12-2012

Network Target Coordination for Design Optimization of Decomposed Systems

Wenshan Wang
Clemson University, wenshaw@clemson.edu

Network Target Coordination for Design Optimization of Decomposed Systems

---

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
Wenshan Wang
December 2012

Accepted by:
Dr. Georges Fadel, Committee Chair
Dr. Vincent Blouin, Committee Co-Chair
Dr. Gang Li
Dr. Lonny Thompson
Dr. Margaret Wiecek

Abstract

A complex engineered system is often decomposed into a number of different subsystems that interact on one another and together produce results not obtainable by the subsystems alone. Effective coordination of the interdependencies shared among these subsystems is critical to fulfill the stakeholder expectations and technical requirements of the original system. The past research has shown that various coordination methods obtain different solution accuracies and exhibit different computational efficiencies when solving a decomposed system. Addressing these coordination decisions may lead to improved complex system design. This dissertation studies coordination methods through two types of decomposition structures, hierarchical, and nonhierarchical.

For coordinating hierarchically decomposed systems, linear and proximal cutting plane methods are applied based on augmented Lagrangian relaxation and analytical target cascading (ATC). Three nonconvex, nonlinear design problems are used to verify the numerical performance of the proposed coordination method and the obtained results are compared to traditional update schemes of subgradient-based algorithm. The results suggest that the cutting plane methods can significantly improve the solution accuracy and computational efficiency of the hierarchically decomposed systems. In addition, a biobjective optimization method is also used to capture optimality and feasibility. The numerical performance of the biobjective algorithm is verified by solving an analytical mass allocation problem.

For coordinating nonhierarchically decomposed complex systems, network target coordination (NTC) is developed by modeling the distributed subsystems as different

agents in a network. To realize parallel computing of the subsystems, NTC via a consensus alternating direction method of multipliers is applied to eliminate the use of the master problem, which is required by most distributed coordination methods. In NTC, the consensus is computed using a locally update scheme, providing the potential to realize an asynchronous solution process. The numerical performance of NTC is verified using a geometrical programming problem and two engineering problems.

To my family for their unwavering support and encouragement over the years.

Acknowledgments

It is now the right time to think about these long yet short, intensive yet fulfilling years during my PhD study at Clemson University. Working with Prof. Georges Fadel and Prof. Vincent Blouin was a dream-come-true for me. And, the longer I have participated in the research group, the more I have enjoyed it. I will certainly miss the wealth of insights and new ideas that Prof. Fadel and Prof. Blouin shared during our weekly meetings. My papers and presentations have benefited from their critiques, and the contents of this dissertation owes much to their guidance and support.

In addition to the opportunity to work for Prof. Fadel and Prof. Blouin, I am also very grateful to Prof. Margaret Wiecek. The last year, in particular, has been a terrific experience. Prof. Margaret Wiecek has provided excellent feedback on the mathematical details of my dissertation. She has also been a great source of encouragement and guidance to the analysis in my dissertation. Prof. Wiecek and I shared occasional weeks of intense, productive discussions that have shaped the direction of my dissertation.

I would also like to recognize long-term members of the research group, including Paolo Guarneri, James Gibert, Chris Czech, Qing Mao, Meng Xu, Jun Hu, Felipe Fernandez, and Ivan Mata, for their fruitful discussions as I prepared for my defense. The research group has certainly benefited from their energy and enthusiasm. Many thanks to all for making my last few months at Clemson particularly enjoyable.

Prof. Gang Li and Prof. Lonny Thompson deserve special recognition for their role as dissertation reading committees. Their constructive feedback and suggestions guided me to focus on specific, challenging and significant research problems in the last year of

my PhD. I am grateful to Barbara Ramirez for her cover-to-cover reading of my dissertation. This dissertation has certainly benefited from her careful review and thoughtful comments.

Many others have contributed to my happiness and success at Clemson University. The administration staff in the Department of Mechanical Engineering, in particular, Gwen Dockins, Garrett Teri, Tameka Boyce, Carol Johnson, and Kathryn Poole deserve credit for guiding me through the mounds of paperwork associated with completing a degree at Clemson University. My friends, in particular, Jinying Chen, Yijie Jiang, Meng Zhang, Xiaozhen Yu, Jinxiang Zhou, Ziyi Hu, Jing Peng, and Jian Wei, provide a source of inspiration and help broaden my knowledge.

The National Science Foundation (NSF) and Automotive Research Center (ARC, a center of excellence of the US Army TACOM) are gratefully acknowledged for their funding support. My studies at Clemson would not have been possible without their graduate research assistantships.

Most importantly, I would like to thank my parents for their support throughout my educational journey. Their constant love and unwavering support accompanied me all through the way from a young girl to a Doctor of Philosophy, making the process full of warmth, encouragement, and happiness.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Optimization and Basic Definitions

Optimization[1], alternatively, mathematical programming, is the science of the best in the sense that it helps designers to make not only a reasonable decision but also the best one subject to certain constraints describing its domain. A very common instance of a constrained optimization problem involves finding the minimum weight of a structure subjected to constraints on stress and deflection.

Mathematical programming models provide appropriate tools for addressing these optimization decision variables in precisely and formally. A mathematical programming problem is formulated as an objective $f(\mathbf{x})$ to be minimized or maximized with respect to a column vector of n real valued decision variables, $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$. The optimal values of $\mathbf{x}$ are searched within the feasible region X, which is specified explicitly in terms of equality $h_i(\mathbf{x}), i = 1, \cdots, m$ and inequality constraints $g_i(\mathbf{x}), i = 1, \cdots, l$. The nonlinear programming problem is represented by

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h_i(\mathbf{x}) = 0, \quad i = 1, \ \cdots, \ m \end{aligned} \tag{1.1}$$

[1]This dissertation uses the term “optimization" to define the mathematical procedures as finding the maximum or minimum of a function.

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, \ \cdots, \ l$$
$$\mathbf{x} \in X,$$

where $f(\mathbf{x})$, $h_i(\mathbf{x})$, and $g_i(\mathbf{x})$ are functions of $\mathbf{x} \in X$, and the feasible region X is a nonempty open set in $\mathbb{R}^n$. A vector $\mathbf{x} \in X$ satisfying all the constraints is called a feasible solution to the problem, with the collection of all such solutions forming the feasible region. The purpose of solving the programming problem is to find an optimal point $\mathbf{x}^* \in X$ such that $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for each feasible point $\mathbf{x}$. If $\mathbf{x}^*$ is an optimal solution of Problem (1.1), then it must satisfy optimality conditions.

The concept of optimality conditions is an important concept in the field of optimization. Typically, optimality conditions provide designers with much more than a termination condition; they often provide insights into the problem frequently suggesting algorithms for solving them. When a feasible solution does not satisfy the optimality conditions, the conditions often suggest how to modify the current solution so that it becomes closer to an optimal one, as measured by a specific underlying metric. The results reported here use the Karush-Kuhn-Tucker (KKT) optimality conditions (Bazaraa et al., 2006). Under the assumption that $f(\mathbf{x})$, $h_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are continuously differentiable with respect to the variables $\mathbf{x} \in \mathbb{R}^n$, the optimality conditions are specified using the Lagrangian function by introducing a Lagrangian multiplier for each relaxed constraint. The resulting Lagrangian function is given by

$$\mathcal{L} = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{i=1}^{l} \mu_i g_i(\mathbf{x}), \tag{1.2}$$

where the sets of Lagrangian multipliers $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m]$ and $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_l]$ are introduced for relaxing the equality and inequality constraints, respectively. Assuming regularity conditions for Problem (1.1), if $\mathbf{x}^*$ is an optimal solution, then there exists a

vector $[\lambda, \mu]$ such that

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{i=1}^{l} \mu_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \\ \mu_i g_i(\mathbf{x}^*) = 0, \quad \text{for } i = 1, \ \cdots, \ m, \\ \mu_i \geq 0, \quad \text{for } i = 1, \ \cdots, \ m, \end{aligned} \tag{1.3}$$

where again λ_i and μ_i are the Lagrangian multipliers associated with the constraints $h_i(\mathbf{x}^*) = 0$ and $g_i(\mathbf{x}^*) \leq 0$, respectively. In addition, $\mu_i g_i(\mathbf{x}^*) = 0$ is referred to as a complementary slackness condition. Under suitable convexity assumptions, the KKT conditions formulated in Eq. (1.3) are also sufficient for optimality according to Bazaraa et al. (2006).

Many nonlinear programming algorithms[2] have been developed to solve Problem (1.1) with a reasonable number of decision variables and constraints. Efficient algorithms are suitable for solving convex and continuously differentiable problems with gradient- or derivative-based methods, such as Rosen's Gradient Projection Method for nonlinear constraints Rosen (1961), Zoutendijk's method of feasible directions Zoutendijk (1960), and Sequential quadratic programming (SQP) methods Han (1976), Powell (1978), Schittkowski (1983).

However, designers may not have the ability to obtain the derivatives in many engineering problems, since often, the needed functions are not explicitly available, and are the results of large codes such as Finite Elements Analysis (FEA) or Computational Fluid Dynamics (CFD). If the objective and constraint functions are nonconvex but differentiable, augmented Lagrangian penalty functions are used to ensure the local convexity of the original problems. Under the convex assumption, if the problem is non-differentiable, e.g. $f(\mathbf{x}) = |\mathbf{x}|$, subgradient methods can be used to approximate the optimal value of the objective via supporting hyperplanes on the epigraphs of convex functions (Bazaraa

[2]Defined by Wolfram MathWorld, an algorithm is a specific set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminate at some point.

et al., 2006, Bertsekas, 1999).

There are also several algorithms that do not involve derivatives, called direct methods. These methods are valuable when gradient information is not readily available or when the evaluation of the gradient is cumbersome and prone to errors. Direct methods are generally robust and introduce a degree of randomness in order to achieve global optimum designs. Some well-known direct methods are cyclic coordinates (Landau and Lifshits, 1989), Hooke and Jeeves method (Hooke and Jeeves, 1961), Rosenbrock method (Rosenbrock, 1960), simplex method of Nelder and Mead (Nelder and Mead, 1965), Powell's method of conjugate directions (Powell, 1964), and simulated annealing, genetic, and differential evolution algorithms.

1.2 Multidisciplinary Design Optimization

Large-scale products such as automobiles and airplanes are complex engineered systems comprised of many interacting subsystems and components. These complex engineered systems are usually solved by employing a systems engineering strategy, which is an architecture that enables and coordinates all the design processes within a large engineering program. However, this approach is showing its limitations, since at present the systems engineering community lacks a general theory to deal with the interactions (Griffin, 2010, Simpson and Martins, 2011). Multidisciplinary design optimization (MDO), which has evolved remarkably since its inception 25 years ago, offers alternative methods[3] to complement and enhance the systems engineering to help address the challenges inherent in the design of complex engineered systems (Simpson and Martins, 2011). More specifically, MDO is a field of systems engineering that uses optimization methods to solve design problems incorporating a number of disciplines.

It is an important part of MDO to rely on computational design, i.e. integrating

[3]In the literature, many terms have been used to describe MDO methods, such as "method," "methodology," "problem formulation," "strategy," "procedure," "algorithm," and "architecture.", This dissertation uses the term "method", which refers to the combination of the design problem formulation and the organizational and algorithmic strategy used to solve it as an MDO method.

analysis software in a “black-box" fashion and employing a surrogate model. Previous computer technology enabled higher fidelity codes to be processed faster, and gradually erased the distinctions between the analysis and optimization codes, leading to large monolithic codes invoking several disciplinary mathematical models in a single optimization cycle (Fulton et al., 1974, Vanderplaats, 1976). The methods adopting this single optimization cycle have been referred to in literature as the multiple discipline feasible (MDF) (Cramer et al., 1994), individual discipline feasible (IDF) (Cramer et al., 1994), and simultaneous analysis and design (SAND) and all-at-once (AAO) methods (Balling and Sobieszczanski-Sobieski, 1996, Cramer et al., 1992, 1994, Haftka, 1985). Due to the large number of design variables involved in each disciplinary model, the intrinsic practical limitations of these monolithic systems of disciplinary codes soon became apparent, leading to the development of decomposition-based MDO methods.

Subsequently, decomposition as an approach to break a large-scale optimization problem into an equivalent set of smaller, independent but interacting subproblems and components was successfully developed and applied to MDO (Balling and Sobieszczanski-Sobieski, 1996, Sobieszczanski-Sobieski and Haftka, 1997, Wagner and Papalambros, 1993). This decomposition is used to compress the execution time by applying additional resources, whether human or computational, to solve the problem at hand (Agte et al., 2010). It became readily apparent that for an internally coupled multidisciplinary system, the optimal system-level design was more than simply a collection of individually optimized subsystems and components (Agte et al., 2010).

In addition, the decomposition implementation for the multidisciplinary system involved many diverse sets of analytical equations whose interaction often created additional problems within itself, and in large applications involved a number of different teams of specialists. As a result, that involvement required (and still requires) dealing with a non-mathematical but crucially important set of human factors as a prerequisite to success (Agte et al., 2010). A few examples of the decomposition-based MDO methods are Concurrent Subspace Optimization (CSSO) and its variants (Bloebaum et al., 1992,

Sobieszczanski-Sobieski, 1988, Wujek et al., 1996), Collaborative Optimization (CO) and its variants (Braun et al., 1997, Demiguel and Murray, 2006, Roth and Kroo, 2008), Bi-Level Integrated Systems Synthesis (BLISS) and its variants (Kodiyalam and Sobiesczanski-Sobieski, 2000, Sobieszczanski-Sobieski et al., 2000, 2003), and Analytical Target Cascading (ATC) and its variants (Kim et al., 2006, 2003, Michelena et al., 1999, 2003, Tosserams et al., 2006a, Wang et al., 2010). Among these decomposition-based MDO methods, a relatively recent contribution of MDO includes methods that allow for finding optimal system designs for a set of desired targets or goals. These are important because the development of products in industry is often driven by the changing requirements caused by technological and societal innovations as seen in Figure 1.1.

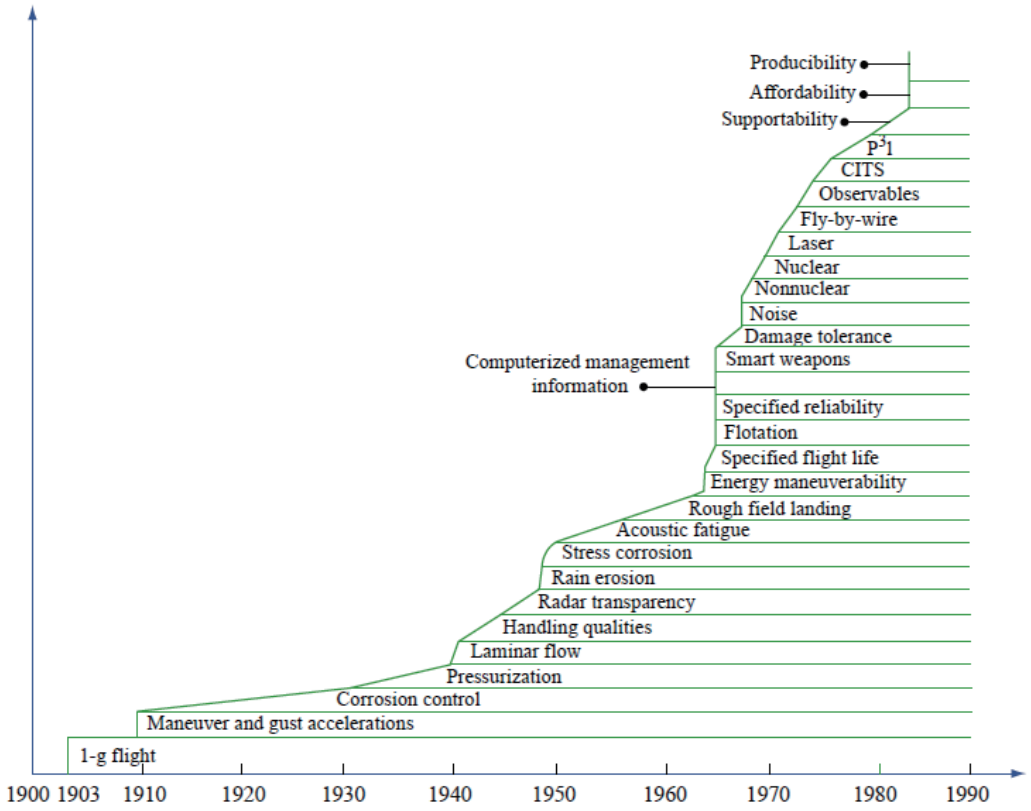


Figure 1.1: Design requirements growth for aerospace vehicles (Allen et al., 2004)

Such methods typically seek one or more solutions that meet a design target within a pre-specified numerical performance. The methods that enable goal-seeking

design for MDO are, among others, Physical Programming (PP) (Messac, 1996), Iso-Performance (IP) (de Weck and Jones, 2006), Analytical Target Cascading (ATC) (Kim et al., 2003), and Augmented Lagrangian Coordination (ALC) (Tosserams et al., 2008b). The latter two, as decomposed-based, goal-seeking MDO methods, are ones with proven solution convergence and equivalence (Bertsekas, 2003, Lassiter et al., 2005, Li et al., 2008). In today's business world, the request for maximum performance has been superseded by a need for a "balance" among performance, product development cost, reliability, maintainability and other "-ilities" (Allen et al., 2004). Consequently, the further development of decomposition-based, goal-seeking MDO methods is still in demand.

1.3 Sequential and Concurrent Engineering Strategies

Since 1980s, concurrent engineering (CE) has been the focus of research. It was studied and widely implemented in industry to compress time to market and cost (Prasad, 1996), thereby improving the productivity and performance of the product development process (PDP). Previously, PDP employed sequential engineering as the business strategy, in which the various design tasks were completed one after another, with all attention and resources focused on that current task. After a task is finished, all resources are then concentrated on the next one. In CE, however, different design tasks are addressed at the same time and not necessarily in sequential order. More precisely, CE is an approach used in PDP by which several teams within an organization work simultaneously to develop new products and services, allowing for a streamlined procedure. In addition, products and processes in CE are closely coordinated to achieve an optimal matching of requirements for effective cost, quality, and delivery.

For the purpose of comparison, a specific problem is very useful in thinking about MDO and sequential and concurrent engineering strategies. For this dissertation, the model problem is an aeroelastic optimization given by Cramer et al. (1994). In static aeroelasticity, a flexible wing of an aircraft in steady flight is considered, as shown in

Figure. The air rushing over the wing causes pressures to be imposed on the wing, which causes the wing to deflect and change shape. This change in wing shape in turn causes the aerodynamic pressures to change. In static aeroelasticity, these physical processes are considered to reach an equilibrium of forces.

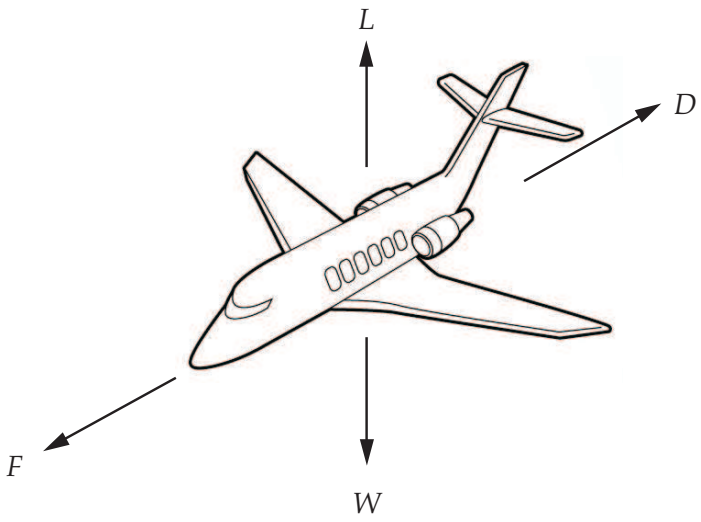


Figure 1.2: A schematic of the forces on an aircraft in steady level flight

The aeroelastic system involves two analysis disciplines, which are aerodynamics and structures. The computational problems for these two disciplines are generally solved by individual analysis codes, i.e., a finite difference CFD code for aerodynamics, and a finite element code for structures. Next, an optimization problem is added to the aeroelastic example (Cramer et al., 1994). For example, a range minimization problem for this aeroelastic model is given by

$$\begin{aligned} \min_{\alpha,\gamma,t,\Lambda} \quad & -R(\alpha,\gamma,t) = -\left[\frac{V}{c} \times \frac{L}{D} \times \ln\left(\frac{W_i}{W_f}\right)\right] \\ \text{subject to} \quad & g = \sigma_{\text{yield}} - \sigma_j(t) \geq 0, \\ & h = L(\alpha,\gamma) - W = 0, \end{aligned} \tag{1.4}$$

where R is the range or the objective function obtained through the Breguet range equation; V is the velocity, which is a system-level variable; L/D is the lift-to-drag ratio,

which is output from the aerodynamics module; c is the specific fuel consumption from the propulsion module and fixed as a constant value; and W_i and W_f are the initial aircraft weight and the final aircraft weight due to fuel burned, both of which are outputs from the structural module. The parameter Λ is the wing sweep range, which is a global design variable; t is the wing thickness, which is a local design variable of the structural module; and α and γ are the wing twist angle and the tail sweep angle, respectively, both of which are local design variables of the aerodynamics module. For an aircraft in steady, level flight, the normal stress on the wing is limited by the yielding stress, being represented as an inequality constraint g; and the lift force is equal to the weight, being represented as an equality constraint h.

Figure 1.3 shows the partitioning of the range minimization problem of the aeroelastic system. Since the coupling variables, i.e., forces, drag, displacements, and weight, must be consistent, each subsystem minimizes the range while satisfying its own constraints by treating the coupling variables solved by the other subsystem as constant values. Consequently, the design tasks of these two subsystems have to be implemented in a sequential order.

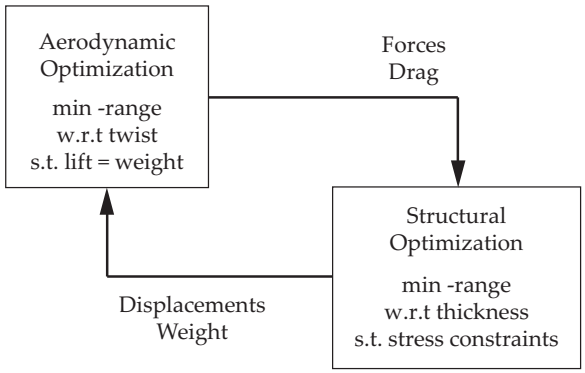
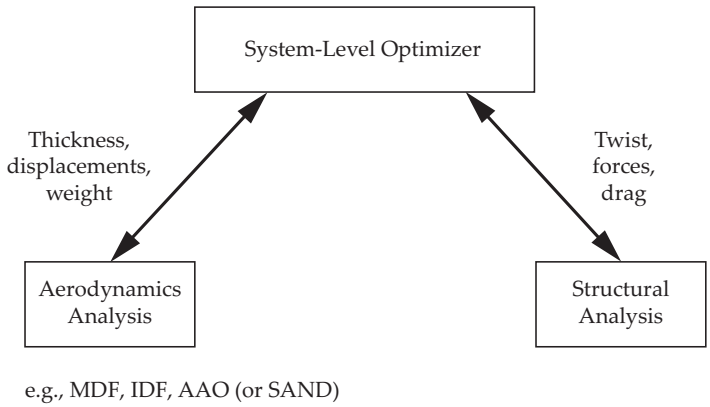


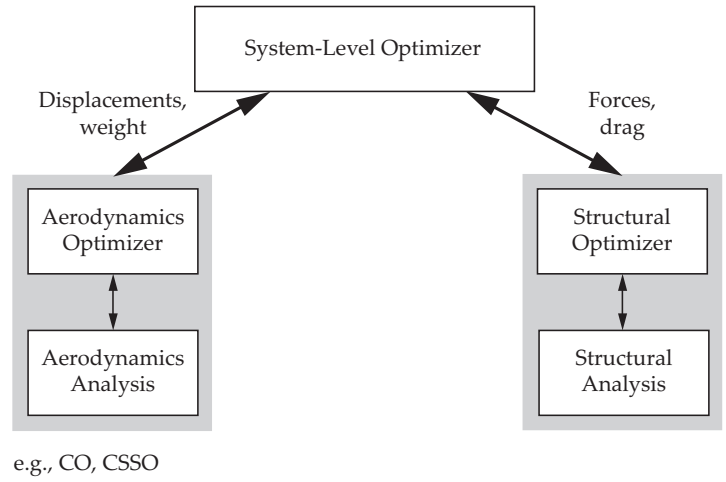
Figure 1.3: Sequential engineering strategy of the range minimization problem

To take advantage of the concurrent engineering strategy, the analysis of the wing design problem can be performed in disciplinary models with a centralized optimization as seen in Figure 1.4 (a). This framework implements the analysis models concurrently, which is realized by decoupling the disciplinary analyses so that they no longer rely on

one another for their coupling variable inputs, and the coupling variables are added to the set of design variables.



(a) Distributed disciplinary analysis models with a non-hierarchical framework



(b) Distributed disciplinary design models with a hierarchical (bi-level) framework

Figure 1.4: Concurrent engineering strategy of the range minimization problem

To provide the discipline autonomy while enforcing the interdisciplinary consistency, the range minimization problem can also be solved with two disciplinary models that perform design tasks including optimization and analysis, see Figure 1.4 (b). In this framework, discipline feasibility is maintained throughout the system-level optimization process since the discipline optimizations are responsible for generating discipline feasible solutions for each system-level iteration.

Similar to CE, parallel computing has been adopted in software development as an alternative to sequential computing. This approach involves the simultaneous use of multiple compute resources to solve a computational problem. When implementing parallel computing, a problem is broken into discrete parts that can be solved concurrently, and then each part is further broken down into a series of instructions. Finally, instructions from each part are executed simultaneously on different CPUs (Barney, 2012). The primary reasons for using parallel computing are to shorten the computational time with potential reduced cost, to solve large-scale and complex problems, and to use computer resources on a wide area network. Over the past 20+ years, the trends shown by ever faster networks, distributed systems, and multi-processor computer architectures indicate that parallelism is the future of computing.

For these reasons, it is necessary to develop a quantitative solution method that can organize the multidisciplinary design tasks of the complex systems by including goal-seeking and concurrent design concepts to meet the design requirements. To address this need, this dissertation investigates coordination methodologies within the concurrent, multidisciplinary optimization and goal-seeking framework.

1.4 Product Development Process

Generic Product Development Process

Produce development process (PDP) is the sequence of steps or activities which an enterprise employs to conceive, design, and commercialize a product (Ulrich and Eppinger, 2007). Many of these steps and activities are intellectual and organizational. In industries, organizations usually define and follow a precise and detailed development process. Every organization employs a process at least slightly different from that of every other organization. In fact, the same enterprise may follow different processes for different types of development projects.

A generic PDP consisting of six phases, depicted in Figure 1.5, is adopted from

Ulrich and Eppinger (2007). The process begins with a planning phase, which is the link to advanced research and technology development activities. The output of the planning phase is the project's mission statement, which is the input required to begin the concept development phase and which serves as a guide to the development team. The conclusion of the product development process is the product launch, at which time the product becomes available for purchase in the marketplace. A summary of the input and output of each phase in the product development process is illustrated in Table 1.1.

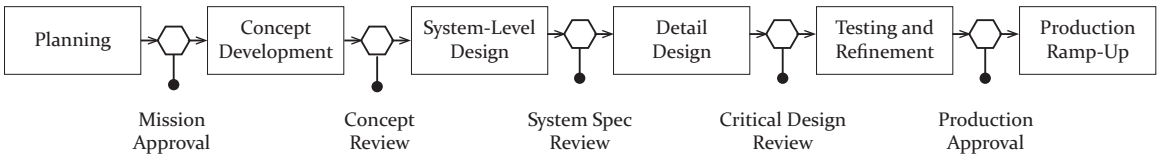


Figure 1.5: Generic product development process (Ulrich and Eppinger, 2007)

Table 1.1: Inputs and outputs of the generic product development process.

Phrases	Inputs	Outputs
1. Planning	Corporate strategy and assessment of technology developments and market objectives.	Project mission statement.
2. Concept development	Project mission statement.	Identification of target market and evaluation of alternative conceptual designs.
3. System-level design	One or more conceptual designs.	A geometric layout of the product, a functional specification of each of the subsystems, and a preliminary process flow diagram for the final assembly process.
4. Detail design	Functional specification of the product.	Complete specifications of the subsystems and components, complete tolerances of all of the unique parts, and the identification of all of the standard parts to be purchased from suppliers.
5. Testing and refinement	Complete specifications and tolerances of each part and standard parts purchased from suppliers.	Construction and evaluation of multiple preproduction versions of the product such as early and later prototypes.
6. Production ramp-up	Prototypes	Work force training and solving the remaining problems in the production processes.

The PDP described in Figure 1.5 and Table 1.1 is generic, and particular processes will differ in accordance with a company's unique context. In addition to the generic PDP, automobile and aerospace industries often adapt a variant of the generic process,

i.e., complex system development process. The characteristics of the complex system development process and the resulting deviations from the generic process are compared in Table 1.2.

Table 1.2: Summary of generic and complex product development processes

	Generic Products	Complex Systems
Description	The team begins with a market opportunity and selects appropriate technologies to meet customer needs.	Systems must be decomposed into several subsystems and many components.
Distinct Features	Process generally includes distinct planning, concept development, system-level design, detail design, testing and refinement and production ramp-up phases.	Subsystems and components are developed by many teams working in parallel, followed by system integration and validation.
Examples	Printers, screwdrivers, rollerblade skate, etc..	Airplanes, engines, automobiles, etc..

Complex System Development Process

Large-scale products such as automobiles and airplanes are often considered as complex systems comprised of many interrelated subsystems and components. Most people without experience in product development are impressed by how much time and money are required to develop a complex system. The reality is that very few complex systems can be developed in less than one year; many require three to five years, and some take as long as ten years. Table 1.3 depicts five engineered products with a table showing the approximate scale of the associated product development efforts along with various distinguishing characteristics of the products (Ulrich and Eppinger, 2007). Vehicles and airplanes are considered typical examples of complex systems; however, scholars argue that small-scale systems, such as micro-accelerometers, involving a large number of interrelationships across subsystems and components, can also be categorized as complex systems (Tosserams et al., 2010).

When developing complex systems, modifications to the generic PDP address a number of system-level issues. Figure 1.6 shows a modification of the generic PDP (Ulrich and Eppinger, 2007). The concept development phase considers the architecture of the entire system with multiple architectures perhaps being considered as competing concepts

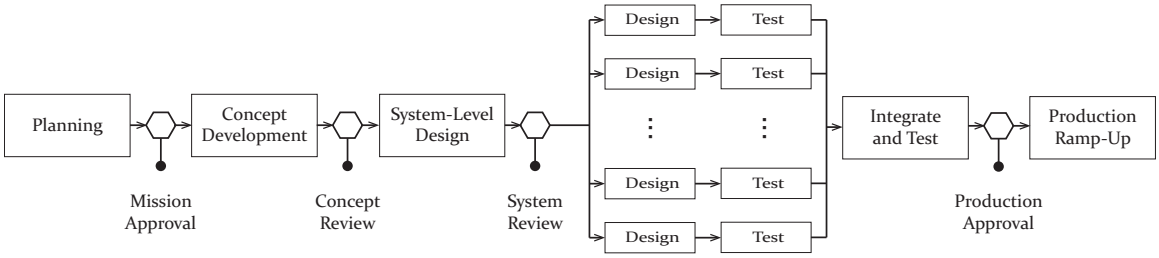


Figure 1.6: Complex system development process (Ulrich and Eppinger, 2007)

Table 1.3: Attributes of five products and their associated development efforts (Ulrich and Eppinger, 2007)

	Stanley Screwdriver	Rollerblade Skate	HP Deskjet Printer	Volkswagen New Beetle	Boeing 777 Airplane
Annual production volume (unit/year)	100,000	100,000	4 million	100,000	50
Sales lifetime (years)	40	3	2	6	30
Number of unique parts	3	35	200	10,000	130,000
Development time (years)	1	2	1.5	3.5	4.5
Internal development team (people)	3	5	100	800	6,800
External development team (people)	3	10	75	800	10,000
Development cost (dollars)	150,000	750,000	50 million	400 million	3 billion
Product investment (dollars)	150,000	1 million	25 million	500 million	3 billion

for the overall system. The system-level design phase becomes critical. During this phase, the original system is decomposed into multiple subsystems and then further into many components. After the decomposition, teams are assigned to develop each component; additional teams are assigned the specific challenge of integrating components into the subsystems and these into the overall system. Consequently, each subsystem is handled by a team that relies on its own design/test tools or methods.

1.5 The Role of the Systems Engineer

In the system-level design phase, designers often face two challenges. One is that each team strives to achieve its own design objective by satisfying its own requirements without knowing how those design decisions influence the behavior of other subsystems or even the overall system. The second is that the detailed design of subsystems and

components requires the implementation of a highly parallel process in which the many development teams work concurrently but not in isolation.

Managing interrelationships across the components and subsystems is the task of many types of system engineers. Complex system engineering seeks a feasible and balanced design in the face of opposing interests and multiple, sometimes competing design requirements. The system engineer must develop skills and instincts for identifying and focusing effort on assessments to optimize the overall design and not favor one subsystem or component at the expense of another (Kapurch, 2007). The art is in knowing when and where to probe. Personnel with these skills are usually tagged as “systems engineers," but they may have other titles, such as lead system engineer, technical manager, or chief engineer. For this dissertation, the term system engineer is used.

The exact role and responsibility of the system engineer may change from project to project depending on the size and complexity of the project and from phase to phase of the development process. For large-scale projects, there may be one or more system engineers. For small projects, sometimes the project manager may perform these responsibilities. No matter who assumes these duties, the complex system engineering functions must be performed. The system engineer ensures that the system technically fulfills the defined design requirements and a proper coordination methodology is being followed. The system engineer oversees the project’s activities as performed by the technical teams and directs, communicates, monitors, and coordinates the design tasks of subsystems and components. The system engineer reviews and evaluates the technical aspects of the project to ensure that the subsystem and component design processes are being implemented properly and evolves the system from concept to product. In summary, the system engineer is skilled in the art and science of balancing organizational and technical interrelationships in complex systems.

1.6 Research Scope

In human organizations, e.g. design teams consisting of project managers and design engineers, various decision makers may have different objectives, which may be different from the goal of the organization. To eliminate resulting conflicts, this dissertation is restricted to discussions for which three assumptions are made.

Assumption 1 There is a well-defined organizational objective, which is determined based on company goals, customer needs, and government regulations.

Assumption 2 The individual decision makers are either physical processors or they are treated as if there were processors with predictable behavior.

Assumption 3 Design optimization of each subsystem or component operating in parallel takes the same amount of time. If one subsystem of component analysis finishes early, it waits on the others to finish.

This dissertation is limited to general concepts and generic descriptions of coordination processes and techniques. Specifically, it provides information on coordination methodologies and algorithms, describes the mathematical model for complex system engineering problems, and analyzes the effort in the proposed methodologies and algorithms using academic version engineering problems. The specifics of the demonstration examples can be seen in the description of the partition structures and the details of the coordination algorithms. Each example varies in these two areas, meaning the reader should refer to the procedural requirements for the problem's partition structure and coordination algorithm.

1.7 Motivation and Research Objectives

The research presented in this dissertation is primarily motivated by recent efforts in the automotive, aerospace, and other industries to formalize the product development

process and take advantage of parallel computation. More specifically, this research is motivated by the observation of the relationship between manufacturers and suppliers. For example, Boeing annually purchases more than $50 billion in components and services from a global network of more than 28,000 suppliers that collectively employ more than 1.2 million people; the suppliers are required to demonstrate their capabilities for providing Boeing-qualified components, on-time delivery, post-delivery support, competitive cost, and swift response to changing requirements. To realize an efficient product development process, the various tasks of designing subsystems and components should be accomplished in parallel but not in isolation from one another (Kim, 2001). The product development process for complex engineering systems results in a network of design teams including various specialists and communicating with their fellow teams over the network regularly.

The product development process, in fact, should be considered as an organizational problem rather than one able to be solved simply by grinding through a mathematical model or computer algorithm (Churchman and Eisenberg, 1969). However, early work in developing a computer-aided environment or software to support complex, unstructured group decision processes within organizations has had an adverse effect, because as a result, decisions could be made quickly on trying to satisfy all requirements (Nunamaker et al., 1988). More specifically, when conducting design tasks in parallel and isolation (i.e., ignoring the existing interactions shared among design teams), each design team could focus on its own task; however, isolating the interactions leads to higher downstream[4] costs (Kim, 2001).

This dissertation investigates fundamental methodologies and numerical applications on two coordination methodologies. The first coordination methodology is the further investigation of analytical target cascading (ATC) for coordinating hierarchically decomposed systems using the cutting plane method and bi-objective optimization methods. The second relates to the development of a new non-hierarchical, decentralized coor-

[4]In software engineering, downstream refers to data sent from a network service provider to a customer.

dination methodology for a multi-agent network model. The new coordination methodology is termed network target coordination (NTC), the goal of which is to collectively optimize the decomposed system without using a master problem.

1.8 Dissertation Outline

The research presented in this thesis adapts several existing optimization algorithms, involving cutting plane methods (Bertsekas, 2003) and alternating direction method of multipliers (ADMM) (Bertsekas and Tsitsiklis, 1989, Boyd et al., 2011, Gabay and Mercier, 1976), to develop coordination methods for complex systems decomposed based on multidisciplinary design optimization (MDO). The first topic of the thesis is to apply and investigate the cutting plane methods for solving hierarchically decomposed problems. The second topic of the thesis is to develop a network target coordination method based on consensus optimization and ADMM for realizing the complete parallelization of nonhierarchically decomposed problems.

Chapter 1 introduced MDO methods in the contexts of systems engineering and product development process, indicating that the primary focus of this dissertation was on mathematical formulations and coordination strategies for both the hierarchically and nonhierarchically decomposed problems. The outline of this dissertation is depicted in Figure 1.7.

Chapter 2 overviews well-known MDO methods based on three steps involved in the solution procedure. In Section 2.1, a consistent notation is introduced to identify general aspects of the distributed optimization methods developed in the literature. In Section 2.2, the problem statements are presented to identify relationships between variables and functions associated with individual subsystems. In Section 2.3, the classification criteria are discussed in more detail. Finally, the general properties of distributed optimization methods are concluded in Section 2.4.

Chapter 3 overviews the rational for using the augmented Lagrangian method to

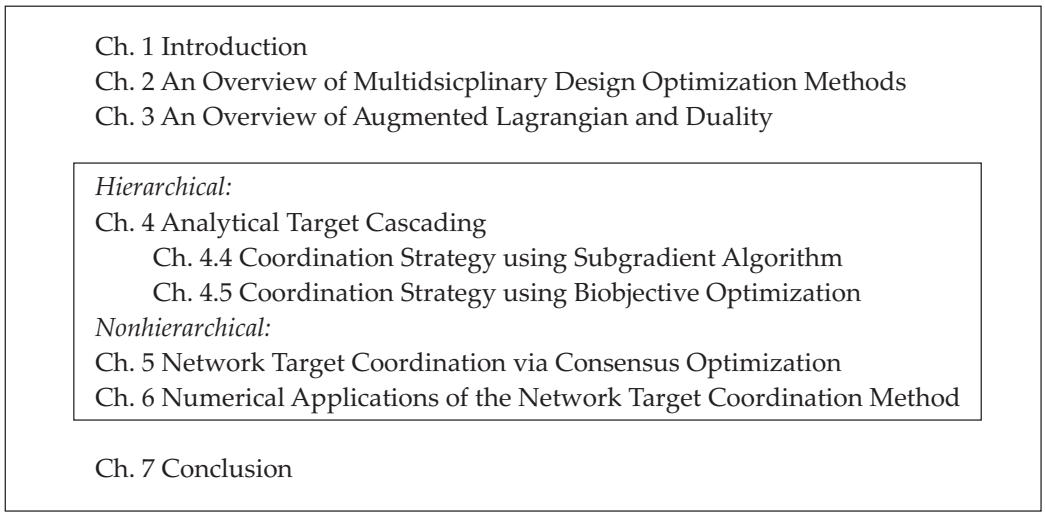


Figure 1.7: Outline of the dissertation

solve decomposable engineering system design problems. The augmented Lagrangian method is an extension of the quadratic penalty method and Lagrangian method introduced in Chapter 2. In Section 3.2, the augmented Lagrangian dual formulation is presented. Then, its augmented Lagrangian dual problem is defined in Section 3.3 and the strong and weak duality are discussed. Section 3.3 presents the first-order optimality conditions for the constrained optimization problem by analyzing the dual and primal feasibility. Section 3.4 discusses the significance of Lagrangian multipliers. Section 3.6 illustrates the augmented Lagrangian coordination with respect to the introduction of copies of coupling variables and the relaxation of consistency constraints. Finally, the duality theorem and the augmented Lagrangian method are concluded in Section 3.7.

Chapter 4 presents the two new coordination methods for solving complex systems decomposed based on ATC. The main contribution of this chapter is to provide a unified duality view of the subgradient update schemes for solving ATC-decomposable problems. In addition, this chapter discusses the advantages and disadvantages of these two cutting plane methods when applied to ATC-decomposable design problems. In Section 4.2, the centralized, hierarchical problem structure is modeled. In Section 4.3, ATC problem statement is presented, and a generic subgradient algorithm with five update

schemes for solving the ATC-decomposed problems are given in Section 4.4. In Section 4.5, bi-objective optimization and its corresponding coordination algorithm are presented. Finally, numerical applications are tested in Section 4.6 and the results are analyzed in Section 4.7.

Chapter 5 proposes a new coordination method via consensus optimization with the alternating direction method of multipliers (CADMM) to solve nonhierarchical decomposed problems. This solution strategy consists of three steps. In the first step, each agent solves for coupling variables locally with fixed Lagrangian multipliers, penalty parameters, and the consensus estimates of all coupling variables. After achieving the convergence of all subsystems, the second step aims to compute the consensus estimates using the results of the coupling variables collected from each subsystem. In the third and final step, the Lagrangian multipliers are updated based on sub-gradient methods.

Chapter 6 demonstrates the performance of the proposed method by solving three nonconvex problems, including a geometric programming problem, the Golinski's speed reducer problem, and the MEMS-based micro-accelerometer design problem. The solution of these three problems are computed, and then the performance of the proposed coordination method via the CADMM approach is analyzed with a comparison to the ATC method via the alternating direction method of multipliers (ADMM) in the literature. The numerical results indicate that NTC with the CADMM is more efficient and robust than the ATC with ADMM approach when solving the example problems.

Chapter 7 summarizes the primary contributions of the research, and gives recommendations for future research. From a mathematical viewpoint, the contribution of the research is the development of mathematical formulations and corresponding algorithms for realizing the large-scale, distributed design of both the hierarchically and nonhierarchically decomposed systems. From an engineering application viewpoint, the research reduces the computational costs and improves the solution accuracy for solving large-scale design problems completed by geographically dispersed teams.

Chapter 2

An Overview of Multidisciplinary Design Optimization Methods

2.1 Introduction

Traditionally, the integration of optimization in engineering design was realized by allowing design teams to contribute discipline analysis models to a system-level configuration team, which executed the models using single and monolithic methods, e.g. All-In-One (AIO), multidisciplinary feasible (MDF), and individual discipline feasible (IDF) methods. These methods are only useful during the early design stage, when the number of decision variables is relatively small, and they are appropriate for a small design team. Because of their limitations, single and monolithic methods face challenges when solving complex systems that are large-scale and multidisciplinary, and require a higher performance with a low cost.

To address these challenges, a variety of multidisciplinary design optimization (MDO) methods have been developed for efficient analysis and optimization of these complex systems. Among others, several well-known methods are Concurrent Subspace Optimization (CSSO) (Sobieszczanski-Sobieski, 1988), Collaborative Optimization (CO) (Braun et al., 1997, Braun, 1996), Bi-level Integrated Systems Synthesis (BLISS) (Sobieszczanski-Sobieski et al., 2000, 2003), Analytical Target Cascading (ATC) (Kim et al., 2003, Miche-

lena et al., 1999, 2003), Quasi-separable Decomposition (QSD) (Haftka and Watson, 2005, 2006), Inexact/Exact Penalty Decomposition (IPD/EPD) (Demiguel and Murray, 2006), Enhanced Collaborative Optimization (ECO) (Roth and Kroo, 2008), and Augmented Lagrangian Coordination (ALC) (Tosserams et al., 2008b). These MDO methods are also summarized as following:

Equality-based MDO methods

Collaborative Optimization (CO) - Copies of the coupling variables are introduced for each subsystem. Discipline subsystems minimize consistency constraints, denoted by the discrepancies between the copies of coupling variables and their responses, by subjecting local design constraints. The system-level problem minimizes the original objective by subjecting to consistency constraints (Braun et al., 1997, Braun, 1996).

Quasi-separable Decomposition (QSD) - Each discipline is assigned a target for a local objective and the discipline subsystems maximize the residual in their local constraints and the discrepancy between the target and local objective. The system-level problem minimizes a coupled objective function and the targets of each discipline by subjecting to coupled design constraints and the residual in each discipline (Haftka and Watson, 2005, 2006).

Concurrent Subspace Optimization (CSSO) - In the system-level problem, disciplinary analyses are conducted using surrogate models. In discipline subsystems, the coupling variables and constraints are solved using surrogate models, and then the solutions of these discipline subsystems are used to update the surrogate models (Sobieszczanski-Sobieski, 1988).

Bi-level Integrated Systems Synthesis (BLISS) - Coupled derivatives of the multidisciplinary analysis are used to construct linear approximations for each discipline subsystem. The derivatives with respect to the solutions of these subsystems are

computed to form the system-level linear approximation, which is then optimized with respect to coupling variables (Sobieszczanski-Sobieski et al., 2000, 2003).

Relaxation-based MDO methods

Inexact/Exact Penalty Decomposition (IPD/EPD) - This method is applicable to solve MDO problems with no coupling objectives and/or constraints. Copies of coupling variables are introduced for each subsystem. The consistency constraints are relaxed using a penalty function. The coordination structure of the discipline subsystems is investigated to compute sensitivities used for solving the system-level problem (Demiguel and Murray, 2006).

Enhanced Collaborative Optimization (ECO) - As in Collaborative Optimization, copies of coupling variables are also introduced. Discipline subsystems minimize quadratic approximations of the objective by subjecting to local constraints and linearized coupling constraints. Coupling variables are solved by the system-level problem, which minimizes the sum of consistency constraints (Roth and Kroo, 2008).

Analytical Target Cascading (ATC) - Both copies of the coupling variables and their corresponding consistency constraints are introduced in discipline subsystems. These consistency constraints are then relaxed using a quadratic penalty function. System-level problem and discipline subsystems independently solve their respective optimization problems. During the iterative process, penalty weights are gradually increased until the desired termination tolerance for consistency constraints is achieved (Kim et al., 2003, Michalek and Papalambros, 2005a,b, Michelena et al., 1999, 2003).

Augmented Lagrangian Coordination (ALC) - Similar to ATC, both copies of coupling variables and corresponding consistency constraints are introduced in discipline subsystems. Unlike ATC, these consistency constraints are relaxed using an augmented Lagrangian function. During the iterative process, both Lagrangian

multipliers and penalty weights are updated using various schemes in order to achieve the desired termination tolerance for consistency constraints (Tosserams et al., 2008b).

The fundamental idea of these MDO methods is to decompose the system into a number of subsystems, and independently solve them using the standard optimization algorithms. The decomposition process of these MDO methods is usually comprised of two steps, i.e. partitioning and coordination (Wagner, 1993). The partitioning step separates the system into many smaller subsystems so that they can be solved by various design teams using object-, aspect-, or model-based partitioning strategies (Wagner and Papalambros, 1993). The coordination step manages interactions among subsystems in order to achieve a consistent, optimal solution, which should be equivalent to the optimal solution for the original system.

Previous research has classified these MDO methods into the following widely used categories:

- Problem formulation for single-level MDO optimization (Cramer et al., 1994),
- Single-level v.s. multi-level optimization (Balling and Sobieszczanski-Sobieski, 1996),
- Positioning of computational costs and organization of the individual subsystems (Sobieszczanski-Sobieski and Haftka, 1997),
- Positioning of consistency constraints (Alexandrov and Lewis, 1999),
- Formulation structure and constraint relaxation (Tosserams et al., 2008a),
- Unified notation and solution procedure parallel to the decomposition and coordination steps (de Wit and van Keulen, 2010), and
- Unified description including optimization problem statements, diagrams, and detailed algorithms (Lambe and Martins, 2011).

This chapter provides an overview of the MDO methods, focusing on three general features for solving complex systems, 1) whether to introduce copies of coupling variables, 2) whether to relax the consistency constraints, and 3) which solution order to choose. Section 2.1 introduces a consistent notation to identify general aspects of the distributed optimization architectures developed in the literature. Section 2.2 illustrates the functional dependent table for identifying relationships between variables and functions associated with individual subsystems. Section 2.3 discusses the classification criteria in more detail and presents the formulations corresponding to MDO methods. Section 2.4 concludes by giving the general properties of distributed optimization architectures.

2.2 Terminology and Notation

Before discussing the classification criteria, the notation used throughout this chapter must be defined. This notation, developed to compare the various problem formulations within MDO architectures, is listed in Table 2.1.

Table 2.1: Mathematical notation for MDO problem data.

Symbol	Definition
$\mathbf{x}$	Vector of local design variables
$\mathbf{y}$	Vector of coupling design variables (outputs from a discipline analysis)
$\mathbf{y}^t$	Vector of coupling variable targets (inputs to a discipline analysis)
f	Objective function
$\mathbf{g}$	Vector of inequality design constraints
$\mathbf{h}$	Vector of equality design constraints
$\mathbf{c}$	Vector of consistency constraints
M	Total number of disciplines (or subsystems)
0	This subscript means that functions are shared by more than one discipline, called coupling functions or variables
j	This subscript means that functions or variables are associated only with discipline j
$\star$	This superscript means that functions or variables are achieved at their optimal values
$\sim$	This symbol denotes an approximation of a given function or vector of functions

In addition, several terms specific to the field of MDO are specified as follows:

- A design variable is one under the explicit control of an optimizer. Design variables may pertain to a single discipline, i.e. be local, or may be shared by multiple disciplines (Lambe and Martins, 2012).
- A discipline analysis is a simulation that models the behavior of one discipline. Conducting a discipline analysis consists of solving a series of equations and returning a set of response variables (Lambe and Martins, 2012).
- A local design variable, denoted by x_j, is a variable that is exclusively associated with discipline j. The vector of local variables is denoted by $\mathbf{x}_j$.
- A linking design variable, denoted by y, is a variable that appears in more than one discipline. This linking variable may be a design variable, or an analysis response computed by one discipline used as an input to others. The vector of linking variables solved by discipline j is denoted by $\mathbf{y}_j$. These linking variables are also referred to as coupling variables.
- In many formulations, copies of the coupling variables must be made to allow the discipline analysis or optimization to run independently and in parallel. These copies are known as target variables denoted by a superscript t (Lambe and Martins, 2012).
- The function f is called, variously, an objective function, a cost function (minimization), a utility function (maximization), or, in certain fields, an energy function or energy functional. A feasible solution that minimizes (or maximizes, if that is the goal) the objective function is called an optimal design solution, which is denoted by $\mathbf{x}^\star$.
- To preserve consistency between the coupling variable inputs and outputs at convergence, a set of auxiliary constraints, i.e. $\mathbf{c}_j = \mathbf{y}^t - \mathbf{y}_j$, is defined as the consistency constraints added to the formulation.

2.3 Problem Statements

The most important consideration in the partitioning for systems optimization is to identify the two types of interactions, i.e. coupling variables $\mathbf{y}$ and coupling functions f_0, $\mathbf{g}_0$, $\mathbf{h}_0$. The problem statement including only the coupling variables is given by

$$\begin{aligned} \min_{\mathbf{y},\, \mathbf{x}_1,\, \cdots,\, \mathbf{x}_M} \quad & \sum_{j=1}^{M} f_i(\mathbf{y}, \mathbf{x}_j) \\ \text{s.t.} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \cdots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \cdots, M. \end{aligned} \tag{2.1}$$

Although the size does not matter in this equation, it is useful to think that $\mathbf{x}_j$ has more components than $\mathbf{y}$. The objective f_j and constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ are almost block-separable in $\mathbf{x}_j$, meaning the subsystems can be solved independently. For this reason, $\mathbf{y}_j$ is referred to as the vector of coupling variables because when it is treated as a constant, Problem (2.1) can be separated into M subsystems. In other words, the vector of coupling variables complicates the problem.

Unlike block-separable Problem (2.1), the problem statement involving only the coupling functions is defined as

$$\begin{aligned} \min_{\mathbf{x}_1,\, \cdots,\, \mathbf{x}_m} \quad & f_0(\mathbf{x}_1,\ \cdots,\ \mathbf{x}_m) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{x}_1,\ \cdots,\ \mathbf{x}_m) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{x}_1,\ \cdots,\ \mathbf{x}_m) = \mathbf{0} \\ & \mathbf{g}_j(\mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \cdots, M \\ & \mathbf{h}_j(\mathbf{x}_j) = \mathbf{0} \quad j = 1, \cdots, M. \end{aligned} \tag{2.2}$$

Although local constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ are block-separable in $\mathbf{x}_j$ for $j = 1,\ 2,\ \cdots,\ M$, the functions f_0, $\mathbf{g}_0$ and $\mathbf{h}_0$ couple these subsystems, making the solving process of the problem complicated.

After combining two forms of interdependencies, a general formulation of the problem with a decomposable structure is given by

$$\begin{aligned} \min_{\mathbf{y},\, \mathbf{x}_1,\, \cdots,\, \mathbf{x}_m} \quad & f_0(\mathbf{y},\, \mathbf{x}_1,\, \cdots,\, \mathbf{x}_m) + \sum_{j=1}^{M} f_j(\mathbf{y}, \mathbf{x}_j) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y},\, \mathbf{x}_1,\, \cdots,\, \mathbf{x}_m) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y},\, \mathbf{x}_1,\, \cdots,\, \mathbf{x}_m) = \mathbf{0} \\ & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0} \quad j = 1, \cdots, M \\ & \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \quad j = 1, \cdots, M. \end{aligned} \tag{2.3}$$

The overall system objective is assumed to be the sum of the coupling objective f_0 and the subsystems objectives $f_j \; \forall \; j = 1, \; \cdots, \; M$. The original constraints are divided into coupling constraints $\mathbf{g}_0$ and $\mathbf{h}_0$ allocated to the system-level problem, and local constraints $\mathbf{g}_j$ and $\mathbf{h}_j$ assigned to subsystem j. Without coupling variables and functions, Problem (2.3) could be partitioned into M subsystems. These subsystems can be solved individually by design teams because there is no interdependency between subsystems and, therefore, there is nothing to coordinate. However, when coupling variables and functions are involved, a coordination method is needed to manage them, and to guide individual subsystems towards the achievement of the optimal system design.

The relationships among local design variables $\mathbf{x}_j$, coupling variables $\mathbf{y}$, objective functions f_0 and f_j, and constraint functions $\mathbf{g}_0$, $\mathbf{h}_0$, $\mathbf{g}_j$, and $\mathbf{h}_j$ with $\forall \; j = 1, \; \cdots, \; M$ has been analyzed by Barthelemy (1989) and Wagner and Papalambros (1993), among others, using functional dependency tables (FDT). An example of the structure of the interactions and its corresponding FDT of Problem (2.3) are depicted in Figure 2.1 and Table 2.2, respectively. The figure shows that four subsystems are interconnected through the coupling variable $\mathbf{y}$, represented by solid lines, and coupling design/constraint functions f_0, $\mathbf{g}_0$, $\mathbf{h}_0$, represented by dashed lines. As in Wagner and Papalambros (1993), a cell of the functional dependency table is shaded if the function of the associated row

depends on the variable of the corresponding column. Without these coupling variables and/or coupling functions, Problem (2.3) can be completely partitioned into M subsystems, each associated with one discipline. If these coupling variables and/or functions are present, a coordination method should be included to ensure their values to be equal to one another.

In addition, functional dependency tables were used to present the strength of the interactions among subsystems de Wit and van Keulen (2010) by identifying four patterns. For the multilevel decomposition, the first pattern illustrates a functional dependency table where the objective function is only associated with the top-level subsystem. The second illustrates the case where there is a smaller number of coupling variables shared among the objective and constraints. The third illustrates that a larger number of coupling variables are shared among the objective and constraints, while the fourth illustrates that the objective and constraints depend on all design variables and the analysis of responses of all subsystems.

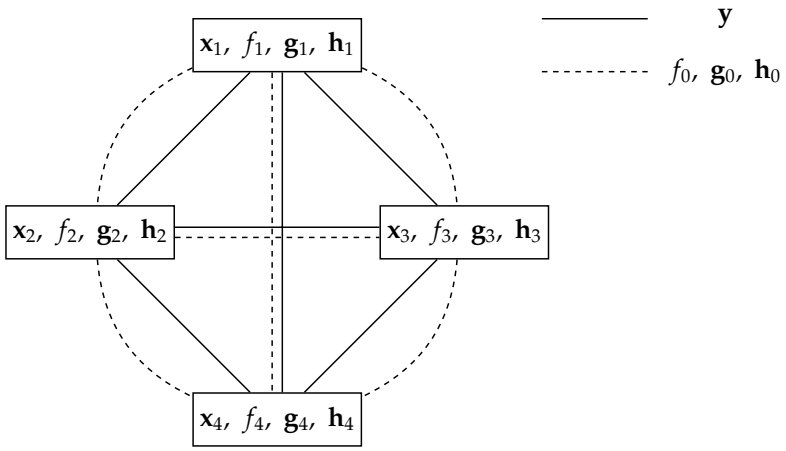


Figure 2.1: Distribution of variables and functions of Problem (2.3) (Tosserams et al., 2008a)

MDO methods are typically defined for optimization problems that belong to one of these four patterns. These MDO methods coordinate data from the individual subsystems such that the optimal solution to the entire system is achieved. Research into

Table 2.2: Functional dependency table of Problem (2.3) (Tosserams et al., 2008a)

	$\mathbf{y}$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
f_0	■	■	■	■	■
f_1	■	■			
f_2	■		■		
f_3	■			■	
f_4	■				■
$\mathbf{g}_0,\ \mathbf{h}_0$	■	■	■	■	■
$\mathbf{g}_1,\ \mathbf{h}_1$	■	■			
$\mathbf{g}_2,\ \mathbf{h}_2$	■		■		
$\mathbf{g}_3,\ \mathbf{h}_3$	■			■	
$\mathbf{g}_4,\ \mathbf{h}_4$	■				■

finding optimal problem decompositions to reduce the effort of coordination has been conducted by, among others, Bloebaum (1995), Chen and Li (2005), and Allison et al. (2007). The criteria for classifying MDO methods are discussed in the next section.

2.4 Classification Criteria

The primary goal of this chapter is to classify various MDO methods. In particular, this classification aims to categorize these architectures by analyzing their similarities and differences. The classification of these MDO methods is determined based on the following three steps of the solution process.

Step 1: Introduction of Copies of Coupling Variables

The first step relates to whether or not the copies of coupling variables $\mathbf{y}_j$ will be added to the set of design variables of subsystem j. If a problem formulation introduces them, consistency constraints must be added to the set of constraints. The residuals of these consistency constraints denote the inconsistencies between the target coupling variables $\mathbf{y}_j^t$ and the auxiliary coupling variables $\mathbf{y}_j$ of subsystem j. If the residuals of consistency constraints are not zero, only the individual discipline feasibility can be maintained during the iterative solution process. Conversely, if a problem statement does not use copies of coupling variables, the introduction of consistency constraints is not needed,

and the multidisciplinary feasibility can be maintained during the iterative solution process.

In general, there are two formulations used frequently to maintain the consistency between the coupling variable and its copy. These formulations, referred to as consistency constraints, can be expressed as

$$\text{Strong consistency constraint:} \quad \mathbf{c} = \mathbf{y}^t - \mathbf{y} = \mathbf{0} \tag{2.4}$$

$$\text{Weak consistency constraint:} \quad \boldsymbol{\lambda}^T \mathbf{c} = \boldsymbol{\lambda}^T \left(\mathbf{y}^t - \mathbf{y}\right) = \mathbf{0}. \tag{2.5}$$

Equality constraint (2.4) is often referred to as the strong consistency constraint since it forces the values of responses to fulfill their targets at each iteration. Equality constraint (2.5) is called the weak consistency constraint, which equals to zero only if the Lagrangian multiplier λ used to relax the consistency, achieves its optimal value, meaning this inconsistency approaches zero (de Wit and van Keulen, 2010).

Step 2: Relaxation of Constraints

The second step relates to whether or not the relaxation of the two types of constraints, design constraints ($\mathbf{g}_0$, $\mathbf{h}_0$, $\mathbf{g}_j$, and $\mathbf{h}_j \ \forall \ j = 1, \cdots, M$) and consistency constraints $\left(\mathbf{c}_j \ \forall \ j = 1, \cdots, M\right)$, will be added to the objective function. Typically, the relaxed constraints, referred to as open constraints, enforce the feasibility only at convergence, whereas the remaining constraints ensuring the feasibility at each iteration are called closed constraints (Alexandrov and Lewis, 1999, Tosserams et al., 2008a).

Nested and Alternating Formulation Structures

There is a difference between the definition of design constraints and consistency constraints: Design constraints are typically linear or nonlinear constraints introduced to represent the design limits of a subsystem, and consistency constraints are typically linear constraints introduced to decouple subsystems. Relaxing consistency constraints means

that the coupled subsystems may obtain different values for the same coupling variable. Similarly, relaxing design constraints implies that they can be violated at a subsystem solution.

To illustrates the open and closed constraints, the problems in Table 2.3 are presented based on nested and alternating formulation structures (Tosserams et al., 2008a). Then, the distinction between nested and alternating formulation structures is summarized in Table 2.4 based on six features, involving 1) problem formulation, 2) difficulty of solving the master system, 3) degree of strict assumptions for achieving convergence proofs, 4) use of gradient-based algorithms for solving the master system, 5) possibility of using approximations to improve the computational efficiency of solving subsystems, and 6) additional computation when using penalty functions to relax complicate constraints.

Table 2.3: Summary of the relaxation of design and consistency constraints

Constraint Relaxation		Formulation Structures	
Design	Consistency	Nested	Alternating
Closed	Closed	$\min_{\mathbf{y}} \quad f(\mathbf{y})$ s.t. $\quad \mathbf{g}_0(\mathbf{y}) \le \mathbf{0}$ $\mathbf{h}_0(\mathbf{y}) = \mathbf{0}$ $b_j(\mathbf{y}) = 0 \quad j = 1, \cdots, M$ where $\quad b_j(\mathbf{y}) = \begin{cases} 0 & \text{if} \quad \exists \mathbf{x}_j : \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \le \mathbf{0},\ \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \\ 1 & \text{if} \quad \nexists \mathbf{x}_j : \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \le \mathbf{0},\ \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0} \end{cases}$	Master: $\min_{\mathbf{y}} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M)$ s.t. $\quad \mathbf{g}_j(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) \le \mathbf{0} \quad j = 0, \cdots, M$ $\mathbf{h}_j(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) = \mathbf{0} \quad j = 0, \cdots, M$ Sub: $\min_{\mathbf{x}_j} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M)$ s.t. $\quad \mathbf{g}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) \le \mathbf{0} \quad i = 0, \cdots, M$ $\mathbf{h}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) = \mathbf{0} \quad i = 0, \cdots, M$
Closed	Open	$\min_{\mathbf{y}} \quad f(\mathbf{y})$ s.t. $\quad \mathbf{g}_0(\mathbf{y}) \le \mathbf{0}$ $\mathbf{h}_0(\mathbf{y}) = \mathbf{0}$ $\mathbf{c}_j = (\mathbf{y} - \mathbf{y}_j^*) = \mathbf{0}, \quad j = 1, \cdots, M$ where $\quad \mathbf{y}_j^*(\mathbf{y}), \mathbf{x}_j^*(\mathbf{y}) = \begin{array}{cc} \arg\min_{\mathbf{y}_j, \mathbf{x}_j} & \phi_j(\mathbf{y} - \mathbf{y}_j) \\ \text{s.t.} & \mathbf{g}_j(\mathbf{y}_j, \mathbf{x}_j) \le \mathbf{0} \\ & \mathbf{h}_j(\mathbf{y}_j, \mathbf{x}_j) = \mathbf{0} \end{array}$	Master: $\min_{\mathbf{y}} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) + \sum_{j=1}^{M} \phi_j(\mathbf{y} - \mathbf{y}_j)$ s.t. $\quad \mathbf{g}_j(\mathbf{y}) \le \mathbf{0}$ $\mathbf{h}_j(\mathbf{y}) = \mathbf{0}$ Sub: $\min_{\mathbf{y}_j, \mathbf{x}_j} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) + \phi_j(\mathbf{y} - \mathbf{y}_j)$ s.t. $\quad \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \le \mathbf{0}$ $\mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0}$
Open	Closed	$\min_{\mathbf{y}} \quad f(\mathbf{y})$ s.t. $\quad \mathbf{g}_0(\mathbf{y}) \le \mathbf{0}$ $\mathbf{h}_0(\mathbf{y}) = \mathbf{0}$ $\phi_j^* \le 0, \quad j = 1, \cdots, M$ where $\quad \phi_j^* = \min_{\mathbf{x}_j} \{\phi_j(\mathbf{g}_j(\mathbf{y}, \mathbf{x}_j), \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j))\}$	Master: $\min_{\mathbf{y}} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) +$ $\sum_{j=0}^{M} \phi_j(\mathbf{g}_j(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M), \mathbf{h}_j(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M))$ Sub: $\min_{\mathbf{x}_j} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) +$ $\sum_{i=0}^{M} \phi_i(\mathbf{g}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M), \mathbf{h}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M))$
Open	Open	$\min_{\mathbf{y}} \quad f(\mathbf{y})$ s.t. $\quad \mathbf{g}_0(\mathbf{y}) \le \mathbf{0}$ $\mathbf{h}_0(\mathbf{y}) = \mathbf{0}$ $\phi_j^* \le 0, \quad j = 1, \cdots, M$ where $\quad \phi_j^* = \min_{\mathbf{y}_j, \mathbf{x}_j} \{\phi_j(\mathbf{y} - \mathbf{y}_j, \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j), \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j))\}$	Master: $\min_{\mathbf{y}} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) + \sum_{j=1}^{M} \phi_j(\mathbf{y} - \mathbf{y}_j) +$ $\theta_0(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M), \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M))$ Sub: $\min_{\mathbf{y}_j, \mathbf{x}_j} \quad f(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M) + \sum_{j=1}^{M} \phi_j(\mathbf{y} - \mathbf{y}_j) +$ $\theta_0(\mathbf{g}_0(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M), \mathbf{h}_0(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M)) +$ $\theta_i(\mathbf{g}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M), \mathbf{h}_i(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M))$

Table 2.4: Features of nested and alternating formulations

	Nested Formulations	Alternating Formulations
Apply to	Problem (2.1)	Problem (2.3)
Advantages	Local optima The use of single-point approximations for subsystems The use of multi-point approximations for subsystems	Differentability of optimization problems Well-posedness of optimization problems The use of gradient-based algorithms The use of approximations based on gradient information The use of multi-point approximations
Disadvantages	Master non-differentability Master ill-posedness A large number of consistency constraints Difficult use of gradient-based algorithms Highly non-trivial approximations of subsystems	Requirement of differentiable objectives Requirement of convex constraint sets Strict assumptions for local convergence of parallel schemes Additional updates of penalty parameters Non-differentiable penalty functions

Hierarchical and Nonhierarchical Decomposition Structures

A multilevel problem typically has a hierarchical structure as shown in Figure 2.2 (Tosserams et al., 2006b). The top-level, referred to as level 1, involves the parent subsystem P_{11}. One level lower, i.e. at level 2, the children subsystems of P_{11} can be found. The subsystems located at level 2 are numerated from left to right. At the third level, the children of level 2 subsystems can be found.

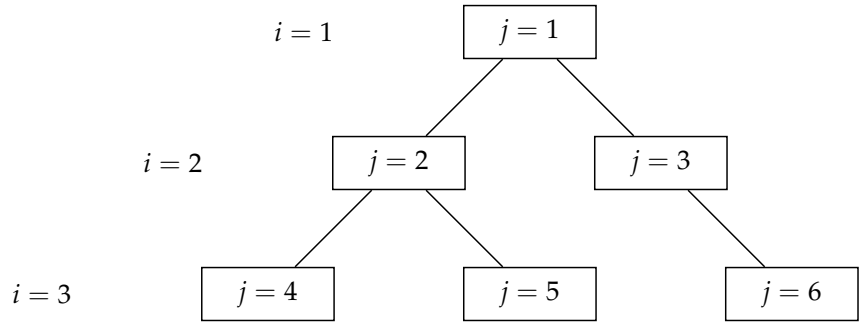


Figure 2.2: Multi-level structure for a three-level hierarchy (Tosserams et al., 2006b)

For solving multilevel problems, two solution sequences, top-down and bottom-up, can be employed to communicate the results of the coupling variables. Figure 2.3 depicts these two schemes for three-level problems by integrating the nested and alternating formulation structures. In the alternating top-down scheme of Figure 2.3 (a), the subsystems at level 1 converge and pass their estimates of coupling variables to subsystems at level 2 and then level 3. This process is repeated until the three subsystems have collaboratively converged to the original optimal solution. The alternating bottom-up scheme of Figure 2.3 (c) is the mirror image of the alternating top-down scheme, i.e. the subsystems at level 3 converge first and then pass their estimates of the coupling variables to the subsystems at level 2 and then level 1. The nested structure of these two schemes requires the convergence of the subsystems outlined in the dashed lines before sending their estimates of coupling variables to subsystems at the bottom-level or top-level.

The hierarchical nature of MDO methods may not be the most suitable approach for decomposing complex problems that do not have a clear hierarchical structure. For

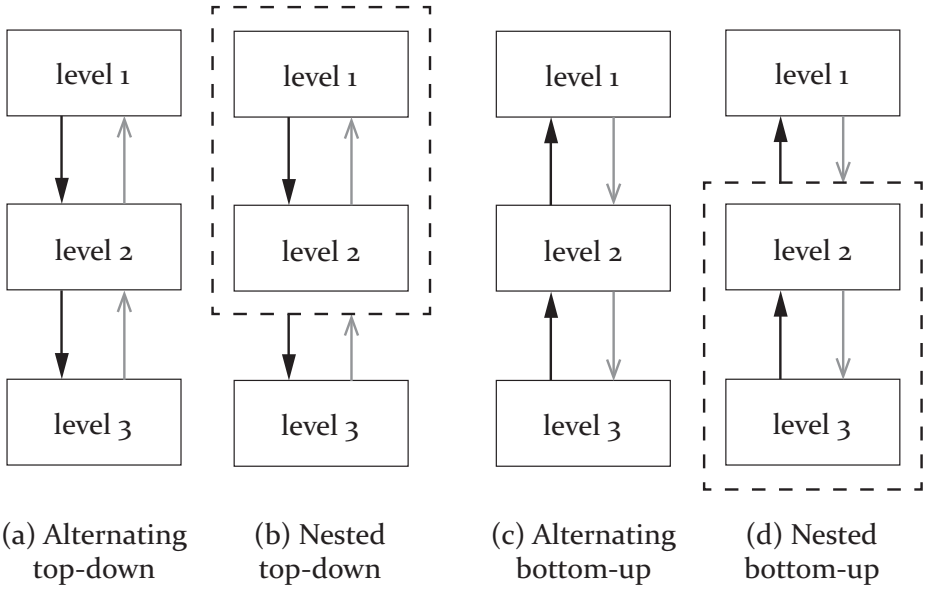


Figure 2.3: Convergence coordination schemes for decomposed optimization problems

those problems, direct communication between subsystems may be more appropriate. For example, classic MDO problems are typically composed of subsystems ordered by analysis disciplines among which no clear hierarchy may exist. These classic MDO methods concern the situation in which subsystems do not prescribe estimates of coupling variables based on one another. Instead, coupling variables are estimated using sensitivity information, e.g. global sensitivity equations (Sobieszczanski-Sobieski, 1988). By introducing these equations to monitor the consistency, the strong consistency constraints do not need to be relaxed to the objective, and the parallel computing of subsystems can be realized by exchanging information only found among subsystems. However, using sensitivity analysis to monitor the consistency is computational inefficient. Therefore, one goal of this dissertation is to extend hierarchical decomposition based on relaxation techniques to include nonhierarchical target response couplings of subsystems such that the nonhierarchical communication between distributed subsystems seen in Figure 2.4 is realized.

Figure 2.5 presents an integrated overview based on two types of consistency constraints, i.e. strong and weak consistency constraints. Strong consistency constraints

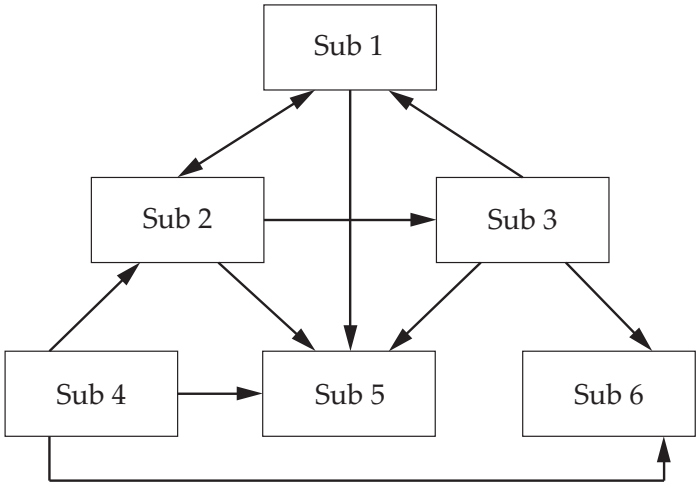


Figure 2.4: Nonhierarchical information flow of distributed subsystems

are accomplished using global sensitivity equations, and weak consistency constraints are accomplished using three typical approaches, involving Lagrangian relaxation, penalty relaxation, and augmented Lagrangian relaxation. Both decomposition techniques can be further decomposed into hierarchal and nonhierarchical formulations.

Figure 2.5: Summary of the decomposition process

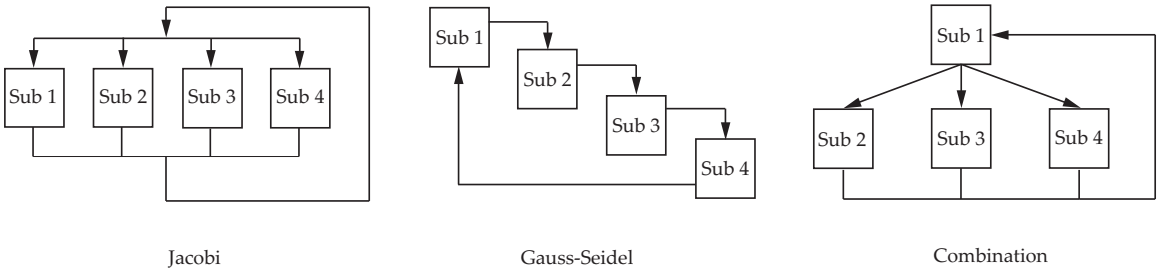


Figure 2.6: Three schemes of solution sequences

Step 3: Solution Sequence

The third step determines the order for solving each subsystem, either nested or alternating. In a nested formulation, all disciplinary subsystems are required to converge to an optimal solution, and then the master problem evaluates the objective and constraint functions and solves for an overall optimal solution. In an alternating formulation, however, an iterative process is implemented between solving the master problem and disciplinary subsystems. For both the nested and alternating formulation structures, disciplinary subsystems can be solved using a Jacobi scheme, a Gauss-Seidel scheme, or a combination of both. A Jacobi scheme solves all subsystems in parallel, after which their solutions are updated. The Gauss-Seidel scheme solves all subsystems sequentially and updates their solutions as soon as they become available. These three schemes are depicted in Figure 2.6.

2.5 Concluding Remarks

This chapter identified three steps used to classify MDO methods. Introduction of the copies of coupling variables was analyzed by constructing two types of consistency constraints, i.e. strong and weak. Furthermore, based on these two types, the nested and alternating problem formulations were presented, which were categorized as hierarchical and nonhierarchical decomposition structures. Finally, the solution sequence using a Jacobi scheme, a Gauss-Seidel scheme, or a combination of both discussed the analysis

presented in this chapter. We can conclude that the multidisciplinary design optimization methods are practical for solving complex systems, when the amount of data exchanged between individual subsystems remains small, and these subsystems are weakly coupled.

Chapter 3

An Overview of Augmented Lagrangian and Duality

3.1 Introduction

The augmented Lagrangian method is an extension of the quadratic penalty method and Lagrangian method (Bazaraa et al., 2006, Bertsekas, 2003), which are applied in developing MDO distributed optimization structures such as Collaborative Optimization (CO) (Braun et al., 1997), Analytical Target Cascading (ATC) (Kim et al., 2003, Lassiter et al., 2005, Michelena et al., 2003), Exact and Inexact Penalty Decomposition (EPD/IPD) (Demiguel and Murray, 2006). The augmented Lagrangian method reduces the possibility of ill-conditioning of the subsystems by introducing Lagrangian multiplier estimates.

This chapter overviews the rational for using the augmented Lagrangian method to solve decomposable engineering system design problems. In Section 3.2, the augmented Lagrangian dual formulation is presented. Then, its augmented Lagrangian dual problem is defined in Section 3.3 and the strong and weak duality are discussed. Section 3.3 presents the first-order optimality conditions for the constrained optimization problem by analyzing the dual and primal feasibility. Section 3.4 discusses the significance of Lagrangian multipliers. Section 3.6 illustrates the augmented Lagrangian coordination with respect to the introduction of copies of coupling variables and the relaxation of con-

sistency constraints. Finally, the duality theorem and the augmented Lagrangian method are concluded in Section 3.7.

3.2 The Augmented Lagrangian Dual Function

In constrained optimization, it is often possible to convert the primal problem, i.e. the original form of the optimization problem, to a dual problem (Bazaraa et al., 2006). The fact motivates the following general philosophy: since dual problem captures the properties of the primal problem, then if the primal optimal value is the same as the dual optimal value, the behavior of the primal problem may be analyzed via its dual counterpart. For a large-scale engineering system design problem, we can apply the decomposition to the problem after introducing dual variables and working with the dual problem.

For expressing an optimization problem without the convexity assumption, a minimization problem is given by

$$\begin{aligned} \min_{\mathbf{x}\in\Omega} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned} \tag{3.1}$$

where the objective function is denoted by $f : \mathbb{R}^n \mapsto \mathbb{R}$, the vector of equality constraint functions $\mathbf{h} : \mathbb{R}^n \mapsto \mathbb{R}^m$, and the vector of design variables $\mathbf{x}$ with $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$, $\mathbf{l} < \mathbf{u}$ and $\Omega = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{l} < \mathbf{x} < \mathbf{u}\}$. The optimal objective value is denoted by $f(\mathbf{x}^\star)$ with respect to the optimal solution $\mathbf{x}^\star$. Assuming that $\mathbf{h}$ has continuous first derivatives on Ω, then its first derivatives are denoted as

$$\nabla \mathbf{h}(\mathbf{x}) = (\nabla h_1(\mathbf{x}), \cdots, \nabla h_m(\mathbf{x})) = \mathbf{h}^{'}(\mathbf{x}) \in \mathbb{R}^{n\times m}. \tag{3.2}$$

Since the discussion in this research only concerns with the relaxation of the consistency constraints, which are equality constraints, the minimization problem is considered only

including the equality constraints.

Lagrangian duality with the elimination of equality constraints is realized by augmenting the objective function with a weighted sum of the equality constraint functions. The augmented Lagrangian $\mathcal{L} : \mathbb{R}^{n\times m} \mapsto \mathbb{R}$ for Problem (3.1) is defined as

$$\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \frac{1}{2}\sum_{i=1}^{p} \rho_i h_i(\mathbf{x})^2. \tag{3.3}$$

The vectors comprising of Lagrangian multipliers and the penalty parameters are defined as $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m] \in \mathbb{R}^m$ and $\boldsymbol{\rho} = [\rho_1, \cdots, \rho_m] \in \mathbb{R}^m_{>0}$, respectively. The Lagrangian multiplier $\boldsymbol{\lambda}$ is also referred to as the dual variable for Problem (3.1). In this research, the augmented Lagrangian function instead of the Lagrangian function or the penalty function is used, because through the augmented Lagrangian, ill-conditioning of the optimization problem can be avoided by using an appropriate strategy to find $\boldsymbol{\lambda}$ arbitrarily close to the its optimal solution $\boldsymbol{\lambda}^\star$ while keeping the penalty parameter $\boldsymbol{\rho}$ relatively small. Unless stated otherwise, the reader is referred to Bazaraa et al. (2006) and Bertsekas (2003) for the following discussion of augmented Lagrangian relaxation techniques.

The augmented Lagrangian dual function $q : \mathbb{R}^m \mapsto \mathbb{R}$ is defined as the minimum value of the augmented Lagrangian over $\mathbf{x} \in \Omega$. For $\boldsymbol{\lambda} \in \mathbb{R}^m$,

$$q(\boldsymbol{\lambda}) = \inf_{\mathbf{x}\in\Omega} \left\{ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \frac{1}{2}\sum_{i=1}^{m} \rho_i h_i(\mathbf{x})^2 \right\}. \tag{3.4}$$

When the Lagrangian dual function is unbounded below with respect to the design variable $\mathbf{x}$, $q(\boldsymbol{\lambda}) = -\infty$. Since the dual function is the pointwise infimum of a family of affine functions of $\boldsymbol{\lambda}$, it is concave when Problem (3.1) is nonconvex (Bertsekas, 2003).

The dual function yields a lower bound $q(\boldsymbol{\lambda})$ on the optimal value $f(\mathbf{x}^*)$ of Problem (3.1) (Bazaraa et al., 2006, Boyd and Vandenberghe, 2004). For any $\boldsymbol{\lambda}$, the relationship between $q(\boldsymbol{\lambda})$ and $f(\mathbf{x}^\star)$ is given by

$$q(\boldsymbol{\lambda}) \leq f(\mathbf{x}^\star). \tag{3.5}$$

This important property can be easily verified. If $\widetilde{\mathbf{x}}$ refers to a feasible point for Problem (3.1), i.e., $h_i(\widetilde{\mathbf{x}}) = 0$ and any $\boldsymbol{\lambda}$. Then we have

$$\sum_{i=1}^{p} \lambda_i h_i(\widetilde{\mathbf{x}}) + \frac{1}{2} \sum_{i=1}^{m} \rho_i h_i(\widetilde{\mathbf{x}})^2 \leq 0, \tag{3.6}$$

since each term in the first sum is nonpositive, and each in the second sum is zero. Thereby,

$$\mathcal{L}_\rho(\widetilde{\mathbf{x}}, \boldsymbol{\lambda}) = f(\widetilde{\mathbf{x}}) + \sum_{i=1}^{m} \lambda_i h_i(\widetilde{\mathbf{x}}) + \frac{1}{2} \sum_{i=1}^{m} \rho_i h_i(\widetilde{\mathbf{x}})^2 \leq f(\widetilde{\mathbf{x}}). \tag{3.7}$$

Based on the above Equation (3.7), then

$$q(\boldsymbol{\lambda}) = \inf_{\mathbf{x} \in \Omega} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) \leq \mathcal{L}_\rho(\widetilde{\mathbf{x}}, \boldsymbol{\lambda}) \leq f(\widetilde{\mathbf{x}}). \tag{3.8}$$

The inequality (3.5) holds except when $q(\boldsymbol{\lambda}) = -\infty$. As a result, the dual function can give a lower bound for the primal objective value $f(\mathbf{x}^\star)$ if $q(\boldsymbol{\lambda}) > -\infty$.

3.3 The Augmented Lagrangian Dual Problem

For any $\boldsymbol{\lambda}$, the Lagrangian dual function gives a lower bound on the optimal value $f(\mathbf{x}^\star)$ of the optimization problem (3.1). Thus we have a lower bound that depends on some parameters $\boldsymbol{\lambda}$. A natural question is: What is the best lower bound that can be obtained from the Lagrangian dual function? This leads to an optimization problem, i.e.,

$$\max_{\boldsymbol{\lambda}} \quad q(\boldsymbol{\lambda}) \tag{3.9}$$

This Lagrangian dual problem is associated with Problem (3.1), which is referred to as the primal problem in this context. Using the term dual feasible to describe $\boldsymbol{\lambda}$ with $q(\boldsymbol{\lambda}, \boldsymbol{\nu}) > -\infty$ means that $\boldsymbol{\lambda}$ is a feasible solution for the dual problem (3.9); $\boldsymbol{\lambda}^\star$ is referred to as a dual optimum or an optimal Lagrangian multiplier if it is an optimal solution for Problem (3.9). If the primal problem (3.1) is nonconvex, the Lagrangian dual prob-

lem (3.9) can still be transformed to a convex optimization problem, since the objective to be maximized is concave and the constraint is convex.

The example above shows that frequently the domain of the dual function

$$\text{dom } q = \{\boldsymbol{\lambda} |\ q(\boldsymbol{\lambda}) > -\infty\} \tag{3.10}$$

has a dimension smaller than p. In many cases, the affine hull of dom q can be described as a set of linear equality constraints, meaning that the equality constraints that are “hidden" or “implicit" in the objective q of the dual problem (3.9) can be identified. In this case, an equivalent problem in which these equality constraints are given explicitly as constraints can be formed.

Weak Duality The optimal value of the Lagrangian dual problem denoted as $q^\star$ is, by definition, the best lower bound on $f^\star$ that can be obtained from the Lagrangian dual function. The relationship between the optimal dual objective and the optimal primal objective is referred to as

$$q(\boldsymbol{\lambda}^\star) \leq f(\mathbf{x}^\star), \tag{3.11}$$

which holds even if the primal problem is nonconvex. This property is called weak duality, which holds even if $q^\star$ and $f^\star$ are infinite. If the primal problem is unbounded below so that $f^\star = -\infty$, then $q^\star = -\infty$, i.e. the Lagrangian dual problem is infeasible; conversely, if the dual problem is unbounded above so that $q^\star = \infty$, then $f^\star = \infty$, i.e. the primal problem is infeasible. If the primal problem is bounded below, then the term $f(\mathbf{x}^\star) - q(\boldsymbol{\lambda}^\star)$ is often referred to as the optimal duality gap, indicating the discrepancy between the primal optimal value and its best lower bound.

Strong Duality If the relationship

$$q(\boldsymbol{\lambda}^\star) = f(\mathbf{x}^\star), \tag{3.12}$$

holds, i.e., the optimal duality gap is zero, then a property called strong duality holds, meaning that the best lower bound obtained by solving the augmented Lagrange dual problem is exactly the same as the optimal primal objective. Strong duality is not, in general, present. However, if the primal problem (3.1) is convex, i.e. of the form

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \boldsymbol{A}\mathbf{x} = \mathbf{b}, \end{aligned} \tag{3.13}$$

with $f(\mathbf{x})$ convex, usually (but not always), strong duality is exhibited.

3.4 First-Order Optimality Conditions

Sub-optimality and stopping criteria If a dual feasible variable $\boldsymbol{\lambda}$ can be found, a lower bound on the optimal value of the primal problem $q(\boldsymbol{\lambda}) \leq f(\mathbf{x}^\star)$ is exhibited. Thus, a dual feasible point $\boldsymbol{\lambda}$ provides a design such that $q(\boldsymbol{\lambda}) \leq f(\mathbf{x}^\star)$, and strong duality indicates there exist many feasible and optimal designs (Bazaraa et al., 2006). When $\mathbf{x}$ is primal feasible and $\boldsymbol{\lambda}$ is dual feasible, then

$$f(\mathbf{x}) - f(\mathbf{x}^\star) \leq f(\mathbf{x}) - q(\boldsymbol{\lambda}). \tag{3.14}$$

If the value of $f(\mathbf{x}) - q(\boldsymbol{\lambda})$ is set to $\varepsilon > 0$, the inequality (3.14) establishes that $\mathbf{x}$ is only ε-suboptimal for the primal problem, and $\boldsymbol{\lambda}$ is only ε-suboptimal for the dual problem (Bertsekas, 2003). If $f(\mathbf{x}) - q(\boldsymbol{\lambda}) = 0$, then $\mathbf{x}$ is the primal optimal solution and $\boldsymbol{\lambda}$ is the dual optimal solution.

The relationships between $f(\mathbf{x})$ and $q(\boldsymbol{\lambda})$ can be used in providing the stopping criteria for optimization algorithms. If an algorithm produces a sequence of primal and dual feasible points such as $\mathbf{x}^k$ and $\boldsymbol{\lambda}^k$ for $k = 1, 2, \cdots$, and $\varepsilon > 0$ is a given termination tolerance, then the stopping criterion is given by

$$f(\mathbf{x}^k) - q(\boldsymbol{\lambda}^k) \leq \varepsilon, \tag{3.15}$$

which guarantees that $\mathbf{x}^k$ is ε-suboptimal when the algorithm terminates (Bazaraa et al., 2006).

Karush-Kuhn-Tucker Conditions Based on the above discussions, the functions f, g_i, $\cdots$, g_m, h_1, $\cdots$, h_p are continuous and differentiable, but not necessarily convex; the design variables $\mathbf{x}^\star \in \Omega$ and $\boldsymbol{\lambda}^\star \in \mathbb{R}^m$ are any primal and dual optimal points with zero duality gap. Since $\mathbf{x}^\star$ minimizes the augmented Lagrangian function $\mathcal{L}_\rho(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ over Ω, then if $F = \nabla f$, its gradient must vanish at $\mathbf{x}^\star$, i.e.

$$\frac{\partial \mathcal{L}_\rho}{\partial \mathbf{x}} = F(\mathbf{x}^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla h_i(\mathbf{x}^\star) + \sum_{i=1}^{m} \rho_i h_i(\mathbf{x}^\star) \nabla h_i(\mathbf{x}) = 0. \tag{3.16}$$

Thus, the Karush-Kuhn-Tucker (KKT) conditions for Problem (3.1) are defined as

$$\text{Dual Feasibility:} \quad \nabla f(\mathbf{x}^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla h_i(\mathbf{x}^\star) + \sum_{i=1}^{m} \rho_i h_i(\mathbf{x}^\star) \nabla h_i(\mathbf{x}^\star) = 0, \tag{3.17}$$

$$\text{Primal Feasibility:} \quad h_i(\mathbf{x}^\star) = 0, \quad i = 1, \cdots, m \tag{3.18}$$

For any optimization problem with differentiable objective and constraint functions exhibiting strong duality, any pair of primal and dual optimal points must satisfy the KKT conditions (3.17). If the primal problem is convex, the KKT conditions are also sufficient for the points to be globally primal or dual optimal. In other words, if f is convex, h_i is an affine, and $\widetilde{\mathbf{x}}$, $\widetilde{\boldsymbol{\lambda}}$, and $\widetilde{\nu}$ are any points that satisfy the KKT conditions

$$\text{Dual Feasibility:} \quad \nabla f(\widetilde{\mathbf{x}}) + \sum_{i=1}^{m} \widetilde{\lambda}_i \nabla h_i(\widetilde{\mathbf{x}}) + \sum_{i=1}^{m} \rho_i h_i(\widetilde{\mathbf{x}}) \nabla h_i(\widetilde{\mathbf{x}}) = 0, \tag{3.19}$$

$$\text{Primal Feasibility:} \quad h_i(\widetilde{\mathbf{x}}) = 0, \; i = 1, \cdots, m \tag{3.20}$$

then $\widetilde{\mathbf{x}}$ and $\widetilde{\boldsymbol{\lambda}}$ are global optimal solutions for the primal and dual problems, respectively (Bertsekas, 2003).

For strong duality and a dual optimal solution $(\boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$, then any primal optimal

point is also a minimizer of $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$. This fact sometimes allows for the computation of a primal optimal solution via solving the dual problem. More precisely, strong duality and an dual optimal solution $(\boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ are known, and the minimizer of $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$, i.e. the solution of

$$\min \quad f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i^\star h_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i^\star h_i(\mathbf{x})^2, \tag{3.21}$$

is unique, then if the solution of problem (3.21) is primal feasible, it must be primal optimal; if it is not primal feasible, then no primal optimal point can be achieved (Bertsekas, 2003).

3.5 Significance of Lagrangian Multipliers

Lagrange multipliers are a very useful technique in multivariable calculus, but all too often their significance is not well understood. This section will make the concept and the applications of Lagrangian multipliers clearer.

First, a case where an optimization problem depends on a parameter a is given by

$$\begin{aligned} \min_{\mathbf{x}\in\Omega} \quad & f(\mathbf{x}; a) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}; a) = \mathbf{0}. \end{aligned} \tag{3.22}$$

The Lagrangian function for Problem (3.22) is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}; a) = f(\mathbf{x}; a) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}; a). \tag{3.23}$$

Then, the KKT conditions with respect to $\mathcal{L}$ gives that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{h}}{\mathbf{x}} = \mathbf{0}, \tag{3.24}$$

which represents the relationship between the first-order derivative of the objective func-

tion and the first-order derivative of the equality constraints, i.e.

$$\frac{\partial f^T}{\partial \mathbf{x}} = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{h}^T}{\mathbf{x}}. \tag{3.25}$$

In multivariable calculus, the partial first-order derivative of a function such as $f(\mathbf{x})$ or $\mathbf{h}(\mathbf{x})$ (written as $\frac{\partial}{\partial \mathbf{x}}$) forms a normal vector to a curve (in two dimensions, i.e. $\mathbf{x} \in \mathbb{R}^2$) or a surface (in higher dimensions), and that a particular point on these partial first-order derivatives are constants. The length of the normal vector does not matter, meaning that any constant multiplying $\frac{\partial \mathbf{h}^T}{\mathbf{x}}$ does not change the direction of the normal vector. For Problem (3.22), the normal vectors of functions $f(\mathbf{x}; a)$ and $\mathbf{h}(\mathbf{x}; a)$ are parallel. Thus, the normal vector of the objective is written as a linear combination of the normal vectors of the equality constraints. The inclusion of the unknown constant multiplier $\boldsymbol{\lambda}$ is necessary because the magnitudes of the normal vectors may be different.

Since the optimization problem also depends on the parameter a, then the first-order derivative of the objective function with respect to a is calculated as

$$\frac{df}{da} = \frac{\partial f}{\partial a} + \frac{\partial f^T}{\partial \mathbf{x}} \frac{d\mathbf{x}}{da}. \tag{3.26}$$

By substituting Equation (3.25) into Equation (3.26), the Lagrangian multipliers can be used to describe the sensitivity of the objective to changes in the equality constraints. In equations, this statement reads

$$\frac{df}{da} = \frac{\partial f}{\partial a} + \boldsymbol{\lambda} \frac{\partial \mathbf{h}^T}{\partial a}. \tag{3.27}$$

The intuitive meanings of the Lagrangian multipliers can be interpreted in both physics and economics. A tutorial "The Introduction to Lagrangian Multipliers[1]" introduces that the equality constraint function $\mathbf{h}(\mathbf{x})$ can be considered as competing with the objective function $f(\mathbf{x})$ to pull the point $\mathbf{x}$ to its minimum. During the pulling process, the Lagrange multiplier $\boldsymbol{\lambda}$ can be considered as a measure of how difficult $\mathbf{h}(\mathbf{x})$ has to pull the

[1]The sources is obtained from http://www.slimy.com/ steuard/teaching/tutorials/Lagrange.html

point in order to achieve an equilibrium of the forces on the constraint curve (or surface). This tutorial also emphasizes that in physics based on Lagrange multipliers in the calculus of variations, the Lagrangian multiplier $\boldsymbol{\lambda}$ is the force of constraint. Moreover, the economic meaning of the Lagrange multiplier $\boldsymbol{\lambda}$ is interpreted in a tutorial "Optimization with An Equality Constraint: Interpretation of Lagrange Multiplier[2]". When the objective of the problem is to minimize the overall cost subject to a limited resource, then $\boldsymbol{\lambda}$ can be considered as either marginal value or shadow price of the resource. By conclusion, the rate or sensitivity at which the value of the objective $f(\mathbf{x})$ changes along with the constraint $\mathbf{h}$ is represented by the value of the Lagrange multiplier.

3.6 Augmented Lagrangian Coordination

The augmented Lagrangian relaxation techniques discussed in the previous sections provide a basis for solving large-scale, complex system design problems. To determine an optimal design of the system, the multidisciplinary design optimization (MDO) approach (Cramer et al., 1994) is adopted and is given by:

$$\begin{aligned} \min_{\mathbf{z}=\left[\mathbf{y}^T,\mathbf{x}_1^T,\cdots,\mathbf{x}_M^T\right]^T} \quad & f_0\left(\mathbf{y},\mathbf{x}_1,\cdots,\mathbf{x}_M\right)+\sum_{j=1}^{M} f_j(\mathbf{y},\mathbf{x}_j) \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{y},\mathbf{x}_1,\cdots,\mathbf{x}_M)\leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{y},\mathbf{x}_1,\cdots,\mathbf{x}_M)= \mathbf{0} \\ & \mathbf{g}_j(\mathbf{y},\mathbf{x}_j)\leq \mathbf{0} \quad j=1,\cdots,M \\ & \mathbf{h}_j(\mathbf{y},\mathbf{x}_j)= \mathbf{0} \quad j=1,\cdots,M \end{aligned} \tag{3.28}$$

where the vector of design variables $\mathbf{z}=\left[\mathbf{y}^T,\mathbf{x}_1^T,\cdots,\mathbf{x}_M^T\right]^T\in\mathbb{R}^n$ consists of a number of coupling variables $\mathbf{y}\in\mathbb{R}^{n^y}$, and a number of local variables $\mathbf{x}_j\in\mathbb{R}^{n_j^x}$ associated only with subsystem j, and $n^y+\sum_{j=1}^{M} n_j^x=n$, where n denotes the number of design variables. The linking variables may be common design variables shared by multiple subsystems, and

[2]The source is obtained from http://www.economics.utoronto.ca/osborne/MathTutorial/ILMF.HTM.

interdisciplinary coupling variables that link the analysis models of different subsystems. The coupling objective $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ and coupling design constraints $\mathbf{g}_0 : \mathbb{R}^n \mapsto \mathbb{R}^{m_0^g}$ and $\mathbf{h}_0 : \mathbb{R}^n \mapsto \mathbb{R}^{m_0^h}$ are non-separable and may depend on all design variables $\mathbf{z}$. Local objectives $f_j : \mathbb{R}^{n_j} \mapsto \mathbb{R}$, and local design constraints $\mathbf{g}_j : \mathbb{R}^n \mapsto \mathbb{R}^{m_j^g}$ and $\mathbf{h}_j : \mathbb{R}^n \mapsto \mathbb{R}^{m_j^h}$ are associated exclusively with subsystem j, and may depend on the coupling variables $\mathbf{y}$ and the local variables $\mathbf{x}_j$ of only a single subsystem j, such that $n_j = n^y + n_j^x$. In addition, $m_0^g + \sum_{j=1}^{M} m_j^g = m^g$ and $m_0^h = \sum_{j=1}^{M} m_j^h = m^h$, where m^g denotes the number of inequality constraints and m^h the number of equality constraints. Unless indicated otherwise, all vectors in this chapter are assumed to be column vectors.

Without the coupling variables $\mathbf{y}$ and the coupling functions f_0, $\mathbf{g}_0$, and $\mathbf{h}_0$, i.e. without interactions, the above problem can be partitioned into M smaller subsystems, each associated with one discipline. When coupling variables and/or coupling functions are present, a coordination method is required to guide the individual subsystem designs to achieve an optimal design for the overall system.

To realize the situation in which all coupled subsystems are separated with respect to the design variables of the individual disciplines, i.e., the design of each subsystem depends only on the variables of its discipline, the augmented Lagrangian relaxation techniques are applied by introducing the copies of the coupling variables and/or coupling functions through the following two steps:

Step 1: Introduction of Copies of Coupling Variables The first step of the transformation is to introduce the copies of the coupling variables $\mathbf{y}_j$ for each subsystem, such that consistency constraints $\mathbf{c}_j = \mathbf{y}_i - \mathbf{y}_j,\ i \neq j$ can be used to insure that introduced copies achieve the same values, i.e. $\mathbf{y}_1 = \mathbf{y}_2 = \cdots = \mathbf{y}_M$. As a result, the local constraint functions $\mathbf{g}_j$ and $\mathbf{h}_j$ only depend on the design variables $\mathbf{y}_j$ and $\mathbf{x}_j$ of system j. However, since the consistency constraint $\mathbf{c}$ still depend on the coupling variables of multiple subsystems, the augmented Lagrangian relaxation technique is required to realize the separability of the subsystems.

Step 2: Relaxation of the consistency constraints and/or coupling functions The second step of the transformation is to relax the consistency constraint $\mathbf{c}$ and/or the coupling functions f_0, $\mathbf{g}_0$, and $\mathbf{h}_0$ through an augmented Lagrangian function $\mathcal{L}_\rho$. After the relaxation, the local constraint functions $\mathbf{g}_j$ and $\mathbf{h}_j$ are separable with respect to the design variables $\mathbf{y}_j$ and $\mathbf{x}_j$ of system j. Due to the relaxation, an error between the primal and the dual problems is introduced, and an algorithm should be developed to reduce this error into zero. The resulting relaxed optimization problem for Subsystem j is presented as

$$\begin{aligned}
\min_{\mathbf{x}_j, \mathbf{y}_j} \quad & f_j(\mathbf{y}_j, \mathbf{x}_j) + f_0\left(\mathbf{y}, \mathbf{x}_1, \cdots, \mathbf{x}_M\right) + \\
& \lambda_{\mathbf{y}}^T \mathbf{c}_j + \frac{\rho_{\mathbf{y}}}{2} \|\mathbf{c}_j \circ \mathbf{c}_j\|_2^2 + \\
& \lambda_{\mathbf{h}}^T \mathbf{h}_0 + \frac{\rho_{\mathbf{h}}}{2} \|\mathbf{h}_0 \circ \mathbf{h}_0\|_2^2 + \\
& \lambda_{\mathbf{g}}^T \mathbf{g}_0 + \frac{\rho_{\mathbf{g}}}{2} \|\mathbf{g}_0 \circ \mathbf{g}_0\|_2^2 \\
\text{s.t.} \quad & \mathbf{g}_j(\mathbf{y}, \mathbf{x}_j) \leq \mathbf{0}, \\
& \mathbf{h}_j(\mathbf{y}, \mathbf{x}_j) = \mathbf{0}.
\end{aligned} \tag{3.29}$$

where $\lambda_{\mathbf{h}} \in \mathbb{R}^{m_0^{\mathbf{h}}}$ is the vector of Lagrangian multiplier estimates for the system-wide equality constraints and $\rho_{\mathbf{h}} \in \mathbb{R}^{m_0^{\mathbf{h}}}$ is the vector of penalty parameters; similarly, $\lambda_{\mathbf{g}} \in \mathbb{R}^{m_0^{\mathbf{h}}}$ is the vector of Lagrangian multiplier estimates for the system-wide equality constraints and $\rho_{\mathbf{g}} \in \mathbb{R}^{m_0^{\mathbf{g}}}$ is the vector of penalty parameters. When solving Subsystem j, the copies of coupling variables obtained from other subsystems $i = 1, 2, \cdots, m,\ j \neq j$ are taken as constants, so that all subsystems can be designed independently.

3.7 Concluding Remarks

The duality theory in nonlinear programming sparks a great deal of interest in solving Multidisciplinary Design Optimization (MDO) problems. The first principal ap-

plication of duality theory is that it enables us to derive systems of demand equations which are consistent with minimizing the objective function simply by differentiating an augmented Lagrangian function, as opposed to solving explicitly a constrained minimization problem. The second principal advantage of duality theory is that it enables us to understand in an effortless way the sensitivity of the objective function changes in constraints. During last forty years, several duality formulations that enjoy many of the properties of linear dual programs have been derived, including the Lagrangian dual problem, the conjugate dual problem, the surrogate dual problem, and the mixed Lagrangian and surrogate, or composite dual problem. The reader can refer to the work of Geoffrion (1971) and Karamardian (1971) on various duality formulations and their interrelationships.

For constrained optimization problems, the augmented Lagrangian method and the corresponding algorithm, method of multipliers, were first proposed in the late 1960s by Hestenes (1969) and Powell (1969). Augmented Lagrangian methods were developed in part to bring robustness to the Lagrangian or penalty method, and in particular, to yield convergence by assuming that the optimization problem is locally convexified. Many of the early numerical experiments on the method of multipliers are due to Miele et al. (1971a), Miele et al. (1971b), and Miele et al. (1972). Much of the early work is consolidated in a monograph by Bertsekas (1982), who also discusses similarities to older approaches using Lagrangian or penalty functions (Arrow and Solow, 1958, Arrow et al., 1958, Fiacco and McCormick, 1964), as well as a number of generalizations. For solving MDO problems, the reader can refer to the work of Tosserams et al. (2006a) and Blouin et al. (2005). In order to ensure the decomposability of the MDO problems with the superior convergence properties of the method of multipliers, the alternating direction method of multipliers (ADMOM) is introduced in Tosserams et al. (2006a) and extended in Tosserams (2008).

Chapter 4

Analytical Target Cascading

4.1 Introduction

Analytical target cascading (ATC) is a hierarchical, multilevel, multidisciplinary coordination method used to solve large-scale system design optimization problems (Kim, 2001, Kim et al., 2003). Unlike other multilevel coordination methods (Braun, 1996, Demiguel and Murray, 2006, Haftka and Watson, 2005, Sobieszczanski-Sobieski, 1988, Sobieszczanski-Sobieski et al., 2000), ATC is one with proven solution convergence and equivalence (Bertsekas, 2003, Lassiter et al., 2005, Li et al., 2008). The research presented here continues efforts to solve ATC-decomposable problems based on the Lagrangian duality theory (Bazaraa et al., 2006, Bertsekas, 2003). A dual counterpart of the primal problem is solved by using the subgradient algorithm, the application of which includes five schemes for updating dual variables. Three of these schemes, i.e., K, M, and O updates, have been studied by Lassiter et al. (2005), Blouin et al. (2005), and Kim et al. (2006). Two new schemes, known as the linear cutting plane method (LCP) and the proximal cutting plane method (PCP) (Bertsekas, 2003), are investigated in this chapter to overcome the convergence efficiency weaknesses of the existing subgradient update schemes.

After ATC was formalized (Kim, 2001), researchers for the last ten years have focused on its efficiency for solving engineering problems. For example, Kim et al. (2003)

demonstrated the efficiency of the ATC modeling and solution process in the chassis design of a sports-utility vehicle, and Blouin et al. (2004) applied ATC to the design of a continuously variable transmission for optimum vehicle performance, analyzing the trade-off between subsystem optimality and overall design consistency.

Michalek and Papalambros (2005b) applied a quadratic penalty method (QP) for ATC, requiring large penalty weights in order to find more accurate solutions. These large weights, however, introduce ill-conditioning and cause computational difficulties. To address such problems, they subsequently proposed an update method to determine the minimal required weights for achieving the estimated violation of the consistency constraints.

Lassiter et al. (2005) proposed an alternative method for relaxing the consistency constraints using an ordinary Lagrangian function (OL) based on the Lagrangian duality theory. They developed a subgradient algorithm for updating dual variables and enforcing convergence. In contrast to QP, this method allows the separability of coupled subsystems. When applied to nonconvex problems, however, the OL method may not be able to achieve the original optimal solution due to a duality gap (Li et al., 2008).

To further improve the solution accuracy and efficiency, the augmented Lagrangian function based on the penalty method was applied by Tosserams et al. (2006a) using the alternating direction method of multipliers (ADMM). Concurrently, an augmented Lagrangian coordination for ATC using the separable augmented Lagrangian dual method (SALD) was introduced by Blouin et al. (2005) and further researched by Kim et al. (2006). This method combines the ordinary Lagrangian duality with the augmented Lagrangian duality, providing a subgradient algorithm without imposing the convexity and differentiability conditions required for convergence.

As parallel computing became widespread, Li et al. (2008) applied the diagonal quadratic approximation method (DQA) and the truncated diagonal quadratic approximation method (TDQA) to linearize the cross term of the augmented Lagrangian function in order to create separable subsystems, desirable for improving the computational effi-

ciency by solving subsystems concurrently. The advantage of the TDQA is that it limits the number of inner loop iterations, thereby reducing the computation cost.

The main contribution of the research here is to provide a unified duality view of the subgradient update schemes for solving ATC-decomposable problems. In addition, this chapter discusses the advantages and disadvantages of these two cutting plane methods when applied to ATC-decomposable design problems. Section 4.2 models the centralized, hierarchical problem structure. Section 4.3 presents the ATC problem statement. Section 4.4 proposes a generic subgradient algorithm and the five update schemes used to solve ATC-decomposed problems. Section 4.5 proposes a biobjective optimization approach for solving ATC-decomposed problems. Finally, the numerical applications are tested in Section 4.6, and the results are analyzed in Section 4.7.

4.2 Modeling Centralized, Hierarchical Problem Structure

Considering a finished product as a supersystem involving multiple systems, a large-scale engineering systems design problem is typically decomposed into subsystems, subsystems are decomposed into components, and so on, as seen in Figure 4.1 (Kim et al., 2003). Each element in the hierarchy is assigned to an optimal design model P and an analysis model $\mathbf{a}$ as shown in Figure 4.2 (Kim et al., 2003). The analysis model evaluates the responses of an element by calling its respective optimal design model.

Analytical target cascading (ATC) is an approach for solving these hierarchically decomposed systems design problems (Kim, 2001, Kim et al., 2002, 2003, Michelena et al., 1999, 2003). As shown in Figure 4.2, ATC introduces design targets and cascades them to elements located at lower levels through the model-based hierarchy. An analysis model is employed by each element to compute responses to the targets assigned by the upper-level element. An optimal design model is developed for each element to minimize the discrepancies between the responses and targets. If elements located at the same level share common design variables, these variables are coordinated by their parent element

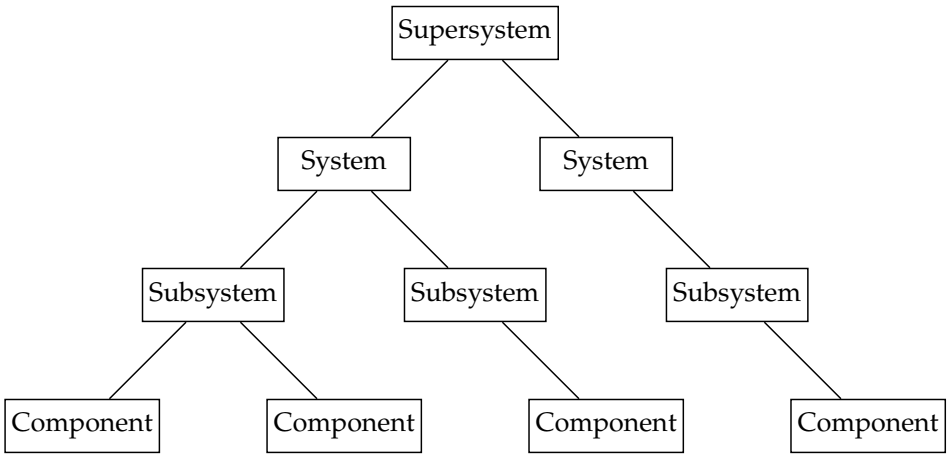


Figure 4.1: Hierarchy model of a large-scale system design problem (Kim et al., 2003)

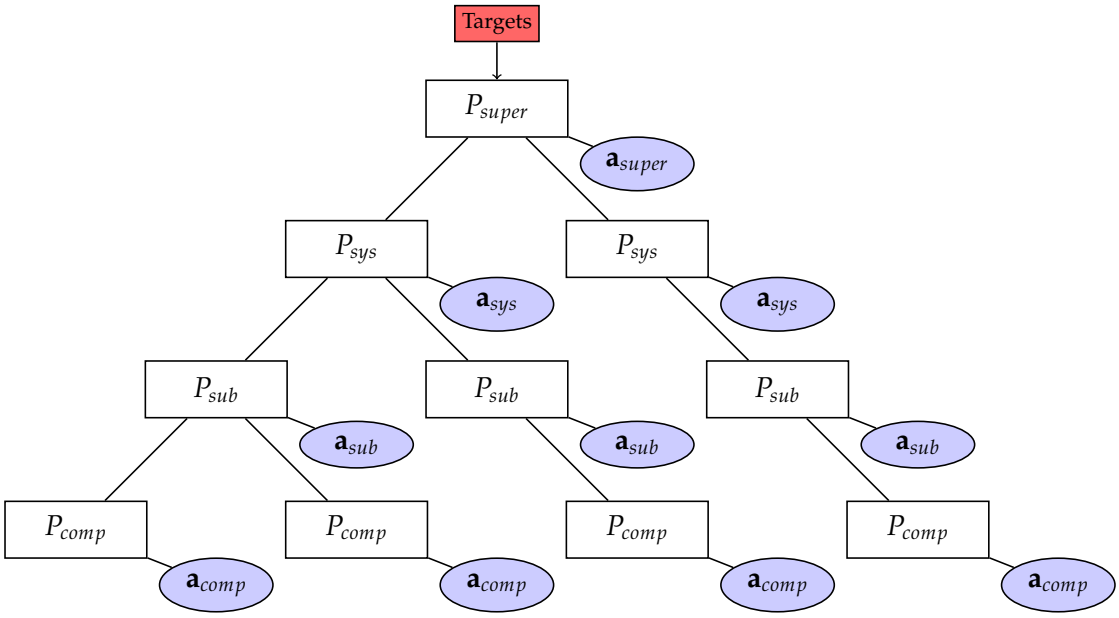


Figure 4.2: Design and analysis models in the modeling hierarchy (Kim et al., 2003)

located at the upper-level.

To clarify the modeling hierarchy, notations are defined in Table 4.1. In addition, the information flow of the ATC process is also illustrated in Figure 4.3, showing the relations between the analysis models and design models using element j in level i as an example. Usually, during the conceptual design phase, analysis models can be the response surface models based on design of experiments, approximated models based on

design sensitivity analysis, spreadsheet models, or mathematical descriptions.

Table 4.1: Nomenclature

Symbol	Description
P_{super}	Supersystem level optimal design problem
P_{sys}	System level optimal design problem
P_{sub}	Subsystem level optimal design problem
P_{comp}	Component level optimal design problem
f	Objective function for the original design problem
$\mathbf{g}$	Inequality constraints of the original design problem
$\mathbf{h}$	Equality constraints of the original design problem
i	Level
j	Problem number
$\mathbf{t}_{ij}$	Targets assigned to subsystem j at level i
$\mathbf{r}_{ij}$	Responses computed by the analysis model of subsystem j at level i
$\mathbf{Y}_{ij}$	Linking targets of subsystem j at level i
$\mathbf{y}_{ij}$	Linking responses of subsystem j at level i
f_{ij}	Local objective function of subsystem j at level i
$\mathbf{g}_{ij}$	Local inequality constraints of subsystem j at level i
$\mathbf{h}_{ij}$	Local equality constraints of subsystem j at level i
$\mathbf{x}_{ij}$	Local design variables of subsystem j at level i
$\overline{\mathbf{x}}_{ij}$	Vector of all design variables
$\mathbf{a}_{ij}$	Analysis model of subsystem j at level i
$\mathcal{C}_{ij}$	The set of c_{ij} children of discipline j at level i labeled l_1 through $l_{c_{ij}}$
$\mathcal{E}_i$	The set of elements at level i

In ATC, the optimal design model is not based on using only one analysis model. The top-level subsystem and the bottom-level subsystems of the hierarchy are special cases. At the top-level, there is only one element, e.g., supersystem, and the given system design target is specified. The bottom-level subsystems, e.g. components, are also special cases since they have no lower-level responses. In addition, it should be emphasized that ATC is not merely a design optimization methodology; it addresses the conceptual design of the product development process as seen in Figure 4.5 (Kokkolaras and Papalambros, 2008). The purpose of including ATC in the product development process is to account for the interrelations of the system parts, to identify possible tradeoffs, and to determine optimal and consistent design specifications to match design targets as close as possible, i.e. it can also be used to check whether the design targets can be achieved using the

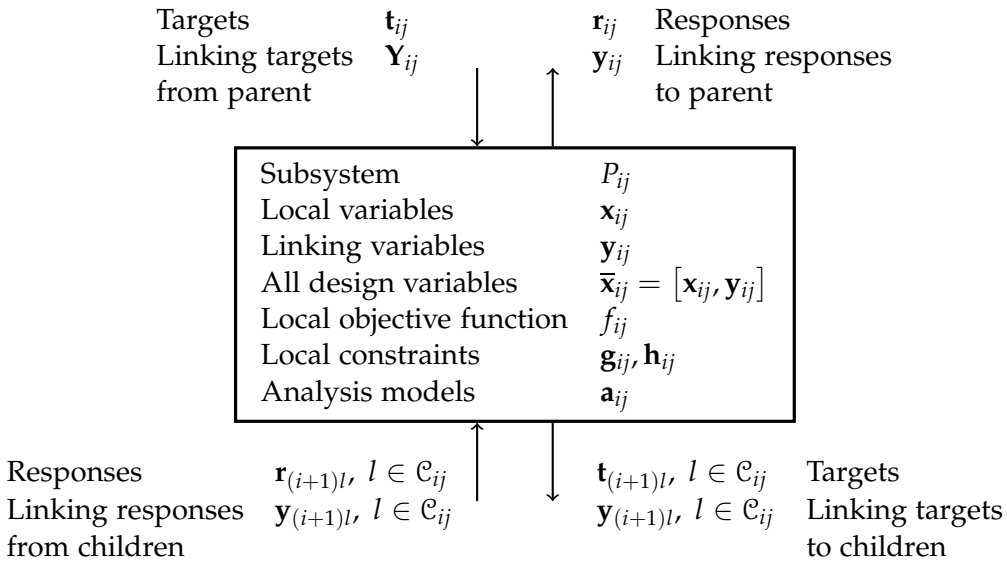


Figure 4.3: Information flow for ATC subsystem

available means (Kokkolaras and Papalambros, 2008). Once the problem is decomposed, the design embodiment for each part can be created concurrently or sequentially.

In preparation for the introduction of the ATC problem statement, first the general procedure for decomposing the original design problem of a system into ATC subsystems is discussed. The mathematical problem formulations adapted in this chapter are based on the notations of Tosserams et al. (2006a) and Li et al. (2008). The original design problem of a system, denoted as the all-in-one (AIO) problem, is to find a design that minimizes the objective and satisfies all constraints using the notations in Table 4.1, the formulation of the AIO problem, which has N levels with M elements, is defined as

$$
\begin{aligned}
\min_{\overline{\mathbf{x}}_{11},\cdots,\overline{\mathbf{x}}_{NM}} \quad & \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) \\
\text{subject to} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0} \\
& \mathbf{r}_{ij} - \mathbf{a}_{ij}\left(\mathbf{x}_{ij}, \mathbf{r}_{(i+1)l}\ l \in \mathcal{C}_{ij}\right) = \mathbf{0} \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{r}_{(i+1)l}\ l \in \mathcal{C}_{ij}\right],\ \forall j \in \mathcal{E}_i,\ i = 1, \cdots,\ N.
\end{aligned}
\tag{4.1}
$$

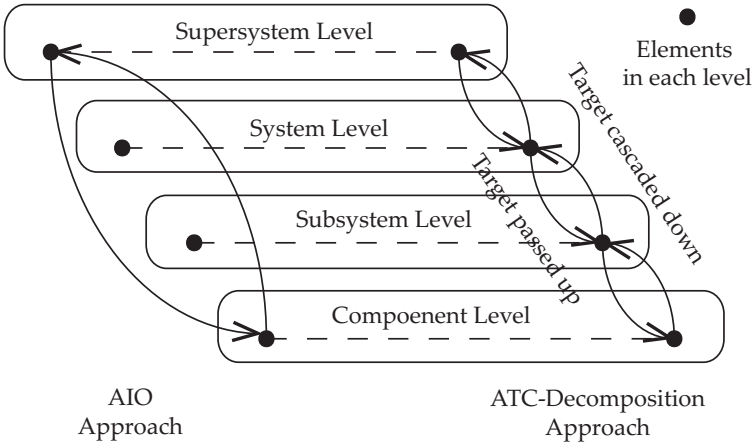


Figure 4.4: AIO and ATC-decomposition approaches to optimal design

where $\mathcal{E}_i$ denotes the set of elements at level i; $\mathcal{C}_{ij}$ the set of children of element j at level i, and c_{ij} is the total number of element j's children, for which $\mathcal{C}_{ij} = \{l_1, \cdots, l_{c_{ij}}\}$. The element j consists of a local objective function f_{ij}, inequality constraint function $\mathbf{g}_{ij}$, and equality constraint function $\mathbf{h}_{ij}$. These functions depend on a vector of design variables $\overline{\mathbf{x}}_{ij}$, which is coupled with a parent at level $i-1$ through the vector of response variables $\mathbf{r}_{ij}$ and is linked with children at level $i+1$ through a number of response vectors $\mathbf{r}_{(i+1)l_1}, \cdots, \mathbf{r}_{(i+1)l_{c_{ij}}}$.

In theory, given the design and analysis models for each element, solving the AIO problem (4.1) may be possible using classical optimization techniques. However, the AIO approach is often impractical and even computational impossible because the size of the problem, the interaction between the coupled subsystems, and the professional expertise required prohibit the system from being solved. As an alternative, the ATC-decomposition approach formulates and solves an optimization problem for individual elements in the hierarchy. The ATC-decomposition approach partitions the system design problem into multiple subsystems, thereby effectively reducing the computational difficulty and cost. The cascading process corresponding to these two approaches is illustrated in Figure 4.4 (Tosserams, 2008).

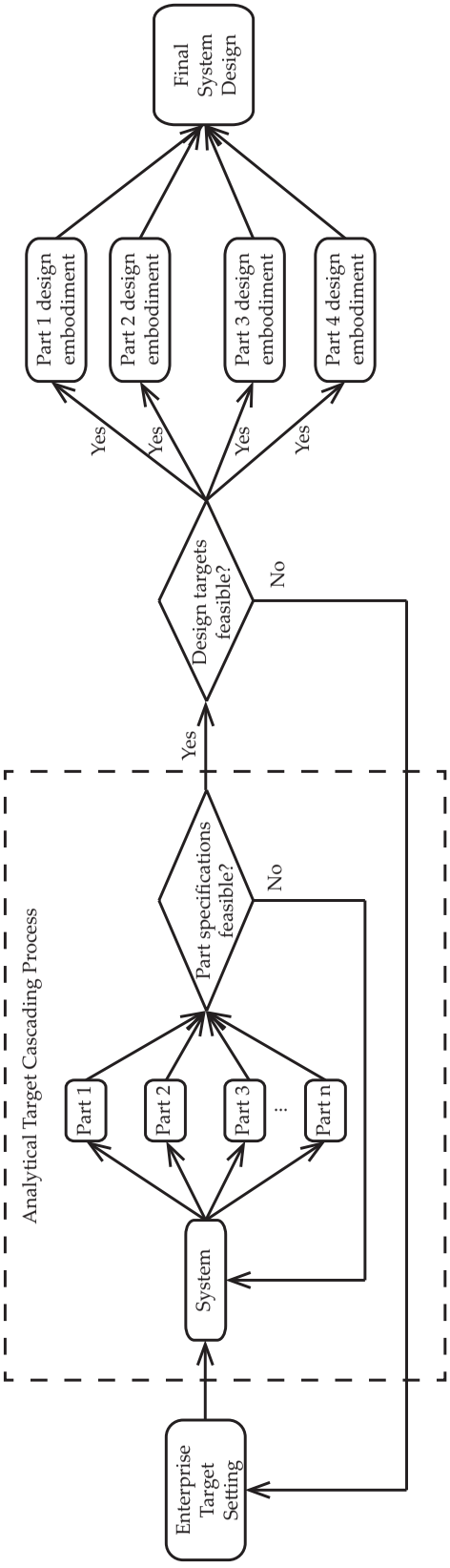


Figure 4.5: The product development process of ATC (Kokkolaras and Papalambros, 2008)

4.3 Analytical Target Cascading Design Problem Statement

In designing an engineering system, the relationship between the knowledge of goals and iterations during the design process can be seen in Figure 4.6. ATC can be utilized to communicate technical objectives to different design teams, knowing a *priori* that these goals are achievable without conflicting with the goals of other teams. Consistent system design can then be realized with minimum communication overhead, i.e., maximum efficiency, to avoid costly iterations late in the process.

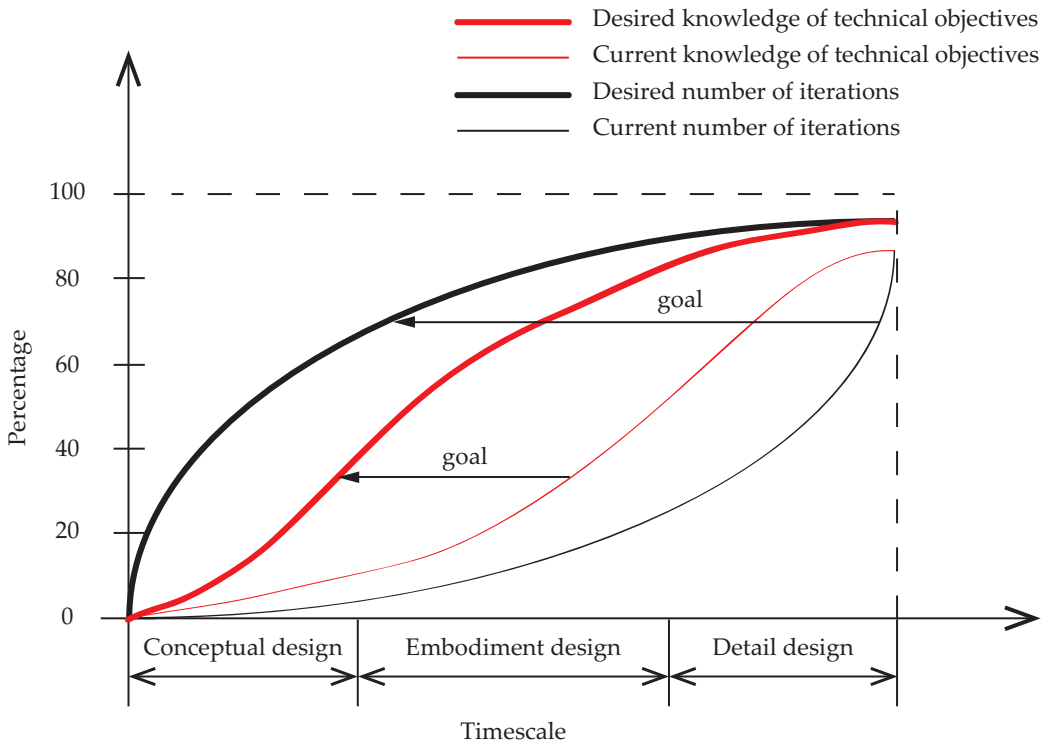


Figure 4.6: The relationship between the knowledge of goals and the number of iterations

ATC as design methodology was motivated originally by design cases in the automotive industry. In the design of a vehicle, a project manager determines the design requirements, including powertrain, body, chassis, engine, transmission, wheels, gear box, axis, and differential. These design requirements can then be translated into quantifiable design targets. These design targets are assigned to the supersystem and propagated

throughout the rest of the systems, subsystems, and components. Figure 4.7 illustrates the systems of a vehicle in the hierarchy. For each element of the hierarchy, i.e., supersystem, systems, subsystems, and components, design and analysis tasks are executed. To account for the interactions between elements, the design and analysis tasks are repetitively conducted until a feasible and consistent design is achieved.

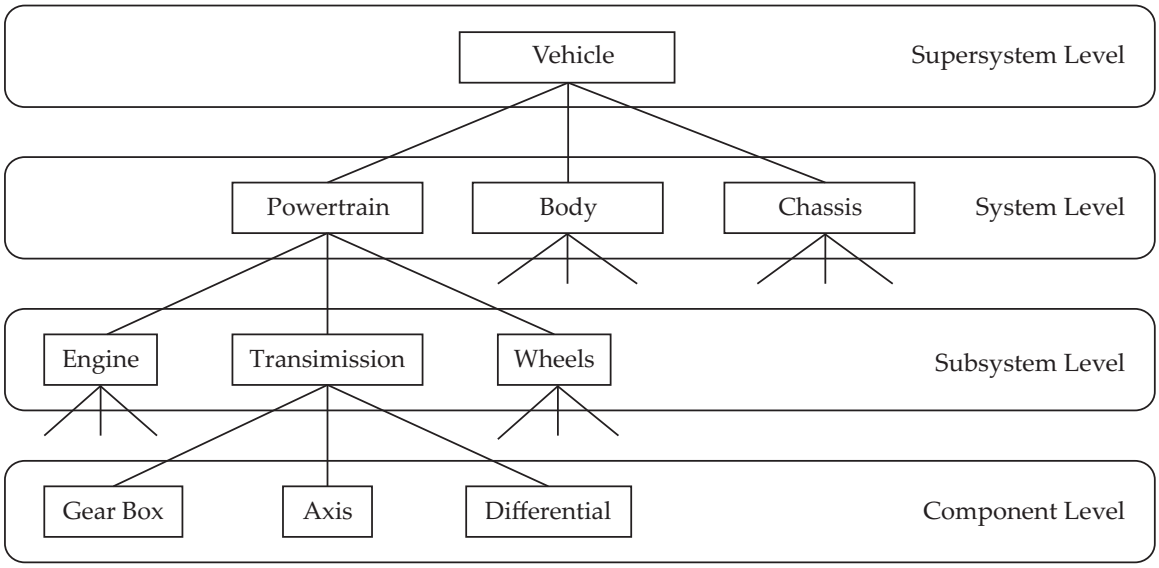


Figure 4.7: Decision hierarchy of a vehicle design problem (Kim, 2001)

The Primary Steps of Analytical Target Cascading Process

ATC is a process of determining the appropriate targets for each element for a specified set of the overall targets. Implementation of ATC in a product development process is summarized below:

1. Specification of overall targets to quantify the design requirements,
2. Propagation of the overall targets to the individual targets of the systems, subsystems, and components,
3. Design and analysis of the systems, subsystems, and components to achieve the responses that meet the respective targets,

4. Validation of the design of the decomposed problem with respect to the overall targets.

Figure 4.8 further illustrates these four steps:

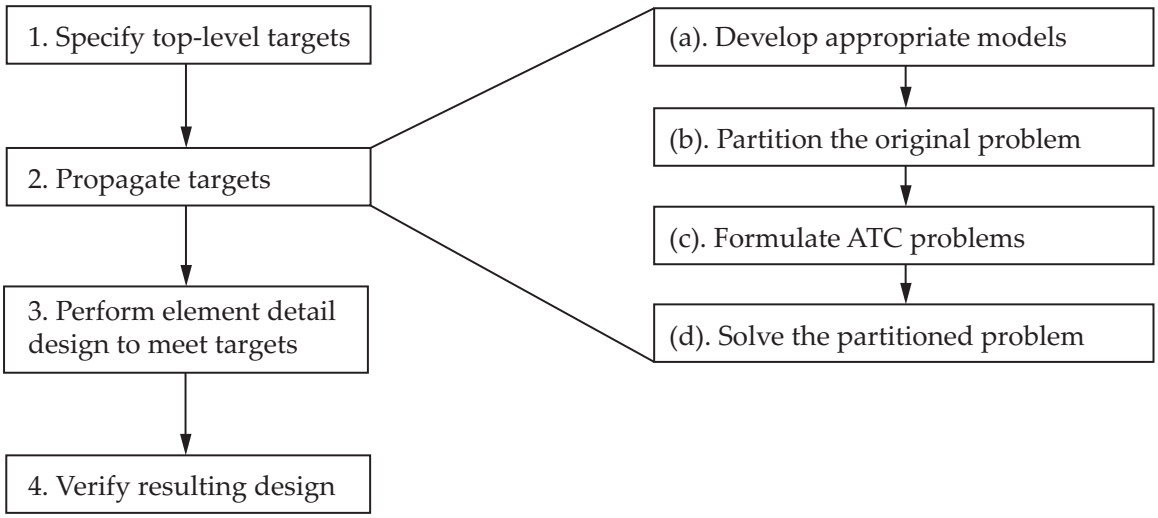


Figure 4.8: Four steps of the ATC process

If the individual targets of the elements (i.e., systems, subsystems, and components) cannot be met, then the process is repeated beginning at Step 2 with updated information obtained from the previous attempt. If the top-level targets cannot be met, then the decision maker, e.g., the executive manager, is presented with the design that best fulfills the top-level targets and is required to make a decision about whether to accept the design or to refine the targets and repeat the entire process.

Primal Problem Statement

The AIO Problem (4.1) is ATC-decomposable due to its special multilevel structure and the vector of response variables $\mathbf{r}_{ij}$ (see Figure 4.9). However, it is still not separable because the subsystems are coupled through the response $\mathbf{r}_{ij}$. To separate the AIO problem, the target variable $\mathbf{t}_{ij}$ is introduced, and the consistency constraint

$$\mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \tag{4.2}$$

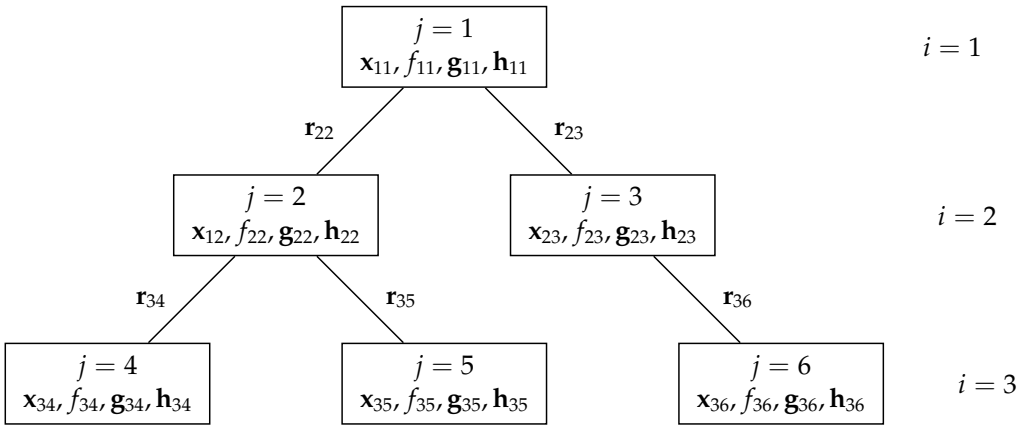


Figure 4.9: An example of the ATC hierarchical structure

is created to force the response variable to fulfill the target variable. This constraint is added to Problem (4.1), which is referred to as Primal Problem (4.3).

$$
\begin{aligned}
\min_{\overline{\mathbf{x}}_{11},\cdots,\overline{\mathbf{x}}_{NM}} \quad & \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) \\
\text{subject to} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0} \\
& \mathbf{r}_{ij} - \mathbf{a}_{ij}\left(\mathbf{x}_{ij}, \mathbf{r}_{(i+1)l}\ l \in \mathcal{C}_{ij}\right) = \mathbf{0} \\
& \mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{r}_{(i+1)l}\ l \in \mathcal{C}_{ij}\right] \\
& \forall j \in \mathcal{E}_i,\ i = 1, \cdots,\ N.
\end{aligned}
\tag{4.3}
$$

The solution to Problem (4.3) also solves the original problem (4.1) (Tosserams et al., 2006a).

Decomposed Problem Statement

For the purpose of decomposition, inconsistency between the target and the response is allowed. However, by allowing this inconsistency, the solutions obtained at each subsystem may be infeasible for the overall problem. This issue can be resolved by using

relaxation techniques, such as the quadratic penalty function (Michalek and Papalambros, 2005a), the ordinary Lagrangian function (Lassiter et al., 2005), or the augmented Lagrangian function (Tosserams et al., 2006a). In this research, consistency constraints are relaxed using the augmented Lagrangian function.

The relaxed primal problem is partitioned into M relaxed primal subsystems, also referred to as decomposed problems. The subsystem j at level i of the decomposed problem is defined as

$$\begin{aligned}
\min_{\overline{\mathbf{x}}_{ij}} \quad & f_{ij}(\overline{\mathbf{x}}_{ij}) - \mathbf{v}_{ij}^T \mathbf{r}_{ij} + \left\| \mathbf{w}_{ij} \circ \left(\mathbf{t}_{ij} - \mathbf{r}_{ij} \right) \right\|_2^2 + \\
& \sum_{l \in \mathcal{C}_{ij}} \mathbf{v}_{(i+1)l}^T \mathbf{t}_{(i+1)l} \; + \sum_{l \in \mathcal{C}_{ij}} \left\| \mathbf{w}_{(i+1)l} \circ \left(\mathbf{t}_{(i+1)l} - \mathbf{r}_{(i+1)l} \right) \right\|_2^2 \\
\text{subject to} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0} \\
& \mathbf{r}_{ij} - \mathbf{a}_{ij} \left(\mathbf{x}_{ij}, \mathbf{t}_{(i+1)l} \; l \in \mathcal{C}_{ij} \right) = \mathbf{0} \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)j} \; \forall l \in \mathcal{C}_{ij} \right] \\
& \mathbf{r}_{ij} = \mathbf{a}_{ij}(\overline{\mathbf{x}}_{ij}).
\end{aligned} \tag{4.4}$$

The vector $\mathbf{v}_{ij}^T$ denotes the transposition of the dual variable vector for the ordinary Lagrangian term, and $\mathbf{w}_{ij}$ is the penalty vector for the quadratic term. The $\circ$ symbol denotes the element-wise product defined as $[a_1, a_2, \cdots, a_n] \circ [b_1, b_2, \cdots, b_n] = [a_1 b_1, a_2 b_2, \cdots, a_n b_n]$. The dual variable $\mathbf{v}_{ij}$ and the weight $\mathbf{w}_{ij}$ are fixed when solving this problem. In addition, since the target $\mathbf{t}_{ij}$ and $\mathbf{r}_{(i+1)l}$ are constants for subsystem j at level i, the terms $\mathbf{v}_{ij}^T \; \mathbf{t}_{ij}$ and $\mathbf{v}_{ij}^T \; \mathbf{r}_{(i+1)l}$ are eliminated from the objective function.

The resulting dual problem for the relaxed primal problem is given by

$$\begin{aligned}
\max_{\mathbf{v}} \quad & \psi(\mathbf{v}) \\
\text{where} \quad & \psi(\mathbf{v}) = \left\{ \min_{\overline{\mathbf{x}}_{11}, \cdots, \overline{\mathbf{x}}_{NM}} \; \sum_{i=1}^{N} \sum_{j=\mathcal{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) + \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \mathbf{v}_{ij}^T \; \left(\mathbf{t}_{ij} - \mathbf{r}_{ij} \right) + \right.
\end{aligned}$$

$$
\begin{aligned}
& \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \left\| \mathbf{w}_{ij} \circ \left(\mathbf{t}_{ij} - \mathbf{r}_{ij} \right) \right\|_2^2, \\
\text{s.t.} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\
& \mathbf{r}_{ij}(\overline{\mathbf{x}}_{ij}) - \mathbf{a}_{ij} \left(\mathbf{x}_{ij}, \mathbf{t}_{(i+1)l} \; \forall l \in \mathcal{C}_{ij} \right) = \mathbf{0} \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)l} \; \forall l \in \mathcal{C}_{ij} \right] \\
& \forall j \in \mathcal{E}_i, \; i = 1, \cdots, N \Bigg\}.
\end{aligned}
\tag{4.5}
$$

Although the augmented Lagrangian function can be applied to ATC to eliminate the duality gap, the resulting primal subsystem loses its separability due to the quadratic penalty terms (Li et al., 2008, Michalek and Papalambros, 2005a,b). To overcome this difficulty, Stephanopoulos and Westerberg (1975), Ruszczynski (1995), and Li et al. (2008) proposed to maintain through linearization the quadratic term using the Diagonal Quadratic Approximation Method (DQA). To realize the separation of the original system, the coordination approach proposed in this research makes copies of the quadratic term for coupled subsystems without linearize them, applies a strategy to update the penalty parameter $\mathbf{w}$ for the quadratic term, and solves subsystems with fixed value of penalty parameter.

4.4 Coordination Strategy Using Subgradient Algorithm

The goal of ATC is to identify interactions and possible tradeoffs among elements early in the design development process and to determine specifications that yield consistent system design with minimized deviations from desired targets. To accomplish this goal, the dual problem is solved indirectly since it and the primal problem have the same optimal AIO value. A solution to the dual problem is found using a subgradient algorithm. Since the dual function $\psi(\mathbf{v})$ is not differentiable, a subgradient of the dual

function $\mathbf{t}_{ij} - \mathbf{r}_{ij}$ is calculated with respect to the dual variable $\mathbf{v}_{ij}$ for subsystem j at level i (Bazaraa et al., 2006). At each iteration, the dual problem is solved in two steps. The first step solves decomposed Problem (4.4) of each subsystem for some values of dual variables, and the second step updates the dual variables.

The vector of primal design variables obtained by solving primal subsystems at iteration k is denoted by $\overline{\mathbf{x}}^k$. The dual variables and penalty weights at iteration k are denoted by $\mathbf{v}^k$ and $\mathbf{w}^k$. The subgradient of the dual objective function of subsystem j at level i at iteration k is denoted as $\mathbf{t}_{ij}^k - \mathbf{r}_{ij}^k$. As a computational framework for this work, the following generic subgradient algorithm is used.

Step 1: Initialize $\overline{\mathbf{x}}^0$, $\mathbf{v}^1$, and $\mathbf{w}^1$. Set $k = 1$.

Step 2: Individually solve subsystems (4.4) in parallel or sequentially.

Step 3: For each subsystem, calculate the subgradient $\mathbf{t}_{ij}^k - \mathbf{r}_{ij}^k$. If the current primal solution is feasible for the AIO problem and the algorithm converged, stop.

Step 4: If not, update the dual variable $\mathbf{v}^k$ to obtain $\mathbf{v}^{k+1}$ and update the penalty weight $\mathbf{w}^k$ to obtain $\mathbf{w}^{k+1}$. Set $k = k + 1$. Repeat **Step 2** to **Step 4**.

Subgradient Methods

Dual update schemes significantly impact the effectiveness of the algorithm convergence. At iteration $k + 1$, the dual variable $\mathbf{v}_{ij}$ is updated as

$$\mathbf{v}_{ij}^{k+1} = \mathbf{v}_{ij}^k + s_{ij}^k \left(\mathbf{t}_{ij}^k - \mathbf{r}_{ij}^k \right), \tag{4.6}$$

where s_{ij}^k is the step size calculated based on one of the update schemes shown in Table 4.2, namely the K, M, or O update schemes. For the O update scheme, the step size depends on the optimal AIO dual objective value $\psi(\mathbf{v}^\star)$, which is generally unknown, thus making the method impractical. However, it is included here for the purpose of comparison.

Table 4.2: Stepsize of the traditional dual update schemes

Schemes	s_{ij}^k	Ref.
K	$\frac{1}{k\|\mathbf{t}_{ij}^k-\mathbf{r}_{ij}^k\|_2}$	(Lassiter et al., 2005)
M	$\frac{(1+5)}{(k+5)\|\mathbf{t}_{ij}^k-\mathbf{r}_{ij}^k\|_2}$	(Bertsekas, 2003)
O	$\frac{a^k\left(\psi^\star-\psi\left(\mathbf{v}^k\right)\right)}{5\|\mathbf{t}_{ij}^k-\mathbf{r}_{ij}^k\|_2}, \quad 1<a<2$	(Gasimov, 2002)

Cutting Plane Methods

Two cutting plane methods are applied here to solve for the dual variable $\mathbf{v}$. The purpose of these methods is to generate a piecewise linear approximation of the dual function during an iterative process. This approximation is refined at each iteration by adding a hyperplane, also called a cut, and used in lieu of the dual function in the optimization process.

Compared to the traditional update schemes seen in Table 4.2, the cutting plane methods have the following advantages that make them attractive choices: 1) they do not require differentiability of the objective and constraint functions of the dual problem; and 2) they do not require evaluation of the objective and constraint functions at each iteration.

Linear Cutting Plane Method The first cutting plane method is the linear cutting plane method (LCP). Let $\theta^{k+1}(\mathbf{v})$ be a piecewise linear approximating function of the dual function based on the dual variables, the dual function values, and the corresponding subgradients obtained in all previous iterations. The resulting LCP problem is defined as

$$\begin{aligned} \max_{\mathbf{v}} \quad & \theta^{k+1}(\mathbf{v}) \\ \text{where} \quad & \theta^{k+1}(\mathbf{v}) = \min_{\mathbf{v}} \left\{ \psi\left(\mathbf{v}^1\right) + \left(\mathbf{v}-\mathbf{v}^1\right)^T \left(\mathbf{t}^1-\mathbf{r}^1\right), \cdots, \right. \\ & \left. \psi\left(\mathbf{v}^k\right) + \left(\mathbf{v}-\mathbf{v}^k\right)^T \left(\mathbf{t}^k-\mathbf{r}^k\right) \right\}. \end{aligned} \tag{4.7}$$

In each iteration, a linearization of the form $\psi(\mathbf{v}^k) + (\mathbf{v} - \mathbf{v}^k)^T \left(\mathbf{t}^k - \mathbf{r}^k\right)$ is added to the approximating function, meaning Problem (4.7) is equivalent to

$$\begin{aligned} \min_{q,\,\mathbf{v}} \quad & -q \\ \text{s.t.} \quad & q \leq \Big\{ \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}\left(\overline{\mathbf{x}}_{ij}^{\eta}\right) + \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \mathbf{v}_{ij}^{\eta^T} \left(\mathbf{t}_{ij}^{\eta} - \mathbf{r}_{ij}^{\eta}\right) + \\ & \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \left\| \mathbf{w}_{ij}^{\eta} \circ \left(\mathbf{t}_{ij}^{\eta} - \mathbf{r}_{ij}^{\eta}\right) \right\|_2^2, \quad \eta = 1, \cdots, k. \Big\} \end{aligned} \tag{4.8}$$

Problem (4.8) has the advantage of being a linear programming problem with respect to q and $\mathbf{v}$ with a finite number of inequality constraints. By integrating Problem (4.8) into the generic algorithm and solving it at each iteration, the dual variable is updated and used to solve the primal subsystem at the next iteration.

Proximal Cutting Plane Method One characteristic of the linear cutting plane method is that it generally creates large step sizes, an advantage when far from the optimal solution, but a disadvantage when approaching the optimum solution and, thus, may induce instability (Bertsekas, 2003). A way to limit the effects of this phenomenon is to add a quadratic term to the piece-wise linear function, thereby reducing large deviations from the current point. This method is referred to as the proximal cutting plane method (PCP), is defined as

$$\begin{aligned} \max_{\mathbf{v}} \quad & \theta^{k+1}(\mathbf{v}) - \frac{1}{2\mu^k} \left\| \mathbf{v} - \mathbf{v}^k \right\|_2^2 \\ \text{where} \quad & \theta^{k+1}(\mathbf{v}) = \min_{\mathbf{v}} \Big\{ \psi\left(\mathbf{v}^1\right) + \left(\mathbf{v} - \mathbf{v}^1\right)^T \left(\mathbf{t}^1 - \mathbf{r}^1\right), \ \cdots, \\ & \psi\left(\mathbf{v}^k\right) + \left(\mathbf{v} - \mathbf{v}^k\right)^T \left(\mathbf{t}^k - \mathbf{r}^k\right) \Big\}. \end{aligned} \tag{4.9}$$

where $\theta^{k+1}(\mathbf{v})$ is as defined in Problem (4.7) and μ^k is a positive non-decreasing scalar parameter sequence (Bertsekas, 2003).

Methods for Updating Penalty Parameters

In previous research reported in (Kim et al., 2006), the penalty parameter $\mathbf{w}_{ij}$ was updated using $\mathbf{w}_{ij}^{k+1} = \left|\mathbf{v}_{ij}^{k}\right|$, which controls the convergence. In this paper, the following method, which, according to Bertsekas (2003) and Tosserams et al. (2006a), can accelerate the convergence process, is used.

$$\mathbf{w}_{ij}^{k+1} = \begin{cases} \mathbf{w}_{ij}^{k}, & \text{if } \left|\mathbf{c}_{ij}^{k}\right| \leq \alpha \left|\mathbf{c}_{ij}^{k-1}\right| \\ \beta\, \mathbf{w}_{ij}^{k}, & \text{if } \left|\mathbf{c}_{ij}^{k}\right| > \alpha \left|\mathbf{c}_{ij}^{k-1}\right| \end{cases}, \tag{4.10}$$

where $0 < \alpha < 1$ and $1 \leq \beta < 3$ are recommended.

4.5 Coordination Strategy Using Biobjective Optimization

The previous section applies the dual methods based on subgradient optimization (Kim et al., 2006, Wang et al., 2010) for solving ATC-decomposed problems. While all these efforts have advanced the computational effectiveness of ATC, they have left room for further improvements. In this section, the ATC-decomposed problems are solved using a biobjective optimization method, which has been published recently (Gardenghi et al., 2012). Readers can also refer to Gardenghi (2009) for more details of the methodology, convergence proofs, and numerical applications of the biobjective optimization method. The following paragraphs are near-verbatim copies of a paper co-authored by the author and collaborators (Gardenghi et al., 2012), in which the author primarily contributes to the engineering application of the method using an analytical mass allocation problem taken from (Allison et al., 2005, Tosserams et al., 2006a).

ATC-decomposed Problems are in general inherently biobjective, since they seek to optimize two performance measures of the system by means of minimizing the deviations from the fixed targets. The measuring of these deviations can be approached in two ways. First, one can consider a scalar measure of the deviations, such as a norm of

the vector of deviations, and formulate a single objective problem with the norm as a scalar objective, which has commonly been done in the ATC literature. Second, one can consider the vector of deviations and formulate a multi-objective problem with this vector as the objective function.

ATC partition and coordination incorporates a compromise between the objectives that measure the performance of the system and the demands of the subproblems reflected in their constraints. Solving a subsystem independently of the system yields a subsystem optimal solution that has the best objective value for this subsystem but may produce targets that are not achievable by other subsystems. Accepting deterioration of that best objective value increases the achievability of the targets. This compromise between the optimality and achievability lends itself neatly to biobjective optimization, which, by definition, models optimization problems with conflicting objectives. Therefore, this section employs biobjective optimization in the algorithms and shows their effectiveness on one engineering example problem.

To compromise between optimality and achievability, the algorithm is designed for systems with a single upper-level and a single lower-level subsystem. This research considers a two-level system with an objective function for only the upper-level problem in which the deviation between the vector of upper-level objectives f_{11} and a fixed target vector $\mathbb{T}$ determined by the designer is minimized. The Euclidean norm is used to measure these deviations. Hence, Problem 4.1 results in the AIO problem

$$\begin{aligned} \min_{\mathbf{x}_{11},\, \mathbf{r}_{22},\, \mathbf{x}_{22}} \quad & \left\| f_{11}\left(\mathbf{x}_{11}, \mathbf{r}_{22}\right) - \mathbb{T} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11}, \mathbf{r}_{22}\right) \leq \mathbf{0} \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11}, \mathbf{r}_{22}\right) = \mathbf{0} \\ & \mathbf{g}_{22}\left(\mathbf{x}_{22}\right) \leq \mathbf{0} \\ & \mathbf{h}_{22}\left(\mathbf{x}_{22}\right) = \mathbf{0} \\ & \mathbf{r}_{22} = \mathbf{a}_{22}\left(\mathbf{x}_{22}\right). \end{aligned} \tag{4.11}$$

The bi-level structure and the definition of the response, $\mathbf{r}_{22}$, make the AIO problem ATC-decomposable. However, the problem is not yet separable since the response links the two subsystems together. To make this problem separable, new variables, $\mathbf{t}_{22}$, are introduced in the form $\mathbf{t}_{22} - \mathbf{r}_{22} = \mathbf{0}$, which is called the consistency constraint, yielding the following problem formulation

$$\begin{aligned} \min_{\mathbf{x}_{11},\, \mathbf{r}_{22},\, \mathbf{x}_{22},\, \mathbf{t}_{22}} \quad & \left\| f_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) - \mathbb{T} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) \leq \mathbf{0} \\ & \mathbf{g}_{22}\left(\mathbf{x}_{22}\right) \leq \mathbf{0} \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) = \mathbf{0} \\ & \mathbf{h}_{22}\left(\mathbf{x}_{22}\right) = \mathbf{0} \\ & \mathbf{r}_{22} = \mathbf{a}_{22}\left(\mathbf{x}_{22}\right),\ \mathbf{t}_{22} - \mathbf{r}_{22} = \mathbf{0}. \end{aligned} \tag{4.12}$$

The consistency constraints are relaxed and the biobjective problem is created as follows

$$\begin{aligned} \min_{\mathbf{x}_{11},\, \mathbf{r}_{22},\, \mathbf{x}_{22},\, \mathbf{t}_{22}} \quad & \left\{ \left\| f_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) - \mathbb{T} \right\|_2^2, \left\| \mathbf{t}_{22} - \mathbf{r}_{22} \right\|_2^2 \right\} \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) \leq 0 \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) = 0 \\ & \mathbf{g}_{22}\left(\mathbf{x}_{22}\right) \leq 0 \\ & \mathbf{h}_{22}\left(\mathbf{x}_{22}\right) = 0 \\ & \mathbf{r}_{22} = \mathbf{a}_{22}\left(\mathbf{x}_{22}\right). \end{aligned} \tag{4.13}$$

This problem is then decomposed into two subsystems. Assuming r_{22}^k to be a fixed value

of r_{22}, the upper-level bi-objective subsystem is

$$\begin{aligned} & \textit{Upper Biobjective Sub}: \\ \min_{\mathbf{x}_{11},\, \mathbf{t}_{22}} \quad & \left\{ \left\| f_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) - \mathbb{T} \right\|_2^2, \left\| \mathbf{t}_{22} - \mathbf{r}_{22}^k \right\|_2^2 \right\} \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) \leq \mathbf{0} \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) = \mathbf{0}. \end{aligned} \tag{4.14}$$

Assuming $\mathbf{t}_{22}^{k-1}$ to be a fixed value of $\mathbf{t}_{22}$, the lower-level subsystem is given by

$$\begin{aligned} & \textit{Lower Biobjective Sub}: \\ \min_{\mathbf{x}_{22},\, \mathbf{r}_{22}} \quad & \left\| r_{22} - t_{22}^{k-1} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{22}\left(\mathbf{x}_{22}\right) \leq 0 \\ & \mathbf{h}_{22}\left(\mathbf{x}_{22}\right) = 0 \\ & \mathbf{r}_{22} = \mathbf{a}_{22}\left(\mathbf{x}_{22}\right). \end{aligned} \tag{4.15}$$

Given the problem statements of subsystems (4.14) and (4.15), the initial upper-level, a single objective subsystem, is then defined as

$$\begin{aligned} & \textit{Init Upper Sub}: \\ \min_{\mathbf{x}_{11},\, \mathbf{t}_{22}} \quad & \left\| f_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) - \mathbb{T} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) \leq 0 \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11}, \mathbf{t}_{22}\right) = 0. \end{aligned} \tag{4.16}$$

Using Subsystems (4.14), (4.15), and (4.16), a generic algorithm is proposed for Problem (4.11). The algorithm is based on principles of biobjective optimization, which are relevant to ATC-decomposable problems. Its premise is that if an optimal solution of Subsystem (4.16) yields a target, t_{22}, that is achievable for the lower-level subsystem (4.15),

then the algorithm will stop during initialization, indicating that the lower-level problem is capable of achieving the target that the upper-level problem desires. However, if the constraints of the lower-level subsystem inhibit the desired target, then the algorithm will require multiple iterations to negotiate a target that is achievable by the lower-level and yet allows for an acceptable objective value for the upper-level.

Given a predefined optimality tolerance, $\varepsilon > 0$, the pseudocode for the generic algorithm, $\mathcal{A}lg_{gen}$, is presented in Pseudocode 1. During the initialization, the upper-level Subsystem (4.16) is solved, and it is likely that the intermediate solution obtained achieves a smaller (for minimization problems) value than the true value of the AIO optimal solution by violating some constraints. This solution value becomes a lower bound for the AIO solution value since the lower-level constraints have been relaxed. At the same time, the target, $\mathbf{t}_{22}^0$, for the lower-level subproblem (4.15) is set.

Pseudocode 1

algorithm $\mathcal{A}lg_{gen}$
input: $\varepsilon > 0$, *Init Upper Sub*, *Upper Biobjective Sub*, and *Lower Sub*
begin
  solve *Init Upper Sub* for $(\mathbf{x}_{11}^0, \mathbf{t}_{22}^0)$;
  set $k = 1$;
  solve *Lower Sub* for $(\mathbf{x}_{22}^k, \mathbf{r}_{22}^k)$;
  while $\left\| \mathbf{r}_{22}^k - \mathbf{t}_{22}^{k-1} \right\|_2^2 \geq \varepsilon$ **do**
  begin
    solve *Upper Biobjective Sub* for $(\mathbf{x}_{11}^k, \mathbf{t}_{22}^k)$;
    $k = k + 1$;
    solve *Lower Sub* for $(\mathbf{x}_{22}^k, \mathbf{r}_{22}^k)$;
  end while
end
output: AIO Solution

In the main step of the algorithm, and in accordance with target cascading, the target obtained at the upper-level is sent down to the lower-level subproblem as the initial target that would be ideal if it is achievable. If the subproblem can achieve this target, or in other words, a solution feasible for the lower-level subproblem can be computed so that its response matches the target, the algorithm stops. Otherwise, an optimal solution

that is feasible for the subproblem is computed so that its response, r_{22}^k, is the closest to the target. This closest response is then sent to the upper level where the bi-objective optimization problem (4.14) is solved.

In every iteration of the main step of the algorithm, another bi-objective problem is solved based on different subproblem responses. The bi-objective problem involves the simultaneous optimization of two objective functions, the AIO objective and the deviation objective produced by the relaxed consistency constraint, subject to the upper-level design constraints. The bi-objective problem models the conflict between the optimality of the AIO problem and the achievability of the target by the subproblem response. If the lower-level response cannot achieve the target, then the optimal value of the AIO objective has to be degraded so that a new target will become "more achievable."

Although the upper and lower-level subsystems use terms that resemble quadratic penalty terms, their role is different from that in the penalty approach (Li et al., 2008, Tosserams et al., 2006a), as can be seen when comparing the weights accompanying these terms. In the proposed biobjective approach the weights are continually constant, while in the penalty approach they keep increasing. The constant values are implied by the bi-objective context while the quadratic penalty method requires the weights to be updated at every iteration.

When solving the bi-objective problem, it is necessary to compute its specific Pareto solutions (Ehrgott, 2005). Pareto points are sought that favor the minimization of the deviation objective value at the expense of the deterioration (here, an increase) of the AIO objective value obtained in the initialization. For the conceptual development of the algorithm, the method for computing these Pareto solutions is unimportant, and, therefore, the algorithm is referred to as generic.

The fact that the subsequent iterates are in the Pareto set helps the user understand the tradeoff between the system optimality and the achievability of targets for the subproblems. From iteration to iteration of the generic algorithm, the optimality degrades while the achievability improves. If the targets are achievable, the user can continue the

algorithm until it converges. However, if the targets are not achievable, which is the expected case in real-life applications, the user can stop the algorithm at any iteration accepting the current tradeoff between optimality and achievability. In this way, the algorithm guides users and enables them to make an informed decision regarding termination. The proposition presenting a proof for this case can be referred to Gardenghi et al. (2012).

Algorithm for Two Subproblems In the implementation of the algorithm, the biobjective problem (4.14) is replaced with the following scalarized subproblem

$$\begin{aligned} & Upper\ Sub: \\ \min_{\mathbf{x}_{11},\, \mathbf{t}_{22}} \quad & (1-\alpha)\left\|f_{11}\left(x_{11}, t_{22}\right)-\mathbb{T}\right\|_2^2+\alpha\left\|t_{22}-r_{22}^k\right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_{11}\left(\mathbf{x}_{11},\ \mathbf{t}_{22}\right) \leq \mathbf{0}, \\ & \mathbf{h}_{11}\left(\mathbf{x}_{11},\ \mathbf{t}_{22}\right)=\mathbf{0}, \end{aligned} \tag{4.17}$$

where $0 < \alpha < 1$. Pseudocode 1 requires two changes. The *Upper Sub* in the input is substituted for *Upper Biobjective Sub*, and the line that reads "solve *Upper Biobjective Sub*" is replaced with the line "solve *Upper Sub*" which calls subproblem (4.17) to be solved. The resulting algorithm is referred to as $\mathcal{A}lg_{2S}$.

In the weighted-sum problem we are able to easily select weights that yield desired Pareto solutions of subproblem (4.14) due to its biobjective nature. The choice of the weights results from the assumption that users are unwilling to accept that the lower-level subproblem responses are far from achieving their targets even though the AIO objective value is small. Therefore, the small AIO objective value is sacrificed to find a lower-level subproblem response that is closer to the target. Under this premise it is unnecessary to solve the biobjective problem (4.14) for more than a single well chosen Pareto solution. The algorithm only seeks the Pareto solution that places a priority on minimizing $\left\|\mathbf{t}_{22}-\mathbf{r}_{22}^k\right\|_2^2$ while allowing an increase of the AIO objective. We choose the weight α to

be very close to but not equal to 1. If we were to let $\alpha = 1$ and the problem had alternate optimal solutions that minimized the deviations, then the solution returned by the optimizer may not be the solution that also accounts for minimizing $\|f(\mathbf{x}_{11},\ \mathbf{t}_{22}) - \mathbb{T}\|_2^2$. Hence, $\alpha < 1$ is required.

The stopping criterion for algorithm $\mathcal{A}lg_{2S}$, as given in Pseudocode 1, is the amount of deviation (or lack of feasibility with respect to the consistency constraints) that is acceptable between the target and the actual response of the lower-level problem. It is generally given by a predetermined acceptable tolerance, $\varepsilon > 0$, based on the problem at hand as well as the numerical precision of the optimizer.

4.6 Numerical Applications

Numerical Applications for Subgradient Algorithm

For the purpose of comparing with the K, M, and O update schemes, as well as all other ATC methods (Li et al., 2008, Tosserams et al., 2006a), three examples are used to demonstrate the cutting plane methods. The decomposition configuration for these three examples is illustrated in Figure 4.10. In each decomposition configuration, design variables, objective functions, and constraint functions are associated with each element. The coupling variables are depicted between connected elements.

In this research, three criteria are used to evaluate performance: the solution error, the number of iterations, and the number of function evaluations. All these criteria are defined by Tosserams et al. (2006a) and Li et al. (2008). For all examples, markers (from left to right) shown in Figures 4.11, 4.12, and 4.13 represent numerical experiments with the consistency error termination tolerance set to $\epsilon = \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. The three examples are implemented and solved using Matlab's algorithms: the sequential quadratic programming for all subsystems, the medium-scale linprog simplex algorithm for the linear cutting plane method, and the interior-point-convex method for the proximal cutting plane method.

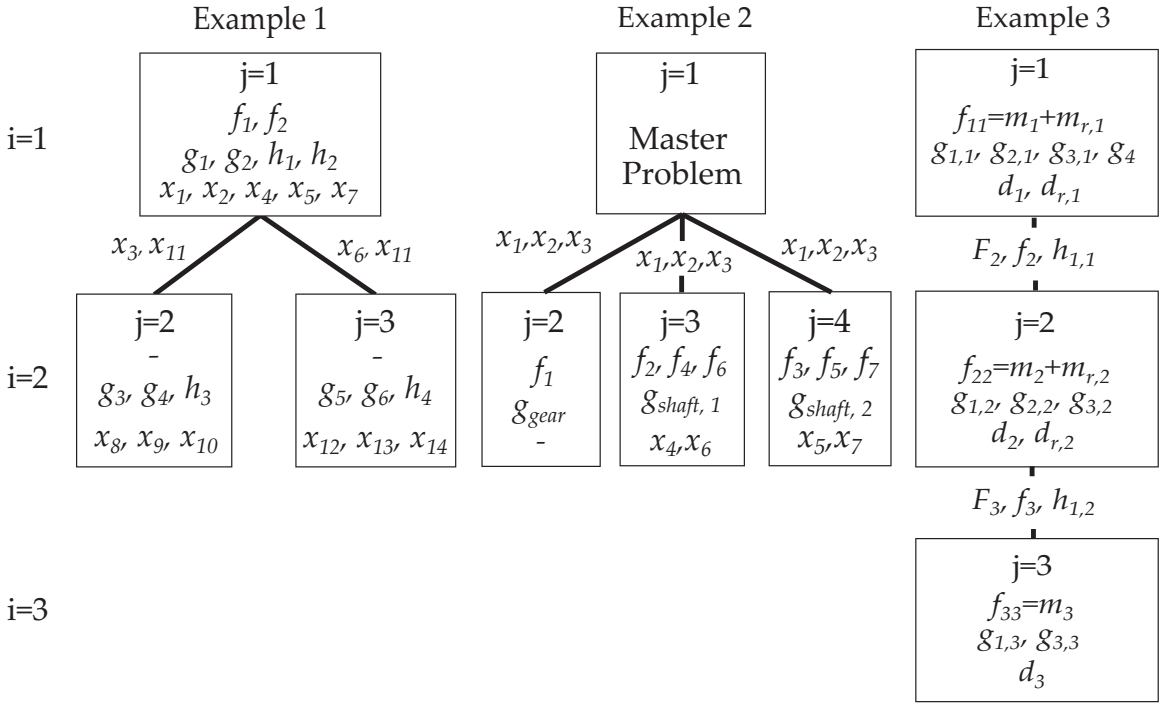


Figure 4.10: Decomposition configurations of examples.

Example 1: Nonconvex Nonlinear Programming Problem

The first example is a two-level decomposition of a geometric programming problem studied by Kim et al. (2006), Kim (2001), Tosserams et al. (2007). Its AIO problem is given by

$$
\begin{aligned}
\min_{x_1,\ x_2,\cdots,\ x_{14}} \quad & f = x_1^2 + x_2^2 \\
\text{s.t.} \quad & g_1 = \left(x_3^{-2} + x_4^2\right) x_5^{-2} - 1 \leq 0 \\
& g_2 = \left(x_5^2 + x_6^{-2}\right) x_7^{-2} - 1 \leq 0 \\
& g_3 = \left(x_8^2 + x_9^2\right) x_{11}^{-2} - 1 \leq 0 \\
& g_4 = \left(x_8^{-2} + x_{10}^2\right) x_{11}^{-2} - 1 \leq 0 \\
& g_5 = \left(x_{11}^2 + x_{12}^{-2}\right) x_{13}^{-2} - 1 \leq 0 \\
& g_6 = \left(x_{11}^2 + x_{12}^2\right) x_{14}^{-2} - 1 \leq 0
\end{aligned}
\tag{4.18}
$$

$$
\begin{aligned}
h_1 &= \left(x_3^2 + x_4^{-2} + x_5^2\right) x_1^{-2} - 1 = 0 \\
h_2 &= \left(x_5^2 + x_6^2 + x_7^2\right) x_2^{-2} - 1 = 0 \\
h_3 &= \left(x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2\right) x_3^{-2} - 1 = 0 \\
h_4 &= \left(x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2\right) x_6^{-2} - 1 = 0 \\
\text{where} \quad & x_1,\ x_2, \cdots,\ x_{14} \geq 0.
\end{aligned}
$$

Figure 4.11 displays the number of iterations and the number of function evaluations as functions of the solution error. For all updates, a starting point is randomly selected from the initial designs given by Tosserams et al. (2007). The initial dual variable and weight are set to $\mathbf{v}^1 = \mathbf{0}$ and $\mathbf{w}^1 = \mathbf{2}$, where $\mathbf{w}^1 = \mathbf{2}$ is the critical weight observed in (Tosserams et al., 2007). The parameters used for updating $\mathbf{w}$ are set to $\alpha = 0.25$ and $\beta = 1.1$. The parameter used in PCP is set to $\mu = 2$.

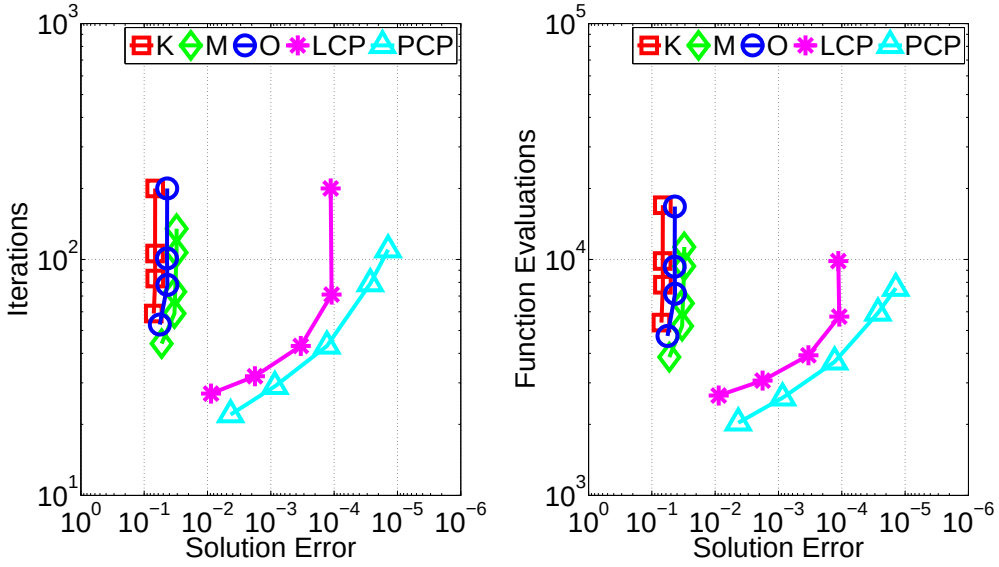


Figure 4.11: Example 1: the number of iterations and the number of function evaluations as functions of the solution error.

Example 2: Golinski's Speed Reducer Problem

The second example is the Golinski's speed reducer problem studied by Golinski (1970), Tosserams et al. (2007), and others, consisting of minimizing the weight of a reducer while satisfying the stress, deflection, and geometric constraints imposed by gear and shaft design practices. The AIO problem is given by

$$
\begin{aligned}
\min_{x_1,\cdots,x_7} \quad & f = \sum_{i=1}^{7} f_i, \\
\text{s.t.} \quad & \mathbf{g}_{\text{gear}} = [g_5, g_6, g_9, g_{10}, g_{11}]^T \leq \mathbf{0} \\
& \mathbf{g}_{\text{shaft,1}} = [g_1, g_3, g_7]^T \leq \mathbf{0} \\
& \mathbf{g}_{\text{shaft,2}} = [g_2, g_4, g_8]^T \leq \mathbf{0} \\
& 2.6 \leq x_1 \leq 3.6,\ 0.7 \leq x_2 \leq 0.8,\ 17.0 \leq x_3 \leq 28.0,\ 7.3 \leq x_4 \leq 8.3, \\
& 7.3 \leq x_5 \leq 8.3,\ 2.9 \leq x_6 \leq 3.9,\ 5.0 \leq x_7 \leq 5.5, \\
\text{where} \quad & f_1 = 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9335 x_3 - 43.0934), \\
& f_2 = -1.5079 x_1 x_6^2,\ f_3 = -1.5079 x_1 x_7^2, \\
& f_4 = 7.477 x_6^3,\ f_5 = 7.477 x_7^3, \\
& f_6 = 0.7854 x_4 x_6^2,\ f_7 = 0.7854 x_5 x_7^2 \\
& g_1 = \frac{1}{110 x_6^3} \sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 1.69 \cdot 10^7} - 1 \qquad (4.19) \\
& g_2 = \frac{1}{85 x_7^3} \sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 1.575 \cdot 10^8} - 1 \\
& g_3 = \frac{1.5 x_6 + 1.9}{x_4} - 1,\ g_4 = \frac{1.1 x_7 + 1.9}{x_5} - 1 \\
& g_5 = \frac{27}{x_1 x_2^2 x_3} - 1,\ g_6 = \frac{397.5}{x_1 x_2^2 x_3} - 1 \\
& g_7 = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1,\ g_8 = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \\
& g_9 = \frac{x_2 x_3}{40} - 1,\ g_{10} = \frac{5 x_2}{x_1} - 1,\ g_{11} = \frac{x_1}{12 x_2} - 1.
\end{aligned}
$$

Figure 4.12 displays the number of iterations and the number of function evaluations as functions of the solution error. For all updates, a starting point is randomly selected from the initial point given by Tosserams et al. (2007). The initial dual variable and weight are set to $\mathbf{v}^1 = \mathbf{0}$ and $\mathbf{w}^1 = \mathbf{5}$, where $\mathbf{w}^1 = \mathbf{5}$ is the critical weight observed in Tosserams et al. (2007). The parameters used for updating $\mathbf{w}$ are defined as $\alpha = 0.25$ and $\beta = 1.1$. The parameter in PCP is defined as $\mu = 1.1$.

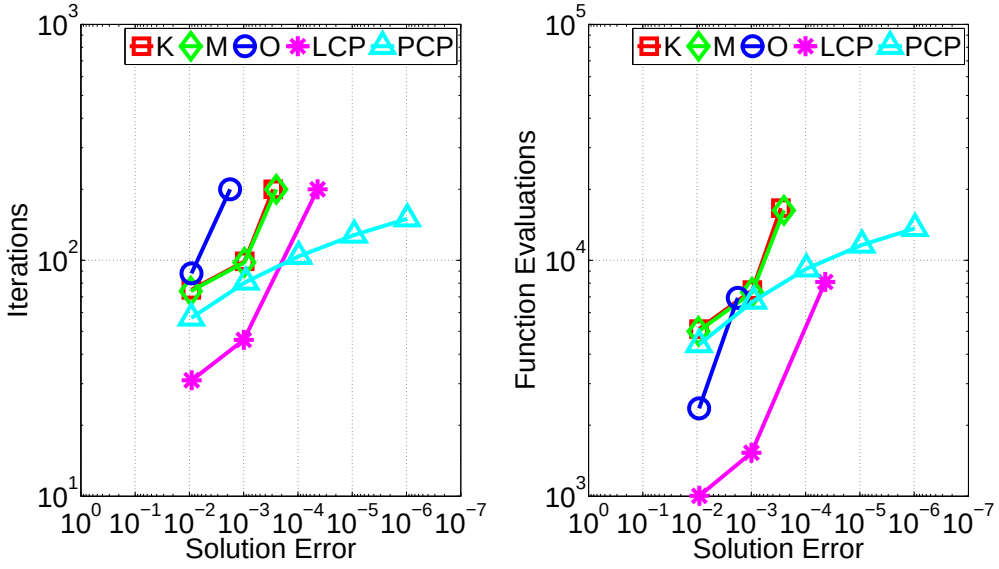


Figure 4.12: Example 2: the number of iterations and the number of function evaluations as functions of the solution error

Example 3: Anchor Beam Problem

The third example is a structural problem modified from Allison et al. (2005) and used by Tosserams et al. (2006a) and Li et al. (2008). The objective function consists of minimizing the total mass with constraints being imposed on stresses, deflections, and transmitted forces. Design variables of this structural optimization problem are the diameters of three beams $d_i, i = 1, 2, 3$ and rods $d_{r,j}, j = 1, 2$. The AIO problem is defined

as

$$
\begin{aligned}
\min_{d_1,d_2,d_3,d_{r1},d_{r2}} \quad & \sum_{i=1}^{3} m_i + \sum_{j=1}^{2} m_{r,j} \\
\text{s.t.} \quad & g_{1,i} = \frac{\sigma_{b,i}}{\overline{\sigma}} - 1 \leq 0,\ i = 1,2,3 \\
& g_{2,j} = \frac{\sigma_{a,j}}{\overline{\sigma}} - 1 \leq 0,\ j = 1,2 \\
& g_{3,i} = \frac{F_{t,i}}{\overline{F}} - 1 \leq 0,\ i = 1,2,3 \\
& g_4 = \frac{f_1}{\overline{f}_1} - 1 \leq 0 \\
& h_i = f_i - f_{i+1} - f_{r,i} = 0,\ i = 1,2 \\
\text{where} \quad & m_i = \frac{\pi}{4} d_i^2 L\rho,\ i = 1,2,3 \\
& m_{r,j} = \frac{\pi}{4} d_{r,j}^2 L\rho,\ j = 1,2 \\
& \sigma_{b,i} = \frac{32L(F_i - F_{i+1})}{\pi d_i^3},\ i = 1,2,3 \\
& f_i = \frac{64L^3(F_i - F_{i+1})}{3\pi E d_i^4},\ i = 1,2,3 \\
& \sigma_{a,j} = \frac{4F_{j+1}}{\pi d_{r,j}^2},\ j = 1,2 \\
& f_{r,j} = \frac{4F_{j+1}L}{\pi E d_{r,j}^2},\ j = 1,2.
\end{aligned}
\tag{4.20}
$$

In this problem, m_i is the mass of beam i, $m_{r,j}$ the mass of rod j, $\sigma_{b,i}$ the bending stress in beam i, $\sigma_{a,j}$ the axial stress in rod j, $F_{t,i}$ the force transmitted at the clamped end of beam i, and δ_1 the vertical deflection of beam 1. Additional compatibility constraints $h_i, i = 1,2$ are employed because the number of unknown forces and moments exceeds the number of equilibrium equations, and the problem is statically indeterminate Tosserams et al. (2006a). The parameters used in this example are set to $\overline{\sigma} = 127 \cdot 10^6$ N/m^2, $\overline{F} =$ 400 N, $\overline{\delta} = 27$ mm, $\rho = 2700$ kg/m^3, $E = 70$ GPa, $L = 1$m, and $F_1 = 1000$ N.

Figure 4.13 displays the number of iterations and the number of function evaluations as functions of the solution error. The starting point for the primal problem is set

to $\mathbf{x}^0 = [0.035, 0.035, 0.03, 0.003, 0.003]$ (Tosserams et al., 2006a), which is slightly different from the initial point used by Allison et al. (2005) and was intentionally selected as infeasible in order to demonstrate that the method does not require a feasible starting point. The initial dual variable and weight are set to $\mathbf{v}^1 = \mathbf{0}$ and $\mathbf{w}^1 = \mathbf{1}$. For all update schemes, the parameters used to update the dual variable $\mathbf{w}$ are defined as $\alpha = 0.25$ and $\beta = 1.1$. The parameter used in the PCP is defined as $\mu = 2$.

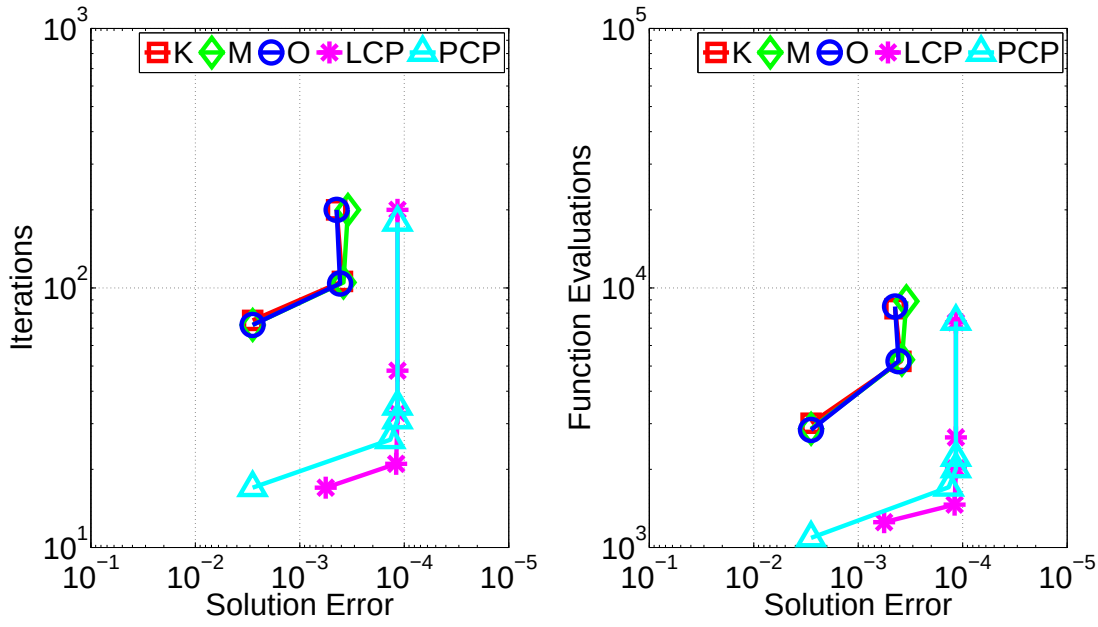


Figure 4.13: Example 3: the number of iterations and the number of function evaluations as functions of the solution error

Numerical Applications for Biobjective Optimization

One example problem is used to demonstrate the effectiveness of the biobjective optimization in comparison with other ATC coordination methods. To this end, this section applies the algorithm to the analytical mass allocation problem and reports the errors obtained by the algorithms. The objective error is defined as

$$\frac{|f^k - f^\star|}{f^\star} \tag{4.21}$$

where f^k is the AIO objective value at the end of iteration k. The solution or feasibility error is measured by the violation of the consistency constraints

$$\sum_{\mathcal{C}_{11}} \left\| t_{22}^k - r_{22}^k \right\|_2^2, \tag{4.22}$$

where $\mathcal{C}_{11}$ denotes the set of subproblems being the children of problem 1 at level 1. Additionally, we report the tradeoff between the AIO problem optimality and the achievability of the targets by the subproblems, which is a special feature of the biobjective algorithms. This tradeoff is measured by the AIO objective value at the end of iteration k, which with respect to problem (4.11) is also given by

$$f^k = \left\| f_{11}\left(x_{11}^k, t_{22}^k\right) - \mathbb{T} \right\|_2^2, \tag{4.23}$$

and the violation of the consistency constraints (4.22) in the same iteration.

Whenever possible, the performance of the proposed algorithm on these examples is compared with the algorithms that have the best performance on the same examples, as reported in the literature. We do not compare computational results obtained with the ATC coordination and the AIO approach because our goal is not to improve computational efficiency by decomposition, but to demonstrate the effectiveness of our biobjective-based coordination scheme as compared to other coordination schemes. All computations, except for the mass allocation problem (Section 4.1.5) and the portal frame design problem (Section 4.3), have been run using Matlab version 7.10.0.499 (R2010). The two engineering design problems have been solved with Matlab version 7.12.0.635 (R2011a).

Analytical Mass Allocation Problem The structural optimization problem described in Section 4.6 is also used to demonstrate the performance of the biobjective optimization. The goal of this problem is to find the dimensions of the beams and rods that minimize the overall mass. This problem is solved for four values of the weight $\alpha = 0.7, 0.8, 0.9$, and

0.999, which offers more insight into the weight selection process. Figure 4.14 shows that the minimal objective error is 0.0488 and Figure 4.15 shows that the solution error is in the range from 10^{-4} to 10^{-9}.

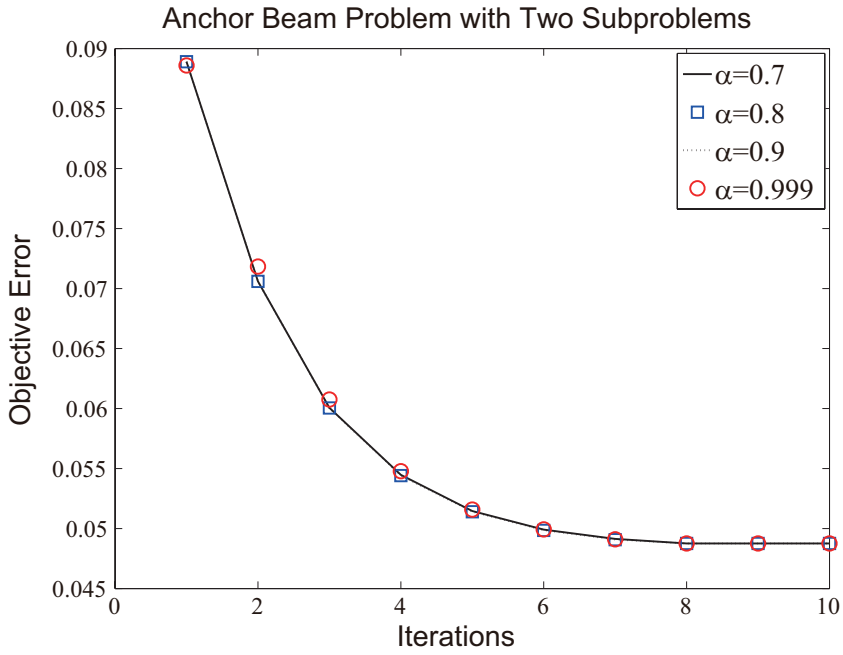


Figure 4.14: Objective error obtained with Algorithm $\mathcal{A}lg_{2S}$ in iteration k for the mass allocation problem

Note that for this example the objective error does not change with the change of the weight while the solution error does. As the weight increases, the solution error decreases because this error results directly from the deviations that decrease due to a higher weight applied to the them. On the other hand, the objective error measures the lack of optimality and remains the same despite the fact that the objective function remains in conflict with the solution error and therefore could deteriorate. In comparison to the other weights, the highest weight of 0.999 guarantees the fastest target achievability at no expense to system optimality.

Figure 4.16 presents the tradeoff between the AIO objective value at the subsequent solutions returned by the algorithm at the end of iteration k and the deviation measured by $\left\|\mathbf{t}_{22}^k - \mathbf{r}_{22}^k\right\|_2^2$ in the same iteration. The curve associated with the highest

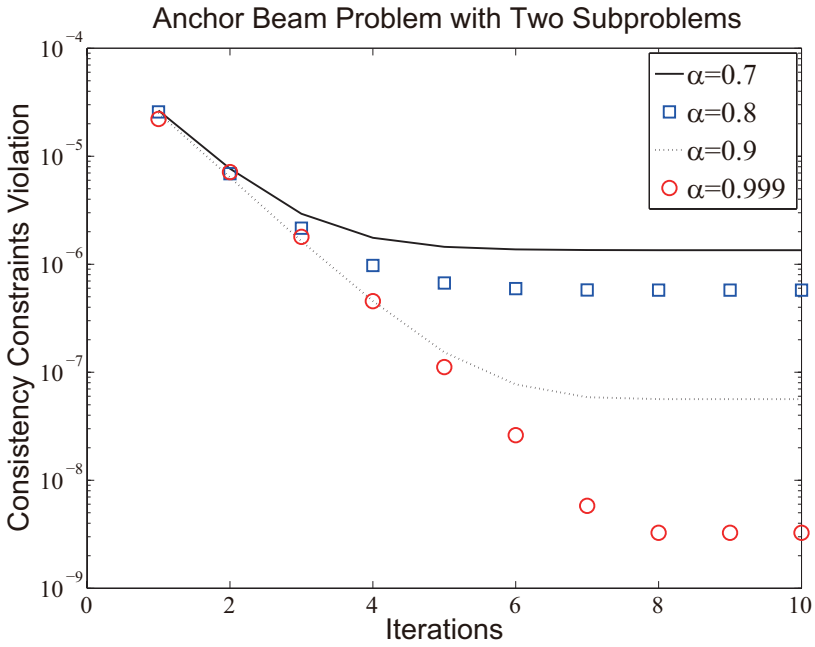


Figure 4.15: Solution error obtained with Algorithm $\mathcal{A}lg_{2S}$ in iteration k for the mass allocation problem

weight confirms that, in this problem, this weight offers the fastest target achievability at no expense to AIO optimality. The same problem was also solved with a subgradient optimization method (Wang et al., 2010) that required 37 iterations to achieve the objective error of 0.0585 and the solution error of 10^{-9}.

4.7 Concluding Remarks

Conclusion for Subgradient Algorithm

These three numerical experiments show that significant computational benefits can be achieved by using the cutting plane methods (LCP and PCP) rather than the traditional subgradient update schemes (K, M, and O updates). This result is due to two aspects. First, both the traditional subgradient update schemes and the cutting plane methods calculate a single subgradient at each iteration, which is used to update the dual variable. However, LCP and PCP update the dual variable based on not only the current

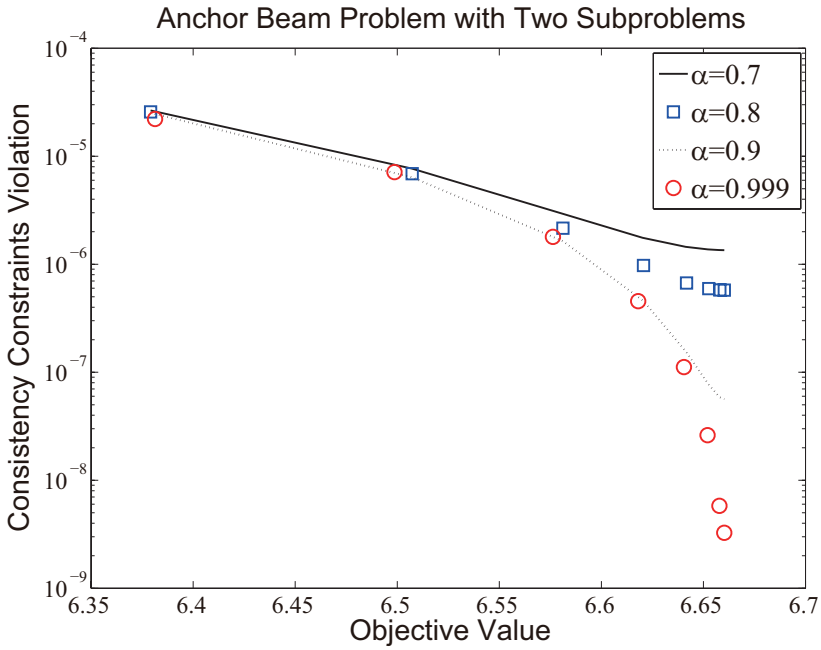


Figure 4.16: Iterative solutions obtained with Algorithm $\mathcal{A}lg_{2S}$ for the mass allocation problem: tradeoff between the AIO objective value versus the infeasibility of the solution with respect to the consistency constraints in iteration k

subgradient but also all previously generated ones. Second, the strategies for using the subgradient are quite different. The K, M, and O updates are based on algebraic formulations and were developed from the gradient and gradient-projection methods, using the subgradient as a direction for updating $\mathbf{v}^k$ (Bertsekas, 2003). Conversely, LCP and PCP, being more sophisticated, are based on geometrical formulations and continually improve the piece-wise linear approximations of the dual objective during the iterative process.

Both LCP and PCP have advantages and disadvantages. LCP generates a large step size, which leads to quick convergence at the beginning of the iterative process. PCP generates a smaller step size, which may explain why Example 2 shows a poor convergence of PCP at the beginning of an iterative process. However, as the number of iterations increases, PCP improves the convergence process by controlling and limiting the step size in updating the dual variable $\mathbf{v}^k$. This improvement is achieved by introducing the quadratic term of Equation (4.9) into the dual objective function. In searching for a

new dual variable $\mathbf{v}^{k+1}$, i.e. solving Problem (4.8), the initial point is the value of the dual variable $\mathbf{v}^k$, which may not be an appropriate starting point. The use of parameter μ^k, a non-decreasing sequence, mitigates this issue (Bertsekas, 2003).

The results of this computational effort are shown in Figure 4.17 as functions of the solution error. Both LCP and PCP outperform the K, M, and O updates, except in Example 2, where the O update performs better than PCP at the beginning of the iterative process. However, one should recall that the O update, as defined in (Goffin, 1977), does not have a practical merit since it assumes that the AIO dual objective value is known. Therefore, the cutting plane methods seem to outperform all practical subgradient methods in these examples.

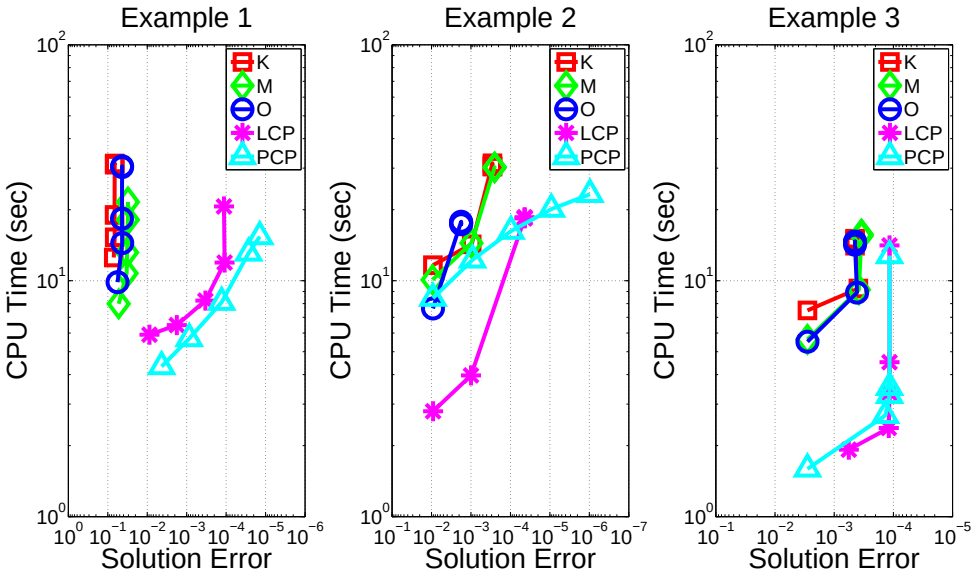


Figure 4.17: CPU time (sec) as functions of the solution error for three examples

For each of the three example problems, several starting points for the primal design variables were tested. Example 3 was solved using an infeasible starting point demonstrating that a feasible starting point is not required. The expected nonconvexity of the three problems means that the starting points may have a significant effect on the results. In addition, a starting point must be defined for the dual variables. In theory, the starting point for the dual variables could affect the speed of convergence but should not

affect the dual solution since the dual function is concave. However, the presence of the inner loop optimization of the primal problem, depending on its convexity, may influence the effect of the starting point in searching for the primal solution. This issue should be investigated further.

Conclusion for Biobjective Optimizatin

Table 4.3 presents the number of complete iterations of Algorithm $\mathcal{A}lg_{2S}$ required after initialization to find a solution of each example problem with a sufficiently small error. For two problems, the nonconvex and reducer problem, no actual iterations of the algorithm were required. The solutions generated in the initialization were optimal. Table 4.3 also reports the value for α used (which remained fixed throughout the trials), the final objective value achieved, and the solution error of the final solution. The solution error is given by the value of $\|t_{22} - r_{22}\|_2^2$.

Table 4.3: A summary of results obtained with $\mathcal{A}lg_{2S}$

Problem	Mass Alloc.
Algorithm Iterations	8
Fixed Value of	0.999
Final Objective Value	6.6602
Solution Error	3.2604e-9

It is clear from all five examples that Algorithm $\mathcal{A}lg_{2S}$ can quickly and efficiently solve both convex and nonconvex problems with two subproblems because it can easily detect the achievability of the targets for the lower-level subproblem with respect to its feasibility constraints and because the biobjective context provides information on the magnitude of the weight.

The strength of Algorithm $\mathcal{A}lg_{2S}$ lies in its initialization, which is what makes it different from other ATC methods. It immediately identifies problems whose optimal solution is constrained by the upper-level constraints, and it provides intelligent (as opposed to randomly selected) targets for the lower-level subproblem. The quadratic

penalty methods or subgradient optimization could exhibit a similar behavior when zero Lagrange multipliers or weights are assumed in the first iteration. In Tosserams et al. (2006a) and Li et al. (2008), the authors discuss such experiments and report that some of those algorithms performed unnecessary iterations before returning to the optimal weights.

In summary, by analyzing coordination methods using either the subgradient algorithm or the biobjective optimization, it is noticed that there are often two types of engineering systems involved in practice. One type of systems is developed for a particular target, which can be achieved by decomposing the systems into smaller elements with a hierarchical, multi-level structure. The other type of systems may lack a centrally-agreed target, so that the hierarchical coordination methods cannot be easily applied to solve these systems. To address this issue, a network target coordination method is studied in the next chapter.

Chapter 5

Network Target Coordination

5.1 Introduction

Multidisciplinary design optimization (MDO) is concerned with systematic approaches to achieve the optimal design of complex, coupled engineering systems, where “multidisciplinary" refers to various objects or aspects that need to be considered in the design of engineering systems (Alexandrov and Lewis, 2000). The design process is complex because the size, the coupling, and the required expertise prohibit an engineering system from being solved with an all-in-one (AIO) method, where it is treated as a fully integrated single system. Consequently, it is decomposed into multiple subsystems, each of which is solved by a design team relying on its own design tools or methods. After the decomposition, each design team attempts to achieve its own design objective by satisfying the subsystem constraints without knowing how its design decision influences the other subsystems or the overall system. Therefore, a coordination method that enables subsystems to collaboratively optimize the original system is needed.

In the field of engineering design, several coordination methods have been proposed for the optimal design of decomposed systems. These methods are generally classified into three categories: 1) interaction approximation methods, 2) bi-level programming methods, and 3) penalty and Lagrangian relaxation methods (Tosserams et al., 2008b).

Among these methods, the Lagrangian relaxation methods have recently become very popular because they efficiently handle problems that cannot be solved using the conventional interaction approximation methods and bi-level programming methods. These Lagrangian relaxation methods are introduced for MDO problems based on analytical target cascading (ATC) in Kim et al. (2003). In ATC, target-response pairs are employed to represent the interdependencies of MDO-decomposed problems, and consistency constraints are used to match the targets with their respective responses. To reduce the inconsistency of the decomposed system, the consistency constraints are relaxed to the objective function. During the solution process, the residuals of the consistency constraints are minimized using either the ordinary Lagrangian function (Lassiter et al., 2005) or the augmented Lagrangian function (Blouin et al., 2005, Tosserams et al., 2006a, Wang et al., 2010).

To separate the coupled subsystem, the ordinary Lagrangian function based on duality theorem is introduced by Lassiter et al. (2005) in the field of engineering design. This method, however, can achieve an accurate solution only under restrictive assumptions that specifically include the convexity of the original problem. When a system is nonconvex, a duality gap may occur, preventing ordinary Lagrangian duality from obtain the optimal solution (Bertsekas, 2003). Because of numerical difficulties with ordinary Lagrangian function when solving nonconvex problems, the augmented Lagrangian function is introduced for ATC-decomposed problems by Blouin et al. (2005) and Tosserams et al. (2006a). The augmented Lagrangian function consists of an ordinary Lagrangian relation term and a penalty-like quadratic relation term. This quadratic term is introduced into the objective function so that the problem can be locally convexified (Bertsekas, 2003). Bertsekas (2003) also used the augmented Lagrangian function with the method of multipliers to solve the relaxed problem. Then to reduce the computational cost of the method of multipliers, the alternating direction method of multipliers (ADMM), originally proposed by Gabay and Mercier (1976), is applied by Tosserams et al. (2006a) on ATC-decomposed problems.

However, the inclusion of the quadratic term makes the ATC-decomposed systems non-separable, and the desired parallelization of decomposed problems may not be realized. For parallel computing of subsystems, a block coordinate descent (BCD) method is applied to so called quasi-separable problems (Tosserams et al., 2007), in which the proposed method iterates between solving a master problem and solving all subsystems concurrently. By depending on this master problem and solving it analytically, the BCD method realizes parallel computing of all subsystems. However, this method partially realizes the parallelization of decomposed problems, since the master problem cannot be solved until the solutions of all subsystems are available. To solve a similar problem, a diagonal quadratic approximation (DQA) is also applied to linearize the quadratic term of the augmented Lagrangian function in Li et al. (2008). However, one primary limitation of the DQA method is that it can achieve good numerical performance only when the total number of coupling variables is small (Li et al., 2008). In addition, another obstacle to progress in realizing the parallelization of decomposed problems is that the proposed BCD and DQA methods seem to be too complex. Although the designers of these two methods are often able to convey an intuitive understanding of how their methods work, it is often difficult to make this intuition formal and precise.

In the field of distributed computing, ADMM has recently become very popular due to its abilities to solve large-scale or distributed problems. ADMM is often employed for solving consensus optimization problems (Bertsekas, 2003, Bertsekas et al., 2003, Lynch, 1996, Olfati-Saber et al., 2007). A survey by Olfati-Saber et al. (2007) describes that “consensus problems have a long history in computer science and provide the foundation of the field of distributed computing." In the field of distributed computation over networks, “consensus" means that all agents can agree on a certain interest by considering the state of all agents; a “consensus optimization problem" is then solved to determine the optimal value for the consensus; and a “consensus algorithm" is the rule that illustrate the interaction and the information exchange among agents on the network (Olfati-Saber et al., 2007).

Consensus optimization problems have been solved using ADMM in the field of distributed computing in the 1980s by Bertsekas and Tsitsiklis (1989). In a recent review of the distributed consensus problems in the context of ADMM by Boyd et al. (2011), the authors argues that ADMM is well suited to solve distributed convex optimization problems. In addition, a recent survey by Nedić and Ozdaglar (2010) and several recent applications in fields of signal processing (Mateos et al., 2010) and wireless communications (Schizas et al., 2008, Zhu et al., 2010) are also good resources to understand the consensus optimization problems, especially in conjunction with ADMM. Since there are many variations of ADMM and many convergence results for ADMM in the literature, the works of Bertsekas and Tsitsiklis (1989) and Boyd et al. (2011) are referred to for their very influential discussions of the method. The applications of the method assume that 1) the real valued objective and constraints are closed, proper, and convex with respect to the design variables; and the ordinary Lagrangian function of the optimization problem has a saddle point, meaning that the strong duality holds for the problem (see Chapter 3 for the background).

In this chapter, the research employs both the consensus optimization and ADMM to address three issues concerned with engineering design problems. The first issue is that the decomposed design problems often lack a centralized access to determine values of coupling variables, e.g. a master problem. The second issue is that these problems usually involve a number of geographically distributed design teams that prefer direct communications among one another. The third issue is that there are a lot of changes during the design process so that a failure of a subsystem should not influence the work of all others.

To address these issues, a new, yet simple, coordination method is developed for nonhierarchically decomposed systems in the field of engineering design. The approach is developed such that the centralized access is eliminated. As explained in detail in the next sections, the proposed approach uses augmented Lagrangian relaxation, consensus optimization, and ADMM. The proposed solution strategy consists of two steps. In the first

step, an optimization problem associated with each agent is solved based on the results of the consensuses and the Lagrangian multipliers. In the second step, the Lagrangian multiplier is updated when the local coupling does not match with the consensus. The update of the consensus is achieved using a locally averaging step.

The outline of this chapter is as follows. Section 5.2 models the consensus optimization problem for MDO-decomposed problems based on a multi-agent network. In Section 5.3, the strategy is discussed for a single variable consensus problem. In Section 5.4, the strategy is extended to a multiple variable consensus optimization problem. The chapter is concluded in Section 5.5.

5.2 Modeling Network Architecture

The primary motivation for proposing a network model is that many engineering systems cannot be decomposed based on the hierarchical, multi-level structure proposed by ATC. Structures representing the organizations of these systems are considered to be non-hierarchical, distributed networks and an example structure is shown in Figure 5.1. In the structure, the rectangular boxes represent the subsystems and the arcs represent the coupling variables shared by the subsystems.

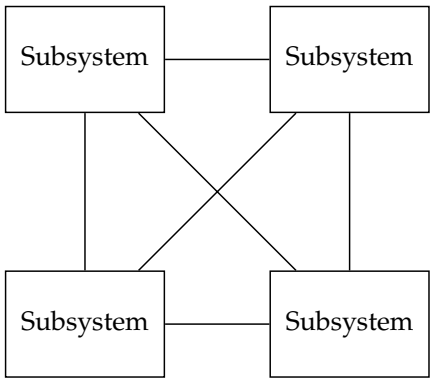


Figure 5.1: A nonhierarchical model of a system design problem

This figure illustrates the differences between ATC and the non-hierarchical, distributed network structure. In ATC, a single top-level subsystem represents the overall

system and each lower level discipline a subsystem or components of its parent subsystem. However, there is no level in the distributed network structure. In addition, ATC requires the parent subsystem to compute the coupling variables shared by lower level subsystems, whereas the distributed network model allows direct information flows between coupled subsystems. A similarity is also observed between the ATC and distributed network. Both structures require the use of individual design model P and analysis model $\mathbf{a}$ as seen in Figure 5.2, that considers inputs from coupled subsystems.

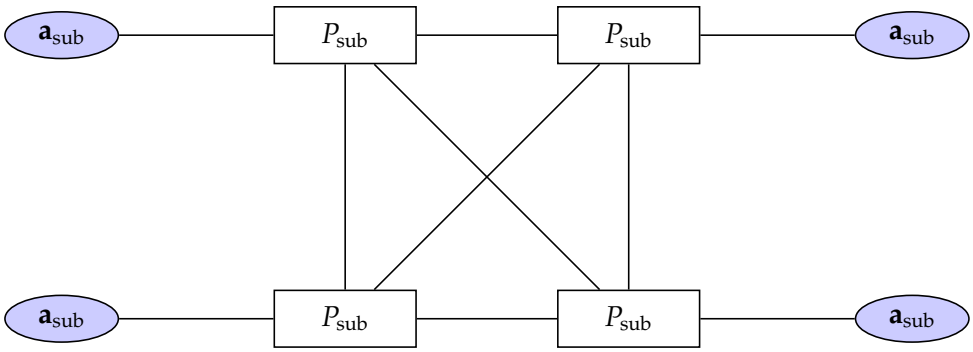


Figure 5.2: Design and analysis models in a nonhierarchical model

More formally, this research uses a new coordination method based on a multi-agent network model, consisting of a set $\mathcal{N} = \{1, 2, \cdots, m\}$ of agents and a set $\mathcal{A} = \{(1,2), (1,3), \cdots, (m-1, m)\}$ of arcs. The node adjacency list $\mathcal{N}(i)$ is the set of nodes adjacent to node i; in this case, $\mathcal{N}(i) = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}$. The arc adjacency list $\mathcal{A}(i)$ of agent i is the set of arcs emanating from that node, that is $\mathcal{A}(i) = \{(i,j) \in \mathcal{A} : j \in \mathcal{N}\}$. A simple example of the multi-agent model is shown in Figure 5.3 in which $\mathcal{N} = \{1,2,3\}$, $\mathcal{A} = \{(1,2), (2,3), (1,3)\}$, $\mathcal{N}(1) = \{2,3\}$, $\mathcal{N}(2) = \{1,3\}$, $\mathcal{N}(3) = \{1,2\}$, $\mathcal{A}(1) = \{(1,2), (1,3)\}$, $\mathcal{A}(2) = \{(2,1), (2,3)\}$ and $\mathcal{A}(3) = \{(3,1), (3,2)\}$.

5.3 Single Variable Unconstrained Consensus Optimization

To provide an initial illustration of the network target coordination based on consensus optimization, an unconstrained optimization problem with one design variable is

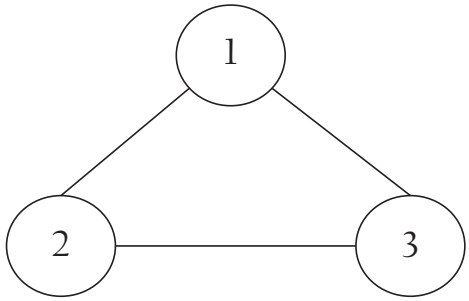


Figure 5.3: An example multi-agent network model

used and defined as

$$\min \quad f(y) = \sum_{i=1}^{m} f_i(y) \tag{5.1}$$

where $y \in \mathbb{R}$, and $f_i : \mathbb{R} \to \mathbb{R}$ is a convex objective function associated with agent i. The function f_i denotes the i-th term in the objective function.

In each subsystem, a copy of the design variable y is made and denoted y_i as a coupling design variable. In addition, a variable, referred to as a global consensus, is introduced to ensure that all agents agree on the same value of the local coupling variables $y_i,\ i = 1,\ 2,\ \cdots,\ m$. Problem (5.1) can be rewritten as

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} f_i(y_i) \\ \text{subject to} \quad & y_i - z = 0, \quad i = 1, 2, \cdots, m. \end{aligned} \tag{5.2}$$

The global consensus is collaboratively determined by all agents and Problem 5.2, referred to as a consensus problem, is solved for the optimal solutions of all local coupling variables and the global consensus. Due to the introduction of the consensus z, Problem 5.2 can be separated into m subsystems and assigned to their corresponding agents. ADMM for Problem (5.2) can be obtained directly based on the augmented Lagrangian function

$$\mathcal{L}_\rho(y_1, \cdots, y_m, z, \nu_1, \cdots, \nu_m) = \sum_{i=1}^{m} \left(f_i(y_i) + \nu_i(y_i - z) + \frac{\rho}{2}\|y_i - z\|_2^2 \right) \tag{5.3}$$

The resulting ADMM is described as following (Boyd et al., 2011)

$$y_i^{k+1} = \underset{y_i}{\operatorname{argmin}} \left(f_i(y_i) + \nu_i(y_i - z) + \frac{\rho}{2}\|y_i - z\|_2^2 \right), \tag{5.4}$$

$$z^{k+1} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i^{k+1} + \frac{1}{\rho}\nu_i^k \right), \tag{5.5}$$

$$\nu_i^{k+1} = \nu_i^k + \rho\left(y_i^{k+1} - z^{k+1} \right). \tag{5.6}$$

Here, the first and last steps are carried out independently for each agent. The step that updates the consensus estimate z is occasionally referred to as the central collector (Boyd et al., 2011), in which, the consensus is analytically solved with the projection of $y^{k+1} + \frac{1}{\rho}\nu^k$ onto each agent.

To further simplify the ADMM, the consensus update step can also be rewritten as

$$z^{k+1} = \bar{y}^{k+1} + \frac{1}{\rho}\bar{\nu}^k, \tag{5.7}$$

where $\bar{y}^{k+1}$ denotes the average value of the sum of all local couplings $\sum_{i=1}^{m}\ y_i^{k+1}$ and $\bar{\nu}^k$ denotes the average value of the sum of all Lagrangian multipliers $\sum_{i=1}^{m}\ \nu_i^k$ corresponding to each agent. Then, the ν_i update step can also be averaged as

$$\begin{aligned}
\bar{\nu}^{k+1} &= \frac{\sum_{i=1}^{m}\nu_i^{k+1}}{m} \\
&= \frac{\sum_{i=1}^{m}\nu_i^k + \rho\left(\sum_{i=1}^{m} y_i^{k+1} - \sum_{i=1}^{m} z^{k+1}\right)}{m} \\
&= \bar{\nu}^k + \rho\left(\bar{y}^{k+1} - z^{k+1}\right) \\
&= \bar{\nu}^k + \rho\left[\bar{y}^{k+1} - \left(\bar{y}^{k+1} + \frac{1}{\rho}\bar{\nu}^k\right)\right] \\
&= 0.
\end{aligned} \tag{5.8}$$

Substituting Eq. (5.7) into Eq. (5.8) means that $\bar{\nu}^{k+1} = 0$, i.e. the dual variables have an average value of zero after the first iteration. This is an important indicator for eliminating the z update step. At iteration $k > 1$, using $z^k = \bar{y}^k$, the ADMM can be reduced to a

simpler form

$$\begin{aligned} y_i^{k+1} &= \underset{y_i}{\operatorname{argmin}} \left(f_i(y_i) + \nu_i(y_i - \bar{y}^k) + \frac{\rho}{2} \|y_i - \bar{y}^k\|_2^2 \right) \\ \nu_i^{k+1} &= \nu_i^k + \rho \left(y_i^{k+1} - \bar{y}^{k+1} \right). \end{aligned} \tag{5.9}$$

The ADMM algorithm is an intuitive algorithm with the Lagrangian multiplier being updated separately to drive the local coupling to the consensus, and quadratic term helping pull all the local couplings reach their average value while still attempting to minimize each local objective function f_i. Then, after eliminating the consensus update step, the ADMM algorithm is referred to as the consensus alternating direction method of multipliers (CADMM). When applying this method after the first iteration, the objective and constraint sets are distributed across multiple agents, and each agent only has to solve its own objective and constraint functions, plus a linear term and a quadratic term which are updated at each iteration.

For CADMM, the primal and dual residuals are given by

$$\begin{aligned} r^k &= \left(y_1^k - \bar{y}^k, \cdots, y_m^k - \bar{y}^k \right), \\ s^k &= -\rho \left(\bar{y}^k - \bar{y}^{k-1}, \cdots, \bar{y}^k - \bar{y}^{k-1} \right), \end{aligned} \tag{5.10}$$

meaning their squared norms are

$$\begin{aligned} \left\| r^k \right\|_2^2 &= \sum_{i=1}^{m} \left\| y_i^k - \bar{y}^k \right\|_2^2, \\ \left\| s^k \right\|_2^2 &= m \, \rho^2 \left\| \bar{y}^k - \bar{y}^{k-1} \right\|_2^2. \end{aligned} \tag{5.11}$$

The first term in Eq. (5.11) is m times the standard deviation of the points $y_i, \cdots, y_m$, a natural measure showing the lack of consensus. The second term in Eq. (5.11) also has an intuitive meaning, i.e., the average value $\bar{y}^k$ of the local parameter estimates in the previous iteration can be modified slightly by ν_i^k, which is the cost of the i-th agent disagreeing with the consensus in the previous iteration. The use of different forms of penalty functions in the augmented Lagrangian term will lead to corresponding changes in the prior

distribution (Boyd et al., 2011). For example, using a matrix penalty P rather than a scalar ρ means that the Gaussian prior distribution has been employed in Tosserams et al. (2006a).

5.4 Multiple Variable Constrained Consensus Optimization

In this section, a constrained consensus optimization problem with respect to multiple consensuses is considered. This minimization problem has local design variables $\mathbf{x}_i \in \mathbb{R}^{n_x}$ and local coupling variables $\mathbf{y}_i \in \mathbb{R}^{n_y}$, $i = 1, \cdots, m$, with the objective function $f_1(\mathbf{x}_1,\ \mathbf{y}_1) + \cdots + f_m(\mathbf{x}_m,\ \mathbf{y}_m)$ separable with respect to $\mathbf{x}_i$ and $\mathbf{y}_i$. Each of these coupling variables consists of a selection from the components of the global consensus set $\mathbf{z} \in \mathbb{R}^n$, i.e., each component of each local coupling variables corresponds to some global consensus component $\mathbf{z}_g$. The mapping from local coupling indices onto the global variable index can be written as $g = \mathcal{G}(i, j)$, meaning that the j-th component of the local coupling variables $\mathbf{y}_{ij}$ should agrees on the value of the global consensus $\mathbf{z}_g$.

Achieving the same value for the local couplings and the global consensus also means that

$$\mathbf{y}_{ij} = \mathbf{z}_{\mathcal{G}(i,j)}, \quad i = 1, \cdots, m, \quad j = 1, \cdots, n_i \tag{5.12}$$

If $\mathcal{G}(i, j) = j$ for all agents, each local coupling variable is only a copy of the global consensus $\mathbf{z}_{\mathcal{G}(i,j)}$, resulting in a single-dimensional consensus optimization problem, similar to the one discussed in the previous section. General consensus is of interest in cases where $n_i \ll n$ so each vector of local coupling variables is comprised of only a small number of the global consensus.

In the context of MDO problems, the multidimensional form of consensus optimization naturally arises when the global variable $\mathbf{z}$ denotes the full set of design specifications, and its different subsets of the design specifications are distributed among m agents. Then $\mathbf{y}_i$ is the sub-vector of $\mathbf{z}$ corresponding to the design specifications that appear in the i-th subsystem. In other words, each agent solves only its block of coupling

design specifications.

For ease of notation, $\mathbf{z}_i \in \mathbb{R}^{n_i}$ is defined by $\mathbf{z}_{ij} = \mathbf{z}_{\mathcal{G}(i,j)}$. Since the global variables $\mathbf{z}_i$ are the ideal values of the local variables $\mathbf{y}_i$, the consistency constraints can then be written as $\mathbf{c}_i = \mathbf{y}_i - \mathbf{z}_i = 0,\ i = 1, \cdots, m$. The general form consensus optimization is then given by

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} f_i(\mathbf{x}_i,\ \mathbf{y}_i) \\ \text{subject to} \quad & \mathbf{g}_i\left(\mathbf{x}_i,\ \mathbf{y}_i\right) \leq \mathbf{0}, \quad i = 1, \cdots, m \\ & \mathbf{h}_i\left(\mathbf{x}_i,\ \mathbf{y}_i\right) = \mathbf{0}, \quad i = 1, \cdots, m \\ & \mathbf{c}_i\left(\mathbf{y}_i,\ \mathbf{z}_i\right) = \mathbf{y}_i - \mathbf{z}_i = \mathbf{0}, \quad i = 1, \cdots, m \\ \text{w.r.t.} \quad & \bar{\mathbf{x}} = \left[\mathbf{x}_1, \cdots, \mathbf{x}_m,\ \mathbf{y}_i,\ \cdots,\ \mathbf{y}_m,\ \mathbf{z}_1,\ \cdots,\ \mathbf{z}_m\right]. \end{aligned} \tag{5.13}$$

A simple model of a multidimensional consensus optimization is shown in Figure 5.4. In this example, there are $m = 3$ subsystems, the global variable dimension $n = 3$, and the local coupling variable dimensions $n_1 = 2$, $n_2 = 3$, and $n_3 = 2$. The agents and global variables form a bipartite graph, with each edge representing a consensus consistency constraint between a local coupling variable component and a global coupling variable component.

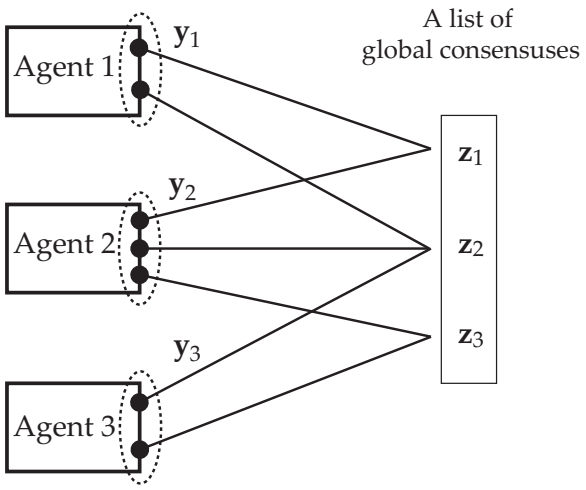


Figure 5.4: A general form of the consensus optimization

The augmented Lagrangian function for Problem (5.15) is given by

$$\mathcal{L}_\rho = \sum_{i=1}^{m} \left(f_i(\mathbf{x}_i,\ \mathbf{y}_i) + \mathbf{v}_i \left(\mathbf{y}_i - \mathbf{z}_i\right) + \frac{\rho}{2} \left\|\mathbf{y}_i - \mathbf{z}_i\right\|_2^2 \right), \tag{5.14}$$

with dual variables $\mathbf{v}_i \in \mathbb{R}^{n_i}$. Then, CADMM consists of the iterations

$$\begin{aligned}
\mathbf{y}_i^{k+1} &= \underset{(\mathbf{y}_i,\ \mathbf{y}_i)\in X_i}{\operatorname{argmin}} \left\{ f_i(\mathbf{x}_i, \mathbf{y}_i) + \mathbf{v}_i^k \mathbf{y}_i + \frac{\rho}{2} \left\| \mathbf{y}_i - \mathbf{z}_i^k \right\|_2^2 \right\} \\
\mathbf{z}^{k+1} &= \underset{\mathbf{z}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{m} \left(-\mathbf{v}_i^k \mathbf{z}_i + \frac{\rho}{2} \left\| \mathbf{y}_i^{k+1} - \tilde{\mathbf{z}}_i \right\|_2^2 \right) \right\} \\
\mathbf{v}_i^{k+1} &= \mathbf{v}_i^k + \rho \left(\mathbf{y}_i^{k+1} - \mathbf{z}_i^{k+1} \right) \\
X_i &= \left\{ (\mathbf{x}_i,\ \mathbf{y}_i) : \begin{array}{l} \mathbf{g}_i\left(\mathbf{x}_i,\ \mathbf{y}_i\right) \leq \mathbf{0} \\ \mathbf{h}_i\left(\mathbf{x}_i,\ \mathbf{y}_i\right) = \mathbf{0} \end{array} \right. \quad i = 1,\ 2,\ \cdots,\ m,
\end{aligned} \tag{5.15}$$

where the $\mathbf{y}_i^{k+1}$ and $\mathbf{v}_i^{k+1}$ update schemes can be calculated independently in parallel for each agent i.

The $\mathbf{z}$-update step decouples across the components of $\mathbf{z}$, since $\mathcal{L}_\rho$ is fully separable from its components:

$$\mathbf{z}_g^{k+1} = \frac{\sum_{\mathcal{G}(i,j)=g} \left(\mathbf{y}_{ij}^{k+1} + \frac{1}{\rho} \mathbf{v}_{ij}^k \right)}{\sum_{\mathcal{G}(i,j)=g} 1}, \tag{5.16}$$

meaning $\mathbf{z}_g$ is found by averaging all entries of $\mathbf{y}_i^{k+1}(e) + \frac{1}{\rho}\mathbf{v}_i^k(e)$ that correspond to the global index g. Similar to single-dimensional consensus optimization, after the first iteration,

$$\sum_{\mathcal{G}(i,j)=g} \mathbf{v}_{ij}^k = 0, \tag{5.17}$$

i.e., the sum of the dual variable entries that correspond to any given global index g is zero. Thus, the $\mathbf{z}$-update step can be written as

$$\mathbf{z}_g^{k+1} = \frac{1}{n_g} \sum_{\mathcal{G}(i,j)=g} \left(\mathbf{y}_{ij}^{k+1} \right), \tag{5.18}$$

where n_g is the total number of local couplings that correspond to the global consensus $\mathbf{z}_g$. In other words, the global consensus update step is only a local averaging step for each component $\mathbf{z}_g$, rather than a global averaging. In other words, only the agents that are linked to a component $\mathbf{z}_g$ will impact on the value of $\mathbf{z}_g$ (Boyd et al., 2011).

5.5 Concluding Remarks

This chapter details a network target coordination (NTC) method based on consensus optimization and alternating direction method of multipliers (CADMM) for engineering design problems. In this method, the original system is decomposed into multiple subsystems, which are solved by agents to generate and maintain the estimates of the coupling variables. All agents communicate the values of these estimates to their neighboring agents over a connected multi-agents network. The distributed coordination method utilize techniques found in the nonlinear programming literature. The primary techniques used here are augmented Lagrangian relaxation, consensus optimization, and alternating direction method of multipliers.

The NTC method offers a large degree of flexibility to the designer through the introduction of consensus optimization and a multi-agent network model. Based on the consensus estimate, the coupling variables shared by multiple agents are coordinated through an arithmetic mean agreed by these agents. Because of the disagreement term, each agent is able to modify the consensus estimate to achieve the optimality and feasibility of individual subsystems. The multi-agent network model provides the designer with the opportunity to coordinate the coupling variables in a nonhierarchical organizational structure of the design problem, which may be desired if the design problem did not fit as prescribed by the hierarchical, multi-level coordination method (ATC).

More specifically, the ATC method formulated the problem with a multi-level hierarchical structure and required the coupling variables shared by multiple subsystems at the same level to be coordinated by their parent subsystem. This type of coordination

method introduces additional consistency constraints in order to ensure that the coupling variables computed by the parent subsystems have the same value for the children subsystems. Unlike ATC, NTC has a non-hierarchical structure and coordinates the coupling variables locally at each agent. Thus, there is no parent-children relationship between agents and no need to introduce additional consistency constraints. Furthermore, the NTC method potentially allows all agents to be updated either in a Jacobi-type or in a Gauss-Seidel iteration schemes.

In summary, the network coordination method using the CADMM algorithm provides: 1) a completely separable formulation for representing the decomposed systems, 2) a flexible distributed coordination method for optimizing decomposed systems, 3) an explicit, efficient algorithm for solving subproblems in-parallel, and 4) a distributed, simple, and efficient guide for updating the Lagrangian multipliers $\mathbf{v}_i$ in each node. In addition, the vector of Lagrangian multipliers $\mathbf{v}_i$ can also be calculated by using subgradient algorithms (Wang et al., 2010, 2012) as presented in Chapter 4, extending the method to solve non-differentiable optimization problems.

Chapter 6

Numerical Applications of Network Target Coordination

6.1 Introduction

In this chapter, numerical applications of the network target coordination (NTC) method via the consensus alternating direction method of multipliers (CADMM) proposed in the previous chapter is implemented using three example problems. These problems exhibit the distributed coordination, in which all agents are solved concurrently and each agent is characterized by a local objective function and constraint set. The first example is a nonconvex geometric programming problem used by Kim et al. (2006) and Tosserams et al. (2006a). The second example is the speed reducer problem, a nonconvex engineering problem, taken from the works of Golinski (1970), Tosserams et al. (2007), and Lu and Kim (2010). The third example is based on analysis models of an ADXL150 micro-accelerometer from Devices (1998), Zhou (1998), Samuels (1996), and Tosserams et al. (2010), also a nonconvex engineering problem.

The results obtained using the NTC method via CADMM are also compared to the ones solved by using analytical target cascading (ATC) via alternating direction method of multipliers (ADMM) (Tosserams et al., 2006b). The numerical performance of these two coordination methods is quantified using three measurements, namely the solution

error e_{sol}, the maximal design constraint violation max-con, and the number of iterations k. The solution accuracy e_{sol} is defined as

$$e_{\text{sol}} = \left\| \mathbf{1} - \bar{\mathbf{x}}^k_{\text{scaled}} \right\|_\infty \tag{6.1}$$

where $\bar{\mathbf{x}}^k_{\text{scaled}}$ consists of the components of the optimal solution for the decomposed problem $\bar{\mathbf{x}}^k$ scaled using the AIO optimal solution $\mathbf{x}^*$, i.e., $\bar{\mathbf{x}}^k_{\text{scaled}} = \bar{\mathbf{x}}^k / \bar{\mathbf{x}}^*$. The infinity norm is defined as $\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \cdots, |x_n|)$.

To ensure that the optimal solutions obtained by these two coordination methods respectively are consistent for the overall system, two criteria are adopted from Tosserams et al. (2007) to check that the consistency constraint can converge to zero. First, when the reduction of the consistency constraint after two successive iterations became smaller than a user-defined termination tolerance ε, the solution procedure for all example problems is considered converged, i.e.

$$\frac{\left| \mathbf{c}^k_{ij} - \mathbf{c}^{k-1}_{ij} \right|}{1 + |\mathbf{y}^k_{ij}|} < \varepsilon, \quad j = 1, \cdots, m^{\mathbf{c}}, \tag{6.2}$$

where for agent i, the j-th component of the consistency constraint $\mathbf{c}_i$ is denoted by $\mathbf{c}_{ij}$. Similarly, $\mathbf{y}_{ij}$ is the j-th component of the vector of local coupling variables $\mathbf{y}_i$. The superscript k denotes the number of iterations. The denominator, $1 + |\mathbf{y}_{ij}|$, is used for scaling the consistency constraint and for ensuring that the denominator is not zero. Second, the maximal consistency constraint violation is also considered as a convergence criterion, i.e.

$$\frac{|\mathbf{c}^k_{ij}|}{1 + |\mathbf{y}^k_{ij}|} < \varepsilon. \tag{6.3}$$

Based on the duality theorem (Bertsekas, 2003), the consistency constraints of each agent are the gradients of a dual problem, which transform the relaxed subsystem design problem (5.15) into a maximization problem by finding the optimal Lagrangian multipliers $\mathbf{v}_i$. Since the Lagrangian multipliers are unbounded when solving the dual problem, the

optimal values of the Lagrangian multipliers can be found when the corresponding gradients of the dual problem are zero. Therefore, if the consistency constraints $\mathbf{c}_i$ are smaller than the termination tolerance $\varepsilon > 0$, the solution procedure for the relaxed subsystem design problem (5.15) is finished.

6.2 Example 1: Nonconvex Geometric Programming Problem

The first example, a nonconvex geometric programming problem used in earlier work on ATC (Kim, 2001, Michalek and Papalambros, 2005a, Tosserams et al., 2006a, Tzevelekos et al., 2003), was used to demonstrate that the solutions obtained with NTC using the CADMM approach can converge to the solutions found using an AIO approach. The example has 14 variables, six inequality constraints, and four equality constraints. The objective is convex, but the constraints are non-convex, violating the assumptions of the convergence proof of the augmented Lagrangian coordination algorithms. The AIO problem is given by

$$
\begin{aligned}
&\min f = x_1^2 + x_2^2 \\
&\text{s.t. } g_1 = \left(x_3^{-2} + x_4^2\right) x_5^{-2} - 1 \leq 0, \\
&\quad g_2 = \left(x_5^2 + x_6^{-2}\right) x_7^{-2} - 1 \leq 0, \\
&\quad g_3 = \left(x_8^2 + x_9^2\right) x_{11}^{-2} - 1 \leq 0, \\
&\quad g_4 = \left(x_8^{-2} + x_{10}^2\right) x_{11}^{-2} - 1 \leq 0, \\
&\quad g_5 = \left(x_{11}^2 + x_{12}^{-2}\right) x_{13}^{-2} - 1 \leq 0, \\
&\quad g_6 = \left(x_{11}^2 + x_{12}^2\right) x_{14}^{-2} - 1 \leq 0, \\
&\quad h_1 = \left(x_3^2 + x_4^{-2} + x_5^2\right) x_1^{-2} - 1 = 0, \\
&\quad h_2 = \left(x_5^2 + x_6^2 + x_7^2\right) x_2^{-2} - 1 = 0, \\
&\quad h_3 = \left(x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2\right) x_3^{-2} - 1 = 0, \\
&\quad h_4 = \left(x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2\right) x_6^{-2} - 1 = 0,
\end{aligned}
\tag{6.4}
$$

where $\bar{\mathbf{x}} = [x_1, x_2, \cdots, x_{14}] \geq \mathbf{0}$.

The unique optimal solution, obtained through the sequential quadratic programming (SQP) method offered by fmincon in MATLAB R2011(a), to the AIO problem is found at $\mathbf{x}^* = [2.84,\ 3.09,\ 2.36,\ 0.76,\ 0.87,\ 2.81,\ 0.94,\ 0.97,\ 0.87,\ 0.80,\ 1.30,\ 0.84,\ 1.76,\ 1.55]$ and the corresponding objective value is $f(\mathbf{x}^*) = 17.59$.

Problem Decomposition

Four decompositions were selected to illustrate the performance of the proposed method, the details being presented in Figures 6.1 and 6.2. Each box represents an agent, and the link between two boxes represents coupling variables shared by both the agents. The variables, objective function, and constraint functions are distributed among corresponding boxes. The number of coupled variables increases with the decomposition index, while the number of local variables and functions decreases with the decomposition index. For simplification, the augmented term for each agent in Eq. (5.15) is denoted by $\phi_i,\ i = 1, \cdots, m$.

Experimental Setup

All four decompositions were solved using both the CADMM and ADMM approaches. The initial design for the problem was randomly selected from the five initial designs given by Tosserams et al. (2006a). For ADMM, the initial weight setting strategy was the one presented in Tosserams et al. (2006a), where the initial dual variable and penalty parameter are set to $\mathbf{v} = \mathbf{0}$ and $\mathbf{w} = \mathbf{10}^{-3}$, respectively. The parameters for updating $\mathbf{w}$ of ADMM are set to $\beta = 1.1, \gamma = 0.9$. Several initial values for the penalty parameter ρ were tried using the following scheme

$$\rho = \frac{2|\hat{f}_i|}{\mathbf{c}_i^T \mathbf{c}_i}, \tag{6.5}$$

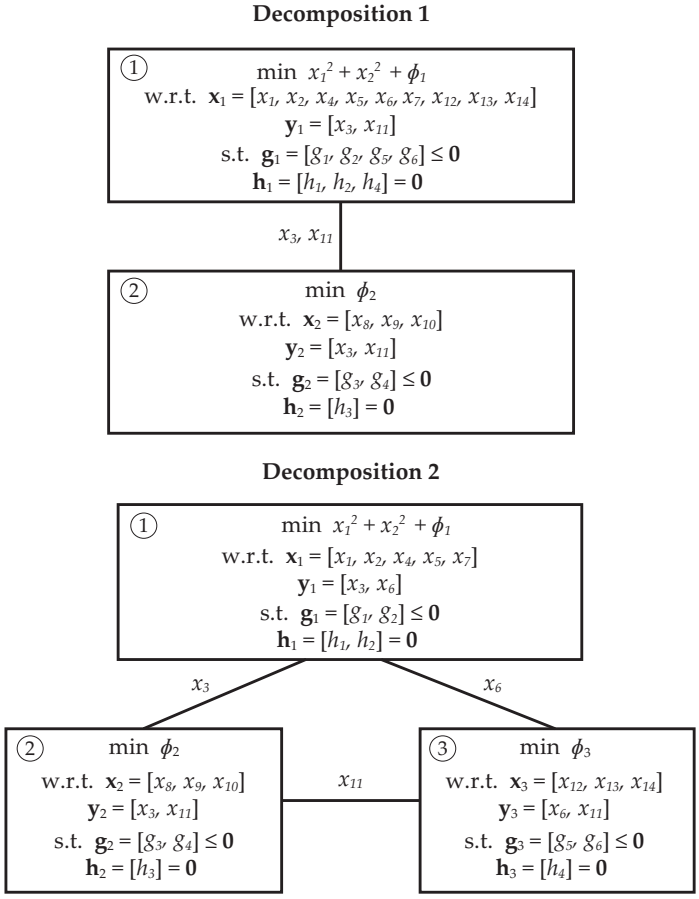


Figure 6.1: Network frameworks of the geometric programming problem: decompositions 1 and 2

where agent i has both local objective function and consistency constraints, and their approximated values were set to $|\hat{f}_i| = 10$ and $\mathbf{c}_i = \mathbf{1}$. The corresponding penalty parameters for all four decomposition cases were set to $\rho = \frac{20}{2} = 10$. There is no need to update the penalty parameter ρ during the solution procedure. This problem was solved considering the termination tolerances $\varepsilon = [10^{-3},\ 10^{-4},\ 10^{-5},\ 10^{-6}]$.

Numerical Results

The results for the numerical experiments, summarized in Table 6.1, show that both the ATC with ADMM and NTC with CADMM were able to find an optimal solution close to the AIO solution. More specifically, both the ADMM and CADMM

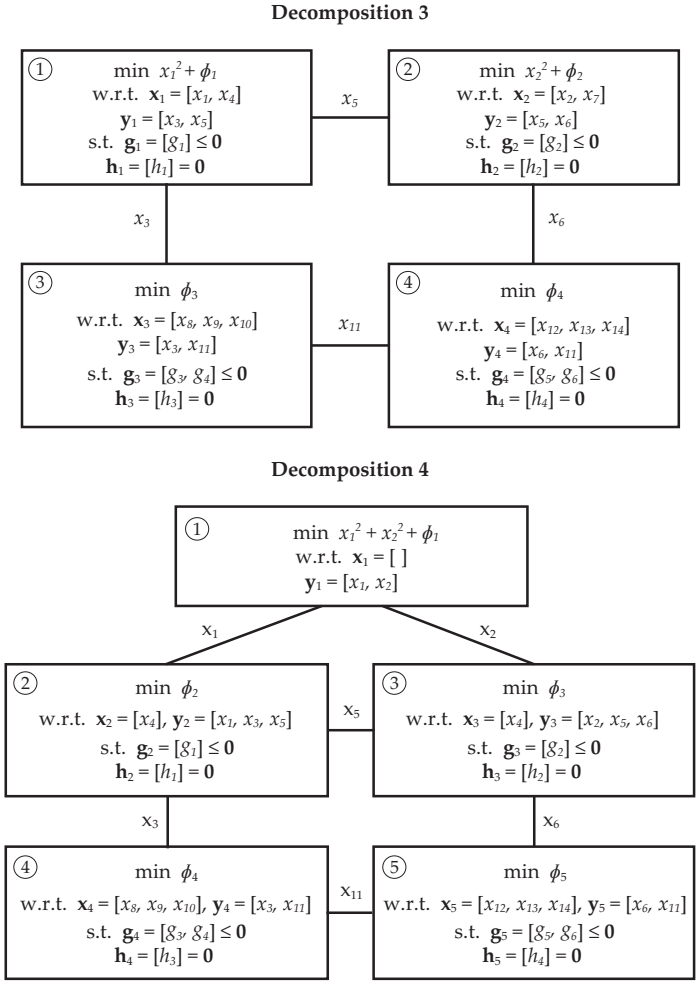


Figure 6.2: Network frameworks of the geometric programming problem: decompositions 3 and 4

were improved by taking more iterations with smaller termination tolerances. Considering the maximum constraint violations, the results indicate that CADMM can satisfy a very strict constraint tolerance. Considering the accuracy in terms of solution error e_{sol}, CADMM performs better than ADMM, resulting in an improved magnitude of solution error by a factor of 100 or 1000. The optimal solutions obtained here using the ADMM approach (Tosserams et al., 2006a) are slightly different from the ones published, and the possible reason may be the use of difference nonlinear programming algorithms for solving each subsystem.

Table 6.1: Results for the decomposed geometrical programming problem

Decomp		ATC-ADMM				NTC-CADMM			
	ε	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}
	$1e-3$	17.58	21	0.0014	6.304e-4	17.58	13	3.141e-4	3.893e-4
1	$1e-4$	17.59	35	1.056e-4	1.842e-4	17.58	18	1.907e-6	2.130e-5
	$1e-5$	17.59	98	1.499e-5	5.256e-5	17.59	21	1.294e-6	8.081e-6
	$1e-6$	17.59	151	6.524e-7	9.399e-4	17.59	25	5.774e-7	4.630e-7
	ε	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}
	$1e-3$	17.57	21	0.0015	8.157e-4	17.57	40	8.896e-5	8.120e-4
2	$1e-4$	17.59	30	6.156e-5	3.870e-5	17.59	52	1.510e-4	1.360e-4
	$1e-5$	17.59	112	5.889e-6	8.147e-4	17.59	68	4.376e-6	1.101e-5
	$1e-6$	17.59	152	6.925e-7	0.0010	17.59	84	1.739e-7	1.323e-6
	ε	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}
	$1e-3$	17.57	20	0.0023	0.0012	17.58	40	7.893e-5	7.648e-4
3	$1e-4$	17.59	49	0.269e-4	5.260e-4	17.59	52	1.417e-4	1.290e-4
	$1e-5$	17.59	134	1.828e-8	8.875e-4	17.59	68	4.183e-6	1.054e-5
	$1e-6$	17.59	164	1.163e-6	5.091e-4	17.59	84	1.315e-7	1.115e-6
	ε	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}
	$1e-3$	17.61	21	0.0012	0.0014	17.54	46	0.0023	0.0021
4	$1e-4$	17.59	39	3.434e-4	4.522e-4	17.59	73	1.421e-4	1.631e-4
	$1e-5$	17.59	114	1.622e-5	9.732e-4	17.59	99	4.864e-6	1.157e-5
	$1e-6$	17.59	162	1.807e-6	8.067e-4	17.59	121	5.246e-7	1.250e-6

The objective value $f(\bar{\mathbf{x}}^*)$ is the optimal objective value obtained with respect to the optimal solution $\bar{\mathbf{x}}^*$ achieved at iteration k.
The parameter max-con denotes the maximum constraint violation achieved at iteration k.
The parameter e_{sol} denotes the solution error achieved at iteration k.

6.3 Example 2: Golinski's Speed Reducer Problem

The second example is a speed reducer problem taken from the works of Golinski (1970), Tosserams et al. (2007), and Lu and Kim (2010). The speed reducer includes two mating gears inside a gear box and two shafts, one input and one output. The design variables are the dimensions of the gears (x_1, x_2, x_3) and of both shafts (x_4, x_6 and x_5, x_7) depicted in Figure 6.3.

The problem was decomposed into three subsystems, one subsystem minimizing the weight of the reducer and the other two minimizing the stresses on their assigned shafts by satisfying the stress, deflection, and geometric constraints. The AIO problem

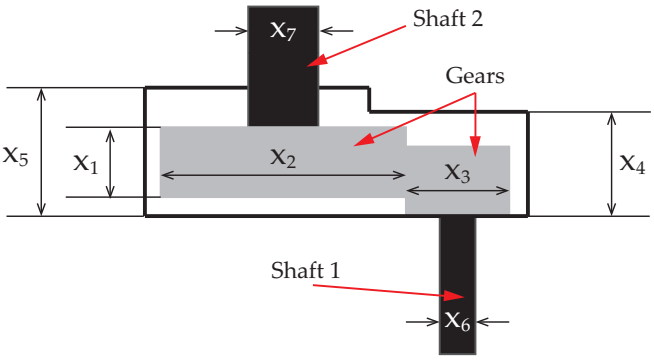


Figure 6.3: A schematic of Golinski's speed reducer

for designing the speed reducer is given by

$$\min_{\bar{\mathbf{x}}=[x_1,\cdots,x_7]} f(\mathbf{x}) = \sum_{i=1}^{7} f_i$$

$$\begin{aligned}
\text{s.t.} \quad & \mathbf{g}_{\text{gear}} = [g_5, g_6, g_9, g_{10}, g_{11}]^T \leq \mathbf{0}, \\
& \mathbf{g}_{(\text{shaft, 1})} = [g_1, g_3, g_7]^T \leq \mathbf{0}, \\
& \mathbf{g}_{(\text{shaft, 2})} = [g_2, g_4, g_8]^T \leq \mathbf{0}, \\
& 2.6 \leq x_1 \leq 3.6,\ 0.7 \leq x_2 \leq 0.8, 17.0 \leq x_3 \leq 28.0, \\
& 7.3 \leq x_4 \leq 8.3,\ 7.3 \leq x_5 \leq 8.3,\ 2.9 \leq x_6 \leq 3.9,\ 5.0 \leq x_7 \leq 5.5, \qquad (6.6) \\
\text{where} \quad & F_1 = 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9335 x_3 - 43.0934), \\
& F_2 = -1.5079 x_1 x_6^2,\ F_3 = -1.5079 x_1 x_7^2,\ F_4 = 7.477 x_6^3, \\
& F_5 = 7.477 x_7^3,\ F_6 = 0.7854 x_4 x_6^2,\ F_7 = 0.7854 x_5 x_7^2, \\
& g_1 = \frac{1}{110 x_6^3} \sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 1.69 \cdot 10^7} - 1, \\
& g_2 = \frac{1}{85 x_7^3} \sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 1.575 \cdot 10^8} - 1, \\
& g_3 = \frac{1.5 x_6 + 1.9}{x_4} - 1,\ g_4 = \frac{1.1 x_7 + 1.9}{x_5} - 1,\ g_5 = \frac{27}{x_1 x_2^2 x_3} - 1, \\
& g_6 = \frac{397.5}{x_1 x_2^2 x_3} - 1,\ g_7 = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1,\ g_8 = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1, \\
& g_9 = \frac{x_2 x_3}{40} - 1,\ g_{10} = \frac{5 x_2}{x_1} - 1,\ g_{11} = \frac{x_1}{12 x_2} - 1.
\end{aligned}$$

The optimal objective value for the AIO problem is achieved at $f(x^*) = 2994.4$ with the optimal solution being $\mathbf{x}^* = [3.5000,\ 0.7000,\ 17.0000,\ 7.3000,\ 7.7153,\ 3.3502,\ 5.2867]^T$.

Problem Decomposition

The problem decomposition used here is depicted in Figure 6.4. The design variables, objective functions, and constraint functions are distributed among three agents. The AIO design variable set $\bar{\mathbf{x}}$ was also assigned to these three subsystems: x_4 and x_6 being assigned to shaft 1 subsystem, and x_5 and x_7 to shaft 2 subsystem; the gear subsystem did not have any local design variables. Design variables x_1, x_2, and x_3 represent coupling variables shared by three subsystems. In the language of the design specifications, the three subsystems collaboratively determined the values of the thickness and radiuses of shafts 1 and 2.

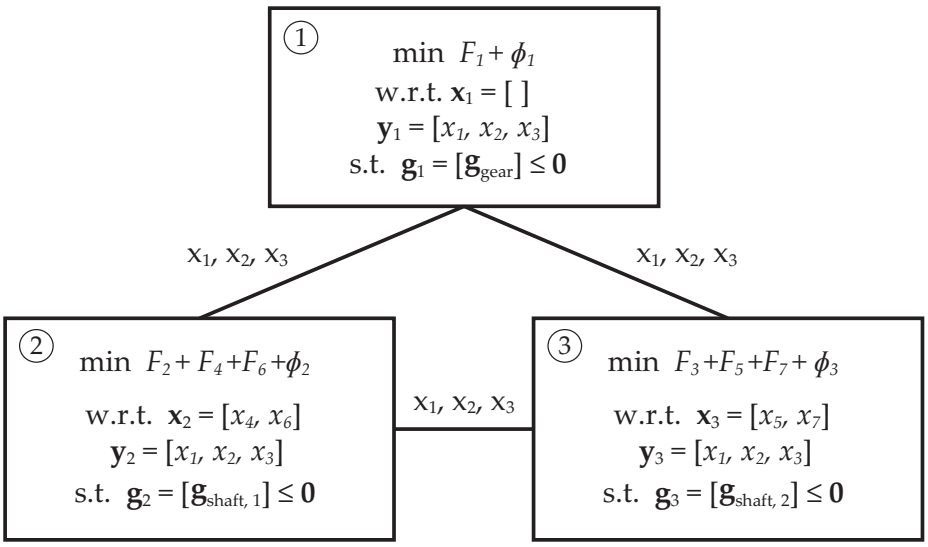


Figure 6.4: Network framework of Golinski's speed reducer problem

Experimental Setup

As in the previous example, the performances of the NTC via CADMM and ATC via ADMM were compared. For ADMM, the initial penalty weight is set to $w^0 = 10^3$. The parameters for updating $\mathbf{w}$ of ADMM are set to $\beta = 1.1$ and $\gamma = 0.9$. For CADMM, the

initial guess for the Lagrangian multiplier $\mathbf{v}_i$ was set to $\mathbf{0}$. The penalty parameter was set to $\rho = 10^2$. For both the CADMM and ADMM approaches, five randomly selected initial values were tested, similar to Tosserams et al. (2007). All subsystems were solved using the interior-point algorithm offered by fmincon using default settings. This problem is solved with the termination tolerance set to $\varepsilon = 10^{-3},\ 10^{-4},\ \cdots,\ 10^{-10}$.

Numerical Results

As in the previous demonstrated example problem, different algorithms are compared: the ADMM approach with the quasi-separable coordination method and the CADMM approach with distributed coordination methods. The results are presented in Table 6.2, showing that the optimal solution can be obtained using these two algorithms and are in excellent agreement with the AIO optimal solution. When comparing the number of iterations k, the CADMM approach can achieve a better computational efficiency than the ADMOM approach. When considering the maximum constraint violations, the results indicate that the CADMM approach can satisfy a very strict constraint tolerance, and the inequality constraint corresponding to this max-con is active at the optimal solution[1]. When comparing the performance of these two algorithms in terms of the solution error e_{sol}, the CADMM approach can converge to a more accurate optimal solution than the ADMM approach.

6.4 Example 3: Micro-accelerometer Design Problem

The third example is a micro-accelerometer design problem based on an ADXL150 style accelerometer from Analog Devices (Devices, 1998, Samuels, 1996). Many MEMS-based micro-accelerometers use a capacitive-sensing scheme for acceleration detection and have been widely used in the automotive, the robotics, and other industries. For this research, the example problem includes four analysis models, representing design aspects

[1]Active constraint. M. Hazewinkel (originator), Encyclopedia of Mathematics. http://www.encyclopediaofmath.org/index.php?title=Active_constraint&oldid=14642

Table 6.2: Results for the decomposed Golinski's speed reducer design problem

Tolerance	ATC-ADMM				NTC-CADMM			
ε	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}	$f(\bar{\mathbf{x}}^*)$	k	max-con	e_{sol}
$1e-3$	2994.0	26	2.7561e-04	2.1431e-04	2996.6	19	9.0518e-4	3.1231e-4
$1e-4$	2994.3	35	2.7532e-05	2.1413e-05	2994.5	24	5.8782e-10	6.4458e-5
$1e-5$	2994.4	46	1.8544e-06	1.4423e-06	2994.3	28	5.9880e-06	4.6573e-6
$1e-6$	2994.4	58	1.5657e-07	1.2178e-07	2994.4	35	5.8782e-10	8.2286e-8
$1e-7$	-	-	-	-	2994.4	43	3.5890e-8	2.7915e-8
$1e-8$	-	-	-	-	2994.4	50	5.8782e-10	5.1624e-10
$1e-9$	-	-	-	-	2994.4	57	5.8782e-10	3.9815e-10
$1e-10$	-	-	-	-	2994.4	74	5.8782e-10	1.6485e-10

The objective value $f(\bar{\mathbf{x}}^*)$ is the optimal objective value obtained with respect to the optimal solution $\bar{\mathbf{x}}^*$ achieved at iteration k.
The parameter k denotes the iteration number.
The symbol $-$ indicates that the solution cannot be obtained using ADMM.

of structures, electrostatics, dynamics, and electronics (Mukherjee, 1998, Senturia, 2001, Tosserams et al., 2010, Zhou, 1998). A simplified schematic of such a capacitive-sensing accelerometer is shown in Figure 6.5.

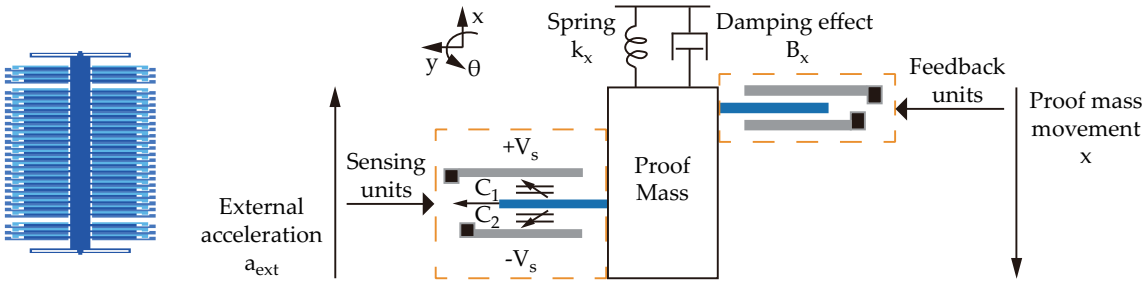


Figure 6.5: Schematic of a capacitive-sensing accelerometer (Tosserams et al., 2010)

The micro-accelerometer design problem is formulated as four optimization problems, each attempting to find optimal settings for input parameters such that the design specifications and desired device behavior are obtained (Tosserams et al., 2010). These cases differ with respect to which design specifications are selected as optimization variables; for example, Case 1 includes only the proof mass and U-spring dimensions as optimization variables, while Case 4 includes 22 optimization variables related to elec-

trostatics, dynamics, circuit, and responses subproblems. The analysis model given by Tosserams et al. (2010) requires that values are assigned to all input parameters (design specifications of the micro-accelerometer analysis model). A baseline design is then defined by using the original ADXL150 accelerometer parameters for a maximum measure frequency of $\omega = 1000$ Hz. Based on the baseline design, the formulation of the optimal design problem of the accelerometer is given by:

$$\begin{aligned} \min_{\mathbf{x},\, A_{\max}} \quad & A_{\max} \\ \text{subject to} \quad & g_{c,1} = -\frac{S(\mathbf{x})}{S_{\min}} + 1 \leq 0 \\ & g_{c,2} = \frac{a_n(\mathbf{x})}{a_{\min}} - 1 \leq 0 \\ & g_{c,5} = -\frac{a_{fs(\mathbf{x})}}{a_{\text{means}}} + 1 \leq 0 \\ & \mathbf{g}_f(\mathbf{x}) \leq 0 \\ & \mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub} \\ & \bar{\mathbf{x}} = [\mathbf{x}, A_{\max}] \end{aligned} \tag{6.7}$$

where $\mathbf{x}$ represents the vector of the input parameters that are selected as design variables. Parameter $A_{\max}$ of constraints $g_{s,16}$ and $g_{s,17}$ is included as an artificial optimization variable to avoid nonsmoothness in the definition of the area A (Tosserams et al., 2010). The design constraints ensure that performance with respect to sensitivity, noise, and range is at least as good as the baseline design (i.e., $S_{\min} = S(\mathbf{x}_{\text{base}})$, $a_{\min} = a_n(\mathbf{x}_{\text{base}})$, and $a_{\text{means}} = a_{fs}(\mathbf{x}_{\text{base}})$, where $\mathbf{x}_{\text{base}}$ is the baseline design.) The functional constraints $\mathbf{g}_f = [g_{s,1-17}, g_{e,1-6}, g_{d,1-4}, g_{c,3}, g_{c,4}, g_{c,6}]$ assure the performance and functioning of four subsystems, i.e., structures, electrostatics, circuits, and dynamics.

The optimal solutions for the AIO problem are obtained by analyzing three cases. Case 1 considers seven variables of the proof mass and U-springs as design variables. Case 2 adds nine sense and feedback unit variables to the first case. In Case 3, four more circuit design variables are added. The results for solving the AIO problem are shown

in Table 6.3. The optimal results obtained are slightly different from the results given by Tosserams et al. (2010), e.g., the width of beam 2 of the U-spring $w_{b2} = 5.500$, which increases the maximum area $A_{\max}$ of the micro-accelerometer from 0.1880 to 0.1916. This slight difference might due to the settings of the optimization solver. All three cases in this research are solved using the sequential quadratic programming (SQP) solver offered by fmincon of Matlab R2011(a) with default settings except for "Maxfunevals = 100000" and "Maxiter = 200". The modification of these two settings prevents iterations that do not terminate in fmincon. Similar to the scaling scheme used by Tosserams et al. (2010), the vector of the design variables $\mathbf{x}$ is also scaled with respect to their baseline values, i.e. $\bar{\mathbf{x}}_{\text{scaled}} = \bar{\mathbf{x}}/\bar{\mathbf{x}}_{\text{base}}$.

Table 6.3: Optimal results for three cases of the AIO micro-accelerometer problem

Components	Variables ($\bar{\mathbf{x}}$)		Boundary		Optimal Designs			
			Lower	Upper	Baseline	Case 1	Case 2	Case 3
Objective Value	$A_{\max}$	$[mm^2]$	0.01	1.0	0.5000	0.1916	0.0962	0.0807
Proof mass & U-springs	l_p	$[\mu m]$	2	700	500	476.7	354.7	355.9
	w_p	$[\mu m]$	2	400	50	**47.10**	80.80	81.59
	l_{b1}	$[\mu m]$	2	400	125	107.0	124.4	103.2
	l_{b2}	$[\mu m]$	1	200	6.00	2.100	1.500	1.500
	w_b	$[\mu m]$	2	10	2.40	2.000	2.000	2.000
	w_{b2}	$[\mu m]$	2	10	4.00	**5.500**	2.000	2.000
	$l_{b3}{}^*$	$[\mu m]$	2	400	105	**88.48**	**88.95**	**67.41**
	$l_{ls}{}^*$	$[\mu m]$	1	100	2.00	1.400	1.000	1.000
Sense & Feedback	l_f	$[\mu m]$	2	400	120	120	39.00	17.41
	l_{ov}	$[\mu m]$	2	400	114	114	37.80	16.30
	g_s	$[\mu m]$	0.5	20	1.3	1.3	0.844	0.800
	g_{su}	$[\mu m]$	0.5	20	1.3	1.3	0.500	0.500
	g_f	$[\mu m]$	0.5	20	1.3	1.3	0.795	0.953
	g_{fu}	$[\mu m]$	0.5	20	1.3	1.3	0.500	0.500
	g_x	$[\mu m]$	0.5	20	0.7	0.7	0.500	0.500
	w_f	$[\mu m]$	2	20	4.00	4.00	5.157	5.202
	w_s	$[\mu m]$	2	20	4.00	4.00	2.000	2.000
Circuit	V_{s0}	$[V]$	1	4	2.5	2.5	2.5	4.000
	V_d	$[V]$	1	4	2.5	2.5	2.5	1.640
	C_a	$[fF]$	1	1000	350	350	350	231
	G_{ni}	$[-]$	1	20	17	17	17	20

- Values colored in light grey are not optimized but inherited from the baseline design.
- Dependent variables marked with an asterisk (*) are calculated based on $l_{b3} = l_{b1} + w_a/2 - w_p/2$ and $l_{ls} = l_{b2} - g_x$.

Problem Decomposition

The problem decomposition presented in Figure 6.6 illustrates a method by which the micro-accelerometer design problem can be partitioned into three subsystems, i.e., circuit, sensor dynamics, and sensor geometry. The distribution of design variables, objective functions, and constraint functions are also depicted in this figure. The AIO design variable set $\bar{\mathbf{x}}$ is separated into these three subsystems: C_a and G_{ni} are local design variables of the circuit subsystem; components of the set $\bar{\mathbf{x}}$ except C_a and G_{ni} are local design variables of the sensor geometry subsystem; the sensor dynamics subsystem does not have a local design variable. The linking variables except for V_{s0} are not optimization variables of the AIO problem but intermediate analytical quantities introduced as a result of decomposition. The single arrows in the figure indicate the dependency relationships of these intermediate analytical quantities.

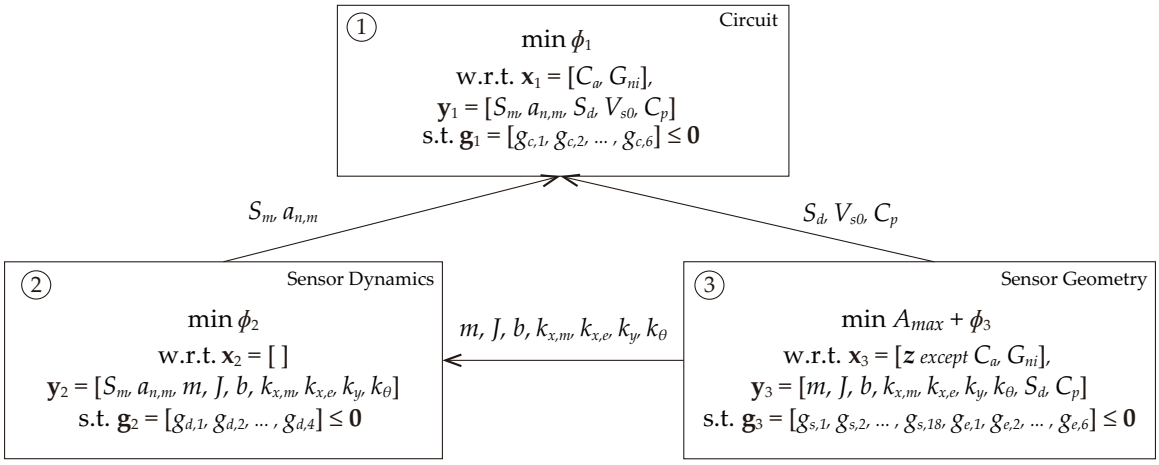


Figure 6.6: Network framework of the micro-accelerometer problem

Experimental Setup

The performance of the CADMM approach is again compared to that of the ADMM approach. For both the CADMM and ADMM approaches, ten randomly selected initial values were investigated. The ADMM approach uses the same initial weight

setting strategy given by Tosserams et al. (2010) with $\mathbf{v} = \mathbf{0}$ and $\mathbf{w} = \mathbf{10}^{-3}$. The parameters for updating the $\mathbf{w}$ in ADMM were set to $\beta = 1.1$ and $\gamma = 0.9$. For the CADMM, the initial values of the dual variables were set to $\mathbf{v} = \mathbf{0}$. Based on the initial settings seen in Eq.(6.5), the estimates were set to $\hat{f}_i = 2.5$ and $\mathbf{c}_i = 1$ (these estimates are scaled values), and the penalty parameter was obtained and set to $\rho = 5/10 = 0.5$ for all three cases. All subsystems were solved with the SQP algorithm offered by fmincon using default settings except for "Maxfunevals = 100000" and "Maxiter = 200". This problem was solved with a termination tolerance set to $\epsilon = 0.001$.

Numerical Results

In all three cases, the results shown in Table 6.4 indicate that both the ATC with ADMM and the NTC with CADMM approaches were able to find optimal solution close to the AIO optimal solution. By comparing the objective value $A_{\max}$, CADMM outperforms ADMM except in Case 1. By comparing the minimum number of iterations, CADMM outerperforms ADMM. A possible explanation may be that the use of consensus estimates and disagreement terms speeds up the convergence. By comparing the maximum constraint violations, CADMM outperforms ADMM since the optimal solutions obtained by using the latter cannot strictly satisfy the constraint tolerance. By comparing the solution accuracy, ADMM outperforms CADMM except in Case 2. A possible explanation for this finding may be that the solution accuracy of NTC with CADMM adopting the Jacobian-type iteration (all agents are simultaneously updated) was no better than the solution accuracy of ATC with ADMM using the Gauss-Seidel (subsystems are updated one at a time) (Tsitsiklis, 1989). For these three cases, both the ADMM and CADMM approaches provided optimal objective values $A_{\max}$ that were not sensitive to the randomly selected initial points.

In addition, both the CADMM and ADMM approaches may have difficulties with nonconvex functions. The reason the solution accuracy of these two approaches is relatively poor for all three cases is still unknown. But a similar situation was observed in the

results published in Tosserams et al. (2010) and it should be noticed that the results (in bold) obtained here using the ATC with ADMM approach are much better than the results published. A possible explanation may be due to the following reasons. The optimization variables for all three cases were involved in high non-convex constraint functions. However, based on the duality theorem, both the objective and constraint functions of the optimization problems are required to be convex for the methods to converge. In addition, the optimization algorithm (such as SQP offered by MATLAB 2011(a)) used for solving subsystems may also lack the ability to find global optimal solutions for problems with high non-convexity, resulting in extra computational difficulty for the coordination procedure.

Table 6.4: Results for the decomposed micro-accelerometer problem

	AIO		ATC-ADMM				NTC-CADMM			
	$A_{\max}$		$A_{\max}$	k	Max-Con	e_{sol}	$A_{\max}$	k	Max-Con	e_{sol}
		Min	0.1934	66	0.0010	1.1723	0.1964	31	0	1.3383
Case 1	0.1916	Mean	0.1934	69	0.0013	1.1807	0.1964	31	6.478e-11	1.3898
100%|100%		Max	0.1934	73	0.0016	1.1909	0.1965	31	2.104e-10	1.6005
		Min	0.0988	56	0.0012	1.7716	0.0973	53	0	1.7487
Case 2	0.0962	Mean	0.0992	57	0.0015	2.5988	0.0978	64	5.651e-4	2.5855
100%|100%		Max	0.1000	60	0.0017	3.9895	0.0982	79	0.0023	3.7677
		Min	**0.0854**	55	8.527e-4	2.0263	0.0820	54	4.707e-14	2.1779
Case 3	0.0807	Mean	**0.0859**	59	0.0015	2.0582	0.0825	68	1.867e-4	2.2123
100%|100%		Max	**0.0864**	62	0.0030	2.0780	0.0830	110	7.042e-4	2.2270

- For each case, the percentage of converged initial points is indicated on the first column.
- Although the solution error e_{sol} obtained for CADMM is comparably larger than the results of ADMM, all these optimal results satisfy the design lower and upper specifications and, therefore, are considered useful for a practical engineering design.
- Values in bold are optimal results significantly different from the results observed by Tosserams et al. (2010), in which Case 3 did not converge to a solution close to its AIO optimal solution.

6.5 Concluding Remarks

This chapter analyzed the numerical performance of NTC via CADMM for solving three nonconvex, nonlinear optimization problems, including a geometrical programming problem, a Golinski's speed reducer design problem, and a micro-accelerometer design

problem. The convergence of the NTC method was studied and preliminary results of numerical experiments were presented. The comparison of the proposed NTC via CADMM and ATC via ADMM was demonstrated using the results of numerical experiments. The results indicate that NTC via CADMM is more efficient and robust than ATC via ADMM. In terms of efficiency, the CADMM can achieve converged optimal solutions after a few iterations. In terms of robustness, the optimal solutions obtained by using CADMM were not sensitive to the randomly selected initial points. The CADMM also has advantages. This approach offers a large degree of freedom in updating the dual variables, favorable when coding the decomposed problems. Additionally, this approach can achieve optimal solutions with efficiency and robustness by using a simplified penalty parameter update scheme.

More specifically, NTC via CADMM seems to outperform ATC via ADMM because the locally averaging step for updating the consensus eliminates the influence of Lagrangian multipliers and penalty parameters. Compared to the analytical master solution of the coupling variables $\mathbf{y}$ defined as

$$\mathbf{y} = \frac{\sum_{i=1}^{M} \mathbf{w}_i \circ \mathbf{w}_i \circ \mathbf{y}_i - \frac{1}{2}\sum_{i=1}^{M} \mathbf{v}_i}{\sum_{i=1}^{M} \mathbf{w}_i \circ \mathbf{w}_i} \tag{6.8}$$

by Tosserams et al. (2006a), the local updating scheme

$$\mathbf{z}_g^{k+1} = \frac{1}{n_g} \sum_{\mathcal{G}(i,j)=g} \mathbf{y}_{ij}^{k+1}$$

given in the previous chapter does not include the information of Lagrangian multipliers and penalty parameters. Although ideally, the penalty parameters may be increased to large values, which drive the term

$$\frac{\frac{1}{2}\sum_{i=1}^{m} \mathbf{v}_i}{\sum_{i=1}^{m} \mathbf{w}_i \cdot \mathbf{w}_i}$$

to a near zero value and may result in ill-conditioning of the problem. Furthermore, in

the NTC method, the agents that are linked to a component $\mathbf{z}_g$ will impact the value of $\mathbf{z}_g$; however, the analytical master solution $\mathbf{y}$ given in Eq. (6.8) indicates that the targets of coupling variables for subsystem i will also be influenced by all Lagrangian multipliers and penalty parameters. Thus, NTC via CADMM uses a Jacobi-type iteration to coordinate all subsystems, while ATC via ADMM uses a combination of Gauss-Siedel and Jacobi iterations to coordinate all subsystems. As a result, NTC via CADMM can provide a simpler yet accurate update scheme for determining the value of the coupling variables during the solution process, resulting in the reduced number of iterations and improved solution accuracy.

Although the numerical performance of the NTC with CADMM approach is efficient and accurate when solving the example problems, further research is still needed. In this research, the micro-accelerometer design problem provides only one type of decomposition and should be investigated further by applying the proposed method to other types of decompositions. An investigation focusing on improving solution accuracy when solving problems with high nonconvexity should be conducted.

Chapter 7

Conclusions

7.1 Contributions

The fundamental contribution of this dissertation is the development of mathematical formulations and corresponding algorithms for realizing the large-scale, distributed design of both hierarchically and nonhierarchically decomposed systems. From an application viewpoint, the research contributes to the reduction of computational costs and the improvement of solution accuracy for solving these design problems potentially completed by geographically dispersed teams.

When studying the hierarchically decomposed complex systems, the research presented in Chapter 4 contributes to the development of two new subgradient-based algorithms. These two new algorithms, linear and proximal cutting plane methods, were implemented based on augmented Lagrangian relaxation techniques and compared to the traditional update schemes used in subgradient algorithms. The results of three nonconvex nonlinear examples suggests that significant computational benefits can be achieved by using the cutting plane methods.

Furthermore, a biobjective approach was also developed for effectively solving hierarchically decomposed problems. Based on the analytical target cascading, this approach was introduced to optimize two performance measures of the system by means

of minimizing the deviations from the fixed design requirements and the local objective. In this biobjective approach, the generic algorithm for two-level, hierarchically decomposed problems was proposed. The algorithm can be easily adapted for problems with any number of lower-level subproblems, as demonstrated by the analytical mass allocation problem. The convergence of the algorithm is also verified based on the established results in nonlinear programming for block coordinate descent methods.

When studying nonhierarchically decomposed complex systems, the network target coordination (NTC) method was proposed using the consensus optimization with the alternating direction method of multipliers (CADMM) in Chapters 5. In this coordination method, a complex system was partitioned into a number of smaller, manageable subsystems, which were independently solved by agents. Since these subsystems are interact with one another, a coordination strategy was introduced to generate the consensus estimates of coupling variables for all agents. When using the proposed NTC method, it was assumed that the design optimization of each subsystem operating in parallel takes the same amount of time, and if one subsystem analysis finishes early, it waits on the others to finish. During the coordination process, all agents communicate the values of these consensus estimates to their neighboring agents over a connected multi-agents network.

The numerical performance of the NTC method was studied and preliminary results were presented in Chapter 6. This chapter demonstrated the performance of the proposed NTC method by solving three nonconvex, nonlinear optimization problems, including a geometrical programming problem, a Golinski's speed reducer design problem, and a micro-accelerometer design problem. The comparison of the proposed NTC with CADMM approach and the ATC with the the alternating direction method of multipliers (ADMM) approach were analyzed using the results of numerical experiments. In general, the results indicated that the CADMM approach was more efficient and robust than the ADMM approach.

This research also provided a classification of the multidisciplinary (MDO) methods in the literature in Chapter 2, and an overview of the rational of using augmented

Lagrangian relaxation techniques to solve decomposable engineering system design problems in Chapter 3. In particular, the classification clearly revealed the similarities and differences among the widely applied MDO methods by focusing on three general features, i.e. the introduction of the copies of coupling variables, the relaxation of constraints, and the solution sequence; the overview of the theoretical rational provided the general properties of the proposed coordination methods for solving both the hierarchical and nonhierarchically decomposed problems.

Specific contributions of the research are summarized into four aspects:

1. Developed cutting plane methods based on duality theorem for ATC-decomposed problems, achieving an optimal solution if the system design optimization problem is convex or a lower bound of the objective if the system design optimization problem is nonconvex.

2. Developed bi-objective optimization to capture ATC features and proposed the generic algorithm for two-level problems.

3. Developed a network target coordination (NTC) framework capturing interactions among distributed design teams, while at the same time allowing design autonomy for each team.

4. Demonstrated actual implementations of both the ATC and NTC successfully using four example problems: a mathematical problem and three engineering design problems.

7.2 Furture Research

For cutting plane methods, future work should study the impact of various parameters such as starting points for primal and dual variables, and the subjective parameters used in updating the dual variables. In addition, the effect of some of the issues generally encountered when solving large scale engineering problems, i.e., non-continuous

variables and functions, time consuming evaluations and large number of levels and subsystems, on the performance of the cutting plane methods should be investigated. Finally, a comparison with other existing methods for ATC, such as the penalty method, should be performed.

For biobjective optimization, further research can be continued in several directions. Convergence of the three-subproblem algorithm allowing for communication at the lower level should be established. Other scalarizing approaches, in addition to the weighted-sum method, to solve the ATC-related biobjective optimization problems should be investigated. While they are not expected to make conceptual changes to the biobjective framework, they may offer computational savings. Furthermore, the proposed biobjective framework can be extended for ATC problems with lower-level objective functions, as well as systems with two or more subproblems on each level and systems with three or more levels.

For network target coordination, the following aspects are worthy of further investigations:

1. Efficiency. Further understanding of efficiency issues of the proposed NTC will be beneficial to reduce the computational costs.

2. Applying to practical large-scale problems. Current NTC is still limited to academia version complex systems, however, more complex problems should be investigated in order to verify the efficiency and accuracy of the proposed method.

3. Intelligent consensus estimates. Agent based design methodology in combination with target distribution have been be applied to allow design agents with appropriate intelligence for follow-up design actions with different options. This idea should be further investigated in order to better model the decision making for achieving the consensus of coupled subsystems.

4. Parameter setting investigation. Comprehensive research should be conducted for setting the values of parameters used for updating the penalty weights and dual

variables.

In summary, this research continues to build on recent methods to address decomposed large-scale, complex systems design problems. It addresses some issues and proposes new algorithms, and the success of these approaches will be in the wide spread use of such methods to address current and future engineering design problems.

Bibliography

J. Agte, O. de Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck. Mdo: Assessment and direction for advancement - an opinion of one international group. *Structural and Multidisciplinary Optimization*, 40:17–33, 2010.

N. M. Alexandrov and R. M. Lewis. Comparative properties of collaborative optimization and other approaches to mdo. In *ASMO UK/ISSMO Conference on Engineering Design Optimization, MCB University Press*, pages 39–46, Bradford, UK, 1999.

N. M. Alexandrov and R. M. Lewis. Analytical and computational aspects of collaborative optimization. Technical report, NASA Langley, 2000.

T. Allen, D. Nightingale, and E. Murman. Engineering systems: An enterprise perspective. Engineering Systems Monograph, April 2004.

J. T. Allison, M. Kokkolaras, M. Zawislak, and P. Y. Papalambros. On the use of analytical target cascading and collaborative optimisation for complex system design. In *Proceedings of the 6^{th} World Congress on Structural and Multidisciplinary Optimization*, 2005.

J. T. Allison, M. Kokkolaras, and P. Y. Papalambros. Optimal partitioning and coordination decisions in system design using an evolutionary algorithm. In *Proceedings of the 7th World Congress on Structural and Multidisciplinary Optimization*, Seoul, South Korea, 2007.

K. J. Arrow and R. M. Solow. *Gradient methods for constrained maxima, with weakened assumptions*. Stanford University Press: Stanford, 1958.

K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming*. Stanford University Press: Stanford, 1958.

R. J. Balling and J. Sobieszczanski-Sobieski. Optimization of coupled systems: a critical overview of approaches. *Journal of Aeronautics and Astronautics*, 34(1):6–17, 1996.

B. Barney. Introduction to parallel computing. Workshop of Lawrence Livermore National Laboratory, July 2012.

J. F. M. Barthelemy. Engineering design applications of heuristic multilevel optimization methods. Technical report, National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, 1989.

M. Bazaraa, H. Sherali, and C. Shetty. *Nonlinear Programming: Theory and Algorithms, 3^{rd} Edition*. Wiley, New York, 2006.

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2^{nd} edition, 2003.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Cambridge, MA, 2003.

C. L. Bloebaum. Coupling strenght-based system reduction for complex engineering design. *Structural Optimization*, 10(2):113–121, 1995.

C. L. Bloebaum, P. Hajela, and J. Sobieski. Non-hierarchic system decomposition in structural optimization. *Engineering Optimization*, 19:171–186, 1992.

V. Y. Blouin, H. B. Samuels, G. M. Fadel, I. U. Haque, and J. R. Wagner. Continuously variable transmission design for optimum vehicle performance by analytical target cascading. *International Journal of Heavy Vehicle Systems, Special Issue on Advances in Ground Vehicle Simulation*, 11(2/3):327–348, 2004.

V. Y. Blouin, J. B. Lassiter, M. M. Wiecek, and G. M. Fadel. Augmented Lagrangian coordination for decomposed design problems. In *Proceeding of the 6^{th} World Congress on Structural and Multidisciplinary Optimization*, 2005.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, 2004.

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, 2011.

R. Braun, A. Moore, and I. Kroo. Collaborative approach to launch vehicle design. *Journal of Spacecraft and Rockets*, 34(4):478–486, 1997.

R. D. Braun. *Collaborative optimization: an architecture for large-scale distributed design*. PhD thesis, Standford University, 1996.

L. Chen and S. Li. Analysis of decomposability and complexity for design problems in the context of decomposition. *Journal of Mechanical Design*, 127(4):545–557, 2005.

C. W. Churchman and H. B. Eisenberg. *Chapter 3: Dekuveration and Judgement. In Human Judgements and Optimality.* John Wiley and Sons, Inc., 1969.

E. J. Cramer, P. D. Frank, G. R. Shubin, J. E. Dennis, and R. M. Lewis. On alternative problem formulations for multidisciplinary design optimization. In *4^{th} AIAA/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, Cleveland, OH, 1992.

E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary optimization. *Journal of SIAM on Optimization*, 4(4):754–776, 1994.

O. L. de Weck and M. B. Jones. Isoperformance: Analysis and design of complex systems with desired outcomes. *System Engineering*, 9(1):45–61, 2006.

A. J. de Wit and F. van Keulen. Overview of methods for multilevel and/or multidisciplinary optimization. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2010.

A. V. Demiguel and W. Murray. A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering*, 7(2):99–133, 2006.

A. Devices. *ADXL150/250 Data Sheet*. Analog Devices, 1998.

M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, 2005.

A. V. Fiacco and G. P. McCormick. The sequential unconstrained minimization technique for nonlinear programing, a primal-dual method. *Management Science*, 10(2):360–366, 1964.

R. E. Fulton, J. Sobieszczanski-Sobieski, O. J. Storaasli, E. Landrum, and D. Loendorf. Application of computer-aided aircraft design in a multidisciplinary environment. *Journal of Aircraft*, 11(7):369–370, 1974.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comput. Math. Appli.*, 2(1):17–40, 1976.

M. Gardenghi. *Multiobjective Optimization for Complex Systems*. PhD thesis, Clemson University, 2009.

M. Gardenghi, M. Wiecek, and W. Wang. Biobjective optimization for analytical target cascading: optimality vs. achievability. *Structural and Multidisciplinary Optimization*, pages 1–23, 2012.

R. N. Gasimov. Augmented Lagrangian duality and nondifferentiable optimization methods in non-convex programming. *Journal of Global Optimization*, 24(2):187–203, 2002.

A. M. Geoffrion. Duality in nonlinear programming: A simplified applications oriented development. *Journal of SIAM Review (SIREV)*, 13(1):1–37, 1971.

J. L. Goffin. On convergence rates of subgradient optimization methods. *Mathematical Programming*, 13(1):329–347, 1977.

J. Golinski. Optimal synthesis problems solved by means of nonlinear programming and random methods. *Journal of Mechanisms*, 5(3):287–309, 1970.

M. D. Griffin. How do we fix the systems engineering? In *61st International Astronautical Congress*, 2010.

R. Haftka and L. Watson. Multidisciplinary design optimization with quasi-separable subsystems. *Optimization and Engineering*, 6(1):9–20, 2005.

R. T. Haftka. Simultaneous analysis and design. *Journal of Aeronautics and Astronautics*, 23 (1):1099–1103, 1985.

R. T. Haftka and L. T. Watson. Decomposition theory for multidisciplinary design optimization problems with mixed integer quasi-separable subsystems. *Optimization and Engineering*, 6(1):135–149, 2006.

S. P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.

M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

S. J. Kapurch. Nasa systems engineering handbook, 2007.

S. Karamardian. Generalized complementarity problem. *Journal of Optimization Theory andApplications*, 8:161–168, 1971.

H. Kim, W. Chen, and M. Wiecek. Lagrangian coordination for enhancing the convergence of analytical target cascading. *Journal of Aeronautics and Astronautics*, 44(10):2197–2207, 2006.

H. M. Kim. *Target Cascading in Optimal System Design*. Phd thesis, University of Michigan, Ann Arbor, 2001.

H. M. Kim, M. Kokkolaras, L. S. Louca, G. J. Delagrammatikas, N. F. Michelena, Z. S. Filipi, P. Y. Papalambros, S. J. L., and D. N. Assanis. Target cascading in automotive vehicle design: a class 6 truck study. *International Journal of Vehicle Design*, 29:361–381, 2002.

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.

S. Kodiyalam and J. Sobiesczanski-Sobieski. Bilevel integrated system synthesis with response surfaces. *Journal of AIAA*, 38(8):1479–1485, 2000.

M. Kokkolaras and P. Y. Papalambros. *Structural Design Optimization Considering Uncertainties: Structures & Infrastructures Book*, volume 1 of *1*. Taylor & Francis, March 2008.

A. Lambe and J. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012.

A. B. Lambe and J. R. R. A. Martins. A unified description of MDO architectures. *Proceedings of the 9th World Congress on Structural and Multidisciplinary Optimization*, June 2011.

L. D. Landau and E. M. Lifshits. *Mechanics (Course of Theoretical Physics, Volume 1)*. Pergamon Press, 3rd edition, 1989.

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*, 67(3):361–381, 2005.

Y. Li, Z. Lu, and J. J. Michalek. Diagonal quadratic approximation for parallelization of analytical target cascading. *Journal of Mechanical Design*, 130(5):1–7, 2008. 051402.

S. Lu and H. M. Kim. A regularized inexact penalty decomposition algorithm for multidisciplinary design optimization problems with complementarity constraints. *Journal of Mechanical Design*, 132(4):1–12, 2010. 041005.

N. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1^{st} edition, 1996.

G. Mateos, J. A. Bazerque, and G. B. Giannakis. Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, Oct. 2010.

A. Messac. Physical programming: Effective optimization for computational design. *Journal of Aeronautics and Astronautics*, 34(1):149–158, 1996.

J. J. Michalek and P. Y. Papalambros. An efficient weighting update method to achieve acceptable inconsistency deviation in analytical target cascading. *Journal of Mechanical Design*, 127(3):206–214, 2005a.

J. J. Michalek and P. Y. Papalambros. Technical brief: weights, norms, and notation in analytical target cascading. *Journal of Mechanical Design*, 127(2):499–501, 2005b.

N. Michelena, H. M. Kim, and P. Y. Paralambros. A system partitioning and optimization approach to target cascading. In *Proceedings of the 12th International Conference on Engineering Design*, Munich, Germany, 1999.

N. Michelena, H. Park, and P. Y. Papalambros. Convergence properties of analytical target cascading. *Journal of Aeronautics and Astronautics*, 41(5):897–905, 2003.

A. Miele, E. E. Cragg, R. R. Iver, and A. V. Levy. Use of the augmented penalty function in mathematical programming problems, part 1. *Journal of Optimization Theory and Applications*, 8(2):115–130, 1971a.

A. Miele, E. E. Cragg, and A. V. Levy. Use of the augmented penalty function in mathematical programming problems, part 2. *Journal of Optimization Theory and Applications*, 8(2):131–153, 1971b.

A. Miele, P. E. Moseley, A. V. Levy, and G. M. Coggins. On the method of multipliers for mathematical programming problems. *Journal of Optimization Theory and Applications*, 10(1):1–33, 1972.

I. S. F. G. K. Mukherjee, T. Optimization-based synthesis of microresonators. *Sensors and Actuators A: Physical*, 70(1-2):118–127, 1998.

A. Nedić and A. E. Ozdaglar. *Cooperative distributed multi-agent optimization in Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

J. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

J. F. Nunamaker, L. M. Applegate, and B. R. Konsynski. Computer-aided deliberation: model management and group decision support. *Journal of the Operations Research*, 36 (6):826–848, 1988.

R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215 –233, jan. 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.887293.

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):303–307, 1964.

M. J. D. Powell. *A Method for Nonlinear Constraints in Minimization Problems*. Optimization ed. by R. Fletcher, Academic Press, New York, NY, 1969.

M. J. D. Powell. *A fast algorithm for nonlinearly constrained optimization calculations," in G.A. Watson (ed.), Numerical Analysis, Lecture Notes in Mathematics*, volume 630. Springer, Berlin, 1978.

B. Prasad. *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*. Prentice Hall, 1996.

J. Rosen. The gradient projection method for nonlinear programming: II. non-linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

B. Roth and I. Kroo. Enhanced collaborative optimization: application to an analytic test problem and aircraft design. In *Proceedings of ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Brooklyn, New York, August 3-6 2008.

A. Ruszczynski. On convergence of an augmented lagrangian decomposition for sparse convex optimization. *Journal of Mathematics of Operations Research*, 20(3):634–656, 1995.

H. Samuels. Single- and dual-axis micromachined accelerometers adxl150 & adxl250: New complete low-noise 50-g accelerometers. *Analog Dialogue*, 30(40), 1996.

K. Schittkowski. On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. *Journal of Mathematische Operationsforschung und Statistik*, 14:197–216, 1983.

I. D. Schizas, A. Ribeiro, and G. B. Giannakis. Consensus in ad hoc wsns with noisy links-part i: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56:350–364, Jan 2008.

S. F. Senturia. *Microsystem Design*. Springer, 2001.

T. W. Simpson and J. R. R. A. Martins. Multidisciplinary design optimization for complex engineered systems: Report from a national science foundation workshop. *Journal of Mechanical Design*, 133(10):101002, 2011.

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*, 1988.

J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural and Multidisciplinary Optimization*, 14(1): 1–23, 1997.

J. Sobieszczanski-Sobieski, J. S. Agte, and R. R. Sandusky, Jr. Bi-level integrated system synthesis. *Journal of Aeronautics and Astronautics*, 38(1):164–172, 2000.

J. Sobieszczanski-Sobieski, T. D. Altus, M. Phillips, and R. Sandusky. Bi-level integrated system synthesis for concurrent and distributed processing. *Journal of Aeronautics and Astronautics*, 41(10):1996–2003, 2003.

G. Stephanopoulos and A. Westerberg. The use of hestenes method of multipliers to resolve dual gaps in engineering system optimization. *Journal of Optimization Theory and Applications*, 15(3):285–309, 1975.

S. Tosserams. *Distributed Optimization for Systems Design: an augmented Lagrangian coordination method*. Phd thesis, Eindhoven University of Technology, 2008.

S. Tosserams, L. Etman, P. Papalambros, and J. Rooda. An augmented Lagrangian relaxation for analytical target cascading using the alternating directions method of multipliers. *Structural and Multidisciplinary Optimization*, 31(3):176–189, 2006a.

S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda. An augmented Lagrangian relaxation for analytical target cascading using the alternating directions method of multipliers. *International Journal for Numerical Methods in Engineering*, 31 (3):176–189, 2006b.

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.

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, 2008a.

S. Tosserams, L. F. P. Etman, and J. E. Rooda. Augmented Lagrangian coordination for distributed optimal design in MDO. *International Journal of Numerical Methods in Engineering*, 73(13):1885–1910, 2008b.

S. Tosserams, L. Etman, and J. Rooda. A micro-accelerometer MDO benchmark problem. *Structural and Multidisciplinary Optimization*, 41(2):255–275, 2010.

J. N. Tsitsiklis. A comparison of jacobi and gauss-seidel parallel iterations. applied. *IEEE Transactions on Automatic Control*, 2:325–332, 1989.

N. Tzevelekos, M. Kokkolaras, P. Y. Papalambros, M. F. Hulshof, L. E. P. Etman, and J. E. 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*, Lido di Jesolo, Venice, Italy, 2003.

K. T. Ulrich and S. D. Eppinger. *Product design and development*. Boston: Irwin/McGraw-Hill, 3rd edition, 2007.

G. N. Vanderplaats. Automated optimization techniques for aircraft synthesis. In *Proceedings of AIAA Aircraft Systems and Technology Meeting*, number 76-909, Dallas, TX, 1976.

J. R. Wagner and P. Y. Papalambros. Implementation of decomposition analysis in optimal design. In *Proceedings of the 19th Annual ASME Design Automation Conference*, pages 327–335, 1993.

T. C. Wagner. *A General Decomposition Methodology for Optimal System Design*. Phd thesis, University of Michigan, Ann Arbor, Michigan, 1993.

W. Wang, V. Blouin, M. Gardenghi, M. M. Wiecek, G. M. Fadel, and B. Sloop. A cutting plane method for analytical target cascading with augmented Lagrangian coordination. In *proceedings of the ASME 2010 Conference IDETC/CIE*, Montreal, Canada, August 15-18 2010.

W. Wang, P. Guarneri, G. Fadel, and V. Y. Blouin. Network target coordination for optimal design of decomposed systems with consensus optimization. In *procedding of the ASME 2012 Conference IDETC/CIE*, 2012.

B. A. Wujek, J. E. Renaud, S. M. Batill, and B. J. B. Concurrent subspace optimization using design variable sharing in a distributed computing environment. *Concurrent Engineering*, 4:361–377, 1996.

Y. Zhou. Layout synthesis of accelerometers. Master's thesis, Carnegie Mellon University, 1998.

H. Zhu, A. Cano, and G. B. Giannakis. Distributed consensus-based demodulation: algorithms and error analysis. *IEEE Transactions on Wireless Communications*, 9(6):2044–2054, 2010.

G. Zoutendijk. *Methods of Feasible Directions*. Elsevier, New York, 1960.