12th World Congress on Structural and Multidisciplinary Optimisation
05th - 09th, June 2017, Braunschweig, Germany

Convergence Strategy for Parallel Solving of Analytical Target Cascading with Augmented Lagrangian Coordination

Yongsu Jung[1], Namwoo Kang[2], Ikjin Lee[3]

[1] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, yongsu50@kaist.ac.kr
[2] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, nwkang@kaist.ac.kr
[3] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, ikjin.lee@kaist.ac.kr

1. Abstract

Analytical Target Cascading (ATC) is a decomposition-based optimization methodology that partitions a system into subsystems and then coordinates targets and responses among subsystems. Augmented Lagrangian relaxation with Alternating Direction method (AL-AD) has been widely used for the coordination process of both hierarchical ATC and non-hierarchical ATC, and theoretically guarantees convergence under the assumption that all subsystem problems are convex and continuous. One of the main advantages of ATC is that it can solve subsystem problems in parallel, thus allowing it to reduce computational cost by parallel computing. Previous studies have proposed AL coordination strategies for parallelization by eliminating interactions among subproblems. This is done by introducing a master problem and support variables or by approximating a quadratic penalty term to make subproblems separable. However, conventional AL-AD does not guarantee convergence in the case of parallel solving. Our study found that, in parallel solving using targets and responses of the current iteration, conventional AL-AD causes mismatch of information in updating the Lagrange multiplier (LM). Therefore, the LM may not reach the optimal point, and as a result, increasing penalty weight causes numerical difficulty in the AL penalty function approach. To solve this problem, we propose a modified AL-AD for parallel solving in non-hierarchical ATC. The proposed algorithm uses the subgradient method with adaptive step size in updating the LM, which is independent of quadratic penalty terms and keeps quadratic penalty terms at the initial value. Without approximation or introduction of an artificial master problem, the modified AL-AD for parallel solving can achieve similar accuracy and convergence with much less computational cost, compared with conventional methods.

2. Keywords: Multidisciplinary Design Optimization (MDO), Analytical Target Cascading (ATC), Parallelization

3. Introduction

Many studies have been published on distributed coordination methodology for Multidisciplinary Design Optimization (MDO) problems and decomposition-based optimization. In the current paper, we focus on the coordination method of ATC specialized in hierarchical multilevel decomposition. ATC was developed originally for translating system-level target to design specifications for the components, and has been shown to be useful as a coordination method that is hierarchically decomposed [13,14]. Top-level design targets are propagated to the lower levels with consecutive optimization to set the target value to a lower level. Moreover, the response corresponding to target value should be calculated in the lower level, from which the top-level design optimization is revised iteratively. To minimize inconsistency between target and response, in each subproblem, an optimization problem is formulated with consistency constraints. Obviously, distributed design optimization, including ATC, usually requires more computational cost than the AiO (All-in-One) strategy. However, the use of decomposition is typically dictated by its inability to solve the problem as AiO, because the disciplinary design teams use specialized FEA model that have been developed respectively in different module; thus, taking all model in one single optimization problem is impractical or impossible. The main issue of the coordination method of ATC is how to deal with inconsistency between target and response in mathematical formulation. A Quadratic Penalty function (QP) of the system consistency constraints is proposed and developed in the beginning [2]. Here, the convergence proof of ATC with quadratic penalty function is proposed, along with the iterative method for finding minimal penalty weight factor that can satisfy user-specified inconsistency tolerances [11,12]. Although large penalty weight is required to achieve convergence and accuracy, it can cause ill-conditioning of the problem and computational difficulties simultaneously [2]. Furthermore, quadratic penalty function is not separable because of penalty term with the form of the l-2 norm, which has a cross term; therefore, subproblems should be solved sequentially. Unlike the linear term, the cross term affects objective function according to the incoming parameters. An alternative choice for the consistency relaxation function is an Ordinary Lagrangian function (OL) [9]. On the basis of the Lagrangian duality theory, the Lagrange multiplier (LM) of the consistency constraint function is defined, and the LM is updated using subgradient method until convergence without quadratic penalty term. However, without proper assumption, it may give rise to a duality gap, cause instability, and limit application [2]. Moreover, because of the absence of the quadratic penalty term, the objective function can be unbounded. Kim et al. [8] proposed the Lagrangian Coordination for Enhancing the Convergence of Analytical Target Cascading developed based on OL. Enhanced coordination algorithm improves convergence and prevents the subproblems from being unbounded. However, for the same reason as QP, each subproblem becomes dependent and must be solved sequentially. Tosserams et al. [19] proposed the Augmented Lagrangian relaxation using Alternating Direction (AL-AD) at a similar time. AL-AD improves the performance of ATC coordination using Augmented Lagrangian Coordination (ALC) with alternating direction method to eliminate the inner loop convergence criterion. This proposed approach demonstrates both good convergence property and low computational cost [18, 21, 22]. However, it also has an issue of separation, because each problem is dependent on others in the current iteration. After developing AL-AD, some previous studies have proposed ALC strategies for parallelization. We focused our research on the parallelization of ATC coordination. Parallelization means parallel computing of each subproblem. With sequential solving, each coupled subproblem should be solved sequentially in an inner loop. Parallel solving could solve all subproblems simultaneously in an inner loop, so if the available resources are sufficient, the time required will be much less than sequential solving. Most of the proposed methods implement parallelization by eliminating communication, such as target and response. Li et al. [10] use first-order Taylor expansion based on previous iteration point at the quadratic penalty term in AL-AD, demonstrating that the quadratic penalty term is separable by linearizing the cross term. In another study, Tosserams et al. [20,23] proposed a new ATC coordination approach that allows non-hierarchical target response coupling between subproblems, introduces system-wide functions that depend on variables of two or more subproblems, and treats the parallelization problem in non-hierarchical ATC by introducing an artificial master problem. However, introducing a support variable can increase the dimensionality of each subproblem and also constructing an artificial master problem and reformulating the non-hierarchical system to a bi-level system are necessary for parallelization. Solving the non-hierarchical system itself with appropriate penalty function and algorithm loop is more natural and convenient rather than constructing the artificial problem and reformulating the whole system. Furthermore, the use of Taylor expansion is based on approximation and can lead to problems, such as accuracy being dependent on the characteristics of the problem. After proposing a non-hierarchical ATC, some studies on the industrial application [1,6,7] and other penalty functions or updating scheme of LM [24], such as Exponential Penalty Function (EPF) [4] and sequential linearization techniques [5] have been proposed, but not in the parallelization of ATC. In this paper, we propose a modified AL-AD for parallel solving in non-hierarchical ATC. The proposed method is based on AL-AD coordination, but also uses the subgradient update method [3] to find the optimum LM because of delaying information for parallelization. We also propose an adaptive step size update algorithm, which allows setting the proper step size for each outer loop based on the path of the LM, thus improving convergence. The proposed method is formulated based on non-hierarchical ATC formulation [23]. The reason why we deal with only non-hierarchical ATC is that hierarchical ATC is just a special case of non-hierarchical ATC, and in reality, most of the decomposition-based MDO problems are non-hierarchcial. All subproblems can communicate directly with other subproblems in non-hierarchical coordination. Therefore, non-hierarchical ATC is more general and natural, hence the decision to use non-hierarchical ATC.

The article is organized as follows. Existing non-hierarchical formulation with AL-AD is presented in Section 4. In Section 5, the proposed coordination methodology and convergence strategy are presented. In Section 6, the 14-variable non-convex geometric programming optimization problem that is widely used in previous ATC paper, is employed to demonstrate the efficiency and feasibility of proposed methodology. We compare the performances of the modified AL-AD for parallel solving with those of existing coordination methods, with our proposed approach demonstrating good performance.

4. Previous Method: Non-hierarchical ATC with Augmented Lagrangian Relaxation

4.1 Non-hierarchical ATC formulation

The traditional ATC formulation uses a hierarchical decomposition that coordinates the consistency constraints of the target and response between parents and children. The target and corresponding responses are exchanged in the outer loop between parents and children. The basic assumption is that the responses of components from the upper-level depend on the responses of the lower-level but not vice versa. However, Tosserams et al. [23] proposed a non-hierarchical ATC that does not have a hierarchical decomposition structure. Any two subproblems can communicate with each other directly, which is impossible in traditional hierarchcial ATC. In this section, we show the existing formulation of AL-AD for non-hierarchical ATC, which is extended from the traditional formulation. Meanwhile, in the non-hierarchical formulation, the subproblems that communicate with one another are called neighbor subproblems and are distinct from the concept of parents or children in hierarchical ATC. In non-hierarchical ATC, the concept of level can be dropped. Instead, non-hierarchical ATC maintains a double index notation, which denotes the direction of communication. The important point is that the concept of level is no longer needed because of the lack of hierarchy, although the concept of solving the flow still exists. Distinguishing whether the subproblem is solved earlier or later than oneself and preserving the unique target–response coupling structure are necessary. Hence, subproblems that are solved earlier give the target values of the current iteration and the subproblems themselves which, when solved later, give the response value of the previous iteration. This is identified with the superscript i in Figure 1, which describes the flow of information based on optimization of the subproblem *j*. Let T_j be the set of neighbors, for which subproblem *j* receives responses and computes the corresponding target value, and let R_j be the set of neighbors, for which the subproblem *j* receives the target and computes the corresponding responses. Figure 1 illustrates the target–response pairs between subproblem *j* and the two types of subproblems, which are solved earlier or later than subproblem *j*. Therefore, the targets from subproblem *n* make up the results of current iteration so it has superscript i . In contrast, the response from subproblem *m* is a result of the previous iteration so it has superscript $i-1$.

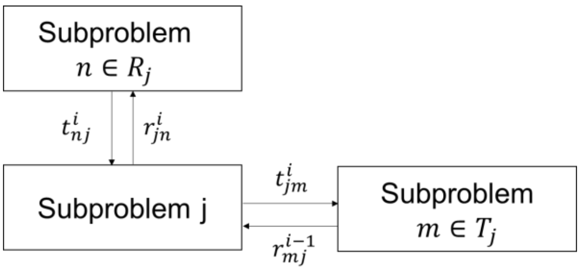

Figure 1: Non-hierarchical target and response flow between subproblem *j* and its neighbors for sequential solving (modified from Tosserams et al. [23])

Hence, the general optimization formulation of subproblem *j* is formulated as

$$\begin{gathered}
\min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j} \theta(\mathbf{t}_{nj}^{(i)} - \mathbf{r}_{jn}^{(i)}) + \sum_{m\in T_j} \theta(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) \\
\text{where } \theta(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \|\mathbf{w}\circ\mathbf{c}\|_2^2 \\
\text{subject to } \mathbf{g}_j(\bar{\mathbf{x}}_j^{(i)}) \le 0 \\
\mathbf{h}_j(\bar{\mathbf{x}}_j^{(i)}) = 0 \\
\text{with } \mathbf{r}_{jn}^{(i)} = \mathbf{S}_{jn}\mathbf{a}_j(\mathbf{x}_j^{(i)}, \mathbf{t}_{jm}^{(i)} \,|\, m \in T_j), \qquad \mathrm{n} \in R_j \\
\bar{\mathbf{x}}_j^{(i)} = [\mathbf{x}_j^{(i)}, \mathbf{r}_{jn}^{(i)} \,|\, n \in R_j, \mathbf{t}_{jm}^{(i)} \,|\, m \in T_j]
\end{gathered} \tag{1}$$

where $\bar{\mathbf{x}}_j$ represents the optimization variables for subproblem *j*, $\mathbf{x}_j$ represents the local design variables, and $\mathbf{t}_{nj}$ represents the targets computed in neighbor subproblem *n* belonging to the set R_j and subproblem *j* return $\mathbf{r}_{jn}$ corresponding to $\mathbf{t}_{nj}$. Similarly, $\mathbf{t}_{jm}$ represents the target computed in subproblem *j* to neighbor subproblem *m* belonging to set T_j and $\mathbf{r}_{mj}$ will be returned in the optimization process of the subproblem *m* corresponding to $\mathbf{t}_{jm}$ .The function f_j is the local objective, and vector functions $\mathbf{g}_j$ and $\mathbf{h}_j$ represents the local inequality and equality constraints, respectively. Function $\mathbf{a}_j$ is used to compute response value, and $\mathbf{S}_{jn}$ is a binary selection matrix that selects components from $\mathbf{a}_j$. The $\circ$ symbol is used to denote a term-by-term multiplication of vectors such that $[a_1, a_2, ..., a_n] \circ [\mathrm{b}_1, \mathrm{b}_2, ..., \mathrm{b}_n] = [a_1 \mathrm{b}_1, a_2 \mathrm{b}_2, ..., a_n \mathrm{b}_n]$ **.** These are used in the common non-hierarchical ATC formulation with the direction of communication. In this paper, we added superscript *i* , which represents the outer loop iteration in the formulation, for the purpose of making a comparison with the proposed coordination.

4.2 Coordination algorithm

The coordination algorithm operates in the inner and the outer loops to achieve convergence of targets and responses, which in turn, determines the optimum of ATC. The basic concept of the inner and outer loops is similar with that of the traditional hierarchical AL-AD [19]. The main purpose of the coordination algorithm in AL-AD using AL function as a penalty function is to find the optimum Lagrange multiplier (LM) associated with the consistency constraints. As mentioned earlier, in the inner loop, the optimization of each subproblem expressed by Eq. (1) is performed with fixed parameters, such as the target from upper-level, the LM and penalty weight given to each subproblem. In the outer loop, the LM is updated in every iteration to perform the new optimization of subproblems in the inner loop with new updated parameters. With the ALC, ill-conditioning of the problem can be avoided by finding the optimum LM instead of selecting penalty weight that is very large for the proper update algorithm. In AL-AD, with alternating direction method, each subproblem is solved only once for the inner loop, instead of solving the iterative inner loop coordination; this process has been shown to reduce the computational cost required for the inner loop. The reader can refer to other papers that treated ALC with coordination algorithm for more detailed descriptions [17,19,20,22]. The important parameters in AL-AD, even if the non-hierarchical ATC is employed, are LM denoted in $\mathbf{v}$ and the penalty weight $\mathbf{w}$. Both determine the performance of ATC, including convergence rate, inconsistency between target and response, and accuracy of the solution. In general, the linear update algorithm employed in the AL penalty method is used for selecting the new LM in AL-AD, as shown in Eq. (2); this is also generally known as a method of multipliers with ALC [2].

$$\mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + 2\mathbf{w}^{(i)} \circ \mathbf{w}^{(i)} \circ (\mathbf{t}^{(i)} - \mathbf{r}^{(i)}) \tag{2}$$

In Eq. (2) above, $\mathbf{c}^{(i)}$ refers to the values of inconsistency between the target and response, which are calculated from the *i* th outer loop iteration. In addition, the penalty weight $\mathbf{w}^{(i)}$ is updated when the reduction of inconsistency constraint value is insufficient, as shown in Eq. (3) [2].

$$\mathbf{w}^{(i+1)} = \begin{cases} \mathbf{w}^{(i)} & \text{if } \left|\mathbf{c}^{(i)}\right| \le \gamma \left|\mathbf{c}^{(i-1)}\right| \\ \beta \mathbf{w}^{(i)} & \text{if } \left|\mathbf{c}^{(i)}\right| > \gamma \left|\mathbf{c}^{(i-1)}\right| \end{cases}, \tag{3}$$

where β >1 and 0< γ <1. Under the strict assumption of convexity, the method of multipliers can be shown to converge to the global optimum for any positive non-decreasing penalty weight. In general, for nonconvex objectives, the quadratic penalty term, including non-decreasing penalty weight, can convexify the objective function with large penalty weight. Hence, the weight update scheme ensures that the weights become large enough to convergence.

5. Proposed Method: Modified AL-AD for Parallel Solving in Non-hierarchical ATC

5.1 Parallel solving

Multidisciplinary Design Optimization (MDO), which includes ATC, aims to find the optimum of the large-scale system problem involving multiple disciplines or components. Therefore, if the system is partitioned to subsystems, the parallel optimization of each subproblem can take full advantage of the characteristics of the MDO problem. In this paper, we discuss the process of reducing the latency by implementing parallel solving of non-hierarchical ATC by modifying traditional coordination algorithm, even if it slightly increases the total number of function evaluation to reach the optimum. We also discuss the implementation of parallel solving by separating all subproblems and proposing a way to treat the problems caused by parallel solving (e.g., the divergence of the LM or oscillation of solution).

The basic idea of parallel solving using traditional ATC, including hierarchical ATC and non-hierarchical ATC, is to use the results obtained from a previous iteration (i.e., the constant parameters) with respect to the subproblem of the current iteration, as depicted in Figure 2. Thus, each subproblem takes one iteration delayed information (e.g., target value). The method of using the values of the previous iteration in AL-AD differs from that of motivation, but the basic concept of formulation in the inner loop optimization is similar with the DQA. The DQA and TDQA, proposed by Li et al. [10], assume that all subproblems can be solved in a parallel manner in a hierarchical ATC. In DQA, the quadratic penalty term that has a cross term in the AL function is linearized to make the subproblems separable. A linearization, which is applied to the cross term included in the quadratic penalty term, is performed by employing first-order Taylor expansion. The cross term is linearized at the point of target and response obtained from the previous iteration; therefore, each subproblem, even though they communicate with one another, can be solved in parallel because of the lack of cross term. However, the DQA method only uses the targets obtained from the previous iteration to make the approximation; not all targets from previous iterations (e.g., the Lagrangian term) are used even if they do not affect the optimization results. The DQA method makes subproblems separable by linearization in the hierarchical AL-AD; results show that the convergence of the inner loop in each subproblem is optimized with fixed parameters regardless of whether the target or response from a previous iteration is used. In this paper, we use the objective function as AL function, so its convergence theorem can be applied to our proposed method in the inner loop. However, given that the double loop that distinguishes the inner loop from the outer loop has much greater computational cost than the single loop (e.g., AL-AD, TDQA, and EPF 2), the proposed algorithm also has a single-loop concept. The single loop concept only has a single inner loop optimization with fixed parameters and the parameters are updated after a single optimization of each subproblem. Unless the LM is not a proper value in each subproblem, the convergence of the inner loop becomes completely irrelevant to the entire system-level optimum. Hence, rather than the inner loop, we focus on the outer loop, especially the updating of the LM.

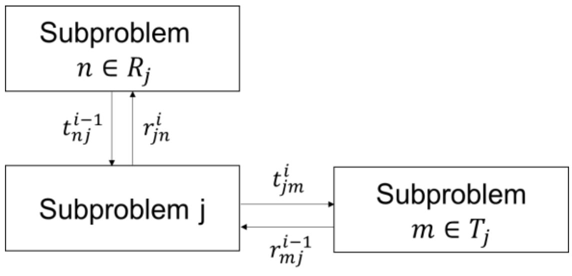

Figure 2: Inner loop of the modified AL-AD for parallel solving in non-hierarchical ATC

5.2 Formulation with modified AL-AD for parallel solving in non-hierarchical ATC

One of the most widely used penalty functions for non-hierarchical ATC is the AL-AD using method of multipliers, which is one way of updating the LM and penalty weight. Due to the presence of quadratic penalty term, however, each coupled subproblem should be optimized consecutively. Hence, the proposed method uses the results of the previous iteration, which are denoted by superscript *i*-1. The general optimization formulation of subproblem *j*, which is based on the notation of Eq. (1), is shown in Eq. (4)

$$\begin{gathered}\min_{\overline{\mathbf{x}}_j} f_j(\overline{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j}\theta(\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)}) + \sum_{m\in T_j}\theta(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) \\ \text{where } \theta(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \left\|\mathbf{w}\circ\mathbf{c}\right\|_2^2 \\ \text{subject to } \mathbf{g}_j(\overline{\mathbf{x}}_j^{(i)}) \le 0 \\ \mathbf{h}_j(\overline{\mathbf{x}}_j^{(i)}) = 0 \\ \text{with } \mathbf{r}_{jn}^{(i)} = \mathbf{S}_{jn}\mathbf{a}_j(\mathbf{x}_j^{(i)}, \mathbf{t}_{jm}^{(i)} \,|\, m\in T_j), \qquad \mathrm{n}\in R_j \\ \overline{\mathbf{x}}_j^{(i)} = [\mathbf{x}_j^{(i)}, \mathbf{r}_{jn}^{(i)} \,|\, n\in R_j, \mathbf{t}_{jm}^{(i)} \,|\, m\in T_j]\end{gathered} \tag{4}$$

In the modified AL-AD for parallel solving in non-hierarchical ATC, each subproblem uses previous optimization results from other subproblems, regardless of the target or response in conventional ATC; therefore, all subproblems can be separated in the current iteration. Compared with Eq. (1), we can see one major difference in the inner loop formulation. The target values from the upper-level in non-hierarchical ATC, $\mathbf{t}_{nj}^{(i)}$ is converted to $\mathbf{t}_{nj}^{(i-1)}$ for parallelization. As mentioned in the previous section, and as shown in Eq. (5), the similar concept of formulation in the inner loop optimization, which is modified for non-hierarchical ATC, is already proposed. However, the DQA method, which uses the approximation of the cross term based on Taylor expansion, uses the target value of the previous iteration but only in the cross term, because the Lagrangian term is constant and does not affect the objective function. In Eq. (5), the LM term with target value $(\mathbf{v}_{nj}^{(i)})^T\mathbf{t}_{nj}^{(i)}$ is used to represent the current target values, but this term does not affect the optimization results because it is constant in the optimization of the subproblem *j*.

$$\min_{\overline{\mathbf{x}}_j} f_j(\overline{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j}((\mathbf{v}_{nj}^{(i)})^T(\mathbf{t}_{nj}^{(i)} - \mathbf{r}_{jn}^{(i)}) + \left\|\mathbf{w}_{nj}^{(i)}\circ(\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)})\right\|_2^2) + \sum_{m\in T_j}((\mathbf{v}_{jm}^{(i)})^T(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) + \left\|\mathbf{w}_{jm}^{(i)}\circ(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)})\right\|_2^2) \tag{5}$$

The DQA, especially TDQA (the single-loop version of the DQA), has numerical instability in non-hierarchical ATC and is sensitive to parameters (e.g., step size of the design progress) that play an important role in the DQA method. As it is based on linearization, it is only accurate in a neighborhood of the point. Sometimes, the LM diverges without a small step size for a good approximation. This phenomenon can be attributed to the outer loop, which updates the LM, penalty weight, and step size of the design progress. It shows good convergence in parallel solving with conservative parameters, but not good enough in non-hierarchical ATC, which directly communicates shared variable. Thus, in the modified AL-AD for parallel solving, the objective function is expressed in Eq. (6) below.

$$\min_{\overline{\mathbf{x}}_j} f_j(\overline{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j}((\mathbf{v}_{nj}^{(i)})^T(\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)}) + \left\|\mathbf{w}_{nj}^{(i)}\circ(\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)})\right\|_2^2) + \sum_{m\in T_j}((\mathbf{v}_{jm}^{(i)})^T(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) + \left\|\mathbf{w}_{jm}^{(i)}\circ(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)})\right\|_2^2) \tag{6}$$

Unlike DQA, all targets are the results of previous iterations of the proposed method. Obviously, each optimization in the inner loop has no effect on the results, because $(\mathbf{v}_{nj}^{(i)})^T\mathbf{t}_{nj}^{(i-1)}$ is constant during the optimization of subproblem *j*. Comparing Eq. (5) and (6), if all parameters are the same, the optimization results in the inner loop, $\overline{\mathbf{x}}_j^{(i)}$, are exactly the same in both cases. However, this modification makes a difference in the convergence of the LM in the outer loop. In the proposed method, we use the subgradient method to update the LM with an adaptive step size rather than the method of multipliers which specialized in sequential solving [3,8]. In comparison, the proposed method intentionally delays information, such as targets and responses from other subproblems. Therefore, the modified LM update with adaptive step size is needed to achieve the convergence of the LM in the outer loop with delayed information. Hence, we use the subgradient update method based on duality theory instead of the method of multipliers in the AL formulation.[8] The effect and meaning of the modified formulation are introduced in Section 5.3, and the updated algorithms of parameters are introduced in Section 5.4

5.3 Modified Lagrange multiplier update using the subgradient method

We propose a new approach of parallel solving in non-hierarchical ATC by using the results of a previous iteration. Without properly updating the parameters, just using the formulation of Eq. (4) can cause oscillation of the solution and divergence of inconsistency due to the instability of the LM. The main difficulty of this formulation is finding a way to solve the divergence of the LM, which begins with the mismatch of information. Thus, we modify the definition of inconsistency after one single inner loop iteration based on the subgradient. In traditional ATC, inconsistency refers to the discrepancy between target and response after the convergence of the inner loop, as shown Eq. (7). The process of updating the scheme of the LM based on method of multipliers is given by Eq. (2) and Eq. (3). In the modified AL-AD for parallel solving, the inconsistency should be defined differently considering the delay of information, as shown in Eq. (8).

$$\mathbf{c}_{nj}^{(i)} = \mathbf{t}_{nj}^{(i)} - \mathbf{r}_{jn}^{(i)} \tag{7}$$

$$\mathbf{c}_{nj}^{(i)} = \mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)} \tag{8}$$

Obviously, the inconsistency expressed in Eq. (8) is not used in a criterion to determine the convergence of the outer loop. The convergence of a whole system is determined based on the consistency between the current target value and the response value, as shown in Eq. (7). Thus, the objective of the whole algorithm is to make two kinds of inconsistency that are close to zero. Eq. (8) represents the inconsistency used in updating the LM with the subgradient method; the proposed method uses target values obtained from the previous iteration, as shown in Eq. (6). The objective function of the dual problem of subproblem *j* in the proposed method is expressed in Eq. (9).

$$\begin{aligned} &\max_{\mathbf{v}_{nj},\mathbf{v}_{jm}} \phi_j(\mathbf{v}_{nj},\mathbf{v}_{jm}) \\ &\text{where } \phi_j(\mathbf{v}_{nj},\mathbf{v}_{jm}) = \min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j}((\mathbf{v}_{nj}^{(i)})^T(\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)}) + \left\|\mathbf{w}_{nj}^{(i)} \circ (\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)})\right\|_2^2) + \\ &\sum_{m\in T_j}((\mathbf{v}_{jm}^{(i)})^T(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) + \left\|\mathbf{w}_{jm}^{(i)} \circ (\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)})\right\|_2^2) \end{aligned} \tag{9}$$

In Eq. (9), the duality theory is implemented in the AL function. The local constraints and other design parameters are based on Eq. (4). Thus, the subgradient of $\phi_j(\mathbf{v}_{nj},\mathbf{v}_{jm})$ with respect to $\mathbf{v}^{(i)}$ is calculated in Eq. (8). As examples, $\frac{\partial\phi_j}{\partial\mathbf{v}_{nj}^{(i)}}$ and $\frac{\partial\phi_j}{\partial\mathbf{v}_{jm}^{(i)}}$ are $\mathbf{c}_{nj}^{(i)}$ and $\mathbf{c}_{jm}^{(i)}$. Hence, $\mathbf{c}_{nj}^{(i)}$ and $\mathbf{c}_{jm}^{(i)}$ can be used as the subgradients of the corresponding LM. The proposed coordination method to find the optimum LM is also based on duality theory. In this paper, for the convexity of the objective function, we also maintain penalty weight $\mathbf{w}$ as minimum initial value just to convexify the objective function and prevent the objective function from being unbounded. More detailed descriptions about ATC dual coordination and subgradient method are discussed in [8,9]. Here, we propose the sequential LM update with the subgradient method, as shown in Eq. (10).

$$\mathbf{v}_{nj}^{(i+1)} = \mathbf{v}_{nj}^{(i)} + \alpha_{nj}^{(i)} \circ (\mathbf{t}_{nj}^{(i-1)} - \mathbf{r}_{jn}^{(i)}) \tag{10}$$

$$\mathbf{w}_{nj}^{(i+1)} = \mathbf{w}_{nj}^{(0)} \tag{11}$$

The step size vector $\alpha_{nj}^{(i)}$ can be $2\mathbf{w}_{nj}^{(i)} \circ \mathbf{w}_{nj}^{(i)}$ in method of multipliers or any step size used in the subgradient method. The important thing is that the proposed Lagrangian update algorithm uses delayed inconsistency based on Eq. (4). In this way, the divergence of the LM does not become an issue if the step size is appropriate, as in the case of a constant or diminishing value used in the subgradient method. However, the convergence of the ATC depends on the inconsistency between the current target and response values. Therefore, the convergence criterion of the outer loop is based on the inconsistency between $\mathbf{t}_{nj}^{(i)}$ and $\mathbf{r}_{jn}^{(i)}$, not $\mathbf{t}_{nj}^{(i-1)}$ and $\mathbf{r}_{jn}^{(i)}$. In addition, even if the proper subgradient is used in updating the scheme, the oscillation of the LM and the current solution may occur due to the quadratic penalty function.

The problem of oscillation in parallel solving is related to the crossing of information and inadequate step size $\alpha_{nj}^{(i)}$ used in updating the LM and excessive quadratic penalty function. In conventional AL-AD, a method of multipliers is used to update the LM, so the step size is defined as $2\mathbf{w}_{nj}^{(i)} \circ \mathbf{w}_{nj}^{(i)}$, which has great convergence property in the sequential solving of each subproblem with updated penalty weight using Eq. (3). In parallel solving, however, it is not quite helpful, because increasing penalty weight and step size can accelerate the oscillation of solution. This is because the quadratic penalty term becomes too dominant in the objective function and the step size becomes too large to reach the optimum, thus causing the solution to oscillate or even diverge. The dominant quadratic penalty function makes each response too close to the target, causing oscillation instead of finding the optimum LM. This is the reason why we use the subgradient method and fixed penalty weight to find the optimum LM and prevent oscillation appears only in parallel solving, as shown in Eq. (10) and Eq. (11).

In the proposed method, the penalty weight term is only required to prevent the objective function from being unbounded and to convexify the objective function; the use of the penalty weight has no impact on directly finding the optimum LM. Thus, the penalty weight term is fixed considering the objective function or variables; the step size is also considered to be independent of penalty weight. The existing step size of the subgradient method, such as constant or non-summable diminishing, is not efficient for finding optimum LM. Thus, we propose a new, efficient step size updating algorithm for parallel solving based on the path of the LM.

5.4 Adaptive step size updating algorithm

The step size updating algorithm plays an important role in the convergence of ATC. In the proposed method, this algorithm is the key to the successful convergence and accuracy of the LM and solution. If subgradient means a direction, step size means a magnitude which related to convergence speed. In the current section, we discuss the process of updating step size based on the path of the LM. So it derives the best step size for the current LM. The role of step size in updating the LM with the subgradient method is simple: if the convergence rate of the LM is slow, then the step size should be increased. On the contrary, if the oscillation of the LM occurs, the step size should be decreased in parallel solving so that the LM can converge to an optimum. Commonly, the step size of subgradient in traditional ATC coordination, inconsistency is the main indicator to update step size. However, we use LM itself because the objective of step size is to find optimum LM and it shows good performance.

Two phases are involved in the adaptive step size updating algorithm. In Phase 1, the current LM is considered to be far from the optimum, and in Phase 2, the current LM is considered to be near the optimum. The criterion for the transition from Phase 1 to Phase 2 is the change in the direction of the LM. In the proposed algorithm, the first and second derivatives of LM with respect to pseudo-time may be calculated. The pseudo-time is related to the evolution of design, so it has the same meaning as the outer loop iteration in AL-AD. Hence, the changes in the direction of LM can be expressed in Eq. (12); if the condition is satisfied such that the product of two consecutive values of first derivative is changed, this means that the current LM has passed the optimum. Thus, Phase 1 is switched to Phase 2.

$$\frac{dv_k^{(i-1)}}{dt} \times \frac{dv_k^{(i)}}{dt} < 0$$
$$\text{where } v_k^{(i)} \text{ is element of } \mathbf{v}^{(i)} \qquad (i = 1, ..., n_c) \tag{12}$$
$$\frac{dv_k^{(i)}}{dt} = \frac{v_k^{(i)} - v_k^{(i-1)}}{\Delta t}$$

The adaptive step size updating algorithm has different update schemes depending on the phase. In phase 1, the LM is far from the optimum; therefore, the speed of convergence to optimum is more important than accuracy. The first derivative of the LM with respect to the outer loop iteration is the performance indicator. The process of updating the step size in Phase is expressed in Eq. (13). Two parameters are introduced in the proposed algorithm.

$$\alpha_k^{(i+1)} = \begin{cases} \alpha_k^{(i)} & \text{if } \dfrac{dv_k^{(i)}}{dt} \geq \eta \\ \beta_1 \alpha_k^{(i)} & \text{if } \dfrac{dv_k^{(i)}}{dt} < \eta \end{cases} \qquad \text{k=1,...,n}_c \tag{13}$$

In the equation above, β_1 is the increasing factor of step size, and η is the parameter used in updating the criterion. Typically, $\beta_1 = 1.1$ is recommended to speed up, and η is dependent on the scale of the LM and the user's preference. Unlike common step size of subgradient method, step size is increased for convergence rate in phase 1. In phase 2, which is switched from Phase 1 based on Eq. (12), the LM is close to the optimum so the large step size is no longer needed. Rather, the decreasing step size can prevent oscillation between the target and response in parallel solving, thereby ensuring the convergence of the LM. The occurrence of oscillation means that the optimum is exceeded and, at the same time, the sign of design inconsistency may be changed. Hence, the change in the direction of the LM (rather than first derivative itself) is the performance indicator that can signify the oscillation. In this case, the step size associated to each inconsistency and the corresponding LM is updated, as shown in Eq. (14).

$$\alpha_k^{(i+1)} = \begin{cases} \alpha_k^{(i)} & \text{if } \dfrac{dv_k^{(i-1)}}{dt} \times \dfrac{dv_k^{(i)}}{dt} \geq 0 \\ \beta_2 \alpha_k^{(i)} & \text{if } \dfrac{dv_k^{(i-1)}}{dt} \times \dfrac{dv_k^{(i)}}{dt} < 0 \end{cases} \qquad \text{k=1,...,n}_c \tag{14}$$

In the equation above, β_2 is the decreasing factor of step size, and η_2 is the parameter used in updating the criterion. Typically, $\beta_2 = 0.9$ is recommended to converge. This approach has a similar concept with Eq. (13). The direction change of LM with respect to the outer loop iteration near the optimum means that the current step size is too large to converge, so it should be decreased. As demonstrated by the numerical example, the proposed algorithm with two phases shows good efficiency in terms of computational cost.

6. Illustration Example

In this section, the 14-variable nonconvex geometric programming problem that is widely used in many ATC research papers is solved by using the proposed modified AL-AD with adaptive step size updating algorithm for parallel solving. The numerical results of each problem are obtained by using the proposed method, and the results are compared with those of the conventional AL-AD and TDQA, which are widely used as benchmark in ATC. Performance indicators and termination criteria are generally used equations in other papers. First, in the performance indicator, three measures are evaluated for quantifying the performance of each method, namely, solution error, total function evaluation to converge, and computational cost in terms of latency. Solution error is defined as the accuracy of a solution obtained from ATC compared with known optimal solution. Solution error e is defined in Eq. (15) widely used in many ATC papers.

$$e = \left\| \mathbf{x}^* - \mathbf{x}^{(i)} \right\|_\infty , \tag{15}$$

where $\mathbf{x}^{(i)}$ is the known optimal solution, and $\mathbf{x}^{(i)}$ is the solution of ATC when inconsistency satisfies the outer loop convergence criterion. The outer loop criterion is based on the convergence of two consecutive solutions of the outer loop and maximum consistency constraint violation, as respectively shown in Eq. (16) and Eq. (17)

$$\left\| \frac{(\mathbf{t}^{(i)} - \mathbf{r}^{(i)}) - (\mathbf{t}^{(i-1)} - \mathbf{r}^{(i-1)})}{(\mathbf{t}^{(i-1)} - \mathbf{r}^{(i-1)})} \right\|_\infty < \tau_{outer} , \tag{16}$$

$$\left\| \mathbf{t}^{(i)} - \mathbf{r}^{(i)} \right\|_{\infty} < \tau_{outer} . \tag{17}$$

Using MATLAB, the total function evaluation and latency are evaluated when outer loop criterion is satisfied with respect to each of termination tolerance $\tau_{outer} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. The latency is the most important indicator that is suitable for the purpose of parallel solving. A 14-variable nonlinear constrained optimization problem is formulated below

$$\begin{aligned}
&\min_{z_1,\cdots,z_{14}} f = z_1^2 + z_2^2 \\
s.t.\quad & g_1 = (z_3^{-2} + z_4^2) z_5^{-2} - 1 \le 0 \\
& g_2 = (z_5^2 + z_6^{-2}) z_7^{-2} - 1 \le 0 \\
& g_3 = (z_8^2 + z_9^2) z_{11}^{-2} - 1 \le 0 \\
& g_4 = (z_8^{-2} + z_{10}^2) z_{11}^{-2} - 1 \le 0 \\
& g_5 = (z_{11}^2 + z_{12}^{-2}) z_{13}^{-2} - 1 \le 0 \\
& g_6 = (z_{11}^2 + z_{12}^{-2}) z_{14}^{-2} - 1 \le 0 \\
& h_1 = (z_3^2 + z_4^{-2} + z_5^2) z_1^{-2} - 1 = 0 \\
& h_2 = (z_5^2 + z_6^2 + z_7^2) z_2^{-2} - 1 = 0 \\
& h_3 = (z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2) z_3^{-2} - 1 = 0 \\
& h_4 = (z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2) z_6^{-2} - 1 = 0 \\
& z_i \ge 0, i = 1, \ldots 14
\end{aligned}$$

The unique optimal solution of this problem is given by $\mathbf{z}^*$ =[2.84,3.09,2.36,0.76,0.87,2.81,0.94,0.97, 0.87,0.8,1.3,0.84, 1.76, 1.55] with all constraints active. We solved the problem according to the two decompositions shown in Figure 4 [22], which consist of three and five non-hierarchical subproblems, respectively. Each subproblem that is connected by shared variables should directly give and receive targets and responses in a non-hierarchical system.

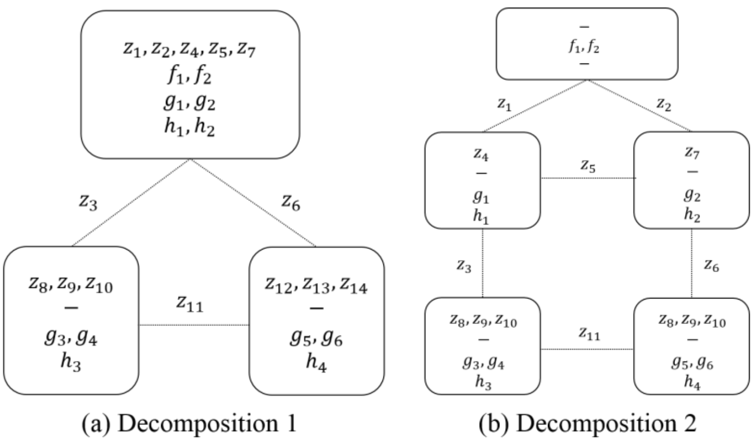

Figure 4: Decomposition details for the geometric optimization problem [22].

Given that the modified AL-AD for parallel solving has a single-loop algorithm, the decomposed problem is also solved using AL-AD and TDQA. In AL-AD, TDQA, and modified AL-AD for parallel solving, the initial parameters are set as $\mathbf{v}^{(0)} = 0$ and $\mathbf{w}^{(0)} = 1$. The initial point is $\mathbf{x}^{(0)} = [0,0,...0]$ for all methods. For AL-AD and TDQA, $\beta = 1$. For the adaptive step size updating algorithm, the parameters are $\beta_1 = 1.1, \beta_2 = 0.9, \eta = 1.5$. One of the main advantages of the proposed method compared with TDQA is robustness with respect to the step size of the design progress; this is distinct from the step size of the LM. Decomposition 2 has more than a few shared variables, and the non-hierarchical structure may cause the TDQA to diverge depending on the step size of the design progress or the step size of the LM. All parameters are the same, and the only difference is the inconsistency used in the LM. In TDQA, if the step size is 1 or 0.8, it cannot be converged. In comparison, the proposed method can treat the non-hierarchical problem even if it has a highly nonlinear property. The total number of function evaluation and latency versus solution error is described in Figure 5. For a fair comparison, the step sizes of the design progress in TDQA are 1 and 0.7, which show the best performances in both cases of decomposition, respectively.

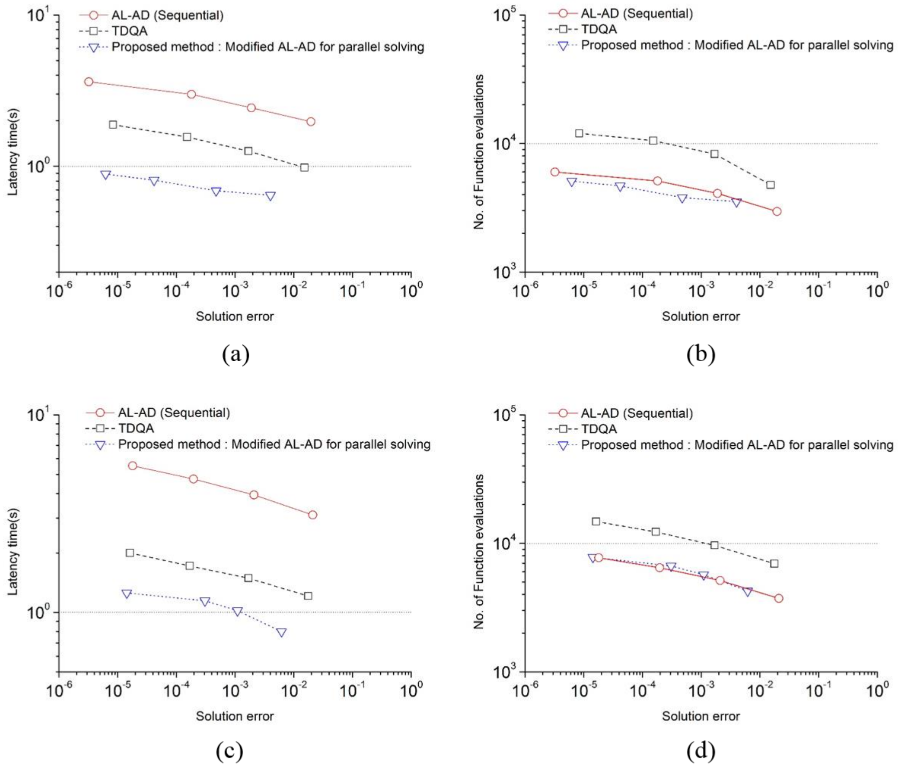

Figure 5: (a) Latency and (b) Total function evaluation versus solution error in decomposition 1 and (c) Latency and (d) Total function evaluation versus solution error in decomposition 2

As can be seen in Figure 5, the proposed coordination algorithm shows good convergence properties with respect to other methods. The modified AL-AD is based on parallel solving, so it is even more efficient in terms of time than the conventional sequential AL-AD when it is divided into many subsystems, such as decomposition 2. The proposed method also outperforms the TDQA as the only existing method of parallelization without reformulation.

7. Conclusion

This paper presents a modified AL-AD approach for parallel solving based on non-hierarchical ATC framework. Originally, two problems in the parallelization of ATC are discussed: the divergence of the LM and the oscillation of the solution. Divergence and Oscillation are due to the crossing of target and response and the non-decreasing penalty weight in the method of multipliers. Thus, the proposed method uses the subgradient method with delayed information to find the optimum LM based on duality theory and also fix the penalty weight to ensure the convexity of the objective function and to prevent it from being unbounded. We also propose an efficient algorithm to update the step size based on the path of the LM which reflects current information because it is updated based on inconsistencies. As a results, the proposed algorithm indicated good performance compared with existing methods in numerical example. The sequential updating of the LM and the fixed penalty weight guarantees right direction to optimum. The adaptive step size update increases efficiency and prevents oscillation in parallel solving. In future works, we will develop the criterion for how to set the initial weight and modify adaptive step size algorithm to identify accurate step sizes as well as investigate the numerical performance of the proposed method using more numerical examples. We will also expand our research to probabilistic domain for solving industrial problems partitioned in many subsystems. Furthermore, we will integrate our parallel solving scheme of distributed coordination on industrial problems dealing with reliability analysis. [15,16].

8. References

[1] Bayrak, A. E., Kang, N., & Papalambros, P. Y. (2016). Decomposition-based design optimization of hybrid electric powertrain architectures: Simultaneous configuration and sizing design. *Journal of Mechanical Design*, 138(7), 071405.

[2] Bertsekas, D. P. (1999). Nonlinear programming (pp. 1-60). Belmont: Athena scientific.

[3] Boyd, S., Xiao, L., & Mutapcic, A. (2003). Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004.

[4] DorMohammadi, S., & Rais-Rohani, M. (2013). Exponential penalty function formulation for multilevel optimization using the analytical target cascading framework. *Structural and Multidisciplinary Optimization*, 47(4), 599-612.

[5] Han, J., & Papalambros, P. Y. (2010). A sequential linear programming coordination algorithm for analytical target cascading. *Journal of Mechanical Design*, 132(2), 021003.

[6] Kang, N., Kokkolaras, M., & Papalambros, P. Y. (2014). Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading. *Structural and Multidisciplinary Optimization*, 50(5), 849-859.

[7] Kang, N., Kokkolaras, M., Papalambros, P. Y., Yoo, S., Na, W., Park, J., & Featherman, D. (2014). Optimal design of commercial vehicle systems using analytical target cascading. *Structural and Multidisciplinary Optimization*, 50(6), 1103-1114.

[8] Kim, H. M., Chen, W., & Wiecek, M. M. (2006). Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA journal*, 44(10), 2197-2207.

[9] Lassiter, J. B., Wiecek, M. M., & Andrighetti, K. R. (2005). Lagrangian coordination and analytical target cascading: solving ATC-decomposed problems with Lagrangian duality. *Optimization and Engineering*, 6(3), 361-381.

[10] Li, Y., Lu, Z., & Michalek, J. J. (2008). Diagonal quadratic approximation for parallelization of analytical target cascading. *Journal of Mechanical Design*, 130(5), 051402.

[11] Michalek, J. J., & Papalambros, P. Y. (2004). An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading. *In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 159-168). American Society of Mechanical Engineers.

[12] Michelena, N., Park, H., & Papalambros, P. Y. (2003). Convergence properties of analytical target cascading. *AIAA journal*, 41(5), 897-905.

[13] Kim, H. M., Michelena, N., Papalambros, P. Y., Jiang T (2003) Target cascading in optimal system design. *Journal of Mechanical Design* 125(3):474–480

[14] Kim, H. M., Rideout, D. G., Papalambros, P. Y., Stein JL (2003) Analytical target cascading in automotive vehicle design. *Journal of Mechanical Design* 125:481–489.

[15] Shin, J.,& Lee, I. (2014). Reliability-Based Vehicle Safety Assessment and Design Optimization of Roadway Radius and Speed Limit in Windy Environments. *Journal of Mechanical Design*, 136(8), 081006.

[16] Shin, J., & Lee, I. (2015). Reliability analysis and reliability-based design optimization of roadway horizontal curves using a first-order reliability method. *Engineering Optimization*, 47(5), 622-641.

[17] Tosserams, S., Etman, L. F. P., & Rooda, J. E. (2008). Augmented Lagrangian coordination for distributed optimal design in MDO. *International journal for numerical methods in engineering*, 73(13), 1885-1910.

[18] Tosserams, S., Etman, L. F. P., & Rooda, J. E. (2008). Performance evaluation of augmented Lagrangian coordination for distributed multidisciplinary design optimization. *In 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference* (p. 1805).

[19] Tosserams, S., Etman, L. F. P., Papalambros, P. Y., & Rooda, J. E. (2006). An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Structural and multidisciplinary optimization*, 31(3), 176-189.

[20] Tosserams, S., Etman, L. F., & Rooda, J. E. (2009). A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization*, 39(5), 503-517.

[21] Tosserams, S., Etman, L. F., & Rooda, J. E. (2009). Block-separable linking constraints in augmented Lagrangian coordination. *Structural and Multidisciplinary Optimization*, 37(5), 521-527.

[22] Tosserams, S., Etman, L. P., & Rooda, J. E. (2007). An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Structural and Multidisciplinary Optimization*, 34(3), 211-227.

[23] Tosserams, S., Kokkolaras, M., Etman, L. F. P., & Rooda, J. E. (2010). A nonhierarchical formulation of analytical target cascading. *Journal of Mechanical Design*, 132(5), 051002.

[24] Wang, W., Blouin, V. Y., Gardenghi, M. K., Fadel, G. M., Wiecek, M. M., & Sloop, B. C. (2013). Cutting plane methods for analytical target cascading with augmented Lagrangian coordination. *Journal of Mechanical Design*, 135(10), 104502.