

8-2015

Accuracy, Efficiency, and Parallelism in Network Target Coordination Optimization

Meng Xu
Clemson University

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Xu, Meng, "Accuracy, Efficiency, and Parallelism in Network Target Coordination Optimization" (2015). *All Dissertations*. 1864.
https://tigerprints.clemson.edu/all_dissertations/1864

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

ACCURACY, EFFICIENCY, AND PARALLELISM IN NETWORK TARGET
COORDINATION OPTIMIZATION

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mechanical Engineering

by
Meng Xu
August 2015

Accepted by:
Dr. Georges Fadel, Committee Chair
Dr. Margaret Wiecek
Dr. Joshua Summers
Dr. Gang Li

Distinguished External Reader: Dr. Panos Papalambros, University of Michigan

Abstract

The optimal design task of complex engineering systems requires knowledge in various domains. It is thus often split into smaller parts and assigned to different design teams with specialized backgrounds. Decomposition based optimization is a multidisciplinary design optimization (MDO) technique that models and improves this process by partitioning the whole design optimization task into many manageable sub-problems. These sub-problems can be treated separately and a coordination strategy is employed to coordinate their couplings and drive their individual solutions to a consistent overall optimum. Many methods have been proposed in the literature, applying mathematical theories in nonlinear programming to decomposition based optimization, and testing them on engineering problems. These methods include Analytical Target Cascading (ATC) using quadratic methods and Augmented Lagrangian Coordination (ALC) using augmented Lagrangian relaxation. The decomposition structure has also been expanded from the special hierarchical structure to the general network structure. However, accuracy, efficiency, and parallelism still remain the focus of decomposition based optimization research when dealing with complex problems and more work is needed to both improve the existing methods and develop new methods.

In this research, a hybrid network partition in which additional sub-problems can either be disciplines or components added to a component or discipline network respectively is proposed and two hybrid test problems are formulated. The newly developed consensus optimization method is applied on these test problems and shows

good performance. For the ALC method, when the problem partition is given, various alternative structures are analyzed and compared through numerical tests. A new theory of dual residual based on Karush-Kuhn-Tucker (KKT) conditions is developed, which leads to a new flexible weight update strategy for both centralized and distributed ALC. Numerical tests show that the optimization accuracy is greatly improved by considering the dual residual in the iteration process. Furthermore, the ALC using the new update is able to converge to a good solution starting with various initial weights while the traditional update fails to guide the optimization to a reasonable solution when the initial weight is outside of a narrow range. Finally, a new coordination method is developed in this research by utilizing both the ordinary Lagrangian duality theorem and the alternating direction method of multipliers (ADMM). Different from the methods in the literature which employ duality theorems just once, the proposed method uses duality theorems twice and the resulting algorithm can optimize all sub-problems in parallel while requiring the least copies of linking variables. Numerical tests show that the new method consistently reaches more accurate solutions and consumes less computational resources when compared to another popular parallel method, the centralized ALC.

Dedication

To my family who have been so supportive and love me unconditionally.

Acknowledgement

I would like to thank my advisor Dr. Georges Fadel for his constant and patient guidance throughout my Ph.D. study. Getting into a new research area and developing new ideas is a challenging journey, yet Dr. Fadel's extensive knowledge and rich experience led me through it. I want to thank Dr. Margaret Wiecek for her insightful remarks and constructive criticism to my research. I am deeply grateful to her for carefully reading and commenting on countless revisions of my papers, from which I learned both the rigorousness and the elegance of math. I also want to thank Dr. Joshua Summers and Dr. Gang Li for the time and effort they volunteered to guide my work.

I would like to thank Dr. Paolo Guarneri and Dr. Wenshan Wang. The fruitful discussions we had in my first two years at Clemson greatly inspired me and helped me a lot in the formation of the ideas presented in this dissertation. Thanks to my fellow labmates in Office 119 for their companionship in the past years. They build up such a warm and relaxing working environment that continuously encourages and supports me. Thanks to our big CEDAR group from which I benefited a lot. We discuss our research on the weekly meetings and participate in all kinds of activities together. It is such a wonderful experience to grow personally and academically with the whole group.

Finally, I want to thank my family for their love. They are my constant source of encouragement, concern, and strength. None of this would have been possible without their support, and I am so fortunate to have such a wonderful family.

Table of Contents

	Page
Title Page	i
Abstract	ii
Dedication	iv
Acknowledgement	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Decomposition based Optimization	1
1.2 Mathematical Foundations	5
1.3 Research Scope and Research Questions	10
1.4 Dissertation Outline.....	13
2 Literature Review of Decomposition Based Optimization	16
2.1 Collaborative Optimization	16
2.2 Analytical Target Cascading	18
2.3 Ordinary Lagrangian Method	22
2.4 Augmented Lagrangian Coordination	22
2.5 Consensus Optimization via Alternating Direction Method of Multipliers	27
2.6 Weight Selection and Update in Decomposition based Optimization.....	30
3 Hybrid Partition with Both Component and Discipline Sub-problems	32
3.1 Hybrid partition	32
3.2 Micro-accelerometer Benchmark Problem and its Hybrid Decomposition.....	34
3.3 Application of CADMM to Hybrid Micro-accelerometer Problems and Results	37
3.4 Summary	39
4 Alternative Structures of ALC Under the Same Partition	40
4.1 Alternative structures of ALC	40

Table of Contents (Continued)

	Page
4.2 Comparison of the Distributed ALC, the Centralized ALC, and the ATC Extended by ALC through Numerical Tests	43
4.3 Summary	52
5 Dual Residual for Centralized ALC Based on Optimality Conditions	53
5.1 Optimality Conditions for Decomposed Problems in Centralized ALC	55
5.2 Weight Update Strategy Based on the Primal and Dual Residual	63
5.3 Numerical Tests	64
5.4 Summary	88
6 Dual Residual for Distributed ALC Based on Optimality Conditions	91
6.1 Optimality Conditions for Distributed ALC and a New Weight Update based on Dual Residuals	92
6.2 Numerical Tests	101
6.3 Summary	114
7 An Efficient Parallel Coordination Method Using Two Duality Theorems	116
7.1 Alternating Direction Method of Multipliers (ADMM)	116
7.2 Derivation	118
7.3 Numerical Tests and Results Analysis	128
7.4 Summary	136
8 Concluding Remarks	138
8.1 Contributions	138
8.2 Future Research	141
Bibliography	143

List of Tables

Table	Page
3.1 Optimization results of CADMM applied to the two kinds of hybrid decompositions (reference solution = 0.0453).....	38
4.1 Results for solving geometric problem using different solving structures	46
4.2 Optimization results of the micro-accelerometer problem (case 3) without decomposition (AIO) [60].....	49
4.3 Results for different solving structures of the micro - accelerometer problem	50
5.1 Optimization results of the micro-accelerometer problem (case 3) without decomposition (AIO) [60].....	79
5.2 Results of the micro-accelerometer problem with a calculated initial weight by Eq. (5.34)	80
5.3 Results of the micro-accelerometer problem with a fine-tuned initial weight.....	84
5.4 Optimization results of the micro-accelerometer problem with different initial weights	88
6.1 Results of different coordination sequences for Example 1	105
6.2 Results of the traditional and proposed update using different coordination sequences for Example 2	106
6.3 Results of the traditional and the proposed updates on Example 3	111
6.4 Optimization results of Example 3 problem with different initial weights	113

List of Figures

Figure	Page
1.1 The process of decomposition based optimization	2
1.2 The simple component based partition (a) and discipline based partition (b) of a car optimization problem	3
2.1 Example of a hierarchical structure (left) and variable allocation (right) [20]	18
2.2 Nested ATC schemes with inner loop (left) and ADMM without inner loop (right) [55].....	24
2.3 Original problem (left) and decoupled problem (right) for CADMM.....	28
3.1 top left - discipline-based decomposition; top right – component based decomposition; bottom – hybrid decompositions (rectangle stands for discipline, circle stands for component)	33
3.2 Microscope image and schematic illustration (right) of the micro-accelerometer [60].....	34
3.3 Hybrid decomposition 1 for the micro-accelerometer problem.....	36
3.4 Hybrid decomposition 1 for the micro-accelerometer problem.....	37
4.1 A four-node decomposed problem with a network structure.....	41
4.2 Four different structures can be used to solve the example problem through ALC.....	42
4.3 Partition used for the geometric optimization problem with five sub-problems	44
4.4 Four possible solving structures for the geometric problem.....	45
4.5 Object based partition of the micro-accelerometer benchmark problem [60]	48
4.6 Three solving structures for the micro-accelerometer problem for object based partition	49
4.7 The effect of number of copies of linking variables in ALC and the tradeoff it causes between accuracy, efficiency, and parallelism.....	50
5.1 Procedure of centralized ALC (problems in grey areas are solved as one problem, dashed lines represent the sub-problem couplings when the number of sub-problem is more than 4)	56
5.2 Comparison of KKT conditions between the AIO and centralized ALC (the equations with shadow background on the left are guaranteed by the equations with shadow background on the right or convergence criteria).....	61

List of Figures (Continued)

Figure	Page
5.3 The structure to solve the geometric programming problem through decomposition.....	67
5.4 Optimization process of the traditional update on the geometric programming problem – initial weight set by Eq. (5.34)	68
5.5 Optimization process of the proposed update on the geometric programming problem – initial weight set by Eq. (5.34)	69
5.6 Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the geometric programming problem – initial weight set by Eq. (5.34)	69
5.7 Optimization process of the traditional update on the geometric programming problem – using fine-tuned initial weight.....	71
5.8 Optimization process of the proposed update on the geometric programming problem – using fine-tuned initial weight.....	71
5.9 Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the geometric programming problem – using fine-tuned initial weight	72
5.10 The structure to solve the Golinski's problem through decomposition	73
5.11 Optimization process of the traditional weight update on the speed reducer problem – initial weight set by Eq. (5.34)	74
5.12 Optimization process of the proposed weight update on the speed reducer problem – initial weight set by Eq. (5.34)	75
5.13 Comparison of the curves of the biggest dual residual for the traditional update and proposed update on the speed reducer problem – initial weight set by Eq.(5.34)	75
5.14 Optimization process of the traditional weight update on the speed reducer problem – using fine-tuned initial weight.....	76
5.15 Optimization process of the proposed weight update on the speed reducer problem – using fine-tuned initial weight.....	77
5.16 Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the speed reducer problem – using fine-tuned initial weight.....	77
5.17 The Structure to solve the micro-accelerometer problem through decomposition..	79

List of Figures (Continued)

Figure	Page
5.18 Optimization process of the centralized ALC – scheme 1 on the micro-accelerometer problem – initial weight set by Eq. (5.34).....	81
5.19 Optimization process of the centralized ALC – scheme 2 on the micro-accelerometer problem – initial weight set by Eq. (5.34).....	81
5.20 Optimization process of the centralized ALC – scheme 3 on the micro-accelerometer problem – initial weight set by Eq. (5.34).....	82
5.21 Optimization process of the centralized ALC – scheme 1 on the micro-accelerometer problem – fine-tuned initial weight	85
5.22 Optimization process of the centralized ALC – scheme 2 on the micro-accelerometer problem – fine-tuned initial weight	86
5.23 Optimization process of the centralized ALC – scheme 3 on the micro-accelerometer problem – fine-tuned initial weight	86
6.1 Comparison of KKT conditions between the AIO and distributed ALC (the equations with shadow background on the left are guaranteed by the equations with shadow background on the right or convergence criteria).....	97
6.2 The partition of Example 1 with four sub-problems.....	102
6.3 The optimization process with the traditional update on Example 1 (solving sequence: 1-2-4-3)	103
6.4 The optimization process with the proposed update on Example 1 (solving sequence: 1-2-4-3)	104
6.5 Number of function evaluations resulting from different μ and τ of the proposed update for Example 1	105
6.6 The optimization process with the traditional update on Example 2 (solving sequence: 1-2-3-4-5)	107
6.7 The optimization process with the proposed update on Example 2 (solving sequence: 1-2-3-4-5)	108
6.8 Number of function evaluations resulting from different μ and τ of the proposed update for Example 2	109
6.9 Object based partition of Example 3 [60].....	110
6.10 The optimization process with the traditional on Example 3	111
6.11 The optimization process with the proposed updates on Example 3	112

List of Figures (Continued)

Figure	Page
7.1 A partition with a three-node network structure for the illustration of selection matrix	120
7.2 The flowchart of the proposed parallel coordination method	125
7.3 Dual transformations in the proposed method and other methods in the literature	126
7.4 The decompositions of the centralized ALC (left) and the proposed method (right) when solving the geometric optimization problem	129
7.5 The curves of biggest primal and dual consistency errors under different ρ	130
7.6 The test results of the centralized ALC and the proposed method on the geometric optimization problem (From right to left: $\varepsilon = 1e-2, 1e-3, 1e-4$ and $1e-5$)	131
7.7 Structure and cross sectional area of the portal frame design problem[59].....	133
7.8 The decompositions of the proposed method and the centralized ALC when solving the portal frame design problem	133
7.9 The objective errors of ALC-ADMM and the proposed method for 20 tests of the portal frame problem.....	136

Chapter 1

Introduction

1.1 Decomposition based Optimization

The optimal design task in engineering is facing rising challenges as products become more and more complex. Because the products consist of many components that are connected or coupled with each other, their design usually involves several interacting disciplines which may share some of the same design variables. The optimal design of a complex system cannot be solved by a single person or even a single design team. It requires experts in different specialties working both collaboratively and independently. By “independently”, it is meant that the original design task is split into many parts and each expert only needs to deal with the part related to his or her expertise, without considering the parts in other domains. By “collaboratively”, it is meant that experts focusing on different parts need to exchange certain design information that are shared among them and guarantee that the aggregation of their designs are consistent and optimal for the whole system. The theoretical foundation for decomposing the optimal design task has been studied since the early 1960s [1][2]. As shown in Figure 1.1, decomposition based optimization consists of two processes [3]:

(1) Partitioning of a system into smaller sub-systems that can be designed autonomously.

(2) Coordination of the individual sub-systems towards an optimal and consistent system.

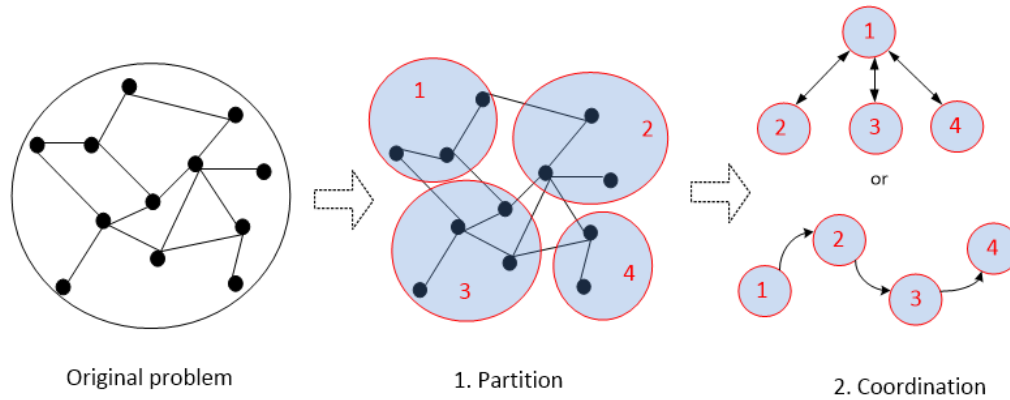
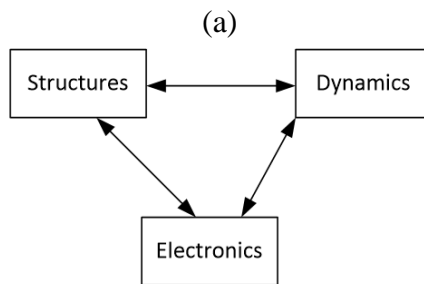
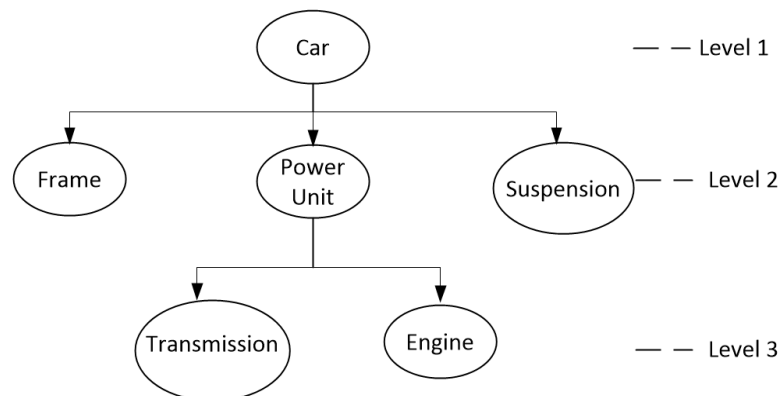


Figure 1.1: The process of decomposition based optimization

A large engineering optimization problem can either be partitioned according to the components it is composed of, or according to the disciplines involved [4]. These decompositions can be solved using Multilevel Optimization (MLO) and Multidisciplinary Design Optimization (MDO) [5][6][7]. Other partitions do exist, for instance based on functions or on flow of information, process, or organization divisions [8][9][10]. Figure 1.2(a) shows an example of a component based partition for the car optimal design which consists of frame, power unit, and suspension design. The power unit sub-problem can be further partitioned into transmission and engine design. These sub-problems form a hierarchical structure in which each sub-problem belongs to a specific level and can only communicate with its parent on the upper level or children on the lower level. MLO is designed to solve this kind of multilevel partitions, while in MDO there is no clear level in the problem partition. In Figure 1.2 (b), the discipline

based partition of the car design has three fully coupled sub-problems: structures, dynamics, and electronics. All the sub-problems are equally important and each of them can freely communicate with the others. This structure is called a network structure. The car design problem shown here is idealized and the only purpose of it is to illustrate the definitions of component and discipline based partition. The real design problem of a car is much more complex.

Unlike single-level methods which have a single, centralized decision-making process, MLO and MDO distribute the decision-making tasks among all sub-problems [11], and both approaches have been applied to various engineering problems such as vehicle design and aircraft design [12][13].



(a)
(b)
Figure 1.2: The simple component based partition (a) and discipline based partition (b) of a car optimization problem

Many studies have also been conducted to find optimal partitions for decomposition-based optimization [14][15][16][17]. In particular, the model-based method of Michelena and Papalambros [15] uses hypergraphs to study the optimal problem partition in decomposition-based optimization. Those methods belong to the partition phase of decomposition based optimization. Although choosing the appropriate partition is important, in decomposition based optimization we often assume that the partition of an optimization problem has previously been determined according to certain rules (component, disciplines, model-based or hybrid) [4][9][11][16], and the main focus of the studies in the literature has been on the coordination process. For example, the very recent methods such as Analytical Target Cascading (ATC) [18][22][23], Augmented Lagrangian Coordination (ALC) [20][24], and Consensus Optimization (CADMM) [25][26][27] all deal with the couplings between partitioned sub-problems which are generated beforehand.

The ATC method is specifically designed to solve the decomposed problems exhibiting a hierarchical structure, which normally results from the component-based partition. In this hierarchical problem there are different levels and the sub-problems on the same level do not interact with each other but are allowed to communicate with the sub-problems on the higher or the lower level. The ATC method fits well the hierarchically decomposed problem and often is chosen as the coordination strategy for this kind of decomposition.

When it comes to decomposed problems of a network structure, a more flexible coordination strategy such as the augmented Lagrangian coordination (ALC) is needed.

Tosserams [20] categorizes the ALC method for MDO into centralized ALC and distributed ALC based on the existence of a master problem. The centralized ALC has a sub-problem acting as a master problem that coordinates all other sub-problems, which results in a bi-level decomposed structure. Once the master problem on the top level has been solved, all the sub-problems on the bottom level can be solved in parallel. Due to this advantage, the centralized ALC is researched extensively and the traditional hierarchical ATC method can be extended to a non-hierarchical version by using the centralized ALC [24]. Also, the master problem in the centralized ALC is typically artificial and its analytical solution can be calculated easily. There is no master problem in the distributed ALC, thus its decomposed sub-problems may depend on each other which prevents parallel computation of all sub-problems.

The network structure (as shown in Figure 1.2 (b)) is the most general structure in which there is no rule for how one sub-problem should connect with other sub-problems, whereas the hierarchical structure in Figure 1.2 (a) can be considered a special case of the network structure. This dissertation studies the general case where the problems are partitioned into network structures, for which the coordination process of sub-problems is called Network Target Coordination (NTC) [25][27]. As a result, the proposed research is applicable to hierarchical structures by default.

1.2 Mathematical Foundations

Mathematical programming is the foundation of most methods in decomposition based optimization, many theories derived from pure mathematical deductions turned out to be also very effective when solving engineering problems. Thus, in order to provide a

comprehensive view of current decomposition based optimization techniques, the related mathematical foundation is described first.

1.2.1 Primal problem

The general optimization problem can be expressed as follows:

$$\begin{aligned}
 & \text{minimize} && f(\mathbf{x}) \\
 & \text{subject to} && g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \dots, m \\
 & && h_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \dots, l \\
 & && \mathbf{x} \in X
 \end{aligned} \tag{1.1}$$

where \mathbf{x} is the vector of design variables. $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are inequality and equality constraints respectively. $f(\mathbf{x})$ is the objective we are trying to optimize. This problem is called the primal problem in contrast to the dual problem introduced next.

1.2.2 Lagrangian Dual Problem

Among various duality formulations, the Lagrangian duality formulation is one of the most studied. It has been proven to be powerful for solving convex, nonconvex and discrete optimization problems [28][29], and it is used throughout this dissertation.

$$\begin{aligned}
 & \text{maximize} && \theta(\mathbf{u}, \mathbf{v}) \\
 & \text{subject to} && \mathbf{u} \geq \mathbf{0},
 \end{aligned} \tag{1.2}$$

$$\text{where } \theta(\mathbf{u}, \mathbf{v}) = \inf \left\{ f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{i=1}^l v_i h_i(\mathbf{x}) : \mathbf{x} \in X \right\}$$

The Lagrangian dual problem is to maximize the minimum of the Lagrangian function $L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{i=1}^l v_i h_i(\mathbf{x})$. In this problem, the original constraints $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ in Eq. (1.1) have been incorporated in the objective function $\theta(\mathbf{u}, \mathbf{v})$ using the Lagrangian multipliers u_i and v_i respectively.

According to the weak duality theorem [28][29], the objective value of any feasible solution to the dual problem in Eq. (1.2) yields a lower bound on the objective

value of any feasible solution to the primal problem in Eq. (1.1). The difference between these two objective values is called the duality gap, which can be eliminated under certain assumptions meeting the strong duality theorem [28][29].

1.2.3 Karush-Kuhn-Tucker (KKT) optimality conditions

KKT optimality conditions are a series of equations built upon Lagrangian multipliers to characterize the optimal solutions of an optimization problem [28][29]. For the problem (1.1), suppose \mathbf{x}^* is a local optimal solution, $g_i(\mathbf{x}^*)$ and $h_i(\mathbf{x}^*)$ are differentiable at \mathbf{x}^* , $\nabla g_i(\mathbf{x}^*)$ for $i \in \{i : g_i(\mathbf{x}^*) = 0\}$ and $\nabla h_i(\mathbf{x}^*)$ are linearly independently, then there exist scalars u_i for $i = 1, \dots, m$ and v_i for $i = 1, \dots, l$ such that

$$\begin{aligned} f(\mathbf{x}^*) + \sum_{i=1}^m u_i g_i(\mathbf{x}^*) + \sum_{i=1}^l v_i h_i(\mathbf{x}^*) &= \mathbf{0} \\ u_i g_i(\mathbf{x}^*) &= 0 && \text{for } i = 1, \dots, m \\ u_i &\geq 0 && \text{for } i = 1, \dots, l \end{aligned} \quad (1.3)$$

Eq. (1.3) is called the KKT necessary conditions. Furthermore, if $f_i(\mathbf{x})$, $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ satisfy certain convex assumptions, Eq. (1.3) can become the KKT sufficient conditions and can be used to check if a feasible solution is the optimal solution.

1.2.4 Block coordinate descent

The optimization can be solved part by part iteratively. Assume an ideal optimization problem where there is no constraint:

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in X \end{aligned} \quad (1.4)$$

The design variable vector \mathbf{x} can be partitioned as

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$$

Under the assumption that f is continuously differentiable over the set X and under certain convergence requirements, the original problem can be solved through the following iterative method, which is called block coordinate descent or nonlinear Gauss – Seidel method [29][30][31][32].

Given the current iterate $\mathbf{x}^k = (\mathbf{x}_1^k, \dots, \mathbf{x}_m^k)$, the next iterate is calculated through

$$\mathbf{x}_i^{k+1} = \arg \min_{\mathbf{x}_i \in X_i} f(\mathbf{x}_1^{k+1}, \dots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^k, \dots, \mathbf{x}_m^k) \quad (1.5)$$

As the iteration process continues, every limit point of $\{\mathbf{x}^k\}$ is a stationary point, which in many cases is an optimal solution to the original problem.

1.2.5 Penalty methods

Penalty methods enable us to eliminate some constraints by adding a penalty term that prescribes an additional high cost to the objective function of infeasible points.

Consider the equality constrained problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && h(\mathbf{x}) = 0 \\ & && \mathbf{x} \in X \end{aligned} \quad (1.6)$$

It can be transformed to the following relaxed problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) + \mu h^2(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in X, \end{aligned} \quad (1.7)$$

where $\mu > 0$ is called a penalty parameter.

When μ is large enough, due to the high cost of infeasibility, the relaxed problem generates a good approximation of the optimal solution of the original problem. This relaxation method is called the quadratic penalty method [28].

A more general form of penalty methods is stated as follows. For the problem

$$\begin{aligned}
& \text{minimize} && f(\mathbf{x}) \\
& \text{subject to} && g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \dots, m \\
& && h_i(\mathbf{x}) = 0 \quad \text{for } i = 1, \dots, l \\
& && \mathbf{x} \in X
\end{aligned} \tag{1.8}$$

The equality and inequality constraints can be relaxed as follows [28]

$$\begin{aligned}
& \text{minimize} && f(\mathbf{x}) + \sum_{i=1}^m \phi[g_i(\mathbf{x})] + \sum_{i=1}^l \psi[h_i(\mathbf{x})] \\
& \text{subject to} && \mathbf{x} \in X
\end{aligned} \tag{1.9}$$

where ϕ and ψ are continuous functions satisfying:

$$\begin{aligned}
\phi(y) &= 0 \quad \text{if } y \leq 0 \quad \text{and} \quad \phi(y) > 0 \quad \text{if } y > 0 \\
\psi(y) &= 0 \quad \text{if } y = 0 \quad \text{and} \quad \psi(y) > 0 \quad \text{if } y \neq 0
\end{aligned}$$

1.2.6 Augmented Lagrangian relaxation method

The Augmented Lagrangian relaxation method is a combination of the Lagrangian dual method and the quadratic penalty method [29].

The constrained equality problem (1.6) can be relaxed as follows:

$$\begin{aligned}
& \text{minimize} && L_c(\mathbf{x}, v) = f(\mathbf{x}) + v^T h(\mathbf{x}) + \frac{c}{2} \|h(\mathbf{x})\|^2 \\
& \text{subject to} && \mathbf{x} \in X,
\end{aligned} \tag{1.10}$$

where v is the Lagrangian multiplier and c is a positive penalty parameter.

The augmented Lagrangian relaxation method can reach optimal solutions without a large penalty parameter c , thus it avoids the ill-conditioning difficulties encountered by the classical penalty methods as the penalty parameter approaches infinity.

1.2.7 Method of multipliers

For the augmented Lagrangian relaxation formulation to yield an accurate approximation to the optimal solution of the original problem, the multiplier v should

tend to the optimal value v^* of the dual problem. Normally this process needs iterations between solving the relaxed problem and updating v toward v^* . The method of multipliers is an update strategy for the multiplier v . Given v^k at iteration k [29],

$$v^{k+1} = v^k + c^k h(x^k) \quad (1.11)$$

where c^k is the quadratic penalty weight parameter at iteration k .

1.3 Research Scope and Research Questions

The literature on decomposition based optimization has focused on the mathematical representation of the “split and assign” process happening in industry. The mathematical models being studied are built upon simplification of the modern engineering optimal design process and do not necessary contain all the factors considered in industry. For instance, the decision making process of an engineering design team is generalized to a mathematical optimization problem. The interactions between different design teams or the couplings between various components and disciplines are interpreted as the exchange of certain variables between these entities. As a research developed upon the literature in decomposition based optimization, this dissertation follows the same simplification rules and studies the generic mathematical model of engineering optimization problems.

For problems solved through NTC, coordination of sub-problems always plays an important role because it eliminates the inconsistencies between the decomposed sub-problems and drives their solutions towards the optimal solution of the original problem. Coordination is often carried out in an iterative manner, and as the complexity of

engineering problems continues to grow rapidly, accuracy, efficiency, and parallelism become critical issues during this process.

Aiming at improving the accuracy, efficiency, and parallelism of Network Target Coordination (NTC), this dissertation is designed to explore the answers of the following research questions:

1. Can NTC methods deal with problems with hybrid partitions?

A hybrid partition is obtained when a new discipline (component) is added to a problem decomposed by components (disciplines) to reflect the inherent dynamics within the design process. The resulting problem becomes a non-hierarchical, network optimization problem that requires a suitable coordination approach. The efficacy of the newly proposed NTC method - Consensus optimization via Alternating Direction Method of Multipliers (CADMM) on this kind of hybrid partition needs to be explored.

2. What are the alternative structures for solving problems using Augmented Lagrangian Coordination (ALC) and how are they compared to each other?

ALC is a very flexible coordination method and when we use it to solve NTC problems, there are many alternative structures which have different characteristics such as the number of multipliers and levels. These structures may have great effects on the optimization results. The performance of these alternative structures needs to be analyzed and compared to provide guidelines for choosing the appropriate structure in the implementation of ALC.

3. Does the optimal solution of decomposition based optimization satisfy the KKT necessary conditions of the original problem? If not, what kind of mechanism can we employ to guarantee these conditions are met for NTC methods?

The KKT optimality necessary conditions represent a series of requirements that must be fulfilled by a feasible solution if this solution is optimal. For the optimal solution of decomposition based optimization to be optimal for the original problem, it must satisfy the KKT conditions of the problem before decomposition. If not, some method needs to be introduced to drive the optimization results to match those KKT conditions. The update strategy of penalty weights is a potential candidate for this purpose.

For the NTC methods using augmented Lagrangian relaxation, the initial setting and update strategy of the penalty weights are critical to the optimization performance. The traditional weight update strategy always increases the weights during the iteration process, and inappropriate initial weights tend to cause the iterative optimization process to converge prematurely and output irrelevant solutions.

4. Instead of just using one duality theorem, can we employ two different duality theorems to develop a new NTC coordination method capable of solving all sub-problems in parallel?

Most popular coordination methods in the literature use duality theorems just once, transforming the primal problem (the original problem) into a dual

problem. What will happen if we apply the duality theorems twice? The resulting formulation will be a dual problem of a dual problem. By choosing the right duality theorems in this process, it is possible to introduce fewer copies of shared variables to the decomposition, which decreases the coordination effort necessary for the optimization to converge, thus the new NTC method is expected to perform better than other parallel methods in terms of efficiency and accuracy.

1.4 Dissertation Outline

The dissertation is organized according to the following outline:

Chapter 2 reviews the popular decomposition based optimization methods in the literature, which includes methods with nested formulation (such as collaborative optimization), methods with hierarchical structure (such as analytical target cascading), and methods with nonhierarchical structure (such as augmented Lagrangian coordination, and consensus optimization). The weight selection and update strategy are also introduced as critical procedures in decomposition based optimization.

Chapter 3 proposes a new type of partition which may emerge from engineering application of decomposition – the hybrid partition. The sub-problems in this new partition can be either a component design problem or a discipline design problem, and they are connected through a network structure. For numerical tests, the complex benchmark problem – micro-accelerometer design problem is reconstructed to form two hybrid partitions, to which the recently developed consensus optimization method is applied and its solution is compared to the AIO solution.

Chapter 4 numerically explores the alternative structures of ALC when the partition of a problem is given. These alternative structures' performance varies greatly in terms of accuracy and efficiency according to their number of multipliers. The distributed ALC consumes the least iterations and functions evaluation, whereas the centralized ALC enables parallel computation of sub-problems.

Chapter 5 applies the KKT necessary conditions to the problem formulations of centralized ALC before and after decomposition. The results show that one dual feasibility condition of the original problem is not guaranteed by the optimal solution of the centralized ALC. The new terms "primal residual" and "dual residual" are introduced to centralized ALC, and a new flexible weight update is proposed based on the residuals. One mathematical and two engineering examples are used to test the new update.

Chapter 6 extends the dual residual theory and the new flexible weight update for centralized ALC in Chapter 5 to the distributed ALC. Numerical tests show a significant increase in solution accuracy and robustness of distributed ALC when employing the new update.

Chapter 7 develops a new coordination method using two duality theorems: the ordinary Lagrangian duality theorem and the alternating direction method of multipliers. The sub-problems in the new method can be solved in parallel and require the least copies of linking variables to be decoupled. The accuracy, efficiency, and robustness of this new method are tested on a math programming problem and a structural optimization problem.

Chapter 8 concludes the dissertation and provides possible directions for future research.

Chapter 2

Literature Review of Decomposition Based Optimization

This chapter reviews the prevalent decomposition based optimization methods in chronological order, which covers collaborative optimization (bi-level nested), analytical target cascading (hierarchical), ordinary Lagrangian method (hierarchical), augmented Lagrangian coordination (network), and consensus optimization (network). The coordination methods using approximations and weight update in coordination are also discussed.

2.1 Collaborative Optimization

Several classical decomposition based methods have been proposed in the 1990's, such as collaborative optimization (CO) [33][34], current subspace optimization (CSSO) [35] and bi-level integrated system simulations (BLISS) [36]. They all belong to the multi-level optimization methods which allow decision making at sub-system levels and provide many insights for follow up studies [37]. Among them, CSSO and BLISS involve the approximations of contributions of sub-systems in their coordination process. Consequently, their numerical performance typically depends on the quality of approximations. Since approximation is not the focus of this research, readers can refer to the above references for the details of CSSO and BLISS.

The Collaborative Optimization method is a bi-level (system and sub-system levels) method. The system optimization is performed with respect to the system targets. Each sub-system receives the system targets and tries to minimize the discrepancy between the system targets and the corresponding responses generated by the sub-problem design variables, subject to the local design constraints.

Assume the design variables \mathbf{x}_j is the vector of local variables associated with sub-system j . The total number of sub-problems is M . \mathbf{y} is the vector of linking variables, which can either be the shared variables that appear in more than two sub-problems, or some intermediate variables required as inputs to some sub-problems but are not design variables of the original problem.

To allow the sub-problems at sub-system level to make decisions on their own, copies of linking variables \mathbf{y}_j are introduced to the sub-problem j . The system level problem can be formulated as:

$$\begin{aligned}
 & \min_{\mathbf{y}} f(\mathbf{y}) \\
 & \text{subject to } \mathbf{g}_0(\mathbf{y}) \leq \mathbf{0} \\
 & \quad \mathbf{h}_0(\mathbf{y}) = \mathbf{0} \\
 & \quad \|\mathbf{y} - \mathbf{y}_j^*\|_2^2 = \mathbf{0}, \quad j = 1, \dots, M
 \end{aligned} \tag{2.1}$$

where \mathbf{y}_j^* come from the optimal solution for the sub-problem j at sub-system level.

The objective of sub-problem j is to minimize the discrepancy between \mathbf{y} from the system level and the copies \mathbf{y}_j .

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) = \|\mathbf{y} - \mathbf{y}_j\|_2^2 \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0}, \quad j = 1, \dots, M
\end{aligned} \tag{2.2}$$

One evaluation of the system level problem needs the optimal solutions of all sub-problems. Tosserams [38] classified CO as a nested formulation which means the sub-problem optimization is nested in the system level optimization. The gradients of the system level constraints are undefined at optimal system solutions, which causes ill-posedness to this method [39][40][41].

2.2 Analytical Target Cascading

Analytical Target Cascading (ATC) [18][23][42] is developed to solve complex systems with a hierarchical structure. Figure 2.1 illustrates a hierarchical structure. For two linked elements (or sub-problems), the one on an upper level is called a parent and the one on a lower level is called a child. Parents and children may share same design variables or intermediate parameters which are not part of the design variables but are dependent on design variables.

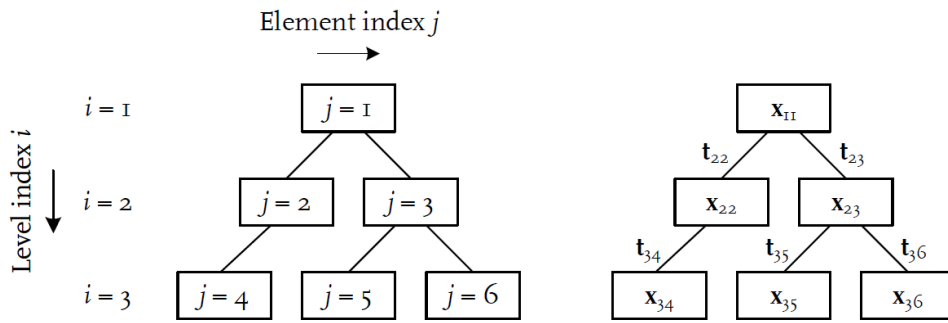


Figure 2.1: Example of a hierarchical structure (left) and variable allocation (right) [20]

Assume we have a hierarchical structured problem with N levels and M elements ($N = 3$ and $M = 6$ in Figure 2.1). For convenience, each element in this structure is assigned an identification number ij , where i is the level this element belongs to and j is the number of the element. Then the All-In-One (AIO) formulation for this problem can be defined as [20]

$$\begin{aligned}
& \min_{\mathbf{x}_{11}, \dots, \mathbf{x}_{NM}, \mathbf{t}_{22}, \dots, \mathbf{t}_{NM}} \sum_{i=1}^N \sum_{j \in \mathcal{E}_i} \mathbf{f}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \\
& \text{s.t. } \mathbf{g}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \leq \mathbf{0} \\
& \quad \mathbf{h}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \leq \mathbf{0} \\
& \quad i = 1, \dots, N, j \in \mathcal{E}_i
\end{aligned} \tag{2.3}$$

where \mathbf{x}_{ij} is the vector of local design variables, \mathbf{t}_{ij} is the vectors of shared variables by element j at levels i with its parent at level $i-1$, which are called targets. The design variables are all the \mathbf{x}_{ij} and \mathbf{t}_{ij} ; \mathcal{E}_i is the set of indices of all elements on level i ; $C_{ij} = \{k_1, \dots, k_{c_{ij}}\}$ is the set of indices of all children of element j and c_{ij} is the size of this set. \mathbf{g}_{ij} and \mathbf{h}_{ij} are inequality and equality constraints for each element.

Copies of target \mathbf{t}_{ij} are then introduced to separate this problem. These copies are called responses \mathbf{r}_{ij} and are assigned to each child, while the target \mathbf{t}_{ij} becomes a design variable for each parent. The value of the responses should match the targets which forms the consistency constraint \mathbf{c}_{ij} :

$$\mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \tag{2.4}$$

The modified AIO problem becomes

$$\begin{aligned}
& \min_{\bar{\mathbf{x}}_{11}, \dots, \bar{\mathbf{x}}_{NM}} \sum_{i=1}^N \sum_{j \in \mathcal{E}_i} \mathbf{f}_{ij}(\bar{\mathbf{x}}_{ij}) \\
& s.t. \quad \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \quad \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \quad \mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \\
& \text{where } \bar{\mathbf{x}}_{ij} = \{\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}\} \\
& \quad i = 1, \dots, N, j \in \mathcal{E}_i
\end{aligned} \tag{2.5}$$

The consistency constraint \mathbf{c}_{ij} is so strict that it prohibits us from solving this problem using an iterative strategy. Therefore, ATC adopts the quadratic penalty function to relax \mathbf{c}_{ij} to the objective.

$$\phi(\mathbf{c}) = \|\mathbf{w} \circ \mathbf{c}\|_2^2 = \sum_{i=2}^N \sum_{j \in \mathcal{E}_i} \|\mathbf{w}_{ij} \circ \mathbf{c}_{ij}\|_2^2 \tag{2.6}$$

where $\|\cdot\|_2^2$ denotes the square of l_2 norm; \circ denotes the Hadamard production [20]:

$$[a_1, \dots, a_n] \circ [b_1, \dots, b_n] = [a_1 b_1, \dots, a_n b_n]$$

At this point the AIO problem can be decomposed into a collection of sub-problems with the following general formulation. For the sub-problem j on level i , its design variables are $\bar{\mathbf{x}}_{ij}$ and the terms that do not include $\bar{\mathbf{x}}_{ij}$ can be considered as constants and dropped.

$$\begin{aligned}
& \min_{\bar{\mathbf{x}}_{ij}} \mathbf{f}_{ij}(\bar{\mathbf{x}}_{ij}) + \|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2 + \sum_{k \in \mathcal{C}_{ij}} \|\mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})\|_2^2 \\
& s.t. \quad \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \quad \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \text{where } \bar{\mathbf{x}}_{ij} = \{\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}\}
\end{aligned} \tag{2.7}$$

For the sub-problem j on level i , the target vector \mathbf{t}_{ij} comes from its parent and the response vector $\mathbf{r}_{(i+1)k}$ comes from its children. They are kept constant for this sub-problem. For the top level sub-problem, there is no $\|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2$ term since the top level sub-problem does not have a parent. For a similar reason, there is no

$\sum_{k \in C_{ij}} \|\mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})\|_2^2$ term for the bottom level sub-problems.

Kim et. al. [22][43] proposes ATC for optimal system design and presented the ATC formulations for top (super-system), middle (system) and bottom levels (sub-system). The efficacy of ATC is demonstrated through the chassis design of a support-utility vehicle given necessary analysis models. Two objectives - ride quality and handing targets are cascaded down to system and sub-system levels. Michelena et. al. [23] studies the convergence properties of ATC and proposes a convergence proof under convex and smooth assumption. Michalek and Papalambros [44] come up with a branch-and-bound approach for ATC to deal with integer variables. Kokkolaras et. al. [45] and Han et. al. [46] investigate solving multi-level problems under uncertainty.

Approximation methods have also been applied to ATC. A sequential linear programming coordination algorithm for ATC is developed and the sub-problem evaluation effort is reduced considerably [47].

Instead of treating the sum of the original objective and penalty terms as one objective, ATC is interrelated as a multi-objective optimization problem and the weighted sum method is applied to balance the tradeoff between the original objective and penalty terms in [48].

Numerical tests on ATC [49][50][51] show that reaching the accurate optimal solution needs significant computational resources. Large penalty weights are required to reduce the consistency errors between targets and response, and many sub-problems optimizations are needed in the coordination process.

2.3 Ordinary Lagrangian Method

Instead of the QP method, the Lagrangian dual function is employed to relax the consistency constraints under the ATC framework and the Ordinary Lagrangian (OL) method is developed based on the Lagrangian duality theory [52]. Since the relaxed dual term is linear, all the sub-problems in this method are independent and thus can be solved in parallel. The dual multipliers in OL are updated through the sub-gradient algorithm towards the optimal multipliers of the original problem. Extension of OL from ATC hierarchical to nonhierarchical structures is available in [53]. The drawback of the OL method is that some sub-problems may become unbounded, which leads to infinite objective values and causes convergence difficulties [54][55].

2.4 Augmented Lagrangian Coordination

Combining both ATC and OL methods, the Augmented Lagrangian Coordination (ALC) method is proposed in [55] as an efficient and robust coordination strategy. A large penalty parameter in the ATC method may cause ill-conditioning in the relaxed problem, making it hard to be solved numerically. The (ALC) method reduces the computational cost associated with ill-conditioning through the use of an Augmented Lagrangian penalty function [29].

$$\phi_{AL}(\mathbf{c}) = \mathbf{v}^T \mathbf{c} + \|\mathbf{w} \circ \mathbf{c}\|_2^2 = \sum_{i=2}^N \sum_{j \in \mathcal{E}_i} (\mathbf{v}_{ij}^T \mathbf{c}_{ij} + \|\mathbf{w}_{ij} \circ \mathbf{c}_{ij}\|_2^2) \quad (2.8)$$

For problem (2.3) in the ATC section, the general formulation for sub-problems can be obtained by replacing the quadratic penalty term $\phi(\mathbf{c})$ with the Augmented Lagrangian penalty term $\phi_{AL}(\mathbf{c})$

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{ij}} \quad & \mathbf{f}_{ij}(\bar{\mathbf{x}}_{ij}) + \mathbf{v}_{ij}^T (\mathbf{t}_{ij} - \mathbf{r}_{ij}) + \|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2 + \dots \\ & \sum_{k \in C_{ij}} \mathbf{v}_{(i+1)k}^T (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) + \sum_{k \in C_{ij}} \|\mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})\|_2^2 \\ \text{s.t.} \quad & \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\ & \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0} \end{aligned} \quad (2.9)$$

$$\text{where } \bar{\mathbf{x}}_{ij} = \{\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{c_{ij}}}\}$$

This is the ALC method applied to the ATC structure [55]. The method of multipliers is employed to update the Lagrangian multipliers after all sub-problems have been solved and the penalty is also updated to increase as the solutions tend to converge.

$$\begin{aligned} \mathbf{v}^{(k+1)} &= \mathbf{v}^{(k)} + 2\mathbf{w}^{(k)} \circ \mathbf{w}^{(k)} \circ \mathbf{c}^{(k)} \\ \mathbf{w}^{(k+1)} &= \beta \mathbf{w}^{(k)}, \quad \beta \geq 1 \end{aligned} \quad (2.10)$$

ATC uses a nested structure to carry out iterations as shown in Figure 2.2. The optimization process iterates between solving sub-problems at level 1 and level 2 until they reach a consistent solution and then their targets are passed down to sub-problems at level 3. This mechanism forms an inner loop inside the outer loop of the multiplier update. To simplify this process, the Alternating Direction Method of Multipliers (ADMM) is introduced to ALC by Tosserrams et. al. [55] which directly abandons the inner loop. All sub-problems are solved once sequentially and multipliers are then updated to prepare for the next iteration. Experiments show that ALC with ADMM

decreases computational costs by orders of magnitude ranging between 10 and 1000 compared to ATC and ALC without ADMM.

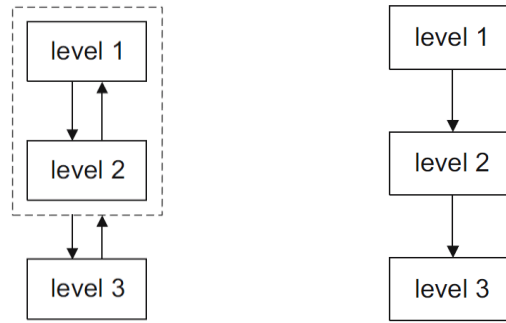


Figure 2.2: Nested ATC schemes with inner loop (left) and ADMM without inner loop (right) [55]

ALC can be applied to an ATC structure but is not limited to it. Through creating an artificial master problem, ALC is able to deal with either hierarchical or non-hierarchical partitions. Subsequent to the development of the ATC based ALC in [55], the ALC approach for quasi-separable problems [56] in MDO is proposed, in which the sub-problems are coupled through shared variables in a non-hierarchical way. The solution process involves an inner loop and an outer loop. In the inner loop, all sub-problems are solved through the block coordinate decent (BCD) method with fixed penalty weights and Lagrangian multipliers. In the outer loop, the penalty weights and Lagrangian multipliers are updated based upon the converged solution of the inner loop. The convergence tolerance of the inner loop can be set as a constant or a small but increasing variable, which results in the inexact nested method of multipliers (INMOM) and exact nested method of multipliers (ENMOM). To avoid the costly inner loop iterations at the beginning of optimization, an extreme case – the alternating direction

method of multipliers (ADMOM) is also proposed in which the inner loop is terminated in just one iteration. Numerical tests show that ADMOM is more efficient and robust than ENMOM and INMOM.

Assuming a quasi-separable problem shown below whose objective and constraint functions are coupled by linking variables \mathbf{y} .

$$\begin{aligned}
 & \min_{\mathbf{z}=[\mathbf{y}; \mathbf{x}_1; \dots; \mathbf{x}_M]} \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\
 & \text{s.t. } \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \\
 & \quad \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \\
 & \quad j = 1, \dots, M
 \end{aligned} \tag{2.11}$$

ALC solves this problem in four steps [56]

- (1) Introduction of auxiliary variables and consistency constraints
- (2) Relaxation of the consistency constraints
- (3) Formulation of the decomposed problem
- (4) Solution of the decomposed problem

The introduction of the master problem enables the sub-problems to be solved in parallel, which is a big advantage over other decomposition methods. Also, the artificial master problem is a simple quadratic programming problem, which can be solved analytically.

$$\min_{\mathbf{y}} \sum_{j=1}^M \phi_j(\mathbf{c}_j(\mathbf{y}, \mathbf{y}_j))$$

From quasi-separable problems, the solvable problems of ALC are expanded to problems with linking variables, coupling objective, and coupling constraints [57]. Two variants of ALC: the centralized ALC and the distributed ALC have been introduced to

offer more flexibility and freedom to the designers. The ALC for quasi-separable problems and ATC proposed before are shown to be subclasses of this general ALC and their convergence proof is presented based on theories in nonlinear programming. The research in [57] is applied to a special class of problems with block-separable coupling constraints [58], for which the coordinating master problem becomes a convex quadratic programming problem and the alternating direction method of multipliers is feasible. The two variants of ALC are derived and the resulting centralized ALC allows the parallel computation of all sub-problems.

ALC's capability of dealing with the non-convexity and multi-modality of complex problems is also studied empirically and the results show that ALC is able to find the global optimal solution even when the problems do not satisfy all the assumptions of its convergence proof [59]. Other complex problems that have been tested on ALC include the micro-accelerometer benchmark design problem [60] and the supersonic business jet design problem [24]. A nonhierarchical ATC is also proposed in [24] which allows nonhierarchical target-response couplings between sub-problems and options to parallel sub-problem optimizations are provided.

Due to the quadratic penalty term in the augmented Lagrangian relaxation, the sub-problems of ALC in the ATC structure are linked with each other and cannot be solved in parallel as in the ordinary Lagrangian relaxation method [52]. In order to address this issue, the Diagonal Quadratic Approximation (DQA) and Truncated Diagonal Quadratic Approximation (TDQA) methods are developed to realize the parallelization of sub-problems in the ATC structure [54]. Several relaxation methods

(Quadratic penalty, Ordinary Lagrangian Relaxation, Augmented Lagrangian, Alternating Direction Augmented Lagrangian) are summarized and tested in this paper and their numerical test results are compared with those of the DQA and TDQA. Other methods based on ALC include the Exponential Penalty Function (EPF) method for multilevel optimization [61]. Unlike the ALC, the EPF employs an exponential penalty function to relax the consistency constraints of linking variables. Double loop EPF and single loop EPF are proposed which are similar to the nested and non-nested ALC formulations. Compared with the above methods, the consensus optimization method shown below has a different structure by introducing the concept of “consensus” to optimization.

2.5 Consensus Optimization via Alternating Direction Method of Multipliers

Assume the following problem formulation

$$\begin{aligned} \min_{\mathbf{x}_j, \mathbf{y}} \sum_{j=1}^3 f_j(\mathbf{x}_j, \mathbf{y}) \\ \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}) \leq \mathbf{0}, \text{ for } i = 1, \dots, 3 \\ \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}) = \mathbf{0}, \text{ for } i = 1, \dots, 3 \end{aligned} \quad (2.12)$$

where \mathbf{x}_j is the local design variable set for each sub-problem, \mathbf{y} is the vector of coupling variables, f_j is the local objective function and $\mathbf{g}_j, \mathbf{h}_j$ are local constraint functions. The above problem is decomposable into three sub-problems as shown in Figure 2.3 (left). For sub-problem 1 the coupling variables are $\mathbf{y}_1 = [\mathbf{y}_{12} \ \mathbf{y}_{13}]$, for sub-problem 2, $\mathbf{y}_2 = [\mathbf{y}_{12} \ \mathbf{y}_{23}]$, and for sub-problem 3, $\mathbf{y}_3 = [\mathbf{y}_{13} \ \mathbf{y}_{23}]$.

To decouple the sub-problems, copies for $\mathbf{y}_{12}, \mathbf{y}_{13}, \mathbf{y}_{23}$ are introduced at each sub-problem $[\mathbf{y}_{12}]_1, [\mathbf{y}_{12}]_2, [\mathbf{y}_{13}]_1, [\mathbf{y}_{13}]_3, [\mathbf{y}_{23}]_2, [\mathbf{y}_{23}]_3$, and $\mathbf{y}_1 = [[\mathbf{y}_{12}]_1 \ [\mathbf{y}_{13}]_1]$, $\mathbf{y}_2 = [[\mathbf{y}_{12}]_2$

$[\mathbf{y}_{23}]_2$, $\mathbf{y}_3 = [[\mathbf{y}_{13}]_3 \ [\mathbf{y}_{23}]_3]$. The consensus variables \mathbf{z}_j are adopted to ensure all these copies are consistent $\mathbf{z}_j - \mathbf{y}_j = \mathbf{0}$.

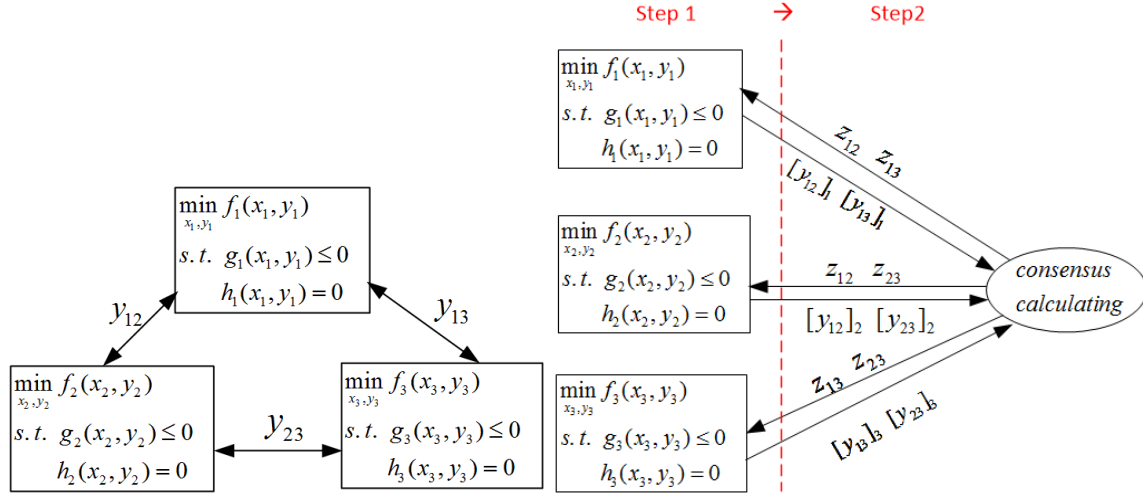


Figure 2.3: Original problem (left) and decoupled problem (right) for CADMM

The decomposed formulation obtained introducing the consensus variables is shown in Figure 2 (right).

$$\begin{aligned}
 & \min_{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3} \sum_{j=1}^3 f_j(\mathbf{x}_j, \mathbf{y}_j) \\
 & \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) < \mathbf{0} \\
 & \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \\
 & \quad \mathbf{c}(\mathbf{z}_j, \mathbf{y}_j) = \mathbf{z}_j - \mathbf{y}_j = \mathbf{0}, \text{ for } j = 1, \dots, 3
 \end{aligned} \tag{2.13}$$

The following formulation is obtained relaxing the consistency constraints to the objective function

$$\begin{aligned}
 & \min_{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3} \sum_{j=1}^3 f_j(\mathbf{x}_j, \mathbf{y}_j) + \sum_{j=1}^3 \mathbf{v}_j^T (\mathbf{y}_j - \mathbf{z}_j) + \sum_{j=1}^3 \frac{\rho}{2} \|\mathbf{y}_j - \mathbf{z}_j\|^2 \\
 & \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
 & \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0}, \text{ for } j = 1, \dots, 3
 \end{aligned} \tag{2.14}$$

where \mathbf{v}_j is the vector of Lagrangian multipliers of sub-problem j , ρ is the weight for the quadratic penalty term .

Problem (2.14) can be solved using the block coordinate descent (BCD) iterative method: the primal variables \mathbf{x}_j , \mathbf{y}_j and \mathbf{z}_j are collected in two blocks $[\mathbf{x}_j, \mathbf{y}_j]$ and $[\mathbf{z}_j]$ that are updated sequentially [25]. Block $[\mathbf{x}_j, \mathbf{y}_j]$ ($[\mathbf{z}_j]$) is obtained solving problem (2.14) while considering $[\mathbf{z}_j]$ ($[\mathbf{x}_j, \mathbf{y}_j]$) constant and equal to the most recent update. The decomposition paradigm is exploited in the first step when $[\mathbf{x}_j, \mathbf{y}_j]$ can be computed optimizing each sub-problem independently (Step 1 in Figure 2.3 (right)). Note that the consensus variables \mathbf{z}_j are obtained in a closed form since problem (2.14) is an unconstrained quadratic problem with respect to \mathbf{z}_j (Step 2 in Figure 2.3 (right)). Once the primal variables are obtained, the dual variables must be calibrated in order to achieve the feasibility of the consistency constraints. These steps read [25][26][27]

$$\begin{aligned}
 \text{step 1: } \mathbf{y}_j &= \arg \min_{\mathbf{x}_j, \mathbf{y}_j} (f_i(\mathbf{x}_j, \mathbf{y}_j) + (\mathbf{v}_j^k)^\top (\mathbf{y}_j - \mathbf{z}_j) + \frac{\rho}{2} \|\mathbf{y}_j - \mathbf{z}_j\|^2) \\
 \text{step 2: } \mathbf{z}_{ij} &= \frac{1}{2} (\mathbf{y}_{ij} + \mathbf{y}_{ji} + \frac{1}{\rho} \mathbf{v}_{ij}^k + \frac{1}{\rho} \mathbf{v}_{ji}^k), \quad i \in N(j) \\
 \mathbf{v}_{ij}^{k+1} &= \mathbf{v}_{ij}^k + \rho (\mathbf{y}_{ij} - \mathbf{z}_{ij}), \quad i \in N(j), j = 1, \dots, 3
 \end{aligned} \tag{2.15}$$

$N(j)$ is the set of indices of sub-problems that share the same coupling variables with sub-problem j . \mathbf{y}_{ij} are the coupling variables shared by sub-problems i and j . \mathbf{v}_{ij} are the Lagrangian multipliers associated with \mathbf{y}_{ij} .

The optimization terminates when the consistency error between the copies of the coupling variables and the consensus are satisfied within a desired tolerance and no improvement on the objective functions is possible. The consistency error is calculated as follows:

$$\mathbf{c}_{ij} = \mathbf{z}_{ij} - \mathbf{y}_{ij}, \quad \text{for } i \in N(j), \quad j = 1, \dots, 3 \quad (2.16)$$

2.6 Weight Selection and Update in Decomposition based Optimization

Assuming \mathbf{c} is the vector of consistency constraints and ϕ is the relaxation function, the relaxation functions in ATC, OL, ALC and CADMM can be summarized in Eq.(2.17). Other relaxation methods in mathematical optimization have also been applied to coordination strategies, such as the exponential penalty function [61], but they are not the focus of this research.

$$\begin{aligned} \text{ATC:} \quad & \phi_Q(\mathbf{c}) = \|\mathbf{w} \circ \mathbf{c}\|_2^2 \\ \text{OL:} \quad & \phi_L(\mathbf{c}) = \mathbf{v}^T \mathbf{c} \\ \text{ALC \& CADMM:} \quad & \phi_{AL}(\mathbf{c}) = \mathbf{v}^T \mathbf{c} + \|\mathbf{w} \circ \mathbf{c}\|_2^2 \end{aligned} \quad (2.17)$$

Here \mathbf{v} is the vector of Lagrangian multipliers and \mathbf{w} is the vector of penalty weights. Both of them need to be updated at each iteration following certain rules. For multipliers, the sub-gradient method and the method of multipliers are widely used to perform the update procedure. Various numerical tests have shown the effectiveness of these update strategies for reaching the optimal multipliers [52][62].

The weight selection and update are crucial to the efficiency and convergence of optimization [29][63][64]. For the weight in the quadratic penalty function, early versions of ATC adopt a fixed weight value through the whole iterative process. As a result, a large weight is required to generate satisfying results in many cases which often causes computational difficulties [50]. To address this issue, a nested iterative weight update method (WUM) is developed to find the appropriate weight for ATC [50]. In the inner loop of WUM, the decomposed problem is solved with a fixed weight. In the outer loop,

the weight is updated based on the results of the inner loop and another inner loop is started until the user-defined inconsistency tolerance is achieved.

For the weight in the augmented Lagrangian relaxation, the ALC method sets the initial weight to a relatively small value, and then increases the weight at each iteration until reaching optimality.

$$\mathbf{w}_j^{k+1} = \begin{cases} \mathbf{w}_j^k & \text{if } |\mathbf{c}_j^k| \leq \gamma |\mathbf{c}_j^{k-1}| \\ \beta \mathbf{w}_j^k & \text{if } |\mathbf{c}_j^k| > \gamma |\mathbf{c}_j^{k-1}| \end{cases} \quad (2.18)$$

where $\beta > 1, 0 < \gamma < 1$

If the decrease of the consistency error \mathbf{c} is large enough, the value of \mathbf{w} is kept to the next iteration. If the decrease is not satisfactory, this means that the penalty applied to the violation of the consistency error is insufficient and \mathbf{w} is multiplied by a factor larger than 1 and is thus increased. In this case \mathbf{w} always increases. Experimental results show that this strategy is much more efficient than WUM and reduces significantly the use of computational resources [55]. However, this monotone update's capability still greatly depends on the initial weight selection. A large weight may cause ill-conditioning and numerical difficulties while a small weight can slow down the convergence rate or even result in a solution far from the optimum [65].

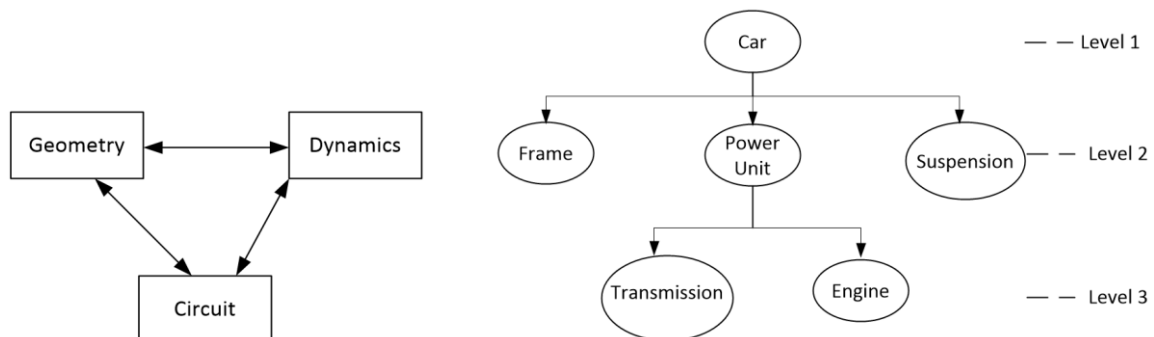
Based on the above literature review and different requirements emerging from engineering design optimization, the proposed research is described in the next chapters. It is aimed at addressing the issues of accuracy, efficiency, and parallelism in decomposition based optimization through NTC.

Chapter 3

Hybrid Partition with Both Component and Discipline Sub-problems

3.1 Hybrid partition

In the practical engineering design process, it is common to modify a decomposition during the design process. The introduction of a new component (discipline) to a discipline- (component-) based decomposition reflects the dynamics of the design process as requirements and criteria evolve. This forms a hybrid network optimization problem in which the interacting nodes are either components or disciplines [53]. Some methods such as Analytical Target Cascading (ATC) imply a hierarchical structure and may not be directly applicable to this kind of hybrid network optimization problem.



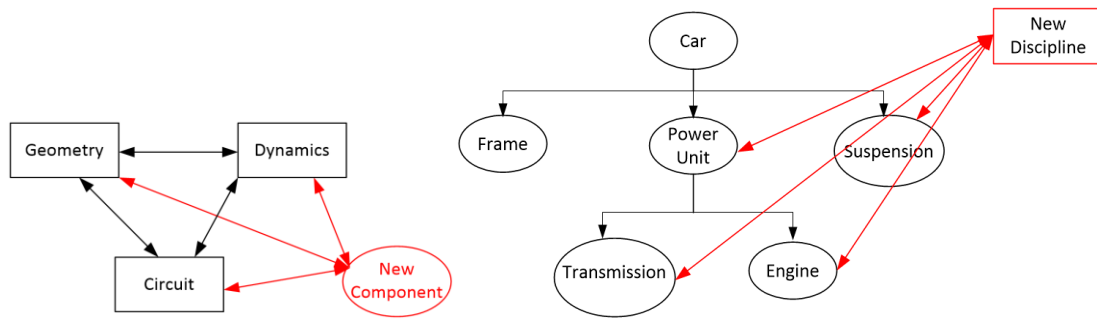


Figure 3.1: top left - discipline-based decomposition; top right – component based decomposition; bottom – hybrid decompositions (rectangle stands for discipline, circle stands for component)

Consensus Optimization via the Alternating Direction Method of Multipliers (CADMM) for Network Target Coordination (NTC) is a distributed coordination method designed to optimize sub-systems that are decomposed in a nonhierarchical fashion. Through the consensus variables, coupled systems are solved concurrently and the method of multipliers is used to efficiently reduce the inconsistency between the linking variables. Its efficacy has been proved by some engineering examples [25][26][27].

In this chapter, we adopt CADMM to solve two hybrid nonhierarchical formulations of the ADXL150 micro-accelerometer design problem [60]. Both problems involve three disciplines (Structures, Dynamics, Electrostatics) and several components (Proof mass, U-spring, Comb, etc.). By adding and removing disciplines and/or components, the flexibility of the hybrid network (nonhierarchical) is explored. The results are discussed and conclusions drawn with respect to the modeling capabilities and computational efficiency of the proposed approach. The research presented in this chapter is also published in [21].

3.2 Micro-accelerometer Benchmark Problem and its Hybrid Decomposition

The micro-accelerometer problem [60][66][67] is a non-convex, nonlinear engineering test problem which is proposed as a benchmark problem for testing and comparing different multidisciplinary optimization methods. It involves four disciplines: Structures, Electrostatics, Dynamics, and Circuits. The four disciplines may depend on the same design variables or one discipline can depend on the output of another discipline. There are 22 design variables at most. The design objective is to minimize the footprint area A (which is proportional to fabrication cost). The design constraints are requirements that make sure the performance with respect to sensitivity, noise, and range is at least as good as the baseline design.

This problem can be decomposed in four ways and solved using the Augmented Lagrangian Method (ALC). Details about this problem can be found in reference [60].

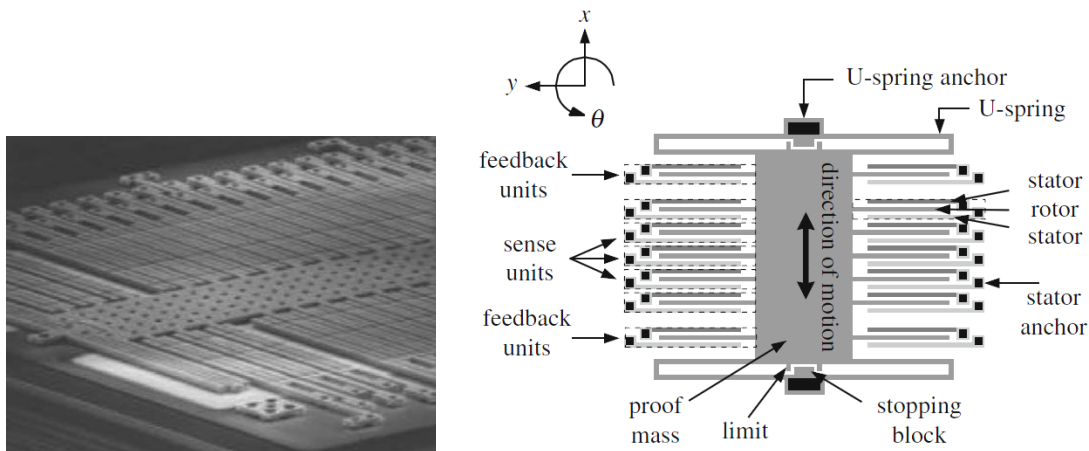


Figure 3.2: Microscope image and schematic illustration (right) of the micro-accelerometer [60]

In this section, by adding and removing disciplines or components, the flexibility of the hybrid network (non-hierarchical) is explored. We propose two hybrid decompositions for solving this problem. In decomposition 1, a new discipline (Dynamics) is added into the original component decomposed problem, resulting in a hybrid problem. In decomposition 2, a new component (Springs) is added into the original discipline decomposed problem. Both decompositions do not include design variables and constraints in the Circuit discipline, which is one aspect considered in [60]. The number of design variables is 16.

3.2.1 Hybrid Decomposition 1

The original problem is decomposed into two components: Springs and Mass-Fingers. These two components share five design variables. Mass & Fingers also needs the output of Springs ($k_{x,m}$) as its parameter. The two components only involve two disciplines: Mechanics and Electrostatics. The constraints g_s and g_e correspond to the two disciplines and g_s is distributed among the different components.

When we add Dynamics as a new discipline to the original problem, the new sub-problem needs the outputs of Springs and Mass & Fingers. As shown in Figure 3.3, for this decomposition, each sub-problem is coupled with the other two sub-problems. It is a typical network target coordination problem.

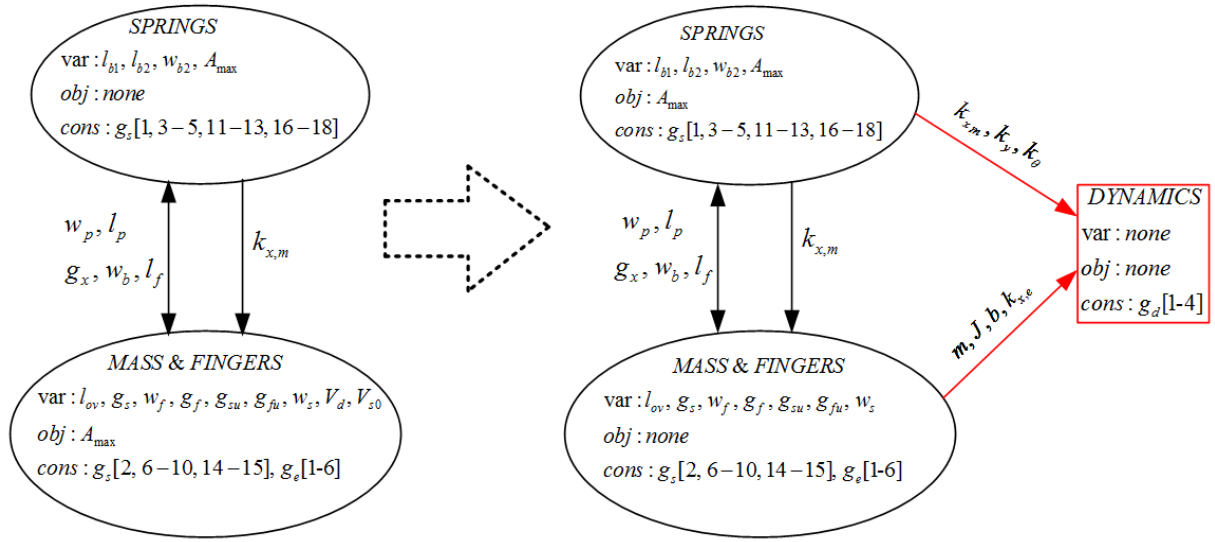


Figure 3.3: Hybrid decomposition 1 for the micro-accelerometer problem

3.2.2 Hybrid Decomposition 2

The original problem is decomposed into three disciplines: Mechanics, Electrostatics and Dynamics. They couple with each other either through sharing the same design variable or linking functions. In addition to g_s and g_e , g_d is the constraint corresponding to the dynamics discipline.

When we add springs as a new component to this original problem, this new sub-problem shares 6 design variables with the mechanics sub-problem, and generates stiffness for the dynamics and electrostatics sub-problems. The resulting hybrid decomposition is shown in Figure 3.4. Each sub-problem in this decomposition network problem is coupled with the other three sub-problems.

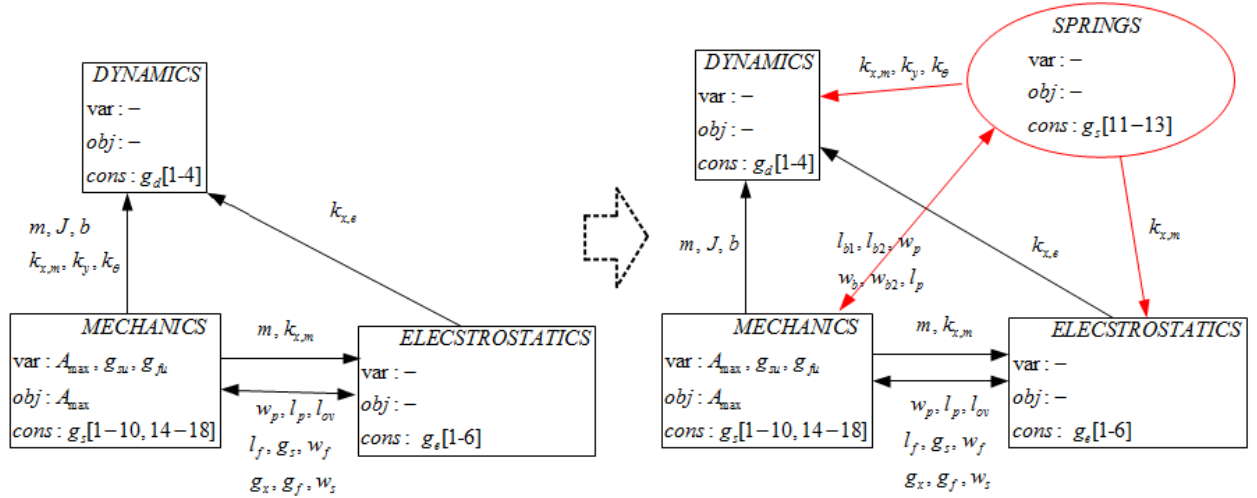


Figure 3.4: Hybrid decomposition 1 for the micro-accelerometer problem

3.3 Application of CADMM to Hybrid Micro-accelerometer Problems and Results

3.3.1 Experimental Setup

In this research, numerical experiments are conducted in Matlab R2012a [68]. The constrained minimization function “*fmincon*” is used.

The limit for the number of iterations of the optimization loop is 1e3. The program converges if the consistency error (Eq. (2.16)) is less than the termination tolerance $\varepsilon = 1e-3$. The initial starting points for each sub-problem are randomly selected between 50% - 150% of the baseline design value. At each iteration, the *exitflag* output of *fmincon* of each sub-problem is checked to ensure they converge. If one sub-problem does not converge within 10 *fmincon* tries, its optimal value from the last iteration is used for the optimization of the other sub-problems and parameter updating.

The initial value of ρ is 0.03. Instead of using a fixed penalty parameter ρ , we update ρ based on the inconsistency error as follows with $\zeta = 0.99$ and $\lambda = 1.01$:

$$\begin{cases} \text{if } \max(\mathbf{c}^k) / \max(\mathbf{c}^{k-1}) > \xi, \rho^{k+1} = \lambda * \rho^k \\ \text{if } \max(\mathbf{c}^k) / \max(\mathbf{c}^{k-1}) \leq \xi, \rho^{k+1} = \rho^k \end{cases} \quad (3.1)$$

In this way, the penalty term is small at the beginning of the optimization process. Thus, during the first iterations, the objective function is minimized and then, as the penalty weight increases, the consistency error becomes more important and the consistency error is reduced.

The optimal objective value of the All-in-one (AIO) formulation is $4.5322\text{e-}08 \text{ m}^2$, which is used as the reference solution for both problems.

3.3.2 Results and Discussion

Table 3.1 summarizes the numerical results of both problems. For each hybrid problem, ten runs are attempted starting from different random initial points. The percentage in Table 3.1 indicates how many runs have converged out of all test runs. All three metrics are statistically categorized into min, mean and max.

Table 3.1: Optimization results of CADMM applied to the two kinds of hybrid decompositions (reference solution = 0.0453)

		<i>Area(mm²)</i>	<i># of iters.</i>	<i>Max_vari_error</i>
Hybrid 1 90%	Min	0.0458	459	0.7549
	Mean	0.0467	538	1.1135
	Max	0.0533	769	2.4103
Hybrid 2 90%	Min	0.0457	367	0.9253
	Mean	0.0460	434	0.9589
	Max	0.0475	589	1.1281

It can be seen that the results for both hybrid decompositions are very close to the reference optimal solution (with mean value 0.0467 for hybrid 1 and 0.0460 for hybrid 2). The high number of iterations may be due to the small initial value for the penalty term ρ

and update parameter λ for ρ . One explanation for the high design variable error is that the problems have many local optimal solutions which are far from the reference solution. Another is that the objective value is not sensitive to some design variables. So although the values for these design variables differ a lot from the reference solution, the objective does not.

3.4 Summary

The hybrid component-discipline based decomposition is proposed in this chapter, in which the sub-problems are coupled in a network way, thus each sub-problem may interact with any of the other sub-problems. This kind of structure is difficult for multilevel hierarchical decomposition methods such as ATC to handle.

By fully exploring the flexibility of decomposition of the micro-accelerometer benchmark problem, two kinds of hybrid network decompositions are proposed. One is adding a new discipline to a component-decomposed problem; the other one is adding a new component to a discipline-decomposed problem. The CADMM is employed to solve these two decompositions. Numerical experiments show that the optimization results of the CADMM are very close to the reference optimal solution (with an error less than 3%). This demonstrates that the CADMM is able to deal with the hybrid network decomposition problem and supports component-discipline decomposition and subsystem optimization to solve the overall problem.

Chapter 4

Alternative Structures of ALC Under the Same Partition

Augmented Lagrangian coordination (ALC) is a powerful and flexible coordination approach for solving decomposed problems with a network structure. Depending on how to handle the couplings between sub-problems and whether a master problem is created, there are different alternative structures for ALC: centralized, distributed or hierarchical. This chapter will compare these structures both analytically and numerically.

4.1 Alternative structures of ALC

Consider a decomposed quasi-separable problem with a nonhierarchical partition shown in Figure 4.1. There are four sub-problems fully coupled with each other through the linking variables y_{ij} , $i, j \in \{1, 2, 3, 4\}$, $i < j$, and $i \neq j$. Several variants of ALC can be employed to solve this problem: The distributed ALC (Figure 4.2(a)) can use the current partition and directly deal with the couplings between sub-problems; The centralized ALC (Figure 4.2(b)) creates an artificial problem as a master problem, cuts off all the original couplings (solid lines) in the partition, adds new links (dash lines) between each sub-problem and the master problem, and coordinate all sub-problems through the master

problem; The extended ATC based ALC (Figure 4.2(c, d)) forms a hierarchical multilevel structure, assigns the roles of parents and children to the previous equal sub-problems, and puts them on different levels; During this process, certain original couplings are cut off and several new couplings associated with the cut couplings are created and passed through intermediate sub-problems. Consider the y_{23} in the original partition which is the linking variable that just exists between sub-problems 2 and 3, in the ATC-3 levels in Figure 4.2(d), the y_{23} is deleted and copies of it are created and the coupling of sub-problems 2 and 3 are dealt through the intermediate sub-problems (sub-problem 1 and 4) in an implicit fashion. Depending on how this cutting and reconnecting process is conducted, different ATC structures with different number of levels can be achieved. Also it should be noted that there are more possible structures for this four-node network-decomposed problem beyond the ones shown in Figure 4.2. In this chapter we mainly discuss distributed ALC, centralized ALC, and extended ATC using ALC.

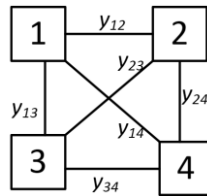


Figure 4.1: A four-node decomposed problem with a network structure

The distributed ALC needs to optimize the sub-problems sequentially because the couplings between sub-problems are handled directly. The centralized ALC adopts a two-level hierarchical structure in which once the optimization of the master problem is finished, which is very easy since it is a quadratic optimization problem with analytical

solutions, all sub-problems can be solved in parallel. Partial parallelization can be achieved for ATC since sub-problems on the same level are not dependent on each other thus can be solved at the same time.

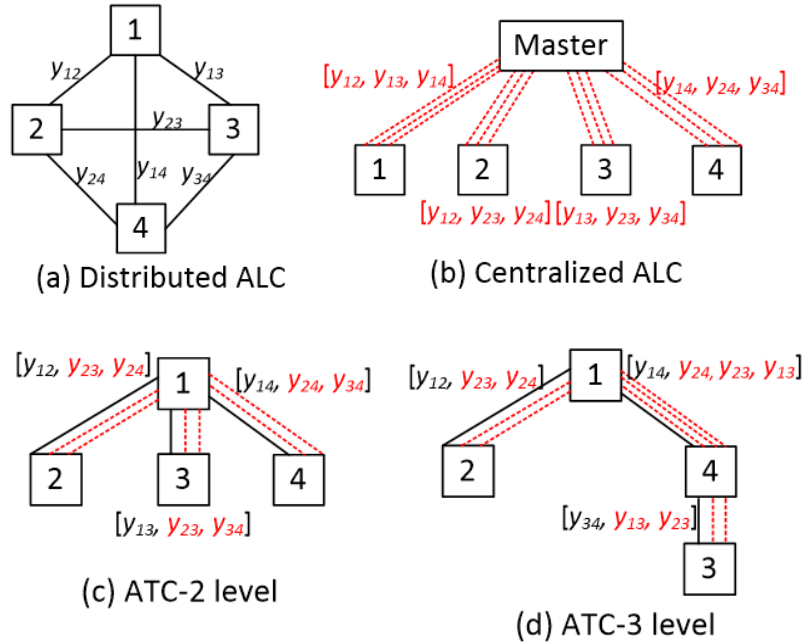


Figure 4.2: Four different structures can be used to solve the example problem through ALC

It can be seen that the methods in Figure 4.2 have a different number of levels and copies of linking variables, which results in a different number of consistency constraints. This makes their dual problems have a different number of design variables and their optimization require a different amount of coordination effort, which leads to different performances. Some of these variants are tested on complex systems[9][20], however the problem partition in those tests are different for different ALC variants, which makes it difficult to analyze the causes of the differences in the results. In this research, the performances of the distributed ALC, the centralized ALC, and the ATC based on ALC

are explored and compared numerically using the same partition. The results show that since the distributed ALC introduces the least copies of linking variables, it can reach a better solution while consuming less computational resources than other alternative ALC structures. However, the decomposed sub-problems in distributed ALC need to be solved sequentially. The introduction of an artificial master problem in the centralized ALC enables all sub-problems to be optimized in parallel, but the solution is not as accurate as that in the distributed ALC.

4.2 Comparison of the Distributed ALC, the Centralized ALC, and the ATC Extended by ALC through Numerical Tests

One mathematical and one engineering test problems are used to test the three variants of ALC. The results presented here are different from our preliminary research in [69]. The reason is that the initial weights in [69] are set based on experience to the values that works best for each ALC, while in this research they are set through Eq. (2.18) to make the results more objective. Also, the name “alternatives of ALC” is used to refer to the “solving structures” in [69].

4.2.1 Geometric optimization problem

The first example is a standard nonconvex test problem which appears in earlier work on ATC and ALC [22][20][54][49][70][71]. It can be partitioned into five sub-problems coupled through a network structure as shown in Figure 4.3. Its AIO problem can be easily solved with a unique solution and it is used as the reference solution.

$$\begin{aligned}
& \min_{z_1, \dots, z_{14}} f = z_1^2 + z_2^2 \\
& \text{subject to} \quad g_1 = (z_3^{-2} + z_4^2)z_5^{-2} - 1 \leq 0, \\
& \quad g_2 = (z_5^2 + z_6^{-2})z_7^{-2} - 1 \leq 0, \\
& \quad g_3 = (z_8^2 + z_9^2)z_{11}^{-2} - 1 \leq 0, \\
& \quad g_4 = (z_8^{-2} + z_{10}^2)z_{11}^{-2} - 1 \leq 0, \\
& \quad g_5 = (z_{11}^2 + z_{12}^{-2})z_{13}^{-2} - 1 \leq 0, \\
& \quad g_6 = (z_{11}^2 + z_{12}^2)z_{14}^{-2} - 1 \leq 0, \\
& \quad h_1 = (z_3^2 + z_4^{-2} + z_5^2)z_1^{-2} - 1 = 0, \\
& \quad h_2 = (z_5^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0, \\
& \quad h_3 = (z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2)z_3^{-2} - 1 = 0, \\
& \quad h_4 = (z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2)z_6^{-2} - 1 = 0, \\
& \quad z_1, \dots, z_{14} > 0
\end{aligned} \tag{4.1}$$

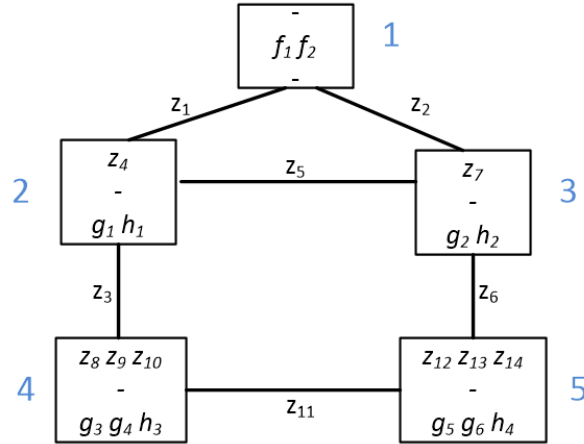


Figure 4.3: Partition used for the geometric optimization problem with five sub-problems

Distributed ALC, Centralized ALC, and two ATC approaches are proposed to coordinate the optimization of the problem with the chosen partition. As shown in Figure 4.4, these methods have a different number of levels (2 to 4) and couplings (equals to the number of multipliers when these couplings are relaxed) (6 to 12). Note that the number of levels for a nonhierarchical solving structure is defined implicitly. In the distributed

ALC, sub-problems 3 and 4 can be solved in parallel which implies that the algorithm is sequentially applied to sub-problems 1, 2, then 3 and 4, and finally to sub-problem 5. The sequential way the algorithm works implicitly defines 4 levels for this structure. All four methods adopt the Augmented Lagrangian Relaxation to relax the introduced equality constraints generated by the auxiliary copies of the linking variables. The distributed ALC solves all the sub-problems sequentially. It does not change anything to the partition in Figure 4.3, thus introduces only 6 copies of linking variables. In the centralized ALC every original coupling is cut off and each sub-problem is coordinated through the newly created master problem, this process introduces the largest number of copies of linking variables.

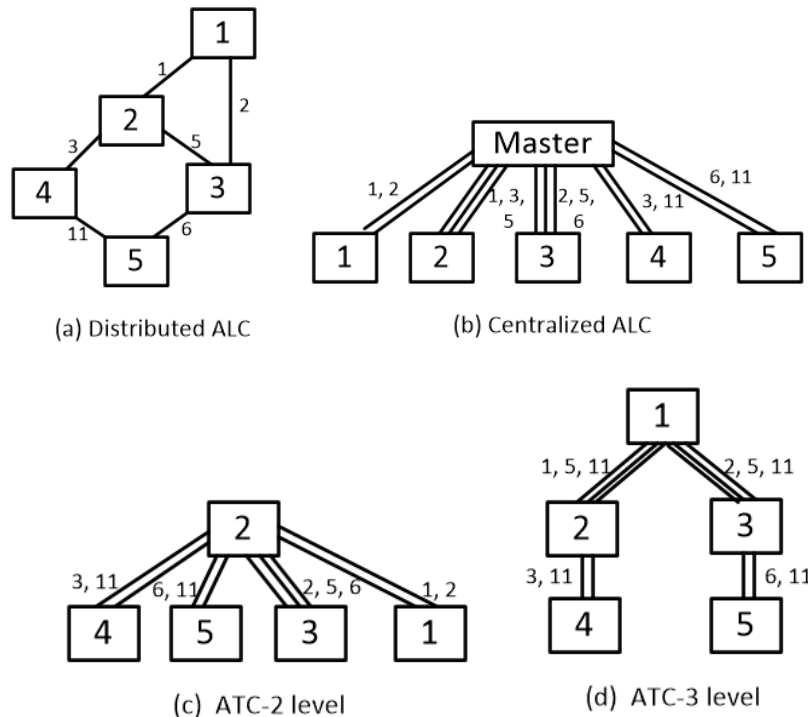


Figure 4.4: Four possible solving structures for the geometric problem

The tests are conducted in MatLab 2012 and the solver “*fmincon*” is chosen to solve the sub-problems, which uses the interior point as the default algorithm but may switch to other algorithms when necessary. The termination tolerance is $\varepsilon = 1e-3$. The initial design variables are set randomly between 0 and 1. The initial Lagrangian multipliers are all set to zero. For each case, ten runs are performed and the average value of the number of iterations and function evaluations are summarized in the following table. The initial weight setting method proposed in [20] is used to determine the initial w :

$$w = \frac{\alpha |\hat{f}|}{\sqrt{\sum_{j=1}^M \|\hat{\mathbf{c}}_j\|_2^2}} \quad (4.2)$$

where \hat{f} and $\hat{\mathbf{c}}_j$ are estimates of the objective value and the consistency error before starting the optimization. α is a fraction number chosen by the user within the range $1e-3 < \alpha < 1$. The initial weight \mathbf{w} is set by Eq. (4.2) with $\hat{f} = 50$, $\alpha = 0.1$, and $\hat{\mathbf{c}}_j$ computed using $\mathbf{w}^0 = 0.001$.

Table 4.1: Results for solving geometric problem using different solving structures

<i>Methods</i>	<i># of levels</i>	<i># of multipliers</i>	<i># of iters.</i>	<i># of func.evals.</i>
Distributed ALC	4	6	49	6828
Centralized ALC	2	12	61	11142
ATC – 2 level	2	9	52	9907
ATC – 3 level	3	10	56	10990

It can be seen that: Distributed ALC performs the best out of the four methods. It consumes the least number of function evaluations and iterations. While centralized ALC needs the most number of function evaluations and iterations. The performance of ATC-2

level and ATC-3 level are better than centralized ALC, but still much worse than distributed ALC. As the number of multipliers gets larger, the number of function evaluations also gets larger, which means more computational resources are needed with the increase of the number of multipliers. In other words, the computational costs of the four methods are proportional to their number of multipliers. Their ratio of number of multipliers is 1:2:1.5:1.7, which is close to their ratio of number of function evaluations - 1:1.6:1.4:1.6. This can be explained from two perspectives: (1) The number of multipliers equals to the number of relaxed consistency constraints. The bigger this number is, the more effort is required to drive the sub-problem optimization towards a consistent solution; (2) The coordination process of ALC can be interpreted as the process of solving the dual problem of the original problem. The number of multipliers equals to the number of design variables for this dual problem, thus a smaller number of design variables can make the optimization converge faster.

Note that this example is relatively simple and all four methods reach the global optimal solution. More research on the solution accuracy of different solving structures is presented in the next test problem.

4.2.2 Micro-accelerometer design problem

The benchmark problem - micro-accelerometer design problem in Chapter 3 is used again. Here case 3 [60] is chosen and the partition shown in Figure 4.5 with three sub-problems coupled in a network structure.

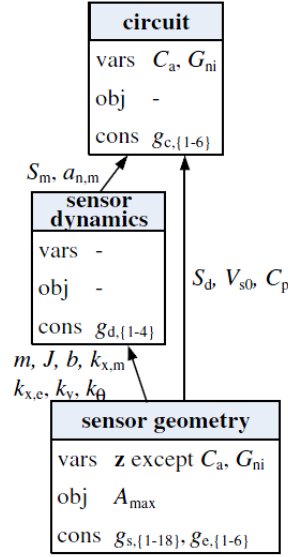


Figure 4.5: Object based partition of the micro-accelerometer benchmark problem [60]

Distributed ALC, centralized ALC and ATC are applied on this problem. As shown in Figure 4.6, distributed ALC directly uses the problem partition, while centralized ALC creates a master problem and ATC transforms the geometry sub-problem to a master problem to coordinate other sub-problems.

The termination tolerance is $\varepsilon = 1e-3$. The initial design variables are set randomly between 50% and 150% of the baseline design. The initial Lagrangian multipliers are all set to zero. The initial weight \mathbf{w} is calculated by Eq. (4.2) with $\hat{f} = 0.5$, $\alpha = 0.1$, and $\mathbf{w}^0 = 0.001$. For each method, ten runs are performed and the average value of the number of iterations and function evaluations are summarized in Table 4.3. The optimal solutions for the All-in-One formulation are summarized in Table 4.2 and the best objective is 0.0807 mm^2 , which is used to calculate the objective error for decomposed based optimization.

Table 4.2: Optimization results of the micro-accelerometer problem (case 3) without decomposition (AIO)[60]

	Min	Mean	Max
Objective (mm ²) (15% of 100 runs have converged)	0.0807	0.0810	0.0817

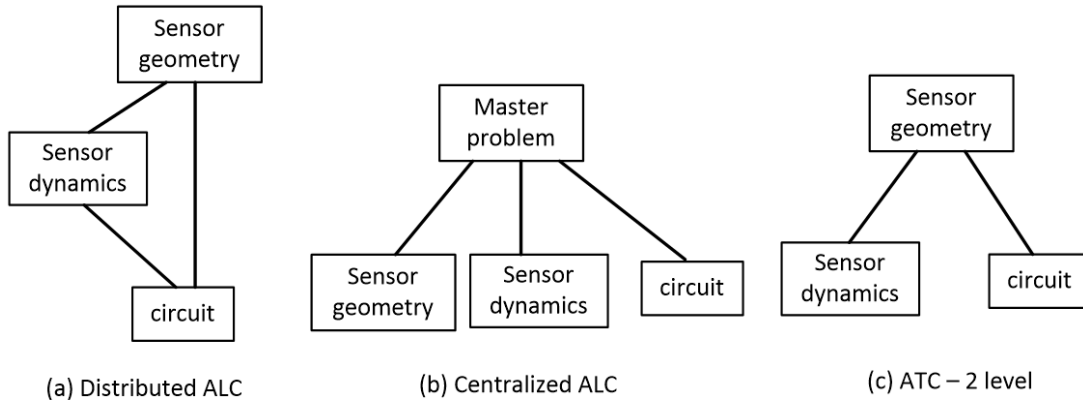


Figure 4.6: Three solving structures for the micro-accelerometer problem for object based partition

From Table 4.3, it is obvious that distributed ALC achieves the best result (objective error = 1.7%) within the least number of function evaluations (18432) among the three coordination methods. Centralized ALC structure with an artificial master problem costs the most computational resource (the average number of function evaluations is 48954, which is 166% more than that of the distributed ALC) and converges to the worst result (error = 12.4%). Again the number of function evaluations and the objective error of the ATC–2 level are between those of distributed ALC and centralized ALC.

Table 4.3: Results for different solving structures of the micro - accelerometer problem

<i>Methods</i>	<i># of levels</i>	<i># of multipliers</i>	<i># of iters.</i>	<i># of func. evals.</i>	<i>Optimal result / error</i>
Distributed ALC	3	12	31	18432	0.0821/1.7%
Centralized ALC	2	24	69	48954	0.0907/12.4%
ATC – 2 level	2	14	48	32906	0.0852/5.6%

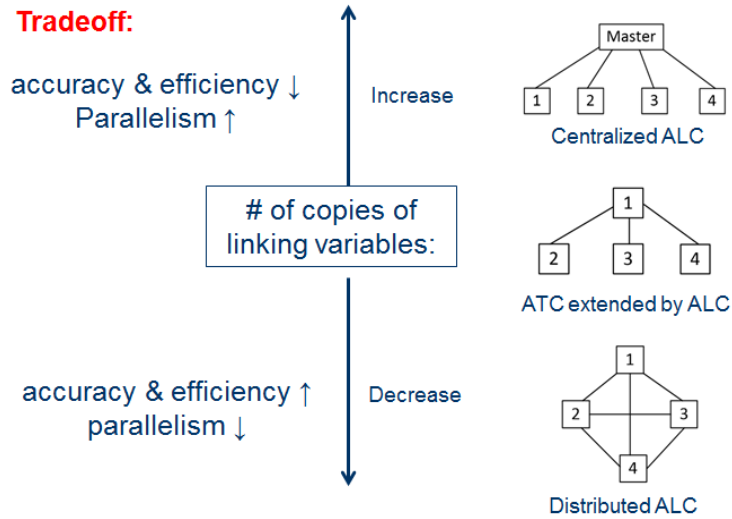


Figure 4.7: The effect of number of copies of linking variables in ALC and the tradeoff it causes between accuracy, efficiency, and parallelism.

Through the above two test problems, we can see that the number of copies of linking variables (generally equals to the number of multipliers) plays an important role when determining the performance of a coordination strategy, which can be illustrated through Figure 4.7. For the alternative structures of ALC discussed in this chapter, the number of copies increases in the order of distributed ALC < ATC < centralized ALC, and so does the parallelism of the three methods, while the accuracy and efficiency decrease in the order of distributed ALC > ATC > centralized ALC. Centralized ALC sacrifices efficiency and accuracy in exchange for the parallel computation of sub-

problems. By creating a master problem and introducing additional copies of linking variables, all sub-problems are put on the same level and can be optimized independently once the master problem optimization is finished, which is very easy since the master problem is an unconstrained quadratic programming problem with analytical solutions. However, for many engineering problems, the number of disciplines they can be partitioned into is limited, thus the number of sub-problems is small. Also, each design team in real life generally is not committed to just one job at a time. When these teams are waiting for inputs from other design teams, they may switch to other tasks instead of remaining idle. In these situations, the parallel computation is not as important as the efficiency and accuracy due to limited resources and increasing quality requirements of the market.

The ATC method just changes the couplings that are necessary to be modified to form a hierarchical structure, thus it needs a few extra copies of linking variables. Its accuracy and efficiency also is much better than that of centralized ALC, and its sub-problems can be solved partially in parallel because sub-problems located on the same level are independent from each other. Distributed ALC is the most natural coordination strategy since it directly handles the couplings that exist in the problem partition and introduces the minimal number of copies of linking variables. Compared to centralized ALC and ATC, Distributed ALC can reach the best solution while consuming the least computational resources, but its sub-problems are fully coupled thus need to be solved sequentially.

4.3 Summary

In this chapter, starting with a fully coupled network-decomposed problem, several alternative ALC structures are proposed to solve the decomposed problem, which have different number of consistency constraints and number of levels. A geometric programming problem is used to illustrate the effects of these structures and the results show that there are great differences between the performances of these ALC alternatives when solving the same problem using the same partition. The numerical tests on the micro-accelerometer results indicate that the distributed ALC method can reach a better design while consuming less function evaluations, compared to the centralized ALC and hierarchical ALC.

The distributed ALC is shown to be the most accurate ALC and requires the least computational resources, but the sub-problems in this method need to be solved sequentially. When the scale of the problems grows much larger, the capability of parallel computation of all sub-problems of the centralized ALC is more desirable and preferable. This tradeoff between solution accuracy, efficiency, and parallel computation indicates that there is no single best ALC method and it totally depends on the engineers to choose which method works better for their specific scenario.

The next two chapters focus on centralized ALC and distributed ALC respectively, and try to improve their performance through the dual residual theory and a new flexible weight update.

Chapter 5

Dual Residual for Centralized ALC Based on Optimality Conditions

Besides the area of decomposition based optimization of complex problems, the augmented Lagrangian relaxation is also widely used in convex optimization, statistics, machine learning, and other areas [72]. In particular, advancements in the Alternating Direction method have been achieved for separable Variational Inequality problems [73][74]. A penalty parameter update strategy is proposed which considers both the “primal residual” and the “dual residual” and drives both of them to zero. This update has been shown to be more flexible and efficient through numerical tests.

Inspired by their work, this chapter explores the “dual residual” in the context of optimization by decomposition, and the potential benefits it can bring to the ALC method. Although the convergence proof of ALC in [20] assumes that the weight \mathbf{w} is non-decreasing during the iterative process, this does not necessarily mean that decreasing \mathbf{w} will lead to optimization failure, which is therefore worth investigating in practice.

In this chapter, the effects of weight \mathbf{w} on the convergence of the centralized ALC are analyzed through the application of the Karush-Kuhn-Tucker (KKT) optimality conditions to the decomposed problem. The new terms, “primal residual” and “dual

residual”, are derived and defined mathematically. A new weight update strategy considering both the primal and dual residuals which drives the dual residual to zero in the optimization process is developed. The proposed strategy can either increase or decrease the weight \mathbf{w} , thus is more flexible and is expected to avoid the ill-conditioning caused by a large initial \mathbf{w} in the monotone weight update. Moreover, since the strategy is based on the optimality conditions of the AIO problem which are not fully covered in the traditional update, the strategy is anticipated to improve the centralized ALC’s ability to generate optimal solutions.

The rest of the chapter is organized as follows. The optimality conditions of AIO and decomposed problems in centralized ALC, including the derivation of the new terms “primal residual” and ”dual residual”, are introduced in Section 5.1. The new non-monotone weight update and an auxiliary convergence check criterion, which drive both primal and dual residuals to zero, are presented in Section 5.2. In Section 5.3, numerical tests on the proposed strategy are conducted on three examples and the results prove the advantages of the proposed update over the traditional update in terms of efficiency, accuracy, and robustness. Conclusions and future directions are presented in Section 5.4. The research presented in this chapter is published in [65][75].

5.1 Optimality Conditions for Decomposed Problems in Centralized ALC

The centralized ALC cuts off all the original links between the existing sub-problems, creates an artificial master problem and adds new links between each sub-problem and the new master problem, thus enabling the parallel computation of all sub-problems. Since this master coordination problem is a simple unconstrained quadratic optimization problem, its optimal solution can be calculated analytically. In this section, the derivation of the centralized ALC is reviewed first in preparation for applying the KKT conditions to the problem formulations.

5.1.1 Derivation of the Centralized ALC

Consider the general quasi-separable optimization problem shown below (Eq. (5.1)) whose objective and constraint functions are coupled by linking variables \mathbf{y} . Its overall or the AII-In-One (AIO) formulation is

$$\begin{aligned} \min_{\mathbf{y}; \mathbf{x}_1, \dots, \mathbf{x}_M} \quad & \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\ \text{s.t.} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \\ & j = 1, \dots, M \end{aligned} \tag{5.1}$$

where M is the number of potential sub-problems.

The structure of the problem is shown in Figure 5.1 (a), where the potential sub-problems are coupled with each other through the linking variables \mathbf{y} . These can either be the design variables shared by more than one sub-problem or some intermediate variables that are outputs of some sub-problems and required as inputs to other sub-problems.

The first step in the decomposition of this AIO problem involves the introduction of copies of the linking variables, enabling each potential sub-problem to have its own copy of \mathbf{y} . The resulting formulation is shown in Eq. (5.2) and is also depicted in Figure 5.1 (b), in which the original links between sub-problems are centralized to the link based on the consistency constraints $\mathbf{c}_j = \mathbf{y} - \mathbf{y}_j = \mathbf{0}$, $j = 1, \dots, M$.

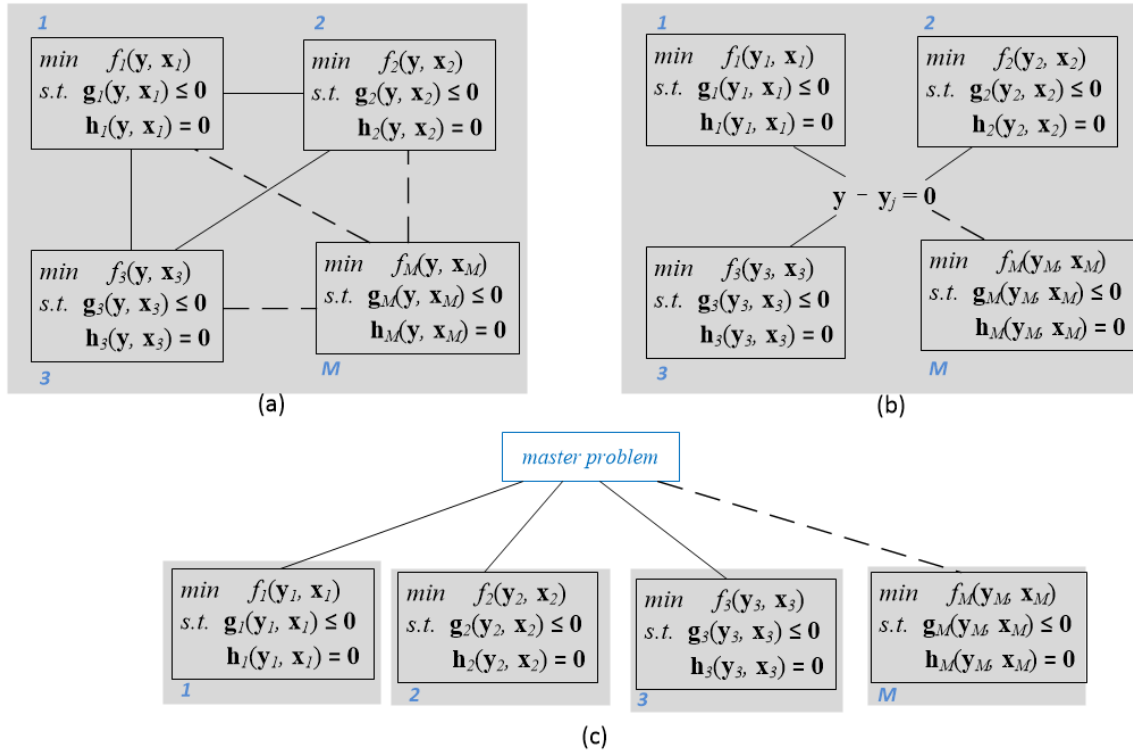


Figure 5.1: Procedure of centralized ALC (problems in grey areas are solved as one problem, dashed lines represent the sub-problem couplings when the number of sub-problem is more than 4)

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j, \mathbf{y}} \sum_{j=1}^M f_j(\mathbf{x}_j, \mathbf{y}_j) \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \\
& \quad \mathbf{c}_j = \mathbf{y} - \mathbf{y}_j = \mathbf{0} \quad j = 1, \dots, M
\end{aligned} \tag{5.2}$$

The consistency constraints are then relaxed to the objectives and result in the following formulation, or the relaxed AIO problem

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j, \mathbf{y}} \sum_{j=1}^M f_j(\mathbf{x}_j, \mathbf{y}_j) + \sum_{j=1}^M \mathbf{v}_j^T (\mathbf{y} - \mathbf{y}_j) + \sum_{j=1}^M \|\mathbf{w}_j \circ (\mathbf{y} - \mathbf{y}_j)\|_2^2 \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad j = 1, \dots, M
\end{aligned} \tag{5.3}$$

where \mathbf{v}_j is the Lagrange multiplier, and \mathbf{w}_j is the weight in the quadratic term.

Finally, according to the Block Coordinate Descent (BCD) method [29], the problem can be solved by iterating between solving the relaxed AIO for a subset of variables, while fixing the other variables at their previous value. The master problem and sub-problems of the centralized ALC generated in this process are shown in Figure 5.1 (c). The master problem is solved first and then its solution \mathbf{y} is passed to all sub-problems as targets. This centralized ALC can be considered as a hierarchical two-level coordination and all sub-problems on the lower level can be solved in parallel. The formulation of the master problem is

$$\min_{\mathbf{y}} \sum_{j=1}^M \mathbf{v}_j^T (\mathbf{y} - \mathbf{y}_j) + \sum_{j=1}^M \|\mathbf{w}_j \circ (\mathbf{y} - \mathbf{y}_j)\|_2^2 \tag{5.4}$$

while the formulation of the sub-problem j is

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \mathbf{v}_j^T (\mathbf{y} - \mathbf{y}_j) + \|\mathbf{w}_j \circ (\mathbf{y} - \mathbf{y}_j)\|_2^2 \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0}
\end{aligned} \tag{5.5}$$

5.1.2 KKT Optimality Conditions for the Centralized ALC

According to the KKT first order necessary conditions [28], and under the assumption that \mathbf{f}_j and \mathbf{g}_j are continuously differentiable, an optimal solution to the AIO problem in Eq. (5.2) must satisfy

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M f_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M (\mathbf{u}_i^{(g)})^T \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i) \right] \\
& + \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M (\mathbf{v}_i^{(h)})^T \mathbf{h}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M \mathbf{v}_i^T (\mathbf{y} - \mathbf{y}_i) \right] = \mathbf{0}
\end{aligned} \tag{5.6}$$

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M f_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M (\mathbf{u}_i^{(g)})^T \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i) \right] \\
& + \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M (\mathbf{v}_i^{(h)})^T \mathbf{h}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M \mathbf{v}_i^T (\mathbf{y} - \mathbf{y}_i) \right] = \mathbf{0}
\end{aligned} \tag{5.7}$$

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{y}} \left[\sum_{i=1}^M f_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}} \left[\sum_{i=1}^M (\mathbf{u}_i^{(g)})^T \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i) \right] \\
& + \frac{\partial}{\partial \mathbf{y}} \left[\sum_{i=1}^M (\mathbf{v}_i^{(h)})^T \mathbf{h}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}} \left[\sum_{i=1}^M \mathbf{v}_i^T (\mathbf{y} - \mathbf{y}_i) \right] = \mathbf{0}
\end{aligned} \tag{5.8}$$

$$\mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \quad (5.9)$$

$$\mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad (5.10)$$

$$\mathbf{c}_j = \mathbf{y} - \mathbf{y}_j = \mathbf{0} \quad (5.11)$$

$$(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad (5.12)$$

$$\mathbf{u}_j^{(g)} \geq \mathbf{0} \quad (5.13)$$

$$j = 1, 2, \dots, M$$

Eqs. (5.6), (5.7), and (5.8) can be simplified to:

$$\frac{\partial}{\partial \mathbf{x}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] = \mathbf{0} \quad (5.14)$$

$$\frac{\partial}{\partial \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] - \mathbf{v}_j = \mathbf{0} \quad (5.15)$$

$$\sum_{i=1}^M \mathbf{v}_i = \mathbf{0} \quad (5.16)$$

$$j = 1, 2, \dots, M$$

Eqs. (5.9), (5.10), and (5.11) are called the primal feasibility of the AIO problem in Eq. (5.2). Eqs. (5.13), (5.14), (5.15), and (5.16) are referred to as the dual feasibility of Eq. (5.2). Eq. (5.12) is called the complementary slackness condition of Eq. (5.2).

The centralized ALC solves the AIO problem by iteratively optimizing its decomposed master problem and the sub-problems. Here we use a superscript to indicate the number of iterations. At iteration $k+1$, the master problem is first optimized using $\mathbf{y}_j^k, j=1, \dots, M$ passed from sub-problems in the last iteration, generating the optimal value for \mathbf{y} at iteration $k+1$: \mathbf{y}^{k+1} . Then all the sub-problems are optimized in parallel using \mathbf{y}^{k+1} and generate $\mathbf{y}_j^{k+1}, j=1, \dots, M$, which are ready to be sent to the master problem in the next iteration. During this process, \mathbf{x}_j^{k+1} and \mathbf{y}_j^{k+1} optimize the sub-problem j (Eq.

(5.5)) with fixed $\mathbf{v}_j^k, \mathbf{w}_j^k$ and \mathbf{y}^{k+1} . Again applying the KKT necessary conditions to the sub-problem j , \mathbf{x}_j^{k+1} and \mathbf{y}_j^{k+1} satisfy

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} \\ + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} = \mathbf{0} \end{aligned} \quad (5.17)$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\ + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} - \mathbf{v}_j^k - 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}^{k+1} - \mathbf{y}_j) \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} = \mathbf{0} \end{aligned} \quad (5.18)$$

$$\mathbf{g}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) \leq \mathbf{0} \quad (5.19)$$

$$\mathbf{h}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) = \mathbf{0} \quad (5.20)$$

$$(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) = \mathbf{0} \quad (5.21)$$

$$\mathbf{u}_j^{(g)} \geq \mathbf{0} \quad (5.22)$$

$$j = 1, 2, \dots, M$$

Note that since the centralized ALC uses the method of multipliers to update the Lagrangian multipliers [20]: $\mathbf{v}_j^k + 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}^{k+1} - \mathbf{y}_j^{k+1}) = \mathbf{v}_j^{k+1}$, Eq. (5.18) reduces to

$$\begin{aligned} \frac{\partial}{\partial \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\ + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} - \mathbf{v}_j^{k+1} = \mathbf{0} \end{aligned} \quad (5.23)$$

$$j = 1, 2, \dots, M$$

As illustrated in Figure 5.2, Eqs. (5.19), (5.20), (5.21), (5.22), (5.17), and (5.23) prove that the optimality conditions in Eqs. (5.9), (5.10), (5.12), (5.13), (5.14), and (5.15), are satisfied automatically at iteration $k+1$ for sub-problem j . Eq. (5.11) is guaranteed by

the convergence criterion when the iteration terminates [20]. The only condition left that needs to be checked is condition (5.16).

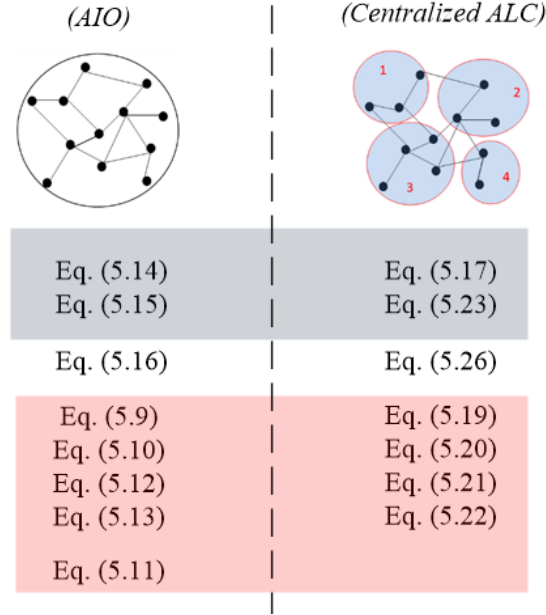


Figure 5.2: Comparison of KKT conditions between the AIO and centralized ALC (the equations with shadow background on the left are guaranteed by the equations with shadow background on the right or convergence criteria)

Similarly to the sub-problems, since \mathbf{y}^{k+1} optimizes the master problem (Eq. (5.4))

with fixed \mathbf{v}_j^k , \mathbf{w}_j^k and \mathbf{y}_j^k , according to the KKT optimality conditions, it must satisfy

$$\frac{\partial}{\partial \mathbf{y}} \left(\sum_{j=1}^M (\mathbf{v}_j^k)^T (\mathbf{y} - \mathbf{y}_j^k) \right) \Big|_{\mathbf{y}=\mathbf{y}^{k+1}} + \frac{\partial}{\partial \mathbf{y}} \left(\sum_{j=1}^M \left\| \mathbf{w}_j^k \circ (\mathbf{y} - \mathbf{y}_j^k) \right\|_2^2 \right) \Big|_{\mathbf{y}=\mathbf{y}^{k+1}} = \mathbf{0} \quad (5.24)$$

The left-hand-side in Eq. (5.24) yields

$$\begin{aligned}
& \sum_{j=1}^M \mathbf{v}_j^k + \sum_{j=1}^M [2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}^{k+1} - \mathbf{y}_j^k)] \\
= & \sum_{j=1}^M \mathbf{v}_j^k + \sum_{j=1}^M [2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}^{k+1} - \mathbf{y}_j^{k+1})] + \sum_{j=1}^M [2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^{k+1} - \mathbf{y}_j^k)] \quad (5.25) \\
= & \sum_{j=1}^M \mathbf{v}_j^{k+1} + \sum_{j=1}^M [2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^{k+1} - \mathbf{y}_j^k)]
\end{aligned}$$

which implies

$$\sum_{j=1}^M \mathbf{v}_j^{k+1} = \sum_{j=1}^M 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^k - \mathbf{y}_j^{k+1}) \quad (5.26)$$

Lets $\mathbf{s}^{k+1} = \sum_{j=1}^M 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^k - \mathbf{y}_j^{k+1})$, then \mathbf{s}^{k+1} can be viewed as the residual for the

dual feasibility condition in Eq. (5.16), and can be further split into $\mathbf{s}^{k+1} = \sum_{j=1}^M \mathbf{s}_j^{k+1}$, where

$\mathbf{s}_j^{k+1} = 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^k - \mathbf{y}_j^{k+1})$ is called the dual residual for sub-problem j . If all

\mathbf{s}_j^{k+1} for $j=1,2,\dots,M$ are equal to 0, then \mathbf{s}^{k+1} is equal to 0 and the optimality condition in

Eq. (5.16) is satisfied.

A solution is feasible to the original AIO problem if the consistency error \mathbf{c} is zero.

Therefore \mathbf{c} is called the primal residual. In summary, in the centralized ALC method, the

primal residual and the dual residual for each sub-problem j at iteration $k+1$ can be

defined as follows:

The primal residual for sub-problem j :

$$\mathbf{r}_j^{k+1} = (\mathbf{y}^{k+1} - \mathbf{y}_j^{k+1}) \quad (5.27)$$

The dual residual for sub-problem j :

$$\mathbf{s}_j^{k+1} = 2\mathbf{w}_j^k \circ \mathbf{w}_j^k \circ (\mathbf{y}_j^k - \mathbf{y}_j^{k+1}) \quad (5.28)$$

5.2 Weight Update Strategy Based on the Primal and Dual Residual

In the centralized ALC, it is necessary to drive both primal and dual residuals to zero to guarantee the primal and dual feasibility of the decomposed problem. However, the definitions of primal residuals \mathbf{r}^{k+1} and dual residual \mathbf{s}^{k+1} suggest a contradiction when considering the elimination of primal and dual residuals. Specifically, a small \mathbf{w} tends to decrease the dual residual, but it reduces the penalty put on the inconsistency thus increasing the primal residual. On the contrary, a large \mathbf{w} can drive the primal residual towards zero quickly, but it also increases the dual residual.

In order to make a tradeoff between the primal and dual residual and keep both of them decreasing during the iterative procedure in the centralized ALC, inspired by ref. [73][74], a new weight update strategy is proposed as follows:

$$\mathbf{w}_j^{k+1} := \begin{cases} \tau^{incr} \mathbf{w}_j^k & \text{if } \|\mathbf{r}_j^k\|_2 > \mu \|\mathbf{s}_j^k\|_2 \\ \mathbf{w}_j^k / \tau^{decr} & \text{if } \|\mathbf{s}_j^k\|_2 > \mu \|\mathbf{r}_j^k\|_2 \\ \mathbf{w}_j^k & \text{otherwise} \end{cases} \quad (5.29)$$

where $\mu > 1$, $\tau^{incr} > 1$ and $\tau^{decr} > 1$ are parameters used to control the rate at which \mathbf{w} increases or decreases. For example, small μ and big τ^{incr} and τ^{decr} means that \mathbf{w} can be easily changed in a speedy fashion. The basic idea of this update is that the weight \mathbf{w} is increased if the primal residual is too large compared to the dual residual and decreased if the dual residual is much larger than the primal residual. Otherwise \mathbf{w} is not changed in the next iteration. Update (5.29) is written in vector form for convenience and each

element w_{ji} of the weight vector \mathbf{w}_j can be updated separately, which is what is used in the following tests.

The dual residual can also be included into the convergence criterion to ensure the dual feasibility condition(5.16) holds at the optimal solution. The traditional convergence check [20], which just considers the primal residual is shown below:

$$\|\mathbf{r}_j^k - \mathbf{r}_j^{k-1}\|_\infty < \varepsilon \quad \text{and} \quad \|\mathbf{r}_j^k\|_\infty < \varepsilon, \quad i = 1, \dots, M \quad (5.30)$$

where ε is the termination tolerance for the optimization process.

In order to guarantee that both the primal and dual residuals are close to zero when the algorithm stops, a new convergence check criterion for the centralized ALC is proposed, which uses the square root of the sum of the squares of a pair of primal and dual residuals

$$\sqrt{(r_{ji}^k)^2 + (s_{ji}^k)^2} < \varepsilon, \quad \text{for all } j = 1, \dots, M \quad (5.31)$$

For the optimization to stop, it is needed that every pair of primal and dual residuals satisfies this criterion.

5.3 Numerical Tests

Three problems are considered to test the efficacy of the proposed update strategy and the convergence check criterion: one mathematical problem and two engineering problems. Four indicators are used to calibrate the performances: the biggest primal and dual residuals, the biggest error in the design variables, and the objective error.

The biggest primal and dual residuals are the biggest elements in the vectors \mathbf{r}_i and \mathbf{s}_i , respectively, following Eq. (5.27) (5.28). The error in the design variable x_i is calculated based on its optimal value in the All-in-One solution:

$$e_x = \left| \frac{x_i^{ALC} - x_i^{AiO}}{x_i^{AiO}} \right| \quad (5.32)$$

The objective error is calculated similarly:

$$e_f = \left| \frac{f^{ALC} - f^{AiO}}{f^{AiO}} \right| \quad (5.33)$$

where f denotes the optimization objective. $f = \sum_{j=1}^M f_j$ in Eq.(5.1).

Since the initial value of the weight \mathbf{w} greatly affects the performance of the centralized ALC, in order to compare the optimization results fairly and comprehensively, these problems are solved and compared with respect to their average behavior and “best case” behavior. The first test starts with the initial \mathbf{w} determined by the proposed initial weight setting method in [20]. The second one starts with a fine-tuned initial \mathbf{w} which has been improved though trial and error. The proposed initial weight setting method in [20] is shown below

$$w = \frac{\alpha |\hat{f}|}{\sqrt{\sum_{j=1}^M \|\hat{\mathbf{c}}_j\|_2^2}} \quad (5.34)$$

where \hat{f} and $\hat{\mathbf{c}}_j$ are estimates of the objective value and consistency error before carrying out the optimization, and α is a fraction parameter chosen by the user within the range $1e-3 < \alpha < 1$.

In this chapter, the centralized ALC-ADMOM method is used due to its parallel computational ability and efficiency. Both the traditional update in Eq. (2.18) and the proposed update in Eq. (5.29) are applied to the centralized ALC-ADMOM and their performances are compared in terms of efficiency, accuracy and robustness. It should be noted that for sub-problems with a convex local objective, [20] suggests \mathbf{w} to be fixed to a small value instead of being updated at every iteration. However, in the follow-up papers of ALC [24][59][60], a small step size β ($1 < \beta < 1.2$) is chosen for ALC-ADMOM to reach a better performance when solving test examples. In this research, to make sure ALC-ADMOM works at top capacity, the same rule is adopted by updating \mathbf{w} slowly in the comparison tests, which is different from the implementation of ALC-ADMOM in references [54][61], where \mathbf{w} is fixed by setting β to 1. The solver “*fmincon*” [68] in MATLAB 2012a, which uses the interior point method as the default algorithm but may switch to other algorithms during execution, is chosen to solve all sub-problems.

5.3.1 Geometric Optimization Problem

The first test problem is a simple geometric optimization problem which has been used in many papers, e.g. [54][55]. Here this problem is partitioned into two sub-problems and uses an artificial master problem to coordinate them as shown in Figure 5.3.

$$\begin{aligned}
& \min_{z_1, \dots, z_7} f = z_1^2 + z_2^2 \\
& \text{subject to } g_1 = (z_3^{-2} + z_4^2)z_5^{-2} - 1 \leq 0, \\
& g_2 = (z_5^2 + z_6^{-2})z_7^{-2} - 1 \leq 0, \\
& h_1 = (z_3^2 + z_4^{-2} + z_5^2)z_1^{-2} - 1 = 0, \\
& h_2 = (z_5^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0,
\end{aligned} \tag{5.35}$$

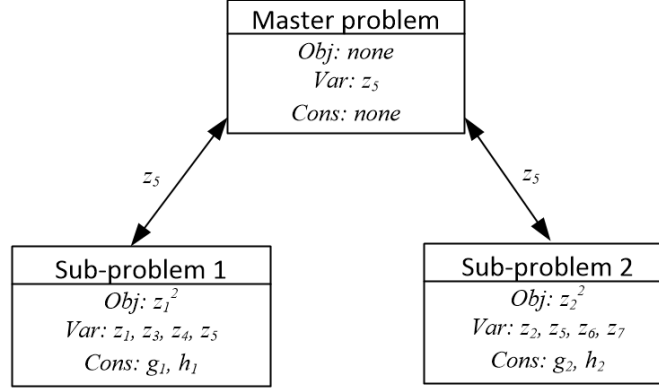


Figure 5.3: The structure to solve the geometric programming problem through decomposition

The parameter settings for the traditional weight update in Eq. (2.18) are $\beta=1.1, \gamma=0.9$, based on the preferred values in reference [20]. For the proposed strategy, $\tau^{incr} = \tau^{decr} < 2$ is suggested. For simple problems, τ^{incr} and τ^{decr} can be assigned a relatively big value (1.5 – 2.0) to accelerate convergence by changing \mathbf{w} quickly, while for complex problems small τ^{incr} and τ^{decr} (1.0 – 1.1) are more preferable since the optimization of complex problems needs much more iterations and small τ^{incr} and τ^{decr} can make this process smooth. For test problem 1, the following are set: $\mu=1.5$ and $\tau^{incr} = \tau^{decr} = 1.5$. Since this is a fairly easy problem, the convergence check (5.30) is adopted with a termination tolerance $\varepsilon = 10^{-3}$. The starting values of all design variables

are all one and the initial multiplier ν is set to zero. For the first set of tests, the initial weight is decided by Eq. (5.34) with $\hat{f} = 1$, $\alpha = 0.1$ and $\mathbf{w}^0 = 0.001$. The optimization process of centralized ALC using the traditional update and the proposed update are shown in Figure 5.4 and Figure 5.5 respectively. The curves of the biggest dual residual of these two updates are compared in Figure 5.6.

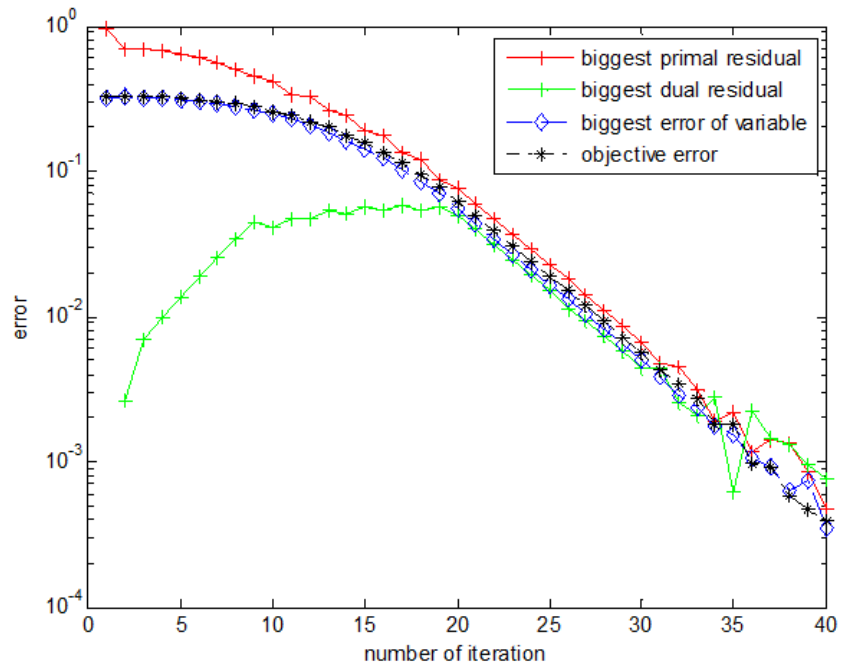


Figure 5.4: Optimization process of the traditional update on the geometric programming problem – initial weight set by Eq. (5.34)

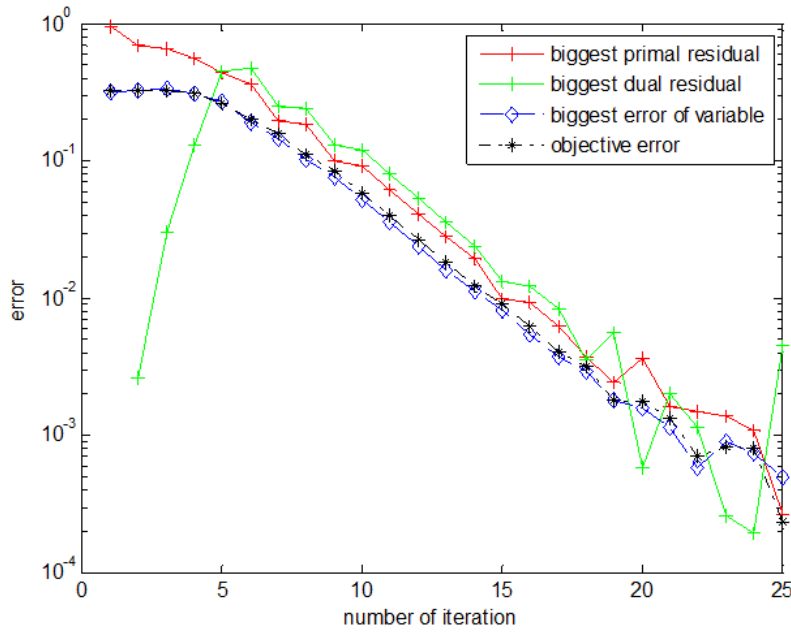


Figure 5.5: Optimization process of the proposed update on the geometric programming problem – initial weight set by Eq. (5.34)

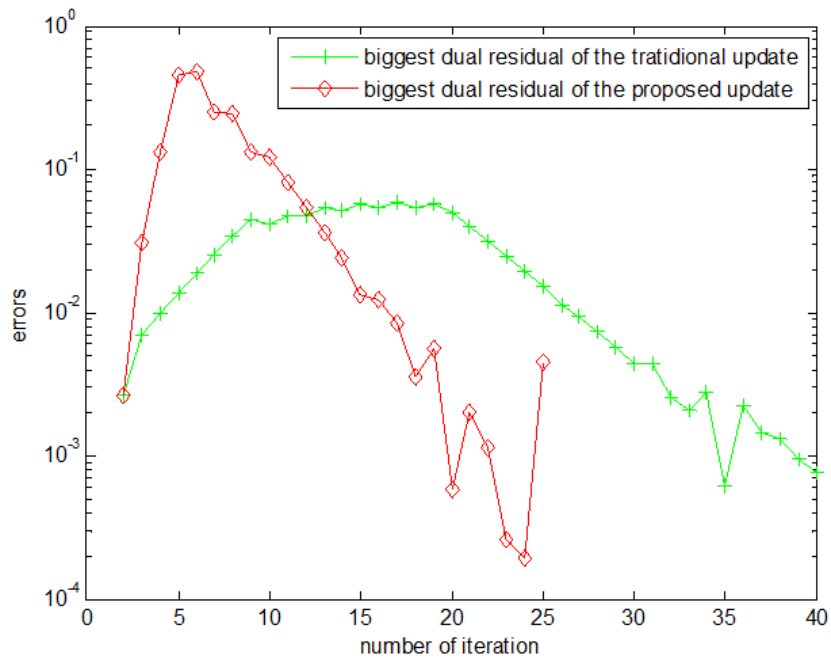


Figure 5.6: Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the geometric programming problem – initial weight set by Eq. (5.34)

It can be seen that the proposed method decreases the number of iterations from 40 to 25 for this problem. Due to the simplicity of the problem, both methods reach a small objective error and a small dual residual error. For the proposed method, the magnitude of the biggest dual residual is the same as that of the primal residual in most iterations, which implies the proposed update is working on adjusting both the primal and dual residuals.

For the second test, the fine-tuned initial weights are used which are 2.5 for the traditional weight update and 5 for the proposed update. The optimization process of the traditional update and the proposed update are shown in Figure 5.7 and Figure 5.8 respectively. These figures show that the traditional and proposed methods reach a similar performance through the tuned initial weight, which is much better than the results depicted in Figure 5.4 and Figure 5.5. The curves of the biggest dual residual of these two updates are compared in Figure 5.9. It can be seen that the final dual residual in the traditional update is bigger than that in the proposed update, although the proposed update starts with a bigger initial \mathbf{w} that contributes more to the dual residual.

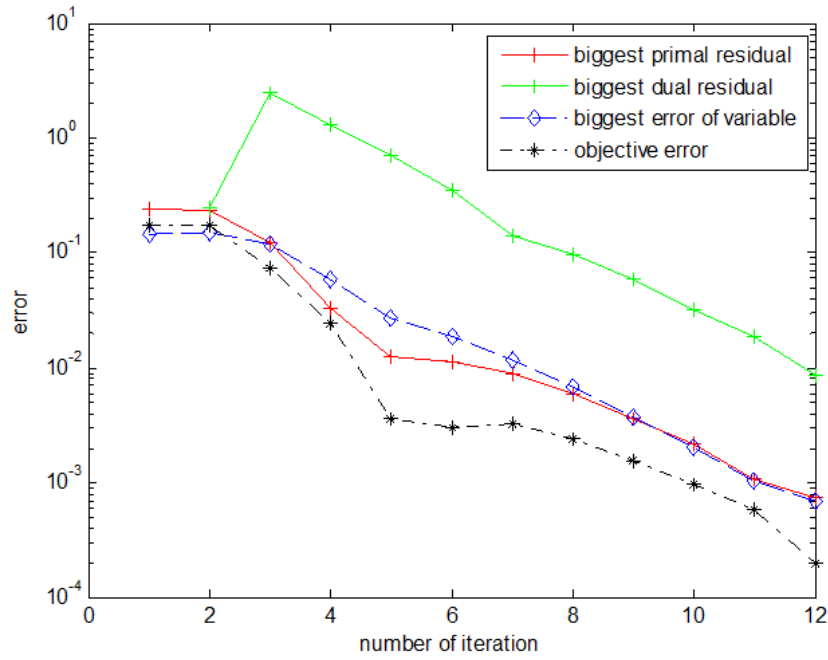


Figure 5.7: Optimization process of the traditional update on the geometric programming problem – using fine-tuned initial weight

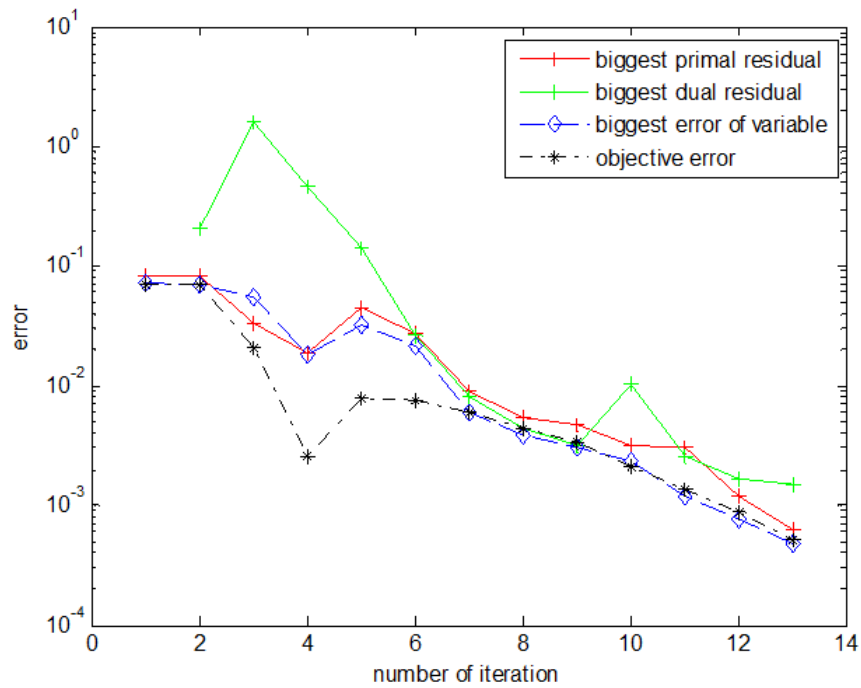


Figure 5.8: Optimization process of the proposed update on the geometric programming problem – using fine-tuned initial weight

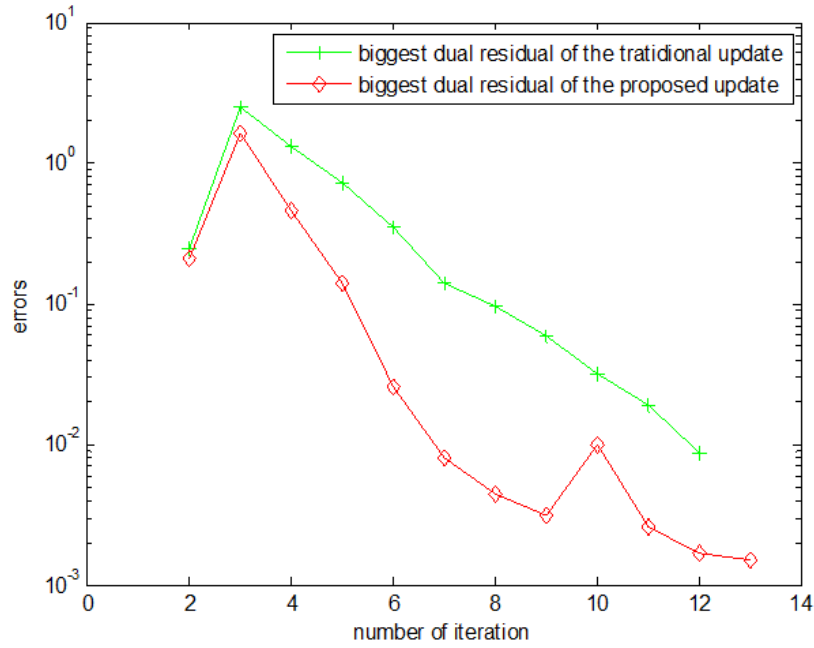


Figure 5.9: Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the geometric programming problem – using fine-tuned initial weight

5.3.2 Golinski’s Speed Reducer Problem

The Golinski’s speed reducer problem has been widely used in the literature [22] [14][54][55][49][70][71][31][76][77]. The objective of this problem is to minimize the volume of a reducer, subject to stress, deflection, and geometric constraints. The design variables are the dimensions of gears (x_1, x_2, x_3) and shafts (x_4, x_5, x_6, x_7). The problem partition in this test is the same as the partition in [20] and has two levels and four sub-problems, which is shown in Figure 5.10. One sub-problem is located at the upper level as a master problem and three sub-problems are located at the lower level. All these four sub-problems are linked through the design variables (x_1, x_2, x_3).

$$\begin{aligned}
& \min_{z=[x_1, \dots, x_7]^T} F = \sum_{j=1}^7 F_j \\
& \text{s.t.} \quad g_1 = \frac{1}{110x_6^3} \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69 \times 10^7} - 1 \\
& \quad \quad g_2 = \frac{1}{85x_7^3} \sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 1.575 \times 10^8} - 1 \\
& \quad \quad g_3 = \frac{1.5x_6 + 1.9}{x_4} - 1, \quad g_4 = \frac{1.1x_7 + 1.9}{x_5} - 1, \\
& \quad \quad g_5 = \frac{27}{x_1x_2^2x_3} - 1, \quad g_6 = \frac{397.5}{x_1x_2^2x_3} - 1, \\
& \quad \quad g_7 = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1, \quad g_8 = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1, \\
& \quad \quad g_9 = \frac{x_2x_3}{40} - 1, \quad g_{10} = \frac{5x_2}{x_1} - 1, \\
& \quad \quad g_{11} = \frac{x_1}{12x_2} - 1. \\
& \quad \quad x_1 \in [2.6, 3.6], x_2 \in [0.7, 0.8], x_3 \in [17, 28], \\
& \quad \quad x_4 \in [7.3, 8.3], x_5 \in [7.3, 8.3], x_6 \in [2.9, 3.9], \\
& \quad \quad x_7 \in [5.0, 5.5]
\end{aligned}$$

where $F_1 = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9335x_3 - 43.0934)$
 $F_2 = -1.5079x_1x_6^2, \quad F_3 = -1.5079x_1x_7^2,$
 $F_4 = 7.477x_6^3, \quad F_5 = 7.477x_7^3,$
 $F_6 = 0.7854x_4x_6^2, \quad F_7 = 0.7854x_5x_7^2.$ (5.36)

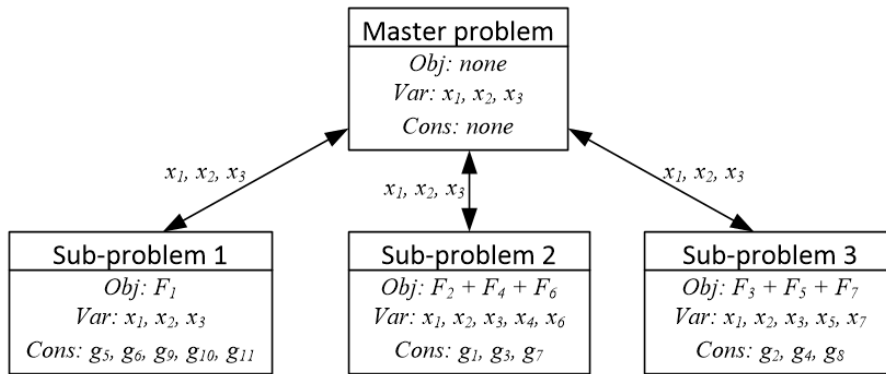


Figure 5.10: The structure to solve the Golinski's problem through decomposition

The parameter settings for the traditional weight update in Eq. (2.18) are $\beta=1.1, \gamma=0.9$. For the proposed strategy, $\mu=2$ and $\tau^{incr} = \tau^{decr} = 1.5$. The convergence check (5.30) is adopted with the termination tolerance $\varepsilon = 10^{-3}$. Since previous research [20] has shown that the initial values of design variables have small effects on optimization results, here all the initial design variables are set to one and the initial multipliers \mathbf{v} are set to zero. For the first set of tests, the initial weight is decided by Eq. (5.34) with $\hat{f} = 10^3$, $\alpha = 0.1$ and $\mathbf{w}^0 = 0.001$. The optimization process of centralized ALC using the traditional update and the proposed update are shown in Figure 5.11 and Figure 5.12 respectively. The curves of the biggest dual residual of these two updates are compared in Figure 5.13.

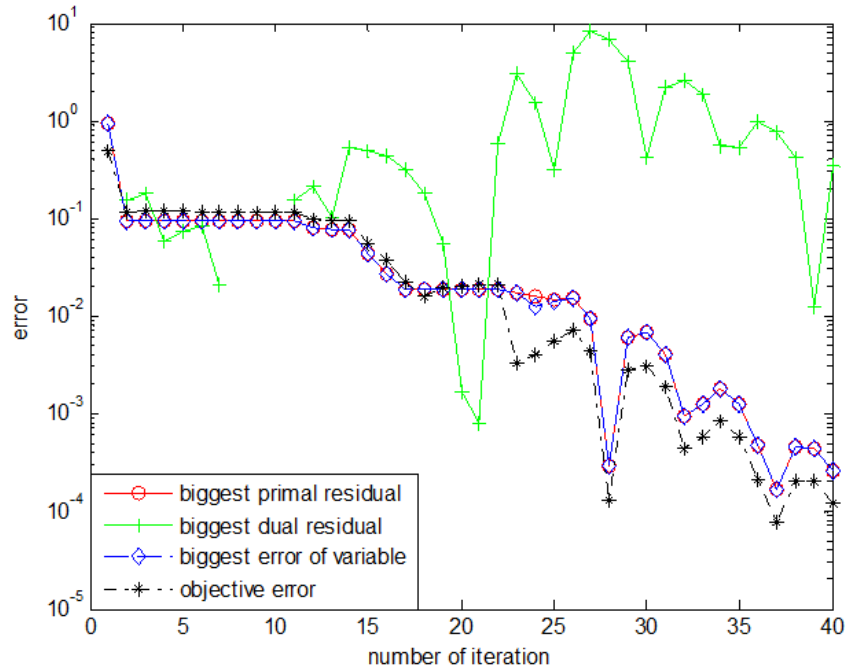


Figure 5.11: Optimization process of the traditional weight update on the speed reducer problem – initial weight set by Eq. (5.34)

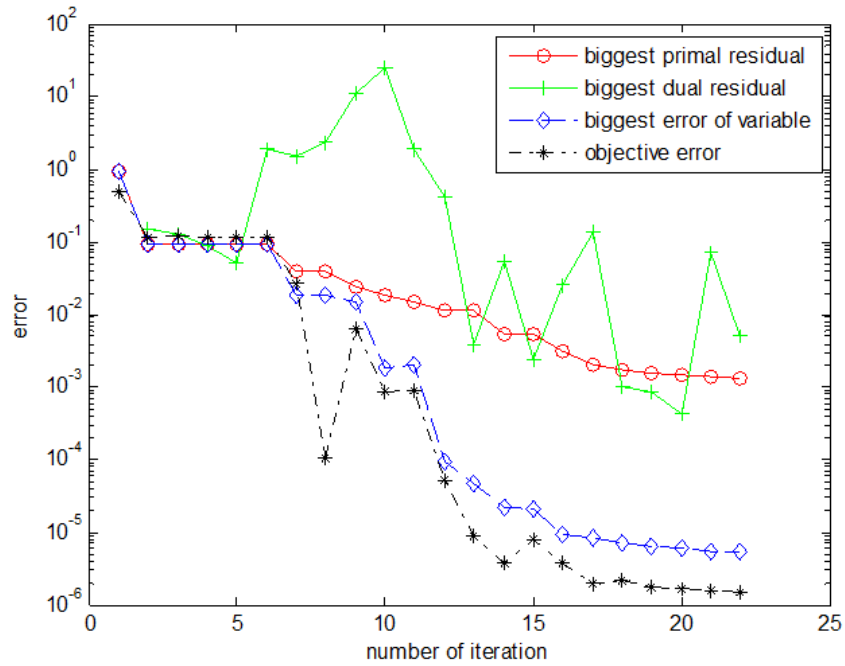


Figure 5.12: Optimization process of the proposed weight update on the speed reducer problem – initial weight set by Eq. (5.34)

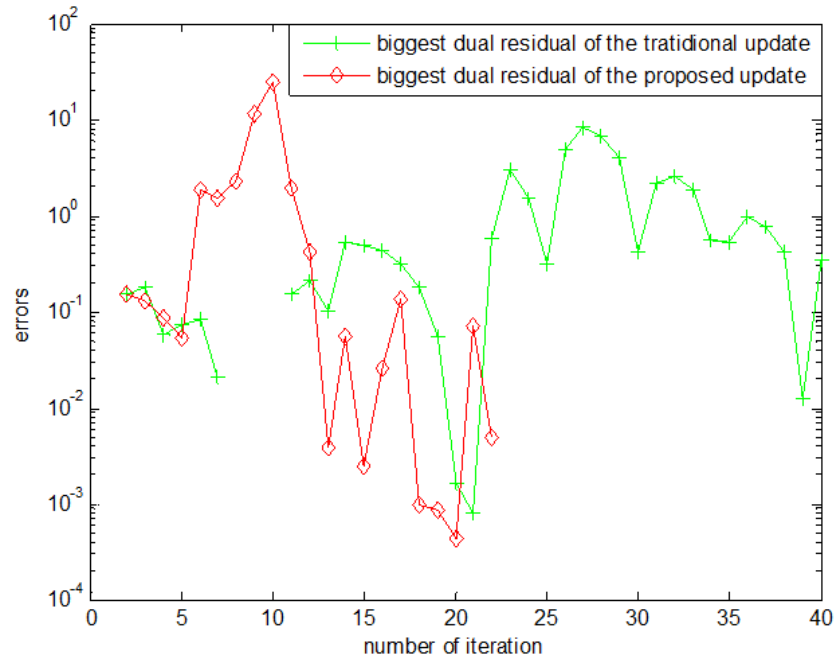


Figure 5.13: Comparison of the curves of the biggest dual residual for the traditional update and proposed update on the speed reducer problem – initial weight set by Eq.(5.34)

It can be seen that, similarly to the results in the first test problem, the proposed strategy decreases the number of iterations from 40 to 27, which means it needs 32.5% less sub-problem redesigns to optimize this problem. Meantime, the errors of variables (around 10^{-5}) and objective (around 10^{-6}) in the proposed method are smaller than those in the traditional update method (around 10^{-4}). Additionally, the dual residual reaches a relatively large value 0.34 in the traditional method, while in the proposed method, it shows a trend of decreasing and being about 0.005 when the program converges.

For the second test, the fine-tuned initial weights are used which are 8 for the traditional weight update and 20 for the proposed update. The optimization process of the traditional penalty parameter update and the proposed update are shown in Figure 5.14 and Figure 5.15 respectively. The curves of the biggest dual residual of these two updates are compared in Figure 5.16.

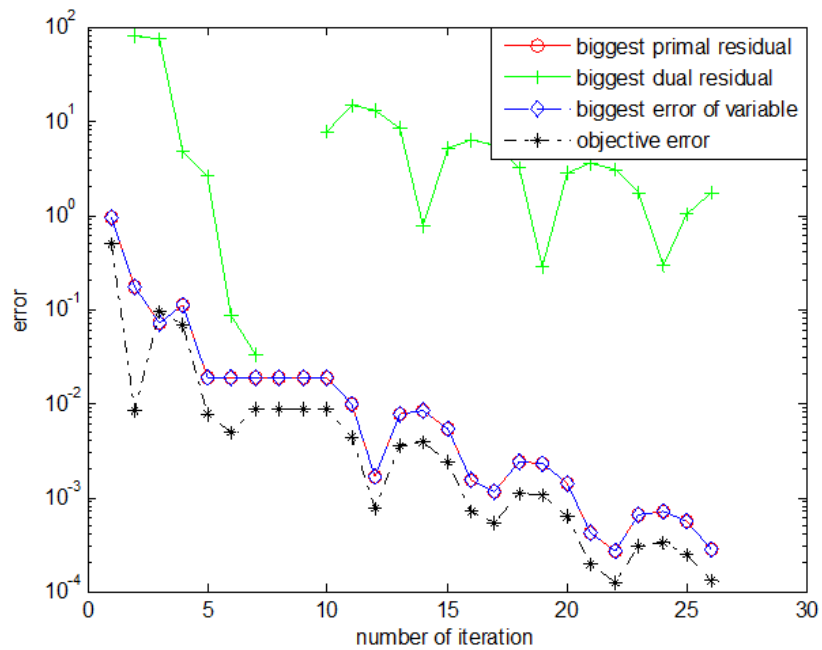


Figure 5.14: Optimization process of the traditional weight update on the speed reducer problem – using fine-tuned initial weight

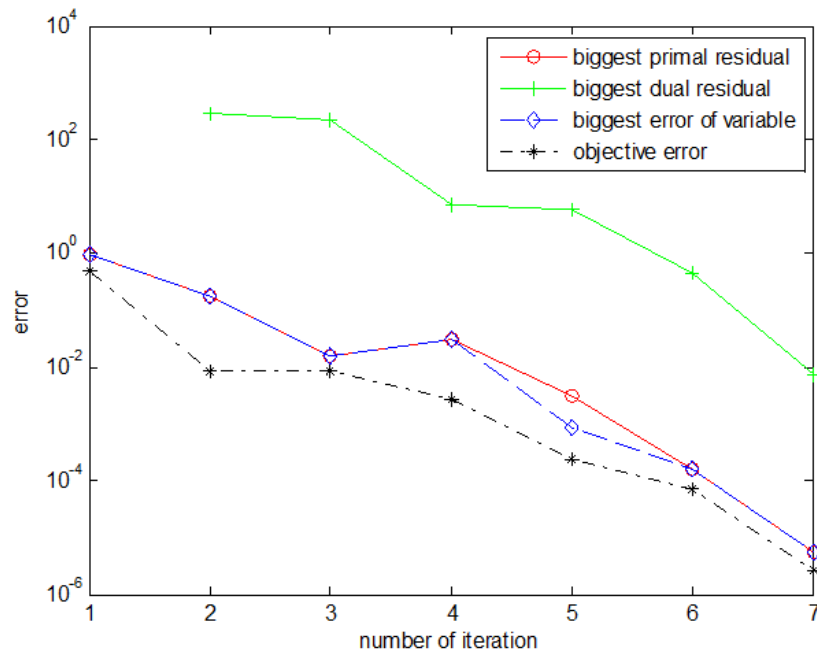


Figure 5.15: Optimization process of the proposed weight update on the speed reducer problem – using fine-tuned initial weight

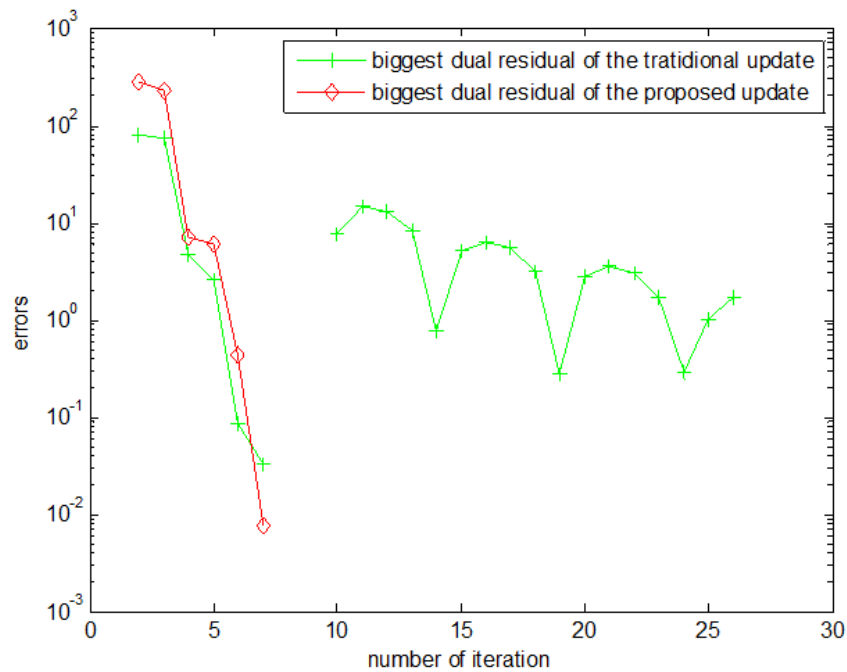


Figure 5.16: Comparison of the curves of the biggest dual residual for the traditional update and the proposed update on the speed reducer problem – using fine-tuned initial weight

The performance of both the traditional and proposed methods with tuned initial weight improves a lot compared to what is shown in Figure 5.11 and Figure 5.12. For the traditional method, the number of iterations is decreased to 26 but the final dual residual is increased to 1.72 due to a larger initial weight. The performance of the proposed method is quite impressive, reaching the optimal solution in 7 iterations while the errors of variables and objective remain under 10^{-5} . The biggest dual residual is also under 10^{-2} . It is clear that the proposed update manages to drive both the primal and dual residuals to zero and thus is able to satisfy the dual optimality condition (5.16) when the program converges, which is not guaranteed in the traditional update. As a result, the proposed method is shown to be more efficient (in terms of the number of iterations) and more accurate (in terms of the errors of variables and objective) than the traditional update in both scenarios, with Eq. (5.34) determining a guessed initial weight and a fine-tuned initial weight.

5.3.3 Micro-accelerometer Design Problem

The benchmark problem - micro-accelerometer design problem in Chapter 3 is used again. Here case 3 [60] and the partition into four sub-problems proposed in [60] are adopted. Among the proposed four cases in [60], case 4 involves mixed-integer optimization which is not the focus of this research and case 3 is the most complex one. The optimal results of this problem (case 3) through the All-in-One formulation without decomposition are shown in Table 5.1. The best solution $f = 0.0807 \text{ mm}^2$ is used to calculate the objective error in the following part. The structure to solve the problem through the centralized ALC is shown in Figure 5.17.

Table 5.1: Optimization results of the micro-accelerometer problem (case 3) without decomposition (AIO)[60]

	Min	Mean	Max
Objective (mm ²) (15% of 100 runs have converged)	0.0807	0.0810	0.0817

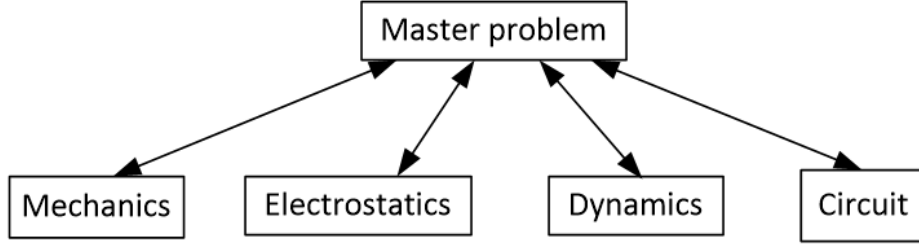


Figure 5.17: The Structure to solve the micro-accelerometer problem through decomposition

The decomposed problem is solved by the centralized ALC under three schemes: scheme 1 - the traditional weight update Eq.(2.18) + traditional convergence check Eq. (5.30); scheme 2 - the proposed weight update Eq. (5.29)+ traditional convergence check Eq. (5.30); scheme 3 - the proposed weight update Eq. (5.29)+ the proposed convergence check Eq. (5.31). Also, as in the previous experiments, all these schemes are tested twice with a calculated initial weight and a fine-tuned initial weight.

In the first set of tests, the initial weight is decided by Eq. (5.34) with $\hat{f} = 0.5$ mm², $\alpha = 0.1$, and $\mathbf{w}^0 = 0.001$. The parameter settings for the traditional weight update in Eq.(2.18) are $\beta = 1.1, \gamma = 0.9$, and those for the proposed strategy are $\mu = 2, \tau^{incr} = \tau^{decr} = 1.02$. The termination tolerance for both methods is $\epsilon = 1e-3$. The initial design variables are set randomly between 50% and 150% of the baseline design. The initial Lagrangian multipliers are all set to zero. Similar to [60], to ensure that each

sub-problem transfers a correct solution to the other sub-problems, the sub-problem optimization is restarted when its optimization does not converge. For each scheme, 10 runs starting from different random points are performed and the average of the number of iterations and objectives are summarized in Table 5.2.

Table 5.2: Results of the micro-accelerometer problem with a calculated initial weight by Eq. (5.34)

	Scheme 1: traditional update + traditional convergence check		Scheme 2: proposed update + traditional convergence check		Scheme 3: proposed update + proposed convergence criterion)	
	Objective(mm ²) /error	# of iters.	Objective(mm ²) /error	# of iters.	Objective (mm ²) /error	# of iters.
Min	0.0969 /20.1%	101	0.0811 /0.5%	187	0.0809 /0.2%	246
Mean	0.0994 /23.2%	116	0.0816 /1.1%	252	0.0812 /0.6%	370
Max	0.1016 /25.9%	143	0.0821 /1.7%	308	0.0814 /0.9%	494

From Table 5.2 it can be seen that the proposed weight update greatly improves the solution accuracy. The mean value of the optimal objective error is decreased from 23.2% for the traditional ALC to 1.1% for ALC using the proposed update, and it is decreased even further to 0.6% by the combination of the proposed update and the proposed convergence check. The expense for this improvement is the increase of the mean number of iterations from 49 to 252 and 370 respectively.

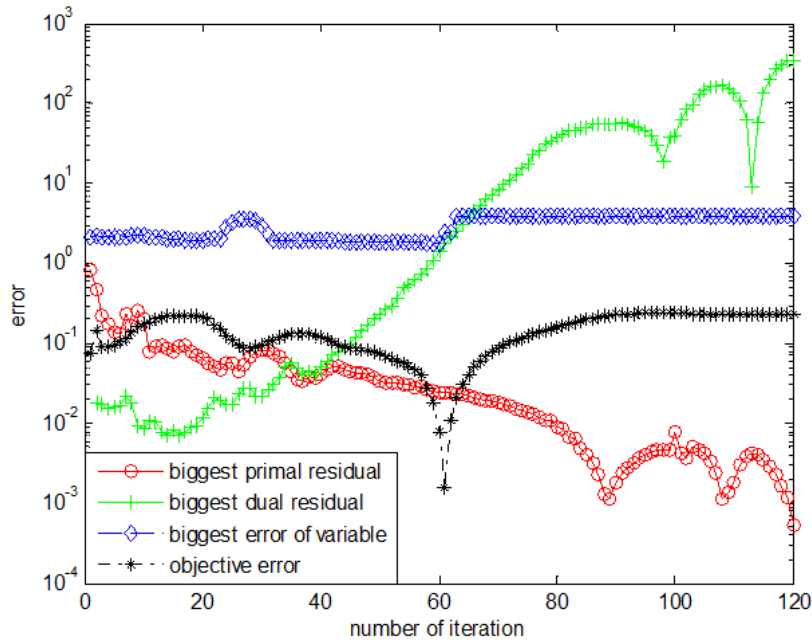


Figure 5.18: Optimization process of the centralized ALC – scheme 1 on the micro-accelerometer problem – initial weight set by Eq. (5.34)

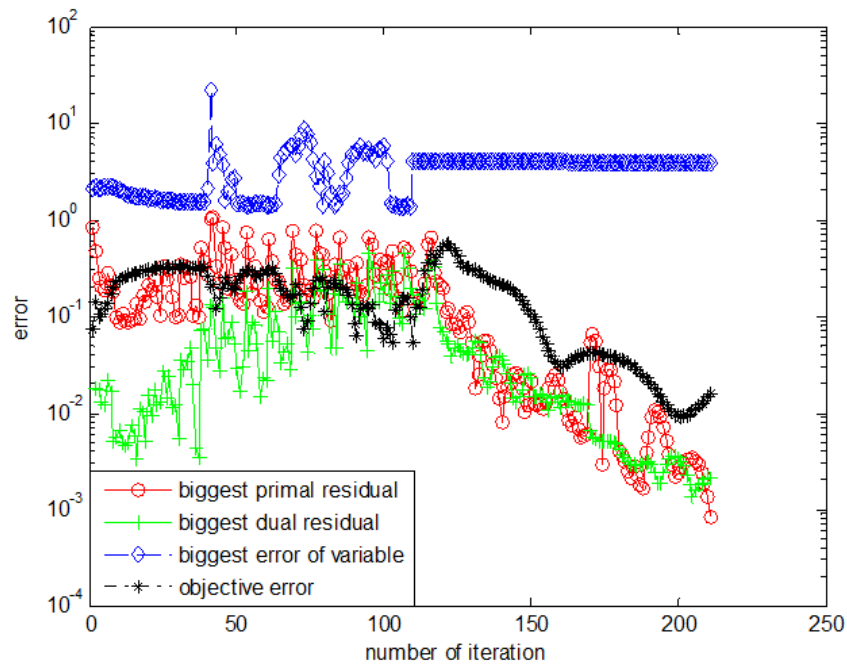


Figure 5.19: Optimization process of the centralized ALC – scheme 2 on the micro-accelerometer problem – initial weight set by Eq. (5.34)

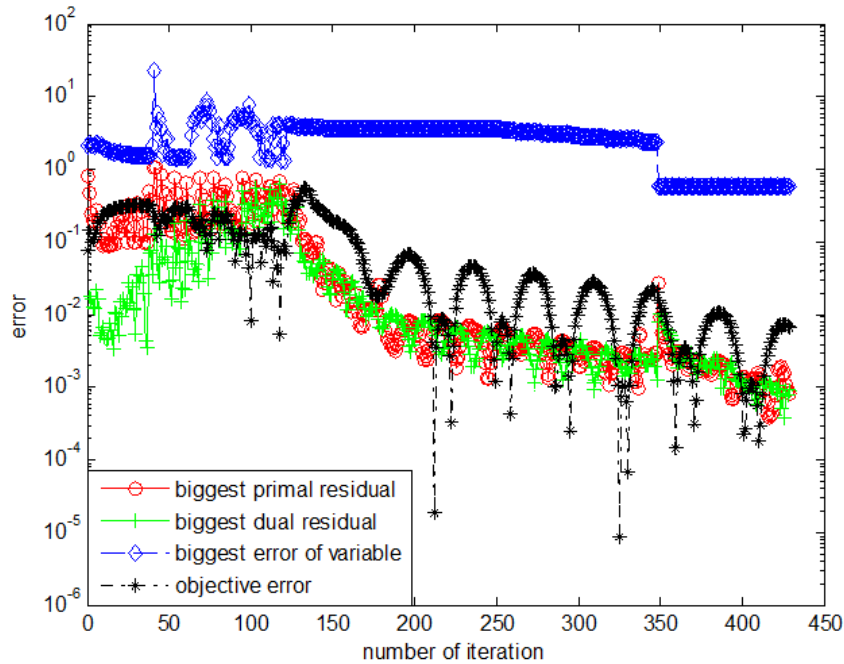


Figure 5.20: Optimization process of the centralized ALC – scheme 3 on the micro-accelerometer problem – initial weight set by Eq. (5.34)

Figure 5.18 shows that the dual residual no longer decreases after the first 20 iterations. Figure 5.19 shows that for the proposed weight update, both the primal residual and dual residual decrease after the iteration process has gone through some fluctuations. The dual residual at the optimal solution is still much larger than the primal residual since only the primal residual is considered in the traditional convergence check criterion. Figure 5.20 shows that using the proposed convergence check criterion, the dual residual is also under 10^{-3} when the algorithm converges, which results in a better solution.

An interesting thing can be observed if the results in the geometric programming and speed reducer problems are compared with those in this section: the proposed strategy decreases the number of iterations needed for the first two test problems while it

increases that number for the micro-accelerometer problem. The reason might be that it is not fair to compare the number of iterations when the objective errors are not at the same order of magnitude. Specifically, the first two test problems are fairly easy and both the traditional and proposed methods perform well and can easily reach a solution that is sufficiently accurate (with the objective error under 10^{-3}). In this case, the objective errors of both updates are at the same order of magnitude and the advantage of the proposed update is arriving at the optimal solution with less iterations. However, the micro-accelerometer problem is a complex engineering problem involving four disciplines and the Finite Element Analysis (FEA) and therefore is difficult for the traditional ALC which does not consider the dual optimality conditions. For this kind of complex problem, the proposed method is able to converge to an accurate solution (objective errors around 1%) while the traditional method stalls at a point that does not satisfy the KKT conditions (objective errors larger than 20%). Therefore, since the new method succeeds where the old method fails, the difference in the number of iterations is irrelevant. The primal and dual residual values are indicative of satisfaction of the KKT conditions of the All-in-One problem and the proposed update enables the centralized ALC to reach these conditions by driving both residuals to zero. The relatively large variable errors reported may be indicative of nonconvexity in the problem formulation itself. The above analysis also applies to the tests with a tuned initial weight that follows.

In the second set of tests, all the settings in the first test are kept the same, except the fine-tuned initial weight is used which is 0.3 for the traditional weight update and 0.5

for the proposed update. The optimization results of the three schemes using the fine-tuned initial weights are summarized in Table 5.3.

Table 5.3: Results of the micro-accelerometer problem with a fine-tuned initial weight

	Scheme 1: traditional update + traditional convergence check		Scheme 2: proposed update + traditional convergence check		Scheme 3: proposed update + proposed convergence criterion)	
	Objective(mm ²) /error	# of iters.	Objective(mm ²) /error	# of iters.	Objective (mm ²) /error	# of iters.
Min	0.0893 /10.7%	40	0.0810 /0.4%	168	0.0807 /0.0%	267
Mean	0.0901 /11.6%	49	0.0815 /1.0%	200	0.0811 /0.5%	322
Max	0.0910 /12.8%	76	0.0817 /1.2%	223	0.0813 /0.7%	447

Using the tuned initial weight, the mean objective error of traditional ALC is decreased from 23.2% to 11.6% while the mean number of iterations is decreased from 116 to 49. Although the accuracy improves a lot, the objective error 11.6% is still relatively large compared with those of scheme 2 and scheme 3, which are 1.0% and 0.5% respectively. Scheme 3 requires more iterations than scheme 2 as it puts a stricter convergence criterion on the dual residual, which results in a further improved solution since the dual optimality condition (5.16) is fully met. Compared to the AIO results in Table 5.1, scheme 3 in Table 5.3 reaches the same minimal objective of 0.0807 mm² which is the reference value and their mean objective values are almost the same. However, only 15% of all tests converge in the AIO test while here all tests of scheme 3 successfully arrive at an accurate solution. This indicates that for some complex problems,

the decomposition-based optimization methods might show a better convergence property than the AIO method.

For scheme 1 with the traditional ALC (Figure 5.21), the objective stops decreasing after a few iterations and the reason might be that the nondecreasing dual residual leads the optimization to an inaccurate solution to which the process starts to converge. Through introducing the dual residual into the update strategy, scheme 2 (Figure 5.22) effectively drives both the primal and dual residual toward zero and the optimization starts fluctuating after it has reached a certain level of accuracy (with an objective under 1%, around 50 iterations) until the primal residual is under 10^{-3} . Scheme 3 (Figure 5.23) has more fluctuations and requires more iterations, but a decreasing trend of the objective error can be observed during this fluctuation process.

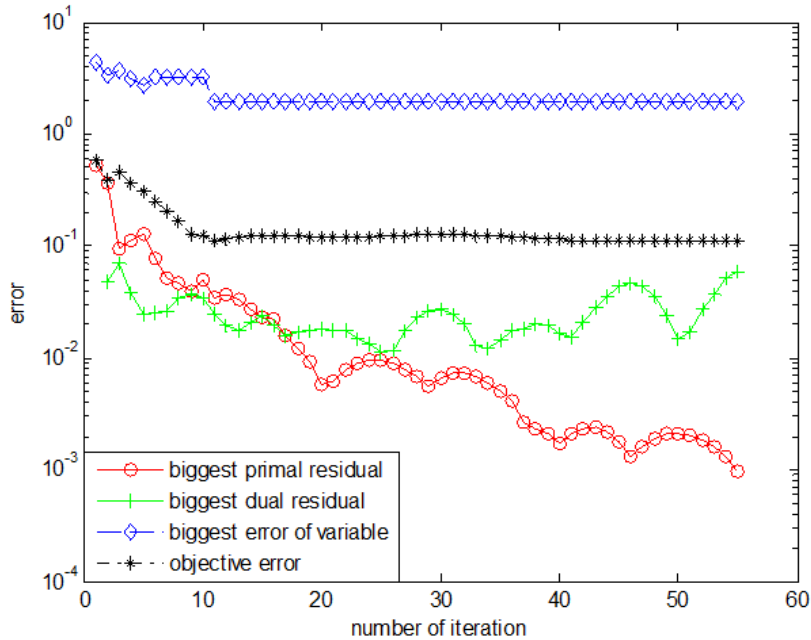


Figure 5.21: Optimization process of the centralized ALC – scheme 1 on the micro-accelerometer problem – fine-tuned initial weight

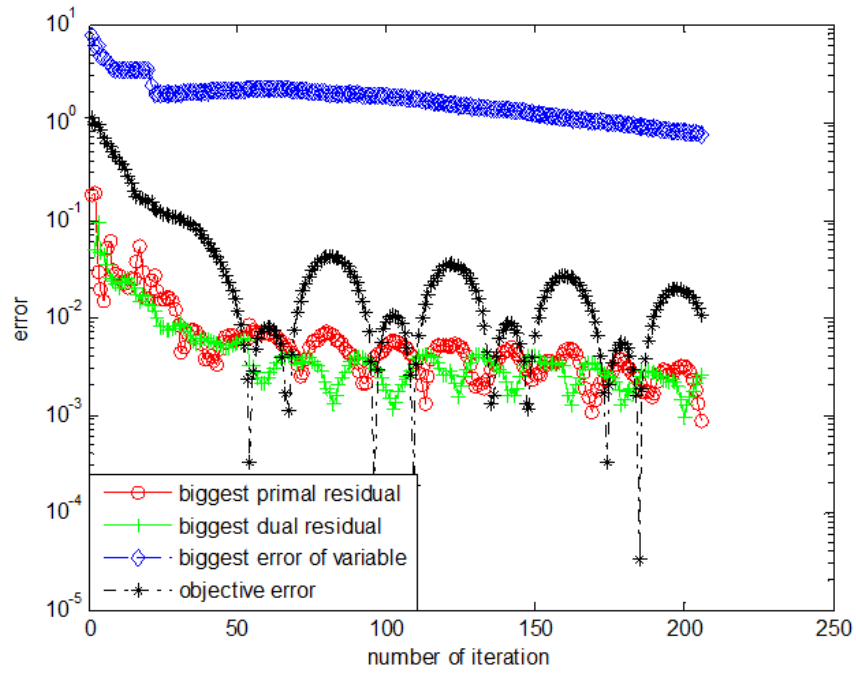


Figure 5.22: Optimization process of the centralized ALC – scheme 2 on the micro-accelerometer problem – fine-tuned initial weight

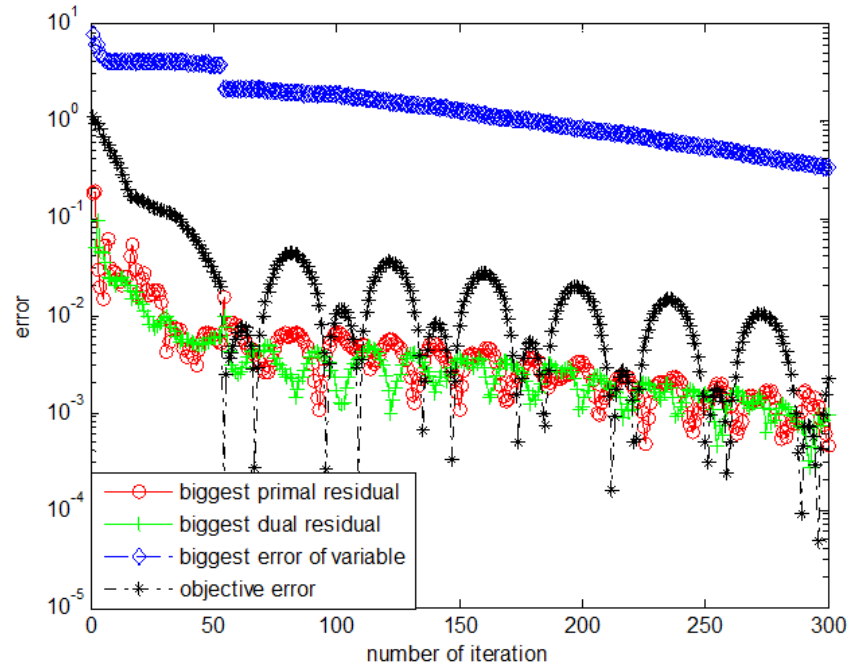


Figure 5.23: Optimization process of the centralized ALC – scheme 3 on the micro-accelerometer problem – fine-tuned initial weight

The performance improvements of the proposed update, that can be observed when comparing Table 5.3 and Table 5.2, are not significant in contrast to the noteworthy improvements over the traditional method. This leads to a question: Is the performance of the proposed update affected greatly by the initial weight as in the traditional update, or is the proposed update more robust than the traditional update? Attempting to answer this question is the topic of the next section.

5.3.4 Robustness of the Proposed Strategy

The initial value of the penalty weight is always a critical parameter for the traditional ALC method [20]. An excessive initial weight often causes the optimization process to converge early to irrelevant solutions while a small initial weight often significantly slows down the convergence rate of optimization. Tosserams [20] proposed to heuristically set the initial weight as shown in Eq. (5.34). But this method is not a fail-safe approach and some trial-and-error runs are often required in practice. As shown in the previous three tests, the performance of the traditional update can be changed greatly by changing the initial weight set by Eq. (5.34).

Since the strategy proposed in this research can either increase or decrease the weight w , it has more capabilities to adjust the weight during the optimization process than the traditional weight update strategy. This may make the performance of ALC less sensitive to the initial weight and bring robustness to it. To verify this conjecture, another set of tests is conducted on the micro-accelerometer problem. All settings are kept the same except for the following three initial weight values: 1, 1e-1, 1e-2. All three cases are performed ten times starting with random initial points between 50% and 150% of the

baseline design. The objective value and the number of iterations reported in Table 5.4 are the mean values of these 10 runs.

Table 5.4: Optimization results of the micro-accelerometer problem with different initial weights

Initial penalty weight	Traditional update		Proposed update	
	Objective(mm ²) /error	# of iters.	Objective(mm ²) /error	# of iters.
1	N/A [*]	N/A	0.0814/0.9%	224 ^{**}
1e-1	N/A	N/A	0.0816/1.1%	293
1e-2	N/A	N/A	0.0809/2.2%	496

* All the tests of the traditional update do not converge or converge to irrelevant solutions;

** 90% of the tests converge to the accurate solutions;

It is clear that none of the above three cases can generate an optimal solution for the traditional update. The appropriate initial weight in Section 5.3.3 is 0.3. The high sensitivity of the traditional update to the initial weight prohibits the convergence of the algorithm starting with the w that is not close enough to 0.3.

Based on the two most right columns in Table 5.4 we note that ALC with the proposed strategy is able to work with any initial weight selected between 1e-2 and 1. Thus the proposed weight update method makes the centralized ALC method robust to the initial weight, which is of great significance to engineers.

5.4 Summary

In this chapter, to solve the issues of setting and updating the penalty weight in the centralized ALC, the KKT necessary optimality conditions are applied to the relaxed AIO problem and the decomposed problems. The analysis results show that most conditions are satisfied automatically in the centralized ALC except for one particular dual optimality condition Eq. (5.16), which is not guaranteed by the traditional weight

update. Based on this, the “primal residual” and “dual residual” are introduced to decomposition-based optimization and their mathematical definitions are given according to the KKT conditions for the master problem in the centralized ALC. Inspired by relevant research on augmented Lagrangian relaxation in the convex optimization area, a new robust weight update considering both the primal and dual residuals is introduced to engineering optimization based on the above theoretical exploration. This penalty weight can either increase or decrease depending on the comparison of the primal and dual residuals. Additionally, a new convergence check criterion is also proposed based on the primal and dual residuals.

Both mathematical and engineering benchmark problems are used to test the proposed update and both a calculated initial weight and a fine-tuned initial weight are adopted to provide a comprehensive comparison between the traditional update and the proposed update. The experimental results on the first two problems indicate that the new update decreases the number of iterations consumed to reach the AIO optimal solution. The results of the micro-accelerometer design benchmark problem show that the proposed strategy greatly increases the solution accuracy by reducing the objective error from above 10% to below 1% which can be further improved using the new convergence check criterion. These results prove that driving the dual residual to zero through our proposed method greatly improves the ability of the centralized ALC to produce a better optimality. More importantly, the new update enables the centralized ALC method to converge regardless of the magnitude of the initial weight since it allows the weight to

either increase or decrease during optimization. This largely reduces the sensitivity of the ALC performance to the initial weight and brings robustness to the centralized ALC.

In the future, it is necessary to study the reasons of the oscillations in the objective error, dual residual, and primal residual in Figure 5.20, Figure 5.22, and Figure 5.23, as these oscillations consume more than half of the overall iterations in those cases. The effects of μ and τ on the performance of the proposed methods need to be explored. Additionally, the applicability of the proposed theory on dual residual to a broader category of problems beyond quasi-separable problems (such as problems with system-wide constraints) is worth investigating.

Chapter 6

Dual Residual for Distributed ALC Based on Optimality Conditions

The update of the penalty weights in the penalty terms of ALC greatly affects the optimization results [20]. The traditional update assigns the initial weight \mathbf{w} a relatively small value and then increases it during the optimization iterations, which is shown in Eq. (2.18). However, this monotone update may lead to irrelevant solutions if starting with an inappropriate initial \mathbf{w} [20]. In the last chapter, a new non-monotone update strategy based on dual residual was proposed recently and has shown to increase the efficiency, accuracy, and robustness of the centralized ALC [65][75]. The dual residual is considered as an indicator of the satisfaction of the Karush-Kuhn-Tucker (KKT) optimality conditions of the All-in-one (AIO) problem and the new update manages to drive the dual residual to zero, which is not achieved by the traditional update.

In this chapter, the theory of dual residual on the centralized ALC is extended to the distributed ALC. A new update based on the dual residual is proposed and demonstrated to improve the performance of the distributed ALC. Section 6.1 examines and compares the KKT optimality conditions for the AIO problem and the decomposed problems, which leads to the definition of primal and dual residual in the distributed ALC. A new weight update is proposed based on both residuals. Section 6.2 presents numerical

tests of the proposed method and the analysis of the results. The chapter is summarized in Section 6.3. The research presented in this chapter is also published in [78].

6.1 Optimality Conditions for Distributed ALC and a New Weight Update based on Dual Residuals

Assume we have a quasi-separable problem whose objective and constraints are coupled by linking variables \mathbf{y} . These variables can be either the shared design variables between constraints and objective or the outputs of one analysis function required as inputs for another analysis function. The general formulation of this quasi-separable problem is shown below.

$$\begin{aligned}
 & \min_{\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M} \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\
 & s.t. \quad \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \\
 & \quad \quad \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \\
 & \quad \quad j = 1, \dots, M
 \end{aligned} \tag{6.1}$$

Distributed ALC can decompose problem (6.1) into M sub-problems and solve these sub-problems iteratively until they reach a consistent and optimal solution. A few steps need to be followed to reach this process [20]. First, $M-1$ copies of linking variables \mathbf{y} : \mathbf{y}_j ($j = 1, \dots, M-1$) are introduced into the problem, and \mathbf{y} is renamed as \mathbf{y}_M . New equality constraints $\mathbf{y}_j - \mathbf{y}_n$ are required to guarantee these copies share the same value. These are called consistency constraints.

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j} \sum_{j=1}^M f_j(\mathbf{x}_j, \mathbf{y}_j) \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \\
& \quad \mathbf{c}_{jn} = \mathbf{y}_j - \mathbf{y}_n = \mathbf{0} \\
& \quad \{n \in N_j \mid n > j\} \quad j = 1, \dots, M
\end{aligned} \tag{6.2}$$

N_j denotes the set of indices of all sub-problems that are coupled with sub-problem j through linking variables. The consistency constraints are then relaxed to the objective by the augmented Lagrangian relaxation [29], resulting in the following relaxed AIO formulation:

$$\begin{aligned}
& \min_{\mathbf{x}_1, \dots, \mathbf{x}_M, \mathbf{y}_1, \dots, \mathbf{y}_M} \sum_{j=1}^M f_j(\mathbf{x}_j, \mathbf{y}_j) + \sum_{j=1}^{M-1} \sum_{n \in N_j | n > j} \mathbf{v}_{jn}^T (\mathbf{y}_j - \mathbf{y}_n) + \\
& \quad \sum_{j=1}^{M-1} \sum_{n \in N_j | n > j} \|\mathbf{w}_{jn} \circ (\mathbf{y}_j - \mathbf{y}_n)\|_2^2 \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \quad \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad j = 1, \dots, M
\end{aligned} \tag{6.3}$$

Now the constraints are fully separable and problem (6.3) can be solved by iteratively optimizing sub-problem j ($j = 1, \dots, M$) with respect to \mathbf{x}_j and \mathbf{y}_j . The general formulation of sub-problem j is:

$$\begin{aligned}
& \min_{\mathbf{x}_j, \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \\
& \sum_{n \in N_j | n > j} \mathbf{v}_{jn}^T (\mathbf{y}_j - \mathbf{y}_n) + \sum_{n \in N_j | n > j} \left\| \mathbf{w}_{jn} \circ (\mathbf{y}_j - \mathbf{y}_n) \right\|_2^2 + \\
& \sum_{n \in N_j | n < j} \mathbf{v}_{nj}^T (\mathbf{y}_n - \mathbf{y}_j) + \sum_{n \in N_j | n < j} \left\| \mathbf{w}_{nj} \circ (\mathbf{y}_n - \mathbf{y}_j) \right\|_2^2 \\
& \text{subject to } \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \\
& \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0}
\end{aligned} \tag{6.4}$$

6.1.1 The necessary optimality conditions for the relaxed AIO problem

Next the KKT optimality necessary conditions are applied to both the AIO problem as it exists before decomposition and to the sub-problem j resulting from decomposition. In order to reach the optimal solution, the distributed ALC must fully satisfy all the KKT conditions of the AIO problem.

The KKT optimality conditions for Eq.(6.2) are

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M f_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M (\mathbf{u}_i^{(g)})^T \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \\
& \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^M (\mathbf{v}_i^{(h)})^T \mathbf{h}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{x}_j} \left[\sum_{i=1}^{M-1} \sum_{n \in N_j | n > i} \mathbf{v}_{in}^T (\mathbf{y}_i - \mathbf{y}_n) \right] = \mathbf{0}
\end{aligned} \tag{6.5}$$

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M f_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M (\mathbf{u}_i^{(g)})^T \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \\
& \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^M (\mathbf{v}_i^{(h)})^T \mathbf{h}_i(\mathbf{x}_i, \mathbf{y}_i) \right] + \frac{\partial}{\partial \mathbf{y}_j} \left[\sum_{i=1}^{M-1} \sum_{n \in N_i | n > i} \mathbf{v}_{in}^T (\mathbf{y}_i - \mathbf{y}_n) \right] = \mathbf{0} \\
& j = 1, 2, \dots, M
\end{aligned} \tag{6.6}$$

$$\mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \leq \mathbf{0} \quad (6.7)$$

$$\mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad (6.8)$$

$$\mathbf{c}_{jn} = \mathbf{y}_j - \mathbf{y}_n = \mathbf{0} \quad \{n \in N_j \mid n > j\} \quad (6.9)$$

$$(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) = \mathbf{0} \quad (6.10)$$

$$\mathbf{u}_j^{(g)} \geq \mathbf{0} \quad (6.11)$$

$$j = 1, 2, \dots, M$$

Eqs. (6.5), (6.6) can be simplified as

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{x}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] + \\ & \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{u}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] = \mathbf{0} \end{aligned} \quad (6.12)$$

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{y}_j} f_j(\mathbf{x}_j, \mathbf{y}_j) + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] + \\ & \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] + \sum_{n \in N_j \mid n > j} \mathbf{v}_{jn}^T - \sum_{n \in N_j \mid n < j} \mathbf{v}_{nj}^T = \mathbf{0} \end{aligned} \quad (6.13)$$

$$j = 1, 2, \dots, M$$

6.1.2 The necessary optimality conditions for distributed ALC

Since the distributed ALC solves sub-problems in an iterative way, the superscripts are used here to indicate the number of iterations. For sub-problem j in Eq (6.4) at iteration $k+1$, $\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}$ are the optimal solutions for the sub-problem. Thus according to the KKT necessary optimality conditions, these solutions must satisfy

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{x}_j} f_i(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} + \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} \\ & \frac{\partial}{\partial \mathbf{x}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{x}_j = \mathbf{x}_j^{k+1}} = \mathbf{0} \end{aligned} \quad (6.14)$$

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{y}_j} f_i(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \\ & \frac{\partial}{\partial \mathbf{y}_j} [(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \sum_{n \in N_j | n > j} \mathbf{v}_{jn}^k + \\ & \sum_{n \in N_j | n > j} [2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_j - \mathbf{y}_n^k)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} - \sum_{n \in N_j | n < j} \mathbf{v}_{nj}^k - \\ & \sum_{n \in N_j | n < j} [2\mathbf{w}_{nj}^k \circ \mathbf{w}_{nj}^k \circ (\mathbf{y}_n^{k+1} - \mathbf{y}_j)] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} = \mathbf{0} \end{aligned} \quad (6.15)$$

$$\mathbf{g}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) \leq \mathbf{0} \quad (6.16)$$

$$\mathbf{h}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) = \mathbf{0} \quad (6.17)$$

$$(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j^{k+1}, \mathbf{y}_j^{k+1}) = \mathbf{0} \quad (6.18)$$

$$\mathbf{u}_j^{(g)} \geq \mathbf{0} \quad (6.19)$$

$$j = 1, 2, \dots, M$$

Now the KKT conditions for distributed ALC in Eqs. (6.14) - (6.19) can be compared to those of the AIO problem in Eqs.(6.7) - (6.13). As illustrated in Figure 6.1, Eqs. (6.14), (6.16) - (6.19) guarantee the conditions in Eqs. (6.12), (6.7), (6.8), (6.10), and (6.11) at iteration $k+1$. Considering that condition (6.9) is reached at the end of the coordination as a convergence check criterion [20], the only condition left unchecked is condition (6.13). Before addressing that condition, some transformations need to be provided based on the method of multipliers used in the ALC [20].

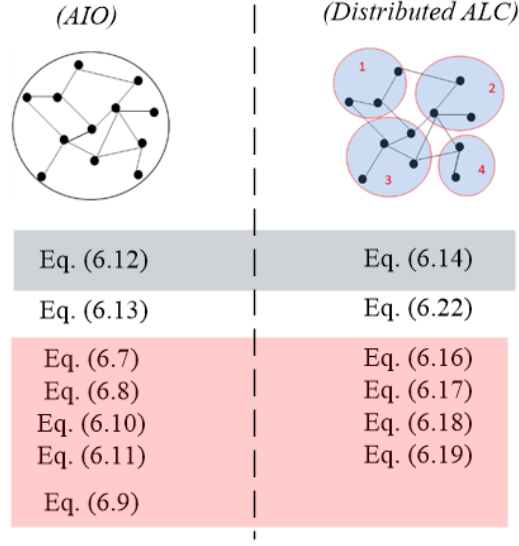


Figure 6.1: Comparison of KKT conditions between the AIO and distributed ALC (the equations with shadow background on the left are guaranteed by the equations with shadow background on the right or convergence criteria)

Since the sub-problems in the distributed ALC are solved sequentially, at iteration $k+1$ sub-problem j can use the latest information passed from sub-problem n ($n \in N_j | n < j$), but it has to use the information of the last iteration for sub-problem n ($n \in N_j | n > j$). Between sub-problem j and sub-problem n ($n \in N_j | n < j$) there is a relationship as follows

$$\begin{aligned}
& \sum_{n \in N_j | n < j} \mathbf{v}_{nj}^k + \sum_{n \in N_j | n < j} \left[2\mathbf{w}_{nj}^k \circ \mathbf{w}_{nj}^k \circ (\mathbf{y}_n^{k+1} - \mathbf{y}_j) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\
&= \sum_{n \in N_j | n < j} \left[\mathbf{v}_{nj}^k + 2\mathbf{w}_{nj}^k \circ \mathbf{w}_{nj}^k \circ (\mathbf{y}_n^{k+1} - \mathbf{y}_j) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\
&= \sum_{n \in N_j | n < j} \mathbf{v}_{nj}^{k+1} \\
& j = 1, 2, \dots, M
\end{aligned} \tag{6.20}$$

Between sub-problem j and sub-problem n ($n \in N_j | n > j$) there is a relationship as follows

$$\begin{aligned}
& \sum_{n \in N_j | n > j} \mathbf{v}_{jn}^k + \sum_{n \in N_j | n > j} \left[2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_j - \mathbf{y}_n^k) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\
&= \sum_{n \in N_j | n > j} \left[\mathbf{v}_{jn}^k + 2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_j - \mathbf{y}_n^k) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} \\
&= \sum_{n \in N_j | n > j} \left[\mathbf{v}_{jn}^k + 2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_j^{k+1} + \mathbf{y}_n^{k+1} - \mathbf{y}_n^{k+1} - \mathbf{y}_n^k) \right] \\
&= \sum_{n \in N_j | n > j} \left[\mathbf{v}_{jn}^k + 2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_j^{k+1} - \mathbf{y}_n^{k+1}) \right] + \\
& \quad \sum_{n \in N_j | n > j} \left[2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^{k+1} - \mathbf{y}_n^k) \right] \\
&= \sum_{n \in N_j | n < j} \mathbf{v}_{jn}^{k+1} + \sum_{n \in N_j | n > j} \left[2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^{k+1} - \mathbf{y}_n^k) \right] \\
& \quad j = 1, 2, \dots, M
\end{aligned} \tag{6.21}$$

Substituting Eqs. (6.20), (6.21) into Eq. (6.15), the equation becomes:

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{y}_j} f_i(\mathbf{x}_j, \mathbf{y}_j) \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \frac{\partial}{\partial \mathbf{y}_j} \left[(\mathbf{u}_j^{(g)})^T \mathbf{g}_j(\mathbf{x}_j, \mathbf{y}_j) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \\
& \frac{\partial}{\partial \mathbf{y}_j} \left[(\mathbf{v}_j^{(h)})^T \mathbf{h}_j(\mathbf{x}_j, \mathbf{y}_j) \right] \Big|_{\mathbf{y}_j = \mathbf{y}_j^{k+1}} + \sum_{n \in N_j | n > j} \mathbf{v}_{jn}^{k+1} - \sum_{n \in N_j | n < j} \mathbf{v}_{nj}^{k+1} \\
&= \sum_{n \in N_j | n < j} \left[2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^k - \mathbf{y}_n^{k+1}) \right] \\
& \quad j = 1, 2, \dots, M
\end{aligned} \tag{6.22}$$

Comparing condition (6.22) to condition (6.13), one can see that the solutions from the distributed ALC are not guaranteed to meet condition (6.13). There are many terms on the right hand side of Eq. (6.22) which depend on the weights \mathbf{w} and on the consistency error. These have to be resolved.

6.1.3 The primal residual and dual residual for distributed ALC

Let \mathbf{s}_j^{k+1} denote the right hand side of Eq. (6.22):

$$\mathbf{s}_j^{k+1} = \sum_{n \in N_j | n < j} \left[2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^k - \mathbf{y}_n^{k+1}) \right].$$

This term can be viewed as the residual of the dual

feasibility condition (6.13) and can be further split into $\mathbf{s}_j^{k+1} = \sum_{n \in N_j | n < j} \mathbf{s}_{jn}^{k+1}$,

where $\mathbf{s}_{jn}^{k+1} = 2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^k - \mathbf{y}_n^{k+1})$. \mathbf{s}_{jn}^{k+1} is called the dual residual of sub-problem j corresponding to the consistency constraint $\mathbf{y}_j - \mathbf{y}_n = \mathbf{0}$ at iteration $k+1$. If all \mathbf{s}_{jn}^{k+1} for $j = 1, \dots, M$ and $n \in N_j | n < j$ are equal to 0, then \mathbf{s}_j^{k+1} is equal to 0, which guarantees the optimality condition in Eq. (6.13).

Based on the above analysis, the primal residual and dual residual for sub-problem j at iteration $k+1$ are defined as follows:

The primal residual:

$$\mathbf{r}_j^{k+1} = \mathbf{y}_j^{k+1} - \mathbf{y}_n^{k+1} \quad (6.23)$$

The dual residual:

$$\mathbf{s}_j^{k+1} = 2\mathbf{w}_{jn}^k \circ \mathbf{w}_{jn}^k \circ (\mathbf{y}_n^k - \mathbf{y}_n^{k+1}) \quad (6.24)$$

6.1.4 A new update strategy for distributed ALC

In distributed ALC, both the primal and dual residuals need to be equal to zero for the coordination to generate an optimal solution satisfying the KKT conditions. From the definitions of the residuals in Eqs. (6.23) and (6.24), it can be seen that the penalty weight \mathbf{w} has contrary effects on the primal and dual residuals: large \mathbf{w} can force the

optimization to quickly converge to a solution with small primal residual, while it makes the dual residual big. Small \mathbf{w} decreases the dual residual but fails to penalize a large primal residual generated during optimization. This indicates that the monotone weight update strategies including the popular one in Eq.(2.18) are not the best for distributed ALC.

Similar to the update proposed in [65][75] and in last chapter, a new weight update for distributed ALC is proposed in Eq. (6.25). The update is non-monotone and considers both the primal and dual residuals when changing \mathbf{w} , while the traditional weight update in Eq. (2.18) only considers the primal residual.

$$\mathbf{w}_j^{k+1} := \begin{cases} \tau^{incr} \mathbf{w}_j^k & \text{if } \|\mathbf{r}_j^k\|_2 > \mu \|\mathbf{s}_j^k\|_2 \\ \mathbf{w}_j^k / \tau^{decr} & \text{if } \|\mathbf{s}_j^k\|_2 > \mu \|\mathbf{r}_j^k\|_2 \\ \mathbf{w}_j^k & \text{otherwise} \end{cases} \quad (6.25)$$

where $\mu > 1$, $\tau^{incr} > 1$ and $\tau^{decr} > 1$ are parameters used to set how quickly \mathbf{w} is increased or decreased. If the primal residual is much larger than the dual residual, \mathbf{w} is multiplied by τ^{incr} to get a larger \mathbf{w} to penalize the primal residual. If the dual residual is much larger than the primal residual, \mathbf{w} is divided by τ^{decr} to reduce the dual residual. Through keeping a balance between the primal and dual residuals, the new update is expected to drive both residuals to zero. During implementation, each element w_{ji} of the weight vector \mathbf{w}_j in Eq. (6.25) can be updated separately, which is what is used in the following tests.

There are no specific values for μ and τ that can work perfectly with any problem, just as the β and γ in ALC, which are given a recommended range instead of fixed values. The parameter μ in Eq. (6.25) controls the gap between the primal residual \mathbf{r} and the dual

residual \mathbf{s} , and a large value of μ allows a large difference between these two residuals and prevents the adjusting of \mathbf{w} accordingly to decrease this difference. Parameter τ in Eq. (6.25) controls the rate at which \mathbf{w} is changed, and a large value of τ can increase or decrease \mathbf{w} quickly, which is good for simple optimization problems but may cause convergence issues when solving complex problems.

Based on the above discussion, for this work, μ is recommended to be around 2 to guarantee that \mathbf{r} and \mathbf{s} are of the same order of magnitude. For simple problems, τ can be assigned a relatively large value (around 1.5) to accelerate the convergence, while for complex problems, a small τ (1.0 – 1.2) is suggested to make the optimization process smoother. The values of τ^{incr} and τ^{decr} used for increasing \mathbf{w} and decreasing \mathbf{w} respectively do not necessarily need to be the same, but they are kept equal in this work. Numerical explorations for the feasible settings of μ and τ are conducted on the first two examples in Section 6.2.

6.2 Numerical Tests

The geometric programming and micro-accelerometer design problems in section 3.2 are used again to test the proposed update. Both the traditional update in Eq. (2.18) and the proposed update in Eq. (6.25) are applied to the distributed ALC and their performance on the test problems are compared in tables and figures. In the optimization process figures there are four curves: the biggest primal and dual residuals (using Eqs. (6.23) and (6.24)), the biggest design variable error (the relative error between the ALC solution and AIO reference solution), and the objective error (the relative error between the ALC objective and AIO reference objective).

6.2.1 Example 1: Convex quadratic programming problem

The first test example is a convex quadratic programming that has been used in [79][48]. It has a unique global optimal solution. The problem is partitioned into four sub-problems as shown in Figure 6.2.

$$\begin{aligned}
 \min_{z_1, \dots, z_{14}} \quad & f = z_1^2 + z_2^2 \\
 \text{s.t.} \quad & g_1 = -z_3 + z_4 - z_5 + 2 \leq 0, \\
 & g_2 = z_5 - z_6 + z_7 + 1 \leq 0, \\
 & g_3 = z_8 + z_9 - z_{11} + 1 \leq 0, \\
 & g_4 = -z_8 + z_{10} - z_{11} + 3 \leq 0, \\
 & g_5 = z_{11} - z_{12} - z_{13} + 2 \leq 0, \\
 & g_6 = z_{11} + z_{12} - z_{14} + 1 \leq 0, \\
 & h_1 = z_3 + z_4 + z_5 - z_1 + 1 = 0, \\
 & h_2 = z_5 + z_6 + z_7 - z_2 + 1 = 0, \\
 & h_3 = z_8 - z_9 - z_{10} + z_{11} - z_3 + 1 = 0, \\
 & h_4 = z_{11} + z_{12} + z_{13} + z_{14} - z_6 = 0, \\
 & z_1, \dots, z_{14} > 0
 \end{aligned} \tag{6.26}$$

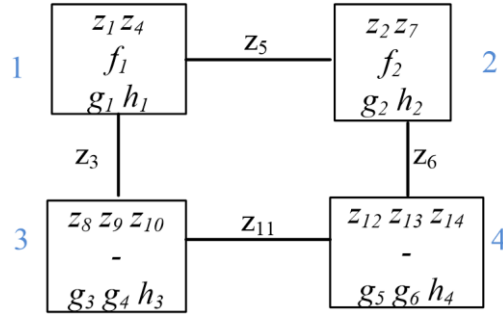


Figure 6.2: The partition of Example 1 with four sub-problems

All the design variables are initialized to one and all Lagrangian multipliers \mathbf{v} are initialized to zero. For the traditional update, the following parameters are set: $\beta = 1.1$ and $\gamma = 0.9$, as suggested in [20]. For the proposed update, the

parameters: $\mu = 2$ and $\tau = 1.5$ are set as suggested in [75]. The termination tolerance is $\varepsilon = 10^{-3}$. The initial weight \mathbf{w} is calculated and is equal to 1.65 according to Eq. (5.34) with $\hat{f} = 500$, $\alpha = 0.1$, and $\mathbf{w}^0 = 0.001$. Since the sequence in which the sub-problems are solved may affect the optimization results, the four sub-problems are optimized in four different orders: 1-2-4-3, 2-4-1-3, 3-4-2-1, and 4-3-2-1. The number of iterations and number of function evaluations of the traditional and proposed update are summarized in Table 6.1. The typical optimization processes of ALC using the traditional and proposed update are shown in Figure 6.3 and Figure 6.4 respectively.

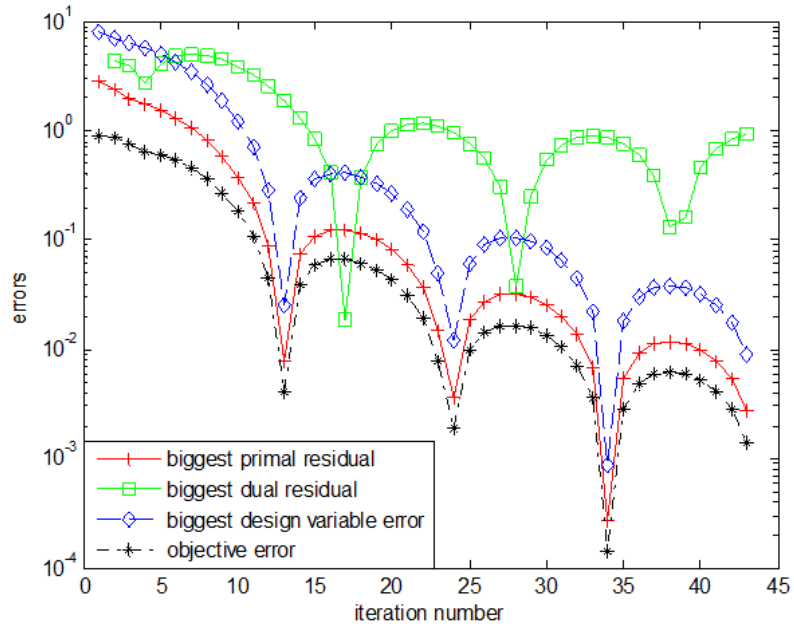


Figure 6.3: The optimization process with the traditional update on Example 1 (solving sequence: 1-2-4-3)

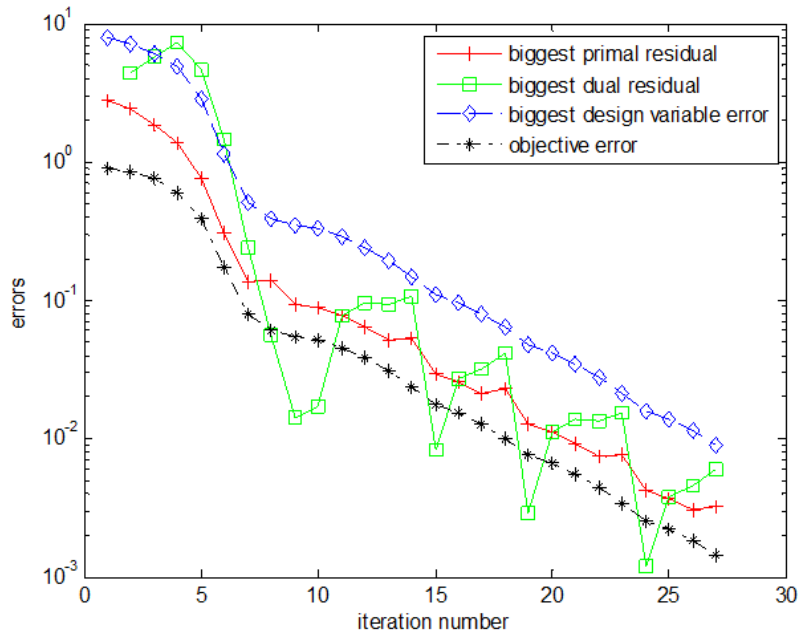


Figure 6.4: The optimization process with the proposed update on Example 1 (solving sequence: 1-2-4-3)

Figure 6.3 shows some fluctuations in the optimization process with the traditional update and the optimization consumes 43 iterations. The biggest primal residual almost decreases to 0.001 while the biggest dual residual is around 1, which is much larger than the biggest primal residual. However in Figure 6.4, both the primal and dual residuals gradually drop to 0.003, which means the proposed update works on decreasing both residuals. As a result, the design variable and objective errors of the proposed update are diminished smoothly toward zero within 26 iterations. Table 6.1 shows that for the same solving sequences, the proposed update can always decrease the number of iterations and number of function evaluations when compared to the traditional update.

Table 6.1: Results of different coordination sequences for Example 1

Sub-problem solving sequence	Traditional update		Proposed update	
	# of iters.	# of func.evals.	# of iters.	# of func.evals.
1-2-4-3	43	2320	27	1338
2-4-1-3	40	2209	27	1466
3-4-2-1	51	2795	37	2051
4-3-2-1	53	2905	38	2204

Another set of tests is also conducted on this convex quadratic programming problem where the values of μ and τ are changed from 1.1 to 2.9 with increments of 0.2, and the problem is solved 100 times using these different μ and τ . The number of function evaluations associated with these μ and τ are recorded and plotted in Figure 6.5. This test does not serve the purpose of choosing appropriate μ and τ for this particular problem, but rather it is designed to empirically investigate the effects of μ and τ on the proposed update.

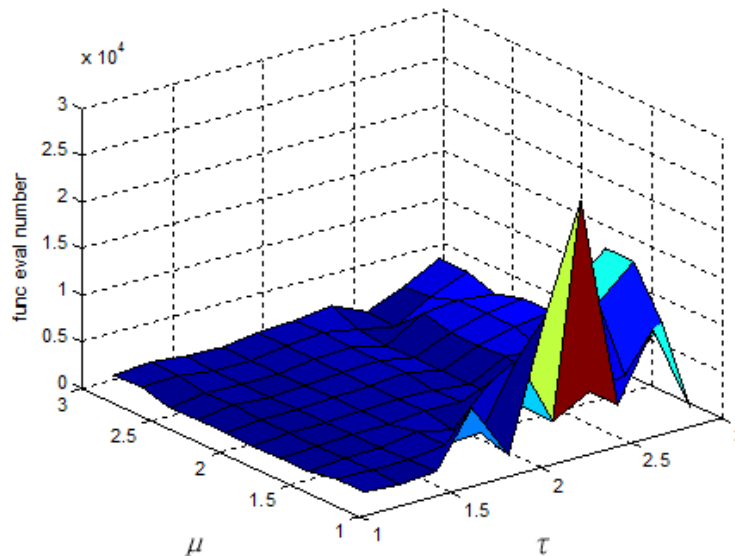


Figure 6.5: Number of function evaluations resulting from different μ and τ of the proposed update for Example 1

The right corner range ($\mu \in [1.1, 1.9]$ and $\tau \in [1.7, 2.9]$) in Figure 6.5 depicts high numbers of function evaluations and so should be avoided, while the left corner shows low numbers consistently hence choosing $\mu \in [1.5, 2.9]$ and $\tau \in [1.1, 1.7]$ can yield a low number of function evaluations and therefore higher efficiency for this problem.

6.2.2 Example 2: geometric programming problem

The second example has appeared in Section 4.2.1. The partition illustrated in Figure 4.4 (a) is used to solve this problem. All the design variables for this example are initialized to one and all Lagrangian multipliers \mathbf{v} are initialized to zero. For the traditional update, the parameters are set to $\beta = 1.1$ and $\gamma = 0.9$. For the proposed update, $\mu = 2$ and $\tau^{incr} = \tau^{decr} = 1.5$. The termination tolerance is $\varepsilon = 10^{-3}$. The initial weight \mathbf{w} is calculated and is equal to 0.02 according to Eq. (5.34) with $\hat{f} = 50$, $\alpha = 0.1$, and $\mathbf{w}^0 = 0.001$. Each update is tested 10 times using random initial points selected between 0 and 1. The decomposed sub-problems are solved in five different sequences: 1-2-3-4-5, 2-1-3-5-4, 3-5-1-2-4, 4-5-2-1-3, and 5-3-4-1-2.

Table 6.2: Results of the traditional and proposed update using different coordination sequences for Example 2

Sub-problem solving sequence	Traditional update		Proposed update	
	# of iters.	# of func.evals.	# of iters.	# of func.evals.
1-2-3-4-5	76	8579	28	4033
2-1-3-5-4	79	9002	28	3843
3-5-1-2-4	76	8559	28	5875
4-5-2-1-3	72	7861	30	4746
5-3-4-1-2	72	13432	35	4652

The optimization results for the tests are summarized in Table 6.2, which shows that the proposed update can effectively decrease the number of iterations and number of function evaluations no matter which solving sequence is used. The typical optimization processes of distributed ALC with the traditional and proposed updates are shown in Figure 6.6 and Figure 6.7. The biggest dual residuals in both figures reach a small value at the end of optimization. However, it takes the traditional update 76 iterations to make the optimization converge while the proposed update only needs 28 iterations. Close inspection reveals that for the proposed update, starting from the 5th iteration, the biggest dual residual keeps producing similar values as the biggest primal residual. But in the traditional update the values of both residuals are far apart from each other until the 16th iteration. This proves that the proposed update has successfully controlled the primal and dual residuals at the same magnitude and has simultaneously driven both of them to zero, which saves many iterations.

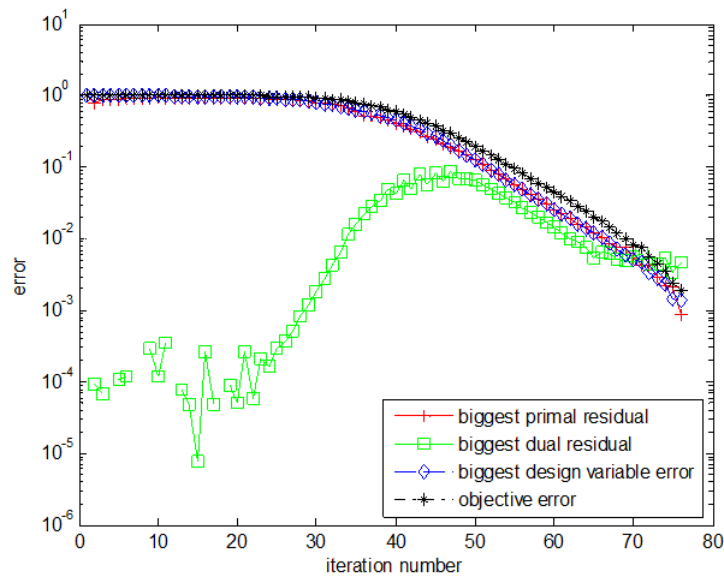


Figure 6.6: The optimization process with the traditional update on Example 2 (solving sequence: 1-2-3-4-5)

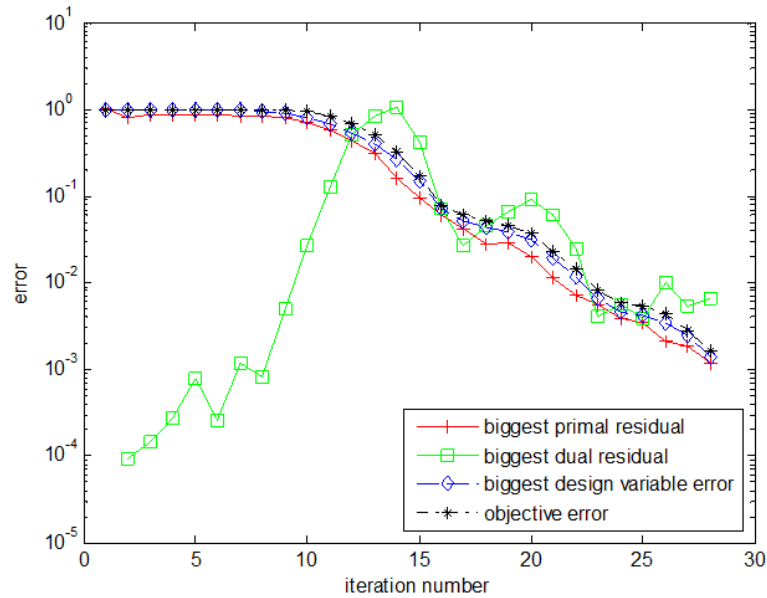


Figure 6.7: The optimization process with the proposed update on Example 2 (solving sequence: 1-2-3-4-5)

Experimental explorations of the effects of μ and τ on the performance of the proposed update are also conducted and the results are shown in Figure 6.8. μ is changed from 1.1 to 2.9 with an increment of 0.2 and τ is changed from 1.1 to 2.7 with an increment of 0.2. It can be seen from Figure 6.8 that choosing $\mu \in [1.1, 2.9]$ and $\tau \in [1.1, 2.1]$ is appropriate for this problem. The value $\tau = 2.9$ is also tested but it turns out to be so inappropriate that the optimization fails.

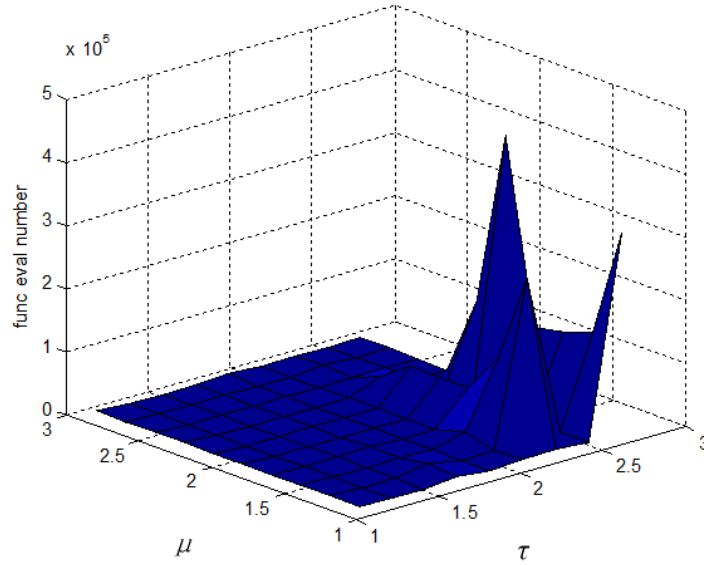


Figure 6.8: Number of function evaluations resulting from different μ and τ of the proposed update for Example 2

6.2.3 Example 3: Micro-accelerometer benchmark design problem

The centralized ALC shown in Figure 4.6 (a) is used again to solve this problem. To get comprehensive results, the initial design variables are set randomly between 50% and 150% of the baseline design. All the parameter settings are the same as in the previous two examples except for $\tau^{incr} = \tau^{decr} = 1.2$ because a small value of the update parameter τ is more suitable for this kind of complex problem. The initial weight \mathbf{w} is calculated and is equal to 0.3 according to Eq. (5.34) with $\hat{f} = 0.5$, $\alpha = 0.1$, and $\mathbf{w}^0 = 0.001$. The problem is solved 10 times for each update and the results are summarized in Table 6.3.

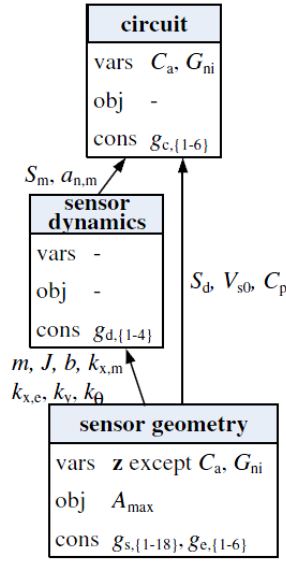


Figure 6.9: Object based partition of Example 3 [60]

Table 6.3 shows that the mean objective error of the distributed ALC is decreased from 1.7% to 0.4% by the proposed update. More iterations (35% increase) and function evaluation (40% increase) are required by the proposed update to drive the optimization to better solutions. In addition to just driving the primal residual to zero as in the traditional update, the proposed update is also in charge of driving the dual residual to zero and keeping a balance between the two residuals. This might be the reason that the proposed update needs more iterations and function evaluations.

Figure 6.10 and Figure 6.11 present the typical optimization process of distributed ALC with both updates. The biggest dual residual in the proposed update in Figure 6.11 reaches a smaller value than that in the traditional update in Figure 6.10, which means the solution in Figure 6.11 better satisfies condition (6.13) than the solution in Figure 6.10, and this leads to the decrease in the objective error. Moreover, if the

results are compared with those of the AIO solutions in Table 4.2, it can be seen that the proposed update attains the same mean objective and a better maximum objective. But only 85% of the AIO optimizations fail to converge while all of the decomposition based optimizations successfully reach the accurate solution. This approach is therefore very promising for decomposition based optimization because it shows that a better convergence property can be achieved without undermining the solution accuracy when solving this problem in decomposed fashion instead of as a whole.

Table 6.3: Results of the traditional and the proposed updates on Example 3

	traditional update			proposed update		
	Obj.(mm ²) /error	# of iters.	# of func.evals.	Obj.(mm ²) /error	# of iters	# of func.evals.
<i>Min</i>	0.0819 / 1.5%	28	16819	0.0808 / 0.1%	39	24301
<i>Mean</i>	0.0821 / 1.7%	31	18897	0.0810 / 0.4%	42	26525
<i>Max</i>	0.0825 / 2.2%	33	22028	0.0814 / 0.9%	46	29549

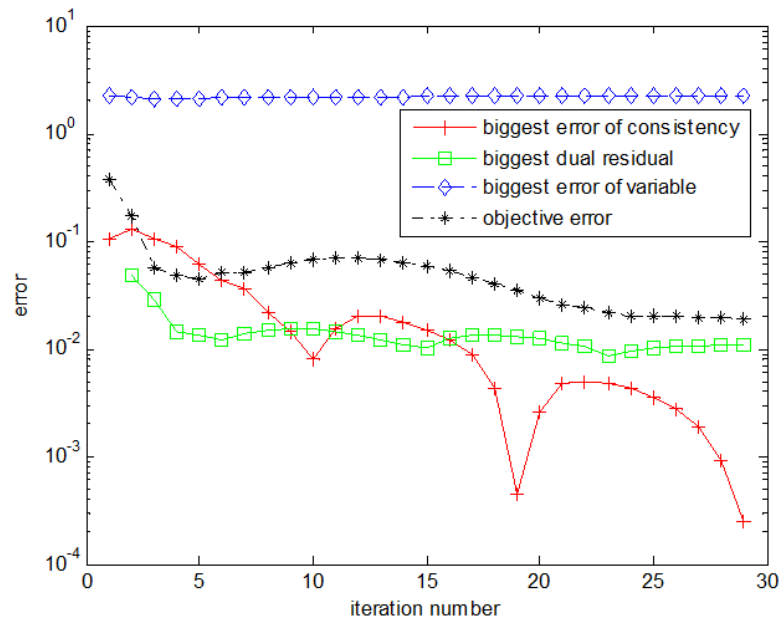


Figure 6.10: The optimization process with the traditional on Example 3

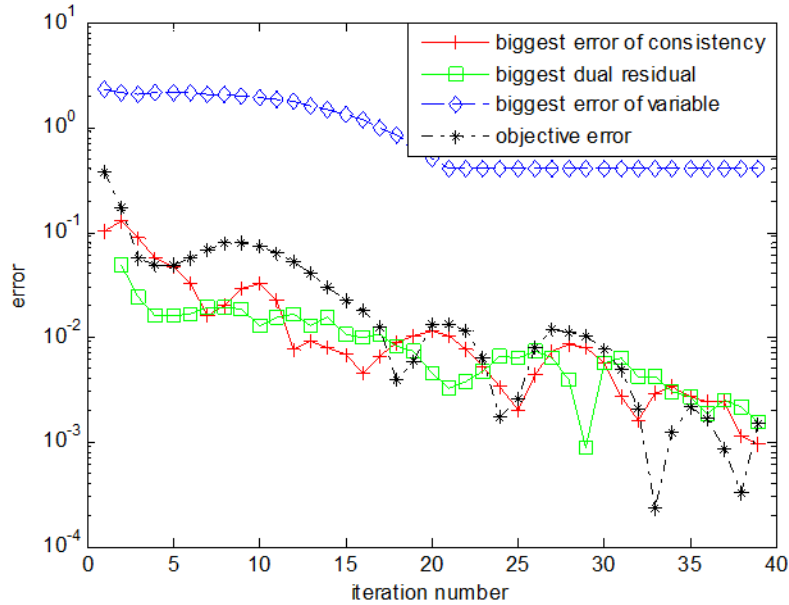


Figure 6.11: The optimization process with the proposed updates on Example 3

6.2.4 Robustness of the proposed strategy

The initial weight \mathbf{w} has a significant effect on the performance of the distributed ALC using the traditional update. An excessive initial weight often causes the optimization process to converge early to irrelevant solutions while a small initial weight often significantly slows down the convergence rate of the optimization. The setting method in Eq. (5.34) is heuristic and often needs some subsequent trial and error runs. Thus, a robust approach to selecting the initial \mathbf{w} is highly desired by the distributed ALC.

Since the proposed update can either increase or decrease \mathbf{w} , it is more flexible than the traditional monotone update and has the potential to show some robustness with respect to initial weights. To verify this assumption, another set of tests on the micro-accelerometer problem is performed, for which all the settings are kept the same as in the previous section except that the initial weights are set to 1, 0.1, 0.01, and 0.001. Both

updates are tested for 10 runs starting with each of these initial weights and their mean values of the objective error and number of function evaluations are presented in Table 6.4.

Table 6.4: Optimization results of Example 3 problem with different initial weights

Initial penalty weight	Traditional update			Proposed update		
	Obj. (mm ²) /error	# of iters.	# of func. evals.	Obj. (mm ²) /error	# of iters.	# of func. evals.
1	0.1127 /39.7%	18	12616	0.0812 /0.6%	46	31217
0.1	0.0929 /15.1%	82	85925	0.0809 /0.2%	111	106568
0.01	0.1397 /73.1%	117	166339	0.0809 /0.2%	148	169699
0.001	0.1017 /26.0%	140	177979	0.0810 /0.4%	157	176066

Table 6.4 shows that regardless of what the magnitude of the initial weight is, the proposed update is always able to consistently drive the optimization of the distributed ALC towards an accurate solution with an objective error less than 0.6%. In contrast, the solutions for the traditional update are much worse because the traditional update is very sensitive to the initial weights. Any inappropriate initial \mathbf{w} may push the optimization away from the optimal solution. This shows that, for this example certainly, the proposed update is more flexible and significantly enhances the robustness of the distributed ALC. The high number of function evaluations with small initial weights in both updates is reasonable, because the small penalty weights allow relatively large primal consistency constraint violation at the beginning of the optimization. As a result, both updates need more computational efforts to eventually reduce this violation to zero.

6.3 Summary

The KKT conditions are valuable indicators of optimality in mathematical programming and are expected to greatly benefit decomposition based optimization as well. The work in Chapter 5 shows that the dual residual theory based on the KKT conditions substantially improves the performance of the centralized ALC. In this chapter, the theory of dual residual is extended from the centralized ALC to the distributed ALC which solves decomposed sub-problems sequentially without a master problem. The KKT optimality conditions for the AIO problem and decomposed sub-problems are examined and compared. The comparison shows that a KKT condition is not guaranteed by the distributed ALC, which leads to the definition of the primal and dual residuals for the distributed ALC and the proposal of a non-monotone update for the penalty weights. The new update is able to drive both residuals simultaneously to zero, thus ensuring the resulting solutions satisfy all the KKT conditions of the AIO problem.

Numerical tests are conducted on the two geometric programming problems and the micro-accelerometer problem that are described in previous chapters. For the first two problems, the distributed ALC using the proposed update converges to the optimal solution much faster than the distributed ALC using the traditional update. The effects of μ and τ on the proposed update are explored experimentally and the results provide reasonable ranges for selecting these parameters. For the complex micro-accelerometer problem, the proposed update improves the objective error from 1.7% to 0.4% at the expense of consuming more function evaluations. When changing the initial weights between 0.001 to 1, the proposed update is always able to guide the optimization to an

accurate solution (with an objective error less than 0.5%) while the traditional update fails to do the same (with an objective error more than 20%).

Chapter 7

An Efficient Parallel Coordination Method Using Two Duality Theorems

7.1 Alternating Direction Method of Multipliers (ADMM)

Some theoretical studies in mathematical optimization have been applied to the decomposition-based optimization process and the Alternating Direction Method of Multipliers (ADMM) is one result of these studies [20][25]. ADMM [72] is a powerful algorithm developed in the field of convex optimization. The algorithm solves problems of the form:

$$\begin{aligned} & \text{minimize} && f_1(\mathbf{v}) + f_2(\mathbf{z}) \\ & \text{subject to} && \mathbf{A}\mathbf{v} + \mathbf{B}\mathbf{z} = \mathbf{e} \end{aligned} \tag{7.1}$$

The augmented Lagrangian function of this problem is

$$L_\rho(\mathbf{v}, \mathbf{z}, \mathbf{p}) = f_1(\mathbf{v}) + f_2(\mathbf{z}) + \mathbf{p}^T (\mathbf{A}\mathbf{v} + \mathbf{B}\mathbf{z} - \mathbf{e}) + (\rho/2) \|\mathbf{A}\mathbf{v} + \mathbf{B}\mathbf{z} - \mathbf{e}\|_2^2 \tag{7.2}$$

ADMM solves the problem in Eq. (7.1) through the following iterations

$$\mathbf{v}^{(k+1)} := \arg \min_{\mathbf{v}} L_\rho(\mathbf{v}, \mathbf{z}^{(k)}, \mathbf{p}^{(k)}) \tag{7.3}$$

$$\mathbf{z}^{(k+1)} := \arg \min_{\mathbf{z}} L_\rho(\mathbf{v}^{(k+1)}, \mathbf{z}, \mathbf{p}^{(k)}) \tag{7.4}$$

$$\mathbf{p}^{(k+1)} := \mathbf{p}^{(k)} + \rho(\mathbf{A}\mathbf{v}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{e}) \tag{7.5}$$

In decomposed-based optimization, the Augmented Lagrangian Coordination

(ALC) adopts ADMM and the resulting algorithm has become one of the most popular coordination methods in recent years. ADMM enables ALC to decouple the sub-problems using the augmented Lagrangian relaxation and to update the dual multipliers through the method of multipliers [20]. There are three types of ALC algorithms which either include an inner loop or not: Exact (ENMOM) inner loop, inexact (INMOM) inner loop, and alternating direction (ADMOM) without inner loop. At each iteration, ENMOM and INMON first use the Block Coordinate Descent (BCD) [29] method to iterate, solving each sub-problem with fixed dual multipliers and penalty weights until they reach a consistent solution, which represents the termination of the inner loop. Then the dual multipliers and penalty weights are updated in the outer loop to prepare the whole optimization process for the next iteration. The difference between the two inner loop methods is that ENMOM uses a small fixed convergence tolerance for the inner loop whereas in INMOM, the inner loop tolerance is relatively large at the beginning and decreases as the optimization process proceeds. ADMOM however, contains no inner loop since it uses ADMM and all sub-problems are solved only once at each iteration. Many tests have proven that ADMOM is the most robust and efficient algorithm among the three ALC algorithms since it does not require the expensive computational efforts of the inner loops when the current solution is far from the optimal solution [20]. Furthermore, the ADMOM algorithm is shown to perform better than other popular coordination methods such as the Analytical Target Cascading (ATC) which is based on a quadratic penalty method [22][23] and the Lagrangian duality-based coordination [52][24] which is based on the ordinary Lagrangian duality theorem.

Similar to ALC-ADMOM, Consensus Optimization via the Alternating Direction Method of Multipliers (CADMM) is another application of ADMM to decomposition-based optimization. At each iteration of CADMM, all sub-problems are solved only once and their solutions are collected together and used to calculate the consensus values for the shared variables, which are then used as targets for sub-problems in the next iteration. The efficacy of CADMM has been verified through several test problems [21][25][26][27].

In either ALC or CADMM, the ADMM is applied directly to the primal problem. In this study, inspired by the research in convex optimization [80], the ADMM is applied to the dual of the primal problem. Specifically, copies of shared variables are first introduced to the original optimization problem to get a primal problem set up for decomposition-based optimization, then the ordinary duality theorem is applied to that primal problem to generate a dual problem, after which ADMM is applied to the dual problem. In the resulting algorithm, all sub-problems are independent from each other and thus can be solved in parallel. The derivations of the algorithm are presented in Section 7.2. Section 7.3 tests the proposed algorithm using both mathematical and engineering problems and compares its performance with that of the parallel ALC-ADMOM. Section 7.4 summarizes this chapter. The research presented in this chapter is also published in [81].

7.2 Derivation

Depending on the ways an optimization problem is partitioned, the resulting partition can have either a hierarchical or a network structure. The hierarchical partition

can be considered a special case of the network partition, which is in great demand since in many cases partitions in the engineering area are non-hierarchical. Multi-disciplinary optimization exemplifies network structures, and several nonhierarchical coordination methods have been proposed in this area [25][24][53][69][75]. This study deals with decomposed sub-problems with a network structure.

Assuming a quasi-separable problem with M potential sub-systems with a network partition structure, the optimization problem is written as:

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{x}_1, \dots, \mathbf{x}_M} \quad & \sum_{j=1}^M f_j(\mathbf{y}, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \\ & j = 1, \dots, M \end{aligned} \quad (7.6)$$

where $\mathbf{y} \in R^N$ is the vector of shared variables, which can either be the linking design variables between two sub-problems or the analysis output of one sub-system which is required as input for another sub-system. $\mathbf{x}_j \in R^K$ is the vector of local design variables which appears only in sub-problem j .

7.2.1 Applying the ordinary duality theorem to the primal problem

To decompose the problem in Eq. (7.6) into M sub-problems, $M-1$ copies of \mathbf{y} are first introduced into the problem:

$$\begin{aligned} \min_{\mathbf{y}_1, \dots, \mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M} \quad & \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) \\ \text{subject to} \quad & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0} \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \\ & \mathbf{S}_1 \mathbf{y}_1 + \dots + \mathbf{S}_M \mathbf{y}_M = \mathbf{0} \\ & j = 1, \dots, M \end{aligned} \quad (7.7)$$

where \mathbf{y} is renamed \mathbf{y}_M and $\mathbf{y}_1, \dots, \mathbf{y}_{M-1}$ are the $M-1$ copies newly created.

The new equality constraints $\mathbf{S}_1\mathbf{y}_1 + \dots + \mathbf{S}_M\mathbf{y}_M = \mathbf{0}$ are called primal consistency constraints, and are added to the formulation to ensure that all the copies of the shared variables have the same value. Since elements in \mathbf{y}_j do not necessarily appear in every row of the consistency constraints, a selection matrix, $\mathbf{S}_j \in R^{P \times N}$, which is similar to those introduced by Michalek and Papalambros [82] for ATC and by Tosserams [20] for ALC, is adopted. The elements of \mathbf{S}_j can be 0, 1 or -1 and each row of \mathbf{S}_j must only contain one “1” or one “-1”. For example, assume a partition in Figure 7.1 with three sub-problems in a network structure coupled through three shared variables:

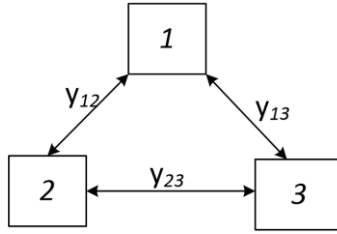


Figure 7.1: A partition with a three-node network structure for the illustration of selection matrix

The vector of shared variables in this partition is $\mathbf{y} = [y_{12}, y_{13}, y_{23}]^T$. After creating two copies of the shared variables $\mathbf{y}_1 = [y_{12}^1, y_{13}^1, y_{23}^1]^T$ and $\mathbf{y}_2 = [y_{12}^2, y_{13}^2, y_{23}^2]^T$, renaming \mathbf{y} as $\mathbf{y}_3 = [y_{12}^3, y_{13}^3, y_{23}^3]^T$, and assigning $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ to the sub-problems 1, 2 and 3 respectively, the resulting consistency constraints for the shared variables are $y_{12}^1 - y_{12}^2 = 0$, $y_{13}^1 - y_{13}^3 = 0$, and $y_{23}^2 - y_{23}^3 = 0$. Then the selection matrixes for the three sub-problems should be $\mathbf{S}_1 = [1 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 0]$, $\mathbf{S}_2 = [-1 \ 0 \ 0; 0 \ 0 \ 0; 0 \ 0 \ 1]$ and $\mathbf{S}_3 = [0 \ 0 \ 0; 0 \ 0 \ 0; -1 \ 0 \ 0]$, thus $\mathbf{S}_1\mathbf{y}_1 + \mathbf{S}_2\mathbf{y}_2 + \mathbf{S}_3\mathbf{y}_3 = \mathbf{0}$ would generate the above consistency constraints. It can be seen that some elements in \mathbf{y}_j are not used in sub-problem j at all, so

in the practical implementation, \mathbf{y}_j may only contain the shared variables relevant to sub-problem j instead of all shared variables.

Given the vector of Lagrangian multipliers $\mathbf{v} \in R^p$, the Lagrangian function L [29] is:

$$\begin{aligned} L(\mathbf{y}_1, \dots, \mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M, \mathbf{v}) &= \sum_{j=1}^M f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{v}^T (\mathbf{S}_1 \mathbf{y}_1 + \dots + \mathbf{S}_M \mathbf{y}_M) \\ &= \sum_{j=1}^M (f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{v}^T \mathbf{S}_j \mathbf{y}_j) \end{aligned} \quad (7.8)$$

The Lagrangian dual problem can be written as

$$\begin{aligned} &\underset{\mathbf{v}}{\text{maximize}} \quad d(\mathbf{v}) \\ &\text{where } d(\mathbf{v}) = \inf_{\substack{\mathbf{y}_1, \dots, \mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M \\ \mathbf{g}_1 \leq \mathbf{0}, \dots, \mathbf{g}_M \leq \mathbf{0}, \mathbf{h}_1 = \mathbf{0}, \dots, \mathbf{h}_M = \mathbf{0}}} L(\mathbf{y}_1, \dots, \mathbf{y}_M, \mathbf{x}_1, \dots, \mathbf{x}_M, \mathbf{v}) \end{aligned} \quad (7.9)$$

Since the Lagrangian function in Eq. (7.8) is separable, the dual problem can be

rewritten as:

$$\begin{aligned} &\underset{\mathbf{v}}{\text{maximize}} \quad \sum_{j=1}^M d_j(\mathbf{v}) \\ &\text{where } d_j(\mathbf{v}) = \inf_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} (f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{v}^T \mathbf{S}_j \mathbf{y}_j) \\ &\quad j = 1, \dots, M \end{aligned} \quad (7.10)$$

7.2.2 Applying the ADMM duality theorem to the dual problem

At this step, the ADMM is applied to the dual problem in Eq. (7.10). M Copies of the Lagrangian multipliers \mathbf{v} are introduced to the dual problem to decouple the objective, and new consistency constraints for the multipliers are added to the formulation. This results in:

$$\begin{aligned} &\underset{\mathbf{v}}{\text{maximize}} \quad \sum_{j=1}^M d_j(\mathbf{z}_j) \\ &\text{subject to } \mathbf{v} - \mathbf{z}_j = \mathbf{0} \\ &\text{where } d_j(\mathbf{z}_j) = \inf_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} (f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{z}_j^T \mathbf{S}_j \mathbf{y}_j) \\ &\quad j = 1, \dots, M \end{aligned} \quad (7.11)$$

where \mathbf{z}_j is the copy of \mathbf{v} at the sub-problem j .

Dual consistency constraints $\mathbf{v} - \mathbf{z}_j = \mathbf{0}$ are added to Eq. (7.11) to guarantee \mathbf{v} and its M copies share the same value. Relaxing the dual consistency constraints for the multipliers using Augmented Lagrangian Relaxation and applying ADMM in Eqs. (7.3) – ((7.5) to the problem result in the following procedures to solve the problem in Eq. (7.11).

$$\mathbf{v}^{(k+1)} = \arg \min_{\mathbf{v}} \left\{ \sum_{j=1}^M ((\mathbf{p}_j^{(k)})^T \mathbf{v}) + \frac{\rho}{2} \sum_{j=1}^M \|\mathbf{v} - \mathbf{z}_j^{(k)}\|_2^2 \right\} \quad (7.12)$$

$$\mathbf{z}_j^{(k+1)} = \arg \max_{\mathbf{z}_j} \left\{ d_j(\mathbf{z}_j) + (\mathbf{p}_j^{(k)})^T \mathbf{z}_j - \frac{\rho}{2} \|\mathbf{v}^{(k+1)} - \mathbf{z}_j\|_2^2 \right\} \quad (7.13)$$

$$\mathbf{p}_j^{(k+1)} = \mathbf{p}_j^{(k)} + \rho(\mathbf{v}^{(k+1)} - \mathbf{z}_j^{(k+1)}) \quad (7.14)$$

where \mathbf{p}_j is the vector of multipliers of the dual consistency constraints. The superscript in $(\cdot)^{(k+1)}$ is used to indicate the value of a term at the iteration $(k+1)$. It should be noted that since the terms $(\mathbf{p}_j^{(k)})^T \mathbf{z}^{(k)}$ and $(\mathbf{p}_j^{(k)})^T \mathbf{v}^{(k+1)}$ are constant and are simply added to the objectives, they are dropped from Eq. (7.12) and Eq. (7.13) respectively.

Eq. (7.12) is a quadratic optimization problem with respect to \mathbf{v} , thus it has an analytical solution:

$$\mathbf{v}^{(k+1)} = \frac{1}{M} \sum_{j=1}^M (\mathbf{z}_j) - \frac{1}{M\rho} \sum_{j=1}^M \mathbf{p}_j^{(k)} \quad (7.15)$$

Substituting $d_j(\mathbf{z}_j)$ from Eq. (7.11) into Eq. (7.13), one can write:

$$\begin{aligned} \mathbf{z}_j^{(k+1)} &= \arg \max_{\mathbf{z}_j} \left\{ \inf_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} [f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{z}_j^T \mathbf{S}_j \mathbf{y}_j] + (\mathbf{p}_j^{(k)})^T \mathbf{z}_j - \frac{\rho}{2} \|\mathbf{v}^{(k+1)} - \mathbf{z}_j\|_2^2 \right\} \\ &= \arg \max_{\mathbf{z}_j} \left\{ \inf_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} [f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{z}_j^T (\mathbf{S}_j \mathbf{y}_j + \mathbf{p}_j^{(k)}) - \frac{\rho}{2} \|\mathbf{v}^{(k+1)} - \mathbf{z}_j\|_2^2] \right\} \end{aligned} \quad (7.16)$$

Let $L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)$ represent the objective function in Eq. (7.16)

$$L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j) = f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{z}_j^T (\mathbf{S}_j \mathbf{y}_j + \mathbf{p}_j^{(k)}) - \frac{\rho}{2} \|\mathbf{v}^{(k+1)} - \mathbf{z}_j\|_2^2 \quad (7.17)$$

According to Theorem 37.3 and 37.6 in [83], if $L(\mathbf{y}_j, \mathbf{x}_j, \cdot)$ and $L(\cdot, \cdot, \mathbf{z}_j)$ are convex functions, $L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)$ has a saddle point and satisfies

$$\max_{\mathbf{z}_j} \{ \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} [L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)] \} = \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \{ \max_{\mathbf{z}_j} [L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)] \} \quad (7.18)$$

Thus, the optimization problem in Eq. (7.16) becomes

$$\begin{aligned} & \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \{ \max_{\mathbf{z}_j} [L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)] \} \\ & = \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \{ \max_{\mathbf{z}_j} [f_j(\mathbf{y}_j, \mathbf{x}_j) + \mathbf{z}_j^T (\mathbf{S}_j \mathbf{y}_j + \mathbf{p}_j^{(k)}) - \frac{\rho}{2} \|\mathbf{v}^{(k+1)} - \mathbf{z}_j\|_2^2] \} \end{aligned} \quad (7.19)$$

Since $L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)$ is a quadratic function with respect to \mathbf{z}_j , the solution to $\max_{\mathbf{z}_j} [L(\mathbf{y}_j, \mathbf{x}_j, \mathbf{z}_j)]$ in vector form is

$$\mathbf{z}_j^{(k+1)} = \mathbf{v}^{(k+1)} + \frac{1}{\rho} (\mathbf{p}_j^{(k)} + \mathbf{S}_j \mathbf{y}_j^{(k+1)}) \quad (7.20)$$

or

$$z_{ij}^{(k+1)} = v_i^{(k+1)} + \frac{1}{\rho} (p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i) \quad (7.21)$$

in scalar form, where $\mathbf{z}_j^{(k+1)} = [z_{1j}^{(k+1)}, \dots, z_{ij}^{(k+1)}, \dots, z_{N_j}^{(k+1)}]^T$. Substituting Eq. (7.20) into Eq.

(7.19) the equation becomes:

$$\begin{aligned}
& \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \left\{ f_j(\mathbf{y}_j, \mathbf{x}_j) + [\mathbf{v}^{(k+1)} + \frac{1}{\rho}(\mathbf{p}_j^{(k)} + \mathbf{S}_j \mathbf{y}_j^{(k+1)})]^\top (\mathbf{p}_j^{(k)} + \mathbf{S}_j \mathbf{y}_j^{(k+1)}) \right. \\
& \quad \left. - \frac{\rho}{2} \left\| \frac{1}{\rho} (\mathbf{p}_j^{(k)} + \mathbf{S}_j \mathbf{y}_j^{(k+1)}) \right\|_2^2 \right\} \\
& = \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \left\{ f_j(\mathbf{y}_j, \mathbf{x}_j) + \sum_{i=1}^N v_i^{(k+1)} [p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i] + \frac{1}{\rho} \sum_{i=1}^N [p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i]^2 \right. \\
& \quad \left. - \frac{1}{2\rho} \sum_{i=1}^N [p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i]^2 \right\} \\
& = \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \left\{ f_j(\mathbf{y}_j, \mathbf{x}_j) + \frac{1}{2\rho} \sum_{i=1}^N [p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i]^2 + \sum_{i=1}^N v_i^{(k+1)} [p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i] \right. \\
& \quad \left. + \frac{\rho}{2} \sum_{i=1}^N (v_i^{(k+1)})^2 - \frac{\rho}{2} \sum_{i=1}^N (v_i^{(k+1)})^2 \right\} \\
& = \min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} \left\{ f_j(\mathbf{y}_j, \mathbf{x}_j) + \frac{\rho}{2} \sum_{i=1}^N [v_i^{(k+1)} + \frac{1}{\rho} (p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i)]^2 - \frac{\rho}{2} \sum_{i=1}^N (v_i^{(k+1)})^2 \right\} \tag{7.22}
\end{aligned}$$

where ρ and $v_i^{(k+1)}$ are constants and thus the last term in Eq. (7.22) can be dropped. The optimization problem in Eq. (7.13) is then simplified to:

$$\min_{\mathbf{y}_j, \mathbf{x}_j, \mathbf{g}_j \leq \mathbf{0}, \mathbf{h}_j = \mathbf{0}} f_j(\mathbf{y}_j, \mathbf{x}_j) + \frac{\rho}{2} \sum_{i=1}^N [v_i^{(k+1)} + \frac{1}{\rho} (p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i)]^2 \tag{7.23}$$

with Eq. (7.20) as the solution for $\mathbf{z}_j^{(k+1)}$.

7.2.3 A new parallel coordination algorithm

In summary, the problem in Eq. (7.6) can be solved through the following steps.

The flowchart of this new algorithm is shown in Figure 7.2.

Step 1: Initialization

Step 2: Compute the master copy of multipliers: $\mathbf{v}^{(k+1)} = \frac{1}{M} \sum_{j=1}^M (\mathbf{z}_j) - \frac{1}{M\rho} \sum_{j=1}^M \mathbf{p}_j^{(k)}$ (7.24)

Step 3: For $j = 1, \dots, M$, optimize the sub-problem j :

$$\underset{\mathbf{y}_j, \mathbf{x}_j}{\text{minimize}} \quad f_j(\mathbf{y}_j, \mathbf{x}_j) + \frac{\rho}{2} \sum_{i=1}^P (v_i^{(k+1)} + \frac{1}{\rho} (p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j)_i))^2 \quad (7.25)$$

$$\text{subject to} \quad \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \leq \mathbf{0}$$

$$\mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0}$$

$\mathbf{z}_j^{(k+1)} = (z_{1j}^{(k+1)}, \dots, z_{n_j}^{(k+1)})$ can be determined by the solution $\mathbf{y}_j^{(k+1)}$:

$$z_{ij}^{(k+1)} = v_i^{(k+1)} + \frac{1}{\rho} (p_{ij}^{(k)} + (\mathbf{S}_j \mathbf{y}_j^{(k+1)})_i) \quad (7.26)$$

Step 4: check if the optimization converges, if not, compute the Lagrangian multipliers for multipliers and go to step 2

$$\mathbf{p}_j^{(k+1)} = \mathbf{p}_j^{(k)} + \rho(\mathbf{v}^{(k+1)} - \mathbf{z}_j^{(k+1)}) \quad (7.27)$$

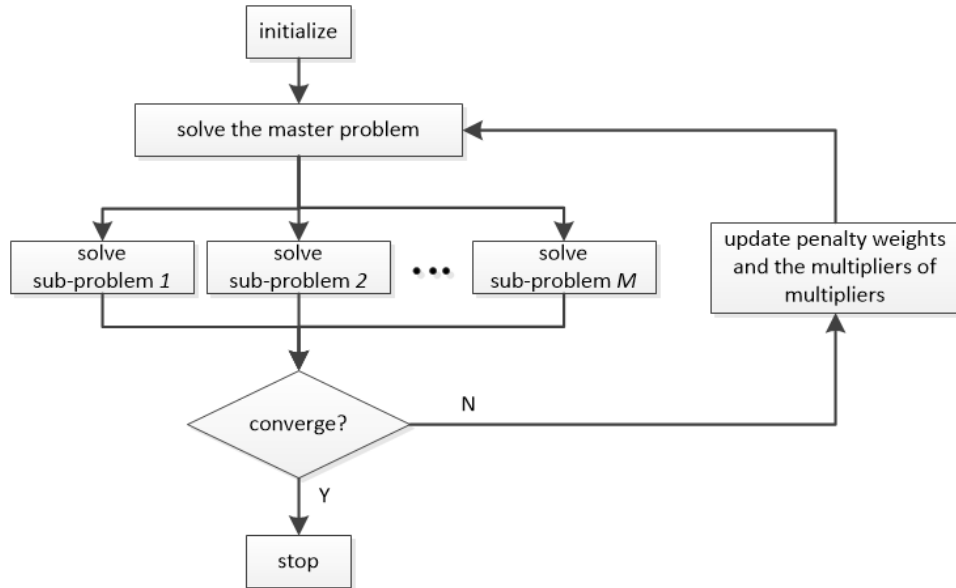


Figure 7.2: The flowchart of the proposed parallel coordination method

During this process, since copies of shared variables and Lagrangian dual multipliers are generated, consistency constraints are introduced to ensure these copies

share the same values. There are two kinds of consistency constraints at iteration k : the primal consistency constraints \mathbf{c}_p and the dual consistency constraints \mathbf{c}_d .

$$\begin{aligned}\mathbf{c}_p^{(k)} &= \mathbf{S}_1 \mathbf{y}_1^{(k)} + \dots + \mathbf{S}_M \mathbf{y}_M^{(k)} = \mathbf{0} \\ \mathbf{c}_d^{(k)} &= \mathbf{v}^{(k)} - \mathbf{z}_j^{(k)} = \mathbf{0}\end{aligned}\tag{7.28}$$

$\mathbf{y}_1, \dots, \mathbf{y}_M$ are shared variables and their copies and \mathbf{c}_p are used to ensure they have the same values at the end of the optimization for the primal problem in Eq. (7.7). \mathbf{v} , $\mathbf{z}_1, \dots, \mathbf{z}_M$ are dual multipliers and their copies and \mathbf{c}_d are used to ensure they have the same values at the end of optimization for the dual problem in Eq. (7.11).

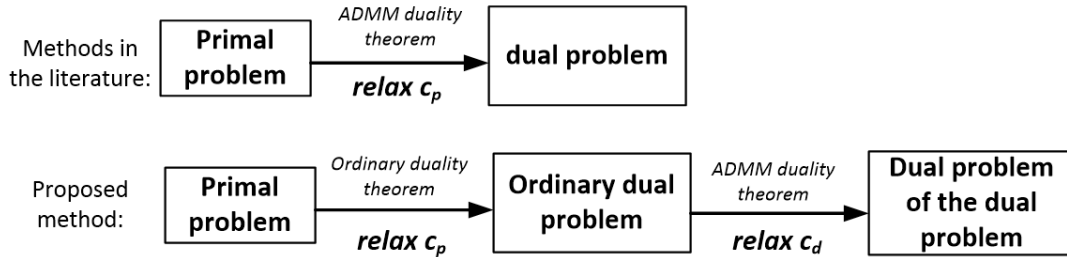


Figure 7.3: Dual transformations in the proposed method and other methods in the literature

Figure 7.3 shows how the duality theorems are applied to the methods in the literature and the proposed method. Most developed methods apply ADMM directly to the primal problem, relax \mathbf{c}_p into the objective and solve the resulting dual problem instead of solving the original problem. In contrast, in the proposed method two kinds of duality theorems are employed. First the ordinary duality theorem is used to relax \mathbf{c}_p and an ordinary dual problem is generated, to which ADMM is then applied to relax \mathbf{c}_d and a dual problem of the first dual problem is formulated. Instead of solving the original problem, the proposed method is solving this dual problem of the dual problem to achieve the optimal solution for the original problem. Since the ordinary duality is

applied initially, the sub-problems in the proposed method are independent from each other and therefore can be solved in parallel. It should be noted that in mathematical programming, the dual problem of the dual problem is supposed to be the primal problem. However, this is not the case here since the two different duality theorems are used in this transformation.

The ultimate goal of the developed algorithm is to solve the primal problem which is difficult to handle directly thus solving the dual of its dual problem is performed. In view of this, as the convergence criteria of the optimization process only the primal consistency constraints \mathbf{c}_p are checked and \mathbf{c}_d is not used. As long as the primal consistency constraints \mathbf{c}_p satisfy certain conditions and are very close to zero, the current solutions can be considered as consistent and feasible for the primal problem, which is the original problem to solve. Thus the convergence criteria for the proposed method are set as

$$\left\| \mathbf{c}_p^{(k)} - \mathbf{c}_p^{(k-1)} \right\|_\infty < \varepsilon \quad \text{and} \quad \left\| \mathbf{c}_p^{(k)} \right\|_\infty < \varepsilon \quad (7.29)$$

where ε is a convergence tolerance, which normally is set to be very small.

The centralized ALC needs M copies of shared variables while the proposed method only needs $M-1$. As a result, the number of elements in \mathbf{c}_p in the proposed method is less than that in the centralized ALC. This is expected to bring certain benefits to the proposed method and is explored in the next section.

7.3 Numerical Tests and Results Analysis

One mathematical problem and one engineering problem are used to test the efficacy of the proposed method. The number of function evaluations is recorded to represent the computational resources consumed. Errors of the design variables and the objective of each problem during the optimization process are also collected. The errors are defined as the relative errors of the decomposition-based optimization results with respect to the reference solution from the All-in-One formulation. All the sub-problem optimizations are solved using “*fmincon*” in Matlab. The centralized ALC-ADMOM is also applied to these test problems for comparison. Through the introduction of an artificial master problem, the centralized ALC-ADMOM allows parallel computation for all decomposed sub-problems and has been proven to be more efficient than other coordination methods [20]. In the following part of this research it is referred to as the centralized ALC for convenience.

7.3.1 Problem 1 – Geometric optimization problem

The first test problem is the simple geometric optimization problem used in Section 5.3.1 Here the problem is partitioned into two sub-problems and both the centralized ALC and the proposed method need a master problem to coordinate the two sub-problems, as shown in Figure 7.4. For the centralized ALC, the linking variables between the master problem and the sub-problems are the shared design variables. In the proposed method, the linking variables are the dual multipliers associated with the shared design variables and their copies. In both methods the master problems are simple quadratic programming problems thus their analytical solutions can be easily calculated.

The solution process of the master problem is merely a parameter update procedure based on the results of the sub-problems.

$$\begin{aligned}
 \min_{z_1, \dots, z_7} \quad & f = z_1^2 + z_2^2 \\
 \text{subject to} \quad & g_1 = (z_3^{-2} + z_4^2)z_5^{-2} - 1 \leq 0, \\
 & g_2 = (z_5^2 + z_6^{-2})z_7^{-2} - 1 \leq 0, \\
 & h_1 = (z_3^2 + z_4^{-2} + z_5^2)z_1^{-2} - 1 = 0, \\
 & h_2 = (z_5^2 + z_6^2 + z_7^2)z_2^{-2} - 1 = 0,
 \end{aligned} \tag{7.30}$$

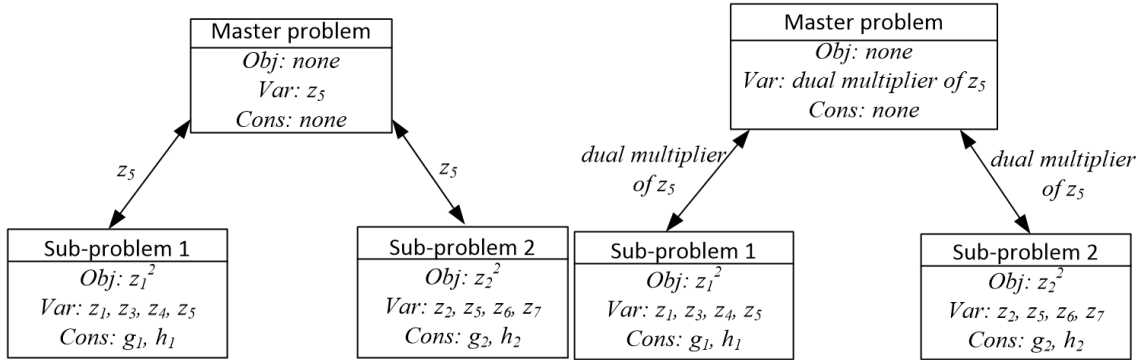


Figure 7.4: The decompositions of the centralized ALC (left) and the proposed method (right) when solving the geometric optimization problem

To explore the effects of the penalty weight ρ on the primal and dual consistency constraints, different values of ρ , 10, 1, 0.1, and 0.01, are considered. The biggest violations of \mathbf{c}_p and \mathbf{c}_d during the optimization process are plotted in Figure 7.5 for different ρ . Here the starting points for the design variables in all cases are set to one and the convergence tolerance ε in Eq. (7.29) is set to 0.001.

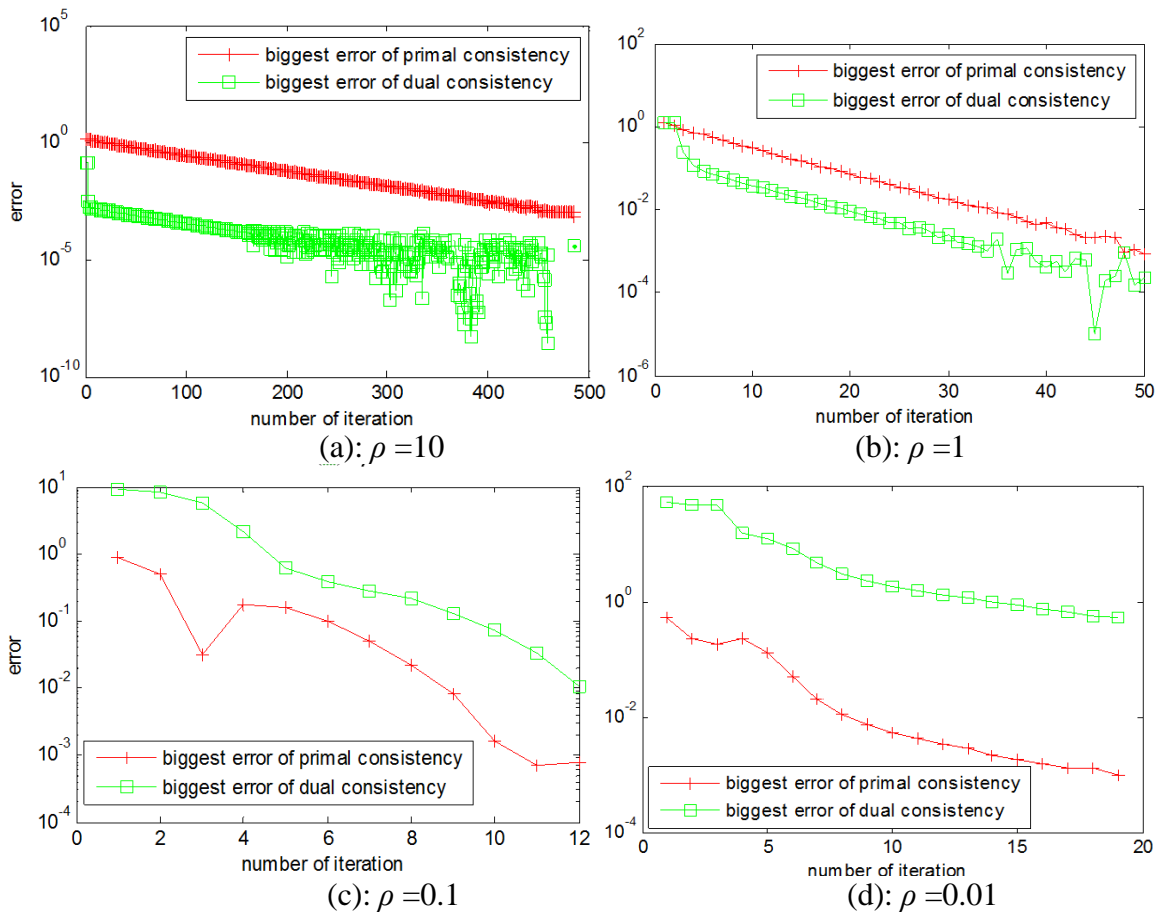


Figure 7.5: The curves of biggest primal and dual consistency errors under different ρ

Figure 7.5 clearly indicates that a large ρ can effectively reduce the dual consistency error, which is reasonable because a large ρ places a high penalty on the differences between the copies of dual multipliers. For $\rho=10$, the dual consistency error reaches around $1e-3$ just after a few iteration while it takes the primal consistency error 487 iterations to reach $1e-3$. As the value of ρ decreases, the primal consistency error drops faster and the dual consistency drops more slowly. For $\rho=0.01$, the dual consistency error is around 0.5 when the primal consistency error reaches $1e-3$.

Inspired by the above observation, it is proposed to decrease the value of ρ throughout the optimization process, instead of fixing it to a constant. Eq. (7.31) is the proposed update scheme for ρ . It is desired to keep ρ relatively big at the beginning to quickly reduce the dual consistency error, and then decrease ρ to reduce the primal consistency error faster thus to accelerate the convergence speed of optimization.

$$\rho^{(k+1)} = \beta\rho^{(k)}, \quad 0 < \beta < 1 \quad (7.31)$$

By adjusting the convergence tolerance in Eq. (7.29), one can control the solution error of decomposition-based optimization. Here, the tolerance ε is set to $1e-2$, $1e-3$, $1e-4$ and $1e-5$ for both the centralized ALC and the proposed method. Their performances are collected in Figure 7.6. The initial ρ in the proposed method and the initial \mathbf{w} in centralized ALC are all set to one. The update parameter β for ρ in Eq. (7.13) is 0.8. For the centralized ALC, the weight update scheme in [20] is used and $\beta = 1.1$ and $\Upsilon = 0.9$, as suggested in [20]. The initial Lagrangian multipliers are set to zero, and the initial design variables are set to one. Figure 7.6 shows that the proposed method is more efficient than the centralized ALC in terms of number of function evaluations.

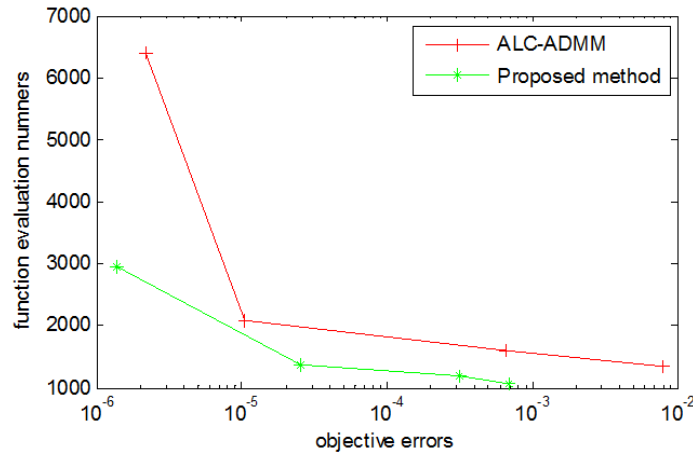


Figure 7.6: The test results of the centralized ALC and the proposed method on the geometric optimization problem (From right to left: $\varepsilon = 1e-2$, $1e-3$, $1e-4$ and $1e-5$)

7.3.2 Problem 2 – Portal frame design problem

The second test problem is a structural optimization problem – the portal frame design problem [84][85][86][87]. The optimization objective is to minimize the volume of the whole structure by changing the dimensions of the cross sectional areas. As shown in Figure 7.7, a horizontal force and a concentrated moment are applied on the structure, which cause normal and shear stresses. The design constraints are to satisfy all stress limits and geometry requirements. The values of stress are computed through Finite Element Analysis (FEA). Readers may find more details about this problem in [59].

$$\begin{aligned}
 & \min_{\mathbf{z}} V(\mathbf{z}) \\
 & \text{subject to} \\
 & \mathbf{g}_{1,i,j,k}(\mathbf{z}) = \frac{\sigma_{jk}^{[i]}}{\sigma_y} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \quad j = 1, 2 \quad k = 1, 2 \\
 & \mathbf{g}_{2,i,j,k}(\mathbf{z}) = -\frac{\sigma_{jk}^{[i]}}{\sigma_y} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \quad j = 1, 2 \quad k = 1, 2 \\
 & \mathbf{g}_{3,i,j}(\mathbf{z}) = \frac{\tau_j^{[i]}}{\tau_y} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \quad j = 1, 2 \\
 & \mathbf{g}_{4,i,j}(\mathbf{z}) = -\frac{\tau_j^{[i]}}{\tau_y} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \quad j = 1, 2 \\
 & \mathbf{g}_{5,i}(\mathbf{z}^{[i]}) = \frac{h^{[i]} - t_1^{[i]} - t_2^{[i]}}{35d^{[i]}} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \\
 & \mathbf{g}_{6,i,j}(\mathbf{z}^{[i]}) = \frac{w_j^{[i]}}{20t_j^{[i]}} - \mathbf{I} \leq \mathbf{0} \quad i = 1, 2, 3 \quad j = 1, 2 \\
 & \mathbf{g}_{7,i}(\mathbf{z}^{[i]}) = \mathbf{I} - \frac{5(h^{[i]} - t_1^{[i]} - t_2^{[i]})d^{[i]}}{A^{[i]}} \leq \mathbf{0} \quad i = 1, 2, 3 \\
 & \mathbf{g}_{8,i}(\mathbf{z}^{[i]}) = \left(\mathbf{I} - \frac{t_1^{[i]}w_1^{[i]}}{2t_2^{[i]}w_2^{[i]}}\right)\left(\mathbf{I} - \frac{t_2^{[i]}w_2^{[i]}}{2t_1^{[i]}w_1^{[i]}}\right) \leq \mathbf{0} \quad i = 1, 2, 3 \\
 & \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}
 \end{aligned} \tag{7.32}$$

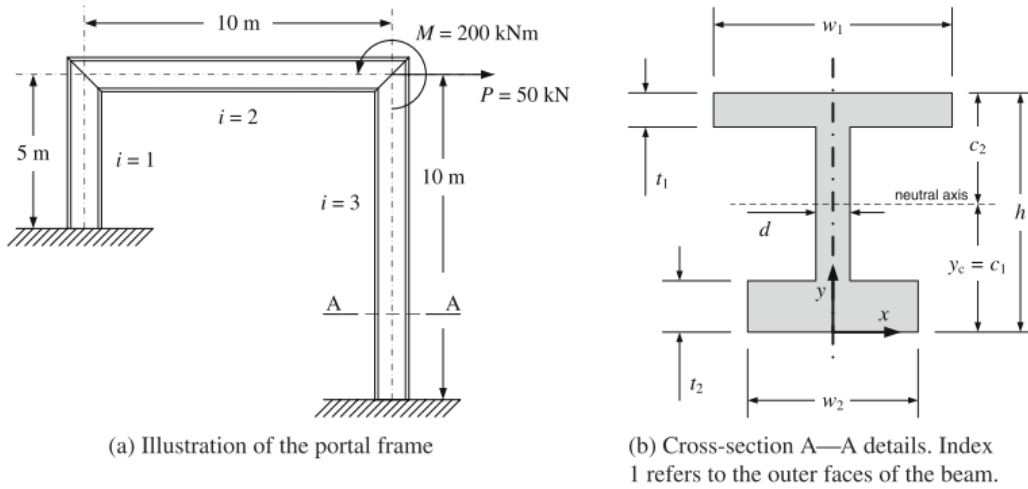


Figure 7.7: Structure and cross sectional area of the portal frame design problem[59]

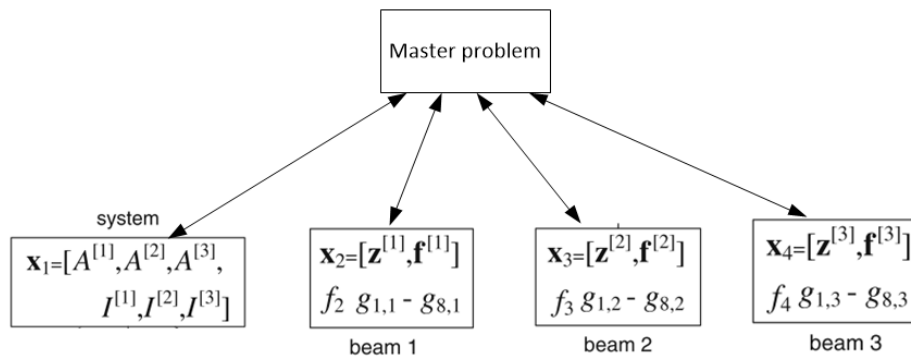


Figure 7.8: The decompositions of the proposed method and the centralized ALC when solving the portal frame design problem

Due to the uncertainty of one set of the geometry constraints in this problem (which requires that either the top flange area is twice as large as the bottom one or vice versa), this problem has many local optimal solutions and one global optimal solution. Previous research has found that ALC with a hierarchical structure could reach the global solution 54 times out of 100 tries [59]. However the ALC in [20] cannot solve the sub-problems in parallel. For this problem, the centralized ALC in [24] is adopted and

compared to the method proposed in this chapter. Both methods use the structure shown in Figure 7.8.

20 tests are performed for the two methods, starting from random points between the lower bounds and upper bounds of the design variables. The objective errors of these test results are collected in Figure 7.9. The convergence tolerance ε is set to $1e-3$ for both the centralized ALC and the proposed method. For the proposed method, the initial ρ is 1 and β is 0.96. For the centralized ALC, a large initial \mathbf{w} such as 1 or 0.1 does not work and always causes the optimization to converge prematurely, so the initial \mathbf{w} is set to 0.01 and the update parameters are set as $\beta = 1.05$ and $\Upsilon = 0.9$. All initial Lagrangian multipliers are set to 0. The AIO optimal objective value is 0.1661, which is used as the global optimal objective value to calibrate the objective error for decomposition-based optimization.

For the centralized ALC, the objective errors for the 20 tests in Figure 7.9 range from 0.53% to 9.95% with an average value of 3.5%. While for the proposed method, all 20 tests in Figure 7.9 reach the same objective value, 0.1660, with the objective error as low as 0.1%, which means that the proposed method has greatly improved the solution accuracy and has high robustness. On the efficiency side, the average number of function evaluations of the proposed method is 181,421 while that number for the centralized ALC is 368,028.

In summary, for this multi-modal engineering problem, starting from random design variables, the proposed method consumes only half of the computational resources of the centralized ALC while it reaches a much better solution than the centralized ALC

does. Also, the results of the centralized ALC vary a lot as the starting points change while the proposed method can consistently reach an accurate solution practically independently of the starting point.

The numerical tests in this study have shown the high efficiency and accuracy of the proposed method over the centralized ALC. These advantages might stem from the major fact that the proposed method creates one less set of copies of all shared variables than the centralized ALC, which makes it easier for the algorithm to coordinate and reach a consistent solution. While in the proposed method copies of dual multipliers are introduced, these copies do not necessarily need to be equal to each other for the algorithm to generate the optimal solution for the original problem. For example, the portal frame design problem originally has 24 shared variables. In the proposed method, 24 duplicated shared variables are introduced and thus 24 primal consistency constraints are added to the decomposition formulation. In the centralized ALC, the number of duplicated shared variables is 48 because each original shared variable requires two copies. As a result, the number of primal consistency constraint increases to 48. The convergence criterion for both methods is the closeness of the primal consistency error to zero, and it is apparently much easier to satisfy this condition with 24 rather than 48 primal consistency constraints.

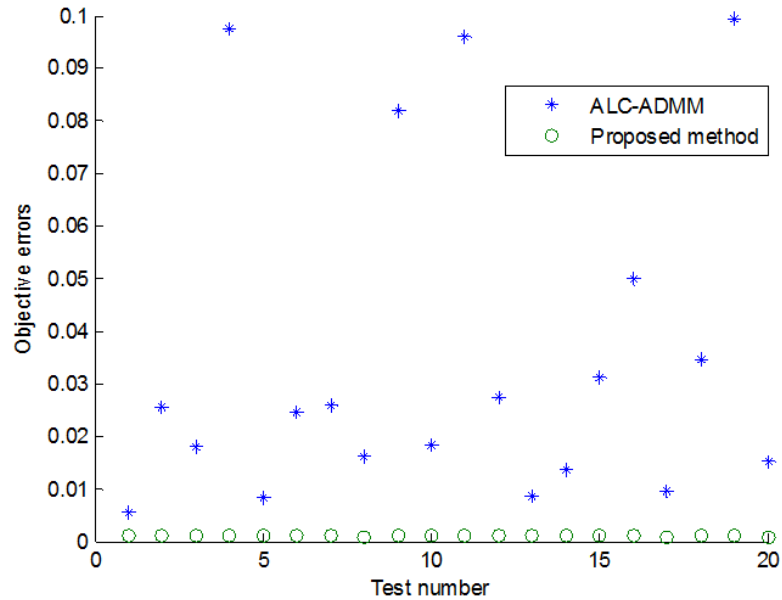


Figure 7.9: The objective errors of ALC-ADMM and the proposed method for 20 tests of the portal frame problem

7.4 Summary

A new parallel coordination method for decomposition-based optimization is proposed in this chapter. Instead of applying ADMM directly to the primal problem, the ordinary duality theorem is first employed to generate an ordinary dual problem, and then ADMM is applied to the dual problem. The resulting formulation is the dual problem of the dual problem, in which all sub-problems are independent from each other and thus can be solved in parallel. The mathematical derivations of the method are presented in the chapter along with its final computation procedures.

One mathematical and one engineering example are used to test the proposed method and the results are compared to those of another popular parallel coordination method: the centralized ALC. The results show that for the mathematical problem, the proposed method consumes less function evaluations than the centralized ALC does

while reaching the same objective accuracy. For the complex multi-modal structural optimization problem, the proposed method is more accurate (with an average objective error as 0.1% compared to 3.5% for the centralized ALC), efficient (needs only half of the number of function evaluations of the centralized ALC), and robust (consistently reaches the same accurate solution while the solutions of the centralized ALC vary when starting from random initial points).

Chapter 8

Concluding Remarks

Decomposition based optimization is a subject relevant to both mathematical programming and engineering optimal design. The motivation of the subject comes from engineering as the optimal design task of modern products is impossible and undesirable to be carried out by one single person or even one design team. The modeling and coordination of decomposed sub-problems rely heavily on mathematical theories of nonlinear programming. In decomposition based optimization, the engineering design tasks are represented as optimization problems with explicit objectives and constraints. The original design problem before decomposition is referred to as the All-in-One (AIO) problem, and the partitioned smaller problems are called sub-problems. The couplings between sub-problems are treated as linking variables, and the decoupling process of these sub-problems are realized by introducing copies of the linking variables and relaxing the newly added equality consistency constraints.

8.1 Contributions

Novel application of existing mathematical theorems and proposal of new theorems have always been the driving force for decomposition based optimization methods. One typical example is the evolutionary course from ATC to ALC, which was

achieved by switching from quadratic penalty relaxation method to augmented Lagrangian relaxation method. Another example is the huge computational cost reduction attained by ALC-ADMOM when the ADMM duality theorem is employed to replace the block coordinate descent method in ALC-ENMOM. Therefore, this research first tries to advance decomposition based optimization from the mathematical perspective and achieves the following contributions:

- For the ALC method, the KKT necessary optimality conditions are applied to the formulations of AIO and the decomposed problems, the terms “primal residual” and “dual residual” are introduced into ALC, and a new update strategy considering both residuals and thus guaranteeing the unmet optimality condition in the traditional update is introduced. Numerical tests show a decrease in the number of iterations and significant improvements in solution accuracy with both calculated and fine-tuned initial weights using the new update. Additionally, the proposed approach is capable to start from a wide range of possible weights and achieve optimality, and therefore brings robustness to ALC.
- Based on the alternating direction method of multipliers (ADMM), a new parallel coordination method with high efficiency is proposed. Instead of applying ADMM directly to the primal problem, the ordinary duality theorem is first employed to generate an ordinary dual problem, and then the ADMM is applied to the dual problem. The resulting formulation is the dual problem of the dual problem, in which all sub-problems are independent from each other thus can be solved in parallel. Numerical tests are conducted on both mathematical and

engineering problems and the results show an increase in efficiency and accuracy for the new method when compared to the centralized ALC, which is one of the most popular parallel coordination methods. Additionally, this increase in performance is consistently displayed by the new method when solving a multimodal structural optimization problem repeatedly starting from different random initial designs, while the centralized ALC fails to show similar robustness.

The motivation of the development of decomposition based optimization is to solve engineering problems. The second part of the contributions of this research is accomplished from the engineering perspective and is related to the implementation of decomposition based optimization methods.

- By exploring the flexibility of decomposition of the micro-accelerometer benchmark problem, two kinds of hybrid network decompositions are proposed. One is adding a new discipline to a component-decomposed problem; the other one is adding a new component to a discipline-decomposed problem. CADMM is employed to solve these two decompositions. Numerical experiments show that the optimization results of CADMM are very close to the reference optimal solution (with an error less than 3%). This demonstrates that CADMM is able to deal with the hybrid network decomposition problem and supports component-discipline decomposition and sub-system optimization to solve the overall problem.
- For a fully coupled network-decomposed problem, several alternative ALC structures are proposed and discussed. Both mathematical and engineering test

problems are used to experimentally explore the performance of different solving structures. The results show that under the same partition and using the same coordination method - ALC, the performance of the decomposition-based optimization may be largely different in terms of efficiency, accuracy and computational resource cost. The results highlight the importance of choosing an optimal solving structure in the implementation of ALC. Several suggestions for guidelines on the selection of an optimal solving structure selection are proposed. While this study is conducted using ALC, the findings are potentially applicable to other NTC methods based on duality theorems.

8.2 Future Research

In this dissertation, the proposed dual residual and the new weight update significantly increase the accuracy and robustness of ALC. The new coordination method based on two duality theorems enables the parallel computation of sub-problems, which increases the efficiency of decomposition based optimization. In the future, more studies need to be conducted to improve these theories and methods.

- The applicability of the proposed theory on dual residual to a broader category of problems beyond quasi-separable problems (such as problems with system- wide constraints) needs to be studied. Also, it is worth investigating whether this theory can improve the consensus optimization method which also employs the augmented Lagrangian relaxation to coordinate decomposed sub-problems.

- For the new flexible weight update, it is necessary to study the reasons of the oscillations in the objective error, dual residual, and primal residual, as these oscillations consume more than half of the overall iterations in several cases. The effects of the update parameters (μ and τ) on the performance of the proposed methods need to be explored.
- For the new parallel coordination method, more numerical tests need to be conducted to compare the proposed method with other parallel coordination methods. It is also worthwhile to investigate how the penalty weight ρ affects the optimization process, and come up with an improved update scheme for ρ .
- For the practical application of the proposed research, more large scale, realistic, complex models for engineering design optimization problems need to be developed to test the proposed methods.

Bibliography

- [1] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, 11(3): 399-417, 1963.
- [2] L. S. Lasdon, "Duality and decomposition in mathematical programming," *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 86-100, 1968.
- [3] J. T. Allison, "Optimal Partitioning and Coordination Decisions in Decomposition-based Design Optimization," Doctoral Dissertation, The University of Michigan, 2008.
- [4] T. C. Wagner and P. Y. Papalambros, "A general framework for decomposition analysis in optimal design," *Advances in Design automation*, 65(2): 315–325, 1993.
- [5] E. J. Cramer, J. E. Dennis Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, "Problem formulation for multidisciplinary optimization," *SIAM Journal of Optimization*, 4(4): 754 – 776, 1994.
- [6] J. Sobieszczanski-Sobieski and R. T. Haftka, "Multidisciplinary aerospace design optimization : survey of recent developments," *Structural Optimization*, 14(1):1–23, 1997.
- [7] J. P. Giesing and J. M. Barthelemy, "A summary of industry MDO applications and needs," *AIAA white paper*, 1998.
- [8] A. Kusak and N. Larson, "Decomposition and representation methods in mechanical design," *Journal of Vibration and Acoustics*, 117(B): 17–24, 1995.
- [9] L. Chen, Z. Ding, and S. Li, "A formal two-phase method for decomposition of complex design problems," *Journal of Mechanical Design*, 127(2): 184-195, 2005.
- [10] M. Dragan, "On decomposing large sparse systems of nonlinear equations," *In the proceedings of the 4th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania, October 9-12, 2002.
- [11] R. J. Balling and J. Sobieszczanski-Sobieski. "Optimization of coupled systems-a critical overview of approaches." *AIAA journal*, 34(1): 6-17, 1996.

- [12] M. Kokkolaras, L. Louca, G. Delagrammatikas, N. Michelena, Z. Filipi, P. Papalambros, J. Stein, and D. Assanis. "Simulation-based optimal design of heavy trucks by model-based decomposition: An extensive analytical target cascading case study." *International Journal of Heavy Vehicle Systems*, 11(3): 403-433, 2004.
- [13] M. Kaufman, V. Balabanov, A. A. Giunta, B. Grossman, W. H. Mason, S. L. Burgee, R. T. Haftka, and L. T. Watson. "Variable-complexity response surface approximations for wing structural weight in HSCT design." *Computational Mechanics*, 18(2): 112-126, 1996.
- [14] J. T. Allison, M. Kokkolaras, and P. Y. Papalambros, "Optimal partitioning and coordination decisions in decomposition-based design optimization," *Journal of Mechanical Design*, 131(8): 081008, 2009.
- [15] N. F. Michelena and P. Y. Papalambros, "A hypergraph framework for optimal model-based decomposition of design problems," *Computational optimization and applications*, 8(2): 173-196. 1997.
- [16] P. T. Lin and H. C. Gea, "A gradient-based transformation method in multidisciplinary design optimization," *Structural and Multidisciplinary Optimization*, 47(5): 715-733, 2013.
- [17] S. Li and L. Chen, "Model-based decomposition using non-binary dependency analysis and heuristic partitioning analysis," *In ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Philadelphia, PY, 2006.
- [18] H. Kim, "Target Cascading in Optimal System Design," Doctoral Dissertation, The University of Michigan, 2001.
- [19] N. Michelena, H. M. Kim, and P. Papalambros, "A system partitioning and optimization approach to target cascading," *In Proceedings of the 12th International Conference on Engineering Design*, 2: 1109-1112, 1999.
- [20] S. Tosserams, "Distributed Optimization for Systems Design An Augmented Lagrangian Coordination Method," Doctoral Dissertation, Eindhoven niversity of Technology, 2008.
- [21] M. Xu, W. Wang, P. Guarneri, and G. Fadel, "CADMM applied to hybrid network decomposition," *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, May 19-24, 2013.
- [22] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang, "Target cascading in optimal system design," *Journal of Mechanical Design*, 125(3): 474-480, 2003.

- [23] N. Michelena, H. Park, and P. Y. Papalambros, "Convergence properties of Analytical Target Cascading," *AIAA journal*, 41(5): 897-905, 2003.
- [24] S. Tosserams, M. Kokkolaras, L. F. P. Etman, and J. E. Rooda, "A Nonhierarchical formulation of analytical target cascading," *Journal of Mechanical Design*, 132(5): 051002, 2010.
- [25] W. Wang, "Network Target Coordination for Design Optimization of Decomposed Systems," Doctoral Dissertaion, Clemson University, 2012.
- [26] W. Wang, M. Xu, P. Guarneri, G. Fadel, and V. Blouin, "'A consensus optimization via alternating direction method of multipliers for network target coordination,' In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, 2012.
- [27] W. Wang, P. Guarneri, G. Fadel, and V. Blouin., "Network target coordination for optimal design of decomposed systems with consensus optimization," In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, IL, USA, 2012.
- [28] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd edition, John Wiley & Sons, New York, NY, 2006.
- [29] D. P. Bertsekas, *Nonlinear programming*. 2nd edition, Athena Scientific, Belmont, MA, 2003.
- [30] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [31] M. Fortin and R. Glowinski, *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. North-Holland, Amsterdam, Netherlands, 1983.
- [32] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization." *Lecture Notes in Computer Science*, 288–300, 2002.
- [33] R. D. Braun, "Collaborative Optimization: An Architecture For Large-Scale Distributed Design," Doctoral Dissertation, Stanford University, 1996.
- [34] R. D. Braun, A. A. Moore, and I. M. Kroo, "Collaborative approach to launch vehicle design," *Journal of Spacecraft and Rockets*, 34(4): 478–486, 1997.

- [35] J. Sobieszczanski-Sobieski, "Optimization by decomposition: a step from hierarchic to non-hierarchic systems," *In Proceedings of the 2nd NASA Air Force Symposium on Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA. NASA CP-3031, 1988.
- [36] J. Sobieszczanski-sobieski, J. S. Agte, and R. R. Sandusky, "Bi-level integrated system synthesis (BLISS)," *AIAA Journal*, 38(1): 164–172, 2000.
- [37] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization : a survey of architectures," *AIAA Journal*, 51(9):2049-2075, 2013.
- [38] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "A classification of methods for distributed system optimization based on formulation structure," *Structural and Multidisciplinary Optimization*, 39(5):503–517, 2009.
- [39] N. M. Alexandrov and R. M. Lewis, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," *AIAA Journal*, 40(2): 301–309, 2002.
- [40] J. G. Lin, "Analysis and enhancement of collaborative optimization for multidisciplinary design," *AIAA Journal*, 42(2):348–360, 2004.
- [41] V. DeMiguel and W. Murray, "A local convergence analysis of bilevel decomposition algorithms," *Optimization and Engineering*, 7(2): 99–133, 2006.
- [42] H. M. Kim, M. Kokkolaras, L. S. Louca, G. J. Delagrammatikas, N. F. Michelena, Z. S. Filipi, P. Y. Papalambros, J. L. Stein, and D. N. Assanis, "Target cascading in vehicle redesign class vi truck study," *International Journal of Vehicle Design*, 29(3): 199-225, 2002.
- [43] H. M. Kim, D. G. Rideout, P. Y. Papalambros, and J. L. Stein, "Analytical target cascading in automotive vehicle design," *Journal of Mechanical Design*, 125(3):481-489, 2003.
- [44] J. J. Michalek and P. Y. Papalambros, "BB-ATC: Analytical target cascading using branch and bound for mixed-integer nonlinear programming," *In Proceedings of the ASME Design Engineering Technical Conferences*, Philadelphia, PA, 2006.
- [45] M. Kokkolaras, Z. P. Mourelatos, and P. Y. Papalambros, "Design optimization of hierarchically decomposed multilevel systems under uncertainty," *Journal of Mechanical Design*, 128(2): 503–508. 2006.

- [46] J. Han and P. Y. Papalambros, “An SLP filter algorithm for probabilistic analytical target cascading,” *Structural and Multidisciplinary Optimization*, 41(6), 935-945, 2010.
- [47] J. Han and P. Papalambros, “A sequential linear programming coordination algorithm for analytical target cascading,” *Journal of Mechanical Design*, 132(2): 021003. 2010.
- [48] M. Gardenghi, M. M. Wiecek, and W. Wang, “Biobjective optimization for analytical target cascading: optimality vs. achievability,” *Structural and Multidisciplinary Optimization*, 47(1): 111-133, 2012.
- [49] N. Tzevelekos, M. Kokkolaras, P. Y. Papalambros, M. F. Hulshof, L. F. P. Etman, and J. E. K. Rooda, “An empirical local convergence study of alternative coordination schemes in analytical target cascading,” *In Proceedings of the 5th world congress on structural and multidisciplinary optimization*, 2003.
- [50] J. J. Michalek and P. Y. Papalambros, “An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading,” *Journal of Mechanical Design*, 127(2):206-214, 2005.
- [51] S. Tosserams, “Analytical Target Cascading Convergence Improvement By Sub-Problem Post-Optimality Sensitivities,” Master’s thesis, Eindhoven University of Technology, 2004.
- [52] J. B. Lassiter, M. M. Wiecek, and K. R. Andrighetti, “Lagrangian coordination and Analytical Target Cascading: solving ATC-decomposed problems with Lagrangian duality,” *Optimization and Engineering*, 6(3): 361-381, 2005.
- [53] P. Guarneri, J. T. Leverenz, M. M. Wiecek, and G. Fadel, “Optimization of nonhierarchically decomposed problems,” *Journal of Computational and Applied Mathematics*, 246:312-319, 2013.
- [54] Y. Li, Z. Lu, and J. J. Michalek, “Diagonal quadratic approximation for parallelization of analytical target cascading,” *Journal of Mechanical Design*, 130(5): 051402, 2008.
- [55] S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda, “An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers,” *Structural and Multidisciplinary Optimization*, 31(3): 176-189. 2006.

- [56] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "An augmented Lagrangian decomposition method for quasi-separable problems in MDO," *Structural and Multidisciplinary Optimization*, 34(3): 211-227, 2007.
- [57] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "Augmented Lagrangian coordination for distributed optimal design in MDO," *International Journal for Numerical Methods in Engineering*, 73(13): 1885-1910, 2008.
- [58] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "Block-separable linking constraints in augmented Lagrangian coordination," *Structural and Multidisciplinary Optimization*, 37(5): 521-527, 2008.
- [59] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "Multi-modality in augmented Lagrangian coordination for distributed optimal design," *Structural and Multidisciplinary Optimization*, 40(1-6): 329-352, 2009.
- [60] S. Tosserams, L. F. P. Etman, and J. E. Rooda, "A micro-accelerometer MDO benchmark problem," *Structural and Multidisciplinary Optimization*, 41(2): 255-275, 2010.
- [61] S. DorMohammadi and M. Rais-Rohani, "Exponential penalty function formulation for multilevel optimization using the analytical target cascading framework," *Structural and Multidisciplinary Optimization*, 47(4): 599-612, 2013.
- [62] W. Wang, V. Y. Blouin, M. K. Gardenghi, M. M. Wiecek, G. M. Fadel, and B. Sloop, "A cutting plane method for analytical target cascading with augmented Lagrangian coordination," *Journal of Mechanical Design*, 135(10): 104502, 2013.
- [63] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York, NY, 1982.
- [64] J. S. Arora, A. I. Chahande, and J. K. Paeng, "Multiplier methods for engineering optimization," *International Journal for Numerical Methods in Engineering*, 32(7): 1485-1525, 1991.
- [65] M. Xu, G. Fadel, and M. M. Wiecek, "Dual residual in augmented Lagrangian coordination for decomposition-based optimization," *In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Buffalo, NY, 2014.
- [66] Y. Zhou, "Layout Synthesis of Accelerometers Master of Science Project Report Committee," MS Project Report, Carnegie Mellon University, 1998.

- [67] T. Mukherjee, Y. Zhou, and G. K. Fedder, "Automated optimal synthesis of microaccelerometers," *In Proceeding of the 12th Annual IEEE InternationalMicro ElectroMechanical System Conference*, Orlando, FL, USA, 1999.
- [68] MATLAB 2012a - version 7.10.0. Natick, Massachusetts, the MathWorks Inc., 2012.
- [69] M. Xu, G. Fadel, and M. M. Wiecek, "Solving structure for network-decomposed problems optimizaed with augmented Lagrangian coordination," *In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* , Buffalo, NY, 2014.
- [70] J. J. Michalek and P. Y. Papalambros, "An efficient weighting update method to achieve acceptable consistency deviation in Analytical Target Cascading," *Journal of Mechanical Design*, 127(2): 206-214, 2005.
- [71] J. Han and P. Y. Papalambros, "A note on the convergence of analytical target cascading with infinite norms," *Journal of Mechanical Design*, 132(3): 034502, 2010.
- [72] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, 3(1): 1-122, 2010.
- [73] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, 106(2): 337-356, 2000.
- [74] S. L. Wang and L. Z. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, 109(2): 415-429, 2001.
- [75] M. Xu, G. Fadel, and M. M. Wiecek, "Dual residual for centralized augmented Lagrangian coordination based on optimality conditions," *Journal of Mechanical Design*, 137(6): 061401, 2015.
- [76] J. Golinski, "Optimal synthesis problems solved by means of nonlinear programming and random methods," *Journal of Mechanisms*, 5(3): 287-309, 1970.
- [77] S. L. Padula, N. Alexandrov, and L. L. Green, "MDO test suite at NASA Langley Research Center," *AIAA Paper*, 96-4028, 1996.

- [78] M. Xu, G. Fadel, and M. M. Wiecek, "Dual residual for distributed ALC based on optimality conditions," *Accepted by ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Boston, MA, 2015.
- [79] H. M. Kim, W. Chen, and M. M. Wiecek, "Lagrangian coordination for enhancing the convergence of analytical target cascading," *AIAA journal*, 44(10): 2197-2207, 2006.
- [80] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, 1(1): 93-111, 1992.
- [81] M. Xu, G. Fadel, M. M. Wiecek, and P. Guarneri, "An efficient parallel coordination method for decomposition-based optimization using two duality theorems," *Accepted by 11th World Congress on Structural and Multidisciplinary Optimization*, Sydney, Australia, June 7-12, 2015.
- [82] J. J. Michalek and P. Y. Papalambros, "Weights, norms, and notation in analytical target cascading," *Journal of Mechanical Design*, 127(3): 499-501, 2005.
- [83] R. T. Rockafellar, *Convex analysis*. Princeton university press, Princeton, NJ, 1970.
- [84] J. Sobieszczanski-Sobieski, B. B. James, and A. R. Dovi, "Structural optimization by multilevel decomposition," *AIAA journal*, 23(11): 1775-1782, 1985.
- [85] G. N. Vanderplaats, Y. J. Yang, and D. S. Kim, "Sequential linearization method for multilevel optimization," *AIAA journal*, 28(2): 290-295, 1990.
- [86] M. A. Arslan and P. Hajela, "Counterpropagation neural networks in decomposition based optimal design," *Computers & Structures*, 65(5): 641-650, 1997.
- [87] B. Liu, R. T. Haftka, and L. T. Watson, "Global-local structural optimization using response surfaces of local optimization margins," *Structural and Multidisciplinary Optimization*, 27(5): 352-359, 2004.