rooth [27]. The formulation is influenced by CO, MCO, and analytical target cascading that will be described in the next section. An overview of ECO can be found in Figure 6. The goal of the system optimiser is to find a consistent design. There are no constraints on the system level, which makes the system optimisation problem straightforward to solve. The objective functions of the subspaces contain the global objective in addition to terms for the deviation from consistency. It is intuitively more appealing for the subspaces to work towards minimising a global objective, rather than to minimise a deviation from consistency as is done in the original CO formulation. The subspaces are subject to local constraints as well as to linearised versions of the constraints in the other subspaces. The inclusion of the latter constraints provides a direct understanding of the preferences of the other subspaces, unlike CO, where this knowledge is only obtained indirectly from the system optimiser. All the numerical problems that are associated with CO in combination with gradient-based algorithms are resolved using ECO. However, the complexity of ECO is a major drawback.

Collaborative optimisation has mainly been used for aerospace applications, for example, aircraft preliminary design [20] and launch vehicles [28]. However, in a recent article, Xue et al. [13] use CO for weight reduction of a car. The variants of CO presented in this section fulfil the requirements imposed on an MDO method for automotive applications defined previously.

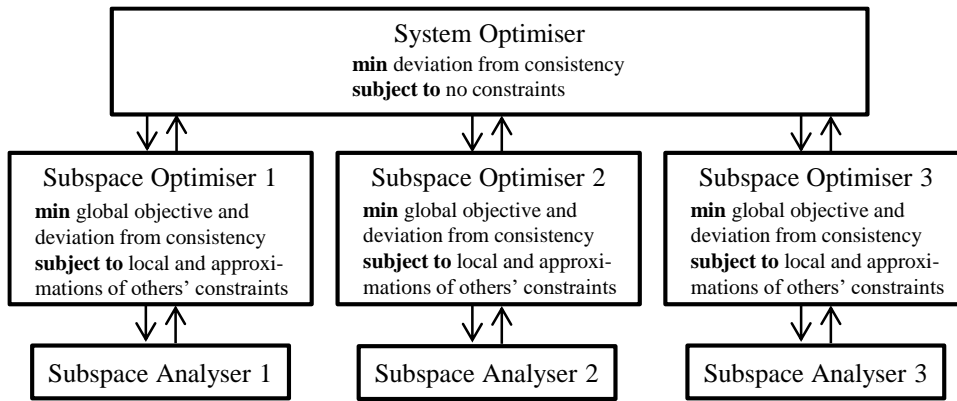


Figure 6. Overview of the enhanced collaborative optimisation formulation.

Analytical target cascading

Analytical target cascading (ATC) was developed for automotive applications at the University of Michigan [29-31]. It was originally intended as an early product development tool for propagating targets on an entire system to targets on smaller parts of the system. The system was therefore decomposed by object into a hierarchical structure. The term subspace has previously been used to designate the second level in a bi-level method. Here, subspace will instead be used as the general term for an element in the hierarchical ATC structure. Each subspace models a part of the system using an analyser and can be connected to subspaces above and below. The objective of each subspace is to meet targets set by the subspace above as closely as possible in addition to striving for a consistent design. The ATC process uses simplified models of the subspaces. After having propagated the targets, the subspaces can be designed independently using detailed models to meet the specified targets.

Analytical target cascading can be used as a multi-level optimisation method if the targets are unattainable. The formulation presented here is adapted from Tosserams et al. [32], which fits better into a general MDO framework than the original formulation. It will only be described for two levels without considering coupling variables. The top subspace optimiser is called system optimiser and the bottom subspace optimisers are denoted subsystem optimisers. In this variant of ATC, the global objective function is assumed to be additively separable and distributed to the subspace optimisers. The objective function of each subspace optimiser consists of two terms; one that is a part of the global objective function and one that ensures consistency. Each subspace has local variables and local constraints. Shared variables are handled in a fashion similar to CO; the system optimiser has target values of the shared variables and the subsystem optimisers have local copies of the shared variables. An illustration of the method can be found in Figure 7.

The system optimisation problem is formulated as:

$$\begin{aligned} \min_{\mathbf{x}_{l0}, \mathbf{x}_s^+} \quad & f_0(\mathbf{x}_{l0}, \mathbf{x}_s^+) + \sum_{j=1}^n \|\mathbf{w}_j \circ (\mathbf{x}_{sj}^+ - \mathbf{x}_{sj})\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{x}_{l0}, \mathbf{x}_s^+) \leq \mathbf{0}, \end{aligned} \quad (6)$$

where \mathbf{x}_{l0} is the vector of local variables, \mathbf{g}_0 is the vector of local constraints, $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$ is a vector of penalty weights, and the symbol \circ denotes a term-by-term

multiplication of vectors. The j^{th} subsystem optimisation problem is formulated as:

$$\begin{aligned} \min_{\mathbf{x}_{lj}, \mathbf{x}_{sj}} \quad & f_j(\mathbf{x}_{lj}, \mathbf{x}_{sj}) + \|\mathbf{w}_j \circ (\mathbf{x}_{sj}^+ - \mathbf{x}_{sj})\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{x}_{lj}, \mathbf{x}_{sj}) \leq \mathbf{0}, \end{aligned} \tag{7}$$

where the notation has been previously defined. Consistency is obtained by including the square of the L_2 -norm of the deviation from consistency multiplied by penalty weights in the subspace objective functions.

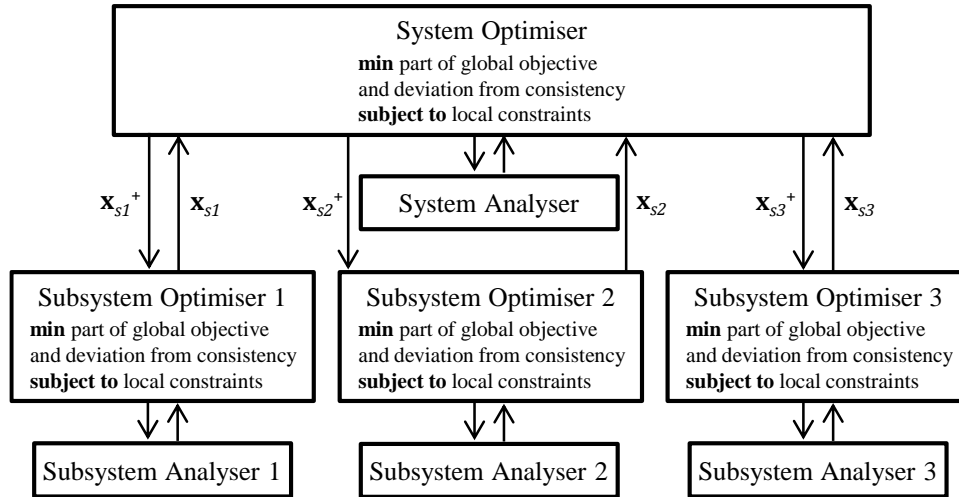


Figure 7. Overview of the analytical target cascading formulation.

A convergence proof for solving the original ATC problem is provided by Michelena et al. [33]. They show that the solution of the ATC problem is identical to the solution of the original undecomposed problem. This is also valid for the variant of ATC presented here if f_j is convex in all subspaces. It is important to set the penalty weights appropriately if the targets are unattainable, since too small weights can yield solutions far from the solution of the original problem while too large weights can cause numerical problems [34]. Quadratic penalty functions are used to ensure consistency in the original formulation and in the formulation presented here. Other types of penalty functions can be found in the literature, see for example Tosserams et al. [32] who use an augmented Lagrangian penalty function.

Analytical target cascading has been used for automotive applications, for example to propagate targets in a chassis design of a sport-utility vehicle [35], to redesign a U.S. class VI truck [36], and more recently, to optimise a hybrid electrical fuel cell vehicle [37] and two commercial vehicle systems [13]. The method has also been used for aerospace applications. Tosserams et al. [38] use ATC to minimise the weight of a supersonic business jet. Analytical target cascading can be used in combination with metamodels for solving an MDO problem without coupling variables, and it fulfils the requirements imposed on an MDO method for automotive structural applications presented previously.

Other methods

Other proposed methods for aerospace applications are concurrent subspace optimisation

(CSSO) and bi-level integrated system synthesis (BLISS). These methods will be covered only briefly, as neither of them is suitable for the type of automotive MDO problems considered in this article.

Concurrent subspace optimisation was introduced by Sobieszczanski-Sobieski [39], but it has diverged into fundamentally different variants. The version presented here was developed by Renaud and Gabriele [40-42] in a series of articles. Many subsequent approaches are based on their work. The main idea is to distribute each shared variable to the subspace for which it influences the objective and constraint functions the most. Each subspace is then optimised with respect to its local variables and a subset of its shared variables, while all other variables are held constant. Thereafter, optimisation of an approximation of the global problem is performed around the combination of optimised variables from the different subspaces. All variables are consequently dealt with at the system level. One of the main motivations for using a multi-level method is to give the individual groups freedom to make their own design decisions. When using the variant of CSSO presented here, the groups will not have this freedom, and the increased complexity from introducing a multi-level method is not motivated.

Bi-level integrated system synthesis was introduced by Sobieszczanski-Sobieski et al. [43]. A reformulation referred to in the literature as BLISS 2000, or simply BLISS, was presented by Sobieszczanski-Sobieski et al. [44]. In this approach, the subspace optimisers control the local variables and the system optimiser the shared variables. The key concept in BLISS 2000 is the use of surrogate models to represent optimised subsystems. To create these surrogate models, a design of experiment is created and a number of subspace optimisation problems are solved with respect to the local variables. In each subspace optimisation problem, the sum of the coupling variables output from that specific subspace multiplied by weighting coefficients is minimised subject to local constraints. The surrogate models represent the coupling variables output from each subspace as functions of the shared variables, the coupling variables input to that subspace, and the weighting coefficients. Polynomial surrogate models are used in the original version of BLISS 2000 but Kriging surrogate models have also been used [45]. In principle, each subspace could be given the freedom to choose their own surrogate model. The system optimiser uses the surrogate models to minimise the global objective subject to consistency constraints. The original implementation of BLISS 2000 concerns four coupled subspaces of a supersonic business jet: structures, aerodynamics, propulsion, and performance. Few other applications have been found in the literature. The method has been developed to handle coupled disciplines and is not relevant for problems lacking coupling variables since the subspace objective functions then vanish.

RESULTS AND DISCUSSION

In this section, the choice between single-level and multi-level optimisation methods will first be assessed in relation to the characteristics of automotive and aerospace MDO problems described previously. When solving large-scale automotive and aerospace MDO problems, it is important that the groups involved can work concurrently and autonomously. For automotive structural applications, this can partly be achieved by the use of metamodels. Having the possibility to incorporate metamodels is essential and is mainly motivated by the high computational cost of many of the detailed simulations. When performing metamodel-based design optimisation, the groups can run the simulations needed, build the metamodels, and check the accuracy of the metamodels before the optimisation process starts, using their preferred methods and tools. The need for

metamodels is not as obvious for aerospace applications. Multi-level optimisation methods are therefore required in order to allow individual groups to work concurrently and autonomously.

The benefits and drawbacks of using single-level and multi-level optimisation methods in combination with metamodels for automotive structural applications will now be assessed. The most straightforward and simple approach to perform MDO of automotive structures is to use a single-level optimisation method in combination with metamodels, as described by Ryberg et al. [46]. Each design group can then be responsible for its own metamodels. The metamodels are sent to a central optimiser that finds the optimum design. All design decisions will be taken on a central level and the groups are therefore not autonomous in this sense. However, this drawback can be relieved by involving the different groups in the setup of the optimisation problem and in the assessment of the results. Another disadvantage of using a single-level optimisation method is that individual groups do not govern optimisation methods and tools. Furthermore, groups that have inexpensive simulations either have to create metamodels, which introduce an unnecessary source of error, or let the central optimiser call their analysers directly, and give up even more of their autonomy. When it comes to the choice of single-level optimisation method, the multidisciplinary feasible and individual discipline feasible methods coincide when there are no coupling variables.

One of the main motivations for using a multi-level optimisation method is to involve the different groups directly in the decision-making process. A multi-level method also allows the groups to govern their own optimisation procedure, including the choice of optimisation methods and tools. The groups can then work concurrently and more autonomously than when using a single-level optimisation method. Metamodels can be used, but groups who use inexpensive simulation models have the option to use direct optimisation instead. The main drawback of multi-level optimisation methods is their complexity. The methods are often complicated to implement and can also be complex to use. The multi-level formulation can become less transparent than the corresponding single-level formulation. One example of this is when the local objective functions do not mirror the global one, which makes it difficult for the individual groups to grasp what they are aiming for during the optimisation process. A comparison of benefits and drawbacks using single-level and multi-level optimisation methods in combination with metamodels is summarised in Table 1.

Table 1. Comparison of benefits and drawbacks using single- and multi-level optimisation methods in combination with metamodels for automotive structural applications.

	Single-Level Methods	Multi-Level Methods
Bene- fits	Simple	Autonomy to make some design decisions Autonomy to govern methods and tools for optimisation
Draw- backs	No autonomy to make design decisions No autonomy to govern methods and tools for optimisation	Complex

Collaborative optimization and analytical target cascading are two multi-level formulations that fulfil the requirements imposed on an MDO method for automotive structural purposes defined in this article. A disadvantage of CO is the lack of transparency for the groups. The purpose of the subspace optimisation problems is to

minimise the deviation from a consistent design, and it is only implicitly through the system optimiser that the global objective is minimised. This can be compared to ATC, where each subspace is given a part of the global objective function and therefore has an understanding of the global goal. Furthermore, CO suffers from a number of numerical problems when used in combination with gradient-based optimisation algorithms, while ATC does not have these shortcomings. Several improvements to CO have been presented in the literature. Enhanced collaborative optimisation resolves the numerical issues and also gives the groups insight into the global objective, but it has the drawback of being complicated. The numerical problems associated with CO will not appear when stochastic optimisation algorithms, which are suitable when performing metamodel-based MDO of automotive structures are used. Another difference between the two formulations concerns the role of the system optimiser. It only coordinates the solution process in CO, while it is also responsible for a part of the MDO problem and has an analyser to evaluate different design variable settings in ATC. Finally, ATC can be used to solve an MDO problem with multiple levels, while CO is restricted to two levels. Both CO and ATC are possible choices of MDO methods for the automotive structural purposes even though ATC has some advantages. However, the performances of the two methods in combination with stochastic optimisation algorithms need to be further investigated.

CONCLUSIONS

In this article, a systematic review of existing MDO methods for automotive structural applications has been presented and guidance on the selection of suitable methods has been provided. Large-scale MDO problems within the automotive industry involve several groups which must work concurrently and autonomously. Furthermore, primarily due to the expense of many of the simulation models used in automotive development, it must be possible to incorporate metamodels. Single-level optimisation methods use a central optimiser to solve the MDO problem. Multi-level optimisation methods are introduced to distribute the decision-making process and the control over optimisation methods and tools. They are more complex than single-level methods, and if they are to be justified, the benefits of using them must compensate for their complexity. Single-level optimisation methods become more attractive in combination with metamodels, as individual groups can work concurrently and autonomously to create the metamodels before the optimisation process starts. In addition, the issue of not participating in design decisions can partly be compensated by involving the different groups in the setup of the optimisation problem and in the assessment of the results. Therefore, it is concluded that a single-level method in combination with the use of metamodels is often the most convenient way of solving non-coupled MDO problems for automotive structural applications involving computationally expensive simulations. If the benefits of multi-level optimisation methods are considered to compensate for their drawbacks, analytical target cascading has a number of advantages over collaborative optimisation, although both methods are possible choices. Further studies comparing the discussed methods on large-scale industrial problems with different characteristics will reveal more details on the efficiency and suitability of the different MDO methods in relation to automotive structural applications.

ACKNOWLEDGEMENTS

The work presented in this article has been carried out with financial support from the Swedish Foundation for Strategic Research through the ProViking project 'ProOpt'

(PV09-0006) and Sweden's Innovation Agency Vinnova through the FFI project 'Robust and multidisciplinary optimization of automotive structures' (2009-00314).

REFERENCES

- [1] Giesing JP, Barthelemy J-FM. A summary of industry MDO applications and needs. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 1998.
- [2] Belie R. Non-technical barriers to multidisciplinary optimization in the aerospace industry. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 2002.
- [3] Agte J, de Weck O, Sobieszczanski-Sobieski J, Arendsen P, Morris A, Spieck M. MDO: assessment and direction for advancement—an opinion of one international group. *Structural and Multidisciplinary Optimization*. 2010;40:17-33.
- [4] Simpson TW, Peplinski J, Koch PN, Allen JK. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*. 2001;17:129-50.
- [5] Ryberg A-B, Bäckryd RD, Nilsson L. Metamodel-based multidisciplinary design optimization for automotive applications: Linköping University; 2012.
- [6] Craig KJ, Stander N, Dooge D, Varadappa S. MDO of automotive vehicle for crashworthiness and NVH using response surface methods, AIAA paper 2002-5607. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 2002. p. 1930-40.
- [7] Sobieszczanski-Sobieski J, Kodiyalam S, Yang RY. Optimization of car body under constraints of noise, vibration, and harshness (NVH) and crash. *Structural and Multidisciplinary Optimization*. 2001; 22,(4):295–306.
- [8] Kodiyalam S, Yang RJ, Gu L, Tho C-H. Multidisciplinary design optimization of a vehicle system in a scalable, high performance computing environment. *Structural and Multidisciplinary Optimization*. 2004;26:256-63.
- [9] Duddeck F. Multidisciplinary optimization of car bodies. *Structural and Multidisciplinary Optimization*. 2008;35:375-89.
- [10] Rakowska J, Chator A, Barthelemy B, Lee M, Morgans S, Laya J, et al. An iterative application of multi-disciplinary optimization for vehicle body weight reduction based on 2015 Mustang product development. *SAE International Journal of Materials and Manufacturing*. 2015;8:685-92.
- [11] Sheldon A, Helwig E, Cho Y-B. Investigation and application of multi-disciplinary optimization for automotive body-in-white development. 8th European LS-DYNA Users Conference; 2011.
- [12] Kang N, Kokkolaras M, Papalambros PY, Yoo S, Na W, Park J, et al. Optimal design of commercial vehicle systems using analytical target cascading. *Structural and Multidisciplinary Optimization*. 2014;50:1103-14.
- [13] Xue Z, Elango A, Fang J. Multidisciplinary design optimization of vehicle weight reduction. *SAE International Journal of Materials and Manufacturing*. 2016;9:393-9.
- [14] Sobieszczanski-Sobieski J, Haftka RT. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*. 1997;14:1-23.

- [15] Balesdent M, Bérend N, Dépincé P, Chriette A. A survey of multidisciplinary design optimization methods in launch vehicle design. *Structural and Multidisciplinary Optimization*. 2012;45:619-42.
- [16] Martins JRR, Lambe AB. Multidisciplinary design optimization: a survey of architectures. *AIAA Journal*. 2013;51:2049-75.
- [17] Cramer EJ, Dennis J, John E, Frank PD, Lewis RM, Shubin GR. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*. 1994;4:754-76.
- [18] Sobieszczanski-Sobieski J, Haftka RT. Interdisciplinary and multilevel optimum design. *Computer aided optimal design: Structural and mechanical systems*: Springer; 1987. p. 655-701.
- [19] Balling RJ, Sobieszczanski-Sobieski J. Optimization of coupled systems- A critical overview of approaches, AIAA paper 94-4330. 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 1994. p. 753-73.
- [20] Kroo I, Altus S, Braun R, Gage P, Sobieski I. Multidisciplinary optimization methods for aircraft preliminary design, AIAA paper 94-4325. 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 1994. p. 697-707.
- [21] Braun RD. Collaborative optimization: an architecture for large-scale distributed design: Stanford University; 1996.
- [22] DeMiguel A-V, Murray W. An analysis of collaborative optimization methods, AIAA paper 2000-4720. 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 2000.
- [23] Alexandrov NM, Lewis RM. Analytical and computational aspects of collaborative optimization for multidisciplinary design. *AIAA Journal*. 2002;40:301-9.
- [24] Braun R, Gage P, Kroo I, Sobieski I. Implementation and performance issues in collaborative optimization, AIAA paper 96-4017. 6th NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 1996. p. 295-305.
- [25] Sobieski IP, Kroo IM. Collaborative optimization using response surface estimation. *AIAA Journal*. 2000;38:1931-8.
- [26] Zadeh PM, Toropov VV, Wood AS. Metamodel-based collaborative optimization framework. *Structural and Multidisciplinary Optimization*. 2009;38:103-15.
- [27] Roth BD. Aircraft family design using enhanced collaborative optimization: Stanford University; 2008.
- [28] Braun RD, Moore AA, Kroo IM. Collaborative approach to launch vehicle design. *Journal of Spacecraft and Rockets*. 1997;34:478-86.
- [29] Kim HM. Target cascading in optimal system design: University of Michigan ; 2001.
- [30] Kim HM, Michelena NF, Papalambros PY, Jiang T. Target cascading in optimal system design. *Journal of Mechanical Design*. 2003;125:474-80.
- [31] Michalek JJ, Papalambros PY. Weights, norms, and notation in analytical target cascading. *Journal of Mechanical Design*. 2005;127:499-501.
- [32] Tosserams S, Etman L, Papalambros P, Rooda J. An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Structural and Multidisciplinary Optimization*. 2006;31:176-89.
- [33] Michelena N, Park H, Papalambros PY. Convergence properties of analytical target cascading. *AIAA journal*. 2003;41:897-905.

- [34] Michalek JJ, Papalambros PY. An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading. *Journal of Mechanical Design*. 2005;127:206-14.
- [35] Kim HM, Rideout DG, Papalambros PY, Stein JL. Analytical target cascading in automotive vehicle design. *Journal of Mechanical Design*. 2003;125:481-9.
- [36] Kim HM, Kokkolaras M, Louca LS, Delagrammatikas GJ, Michelena NF, Filipi ZS, et al. Target cascading in vehicle redesign: a class VI truck study. *International Journal of Vehicle Design*. 2002;29:199-225.
- [37] Han J, Papalambros PY. Optimal design of hybrid electric fuel cell vehicles under uncertainty and enterprise considerations. *Journal of Fuel Cell Science and Technology*. 2010;7:021020-1-9.
- [38] Tosserams S, Kokkolaras M, Etman L, Rooda J. A nonhierarchical formulation of analytical target cascading. *Journal of Mechanical Design*. 2010;132:051002-1-13.
- [39] Sobieszczanski-Sobieski J. Optimization by decomposition: a step from hierarchic to non-hierarchic systems. *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*; 1988. p. 51-78.
- [40] Renaud J, Gabriele G. Sequential global approximation in non-hierarchic system decomposition and optimization. *Advances in Design Automation, 17th Design Automation Conference*; 1991. p. 191-200.
- [41] Renaud J, Gabriele G. Improved coordination in nonhierarchic system optimization. *AIAA Journal*. 1993;31:2367-73.
- [42] Renaud J, Gabriele G. Approximation in nonhierarchic system optimization. *AIAA Journal*. 1994;32:198-205.
- [43] Sobieszczanski-Sobieski J, Agte JS, Sandusky Jr RR. Bi-level integrated system synthesis (BLISS). *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*; 1998. p. 1543-57
- [44] Sobieszczanski-Sobieski J, Altus TD, Phillips M, Sandusky R. Bilevel integrated system synthesis for concurrent and distributed processing. *AIAA Journal*. 2003;41:1996-2003.
- [45] Kim H, Ragon S, Soremekun G, Malone B, Sobieszczanski-Sobieski J. Flexible approximation model approach for bi-level integrated system synthesis, *AIAA paper 2004-4545*. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*; 2004.
- [46] Ryberg A-B, Bäckryd RD, Nilsson L. A metamodel-based multidisciplinary design optimization process for automotive structures. *Engineering with Computers*. 2015;31:711-28.