Modified Augmented Lagrangian Coordination and Alternating Direction Method of Multipliers with Parallelization in Non-hierarchical Analytical Target Cascading

Yongsu Jung

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology,

Daejeon, 34141, South Korea

Namwoo Kang

K-School, Korea Advanced Institute of Science and Technology,

Daejeon, 34141, South Korea

Ikjin Lee*

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology,

Daejeon, 34141, South Korea

Keywords: Multidisciplinary Design Optimization, Analytical Target Cascading, Parallelization, Augmented Lagrangian Coordination

* Corresponding author: ikjin.lee@kaist.ac.kr

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 with Alternating Direction method of multipliers (AL-AD), one of efficient ATC coordination methods, has been widely used in both hierarchical 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 distributed coordination which consists of several non-hierarchical subproblems is that it can solve subsystem problems in parallel and thus reduce computational time. Therefore, previous studies have proposed an augmented Lagrangian coordination strategy for parallelization by eliminating interactions among subproblems. The parallelization is achieved by introducing a master problem and support variables or by approximating a quadratic penalty function to make subproblems separable. However, conventional AL-AD does not guarantee convergence in the case of parallel solving. Our study shows that, in parallel solving using targets and responses of the current iteration, conventional AL-AD causes mismatch of information in updating the Lagrange multiplier. Therefore, the Lagrange multiplier may not reach the optimal point, and as a result, increasing penalty weight causes numerical difficulty in the augmented Lagrangian coordination approach. To solve this problem, we propose a modified AL-AD with parallelization in non-hierarchical ATC. The proposed algorithm uses the subgradient method with adaptive step size in updating the Lagrange multiplier and also maintains penalty weight at an appropriate level not to cause oscillation. Without approximation or introduction of an artificial master problem, the modified AL-AD with parallelization can achieve similar accuracy and convergence with much less computational cost compared with conventional AL-AD with sequential solving.

1. Introduction

Many studies have been published on distributed coordination methodology for multidisciplinary design optimization and decomposition-based optimization. In this paper, we focus on the Analytical Target Cascading (ATC) which is a coordination method specialized in hierarchical decomposition in multilevel and is developed originally for translating a system-level target to design specifications for components. It has been useful as a coordination method that decomposes the system into subsystems hierarchically and successfully applied to large scale engineering problems, e.g., chassis design of an automotive (Kim et al. 2001, 2003a, 2003b). Top-level design targets are propagated to the lower level with consecutive optimizations to obtain system optimum with satisfying the design targets. Obviously, distributed design optimization including ATC usually requires more computational cost than the AiO (All-in-One) strategy. However, the need for decomposition is usually raised when system optimization with a large scale cannot be solved with AiO because disciplinary design teams use specialized FEA models that have been developed in different modules. Thus, taking all models in one single optimization problem is impractical or impossible in such a case.

The main issue of ATC is how to deal with inconsistency between the target and response in mathematical formulation. A Quadratic Penalty function (QP) of system consistency constraints is proposed in the beginning in case of open consistency constraints (Bertsekas et al. 2003). Then, the convergence proof of ATC with the quadratic penalty function is also proposed along with the iterative method for finding minimal penalty weight that can satisfy specified inconsistency tolerances (Michelena et al. 2003, Michalek et al. 2004). However, large penalty weight is required to achieve convergence, and it can cause ill-conditioning of the problem and numerical instability simultaneously (Bertsekas et al. 2003). An alternative for the consistency relaxation function is an Ordinary Lagrangian function (OL) (Lassiter et al. 2005) using the Lagrangian duality theory, and Kim et al. (2006) proposed the Lagrangian coordination for enhancing the convergence of ATC based on OL. Similarly, Tosserams et al. (2006) proposed the Augmented Lagrangian with Alternating Direction method of multipliers (AL-AD) to improve the performance of ATC coordination using the augmented Lagrangian function and eliminating the inner loop. This approach shows both good convergence property and efficiency with low computational cost

(Tosserams et al. 2008b, 2009a, 2009b).

After AL-AD is developed, several strategies using the augmented Lagrangian coordination for parallelization have been proposed. Li et al. (2008) applied Diagonal Quadratic Approximation (DQA) to ATC which uses the first-order Taylor expansion at points of previous iteration for approximating the quadratic penalty function. However, use of the Taylor expansion can lead to numerical instability depending on the characteristics of the problem, such as linearity. In another study, Tosserams et al. (2009a, 2010) proposed a new coordination that allows non-hierarchical target and response flow and implements the parallelization by introducing an artificial master problem, which is called centralized coordination. However, solving the distributed coordination itself with an appropriate coordination is more convenient rather than constructing the artificial problem and reformulating the system to centralized coordination. Thereafter, some studies on the industrial application (Kang et al. 2014a, 2014b, Bayrak et al. 2016), penalty functions (DorMohammadi et al. 2013), updating scheme of the Lagrange multiplier (Wang et al. 2013), and sequential linearization techniques (Han et al. 2010) have been proposed, but not in the parallelization of ATC.

In this paper, we propose a modified AL-AD with parallelization in non-hierarchical ATC. The proposed method is based on AL-AD coordination but also uses the subgradient update method (Boyd et al. 2003) with modification and a guideline for selecting penalty weights to find the optimum Lagrange multiplier. We also propose an adaptive step size updating algorithm which allows setting the proper step size for each outer loop based on the path of the Lagrange multiplier, thus improving convergence. The proposed method is formulated based on non-hierarchical ATC formulation (Tosserams et al. 2010). The reason why we deal with non-hierarchical ATC is that hierarchical ATC is just a special case of the non-hierarchical ATC, and in reality, most of the decomposition-based multidisciplinary design optimization problems have non-hierarchical structure.

The article is organized as follows. Existing non-hierarchical formulation with AL-AD is presented in Section 2. In Section 3, the proposed coordination methodology and convergence strategy are presented. Three mathematical problems that are widely used in the literature and an engineering problem of roof assembly design are employed to demonstrate efficiency and feasibility of the proposed methodology in Section 4, and the conclusion is given in Section 5.

2. Review of Non-hierarchical ATC with Augmented Lagrangian Coordination

2.1 Non-hierarchical ATC formulation

The traditional ATC formulation uses a hierarchical decomposition that coordinates consistency constraints of the target and response between parents and children. Parents mean subproblems which transfer targets to children at the lower level and children mean subproblems which receive targets from parents at the upper level. The basic assumption of the traditional ATC formulation 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. (2010) proposed the non-hierarchical ATC formulation that does not have a hierarchical decomposition structure. Any two subproblems can communicate with each other directly, which is impossible in the traditional hierarchical ATC.

In this section, we explain the formulation of existing AL-AD for non-hierarchical ATC. Figure 1 shows functional dependence structure of hierarchical and non-hierarchical ATC, respectively. The arrows indicate flows of targets. In hierarchical ATC, the target should be transfered to the lower level, and communication between subsystems on the same level is not allowed. Non-hierarchical ATC, on the other hand, has no concept of hierarchy, and all subsystems are communicable with any subsystems.

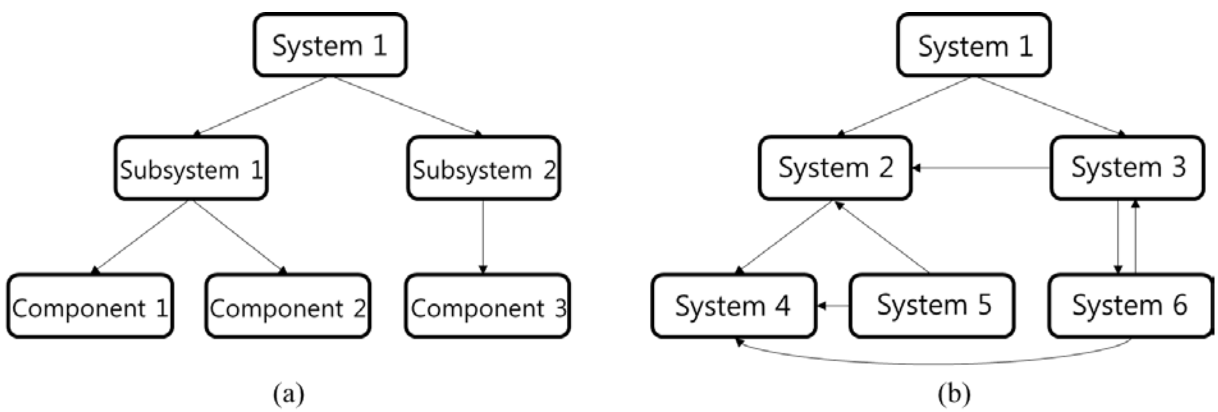

Fig 1 Functional dependence structure of (a) hierarchical and (b) non-hierarchical ATC

In the non-hierarchical ATC formulation, subproblems that communicate with others are called neighbors which are distinct from the concept of parents or children in the hierarchical ATC. Hence, the concept of level can be eliminated in the formulation, but a double index notation which denotes the direction of communication between one subproblem and its neighbor is maintained. The first index denotes the sending subproblem and the second index denotes the receiving subproblem. In other words, it is still necessary to distinguish whether the subproblem is solved earlier or later than oneself and preserve the unique target-response coupling structure.

The iteration is identified with the superscript *i* in Figure 2, which describes the flow of information based on optimization of the subproblem *j*. Let T_j be a set of neighbors, for which the subproblem *j* receives responses and computes the corresponding targets, and let R_j be a set of neighbors, for which the subproblem *j* receives targets and computes the corresponding responses. Figure 2 illustrates the target-response pairs between subproblem *j* and two types of subproblems. Therefore, the targets from subproblem *n* are results of the current iteration, so it has superscript *i*. In contrast, the responses from subproblem *m* are the results of the previous iteration, so it has superscript *i*-1

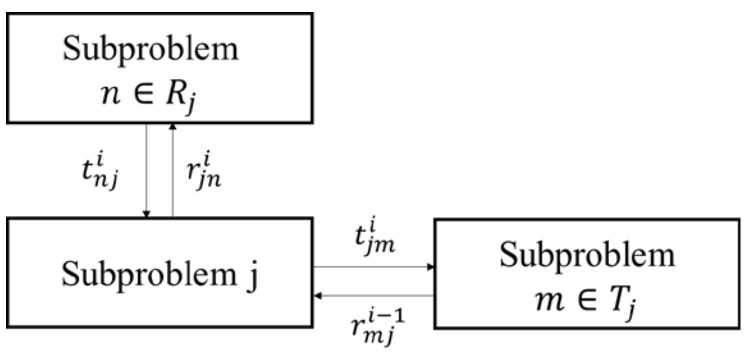

Fig 2 Non-hierarchical target and response flow between subproblem *j* and its neighbors for sequential solving (modified from Tosserams et al. (2010))

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

$$\min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n\in R_j} \pi(\mathbf{t}_{nj}^{(i)} - \mathbf{r}_{jn}^{(i)}) + \sum_{m\in T_j} \pi(\mathbf{t}_{jm}^{(i)} - \mathbf{r}_{mj}^{(i-1)}) \text{ where } \pi(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \|\mathbf{w}\circ\mathbf{c}\|_2^2$$

$$\text{subject to } \mathbf{g}_j(\bar{\mathbf{x}}_j^{(i)}) \le 0,\ \mathbf{h}_j(\bar{\mathbf{x}}_j^{(i)}) = 0 \tag{1}$$

$$\text{with } \mathbf{r}_{jn}^{(i)} = \mathbf{S}_{jn}\mathbf{a}_j(\mathbf{x}_j^{(i)}, \mathbf{t}_{jm}^{(i)} | m\in T_j) \text{ for } n\in R_j,\ \bar{\mathbf{x}}_j^{(i)} = [\mathbf{x}_j^{(i)}, \mathbf{r}_{jn}^{(i)} | n\in R_j, \mathbf{t}_{jm}^{(i)} | m\in T_j]$$

where $\bar{\mathbf{x}}_j$ represents the design variables for subproblem *j*, $\mathbf{x}_j$ represents the local design variables, $\mathbf{t}_{nj}$ represents the targets computed in neighbor subproblem *n* belonging to the set R_j, and subproblem *j* returns $\mathbf{r}_{jn}$ corresponding to $\mathbf{t}_{nj}$. Similarly, $\mathbf{t}_{jm}$ represents the target computed in subproblem *j* transferred to neighbor subproblem *m* belonging to set T_j, and $\mathbf{r}_{mj}$ will be returned to the optimization process of the subproblem *m* corresponding to $\mathbf{t}_{jm}$. f_j, $\mathbf{g}_j$, and $\mathbf{h}_j$ represent the local objective, inequality, and equality constraint functions, respectively. The function $\mathbf{a}_j$ is used to compute responses 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 $[a_1, a_2, ..., a_n]\circ[b_1, b_2, ..., b_n] = [a_1b_1, a_2b_2, ..., a_nb_n]$. These are used in general non-hierarchical ATC formulation with the direction of communication (Tosserams et al. 2010). In this paper, we added superscript *i* which represents the outer loop iteration in the formulation to compare with the proposed coordination.

2.2 Coordination Algorithm

The coordination algorithm operates in the inner and the outer loops to achieve convergence of the targets and the 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 (Tosserams et al. 2006). The main purpose of the coordination algorithm in AL-AD is to find the optimum Lagrange multipliers associated with 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 target from the upper level, the Lagrange multiplier, and penalty weight given to each subproblem. In the outer loop, the Lagrange multipliers and penalty weights are updated in every iteration to perform the optimization of subproblems in the inner loop of the next iteration.

With the augmented Lagrangian coordination, ill-conditioning of the optimization can be avoided by finding the optimum Lagrange multiplier instead of setting large penalty weight for convergence which causes numerical difficulties. With alternating direction method of multipliers, each subproblem is solved only once in the inner loop, instead of solving the iterative inner loop coordination, and it has been shown to reduce the overall computational cost required for convergence (Tosserams et al. 2006, 2007, 2008a, 2009a).

The critical parameters in AL-AD, even if the non-hierarchical ATC is employed, are the Lagrange multiplier denoted as $\mathbf{v}$ and the penalty weight denoted as $\mathbf{w}$. Both parameters determine the performance of ATC including convergence speed, inconsistency between target and response, and accuracy of the solution. In general, the linear update algorithm employed in the augmented Lagrangian coordination is used for updating the Lagrange multiplier in AL-AD as

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

This is also generally known as a method of multipliers (Bertsekas et al. 2003). In Eq. (2), $\mathbf{t}^{(i)} - \mathbf{r}^{(i)}$ refers to the inconsistency between the target and response which is denoted as $\mathbf{c}^{(i)}$ hereafter. Also, the penalty weight $\mathbf{w}^{(i)}$ is updated when the reduction of consistency constraints is insufficient as

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

where β >1 and 0< γ <1. Under the strict assumption of convexity, a method of multipliers can be shown to converge to the global optimum for any positive non-decreasing penalty weight (Bertsekas et al. 1989). In general, for nonconvex objectives, the quadratic penalty function can make the objective function convex with large penalty weight. Hence, the weight update scheme ensures that the weights become large enough to converge.

3. Modified AL-AD with Parallelization in Non-hierarchical ATC

3.1 Parallelization

Multidisciplinary design optimization including ATC aims to find the optimum of a large scale system problem involving multiple disciplines or components. The system is partitioned to many subsystems, and each optimization of subsystem should be conducted autonomously and iteratively while communicating with other subsystems connected by the linking variable. Therefore, parallel optimization of each subsystem can take full advantage of ATC which is decomposed into many subsystems. In this paper, we discuss the process of reducing the latency with implementing parallelization in non-hierarchical ATC and propose a way for parallelization by separating all subproblems and the modified coordination algorithms to treat the numerical difficulties caused by separation (e.g., oscillation and divergence of solution).

The basic idea of parallelization in the coordination of ATC, including hierarchical and non-hierarchical ATC, is to use the results obtained from a previous iteration (i.e., copies of the linking variables) in the current optimization of each subproblem without waiting for optimization of other subproblems, as depicted in Figure 3a. Thus, each subproblem takes delayed information to separate each subproblem. That way, it is not necessary to distinguish the target and the response from other subproblems such as a set R_j and T_j because there is no direction of communication in parallel solving. Therefore, the formulation uses only the linking variables denoted as $\mathbf{z}_{jn}$ in Figure 3b in consistency constraints including all target-response pairs (e.g., coupling variables) or shared variables of subproblem j for simplicity. A more detailed definition of linking variables is described in Papalambros et al. (2017). A set N_j contains all subproblems communicated with subproblem j.

The method of using the values of the previous iteration in AL-AD in the inner loop optimization is similar with DQA. DQA and TDQA, proposed by Li et al. (2008), assume that all subproblems can be solved in a parallel manner in the hierarchical ATC. In DQA, the quadratic penalty function which has a cross term is linearized to make the subproblems separable. However, it differs from the proposed method in that the DQA method only uses the targets obtained from the previous iteration to make the approximation.

In this paper, the proposed method is based on augmented Lagrangian coordination. Also, 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), the proposed algorithm also has a single loop structure. The single loop structure only has a single inner loop optimization with the fixed parameters (e.g., Lagrange multiplier and penalty weight), and those parameters are updated after a single optimization of each subproblem. Unless the Lagrange multiplier is 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 updating of the Lagrange multiplier.

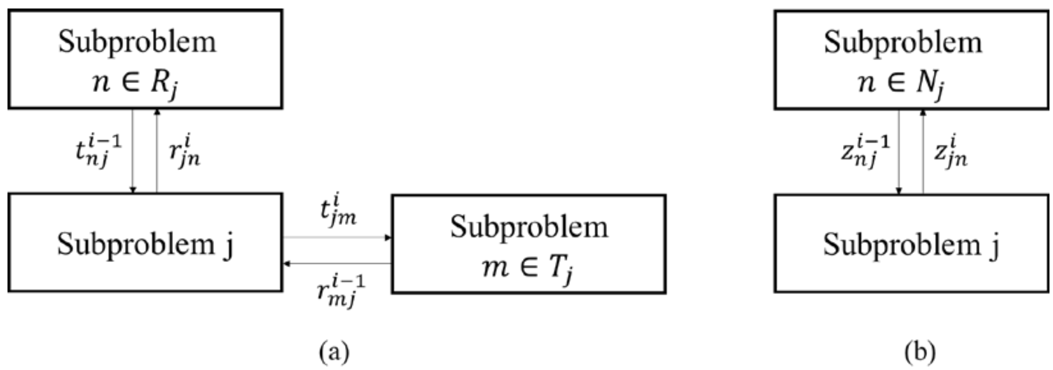

Fig 3 The inner loop of modified AL-AD with parallelization using (a) notations in Tosserams et al. (2010) and (b) linking variables in this paper

3.2 Formulation of Modified AL-AD with Parallelization in Non-hierarchical ATC

One of the most widely used coordination algorithms for non-hierarchical ATC is AL-AD using a method of multipliers. Due to the presence of the quadratic penalty function, however, each coupled subproblem should be optimized consecutively. Hence, the proposed method uses the results of the previous iteration for parallelization, which are denoted as the superscript *i*-1 in the formulation. Through such a basic concept, more general non-hierarchical ATC formulation for parallelization is proposed with linking variable. If parallelization is implemented, the direction of communication which means a sequence of optimization is no longer needed in non-hierarchical ATC. Therefore, it is not necessary to distinguish two types of subproblems which transfer target or response. The only difference is that the responses are the output of functions of design variables and targets can be design variables itself. For simplicity, the vector of linking variables $\mathbf{z}$ is introduced which can be variables or response functions of variables, and include anything that is communicated with other subproblems. Consequently, linking variable $\mathbf{z}$ contains the target-response pair and shared variables of conventional ATC.

In this paper, the proposed modified AL-AD formulation with parallelization in non-hierarchical ATC of subproblem *j* is shown as

$$\begin{aligned} &\min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n \in N_j} \pi(\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) \text{ where } \pi(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \|\mathbf{w} \circ \mathbf{c}\|_2^2 \\ &\text{subject to } \mathbf{g}_j(\bar{\mathbf{x}}_j^{(i)}) \le 0,\ \mathbf{h}_j(\bar{\mathbf{x}}_j^{(i)}) = 0 \\ &\text{with } \mathbf{z}_{jn}^{(i)} = \mathbf{S}_{jn}\mathbf{a}_j(\bar{\mathbf{x}}_j^{(i)}) \text{ for } n \in N_j,\ \bar{\mathbf{x}}_j^{(i)} = [\mathbf{x}_j^{(i)}, \mathbf{L}_{jn}\mathbf{z}_{jn}^{(i)}\ \forall n \in N_j] \end{aligned} \tag{4}$$

The linking variable is used for simplicity of notation and $\mathbf{L}_{jn}$ is introduced which is a binary matrix for selecting design variables in the linking variables. The binary matrix is for identifying design variables because some types of linking variables are the output of functions calculated from design variables, while others are design variables itself which are directly involved in optimization. Hence, design variable $\bar{\mathbf{x}}_j$ consists of $\mathbf{x}_j$ which is local design variables only involved in subproblem *j* and $\mathbf{L}_{jn}\mathbf{z}_{jn}^{(i)}$ for $\forall n \in N_j$ which is selected from linking variables by $\mathbf{L}_{jn}$. The function vector $\mathbf{a}_j$ calculates all linking variables in subproblem *j* from design variables. $\mathbf{S}_{jn}$ is a binary matrix for selecting linking variables associated with subproblem *n*. The detailed description of the notations used in Eq. (4) is shown in Table 1.

Table 1 Detailed description of notations with linking variables in modified AL-AD with parallelization

Notation	Description
N_j	A set of subsystems which communicate with subsystem *j*
$\mathbf{z}_{nj}$	The linking variables calculated in optimization of subsystem *n* which is transferred to subsystem *j*
$\mathbf{z}_{jn}$	The linking variables calculated in optimization of subsystem *j* which is transferred to subsystem *n*
$\bar{\mathbf{x}}_j$	Design variables of subsystem *j*
$\mathbf{x}_j$	Local design variables of subsystem *j*
n_{N_j}	The number of elements in a set N_j
n_{jn}	The number of linking variables between subsystem *j* and subsystem *n*

n_{jn}^{v}	The number of design variables in linking variables between subsystem *j* and subsystem *n*
n_j	The number of all linking variables in subsystem *j*
$\mathbf{L}_{jn}$	$n_{jn}^{v} \times n_{jn}$ binary matrix for selecting design variable in the linking variable $\mathbf{z}_{jn}$
$\mathbf{S}_{jn}$	$n_{jn} \times n_j$ binary matrix for selecting linking variable associated with subsystem *n*
$\mathbf{a}(\overline{\mathbf{x}}_j)$	$n_j \times 1$ column vector of the linking variable in subsystem *j*
$\mathbf{g}_j(\overline{\mathbf{x}}_j)$	Local inequality constraints of subsystem *j*
$\mathbf{h}_j(\overline{\mathbf{x}}_j)$	Local equality constraints of subsystem *j*
$f_j(\overline{\mathbf{x}}_j)$	Local objective function of subsystem *j*

Each subproblem uses previous optimization results as parameters which are the linking variables calculated in other subproblems. Therefore, all subproblems can be separated in the current iteration. The target values from the upper-level in non-hierarchical ATC denoted as $\mathbf{t}_{nj}^{(i)}$ are converted to $\mathbf{z}_{nj}^{(i-1)}$ for parallelization in Eq. (4) unlike Eq. (1). For better understanding of the formulation, assume a simple system which is decomposed into two subproblems as shown in Figure 4.

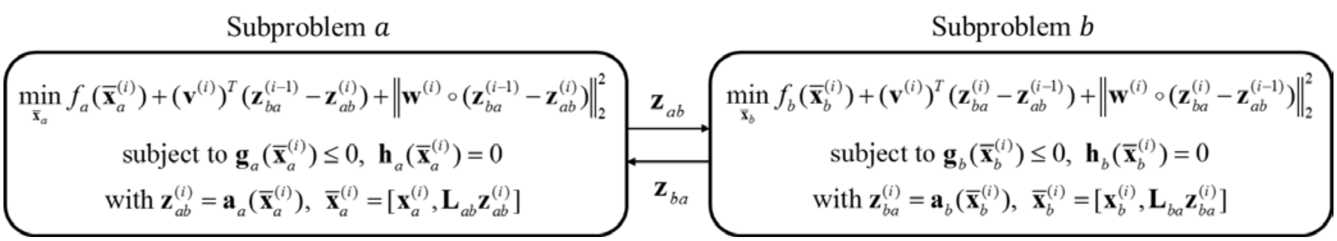

Fig 4 Coordination of modified AL-AD with parallelization in non-hierarchical ATC between two subproblems

As mentioned in the previous section, the similar concept of formulation in the inner loop optimization is already proposed as DQA (Li et al. 2008). However, the DQA method applies the approximation based on Taylor expansion at the target value of the previous iteration only in the cross term, because the Lagrangian function is constant and does not affect the optimum. The objective function of DQA can be formulated as

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

However, DQA, especially TDQA (the single-loop version of DQA), has numerical instability in non-hierarchical ATC and is sensitive to parameters (e.g., step size of the design progress). As it is based on linearization, it is locally accurate only near that point, and also its accuracy depends on nonlinearity of a function. Sometimes, the Lagrange multiplier diverges without a small step size of design progress for approximation. This phenomenon can be attributed to the outer loop, which updates the Lagrange multiplier, penalty weight, and a step size of the design progress. It shows good convergence in parallel solving with conservative parameters, but convergence speed is slow.

Thus, in the modified AL-AD with parallelization, the objective function is expressed as

$$\min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n \in N_j} ((\mathbf{v}_{nj}^{(i)})^T (\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) + \left\| \mathbf{w}_{nj}^{(i)} \circ (\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) \right\|_2^2) \quad (6)$$

Unlike DQA, all linking variables from other subproblems are the results of the previous iterations in the proposed method. Obviously, each optimization in the inner loop has the same optimum because $(\mathbf{v}_{nj}^{(i)})^T \mathbf{z}_{nj}^{(i-1)}$ is constant during the optimization of subproblem j. However, this modification makes a difference in the convergence of the Lagrange multiplier in the outer loop using the subgradient method. In the proposed method, we use the subgradient method to update the Lagrange multiplier with an adaptive step size rather than a method of multipliers specialized in sequential solving (Boyd et al. 2003, Kim et al. 2006). The proposed method intentionally delays information which are linking variables from other subproblems for parallelization. Therefore, the modified Lagrange multiplier and penalty weight update algorithm is needed to achieve the convergence of ATC.

3.3 Modified Lagrange Multiplier and Penalty Weight Update Scheme with Parallelization

We propose a new approach for parallelization in non-hierarchical ATC by using the results of a previous iteration. Without properly updating the Lagrange multiplier, however, just using the formulation of Eq. (4) can cause oscillation of the solution and divergence of inconsistency due to the instability of the Lagrange multiplier. The main difficulty of this formulation is finding a way to treat the divergence of the Lagrange multiplier, which begins with the mismatch of information. This problem does not have a significant effect in the early stages of ATC. However, when the Lagrange multiplier converges to some extent, inconsistency cannot lead to convergence under specified tolerance and can result in divergence in the absence of very conservative parameters.

Thus, we modify the definition of inconsistency after a single inner loop iteration. In the modified AL-AD with parallelization, the delayed inconsistency $\tilde{\mathbf{c}}_{nj,j}^{(i)}$ should be defined differently considering the delay of information as

$$\tilde{\mathbf{c}}_{nj,j}^{(i)} = \mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)} \tag{7}$$

Obviously, the delayed inconsistency expressed in Eq. (7) is not used in a criterion to determine the convergence of the outer loop and added subscript *j* means that inconsistency is computed in subproblem *j*. If it is computed in the corresponding subproblem *n*, the delayed inconsistency should be $\tilde{\mathbf{c}}_{nj,n}^{(i)} = \mathbf{z}_{nj}^{(i)} - \mathbf{z}_{jn}^{(i-1)}$. The convergence of the whole system is determined based on the conventional consistency between the current target value and the response value shown as

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

Thus, the objective of the whole algorithm is to make two kinds of inconsistency close to zero. Eq. (7) represents the delayed inconsistency used in updating the Lagrange multiplier which is denoted as $\mathbf{v}_{nj}^{(i)}$ with the subgradient method which is obtained from Eq. (6).

Based on the Lagrangian duality theory, the dual function of primal objective function with consistency constraints is concave regardless of convexity of the objective function because it is pointwise infimum of a family of affine functions of the Lagrange multiplier (Boyd et al. 2004). Hence, the maximization of the concave function is equivalent to the minimization of the convex problem, so we use the subgradient method to find the optimum Lagrange multiplier of the dual function. The objective function in the dual problem of subproblem *j* regarding consistency constraints in the proposed method is expressed as

$$\max_{\mathbf{v}_{nj}} \phi_j(\mathbf{v}_{nj})$$
$$\text{where } \phi_j(\mathbf{v}_{nj}) = \min_{\bar{\mathbf{x}}_j} f_j(\bar{\mathbf{x}}_j^{(i)}) + \sum_{n \in N_j} ((\mathbf{v}_{nj}^{(i)})^T (\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) + \left\| \mathbf{w}_{nj}^{(i)} \circ (\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) \right\|_2^2) \tag{9}$$

In Eq. (9), the duality theory is implemented in the augmented Lagrangian function, and the local constraints and other design parameters are based on Eq. (4). Thus, the subgradient of $\phi_j(\mathbf{v}_{nj})$ with respect to $\mathbf{v}_{nj}^{(i)}$ which is $\frac{\partial \phi_j}{\partial \mathbf{v}_{nj}^{(i)}}$ is calculated as $\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}$.

For better convergence of the Lagrange multiplier, however, we take into account the bilateral results of corresponding subproblems which are connected by linking variables. Assuming a simple system as in Figure 4, the Lagrange multiplier corresponding to the vectors of linking variable pair denoted as $(\mathbf{z}_{ba}, \mathbf{z}_{ab})$ should be updated with the subgradients of each subsystem. That is, the delayed inconsistency in subsystem *a* which is $\mathbf{z}_{ba}^{(i-1)} - \mathbf{z}_{ab}^{(i)}$ and the delayed inconsistency in subsystem *b* which is $\mathbf{z}_{ba}^{(i)} - \mathbf{z}_{ab}^{(i-1)}$ are computed simultaneously in the outer loop when updating the Lagrange multiplier with the subgradient method. Figure 5 illustrates the difference between the unilateral updating method and the bilateral updating method. The horizontal axis indicates the outer loop iteration and the corresponding points of linking variables are results of optimization of two subproblems in parallel. The red and blue arrows show the delayed inconsistency of each subproblem.

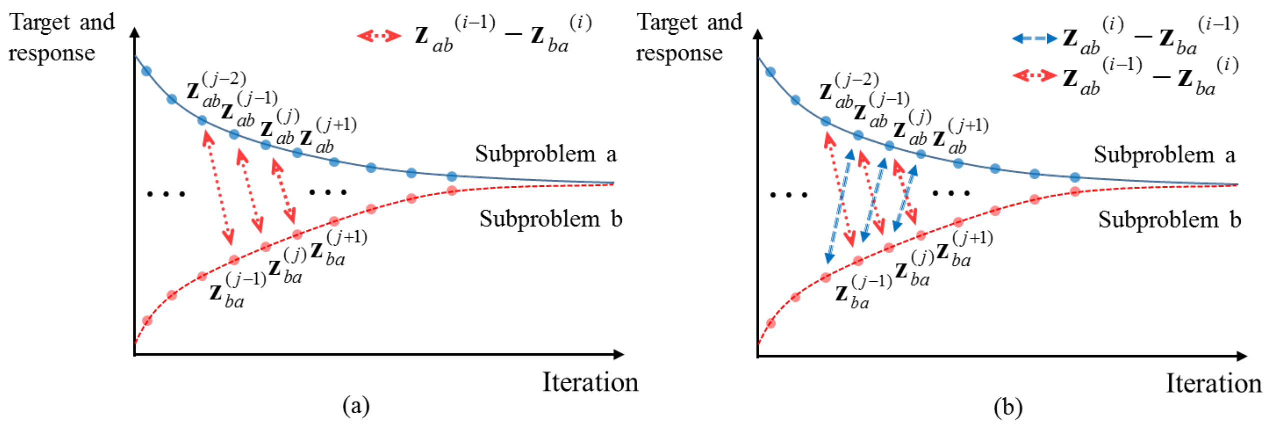

Fig 5 Illustration of the delayed inconsistency used in the updating the Lagrange multiplier considering (a) the unilateral results and (b) the proposed bilateral results.

The subgradient of the Lagrange multiplier should be the bilateral inconsistencies of corresponding subproblems, and therefore we propose a modified Lagrange multiplier updating algorithm with the subgradient method considering bilateral inconsistencies as

$$\mathbf{v}_{nj}^{(i+1)} = \mathbf{v}_{nj}^{(i)} + \boldsymbol{\alpha}_{nj}^{(i)} \circ (\mathbf{n}_j(\mathbf{z}_{nj}^{(i-1)} - \mathbf{z}_{jn}^{(i)}) + (\mathbf{1} - \mathbf{n}_j)(\mathbf{z}_{nj}^{(i)} - \mathbf{z}_{jn}^{(i-1)})) \quad \text{where } \mathbf{0} < \mathbf{n}_j < \mathbf{1} \tag{10}$$

The vector $\mathbf{n}_j$ is the vector of weight factors between two delayed inconsistencies from corresponding subproblems, and we set it to 0.5 in all problems, representing the average of two delayed inconsistencies from corresponding subproblems. The step size vector $\boldsymbol{\alpha}_{nj}^{(i)}$ will be discussed in the next section. More detailed description of the dual coordination in ATC and the subgradient method are discussed in Lassiter et al. (2005) and Kim et al. (2006).

The important thing is that the proposed Lagrange multiplier update algorithm uses delayed inconsistencies as the direction when updating the Lagrange multiplier. However, the convergence of ATC depends on the inconsistency between the current linking variables. In other words, the convergence criterion of the outer loop is based on the inconsistency between $\mathbf{z}_{nj}^{(i)}$ and $\mathbf{z}_{jn}^{(i)}$, not $\mathbf{z}_{nj}^{(i-1)}$ and $\mathbf{z}_{jn}^{(i)}$. Even if the proper subgradient with bilateral delayed inconsistencies is used, the oscillation of the Lagrange multiplier and the current solution may occur due to the improper quadratic penalty function.

The oscillation of conventional AL-AD in parallel solving is related to the crossing of information, inadequate step size $\boldsymbol{\alpha}_{nj}^{(i)}$ used in updating the Lagrange multiplier, and the excessive quadratic penalty function. In conventional AL-AD, a method of multipliers is used to update the Lagrange multiplier, so the step size vector is defined as $2\mathbf{w}_{nj}^{(i)} \circ \mathbf{w}_{nj}^{(i)}$, which has great convergence properties in the sequential solving of each subproblem with increasing penalty weight using Eq. (3). In parallel solving, however, it is not quite helpful, because increasing penalty weight and step size make the quadratic penalty function too dominant in the objective function and also make step size too large to reach the optimum Lagrange multiplier, thus causing the solution to oscillate or even diverge. Thus, we propose a guideline for penalty weight to find the optimum Lagrange multiplier and prevent oscillation considering the scale of Lagrange multiplier and linking variable as

$$\begin{cases} w_{nj,k}^{(i+1)} = \max(1, \sqrt{\left| v_{nj,k}^{(i+1)} \circ \dfrac{1}{2z_{nj,k}^{(i)}} \right|}),\ k = 1, 2, ..., n_{nj} & \text{if phase1} \\ w_{nj,k}^{(i+1)} = w_{nj,k}^{(i)},\ k = 1, 2, ..., n_{nj} & \text{if phase2} \end{cases} \tag{11}$$

where $\mathrm{w}_{nj,k}^{(i+1)}$, $v_{nj,k}^{(i+1)}$, and $z_{nj,k}^{(i)}$ are k-th elements of $\mathbf{w}_{nj}^{(i+1)}$, $\mathbf{v}_{nj}^{(i+1)}$, and $\mathbf{z}_{nj}^{(i)}$ respectively.

In the proposed method, the quadratic penalty function is only required to prevent the objective function from being unbounded and to make the objective function convex. The use of the penalty weight does not directly affect finding the optimum Lagrange multiplier but affects the results of each optimization in subproblems. The phase 1 and the phase 2 are discussed in Section 3.4. The reason for introducing a lower bound is that too small penalty weight may not play a role of the quadratic penalty function itself. Also, the value $\sqrt{\left| v_{nj,k}^{(i+1)} \circ \dfrac{1}{2z_{nj,k}^{(i)}} \right|}$ is calculated from the augmented Lagrangian function $\phi(z_{ab}) = v(z_{ab} - z_{ba}) + w^2(z_{ab} - z_{ba})^2$. It is a quadratic function with respect to z_{ab} and the optimum of the augmented Lagrangian function itself is $\frac{2w^2 z_{ba} - v}{2w^2} = z_{ba} - \frac{v}{2w^2}$. If the weight is too large, the quadratic penalty function should be dominant in the objective function. Hence, we set a guideline to maintain the weight in similar scale with the Lagrange multiplier and linking variable as Eq. (11).

3.4 Adaptive Step Size Updating Algorithm

The step size updating algorithm plays a major role in the convergence of ATC. In the proposed method, this algorithm is the key to the successful convergence and accuracy of the Lagrange multiplier and solution. In updating the Lagrange multiplier, the subgradient and the step size mean the direction and the magnitude in updating the Lagrange multiplier, respectively. The existing step size of the subgradient method, such as constant or non-summable diminishing (Boyd et al. 2003), is not efficient for finding the optimum Lagrange multiplier.

Thus, we propose an efficient step size updating algorithm for parallel solving. The process of updating step size is based on the path of the Lagrange multiplier, and thus it derives the best step size considering the current value of Lagrange multiplier. The role of the step size in updating the Lagrange multiplier with the subgradient method is simple: if the convergence rate of the Lagrange multiplier is slow, then the step size should be increased. On the contrary, if the oscillation occurs, the step size should be decreased so that the Lagrange multiplier can converge to an optimum.

Two phases are involved in the adaptive step size updating algorithm. In phase 1, the Lagrange multiplier is considered to be far from the optimum, and in phase 2, the Lagrange multiplier is deemed to be near the optimum. The criterion for the transition from phase 1 to phase 2 is the change of the direction of the Lagrange multiplier. In the proposed algorithm, previous three values of the Lagrange multiplier are used in the criterion of the phase expressed as

$$\begin{aligned} &(v_k^{(i-2)} - v_k^{(i-1)})(v_k^{(i-1)} - v_k^{(i)}) > 0 : \text{Phase 1} \\ &(v_k^{(i-2)} - v_k^{(i-1)})(v_k^{(i-1)} - v_k^{(i)}) < 0 : \text{Phase 2} \qquad (12) \\ &\text{where } v_k^{(i)} \text{ is the } k\text{-th element of } \mathbf{v}^{(i)} \quad k = 1, \ldots, n_v \end{aligned}$$

If the criterion is satisfied such that the signs of differences between two consecutive values of the Lagrange multiplier are changed, this means that the Lagrange multiplier has passed the optimum. Thus, phase 1 is switched to phase 2 and does not return to phase 1.

The adaptive step size updating algorithm has different update schemes depending on the phase. In phase 1, the Lagrange multiplier is far from the optimum. Therefore, the speed of convergence to the optimum is more important than accuracy. Therefore, the change of Lagrange multiplier is the performance indicator. The process of updating the step size in phase 1 is expressed as

$$\alpha_k^{(i+1)} = \begin{cases} \alpha_k^{(i)} & \text{if } (v_k^{(i-1)} - v_k^{(i)}) \geq \eta \\ \beta_1 \alpha_k^{(i)} & \text{otherwise} \end{cases} \qquad k = 1, \ldots, n_v \tag{13}$$

In the equation above, β_1 is the increasing factor of step size, and η is the reference parameter for satisfying the criterion. Typically, $\beta_1 = 1.1$ is recommended to speed up, and η is dependent on the scale of the Lagrange multiplier and the user's preference. Unlike common step size of the subgradient method, the step size is increased for convergence rate in phase 1.

In phase 2, which is switched from phase 1 based on Eq. (12), the Lagrange multiplier is close to the optimum so the large step size is no longer needed. Rather, the decreasing step size can prevent oscillation between linking variables from two corresponding problems in parallel solving, thereby ensuring the convergence of Lagrange multiplier. The occurrence of the oscillation means that the optimum is exceeded and the sign of inconsistency may be changed. Hence, the change of the direction of Lagrange multiplier (rather than a change of Lagrange multiplier) is the performance indicator that can signify the oscillation. In this case, the step size is updated as

$$\alpha_k^{(i+1)} = \begin{cases} \alpha_k^{(i)} & \text{if } (v_k^{(i-2)} - v_k^{(i-1)})(v_k^{(i-1)} - v_k^{(i)}) \geq 0 \\ \beta_2 \alpha_k^{(i)} & \text{otherwise} \end{cases} \qquad k = 1, \ldots, n_v \tag{14}$$

In the equation above, β_2 is the decreasing factor of step size and typically, $\beta_2 = 0.9$ is recommended for good convergence. However, the increasing factor or decreasing factor does not affect the performance of algorithm so much and it is demonstrated with numerical examples in the next section.

The direction change of the Lagrange multiplier 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. The modified AL-AD with parallelization in non-hierarchical ATC is given by the following algorithm as

1. Set $\overline{\mathbf{x}}$, $\mathbf{v}$, $\mathbf{w}$ and $i = 0$.

2. For each subproblem, solve for $\overline{\mathbf{x}}_j$ in Eq. (4) in parallel and obtain $\overline{\mathbf{x}}_j^{(i)}$.

3. If Eqs. (16) and (17) are satisfied with a user-specified tolerance, then set $\overline{\mathbf{x}}_j^{(i)}$ to be optimum. Otherwise, go to Step 4.

4. For each Lagrange multiplier of consistency constraint, if the Lagrange multiplier is in phase 1, calculate Eq. (12). If it is still in phase 1, update the step size using Eq. (13). Otherwise, update step size using Eq. (14).

5. Update the Lagrange multiplier and the penalty weight using Eqs. (10) and (11).

6. Set $i = i + 1$ and go back to Step 2.

The flowchart for the algorithm explained above is shown in Figure 6.

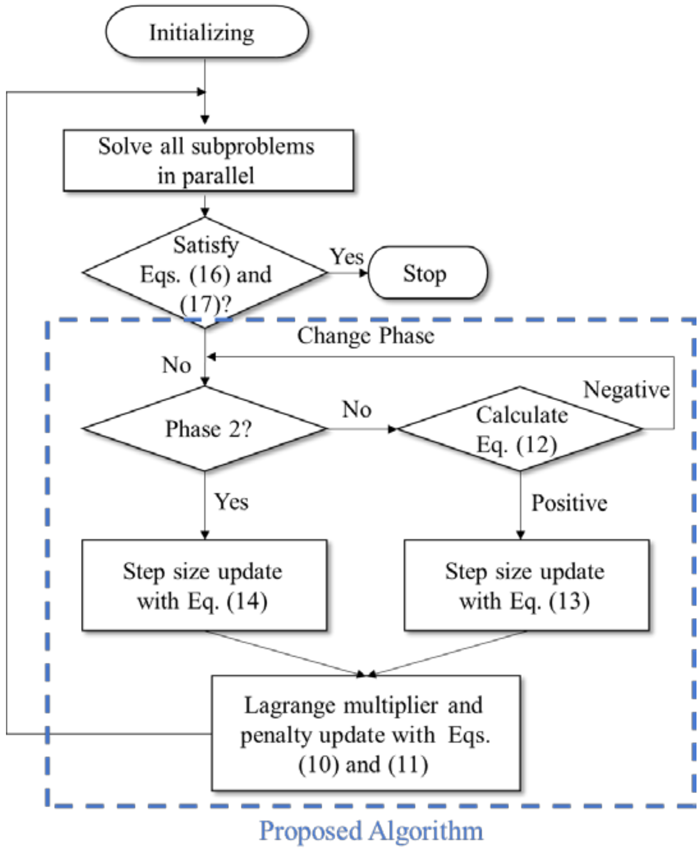

Fig 6 Flowchart of modified AL-AD with parallelization

4. Numerical Study

In this section, several mathematical problems including a fourteen-variable nonconvex geometric programming problem that is widely used in many ATC research papers are solved using the modified AL-AD with parallelization and conventional algorithm for comparison including AL-AD and TDQA, which are widely used as a benchmark in ATC. Performance indicators and termination criteria need to be defined to evaluate the proposed method. In this paper, three indicators are used for quantifying the performance of each method, namely, solution error, total function evaluation to converge, and computational time which is latency.

The latency measures the total CPU time required to evaluate the function in the whole algorithm, considering parallelization. Solution error is defined as the accuracy of a solution obtained from ATC compared with known optimal solution. Solution error e widely used in the literature is defined as

$$e = \left\| \mathbf{x}^{*} - \mathbf{x}^{\mathrm{ATC}} \right\|_{\infty}, \tag{15}$$

where $\mathbf{x}^{*}$ is the known optimal solution, and $\mathbf{x}^{\mathrm{ATC}}$ 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 expressed as

$$\left\| \frac{(\mathbf{z}_{jk}^{(i)} - \mathbf{z}_{kj}^{(i)}) - (\mathbf{z}_{jk}^{(i-1)} - \mathbf{z}_{kj}^{(i-1)})}{(\mathbf{z}_{jk}^{(i-1)} - \mathbf{z}_{kj}^{(i-1)})} \right\|_{\infty} < \tau_{outer} \tag{16}$$

and

$$\left\| \mathbf{z}_{jk}^{(i)} - \mathbf{z}_{kj}^{(i)} \right\|_{\infty} < \tau_{outer}, \tag{17}$$

respectively. The (j,k) denotes all pairs of subproblems which have linking variables in Eqs. (16) and (17).

The total function evaluation and latency are evaluated using Matlab R2016b (Mathworks 2016) when the termination criterion is satisfied with respect to each of tolerance, $\tau_{outer} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. Each optimization of the subproblem is conducted with Sequential Quadratic Programming (SQP) algorithm (fmincon) implemented in Matlab R2016b. All parameters including tolerance in SQP algorithm are default values of Matlab R2016b.

4.1 A Seven-variable Nonlinear Constrained Optimization Problem

The formulation of this problem is given by

$$
\begin{aligned}
\min_{x_1,\cdots,x_7} \quad & f = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 \\
& +10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\
\text{subject to} \quad & \mathrm{g}_1 = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0 \\
& \mathrm{g}_2 = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0 \\
& \mathrm{g}_3 = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0 \\
& \mathrm{g}_4 = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0 \\
\text{where} \quad & -10 \le x_i \le 10, \quad i = 1,\ldots,7
\end{aligned}
\tag{18}
$$

where at the optimum point $\mathbf{x}^*$ = [2.331, 1.951, -0.478, 4.366, -0.625, 1.038, 1.594] and objective function f^* = 680.63. This optimization problem is non-convex optimization since Hessian of the objective function which is continuous and twice differentiable function is not positive semidefinite (Montes et al. 2005, DorMohammadi et al. 2013).

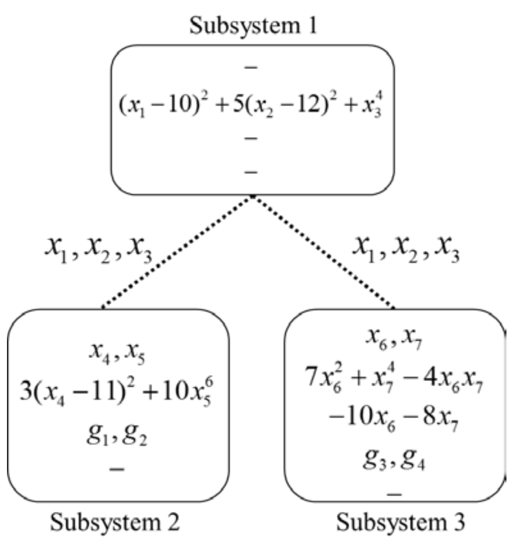

Fig 7 Decomposition details for the seven-variable non-convex problem

Note: Each box represents a subproblem and the connections between the subproblems represent the linking variables. In the box, it means local variables, objective function, inequality constraints and equality constraints from top to bottom.

The system can be decomposed hierarchically as Figure 7 into three subproblems, and each of them may have its own objective function, constraints, and variables. There are three linking variables between subproblems related to consistency constraints. Given that modified AL-AD with parallelization has a single-loop algorithm, the decomposed problem is also solved using AL-AD and TDQA for comparison. Other methods such as QP, OL, AL, enhanced AL, and DQA have proven to be similar or worse than AL-AD and TDQA in several examples of the literature in terms of performance including computational cost.

In AL-AD, TDQA, and the modified AL-AD with parallelization, the initial parameters are set as $\mathbf{v}^{(0)} = [0,0,\ldots,0]$ and $\mathbf{w}^{(0)} = [1,1,\ldots,1]$. The initial point is $\mathbf{x}^{(0)} = [0,0,\ldots,0]$ for all methods. For AL-AD and TDQA, the penalty weight update parameter is set as β = 1. In the adaptive step size updating algorithm, the parameters are set as $\beta_1 = 1.1, \beta_2 = 0.9, \eta = 1.5$. For a fair comparison, the step sizes of the design progress (Li et al. 2008) in TDQA is 0.8, which shows the best performances in this example.

Figure 8 shows the numerical results of each method that solution error, function evaluation, and latency to reach specified tolerance of the outer loop criterion. Table 2 shows the robustness of the proposed method with respect to changes in parameters. The total function evaluation is slightly higher than conventional AL-AD, but the latency is much smaller than conventional AL-AD and TDQA. This means that if parallelization is implemented, each subproblem can be solved in parallel, hence it is not necessary to wait for optimization of other subproblems to receive information of current iteration which is linking variable. It also shows better performance than TDQA which is the existing ATC coordination algorithm with parallelization. In Table 2, with changes in parameters, the proposed algorithm still shows better performance than conventional ones.

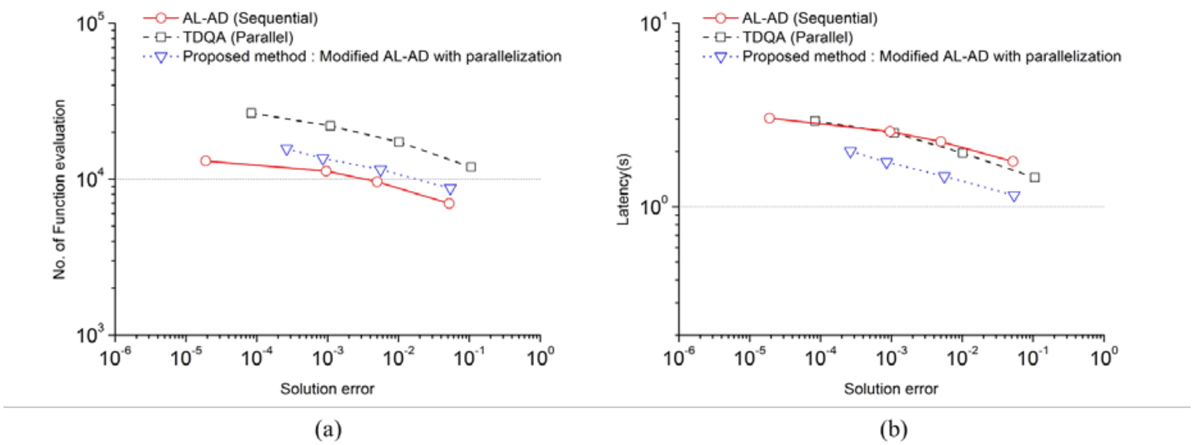

Fig 8 (a) Total function evaluation and (b) latency versus solution error in the seven-variable problem

Table 2 Numerical results of the seven-variable non-convex problem with various parameters

	Tolerance	No. of function evaluations	Solution error	Latency(s)
$(\eta,\beta_1,\beta_2)=(1.5,1.1,0.9)$ – Reference results used in Figure 8				
	0.01	8760	0.054	1.157
	0.001	11525	0.0056	1.466
	0.0001	13615	8.56E-04	1.751
	0.00001	15173	2.65E-04	2.007
$(\eta,\beta_1,\beta_2)=(1.5,1.1,0.7)$				
	0.01	8761	0.0682	1.173
	0.001	12369	0.0052	1.577
	0.0001	15038	8.67E-04	1.936
	0.00001	15803	5.10E-04	2.045
$(\eta,\beta_1,\beta_2)=(3,1.1,0.9)$				
	0.01	7852	0.058	1.093
	0.001	10844	0.0048	1.397
	0.0001	13413	4.82E-04	1.732
	0.00001	13483	4.49E-04	1.761
$(\eta,\beta_1,\beta_2)=(1.5,1.3,0.9)$				
	0.01	11464	0.0029	1.472
	0.001	12473	0.0016	1.571
	0.0001	13376	3.55E-04	1.713
	0.00001	13911	3.36E-05	1.832

4.2 A Ten-variable Nonlinear Constrained Optimization Problem

This problem is formulated as

$$
\begin{aligned}
\min_{x_1,\cdots,x_7} \quad & f = x_1^{\,2} + x_2^{\,2} + x_1x_2 - 14x_1 - 16x_2 + 45 \\
& + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
& + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 \\
\text{subject to} \quad & \mathrm{g}_1 = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0 \\
& \mathrm{g}_2 = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\
& \mathrm{g}_3 = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\
& \mathrm{g}_4 = -3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^{\,2} - 7x_4 - 120 \le 0 \\
& \mathrm{g}_5 = 5x_1^{\,2} + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0 \\
& \mathrm{g}_6 = x_1^{\,2} + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0 \\
& \mathrm{g}_7 = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^{\,2} - x_6 - 30 \le 0 \\
& \mathrm{g}_8 = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0 \\
\text{where} \quad & -10 \le x_i \le 10 \qquad i = 1,\ldots,10
\end{aligned}
\tag{19}
$$

with the optimum point $\mathbf{x}^*$ = [2.172, 2.364, 8.774, 5.095, 0.991, 1.431, 1.322, 9.829, 8.280, 8.376] and objective function f^* = 24.306 (Montes et al. 2005, DorMohammadi et al. 2013). This is a convex problem because all objective functions and constraints have positive definite Hessian matrices. This problem can be decomposed into four subproblems as shown in Figure 9. The format of structure in Figure 9 is the same as format in Figure 7 as mentioned before. All parameters and initial points are also the same as the previous problem for reasonable comparison. For TDQA, we take the step size of design progress as 0.7 which shows the best performance in this example. Figure 10 shows the numerical results of each method with solution error, function evaluation, and latency. In addition, Table 3 shows the robustness of the proposed method with respect to changes in parameters. Similar with the previous mathematical problems, the proposed algorithm is the most efficient algorithm in terms of latency which is most significant performance index in parallel solving.

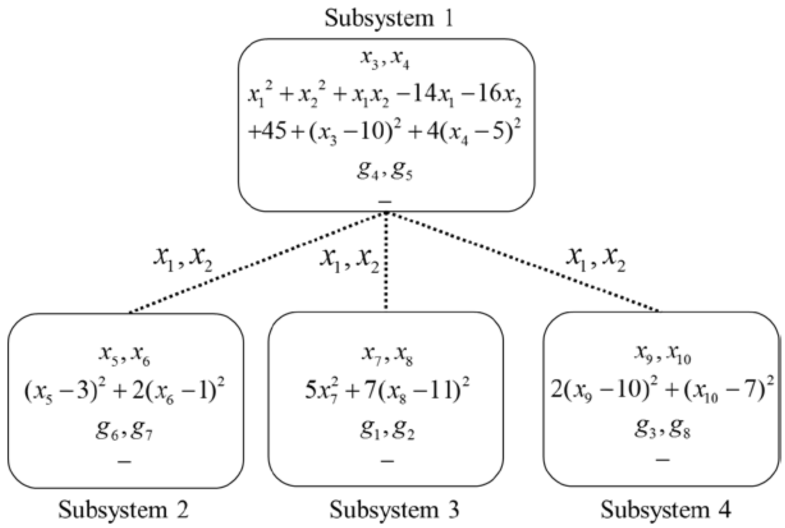

Fig 9 Decomposition details for the ten-variable convex problem

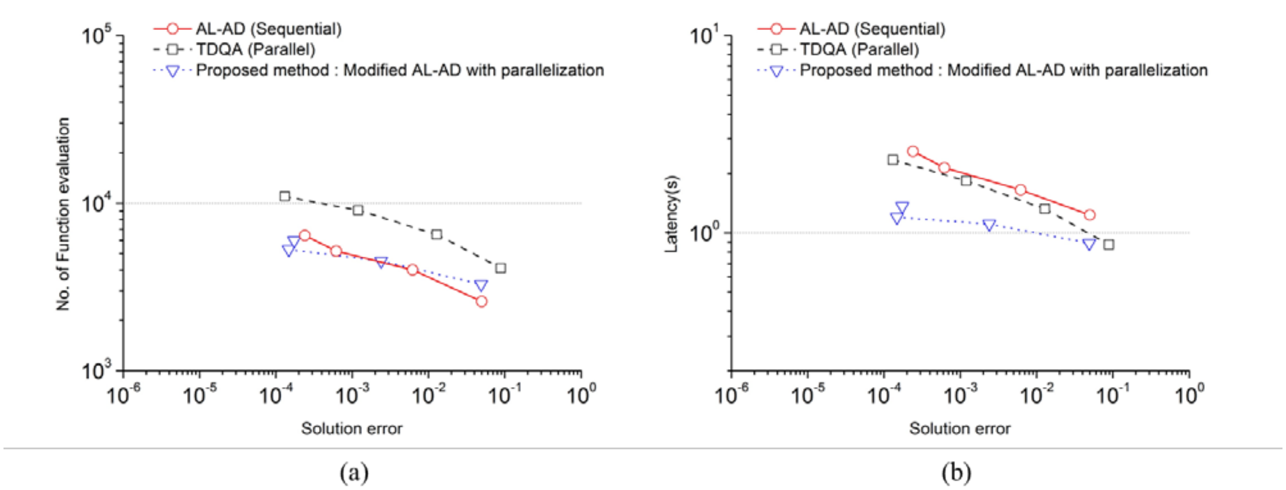

Fig 10 (a) Total function evaluation and (b) latency versus solution error in the ten-variable problem

Table 3 Numerical results of the ten-variable convex problem with various parameters

	Tolerance	No. of function evaluations	Solution error	Latency(s)
$(\eta, \beta_1, \beta_2) = (1.5, 1.1, 0.9)$ – Reference results used in Figure 10				
	0.01	3287	0.049	0.887
	0.001	4522	0.0024	1.111

	0.0001	5304	1.48E-04	1.201
	0.00001	5994	1.74E-04	1.366
$(\eta, \beta_1, \beta_2) = (1.5, 1.1, 0.7)$				
	0.01	3227	0.051	0.871
	0.001	4519	0.0023	1.105
	0.0001	5309	1.39E-04	1.247
	0.00001	6235	7.13E-05	1.481
$(\eta, \beta_1, \beta_2) = (3, 1.1, 0.9)$				
	0.01	3287	0.049	0.944
	0.001	4522	0.0024	1.105
	0.0001	5304	1.48E-04	1.267
	0.00001	5994	1.74E-04	1.405
$(\eta, \beta_1, \beta_2) = (1.5, 1.3, 0.9)$				
	0.01	2958	0.015	0.846
	0.001	3861	8.89E-04	1.009
	0.0001	4671	2.27E-04	1.170
	0.00001	5138	1.98E-04	1.275

4.3 A Fourteen-variable Geometric Programming Problem

This example, posynomial geometric programming, has various kinds of decompositions widely used in ATC literature to demonstrate efficiency and feasibility of methods (Kim et al. 2001, Tosserams et al. 2007). A fourteen-variable nonlinear constrained optimization problem is formulated as

$$
\begin{aligned}
&\min_{z_1,\cdots,z_{14}} && f = z_1^{2} + z_2^{2} \\
&\text{subject to} && g_1 = (z_3^{-2} + z_4^{2}) z_5^{-2} - 1 \le 0 \\
& && g_2 = (z_5^{2} + z_6^{-2}) z_7^{-2} - 1 \le 0 \\
& && g_3 = (z_8^{2} + z_9^{2}) z_{11}^{-2} - 1 \le 0 \\
& && g_4 = (z_8^{-2} + z_{10}^{2}) z_{11}^{-2} - 1 \le 0 \\
& && g_5 = (z_{11}^{2} + z_{12}^{-2}) z_{13}^{-2} - 1 \le 0 \\
& && g_6 = (z_{11}^{2} + z_{12}^{-2}) z_{14}^{-2} - 1 \le 0 \\
& && h_1 = (z_3^{2} + z_4^{-2} + z_5^{2}) z_1^{-2} - 1 = 0 \\
& && h_2 = (z_5^{2} + z_6^{2} + z_7^{2}) z_2^{-2} - 1 = 0 \\
& && h_3 = (z_8^{2} + z_9^{-2} + z_{10}^{-2} + z_{11}^{2}) z_3^{-2} - 1 = 0 \\
& && h_4 = (z_{11}^{2} + z_{12}^{2} + z_{13}^{2} + z_{14}^{2}) z_6^{-2} - 1 = 0 \\
&\text{where} && z_i \ge 0 \qquad i = 1,\ldots,14
\end{aligned}
\tag{20}
$$

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]. We solved the problem according to two decompositions shown in Figure 11 (Tosserams et al. 2007), which consist of three and five non-hierarchical subproblems, respectively. Unlike the previous examples of hierarchical subproblems, each subproblem connected by dotted line can transfer linking variable to any other subproblems regardless of level, such as $\mathbf{z}_5$ and $\mathbf{z}_{11}$.

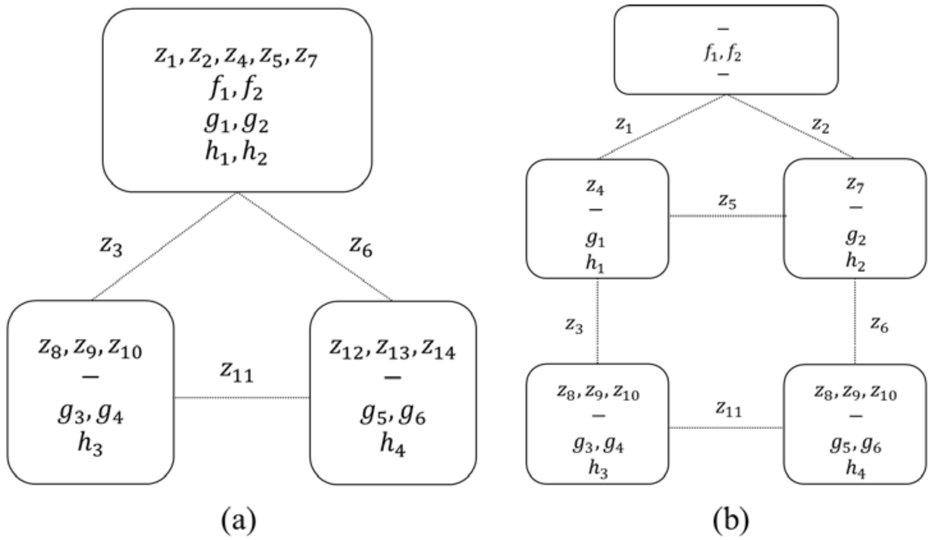

Fig 11 Two decompositions for the geometric optimization problem (Tosserams et al. 2007).

One of the advantages of the proposed method compared to TDQA is its robustness on the step size of the design progress which is distinct from the step size of the Lagrange multiplier. Since the proposed method is not based on approximation, the step size of design progress is not needed like TDQA. Decomposition (b) illustrated in Figure 11 has more linking variables than decomposition (a), and the non-hierarchical structure may cause TDQA to diverge depending on the step size of the design progress or the step size of the Lagrange multiplier. All parameters are the same as in previous examples. In TDQA, if the step size of the design progress is larger than 0.8, it cannot be converged in case of decomposition (b). The total number of function evaluation and latency versus solution error is described in Table 4. For a fair comparison, the step sizes of the design progress in TDQA are 1 and 0.7 in each decomposition, which show the best performances in both cases of decomposition, respectively.

As can be seen in previous three mathematical examples, the proposed coordination algorithm shows better convergence properties in terms of CPU time (i.e., Latency) compared to other methods. The modified AL-AD is based on parallel solving, so it is even more efficient regarding computational time for convergence than the conventional sequential AL-AD when it is divided into many subsystems, such as in decomposition (b) with five subsystems. In other words, the proposed method is more effective than conventional sequential algorithm as the number of subsystems increases. Moreover, it outperforms the TDQA as the only existing method of parallelization without centralized coordination. In all examples, the proposed method demonstrates high efficiency even if the parameters are perturbed. To verify the feasibility when applying to an engineering problem, the roof assembly design problem is described and evaluated in the following section.

Table 4 Numerical results of the fourteen-variable non-hierarchical problem.

	Tolerance	No. of function evaluations	Solution error	Latency(s)
Decomposition (a) with 3 subproblems				
AL-AD (Sequential)				
	0.01	2961	0.0196	1.972
	0.001	4092	0.0019	2.438
	0.0001	5118	1.82E-04	2.992
	0.00001	5996	3.24E-06	3.623
TDQA (Parallel)				
	0.01	4759	0.0153	0.982
	0.001	8290	0.0017	1.267
	0.0001	10528	1.53E-04	1.563
	0.00001	11987	8.40E-06	1.882
Modified AL-AD with Parallelization (Proposed method)				
	0.01	3572	0.0018	0.741
	0.001	3860	0.0024	0.792
	0.0001	4656	2.63E-4	0.845
	0.00001	4885	1.07E-4	0.970
Decomposition (b) with 5 subproblems				
	Tolerance	No. of function evaluations	Solution error	Latency(s)
AL-AD (Sequential)				
	0.01	3733	0.0211	3.112
	0.001	5136	0.0021	3.932
	0.0001	6457	1.96E-04	4.737
	0.00001	7722	1.80E-05	5.529
TDQA (Parallel)				
	0.01	6948	0.01777	1.260
	0.001	9636	0.0017	1.471
	0.0001	12265	1.70E-04	1.916
	0.00001	14764	1.63E-05	2.249
Modified AL-AD with Parallelization (Proposed method)				

0.01	3969	0.0297	0.968
0.001	5461	0.0044	1.148
0.0001	7015	4.77E-04	1.380
0.00001	8665	3.86E-05	1.607

4.4 Engineering Example: A roof assembly problem

This engineering example is originally formulated by Kang et al. (2014b). The original problem is about a bus body structure design to determine the stiffness and mass design specifications for each body assembly. Since the original problem is non-hierarchical sequential ATC, we use the target-response notation in a result table to follow the notation of the original problem instead of linking variable. The objective of the problem is to minimize the mass of the body structure subject to some constraints (e.g., static and dynamic characteristics of structure). The design variables are the dimension of beams forming the roof and side assemblies and material properties of bus body.

In this paper, we have simplified the problem rather than solving whole large system. To demonstrate the feasibility of the proposed method in an engineering example, only a roof assembly with four types of 12 beams is used as in the model used in Kang et al. (2014b). The overall structure and decomposition details of a roof assembly can be seen in Figure 12 and Figure 13. The objective function is to minimize the mass of roof assembly and discrepancy between the target 1st and 2nd mode frequency which are given and 1st and 2nd mode frequency of optimized design. The constraints are for static stiffness. Static stiffness is considered by means of assembly displacements. For a roof assembly, displacements when a single force is applied at the center and a given node of roof assembly can be indicators of bending and torsion stiffness. Hence, the dimensions of four beams are optimized with respect to the objective function and the constraints as mentioned by using FEA and mathematical function evaluation. All responses of roof assembly are obtained by means of simulation model built using Radioss of Altair Engineering (Altair 2012), and mathematical processing including optimization of each subproblem is conducted by Matlab R2016b (Mathworks 2016). The optimization algorithm of subproblems is SQP implemented in Matlab R2016b and default values in Matlab R2016b are used for all parameters in SQP algorithm.

The initial values, target values, parameters, and optimization results are presented in Table 5. The target values of mode frequency in a roof assembly are given from an upper level, and target values of beams (e.g., cross-section) are given from a roof assembly. The design variables including the width, height, and thickness of each cross-section of beam are local design variables. More details about the bus structure model can be obtained from Kang et al. (2014b). Using the proposed method, we set parameters to $\eta = 1.5, \beta_1 = 1.1, \beta_2 = 0.9$, which are the same values set in three previous problems. We set $\mathbf{v}^{(0)} = 0$ and $\mathbf{w}^{(0)} = 1$.

Table 5 and Figure 14 shows all the results summary of the roof assembly problem solved by the proposed method. Figure 14 shows the convergence history of inconsistency between target and response. In the early stage, the inconsistency seems to oscillate a little because of improper parameters, but they converge to near zero. In Table 5, the total mass of a roof assembly is reduced from 491 kg to 437 kg, and the discrepancy between target frequency and frequency of a roof assembly is also reduced. Most importantly, the maximum inconsistency of linking variable which is target-response pair is decreased to 10^{-3} which means the successful convergence of proposed method. As a result, the feasibility of proposed method is demonstrated in an engineering example of the roof assembly design.

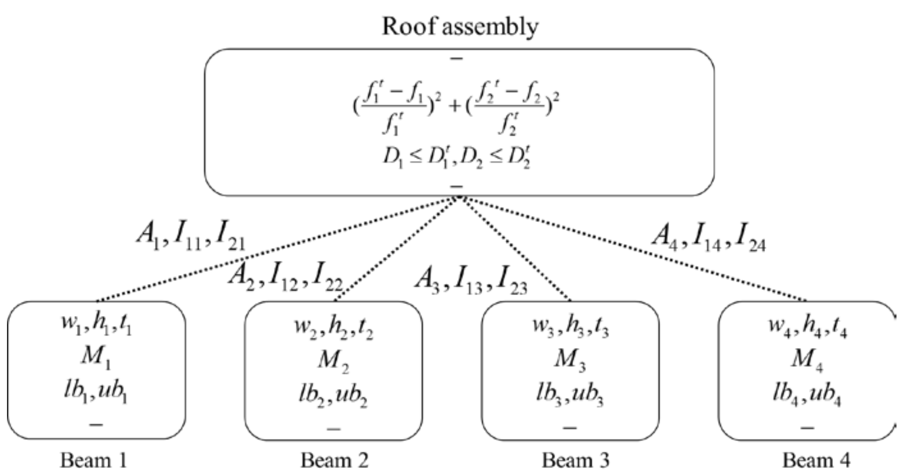

Fig 12 Decomposition details for a roof assembly design problem

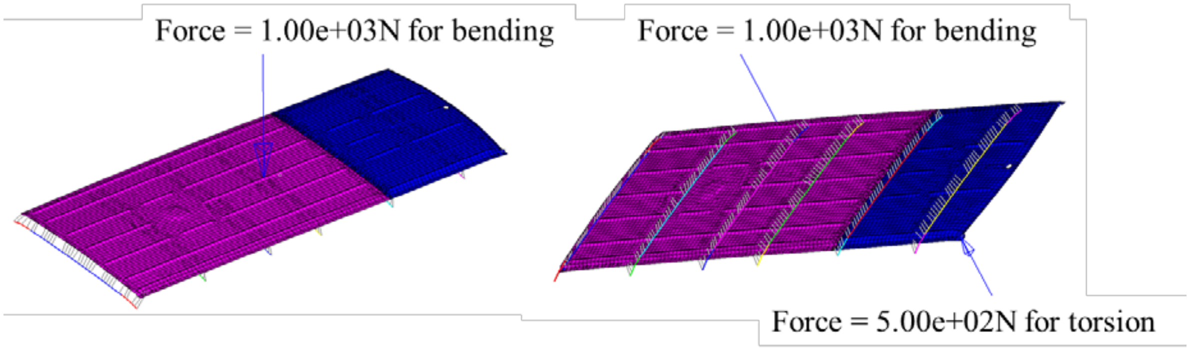

Fig 13 A roof assembly with 4 types of 12 beams and load conditions (Kang et al. 2014b)

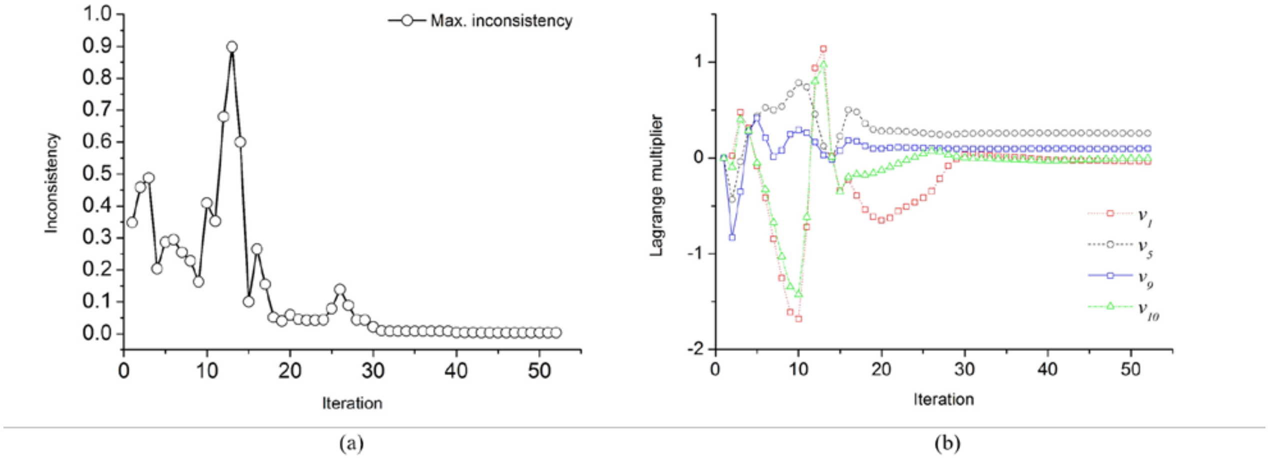

Fig 14 (a) Convergence history of inconsistency between target and response in a roof assembly design

(b) Convergence history of 4 out of 12 Lagrange multipliers

Table 5 All results of a roof assembly design with modified AL-AD with parallelization

		Target value	Initial value	Optimal value	Discrepancy (%)
Roof assembly					
	Mass [kg]	0	491	437	-
	1st mode frequency [Hz]	5.825	5.0129	5.2858	-
	2nd mode frequency [Hz]	8.650	8.3749	8.8224	-
	Displacement for bending [mm]	< 70	66.048	67.565	-
	Displacement for torsion [mm]	< 450	431.5792	435.021	-
1st beam					
	Area [mm^2]	176.59	256	176.83	0.014

	MOI_1 [mm^4]	132769	235125	132697	0.0054
	MOI_2 [mm^4]	47364	114485	47355	0.0017
	Width [mm]	-	80	74.17	
	Height [mm]	-	50	37.95	
	Thickness [mm]	-	1	0.8	
2nd beam					
	Area [mm^2]	177.24	256	176.89	0.20
	MOI_1 [mm^4]	132769	235125	132832	0.047
	MOI_2 [mm^4]	47364	114485	47377	0.028
	Width [mm]	-	80	74.20	
	Height [mm]	-	50	37.95	
	Thickness [mm]	-	1	0.8	
3rd beam					
	Area [mm^2]	191.39	256	191.52	0.067
	MOI_1 [mm^4]	169345	235125	169430	0.050
	MOI_2 [mm^4]	59310	114485	59299	0.018
	Width [mm]	-	80	80.60	
	Height [mm]	-	50	40.70	
	Thickness [mm]	-	1	0.8	
4th beam					
	Area [mm^2]	177.57	256	176.93	0.36
	MOI_1 [mm^4]	132769	235125	132948	0.13
	MOI_2 [mm^4]	47364	114485	47388	0.051
	Width [mm]	-	80	74.23	
	Height [mm]	-	50	37.95	
	Thickness [mm]	-	1	0.8	

5. Conclusion

This paper proposes the modified AL-AD with parallelization based on non-hierarchical ATC framework. The parallelization is necessary, especially in large-scale design problems with many subsystems for efficiency. In conventional AL-AD which shows good performance in sequential solving, however, two difficulties in the parallelization of AL-AD with distributed coordination have discussed: the divergence and the oscillation of the solution. Divergence and oscillation are due to the crossing of linking variables and the increasing penalty weight in the method of multipliers. Thus, the proposed method uses the subgradient method with bilateral delayed inconsistencies of corresponding subproblems to find the optimum Lagrange multiplier based on duality theory and also proposes a guideline for proper penalty weight to ensure the convexity of the objective function and to prevent it from being unbounded and oscillation. Lastly, we propose an efficient algorithm to update the step size based on the path of the Lagrange multiplier using previous three values of Lagrange multiplier.

Numerical study shows that the proposed method has better performance in comparison with existing methods. The modified updating algorithm of the Lagrange multiplier and the proper penalty weight guarantees right direction to optimum and the adaptive step size update increases efficiency and prevents oscillation in parallel solving. The parallelization of the proposed method is much easier and more convenient than centralized coordination with an artificial master problem which should be reformulated from the original problem as bi-level coordination.

In future works, the proper selection of the weight factor used in the Lagrange multiplier update should be discussed analytically and numerically. It may be computed regarding two corresponding inconsistencies for better efficiency. Also, the load balance problem in parallel solving should be discussed. The load balance problem means that the function evaluation of one subsystem (e.g., Large-scale FEA) takes too long compared to the rest of the problem, thus reducing the effect of parallelization. Hence, a new coordination approach is necessary to redistribute loads of function evaluation in the optimization of subsystems considering the computational burden of function evaluation. Finally, we will also extend our research to the probabilistic domain for solving industrial problems partitioned in many subsystems which have input variables and parameters with uncertainty. Therefore, we will integrate our coordination scheme with parallelization of distributed coordination on industrial problems dealing with reliability analysis.

6. Acknowledgement

This work was partially supported by the Altair University Fellowship at Altair Engineering and the National Research Foundation of Korea(NRF) grant funded by the Korea government (N01170150). This support is gratefully acknowledged.

7. References

Altair (2012) Radioss Version 12.0

Bayrak AE, Kang N, Papalambros PY (2016). Decomposition-based design optimization of hybrid electric powertrain architectures: Simultaneous configuration and sizing design. Journal of Mechanical Design, 138(7), 071405.

Bertsekas DP, Tsitsiklis JN (1989). Parallel and distributed computation: numerical methods (Vol. 23). Englewood Cliffs, NJ: Prentice hall.

Bertsekas DP (2003). Nonlinear programming, 2nd edn, Athena scientific, Belmont

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

Boyd S, Vandenberghe L (2004). Convex optimization. Cambridge university press.

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.

Han J, Papalambros PY (2010). A sequential linear programming coordination algorithm for analytical target cascading. Journal of Mechanical Design, 132(2), 021003.

Kang N, Kokkolaras M, Papalambros PY (2014a). Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading. Structural and Multidisciplinary Optimization, 50(5), 849-859.

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

Kim HM (2001) Target cascading in optimal system design, Ph.D. Dissertation, Mechanical Engineering Dept., University of Michigan, Ann Arbor, MI

Kim HM, Michelena N, Papalambros PY, Jiang T (2003a) Target cascading in optimal system design. Journal of Mechanical Design 125(3):474–480

Kim HM, Rideout DG, Papalambros PY, Stein JL (2003b) Analytical target cascading in automotive vehicle design. Journal of Mechanical Design 125:481–489.

Kim HM, Chen W, Wiecek MM (2006). Lagrangian coordination for enhancing the convergence of analytical target cascading. AIAA journal, 44(10), 2197-2207.

Lassiter JB, Wiecek MM, Andrighetti KR (2005). Lagrangian coordination and analytical target cascading: solving ATC-decomposed problems with Lagrangian duality. Optimization and Engineering, 6(3), 361-381.

Li Y, Lu Z, Michalek JJ (2008). Diagonal quadratic approximation for parallelization of analytical target cascading. Journal of Mechanical Design, 130(5), 051402.

Michelena N, Park H, Papalambros PY (2003). Convergence properties of analytical target cascading. AIAA journal, 41(5), 897-905.

Michalek JJ, Papalambros PY (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.

Montes EM, Coello CAC (2005). A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9(1):1–17

MathWorks (2016) Matlab R2012a

Papalambros PY, Wilde DJ (2017). Principles of optimal design: modeling and computation. Cambridge university press.

Tosserams S, Etman LFP, Papalambros PY, Rooda JE (2006). An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. Structural and multidisciplinary optimization, 31(3), 176-189.

Tosserams S, Etman LFP, Rooda JE (2007). An augmented Lagrangian decomposition method for quasi-separable problems in MDO. Structural and Multidisciplinary Optimization, 34(3), 211-227.

Tosserams S, Etman LFP, Rooda JE (2008a). Augmented Lagrangian coordination for distributed optimal design in MDO. International journal for numerical methods in engineering, 73(13), 1885-1910.

Tosserams S, Etman LFP, Rooda JE (2008b). 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).

Tosserams S, Etman LFP, Rooda JE (2009a). A classification of methods for distributed system optimization based on formulation structure. Structural and Multidisciplinary Optimization, 39(5), 503-517.

Tosserams S, Etman LFP, Rooda JE (2009b). Block-separable linking constraints in augmented Lagrangian coordination. Structural and Multidisciplinary Optimization, 37(5), 521-527.

Tosserams S, Kokkolaras M, Etman LFP, Rooda JE (2010). A nonhierarchical formulation of analytical target cascading. Journal of Mechanical Design, 132(5), 051002.

Wang W, Blouin VY, Gardenghi MK, Fadel GM, Wiecek MM, Sloop BC (2013). Cutting plane methods for analytical target cascading with augmented Lagrangian coordination. Journal of Mechanical Design, 135(10), 104502.