DEVELOPMENT OF MULTIDISCIPLINARY DESIGN OPTIMIZATION ALGORITHMS FOR SHIP DESIGN UNDER UNCERTAINTY

by

Shari E. Hannapel

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in The University of Michigan
2012

Doctoral Committee:

Professor Nickolas Vlahopoulos, Chair
Professor Romesh Saigal
Assistant Professor Matthew Collette
Assistant Professor David J. Singer

To my parents.

ACKNOWLEDGEMENTS

I am grateful for funding from a Science, Mathematics, and Research for Transformation (SMART) Scholarship. This research has also supported in part by ONR grant number 008843, TPOC Kelly Cooper.

CFD work and Kriging surrogate model coding was done by Jim He.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

CHAPTER 1

Introduction

Possibly the simplest definition of a design is as a system (Papalambros and Wilde 2000), where a system is composed of multiple interconnected components which interact and must function harmoniously. The designer has the ability to prescribe values for some of the properties of the system, which in turn affect the behavior of the system.

In simple cases, it may be possible for a designer to discern the relationships between the parameters he or she prescribes and the performance of the system; then the designer can make well-informed decisions about the design. However, if the system is complex or if the designer utilizes advanced analysis tools, it is not possible to readily predict the behavior of the design. Therefore, it is necessary to establish a design methodology so that the design process is approached rationally and effectively.

This dissertation is focused on the design of complex systems, and the dissertation presents two new algorithms in the field of multidisciplinary design optimization, including their application in naval architecture. This chapter introduces two concepts in design: design optimization and multidisciplinary design optimization (MDO).

1.1 Design Optimization

Sometimes, a designer may be satisfied with a design that is simply adequate.

However, it is often desirable to improve the design as much as possible. This is not simply a desire for technical improvement (seeking the greatest performance), but also a matter of practicality (for example, making the design as inexpensive as possible). Then design optimization can be defined as the process of improving a design in a rational, systematic way.

In order to improve a design, the designer must answer the very practical question of what makes a design "better" or "best," which may not be an easy question to answer (Papalambros and Wilde 2000). The challenges of selecting appropriate metrics for the design are not discussed here, and it is assumed that for any given design problem, quantifiable design metrics are readily available. Given a measure of performance metric, called the objective function, the goal is to improve the objective function as much as possible. Additionally, the problem may be subject to requirements that must be satisfied, called constraints. Then the general design optimization problem statement is to improve the objective function as much as possible while satisfying the constraints. For example, a structural design problem could be to minimize the weight of a beam while maintaining a bending stress less than a specified value.

The field of optimization is extensive with a long history in both engineering and mathematics. It is not the intention of this dissertation to cover the topic of optimization in detail; instead, the goal of this dissertation is to provide an introduction to optimization techniques as an essential component of MDO, before presenting the new MDO algorithms.

1.2 Multidisciplinary Design Optimization

It is generally not possible to approach the design of a complex system as the solution of a single design problem; instead, the complex design problem can be broken down into subproblems of manageable size. For engineering design problems, a complex system is often divided into disciplines, which are specific areas or subsystems of the design, such as structural design or propulsion in ship design.

One of the earliest reviews on MDO by Sobieszczanski-Sobieski (1989) explains that breaking a complex design into disciplines is intuitive; it is natural to separate a difficult task into parts or divide the work amongst teams of specialists. The traditional approach for handling a complex design is to use a sequential process: proceed linearly from one analysis to the next, iterating as needed for convergence. In the context of ship design, the sequential design process is often referred to as the design spiral (Watson 1998). The sequential design process is illustrated in Figure 1.1.

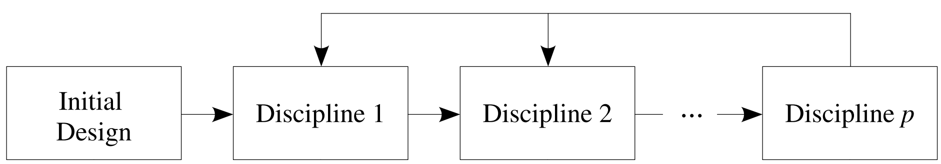


Fig. 1.1. Illustration of the sequential design process.

The simplest approach for including optimization in the design is to add an outer optimization loop to the sequential design process. However, this is often inefficient and, most importantly, does not adequately capture the interaction between disciplines (Sobieszczanski-Sobieski 1989). These deficiencies in the sequential design optimization process motivate the development of MDO methods.

The critical task of MDO methods is to develop a way to coordinate the interactions between disciplines. The development of different methods for coordinating the disciplinary interactions has led to the development of the wide array of MDO algorithms. One example of MDO methods that contrast the sequential design method is the popular class of hierarchical MDO methods. In hierarchical MDO methods, different disciplines are arranged in a hierarchy as illustrated in Figure 1.2; in this way different disciplines communicate indirectly by exchanging information up and down the hierarchy (Sobieszczanski-Sobieski 1989).

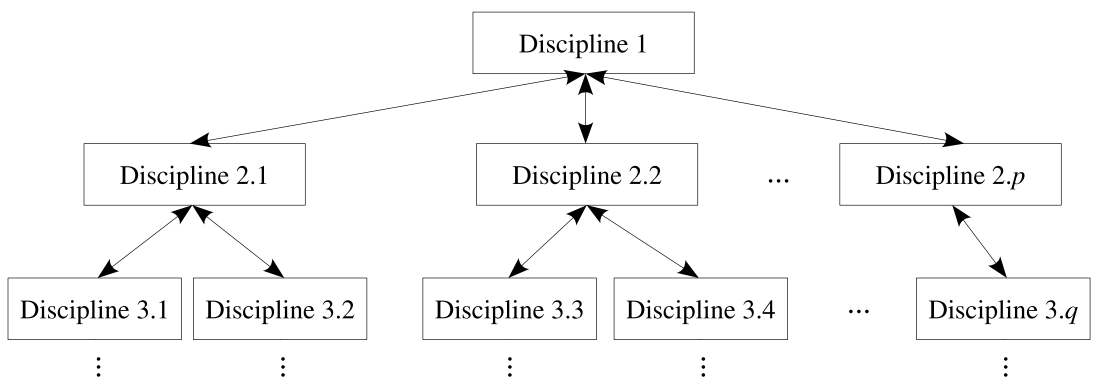


Fig. 1.2. Illustration of a hierarchical MDO method.

In Figure 1.2, the disciplines are labeled *x.y*, where *x* is the level and *y* is the discipline index on level *x*. Use of only the first two levels represents a design that is simply divided into *p* disciplines. Additional levels are useful when analysis calls for further breakdown of the design; for example, if Discipline 2.1 is hull form optimization, Discipline 3.1 could correspond to the bow area and Discipline 3.2 could correspond to the stern area.

MDO methods are valuable for ship design because the design of a ship is normally broken down into disciplines; for example, hull form, structural design, propulsion, or seakeeping performance. Furthermore, the interactions between the different disciplines are complex, particularly when advanced tools are used to evaluate the disciplines, such as computational fluid dynamics for hull form design or finite element analysis for structural design.

1.3 Dissertation Contributions and Overview

There are two phases to this research which correspond to the development of two distinct MDO algorithms. First is the development of a new multilevel MDO algorithm which computes target values for the design variables and uses the target values to drive the process at the system level (MDO with target values). Second is the development of a new MDO algorithm inspired by the principles of set based deign. Research was conducted for both algorithms to demonstrate the value of the algorithms in ship design.

The first contribution of this dissertation is the development of the new multilevel MDO with target values algorithm. The new MDO algorithm seeks to offer an improvement in efficiency over other multilevel MDO methods with target values. Further, the new MDO algorithm provides the capability to give more influence to one discipline which is considered to be most important; this is useful in practical applications because one discipline is often more significant (such as cost).

This dissertation demonstrates that the new multilevel MDO method is capable of handling optimization under uncertainty. Techniques for accounting for uncertainty in optimization (in terms of both reliability and robustness) were implemented with the multilevel MDO algorithm. Additionally, this dissertation shows that the new multilevel

MDO method is capable of utilizing surrogate models in place of expensive solvers to achieve computational savings.

The final contribution of this dissertation is the development of a new MDO algorithm inspired by set-based design. This algorithm employs an optimization strategy that incorporates the principles of set-based design, in particular, rather than pursuing a single point design, the algorithm manages sets (ranges) of the design variables. The size of the sets are gradually reduced during the optimization to yield a reduced design space; by seeking a reduced design space (instead of a single point) the algorithm can respond flexibly to changes encountered as the design evolves and requirements change.

This dissertation is organized as follows. Chapter 2 provides a literature review and mathematical background for multidisciplinary design optimization. The new multilevel MDO algorithm with target values is presented in Chapter 3. In Chapter 4, methods for accounting for uncertainty in optimization are implemented in the new multilevel MDO algorithm, and a ship design analysis with uncertainty is conducted. In Chapter 5, the new multilevel MDO algorithm is used with surrogate models for a ship hull form optimization problem and significant computational time savings are achieved. The new MDO algorithm inspired by the principles of set-based design is presented in Chapter 6. In Chapter 7, the set-based MDO algorithm is applied to a ship design problem. Chapter 8 includes conclusions and recommendations for future work.

Because this dissertation includes elements from many areas of design optimization, the literature review for each topic is included with its corresponding chapter.

CHAPTER 2

Multidisciplinary Design Optimization Background

In general, optimization is the improvement of a particular performance metric under prescribed conditions. Design optimization can be viewed as the application of mathematical optimization techniques to solve design problems.

In real world design problems, there is often more than one area of interest in the design; that is, there may be different disciplines that contribute to the design, such as structures, economics, or hydrodynamics. The disciplines may be studied with different software tools, or the disciplines may be investigated by different teams of engineers. Furthermore, different disciplines are often competing in their objectives; for example, a design with high structural strength is likely to be more expensive than a design with less structural strength. Due to these complexities, the design problem cannot be formulated simply as a single optimization statement.

Therefore, it is necessary to develop methods to address multidisciplinary design problems because they occur in real engineering designs. The goal of multidisciplinary design optimization (MDO) is to develop a method to coordinate different discipline optimizations, and achieve a design that optimizes all disciplines.

This chapter begins with the introduction of optimization terminology, and a brief discussion of the optimization techniques as they relate to MATLAB which is used for programming in this research. History of the development of the field of multidisciplinary design optimization is presented, and a review of multidisciplinary design optimization techniques from the literature follows. Finally, a brief comparison of multidisciplinary design optimization to the related field of multiobjective optimization is presented.

2.1 Optimization Terminology and Techniques

Multidisciplinary design optimization depends on the utilization of optimization techniques; familiarity with the fundamentals of optimization is necessary to understand multidisciplinary design optimization algorithms. Optimization is a vast field with a long history in mathematics and engineering. There are far too many techniques to discuss here, so this section will provide a brief introduction to the areas of interest, and in particular, the techniques included in the MATLAB software which is used in this work. Before discussing the mathematical background on optimization, it is important to define the consistent terminology and notation used in this research.

2.1.1 Optimization Terminology and Notation

Design variables. The design variables are the variables to which the designer can assign values; the values of the design variables are used to determine the state of the design. The design variables are contained in the vector $\mathbf{x} \in \Re^n$, that is, the problem has n scalar design variables. Each design variable x_i has an allowable range between a lower bound and an upper bound; the set of allowable values for $\mathbf{x}$ is defined as χ.

Parameters. Parameters are variables that are not under the control of the designer. Values of parameters may be determined by external factors, for example, environmental parameters and mechanical properties. the vector of parameters for an optimization problem is denoted **p**.

Objective function. The objective function is the function of interest that the designer seeks to optimize. The objective function is denoted $f(\mathbf{x}, \mathbf{p})$ or simply $f(\mathbf{x})$ when there are no parameters in the problem or **p** is constant.

Constraints. The optimization problem may include one or more constraints, which are functions of the design variables and possibly the parameters. In general, a scalar constraint can be written as a function with a value less than or equal to zero: $g(\mathbf{x}) \leq 0$; this is known as negative null form (Papalambros and Wilde 2000). The negative null form for constraints is used throughout this work. In the case of multiple scalar constraints, $\mathbf{g}(\mathbf{x})$ is a vector of length m that contains all of the constraints $g_i(\mathbf{x})$, $i = 1, \ldots, m$.

Finally, the constrained optimization problem can be formulated as:

$$\min_{\mathbf{x} \in \chi} f(\mathbf{x}) \qquad (2.1)$$

$$\text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

Equation (2.1) defines the general constrained optimization problem. This statement requires minimization of the objective function; for cases where the goal is to maximize objective function, the following equivalence can be utilized:

$$\max f(\mathbf{x}) = -\min\left(-f(\mathbf{x})\right) \qquad (2.2)$$

Furthermore, the negative null form for the constraints assumes that the problem contains only inequality constraints, because any equality constraints can be rewritten as inequality using slack variables (Papalambros and Wilde 2000).

The solution to the optimization problem in Equation (2.1) is indicated using the superscript " * ". The value of the function f at its minimum is f^* and the corresponding values of the design variables at the minimum is $\mathbf{x}^*$.

The notation for optimization presented in this section is summarized in Table 2.1. The notation maintains the convention that scalar values are italicized while vectors and matrices are written in boldface.

Table 2.1. Summary of notation for optimization.

n	number of scalar design variables
x_i	scalar design variable with index i, $i = 1, \ldots, n$
$\mathbf{x}$	vector of design variables, $\mathbf{x} \in \Re^n$
χ	set of allowable values for the design variables, $\mathbf{x} \in \chi \subset \Re^n$
$\mathbf{x}^*$	vector of design variables at the optimum point
$\mathbf{p}$	vector of parameters
m	number of constraints
g_i	(scalar) constraint with index i, $i = 1, \ldots, m$ in negative null form
$\mathbf{g}$	vector of constraints, $\mathbf{g} \in \Re^m$
f	(scalar) objective function
f^*	objective function optimum

2.1.2 Classical Constrained Optimization

The classical approach to solving constraint optimization problems analytically is the method of Lagrange multipliers. This section summarizes the discussion presented in Chapter 2 of Rao (2009) on the method of Lagrange multipliers and solution with the Karush-Kuhn-Tucker conditions (Kuhn and Tucker 1951).

The optimization problem with inequality constraints in Equation (2.1) can be rewritten with equality constraints:

$$\min_{\mathbf{x}\in\chi} f(\mathbf{x}) \tag{2.3}$$

$$\text{subject to } \; g_j(\mathbf{x}) + y_j^2 = 0 \quad j = 1, ..., m$$

where y_j are slack variables, held in the vector $\mathbf{y}$.

Then construct the Lagrange function L, for the optimization problem in Equation (2.3):

$$L(\mathbf{x}, \mathbf{y}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_i \left(g_j(\mathbf{x}) + y_j^2 \right) \tag{2.4}$$

The Lagrange function is a function of the variables $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The necessary conditions for a local minimum are found by differentiating the Lagrange function with respect to each variable:

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i}(\mathbf{x}) + \sum_{j=1}^{m} \lambda_i \frac{\partial g_j}{\partial x_i}(\mathbf{x}) = 0 \qquad i = 1, \ldots, n \tag{2.5}$$

$$\frac{\partial L}{\partial \lambda_j} = g_j(\mathbf{x}) + y_j^2 = 0 \qquad j = 1, \ldots, m \tag{2.6}$$

$$\frac{\partial L}{\partial y_j} = 2 \lambda_j y_j = 0 \qquad j = 1, \ldots, m \tag{2.7}$$

Equation (2.6) is simply the statement that all constraints must be satisfied. Equation (2.7) implies that either $\lambda_j = 0$ or $y_j = 0$; when $\lambda_j = 0$, the constraint g_j is inactive, and when $y_j = 0$, the constraint g_j is active. Furthermore, λ_j must be nonnegative for minimization problems with constraints in negative null form.

The Karush-Kuhn Tucker (KKT) conditions are the necessary and sufficient first-order conditions for an optimum. The KKT conditions are equivalent to the statements in Equations (2.5)-(2.7) evaluated at the optimum ($\mathbf{x}^*$, $\boldsymbol{\lambda}^*$):

$$\frac{\partial f}{\partial x_i}(\mathbf{x}^*)+\sum_{j=1}^{m}\lambda_j^*\frac{\partial g_j}{\partial x_i}(\mathbf{x}^*)=0 \qquad i = 1, \ldots, n \tag{2.8}$$

$$\lambda_j^* g_j(\mathbf{x}^*) = 0 \quad j = 1, \ldots, m \tag{2.9}$$

$$\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0} \tag{2.10}$$

$$\boldsymbol{\lambda}^* \geq \mathbf{0} \tag{2.11}$$

Equations (2.8) and (2.9) provide a system of $m + n$ equations with $m + n$ unknowns. For simple problems, it is possible to find the partial derivatives analytically and then solve the system of equations. However, for practical problems the derivatives of the objective function and constraints may be complex or unavailable, and the system of equations could become very large and nonlinear. While the KKT conditions can still provide insight into a complex problem, it is generally not possible to solve Equations (2.8)-(2.11) analytically; instead, numerical optimization methods must be used.

2.1.3 Numerical Constrained Optimization Techniques

Numerical optimization methods are useful for problems that are too difficult to solve analytically, which includes most problems of practical interest. Instead of solving the problem directly, numerical optimization methods operate in an iterative procedure. The fundamental idea behind iterative methods is to begin at a point, and look in the neighborhood of the point for a slightly better point. From the second point, look for another slightly better point, and so on, to iteratively improve the objective functions. This section summarizes some of the content presented in Papalambros and Wilde's (2000) summary of numerical optimization methods.

One approach for handling the complex constrained optimization problem is to solve a simpler approximation for the problem, and sequential quadratic programming (SQP) is an example of such an approach. SQP became popular in the 1970s, and the algorithm continues to be widely used for nonlinear optimization problems because of its efficiency and utility.

The general quadratic programming problem takes the form (Onwubiko 2000)

$$\min \ \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \tag{2.12}$$

$$\text{subject to } \ \mathbf{A}\mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$\mathbf{-x} \leq \mathbf{0}$$

where $\mathbf{Q}$ is a matrix of size $n \times n$, $\mathbf{c}$ is a vector of length n, $\mathbf{b}$ is a vector of length m, and $\mathbf{A}$ is a matrix of size $m \times n$. The quadratic programming program gets its name because the objective function is quadratic in $\mathbf{x}$. The second set of constraints $-\mathbf{x} \leq \mathbf{0}$ simply requires the design variables to be nonnegative (rewritten in negative null form). Furthermore, it is required that the matrix $\mathbf{Q}$ be positive definite.

The KKT conditions can be applied to Equation (2.12) and differentiation of the objective function is necessary for Equation (2.8) of the KKT conditions. Differentiation of the objective function with respect to each design variable (the gradient) yields a linear function:

$$\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}+\mathbf{c}^{\mathrm{T}}\mathbf{x}\right)=2\,\mathbf{Q}\mathbf{x}+\mathbf{c} \tag{2.13}$$

Then all of the KKT conditions for the quadratic programming problem are linear, forming a linear programming program which, as presented earlier, is easy to solve.

The fundamental idea of SQP is to iteratively solve a quadratic approximation for the KKT conditions. The corresponding quadratic programming subproblem is

(Schittkowski and Yuan 2011)

$$\min_{\mathbf{d}} \nabla f(\mathbf{x}_k)^{\mathrm{T}} \mathbf{d} + \frac{1}{2} \mathbf{d}^{\mathrm{T}} \mathbf{H}_k \mathbf{d} \tag{2.14}$$

$$\text{s.t.} \quad g_j(\mathbf{x}_k) + \nabla g_j(\mathbf{x}_k)^{\mathrm{T}} \mathbf{d} \le 0 \quad j = 1, \ldots, m$$

where $\mathbf{x}_k$ is the approximation of the solution (to the optimization problem) at iteration k. $\mathbf{H}_k$ is the Hessian (second derivative matrix) for the Lagrange function at $\mathbf{x}_k$. The solution of the quadratic programming problem in Equation (2.14) is the vector **d** (length n).

The solution **d** of the quadratic programming problem is a vector which identifies the direction used to update the approximation of $\mathbf{x}^*$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d} \tag{2.15}$$

Equation (2.15) can also be modified to include a line search (Schittkowski and Yuan 2011)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d} \tag{2.16}$$

The purpose of the line search is to adjust the step length parameter α_k to minimize the objective function in the direction **d**.

Finally, at each iteration, the Hessian of the Lagrange function $\mathbf{H}_k$ is updated. Explicit evaluation of the Hessian is usually not possible, so $\mathbf{H}_k$ is typically replaced with an approximation for the Hessian, $\mathbf{B}_k$. The purpose of using an approximation for the Hessian is to save computational time (because evaluating the Hessian with finite differences is computationally expensive), and to ensure that the Hessian remains positive-definite. Many formulas for calculating the approximation $\mathbf{B}_k$ are available in the literature.

There are many more optimization methods available in the literature (see Rao

(2009) for a thorough introduction to a variety of methods). Only two additional methods are discussed here because they are encountered in some form in the optimization software used in this research: transformation methods and active set strategies.

Unconstrained optimization problems are generally easier to solve than constrained optimization, and many methods for unconstrained optimization are available in the literature. Transformation methods seek to transform the constrained optimization problem in Equation (2.1) to the following unconstrained optimization problem :

$$\min_{\mathbf{x}\in\chi} f(\mathbf{x})+\Phi(\mathbf{g}(\mathbf{x}),r) \tag{2.17}$$

where Φ is a scalar function, referred to as either a penalty or barrier function depending on its form, and r is a weighting parameter.

Penalty functions (also called exterior penalty functions) penalize the objective function when a constraint is violated. Penalty functions are constructed so that a constraint violation contributes a positive term; one common formulation is

$$\Phi(\mathbf{g}(\mathbf{x}),r)=r^{-1}\sum_{j=1}^{m}\left[\max(0,g_j(\mathbf{x}))\right]^2 \tag{2.18}$$

Penalty function methods are preferred for problems with only equality constraints, because they can generate infeasible solutions for problems with inequality constraints.

Barrier functions (also called interior penalty functions) prevent the design from crossing the constraint boundaries. Barrier functions are constructed so that the objective function approaches infinity when the design approaches a constraint; one common formulation is:

$$\Phi(\mathbf{g}(\mathbf{x}),r)=-r\sum_{j=1}^{m}\ln(-g_j(\mathbf{x})) \tag{2.19}$$

Barrier functions are generally preferred over penalty functions because they ensure that the solution will be feasible; however, they can only be used for inequality constraints.

Optimization problems with inequality constraints can be challenging because it is not known in advance which constraints will be active at the optimum. The purpose of an active set strategy is to methodically study the effect of activating or deactivating constraints during the optimization. The constraints can be sorted into three sets: the active set (constraints that are active), the working set (constraints that are active at the current iteration), and the candidate set (constraints that can be selected to join the working set).

The process of optimization with an active set strategy begins from an initial feasible point and an initial working set. The objective function is minimized assuming that the constraints in the working set are active (that is, they are now equality constraints). Another constraint is added from the candidate set to the working set, and the optimization is repeated to see if the solution improves. Constraints are added to the working set with optimization until the objective function no longer improves.

The KKT condition from Equation (2.8) evaluated at a point $\mathbf{x}_k$ is

$$\nabla f(\mathbf{x}_k) + \sum_j \lambda_j \nabla g_j(\mathbf{x}_k) = \mathbf{0} \tag{2.20}$$

where j is the index of the active constraints. Equation (2.20) can be used to evaluate λ_j at $\mathbf{x}_k$; if all λ_j are nonnegative, $\mathbf{x}_k$ is a KKT point and the algorithm has reached an optimum. If one or more λ_j are negative, further improvement can be achieved by deactivating the constraints. Constraints are removed from the working set one at a time, each time performing an optimization with the remaining active constraints to see if the objective function improves. The process of adding and removing constraints is repeated until the KKT conditions are satisfied.

2.1.4 Optimization in MATLAB

MATLAB software includes powerful capabilities for optimization. The MATLAB function `fmincon` is used for constrained optimization problems in the form of Equation (2.1), and the function is capable of handling both linear and nonlinear constraints. MATLAB's `fmincon` is used throughout this research for solving constrained optimization problems.

`fmincon` uses an SQP algorithm, which as described earlier, seeks to solve the KKT conditions instead of solving the optimization problem directly. The quadratic programming problem used in the SQP algorithm takes the form of Equation (2.14). Because the quadratic programming problem includes inequality constraints, MATLAB's `fmincon` includes an option to utilize an active set strategy to solve the quadratic programming problem. Additionally, in the SQP algorithm, the approximation for the Hessian $\mathbf{B}_k$ is updated at each iteration using the BFGS formula (Schittkowski and Yuan 2011).

The solution of the quadratic programming subproblem returns a search direction $\mathbf{d}$ which is used to update the values of the design variables using a line search as in Equation (2.16). Rather than just minimizing the objective function in the line search, `fmincon` utilizes a penalty function to ensure that the design remains feasible when moving in the direction $\mathbf{d}$. The penalty function resembles that of Equation (2.18); the function minimized during the line search is

$$f(\mathbf{x})+\sum_{j=1}^{m} r_j \max[0, g_j(\mathbf{x})] \tag{2.21}$$

where r_j is a weighting parameter.

2.2 History of Multidisciplinary Design Optimization

It is natural and practical to break down a complex design problem into smaller subproblems (Sobieszczanski-Sobieski 1989), and the practice of dividing complex engineering problems into disciplines is common in many fields. The division of the design into disciplines could be a physical separation, where the disciplines are assigned to different teams of specialists. The division into disciplines could also represent different computer programs that address different aspects of the design. With increases in complexity and advances in technology, modern designs may utilize many design teams spread across the country (or globe) and many different specialized high-fidelity software programs.

While the division of a complex problems into disciplines describes a multidisciplinary design, multidisciplinary design optimization (MDO) requires more: the application of *optimization* to the design, and a systematic *coordination* of the exchange of information between disciplines. Cramer et al. (1994) succinctly define MDO as "... the coupling of two or more analysis disciplines with numerical optimization." The coupling is the critical element that has motivated development in the field of MDO because it necessitates an efficient, logical approach to coordinate the influences of each discipline on the others. In general, a single, independent discipline could be optimized easily; the challenge of MDO is to apply optimization to all disciplines when the disciplines influence each other and the design as a whole.

The purpose of an MDO method is to convert the complex multidisciplinary design problem into a numerical formulation that is solvable using the available optimization techniques (Tedford and Martins 2010). Zhao and Cui (2011) explain that MDO methods developed over three phases. In the first phase, disciplinary analyses were connected in a single large optimization statement; the main disadvantage of these methods is that they could not handle large, complex problems. In the second phase, bi-level methods were introduced; these methods separated the system level optimization statement from discipline optimization statements, and were more capable of managing the coupling between disciplines. In the third phase, techniques for decomposition developed, which allow the MDO problem to be broken down into independent and coordinated optimization problems (Agte et al. 2010).

MDO is generally computationally expensive, because MDO problems typically include a large number of design variables collected from all disciplines, with multiple discipline analyses that are called repeatedly within the overall optimization. Early MDO was limited by the computer technology available; completing a design in a reasonable amount of time required a compromise between the number of design variables, the number of disciplines, and the fidelity of the analysis models (Agte et al. 2010). Today's computer technology enables the use of high-fidelity analysis within large MDO problems, a point which is essential to this work.

MDO originally developed out of the field of structural optimization, specifically with application in aerospace engineering (Agte et al. 2010). One of the most common applications of MDO was (and continues to be) aircraft design with the two disciplines of structures and aerodynamics (Sobieszczanski-Sobieski and Haftka 1997). MDO use in aerospace engineering expanded to include other disciplines, such as performance and propulsion, and eventually included the entire aircraft (Agte et al. 2010).

Today, MDO is valuable in many engineering fields. MDO continues to be popular in advanced aerospace design problems at both the total aircraft level and at the detailed component level. For example, Leifsson et al. (2011) studied an MDO problem with disciplines of aerodynamics, propulsion, weight analysis, aircraft performance, and stability and control. Jouhaud et al. (2007) optimized a two-dimensional wing in turbulent flow with aerodynamics and aeroacoustics analysis using expensive solvers (computational fluid dynamics).

Applications in automotive engineering are also common. Park, da Luz, and Suleman (2008) optimized an electromechanical brake system with disciplines for weight and torque; their analysis includes utilization of expensive solvers (finite element analysis). Ferguson, Kasprzak, and Lewis (2009) designed a family of vehicles (race cars) with disciplines of aerodynamics, handling, and chassis design. In mechanical engineering, Grujicic et al. (2010) optimized a wind turbine blade made of composite material with cost, aerodynamics, and structures analyses. In civil engineering, Geyer (2009) studied the design of a large hall with disciplines for economic performance, environmental impact, and preference.

Applications of MDO in naval architecture are of the most interest in this dissertation. As with aerospace engineering, many naval architecture MDO problems are concerned with structural design and fluid dynamics, such as Diez et al. (2012) where MDO is applied to the design of a keel fin of a sailing yacht with structural and fluid dynamics disciplines. Kalavalapally, Penmetsa, and Grandhi (2006) studied a torpedo design with fluid dynamics (specifically with underwater explosions) and structures disciplines. Besnard et al. (2007) optimized a fast ship hull form using fluid dynamics and structures disciplines.

MDO has also been applied on a larger scale in ship design. Yang et al. (2007) used MDO for preliminary ship design with hull form, powering, and cost analyses. Hefazi et al. (2010) developed an MDO tool for the design of multihull ship, including hull form, powering, and weight analyses.

2.3 Review of Multidisciplinary Design Optimization Techniques

A general MDO problem definition is necessary before presenting MDO techniques. The following form of an MDO problem is common in the literature and it assumes that the MDO problem is described by a single objective function *f*, but with coupling variables which must be evaluated in the discipline analyses. The general MDO problem is

$$\min_{\mathbf{x},\,\mathbf{z}} f\left(\mathbf{x}, \mathbf{z}, \mathbf{y}\left(\mathbf{x}, \mathbf{z}\right)\right) \tag{2.22}$$

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

where $\mathbf{x}$ is a vector of local design variables (design variables that affect only one discipline) and $\mathbf{z}$ is a vector of global design variables. $\mathbf{y}$ is a vector of discipline coupling variables which are determined from the discipline analyses (therefore a function of $\mathbf{x}$ and $\mathbf{z}$). $\mathbf{g}$ is the vector of constraints, which could be divided into local and global constraints.

This definition may seem abstract, so consider an application in ship design. The

MDO problem includes two disciplines: hull resistance (fluid dynamics) and structural design. The design variables **x** and **z** describe the shape of the hull and the properties of the hull structure. The objective function f is the hull resistance (to be minimized). An example of a coupling variable **y** is the structural weight: the structural design discipline determines the structural weight which affects the displacement of the ship. The displacement of the ship then affects the hull resistance, so in order to evaluate the hull resistance for a specific design, the structural weight (coupling variable) information must be passed from the structures discipline to the hull resistance discipline.

The remainder of this section presents popular MDO techniques. In general, each method proposes a way to transform the complex MDO problem into a solvable optimization statement. Regardless of the approach used, optimization is necessary to solve the problem, and the optimization problem could be solved with the methods presented earlier in the chapter or with other methods. The purpose of this section is only to present the structure of each of the MDO methods, so the presentation of many methods has been greatly simplified; details of the exact implementation of each method should be sought in the accompanying references.

It is helpful to approach the MDO methods with classification of different approaches. While many possible ways of classifying MDO methods exist (de Wit and van Keulen 2010), the most common ways of classifying MDO methods is by their structure and the number of levels. An MDO method's structure may be either hierarchic or nonhierarchic, as illustrated in Figure 2.1. In a hierarchic structure, disciplines cannot communicate directly with each other; instead, they can only communicate vertically with analyses above (or below) them (Balling and Sobieszczanski-Sobieski 1996). In nonhierarchic MDO methods, there are no restrictions on the communication between different elements of the MDO problem.

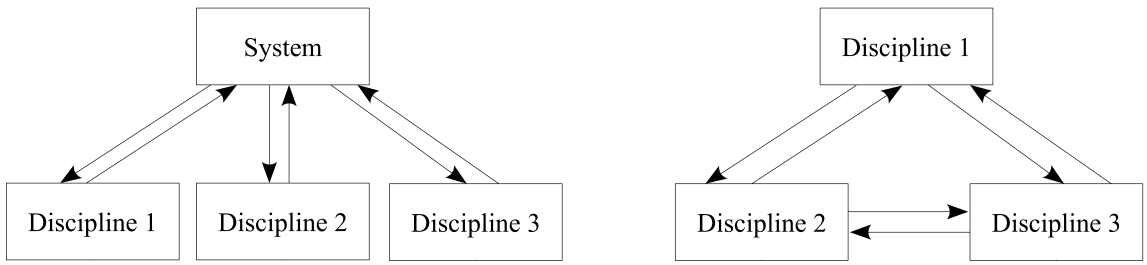


Figure 2.1. Illustration of a hierarchical (left) versus a nonhierarchical (right) MDO system.

MDO methods may also be classified as single level or multi-level. Single level use a single optimizer to determine the local and global design variables. Multi-level methods transform the the original problem into a structure with multiple levels where each level in a multi-level MDO method can have its own optimizer (Balling and Sobieszczanski-Sobieski 1996).

2.3.1 Multidisciplinary Feasible Method

The multidisciplinary feasible (MDF) method (Cramer et al. 1994) is perhaps the most intuitive and direct approach to solving the MDO problem. The MDF method is a single level method that solves the MDO problem in Equation (2.22) directly by performing discipline analyses each time the objective function or constraints must be evaluated (Tedford and Martins 2010).

The advantage of this method is that the design at each iteration is feasible because the discipline analyses are performed at each iteration. A disadvantage of this method is that it can become very computationally expensive when solved using gradient-based solvers (which utilize a finite difference to estimate the gradient) (Tedford and Martins 2010). Use of a gradient-based optimizer can lead to a high number of discipline evaluations, which is problematic when the discipline analyses are expensive.

2.3.2 Individual Discipline Feasible Method

The individual discipline feasible (IDF) method (Cramer et al. 1994) is a single level MDO method that offers improvement over the MDF method by reducing the number of discipline analyses required during the optimization. This is achieved by removing the multidisciplinary feasibility requirement and only enforcing the discipline feasibility at each iteration.

The IDF method uncouples the disciplines by introducing target values for the coupling variables, denoted $\mathbf{y}^t$, which are used as estimates for the coupling variables. The discipline analyses can be performed using the target values instead of evaluating the true values of the coupling variables, which reduces the number of times the discipline analyses must be performed in each iteration. The optimization statement includes a constraint that requires the target values to agree with the coupling variable values at the end of the optimization. The IDF optimization statement is (Tedford and Martins 2010)

$$\min_{\mathbf{x},\mathbf{z},\mathbf{y}^t} f\left(\mathbf{x},\mathbf{z},\mathbf{y}^t\right) \tag{2.23}$$

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

$$\mathbf{y}_i^t - \mathbf{y}_i(\mathbf{x}, \mathbf{y}_j^t, \mathbf{z}) = \mathbf{0}$$

where $\mathbf{y}_i$ are the coupling variables from discipline i evaluated using the target values from the other disciplines, $\mathbf{y}_j^t$.

2.3.3 All-at-Once Method

The all-at-once (AAO) method, is a single level MDO method that does not enforce multidisciplinary feasibility or discipline feasibility at each iteration of the optimization. The reasoning behind this approach is to reduce the time spent seeking feasibility when the design is far from the optimum (Cramer et al. 1994).

To solve the AAO optimization problem, the residual R_i is used to represent the governing equations for the discipline i analysis, where $R_i(\mathbf{x}, \mathbf{z}, \mathbf{y}, \mathbf{u}) = 0$ indicates a solution of the discipline analysis. The vector $\mathbf{u}$ contains state variables which are variables only used within the discipline. Then the optimization statement for the AAO method is (Tedford and Martins 2010)

$$\min_{\mathbf{x},\mathbf{z},\mathbf{u}} f\left(\mathbf{x},\mathbf{z},\mathbf{y}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right)\right) \tag{2.24}$$

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

$$\mathbf{R}(\mathbf{x}, \mathbf{z}, \mathbf{y}(\mathbf{x}, \mathbf{z}, \mathbf{u}), \mathbf{u}) = \mathbf{0}$$

A clear disadvantage of this method is that the user must be able to evaluate the residuals **R**; for practical applications the governing equations for the analysis may not be available, particularly when using external software to evaluate a discipline (Tedford and Martins 2010). Additionally, the AAO method includes a large number of constraints and a very large number of design variables (**x**, **z**, and **u**) (Cramer et al. 1994).

2.3.4 Collaborative Optimization

Collaborative optimization (Braun and Kroo 1995) is a multi-level (bi-level) method which increases the disciplines' autonomy by separating the disciplines into independent optimization problems. The collaborative optimization formulation includes a system level optimization statement with compatibility constraints to ensure agreement between disciplines, which is necessary due to the increased disciplinary autonomy.

The top level optimization statement is (Tedford and Martins 2010)

$$\min_{\mathbf{x}_{obj},\,\mathbf{z},\,\mathbf{y}} f\left(\mathbf{x}_{obj},\mathbf{z},\mathbf{y}\right) \tag{2.25}$$

$$\text{subject to} \quad J_i^* = 0 \quad i = 1, \ldots, p$$

where $\mathbf{x}_{obj}$ are only the local variables that affect the objective function. J_i represents the interdisciplinary compatibility of discipline i, and J_i^* is the solution to (Tedford and Martins 2010)

$$\min_{\mathbf{x}_i,\mathbf{z}_i,\mathbf{y}_{i,j}} J_i = \sum(\mathbf{z}_i - \mathbf{z}_i^t)^2 + \sum(\mathbf{x}_{i,obj} - \mathbf{x}_{i,obj}^t)^2 + \sum(\mathbf{y}_i - \mathbf{y}_i^t)^2 + \sum(\mathbf{y}_{j,i} - \mathbf{y}_{j,i}^t)^2 \tag{2.26}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}_i, \mathbf{z}_i, \mathbf{y}_i(\mathbf{x}_i, \mathbf{y}_{j,i}, \mathbf{z}_i)) \leq \mathbf{0}$$

Collaborative optimization utilizes target values (denoted with superscript t) for the design variables that are used in the system. The use of the target values with the compatibility constraint ($J_i^* = 0$) allows the disciplines to optimize more independently, so the structure of collaborative optimization resembles the operation of disciplinary design teams (Tedford and Martins 2010).

One disadvantage of collaborative optimization is that the compatibility constraint can cause difficulties with convergence during the optimization (Zhao and Cui 2011); without convergence in the compatibility constraint, the disciplines will not agree on the global properties.

2.3.5 Concurrent Subspace Optimization

Concurrent subspace optimization (CSSO) is a multilevel (bilevel) and nonhierarchic MDO method (Sobieszczanski-Sobieski 1988). CSSO uses an approximation for the coupling variables (using response surfaces, for example); the advantage of this approximation is that the system optimization problem can be solved very quickly. The disadvantage of this method is that it requires building the approximation model before beginning the optimization. Additionally, the design variables are divided among the disciplines according to which discipline the design variable has the most impact on; the other variables are held constant during the discipline optimization (Tedford and Martins 2010).

The system level optimization problem is (Tedford and Martins 2010)

$$\min_{\mathbf{x},\,\mathbf{z}} f\left(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}}\right) \tag{2.27}$$

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

where $\tilde{\mathbf{y}}$ is the approximation for the coupling variables $\mathbf{y}$. The discipline optimization problem for discipline i is

$$\min_{\mathbf{x}_i,\,\mathbf{z}_i} f\left(\mathbf{x}_i, \mathbf{x}_{0,} \mathbf{z}_i, \mathbf{z}_0, \mathbf{y}_i\left(\mathbf{x}_i, \mathbf{x}_{0,} \mathbf{z}_i, \mathbf{z}_0, \tilde{\mathbf{y}}_j\right), \tilde{\mathbf{y}}_j\right) \tag{2.28}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}_i, \mathbf{x}_0, \mathbf{z}_i, \mathbf{z}_0, \mathbf{y}_i(\mathbf{x}_i, \mathbf{x}_0, \mathbf{z}_i, \mathbf{z}_0, \tilde{\mathbf{y}}_j), \tilde{\mathbf{y}}_j)$$

where $\mathbf{x}_i$ and $\mathbf{z}_i$ are the design variables used in discipline i, and the variables $\mathbf{x}_0$ and $\mathbf{z}_0$ are held constant during the discipline optimization. $\tilde{\mathbf{y}}_j$ is the approximation for the coupling variables from the other disciplines.

2.3.6 BLISS

Bi-level integrated system synthesis (BLISS) is a multilevel method, with a system-level optimization and discipline optimizations. The system level optimization optimizes the objective function with respect to the system design variables **z** (Sobieszczanski-Sobieski, Agte, and Sandusky 1998):

$$\min_{\Delta \mathbf{z}} f(\mathbf{x}, \mathbf{z}) = f_0 + \frac{df}{dz} \Delta \mathbf{z} \tag{2.29}$$

$$\text{subject to} \quad \mathbf{z}_{lb} \le \mathbf{z} + \Delta\mathbf{z} \le \mathbf{z}_{ub}$$
$$\Delta\mathbf{z}_{lb} \le \Delta\mathbf{z} \le \Delta\mathbf{z}_{ub}$$

where the notation Δ indicates an incremental step and subscripts *lb* and *ub* indicate the lower and upper bounds for the variable, respectively. The subscript 0 indicates information from the present state (initial or previous iteration).

At the discipline level, optimization is performed by varying the local design variables **x** but holding the system-level design variables **z** constant. The discipline-level optimization statement does not directly optimize the objective function *f*, but instead optimizes a function that measures the influence of that discipline's design variables' on the system objective function. The discipline objective function is a weighted sum of the local design variables **x** weighted by an approximation for the derivative of *f* with respect to **x**. The discipline optimization statement is (Sobieszczanski-Sobieski, Agte, and Sandusky 1998)

$$\min_{\Delta \mathbf{x}} \left(\frac{df}{d\,\Delta\mathbf{x}} \right)^{\mathrm{T}} \Delta\mathbf{x} \tag{2.30}$$

$$\text{subject to} \quad \mathbf{g}_i(\mathbf{x} + \Delta\mathbf{x}, \mathbf{z}, \mathbf{y}) \le \mathbf{0}$$

where the discipline i is assumed to have access to the coupling variables $\mathbf{y}$ from all disciplines.

2.3.7 Analytical Target Cascading

Analytical target cascading (ATC) is a multilevel, hierarchical method that is useful for MDO problems with design targets and multiple responses of interest. Instead of the MDO problem defined in Equation (2.22), which has a single objective function f, ATC is interested in problems of the form (Kim et al. 2003)

$$\min_{\mathbf{x}} \left\| \mathbf{T} - \mathbf{R}(\mathbf{x}) \right\| \tag{2.31}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0}$$

where $\mathbf{T}$ is a vector of target values and $\mathbf{R}$ is a vector of response values from the system analysis. Target values are useful because in industry, designs are often defined by requirements, instead of optimal performance (Agte et al. 2010).

The formulation of ATC includes three levels: supersystem, system, and subsystem (disciplines). The goal of the supersystem (subscript "*sup*") level optimization statement is to minimize the difference between the supersystem level responses $\mathbf{R}_{sup}$ and target values $\mathbf{T}_{sup}$. The system level constraints include the system design constraints $\mathbf{g}_{sup}$ and the tolerance constraints which include the deviation tolerances $\varepsilon_{\mathbf{R}}$ and $\varepsilon_{\mathbf{y}}$. The deviation tolerances are supersystem-level design variables that coordinate the responses and linking variables; when the optimization has converged the deviation tolerances should reach zero to achieve consistency. The supersystem optimization statement is (Kim et al. 2003)

$$\min_{\mathbf{x}_{sup},\, \mathbf{y}_s,\, \varepsilon_{\mathbf{R}},\, \varepsilon_{\mathbf{y}}} \quad \left\|\mathbf{R}_{sup} - \mathbf{T}_{sup}\right\| + \varepsilon_{\mathbf{R}} + \varepsilon_{\mathbf{y}} \tag{2.32}$$

$$\text{subject to} \qquad \sum_k \left\|\mathbf{R}_{s,k} - \mathbf{R}^L_{s,k}\right\| \le \varepsilon_{\mathbf{R}}$$

$$\sum_k \left\|\mathbf{y}_{s,k} - \mathbf{y}^L_{s,k}\right\| \le \varepsilon_{\mathbf{y}}$$

$$\mathbf{g}_{sup}(\mathbf{x}_{sup}) \le \mathbf{0}$$

where the superscript L indicates target values determined from the system level optimization. The subscript "*s*" indicates elements evaluated at the system level optimization.

The goal of the system level optimization is to coordinate the supersystem and subsystem (discipline) optimizations. The system level optimization problem is (Kim et al. 2003)

$$\min_{\mathbf{x}_s,\, \mathbf{y}_s,\, \mathbf{y}_{ss},\, \varepsilon_{\mathbf{R}},\, \varepsilon_{\mathbf{y}}} \quad \left\|\mathbf{R}_s - \mathbf{R}^U_s\right\| + \left\|\mathbf{y}_s - \mathbf{y}^U_s\right\| + \varepsilon_R + \varepsilon_y \tag{2.33}$$

$$\text{subject to} \qquad \sum_k \left\|\mathbf{R}_{ss,k} - \mathbf{R}^L_{ss,k}\right\| \le \varepsilon_{\mathbf{R}}$$

$$\sum_k \left\|\mathbf{y}_{ss,k} - \mathbf{y}^L_{ss,k}\right\| \le \varepsilon_{\mathbf{y}}$$

$$\mathbf{g}_s(\mathbf{x}_s, \mathbf{y}_s) \le \mathbf{0}$$

where the subscript "*ss*" indicates the subsystem level and the superscript "*U*" indicates the targets from the supersystem level.

Finally, the goal of subsystem optimization problem is to minimize the difference between the values of responses and linking variables at the subsystem level and the system target values. The subsystem level represents the discipline analyses. The subsystem optimization statement is (Kim et al. 2003)

$$\min_{\mathbf{x}_{ss},\,\mathbf{y}_{ss}} \left\|\mathbf{R}_{ss}-\mathbf{R}_{ss}^{U}\right\| + \left\|\mathbf{y}_{ss}-\mathbf{y}_{ss}^{U}\right\| \tag{2.34}$$

$$\text{subject to} \quad \mathbf{g}_{ss}(\mathbf{x}_{ss}, \mathbf{y}_{ss}) \le \mathbf{0}$$

The ATC method may seem overly complex at first because of its use of multiple levels, however, there are two key features of the method. First, the capability to handle multiple objective functions with target values, and second, the use of the deviation tolerances ε as design variables used to motivate consistency.

2.3.8 Multiobjective Optimization

Multiobjective optimization is a field that is closely related to MDO. Multi-objective optimization is concerned with optimization problems with more than one objective function. Because this work is concerned with MDO problems of the form of with multiple objectives, an understanding of the fundamentals of multiobjective optimization is important.

The standard form of a multiobjective optimization problem is (Marler and Arora 2004)

$$\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x})=[f_1(\mathbf{x})\ f_2(\mathbf{x})\ \ldots\ f_p(\mathbf{x})]^{\mathrm{T}} \tag{2.35}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0}$$

As defined earlier in the chapter, $\mathbf{x}$ represents the vector of design variables (length n). There are p objective functions contained in the vector $\mathbf{f}$. The vector $\mathbf{g}$ contains m constraints. It is also possible to include equality constraints, but only inequality constraints are considered in this work.

Multiobjective optimization problems include more than one objective function and it is not clear how to minimize multiple functions simultaneously, and in general, each of the objective functions will have a different optimum when considered independently. Conflicting objectives are very common in multidisciplinary design problems; for example, in structural design, a design with greater strength will often have increased weight.

Because the objective functions have different optima, it is not possible to select a single optimum for a multiobjective optimization problem. Instead, solutions to multiobjective optimization problems are described by Pareto optimality. The definition of Pareto optimal from Marler and Aurora (2004) is: A point $\mathbf{x}^*$ is Pareto optimal if and only if there does not exist a point $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*)$ and $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for at least one i. In plain language, $\mathbf{x}^*$ "... is Pareto optimal if there is no other point that improves at least one objective function without detriment to another function," (Marler and Aurora 2004). The concept of weakly Pareto optimal is also useful in practice; a point $\mathbf{x}^*$ is weakly Pareto optimal if there are no points that are better in all of the objective functions.

The goal of multiobjective optimization methods is to locate and describe all Pareto optimal points (called the Pareto set) for a given problem. Methods for efficiently finding Pareto optimal points will not be discussed here, as it is not the goal of this research; however, understanding the meaning of the Pareto set is important for interpreting the performance of a proposed solution to a multiobjective (or multidisciplinary) optimization problem.

2.4 MDO Problem Definition

The MDO problem statement in Equation (2.22) is relevant in many applications. However, in this work, a more general problem statement is studied with p disciplines and p objective functions:

$$\min_{\mathbf{x}\in\chi} \ \mathbf{f}(\mathbf{x}) \tag{2.36}$$

$$\text{subject to } \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0} \quad i = 1, \ldots, p$$

where $\mathbf{x}$ is the vector of design variables (length n) and $\mathbf{g}_i$ is the vector of constraints for discipline i. As in Table 2.1, χ denotes the set of allowable values (ranges) for the design variables.

The MDO statement in Equation (2.36) includes multiple objective functions f_i, so for p objective functions, $\mathbf{f}$ is a vector of length p. This definition is more general, because for a multidisciplinary design problem it may not be possible to identify a single objective function. For example, a ship design may be concerned with minimizing hull resistance and minimizing cost, where the two objectives are evaluated in different disciplines. The definition of an MDO problem with multiple objectives is not unusual; the analytical target cascading method also utilized an MDO model with more than one objective function.

CHAPTER 3

Multidisciplinary Design Optimization Algorithm with Target Values

The previous chapter introduced multidisciplinary design optimization (MDO), including MDO techniques and applications of MDO in engineering. In this chapter, a new MDO algorithm is presented. The new algorithm falls into the category of multilevel algorithms and it utilizes target values at the system level to coordinate the exchange of information between disciplines. This chapter begins with the mathematical details of the algorithm and then presents an application of the algorithm to a simple MDO example problem.

3.1 Motivation and Problem Definition

The previous chapter discussed the value of MDO as an engineering tool; the utility of MDO motivates the development of the new MDO algorithm. The reasoning behind the development of this algorithm was to offer improved efficiency over other multilevel MDO methods. For example, the analytical target cascading method presented in Section 2.3.7 utilizes the additional variables $\varepsilon_{\mathbf{R}}$ and $\varepsilon_{\mathbf{Y}}$ (compatibility deviations) as defined in Equation 2.32. While both analytical target cascading and the new MDO algorithm utilize target values, the new MDO algorithm does not use compatibility variables. The formulation without compatibility deviation variables is an attempt to offer improved efficiency.

In many multidisciplinary engineering applications, one discipline addresses a global metric, such as cost or weight, that is considered to be the most significant. The new MDO method allows the user to select one discipline as the "top" level discipline, which is then given more significance during the optimization as is appropriate for a discipline of a global nature.

The general MDO problem was defined in Chapter 2 in Equation (2.36). For the new MDO algorithm, the general MDO problem is rewritten in the following form

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\chi} f_T(\mathbf{x}) & \min\limits_{\mathbf{x}\in\chi} f_i(\mathbf{x}) \quad \text{for all } i \\ \text{subject to } \mathbf{g}_T(\mathbf{x}) \le \mathbf{0} & \text{subject to } \mathbf{g}_i(\mathbf{x}) \le \mathbf{0} \end{array} \tag{3.1}$$

where $\mathbf{x}$ is the vector of design variables, f denotes an objective function, and g denotes a vector of constraints. In the formulation in Equation (3.1), the subscript T indicates the top level discipline. While the MDO problem defined in Equation (3.1) uses different notation, it is mathematically equivalent to the MDO problem in Equation (2.36). Figure 3.1 shows a diagram of the MDO problem formulation with a top level discipline.

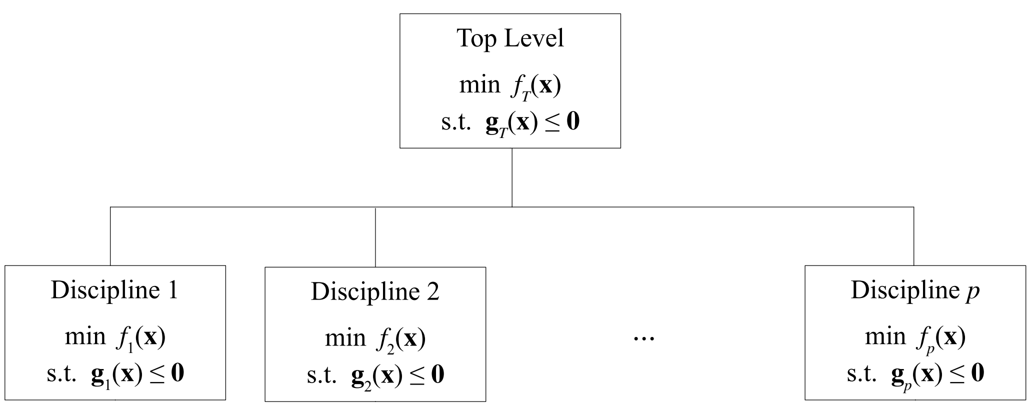


Fig. 3.1. Illustration of the MDO problem with top level discipline.

3.2 MDO Approach Utilizing Target Values

The purpose of this section is to present the mathematical formulation of the new MDO algorithm and demonstrate the use of the MDO algorithm in a simple example problem.

3.2.1 Algorithm Definition

The mathematical formulation includes two elements: the discipline level optimizations and the system level optimization. The system level optimization coordinates the exchange of information between the disciplines, as illustrated in Figure 3.2. The following definition begins with the discipline level optimizations which are used to evaluate the target values for the design variables. The target values are returned to the system level optimization statement which coordinates the entire problem optimization.

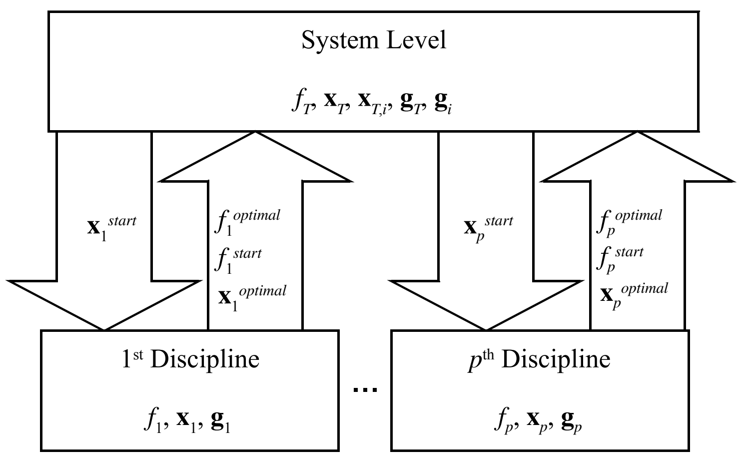


Fig. 3.2. Flow chart of the multidisciplinary design optimization algorithm.

At the beginning of each iteration of the system level optimization, there are current values for the design variables; the current values for the design variables are sent to the disciplines as starting values $\mathbf{x}^{start}$. An optimization of each discipline is performed independently using $\mathbf{x}^{start}$ as starting values for the design variables. For the discipline level optimizations, the design variable ranges are restricted to a subset of their original ranges. In the discipline optimizations, the design variables are allowed to vary from $\mathbf{x}^{start}$ by ±20%; the reduced design variable space is denoted ψ, so that $\mathbf{x} \in \psi \subset \chi$. There are two reasons for performing the discipline optimizations over the subspace ψ. First, optimization over the (same) original design space would often result in the same solution at every iteration, so the system optimization may not progress. Second, optimization over the subspace reduces the difference between $\mathbf{x}^{start}$ and the optimal values of the design variables returned by the discipline optimization; the limits help prevent the design from changing too rapidly which may cause it to converge rapidly in a direction that is suboptimal for the system. (Note that the 20% variation can be adjusted to another value if needed for the specific application.)

When the discipline optimizations are performed, each discipline independently determines optimal values for the design variables; the optimal values for the design variables for discipline i are denoted $\mathbf{x}_i^{optimal}$ and the value of objective function f_i at $\mathbf{x}_i^{optimal}$ is $f_i^{optimal}$. Each discipline returns its values of $\mathbf{x}_i^{optimal}$ and $f_i^{optimal}$ to the system level optimization statement.

The optimal values of the design variables and objective functions from the discipline optimizations are used to evaluate target values for the design variables based on the improvement achieved in the discipline objectives functions:

$$x_j^{target} = \frac{\sum_{i=1}^{N_j} x_{j,i}^{optimal} \left(f_i^{start} - f_i^{optimal} \right)^2}{\sum_{i=1}^{N_j} \left(f_i^{start} - f_i^{optimal} \right)^2} \tag{3.2}$$

In Equation (3.2), x_j^{target} is the target value for the j^{th} design variable. $x_{j,i}^{optimal}$ indicates the optimal value of design variable x_j in the i^{th} discipline. f_i^{start} is the is the initial value of objective function f_i for the current iteration: $f_i^{start} = f_i(\mathbf{x}_i^{start})$. N_j is the total number of disciplines that share the j^{th} design variable.

The target values for the design variables are used to evaluate the system objective function f_{system}:

$$f_{system}(\mathbf{x}) = f_T(\mathbf{x}) + \|\mathbf{x}^{target} - \mathbf{x}\|_2^{\ 2} \tag{3.3}$$

where f_T is the top objective function. Then the system level optimization statement is

$$\begin{aligned} \min_{\mathbf{x}\in\chi} \quad & f_{system}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_{\mathrm{T}}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0} \text{ for all } i \end{aligned} \tag{3.4}$$

The effects of the disciplines are included at the system level by requiring that the design variables approach the target values determined from the discipline optima, and at the same time the top level objective function is improved. All of the discipline level constraints are enforced in the system level optimization along with the top level constraints; this ensures that the optimal point will also be a feasible point for all of the disciplines. The procedure for iteration in the MDO algorithm is illustrated in Figure 3.3.

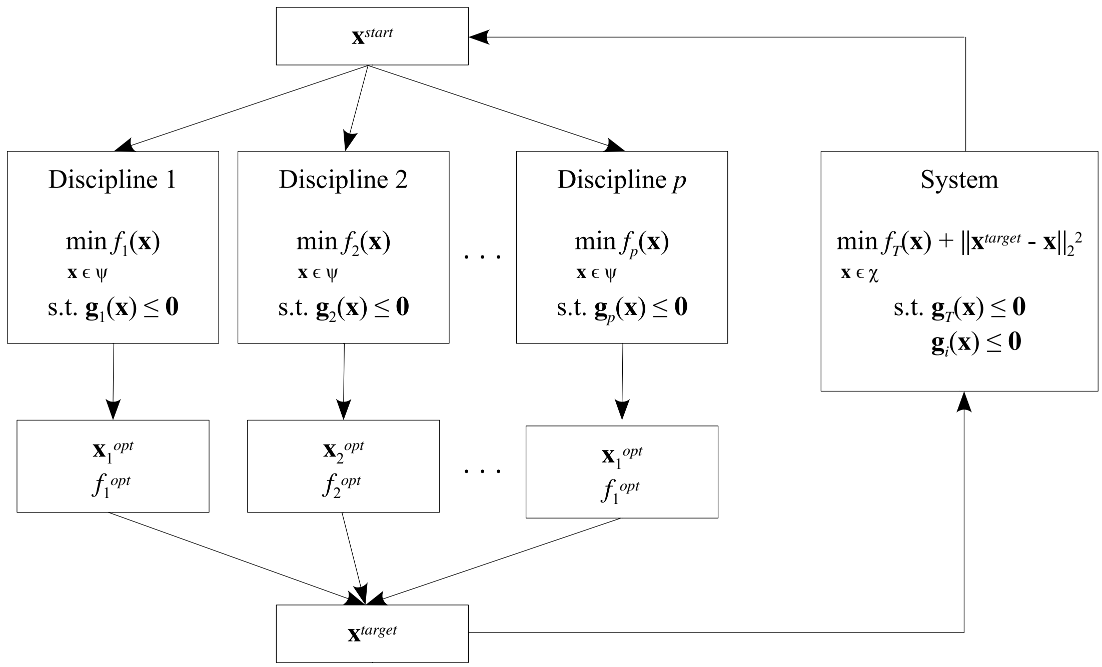


Fig. 3.3. Flow chart for the calculations performed in the iterations of the MDO algorithm.

3.2.2 Analytical Example Problem

A simple analytical example was tested to demonstrate the effectiveness of the MDO algorithm. The example problem used three design variables: x_1, x_2, and x_3, where the range for each design variable was $0 \leq x_i \leq 2$ and the initial value for each of the design variables was 1. Three simple polynomial objective functions were defined as:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 + x_2 + x_3 \\ f_2(\mathbf{x}) &= x_1 + x_2 - x_3^2 \\ f_3(\mathbf{x}) &= -x_1 x_2 x_3 \end{aligned} \tag{3.5}$$

The example is purely analytical, so none of the objectives signify a top level objective; all three objective functions were designated as discipline level objectives. No additional constraints were included in the problem, aside from the bounds on the design variables. Figure 3.4 illustrates the setup of the example problem.

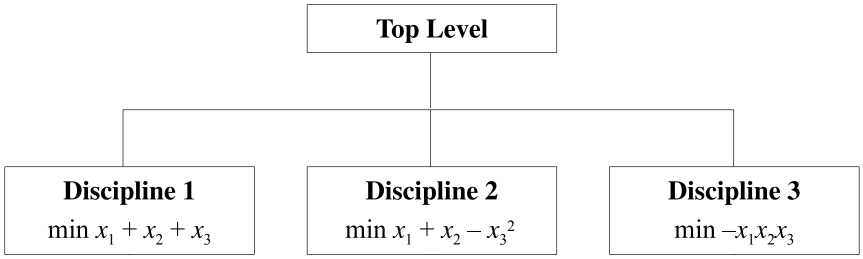


Fig. 3.4. Diagram illustrating the simple MDO example.

The results of minimizing the objective functions independently can provide insight into the MDO problem. The individual discipline optima can easily be observed for this simple example: the minimum for f_1 is 0 at $\mathbf{x}_{f1}^* = (0, 0, 0)$, the minimum for f_2 is -4 at $\mathbf{x}_{f2}^* = (0, 0, 2)$, and the minimum for f_3 is -8 at $\mathbf{x}_{f3}^* = (2, 2, 2)$.

The solution given by the MDO algorithm for this simple analytical example problem is shown in Table 3.1, which includes a comparison to the starting point (which was arbitrarily chosen as the midpoint of the design variables' range). The optimal values of x_1 and x_2 are equal, which is expected because the problem is symmetric in these two design variables. When compared with the initial point, the optimized solution results in a significant improvement in the objective functions f_2 and f_3, while objective function f_1 increased slightly. This result is reasonable because tradeoff between the disciplines is common in MDO problems.

Table 3.1. Results from the simple MDO example.

	Initial Value	Optimal Value
x_1	1	1.128
x_2	1	1.128
x_3	1	1.745
$f_1(\mathbf{x})$	3	4.002
$f_2(\mathbf{x})$	1	-0.790
$f_3(\mathbf{x})$	-1	-2.222

A plot of the Pareto front was generated to visualize the results for the simple analytical example problem. (Points on the Pareto front were determined using a simple Monte Carlo approach, by generating a large number of random feasible points and plotting the dominant points; while this approach is highly inefficient, efficient plotting of the Pareto front was not the purpose of this exercise.) The optimal point returned by the MDO algorithm is plotted along with the Pareto front in Figure 3.5; the MDO optimum is indicated by the (red) star. Figure 3.5 also includes the individual discipline optima for comparison, where $\mathbf{x}_{fi}^*$ indicates that the objective functions are evaluated at the optimum of f_i.

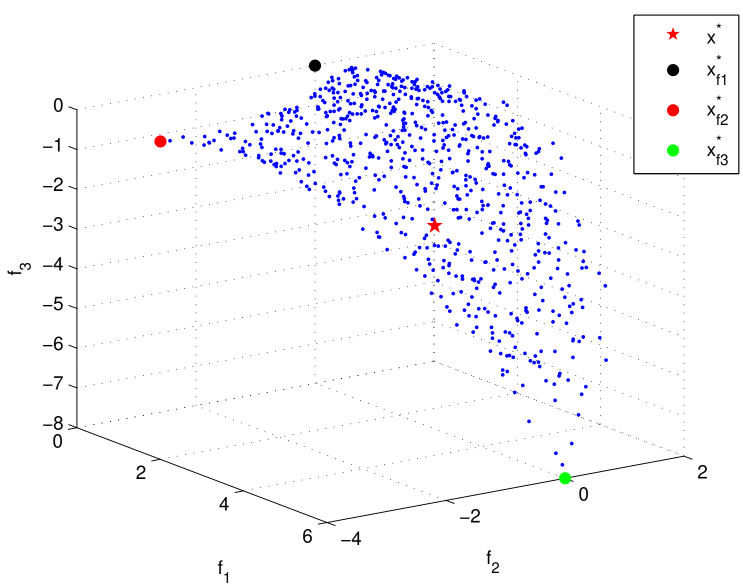


Fig. 3.5. Pareto front for the simple analytical example, with optimum (star).

Most importantly, Figure 3.5 shows that the optimum lies on the Pareto front, indicating that no further improvement is possible. If the optimum did not lie on the Pareto front, further improvement would be possible, indicating that the result returned by the algorithm was suboptimal. Second, the optimum does not lie in a particularly extreme position; that is, it does not appear to strongly favor one discipline's objective over another, indicating that the result provides a balance between the disciplines.

3.3 Chapter Summary

In this chapter, a new MDO algorithm using target values was presented. The algorithm was applied to a simple demonstrative example problem.

CHAPTER 4

Multidisciplinary Design Optimization Under Uncertainty

In traditional optimization, the optimal solution is a single point that minimizes a function of interest while satisfying a prescribed set of constraints. However, in real design problems, design variables and environmental parameters contain uncertainty; for example, manufacturing tolerances and environmental conditions are beyond the control of the designer, but they can affect the performance of the design. Additionally, the simulation tools or regression models which provide performance predictions during design development are additional sources of uncertainty; even highly advanced simulations cannot predict physical performance exactly. Due to the variations in the design variables and parameters with uncertainty, or due to the uncertainty in the performance prediction models, the response of the optimal design will differ from the deterministic expectation. This can lead to violation of constraints on the design and/or deterioration of the expected optimal performance.

Often, the effects of uncertainty are not considered during the ship design process. Capturing the effects of uncertainty in the early stages of the design process can eliminate the need for expensive design modifications at later stages. The importance of accounting for uncertainty in ship design was highlighted recently by an entire special issue of the *Naval Engineers Journal* (Vol. 114, No. 2, 2002) which was devoted to the subject of reliability-based ship structural design.

This study is motivated by the importance of managing uncertainty in the ship design process. In this chapter, two methods for accounting for uncertainty in design optimization are presented: robust optimization and reliability-based design optimization. These approaches are incorporated with the multidisciplinary design optimization algorithm presented in Chapter 3 to study the design optimization of a ship hull form.

4.1 Accounting for Uncertainty in Design Optimization

When discussing the concept of uncertainty in this work, the term "uncertainty" refers specifically to what is known as aleatory uncertainty in the field of uncertainty quantification. As defined by Noor (2005), "Aleatory uncertainty... is used to describe the inherent spatial and temporal variation associated with the physical system or the environment under consideration as well as the uncertainty associated with the measuring device." This is in contrast to epistemic uncertainty which describes the lack of information or understanding about the system. Only aleatory uncertainty is considered in this analysis because the approaches used require the uncertainties to be definable in simple probabilistic terms.

Methods for accounting for uncertainty in optimization (also called nondeterministic approaches) can be classified into three categories (Noor 2005): probabilistic analysis, fuzzy set approaches, and set-theoretical approaches. For probabilistic analysis, the source of uncertainty is assumed to be system parameters and/or design variables that are random variables for which probability distribution functions can be chosen. Fuzzy set and set-theoretical approaches are appropriate for modeling systems with less information about the sources of uncertainty. Methods in probabilistic analysis are used in this work.

4.1.1 Reliability-Based Design

In reliability-based design, the effects of uncertainty in design variables and parameters on the constraints are considered. The goal of reliability is to ensure that the optimal solution will satisfy the constraints in the presence of uncertainty within a prescribed level of confidence.

A simple two-dimensional illustration in Figure 4.1 shows an example of reliability in a constraint. The figure shows the design space for two design variables and a constraint boundary (black line with hatching) defines the feasible space which lies to the left of the constraint. The contour curves (blue) for a single objective function show that the objective function improves while moving to the right. In deterministic optimization, the optimum would be selected at the green circle lying on the constraint boundary because this choice gives the best value of the objective function. However, if the realizations of the design variables are determined by the approximate probability distributions shown along the axes, the actual outcome of the design could lie anywhere within the cloud of points near the deterministic optimum, including points that violate the constraint. A reliable optimum is located further from the constraint boundary, at the red circle, so that the design realizations are more likely to satisfy the constraint.

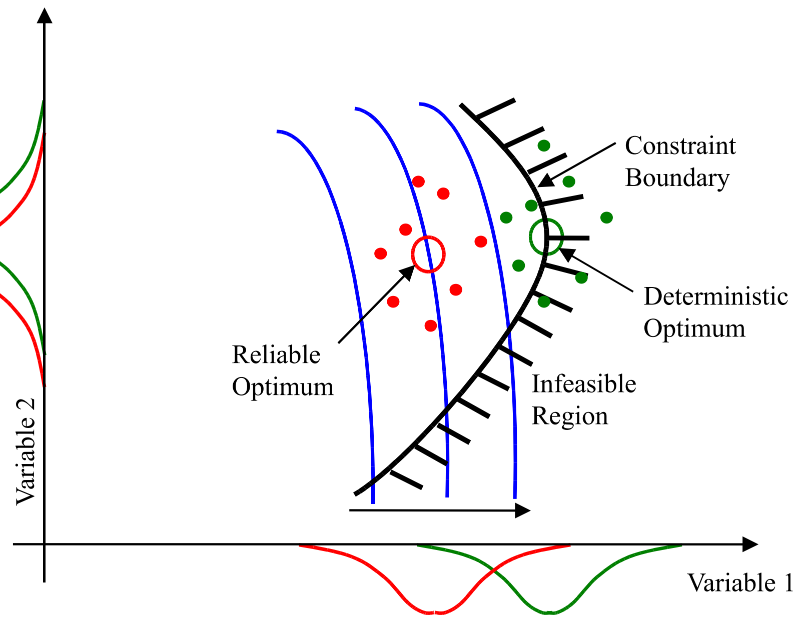


Fig. 4.1. Illustration of reliability in a constraint for the case of two design variables. The black line represents a constraint and the blue lines represent contours of an objective function.

Much of the development in the field of reliability-based design occurred in the field of structural design. However, reliability-based design optimization has more recently been applied to a wide array of engineering design problems in the literature. Acar and Solanki (2009) applied reliability-based design to automobile design for crashworthiness. Allen and Maute (2004) applied reliability-based design to an airplane wing structure. Yang et al. (2005) applied reliability-based design to the design of an exhaust system. Maute and Frangopol (2003) used reliability-based design for the analysis of micro-electromechanical systems (MEMS).

Reliability-based design optimization has also been applied to multidisciplinary design optimization problems. Ahn and Kwon (2006) utilized reliability-based design for the multidisciplinary design optimization of a jet. Sinha (2007) presented an approach for multiobjective optimization with reliability for automobile crashworthiness. Yao et al. (2011) provide a thorough review of reliability methods in multidisciplinary design optimization in aerospace applications.

The purpose of reliability analysis is to assess the probability that a design satisfies constraints. For the case of a single constraint g, the constraint is satisfied when $g(\mathbf{x}) \leq 0$, where $\mathbf{x}$ is the vector of design variables, which are also random variables. Then denote the probability of violating the constraint as p_f, also called the probability of failure (Haldar and Mahadeval 2000), and the probability can be calculated according to

$$p_f = \int \dots \int_{g>0} f_{\mathbf{x}}(x_1, \dots, x_n)\, dx_1 \dots dx_n \tag{4.1}$$

where $f_{\mathbf{x}}$ is the joint probability distribution function (PDF) for $\mathbf{x}$. The integral is performed over the region where the constraint is violated, that is, where $g(\mathbf{x}) > 0$. Furthermore, the reliability index β is related to the probability of failure by

$$p_f = \Phi(-\beta) = 1 - \Phi(\beta) \tag{4.2}$$

where Φ is the the cumulative distribution function for the standard normal PDF.

Evaluation of the integral in Equation (4.1) is task is not trivial because the joint PDF can be very complex (Der Kiureghian 2005). One class of widely-used reliability methods is first-order reliability methods (FORM). First-order reliability methods are referred to as first-order because the constraint function g is approximated by a first-order Taylor series approximation so that the problem can be solved more easily (Haldar and Mahadeval 2000).

Reliability-based design optimization problems can be solved by transforming the probabilistic problem into a deterministic optimization problem. The general reliability-based design optimization problem given by the following expression (Liang, Mourelatos, and Tu 2008):

$$\min_{\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}}} f\left(\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}},\boldsymbol{\mu}_{\mathbf{P}}\right) \tag{4.3}$$

$$\text{s.t.} \quad \mathrm{P}\left[g_i\left(\mathbf{d},\mathbf{X},\mathbf{P}\right)\leq 0\right]\geq R_i \qquad i = 1, \ldots, m$$

$\mathbf{d}$ is the vector of deterministic design variables with the range $\mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U$. $\mathbf{X}$ is the vector of random design variables and $\boldsymbol{\mu}_{\mathbf{X}}$ is the vector of mean values of $\mathbf{X}$ with the range $\boldsymbol{\mu}_{\mathbf{X}}^L \leq \boldsymbol{\mu}_{\mathbf{X}} \leq \boldsymbol{\mu}_{\mathbf{X}}^U$. $\mathbf{P}$ is the vector of random parameters with mean $\boldsymbol{\mu}_{\mathbf{P}}$. f is the objective function and g_i are constraints. The notation P[·] indicates probability. R_i denotes the actual reliability level for the i^{th} constraint so that

$$R_i = 1 - p_{f_i} = 1 - \mathrm{P}\left[g_i\left(\mathbf{d},\mathbf{X},\mathbf{P}\right) > 0\right] \tag{4.4}$$

where the probability of failure is related to the reliability index β by Equation (4.2).

The multidisciplinary optimization method presented in Chapter 3 is capable of solving reliability-based design optimization problems. The optimization method incorporates the reliability-based design optimization framework described by Liang, Mourelatos, and Tu (2008). Traditional methods for solving RBDO problems use double-loop algorithm, with an outer loop to perform the optimization and an inner loop to evaluate reliability. However, the method defined by Liang, Mourelatos, and Tu (2008) is a single-loop algorithm; this method was selected and incorporated in this research because it offers computational savings.

The single-loop deterministic optimization problem statement is:

$$\min_{\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}}} f\left(\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}},\boldsymbol{\mu}_{\mathbf{P}}\right) \tag{4.5}$$

$$\text{s.t. } g_i(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i) \leq 0 \qquad i = 1, \ldots, m$$

again with the ranges $\mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U$ and $\boldsymbol{\mu}_\mathbf{X}{}^L \leq \boldsymbol{\mu}_\mathbf{X} \leq \boldsymbol{\mu}_\mathbf{X}{}^U$. The constraints g_i are functions of **d** and the quantities:

$$\mathbf{X}_i = \boldsymbol{\mu}_\mathbf{X} - \boldsymbol{\sigma}\beta_{t_i}\boldsymbol{\alpha}_i \tag{4.6}$$

$$\mathbf{P}_i = \boldsymbol{\mu}_\mathbf{P} - \boldsymbol{\sigma}\beta_{t_i}\boldsymbol{\alpha}_i \tag{4.7}$$

where $\boldsymbol{\sigma}$ the vector of standard deviations for the random variables **X** and random parameters **P**. β_{t_i} is the target reliability index for the i^{th} constraint (the desired level of reliability). $\boldsymbol{\alpha}_i$ is the normalized gradient for the i^{th} constraint:

$$\boldsymbol{\alpha}_i = \frac{\boldsymbol{\sigma} \cdot \nabla g_i(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i)}{\left\| \boldsymbol{\sigma} \cdot \nabla g_i(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i) \right\|} \tag{4.8}$$

In Equation (4.5), the objective function is evaluated at the mean point ($\mathbf{d}$, $\boldsymbol{\mu}_\mathbf{X}$, $\boldsymbol{\mu}_\mathbf{P}$), and the constraints are evaluated at ($\mathbf{d}$, $\mathbf{X}_i$, $\mathbf{P}_i$), which is called the most probable point. The most probable point is separated from the mean point in the direction away from the constraint, where the direction is determined using the normalized gradient information. The distance from the mean point is determined by the standard deviation (a measure of the amount of uncertainty in the design variables and parameters) and by the target reliability index (a measure of the desired confidence that the constraints will be satisfied).

The single-loop reliability-based design optimization algorithm was implemented with the MDO algorithm from Chapter 3. The reliability-based design optimization algorithm from this section describes a single-discipline optimization, not multidisciplinary optimization. The system level optimization in Equation (3.4) includes the constraints from all disciplines, so the reliability-based design optimization algorithm was applied to the system level objective function and constraints.

4.1.2 Robust Design

In robust design optimization, the effects of uncertainty in design variables and parameters on the objective functions are considered. The goal of robust design optimization is to ensure that the performance of the objective functions at the optimal solution will not severely deteriorate in the presence of uncertainty.

Figure 4.2 shows a simple illustration of robust optimization. In the figure, the horizontal axis represents a single design variable, and the vertical axis represents an objective function that is a function of the design variable. The objective function takes the values shown by the curve in black. The deterministic minimum of the objective function is at the blue dot, at the lowest point in the curve. However, if there is uncertainty in the design variable according to the (blue) bounds shown along the horizontal axis, the actual performance of the objective function could lie anywhere within the blue bounds on the vertical axis; the performance could deteriorate significantly from the deterministic optimum. A robust optimum is located at the red dot to the right, so that the variability in the objective function is reduced under the same variation in the design variable.

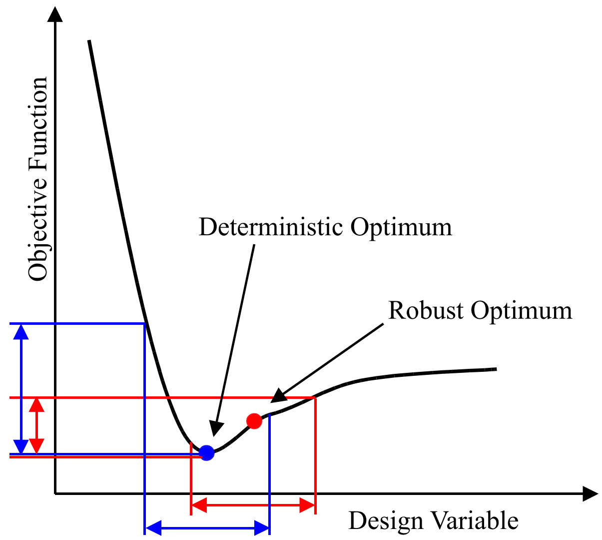


Fig. 4.2. Illustration of robust optimization for one objective function as a function of a single design variable.

Many applications of robust design optimization can be found in the literature. Choi et al. (2008) apply robust optimization is applied to the design of layered plate bonding. Reale-Levis, Romero, and Swiler (2008) use a robust design strategy for a snap-fit device. Steenackers, Guillaume, and Vanlanduit (2009) apply robust optimization to an airplane component. Delpiano and Sepúlveda (2006) introduce robustness in optimizing the design of a transistor device.

Robust optimization has also been applied to multidisciplinary design problems. Wang et al. (2009) use robust optimization with a multi-objective design application to a V6 engine. Ray and Smith (2006) apply a multi-objective evolutionary algorithm for robust design of a welded beam and a bulk carrier. Kovach and Cho (2008) use multidisciplinary robust design for chemical filtration optimization. Goh et al. (2010) compare different evolutionary multi-objective optimization algorithms for robust optimization.

Much of the development of robust design is credited to Taguchi, who worked from the perspective of quality engineering (Taguchi 1986). Taguchi viewed the variability in a product in terms of quality loss and developed the loss function to describe the loss in performance due to uncertainty. For the case of a performance metric y that is to be minimized under uncertainty, the loss function L is

$$L(y) = ky^2 \tag{4.9}$$

where k is a constant. The expected value of the loss function for y is

$$L = k(\sigma^2 + \mu^2) \tag{4.10}$$

where σ and μ are the standard deviation and mean, respectively, for y (Park 1996). This formulation for the loss function forms the basis for many robust optimization approaches.

The field of robust optimization is well-developed and many references for robust optimization methods are available in the literature. For an introduction and overview of the subject, Beyer and Sendhoff (2007) provide a detailed review of current robust optimization techniques. Schuëller and Jensen's (2008) review of approaches for optimization under uncertainty includes robust design optimization. Yao et al. (2011) include robust optimization techniques in their review of methods for multidisciplinary design optimization under uncertainty.

Before presenting the mathematical details of robust optimization methods, consider the following optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}) \tag{4.11}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0$$

where **x** denotes the vector of design variables and **p** is a vector of parameters. The value of the objective function may be very sensitive to changes in the design variables and parameters, so that the actual outcomes of the selected solution vary significantly (Schuëller and Jensen 2008).

The goal of robust design optimization is to address the optimization problem in Equation (4.11) and select a solution which is insensitive to changes in the design variables and parameters. In practice, there are two categories of robust optimization methods: methods which evaluate robustness metrics numerically, and methods which treat the uncertainties directly using simulation (Beyer and Sendhoff 2007). The approach for robust optimization used in this work evaluates robustness metrics; this approach is popular and practical because it transforms the complex robust optimization problem into a new problem in terms of the robustness metrics. Standard optimization techniques can then be used to solve the optimization problem with the robustness metrics (Schuëller and Jensen 2008).

Standard metrics for robustness are the mean and variance of the objective function at the selected design point, using the notation

$$\mathrm{E}[f \mid \mathbf{x}] = \mu_f \tag{4.12}$$

$$\mathrm{Var}(f \mid \mathbf{x}) = \sigma_f^2 \tag{4.13}$$

These metrics are useful in the robust optimization problem because the mean is the expected value for the objective function while the variance describes the variability in the objective function at the selected point **x**.

In order to evaluate the mean and variance during the optimization, the robust optimization in Equation (4.11) is redefined in the following manner:

$$\min_{\mathbf{x}} f(\mathbf{x}+\boldsymbol{\delta}, \mathbf{p}) \tag{4.14}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}+\boldsymbol{\delta}, \mathbf{p}) \leq 0$$

In this formulation, the vector of design variables $\mathbf{x}$ is considered deterministic, but with a random variation term $\boldsymbol{\delta}$ so that the probabilistic input value is described by $\mathbf{x} + \boldsymbol{\delta}$. Then the mean and variance of the objective function f can be evaluated by (Beyer and Sendhoff 2007):

$$\mu_f(\mathbf{x}) = \int f(\mathbf{x}+\boldsymbol{\delta}, \mathbf{p}) P(\boldsymbol{\delta}, \mathbf{p}) \, d\boldsymbol{\delta} \, d\mathbf{p} \tag{4.15}$$

$$\sigma_f^2(\mathbf{x}) = \int \left(f(\mathbf{x}+\boldsymbol{\delta}, \mathbf{p}) \right)^2 P(\boldsymbol{\delta}, \mathbf{p}) \, d\boldsymbol{\delta} \, d\mathbf{p} - \left(\mu_f(\mathbf{x}) \right)^2 \tag{4.16}$$

$P(\boldsymbol{\delta}, \mathbf{p})$ denotes the joint probability distribution function for $\boldsymbol{\delta}$ and $\mathbf{p}$. The analytical evaluation of the integrals in Equations (4.15) and (4.16) is generally not possible, so numerical approximation is used in practice (Beyer and Sendhoff 2007).

The general robust optimization statement using the mean and variance as metrics is (Yao et al. 2011)

$$\min_{\mathbf{x}} F(\mu_f(\mathbf{x}, \mathbf{p}), \sigma_f(\mathbf{x}, \mathbf{p})) \tag{4.17}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0$$

where F is a new objective function in terms of the mean and standard deviation (or variance). The simplest form of F is a weighted sum which is the form used in this research; then the robust objective function R_f for an objective function f is defined as

$$R_f(\mathbf{x}) = a\mu_f(\mathbf{x}) + (1 - a)\sigma_f(\mathbf{x}) \tag{4.18}$$

where a is a weighting parameter, $0 \leq a \leq 1$, that can be adjusted for the relative importance of the mean and standard deviation for the particular application.

The robust design optimization method was implemented with the MDO algorithm from Chapter 3. Although the robust design optimization method defined in this chapter describes single-discipline optimization, robust optimization can be applied to a multidisciplinary problem by transforming each of the discipline objective functions according to Equation (4.18).

4.2 Preliminary Design of a Bulk Carrier

The design process for a ship is typically divided into four phases, in order of increasing detail: concept design, preliminary design, contract design, and functional (or detail) design. The purpose of the first design phase, concept design, is to perform feasibility studies and identify a design that satisfies the design requirements, without necessarily performing detailed analyses (Gale 2003). This level of design is suitable for study in multidisciplinary design optimization because detailed engineering analysis is not required; instead, regression equations and empirical models that are simple to evaluate can be used to predict the design's performance.

Sen and Yang (1998) developed a multiobjective model for the concept design of a bulk carrier. Sen and Yang's model for the bulk carrier design is a collection of empirical and regression models, with a focus on economic aspects of the design. A further economic analysis of the design can be found in Hunt and Butman (1995). Details on the development of the bulk carrier model are available in Scher and Benford (1980).

The optimization of the bulk carrier concept design has been studied in the literature. Parsons and Scott (2004) applied a multicriterion design optimization approach to the bulk carrier model. Ray and Smith (2006) utilized neural networks to reduce the computational effort of the optimization. Hart and Vlahopoulos (2010) implemented a particle swarm optimizer to perform multidisciplinary design optimization of the bulk carrier. Xuebin (2009) performed optimization of the mutltiobjective bulk carrier model using multiple attribute decision making (an approach for evaluation of preference among Pareto-optimal solutions).

4.2.1 Bulk Carrier Model Definition

In Sen and Yang's (1998) model, the ship is described by six design variables: length L, beam B, depth D, draft T, block coefficient C_B, and speed V_k. The design variables are used to evaluate the three objective functions: lightship weight LS, annual cargo AC, and transportation cost TC. The problem also includes nine constraints. Table 4.1 shows the mathematical formulation for the objective functions, design variables, constraints, and other calculations used in this study of the bulk carrier design.

The goal of the multidisciplinary design optimization for the bulk carrier is to solve the problem

$$\begin{array}{ccc} \min LS(\mathbf{x}) & \min AC(\mathbf{x}) & \min TC(\mathbf{x}) \\ \text{subject to } \mathbf{g}(\mathbf{x}) \leq 0 & \text{subject to } \mathbf{g}(\mathbf{x}) \leq 0 & \text{subject to } \mathbf{g}(\mathbf{x}) \leq 0 \end{array} \tag{4.19}$$

where $\mathbf{x}$ is the vector containing the design variables, $\mathbf{x} = (L, B, D, T, C_B, V_k)$, and $\mathbf{g}$ is the vector containing all of the constraints (defined in Table 4.1).

Table 4.1. Bulk carrier model definition, from Sen and Yang (1998).

Objective Functions:		Design Variables:	
Lightship Weight (t)	$LS = W_S + W_O + W_M$	Length, L (m)	$150 \leq L \leq 274.32$
Annual Cargo (t/yr)	$AC = DW_C \times RTPA$	Beam, B (m)	$20 \leq B \leq 32.31$
Transportation Cost (£/yr)	$TC = \frac{C_A}{AC}$	Depth, D (m)	$13 \leq D \leq 25$
		Draft, T (m)	$10 \leq T \leq 11.71$
		Block Coefficient, C_B	$0.63 \leq C_B \leq 0.75$
		Ship Speed, V_k (knots)	$14 \leq V_k \leq 20$

Constraints:	
Length-to-beam ratio	$\frac{L}{B} \geq 6$
Length-to-depth ratio	$\frac{L}{D} \leq 15$
Length-to-draft ratio	$\frac{L}{T} \leq 19$
Froude number	$Fn \leq 0.32$
Deadweight	$25{,}000 \leq DW \leq 500{,}000$
Empirical constraint on T and DW	$T - 0.45DW^{0.31} \leq 0$
Empirical constraint on T and D	$T - 0.7D - 0.7 \leq 0$
Empirical constraint for stability	$0.07B - 0.53T - \frac{(0.085C_B - 0.002)B^2}{TC_B} + 1 + 0.52D \leq 0$

Calculations for Model:	
Steel Weight	$W_S = 0.034L^{1.7}B^{0.7}D^{0.4}C_B^{0.5}$
Outfit Weight	$W_O = L^{0.8}B^{0.6}D^{0.3}C_B^{0.1}$
Coefficient for P Calculation	$a = 4977.06C_B^2 - 8105.61C_B + 4456.51$
Coefficient for P Calculation	$b = -10847.2C_B^2 + 12817C_B - 6960.32$
Froude Number	$Fn = \frac{0.5144V_k}{\sqrt{9.8065L}}$
Displacement	$\Delta = 1.025LBTC_B$
Power	$P = \frac{\Delta^{2/3}V_k^3}{a + b \cdot Fn}$
Machinery Weight	$W_M = 0.17P^{0.9}$
Deadweight	$DW = \Delta - LS$
Daily Fuel Consumption	$DC = 0.2 + \frac{0.19 \times 24}{1000}P$
Sea Days	$D_S = \frac{5000}{24V_k}$
Fuel Carried	$FC = DC(D_S + 5)$
Crew, Stores, and Water	$CSW = 2DW^{0.5}$
Cargo Deadweight	$DW_C = DW - FC - CSW$
Port Days	$D_P = 2\frac{DW_C}{8000} + 1$
Round Trips per Year	$RTPA = \frac{350}{D_S + D_P}$
Ship Cost	$C_S = 1.3(2000W_S^{0.85} + 3500W_O + 2400P^{0.8})$
Capital	$C = 0.2C_S$
Running Cost	$C_R = 40000DW^{0.3}$
Voyage Cost	$C_V = 1.05DC \times D_S \times 100 + 6.3DW^{0.8}$
Annual Cost	$C_A = C + C_R + C_V \times RTPA$

4.2.2 MDO Without Uncertainty

The MDO algorithm presented in Chapter 3 was applied to the bulk carrier design problem given in Equation (4.19). When the model is defined as in Table 4.1, there is no uncertainty in the design variables and parameters; solution of this MDO problem gives a baseline for comparison to other results. The results are listed in Table 4.2 in the column labeled "Deterministic Optimum."

Table 4.2. Summary of results from multiple MDO analyses.

	Deterministic Optimum	Reliable Optimum	Reliable & Robust Optimum
Transportation Cost	8.424	8.635	8.838
Annual Cargo	506,320	550,140	566,690
Lightship Weight	8,198	8,803	8,462
L	188.90	193.86	182.61
B	31.485	32.310	30.435
D	15.729	15.364	15.729
T	11.710	11.455	11.710
C_B	0.630	0.657	0.750
V_k	14.000	15.027	15.027

Because the purpose of the reliability analysis is to study the effect of uncertainty on the constraints, the properties of the constraints were evaluated at the deterministic optimum. The results are listed in Table 4.3. The table shows that all of the constraints are satisfied at the deterministic optimum (as expected), and also that several constraints are active: the minimum length-to-beam ratio, the deadweight constraint based on draft, and the draft-depth relation.

Table 4.3. Constraint evaluation using deterministic design optimization.

Constraint	Constraint Value
$\frac{L}{B} \geq 6$	6.0000
$\frac{L}{D} \leq 15$	12.010
$\frac{L}{T} \leq 19$	16.131
$Fn \leq 0.32$	0.167
$25{,}000 \leq DW \leq 500{,}000$	36,775
$T - 0.45DW^{0.31} \leq 0$	6.052×10^{-4}
$T - 0.7D - 0.7 \leq 0$	2.243×10^{-14}
$0.07B - 0.53T - \frac{(0.085C_B - 0.002)B^2}{TC_B} + 1 + 0.52D \leq 0$	-1.751

4.2.3 MDO with Reliability-Based Design

The next step in the bulk carrier optimization was to introduce uncertainty. Uncertainty was included in the bulk carrier model in two ways: uncertainty in a design variable and uncertainty in a parameter. Uncertainty in a design variable was introduced for the ship speed. For a selected design ship speed V_{design}, the actual ship speed follows a normal distribution about V_{design}; that is,

$$V_{actual} = V_{design} + v \tag{4.20}$$

where V_{actual} is the actual ship speed and v is a normally distributed random variable with zero mean. The uncertainty in the ship speed was defined in this manner to agree with the form of Equation (4.14). The physical situation that Equation (4.20) describes is the variance v from the intended ship speed V_{design} due to sea conditions or other factors. For this analysis, v follows a normal distribution with zero mean and standard deviation of 0.5 knots.

Next, uncertainty in a parameter was introduced in the model. The model defined by Sen and Yang (1998) as given in Table 4.1 uses the following regression equation to calculate the steel weight W_S of the ship:

$$W_S = 0.034L^{1.7}B^{0.7}D^{0.4}C_B{}^{0.5} \tag{4.21}$$

To apply uncertainty in a parameter to this model, the exponent on the ship length L is considered to be a random parameter, ε:

$$W_S = 0.034L^{\varepsilon}B^{0.7}D^{0.4}C_B{}^{0.5} \tag{4.22}$$

For this analysis, the parameter ε follows a normal distribution with mean 1.7 and standard deviation 0.05.

There are five total constraints affected by these uncertainties. The simplest constraints are the upper and lower limits on the ship speed. The other constraints affected by the uncertainties (repeated from Table 4.1) are the constraints on Froude number Fn and deadweight DW:

$$Fn \leq 0.32 \tag{4.23}$$

$$25{,}000 \leq DW \leq 500{,}000 \tag{4.24}$$

$$T - 0.45DW^{0.31} \leq 0 \tag{4.25}$$

The multidisciplinary optimization was run including the uncertainty in V_k and ε and the results are included in Table 4.2. A 98% reliability level was prescribed for the probabilistic constraints.

It is immediately clear that the performance in all three objectives at this selected optimum is inferior to the performance at the deterministic optimum. This is expected because a safety margin has been introduced in order to satisfy the constraints in the presence of uncertainty. The constraints were evaluated at the optimum point and the values are shown in Table 4.4; the constraints that are affected by uncertainty are indicated with a check mark in the column labeled "Uncertainty." Two constraints are active: the constraint on the minimum length-to-beam ratio and the draft and depth relation. The constraints which are affected by uncertainty are not active; the constraint on the Froude number and the bounds on the deadweight were not active in the deterministic solution so it is not surprising that they are not active in the reliable solution. However, the constraint on the relationship between the draft and deadweight was active in the deterministic solution but now has an appropriate margin to account for the variability in the ship speed V_k and exponent ε.

Table 4.4. Constraint evaluation using reliability-based design optimization.

Constraint	Constraint Value	Uncertainty
$\frac{L}{B} \geq 6$	6.000	
$\frac{L}{D} \leq 15$	12.618	
$\frac{L}{T} \leq 19$	16.924	
$Fn \leq 0.32$	0.177	✓
$25{,}000 \leq DW \leq 500{,}000$	39,435	✓
$T - 0.45DW^{0.31} \leq 0$	-0.511	✓
$T - 0.7D - 0.7 \leq 0$	-2.020×10^{-14}	
$0.07B - 0.53T - \frac{(0.085C_B - 0.002)B^2}{TC_B} + 1 + 0.52D \leq 0$	-2.289	

4.2.4 MDO with Reliability and Robust Design

The final step in the bulk carrier optimization was to include robustness. Uncertainty was considered in the design variable V_k and the parameter ε as described in Section 4.2.3.

The robust objective function from Equation (4.18) was scaled to define the robust objective function for each objective function f in the bulk carrier example:

$$R_f(L,B,D,T,C_B,V_k)=a\frac{\mu_f(L,B,D,T,C_B,V_k)}{f_0}+(1-a)\frac{\sigma_f(L,B,D,T,C_B,V_k)}{\sigma_0} \quad (4.26)$$

The mean μ_f and standard deviation σ_f are evaluated at the design point (L, B, D, T, C_B, V_k). The variance is scaled by σ_0, the value of the variance at the initial point. The mean is scaled by f_0, the value of the objective function f at the initial point (not the mean value at the initial point) to allow comparison between the robust optimization results and the other results.

To calculate the necessary mean and standard deviation, Equations (4.15) and (4.16) are used with the two random variables v and ε. Then the equations become:

$$\mu(L,B,D,T,C_B,V_k)=\iint f(L,B,D,T,C_B,V_k+v,\varepsilon)\,p(v,\varepsilon)\,dv\,d\varepsilon \quad (4.27)$$

$$\sigma^2(L,B,D,T,C_B,V_k)= \quad (4.28)$$
$$\iint f(L,B,D,T,C_B,V_k+v,\varepsilon)^2\,p(v,\varepsilon)\,dv\,d\varepsilon-\mu(L,B,D,T,C_B,V_k)^2$$

where f is any of the objective functions and $p(v, \varepsilon)$ is the joint PDF for v and ε. The ship speed variation and the exponent are assumed to be independent, so the equations become

$$\mu(L,B,D,T,C_B,V_k)=\iint f(L,B,D,T,C_B,V_k+v,\varepsilon)\,p_v(v)\,p_\varepsilon(\varepsilon)\,dv\,d\varepsilon \quad (4.29)$$

$$\sigma^2(L,B,D,T,C_B,V_k)= \quad (4.30)$$
$$\iint f(L,B,D,T,C_B,V_k+v,\varepsilon)^2\,p_v(v)\,p_\varepsilon(\varepsilon)\,dv\,d\varepsilon-\mu(L,B,D,T,C_B,V_k)^2$$

where $p_v(v)$ and $p_\varepsilon(\varepsilon)$ are the PDFs for v and ε, respectively.

The integrals are evaluated numerically during the optimization. Each PDF is integrated over $\pm 4\sigma$ to cover approximately 99.99% of the area under the normal curve. The numerical integration is performed using 200 points in each random variable, or 40,000 points total over the integrand. The effect of increasing or decreasing the number of points was investigated and it was found that 200 points led to a satisfactory compromise of computation time and accuracy. Additionally, the results of the integration were compared to a simple Monte Carlo simulation, which randomly placed 1000 points in the design space according to the distributions of v and ε. The mean and variance resulting from the Monte Carlo simulations confirmed the results of the numerical integration.

The robust and reliable optimization was performed using the parameter value a = 0.5 and the results are included in Table 4.2. The constraints at the robust and reliable optimum were evaluated and are included in Table 4.5. All of the constraints are satisfied, with two of the constraints active: the constraint on the length-to-beam ratio and the empirical constraint on the draft and depth. The safety margins introduced in the constraints are comparable to the margins in the probabilistic constraints of the MDO reliability analysis. However, the margins differ because in the current case the optimum has shifted due to the robust considerations.

Table 4.5. Constraint evaluation using reliable and robust optimization.

Constraint	Constraint Value	Uncertainty
$\frac{L}{B} \geq 6$	6.000	
$\frac{L}{D} \leq 15$	11.610	
$\frac{L}{T} \leq 19$	15.595	
$Fn \leq 0.32$	0.183	✓
$25{,}000 \leq DW \leq 500{,}000$	41,569	✓
$T - 0.45 DW^{0.31} \leq 0$	-0.453	✓
$T - 0.7D - 0.7 \leq 0$	2.243×10^{-14}	
$0.07B - 0.53T - \frac{(0.085C_B - 0.002)B^2}{TC_B} + 1 + 0.52D \leq 0$	-1.410	

To study the effects of the robust optimization on the optimum design, the statistics at the three different optima were computed; Table 4.6 summarizes the statistics for the three optimization approaches. The deterministic optimum shows the best performance for the transportation cost (the top level objective) and the lightship weight. However, for the deterministic case, without the consideration of uncertainty in the constraints, it is unlikely that the constraints would be satisfied when considering the uncertainty of the ship speed V_k and exponent ε. For the robust and reliable case, the standard deviation has been reduced from the reliable case for all three of the objective functions.

Table 4.6. Statistics for MDO results in the bulk carrier example.

	Deterministic Optimum	Reliable Optimum	Robust & Reliable Optimum (a = 0.5)
Transportation Cost	8.4240	8.6351	8.8380
Transportation Cost Mean	8.5401	8.7523	8.9472
Transportation Cost Standard Deviation	0.7174	0.7241	0.6792
Annual Cargo	506,320	550,140	566,690
Annual Cargo Mean	503,650	547,300	564,160
Annual Cargo Standard Deviation	20,504	21,022	19,266
Lightship Weight	8,198	8,803	8,462
Lightship Weight Mean	8,434	9,137	8,698
Lightship Weight Standard Deviation	1,858	2,012	1,857

4.2.5 Optimization Under Uncertainty Results

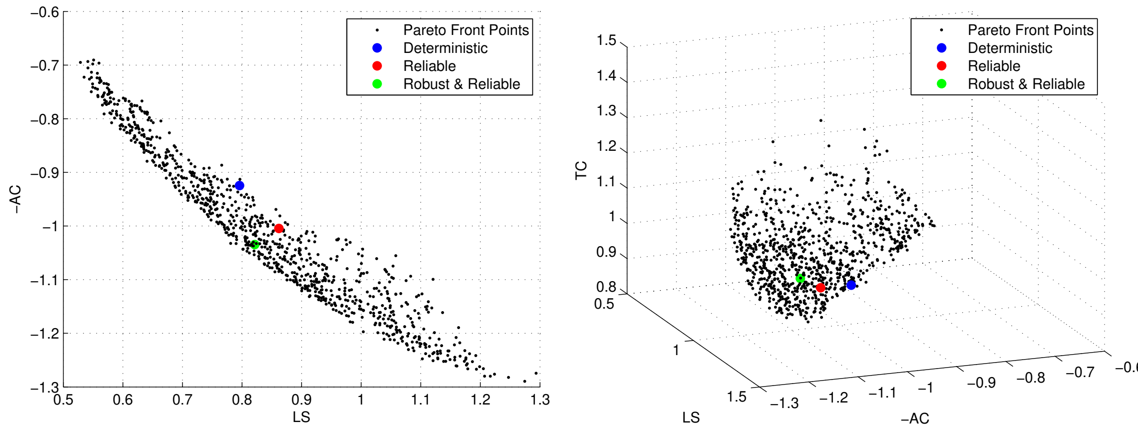
The bulk carrier design problem was first solved using deterministic multidisciplinary design optimization; this solution does not consider any sources of uncertainty. The resulting solution was (L, B, D, T, C_B, V_k) = (188.90, 31.485, 15.729, 11.710, 0.630, 14.000). However, if the ship speed V_k and parameter ε include uncertainty with the properties given in Section 4.2.3, a simple Monte Carlo simulation can be performed to generate random realizations of the ship design about the deterministic optimum design. For a Monte Carlo simulation of 10,000 randomly generated points about the deterministic optimum, only 2,424 satisfied all of the constraints.

To illustrate the effect of selecting a reliable optimum, the reliable design optimum design was (L, B, D, T, C_B, V_k) = (193.86, 32.310, 15.364, 11.455, 0.657, 15.027). When a Monte Carlo simulation was performed to generate random realizations about the reliable optimum design, 9,574 out of 10,000 randomly generated points satisfied all of the constraints. This is a very significant improvement over the deterministic case.

For the reliable and robust optimum design (L, B, D, T, C_B, V_k) = (182.61, 30.435, 15.729, 11.710, 0.750, 15.027), 9,595 out of 10,000 randomly generated points satisfied all of the constraints in the presence of uncertainty. The results for the reliable optimum and the robust and reliable optimum are reasonable for the 98% reliability level used in the analysis.

To visualize the results, the values of the objective functions at the optima can be plotted. A simple Monte Carlo technique was used to approximate the Pareto front for this problem (while inefficient, this method was selected for its simplicity because the purpose of this work was not to find the Pareto front). The optima are shown in two plots in Figure 4.3: the deterministic optimum is shown in blue, the reliable optimum is shown in red, and the robust and reliable optimum is shown in green.

In Figure 4.3, the objective functions are scaled so that all of the objective functions have a magnitude of one at the initial point (L, B, D, T, C_B, V_k) = (195, 32.31, 20, 10.5, 0.7, 16). Figure 4.3(a) shows the scaled values of all three objective functions, and Figure 4.3(b) shows the scaled annual cargo versus the scaled transportation cost. Figure 4.3 demonstrates that the MDO algorithm selects ship designs that perform relatively well because the optima lie on or near the Pareto front. Additionally, the algorithm seeks to reduce the top level objective as much as possible (transportation cost) while the discipline level objectives are not as important.



(a) (b)

Fig. 4.3. Approximate Pareto front with optima for the MDO of the bulk carrier concept design: (a) Plot showing all three objectives' axes, (b) Plot with two-dimensional projected view.

4.3 Chapter Summary

In this chapter, two methods for accounting for uncertainty in multidisciplinary design optimization were integrated in the MDO algorithm with target values: reliability-based design optimization and robust optimization. The purpose of reliability-based design optimization is to ensure that realizations of the optimum satisfy the constraints with a prescribed probability. The purpose of robust design optimization is to ensure that the performance of objective function at the optimum does not deteriorate significantly. Reliability-based design optimization and robust optimization were applied to an MDO problem for the concept design of a bulk carrier; the problem was solved using the multidisciplinary design optimization algorithm presented in Chapter 3.

This research demonstrated that techniques for accounting for uncertainty in design optimization can be applied to the MDO algorithm presented in Chapter 3, and that the MDO algorithm with uncertainty can be effectively utilized in a ship design problem.

CHAPTER 5

Surrogate Models in Multidisciplinary Design Optimization

Functions that are computationally expensive to evaluate present a challenge in design optimization, because the iterative and often complex process of optimization requires many function calls. As functions become more expensive, the cost of the optimization increases, requiring increasingly long computation times. The problem of the high cost of optimization is common in complex engineering analysis codes such as computational fluid dynamics (CFD) or finite element analysis (FEA), where a single evaluation may take hours or even days. The situation becomes even more challenging for multidisciplinary design optimization, where multiple discipline optimization problems must be solved repeatedly.

Surrogate models (also called metamodels or response surfaces) are mathematical interpolation models that approximate the behavior of expensive functions with acceptable accuracy at a reduced computational cost. Surrogate models are developed from a limited number of function evaluations at specific sample points; once surrogate models are available they can be used to replace the true functions during the optimization process.

This chapter begins with the mathematical background for several commonly-used surrogate modeling techniques, including Kriging, the technique used in this research. A discussion on the use of surrogate models in optimization from the literature follows. Finally, the multidisciplinary optimization of a ship's hull form is performed; the hull form is evaluated with computationally expensive solvers, so surrogate models are developed and utilized instead of the actual solvers during the optimization.

5.1 Surrogate Models

Surrogate models are useful in cases where the function of interest is not explicitly known (a "black box") or where the function is very expensive to evaluate. The goal of the surrogate model is to define an expression for the function that is relatively simple to evaluate, but still adequately accurate. In this section, brief mathematical background information on several of the most commonly used types of surrogate models is reviewed. Before presenting mathematical details, the notation used in surrogate modeling techniques is defined in this section.

Forrester, Sóbester, and Keane (2008) provide a guide on the procedure of constructing surrogate models. The first step in creating a surrogate model is to select the variables that will be used as inputs; define the vector of input variables as $\mathbf{x}$, which has k elements. Let f be the true function which determines the performance of interest as a function of $\mathbf{x}$: $f(\mathbf{x})$.

The second step is to select a set of sample points; this process is called design of experiments. The task of selecting appropriate sample points is not trivial, however, design of experiments will not be covered in this dissertation; see Kleijnen (1998) for an introduction to design of experiments. Let $\mathbf{x}^{(i)}$ be the input values selected for sample point i. Then the true function is evaluated at each sample point: $y^{(i)} = f(\mathbf{x}^{(i)})$.

The next step is to select a surrogate modeling technique. Each surrogate modeling technique uses parameters to describe the behavior of the function f, and values for the parameters can be determined based on the sample data $\{\mathbf{x}^{(i)}, y^{(i)}\}$. Once the surrogate model has been built, the surrogate model for $f(\mathbf{x})$ is denoted $\hat{f}(\mathbf{x})$.

The notation for surrogate models used in this chapter is summarized in Table 5.1.

Table 5.1. Summary of notation for surrogate models.

k	number of input variables
x_i	(scalar) input variable with index $i = 1, \ldots, k$
$\mathbf{x}$	vector of input variables, $\mathbf{x} \in \Re^k$
n	number of sample points
$\mathbf{x}^{(i)}$	input variables for a sample point with index i, $i = 1, \ldots, n$
$y^{(i)}$	value of the true function at sample point $\mathbf{x}^{(i)}$
$\mathbf{y}$	vector of the true function values at sample points $y^{(i)}$, $\mathbf{y} \in \Re^n$
f	true function
$\hat{f}$	surrogate model for function f

The remaining portion of Section 5.1 provides an introduction to several popular surrogate modeling techniques. This brief mathematical introduction begins with the simplest surrogate modeling technique and then describes techniques with increasing complexity. The final technique is Kriging which is presented in detail because it is utilized later in this work. The purpose of providing information on other surrogate modeling techniques is to gradually build up from the simpler techniques to the more complex, and to provide a basis for comparison to the Kriging method.

5.1.1 Linear Least Squares (Polynomial) Regression

A polynomial model can be formulated in the following general form:

$$\hat{f}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_i \sum_j \beta_{ij} x_i x_j + \cdots \tag{5.1}$$

where β_i are model-fitting parameters. The expression in Equation (5.1) is quadratic in the input variables x_i, but it may be expanded to include higher order terms. For the least squares regression, $\boldsymbol{\beta}$ is an $(m + 1)\times 1$ vector that contains all of the parameters in Equation (5.1), where $m + 1$ is the number of coefficient parameters.

Let $\mathbf{X}$ be the $n\times(m + 1)$ matrix that contains the design variable information corresponding to Equation (5.1), with one row per sample point; that is, each row i is of the form:

$$[1 \;\; x_1^{(i)} \ldots x_k^{(i)} \;\; (x_1^{(i)})^2 \ldots (x_k^{(i)})^2 \;\; x_1^{(i)} x_2^{(i)} \ldots] \tag{5.2}$$

Then the least squares estimate $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ is given by Kleijnen (2009)

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{5.3}$$

The estimate $\hat{\boldsymbol{\beta}}$ can be used in place of $\boldsymbol{\beta}$ in Equation (5.1) to estimate the response at a new point $\mathbf{x}$.

The least squares estimate determines the values of $\hat{\boldsymbol{\beta}}$ to minimize the sum of the squares of the error between the true sample point responses $y^{(i)}$ and the predicted responses $\hat{f}(\mathbf{x}^{(i)})$ (Papalambros and Wilde 2000).

The least squares regression does not require that the regression equation be a polynomial in $\mathbf{x}$, even though polynomials are the most common. The only requirement for Equation (5.3) to hold is that the expression in Equation (5.1) must be linear in $\boldsymbol{\beta}$; one could, for example, have terms of the form $\beta_i \ln(x_i)$.

5.1.2 Neural Networks

Neural networks utilize a network of simple functions called nodes to build a surrogate model. One common model for node behavior is a logsig sigmoid function (Papalambros and Wilde 2000):

$$\operatorname{logsig}(z_1, \dots, z_m) = \frac{1}{1+\exp\left(b - \sum w_i z_i\right)} \tag{5.4}$$

where b is called the bias for the node and w_i are weights for the inputs z_i.

The nodes are arranged in the network in layers, usually three: the input layer, a hidden layer, and the output layer (Chen et al. 2006). The input layer node(s) take the input variables $\mathbf{x}$ as inputs. The outputs from the input layer nodes are used as input for the hidden layer nodes. Finally, the output from the hidden layer is input to the output layer node(s), and the output layer returns the approximation $\hat{f}(\mathbf{x})$. Values for the parameters (biases b and weights w_i) can be determined by minimizing the sum of the squares of the error in the prediction at the sample points (Papalambros and Wilde 2000).

5.1.3 Radial Basis Functions

The radial basis function model takes the form (Chen et al. 2006; Forrester, Sóbester, and Keane 2008):

$$\hat{f}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{m} \beta_i b\left(\left\|\mathbf{x} - \mathbf{c}^{(i)}\right\|\right) \tag{5.5}$$

where β_i are coefficients, b is a basis function, and $\mathbf{c}^{(i)}$ are centers for the basis function. The centers $\mathbf{c}^{(i)}$ can be chosen as the sample points: $\mathbf{c}^{(i)} = \mathbf{x}^{(i)}$. The most common choices for the basis function b are

- linear: $b(z) = z$

- cubic: $b(z) = z^3$
- thin plate spline: $b(z) = z^3 \ln(z)$
- Gaussian: $b(z) = \exp(-z^2/(2\sigma^2))$
- multi-quadratic: $b(z) = (z^2 + \sigma^2)^{1/2}$
- inverse multi-quadratic: $b(z) = (z^2 + \sigma^2)^{-1/2}$

The last three basis functions listed include an additional parameter σ for improved model-fitting.

Once a basis function has been selected, values for the coefficients β_i can be determined by minimizing the sum of the squares of the error of the prediction at the sample points (Forrester, Sóbester, and Keane 2008).

5.1.4 Kriging

The Kriging method is employed in this work for constructing surrogate models. A number of reviews (Chen et al. 2006; Kleijnen 2009; Simpson et al. 1998; Simpson et al. 2001; Wang and Shan 2007; Zhao and Xue 2010) have compared Kriging to other surrogate modeling techniques, including linear regression, MARS, neural networks, and radial basis functions. In general, the reviews found that while Kriging models can be complex to build and evaluate, Kriging models provide accurate predictions and are very flexible, that is, they can model a wide range of function behavior. Another advantage of the Kriging method is that Kriging models return the true value at sample points (Kriging is an exact interpolator); this is the preferred behavior for surrogate models for deterministic functions.

The mathematical approach for Kriging shown here is based on Sacks et al. (1989) and Martin and Simpson (2005). The fundamental idea of Kriging is to model the response of the function as the sum of a simple regression model and a stochastic process.

The Kriging model $\hat{f}(\mathbf{x})$ for the true response $f(\mathbf{x})$ is

$$\hat{f}(\mathbf{x})=\sum_{i=1}^{m}\beta_i b_i(\mathbf{x})+Z(\mathbf{x})=\mathbf{b}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}+Z(\mathbf{x}) \tag{5.6}$$

where the summation term is a regression model using regression functions b_i. $Z(\mathbf{x})$ represents a stochastic process which is assumed to have zero mean and a covariance given by

$$\mathrm{Cov}(Z(\mathbf{x}), Z(\mathbf{w})) = \sigma^2 R(\mathbf{x}, \mathbf{w}) \tag{5.7}$$

where σ^2 is the process variance. $R(\mathbf{x}, \mathbf{w})$ is the spatial correlation function for two points $\mathbf{x}$ and $\mathbf{w}$; the spatial correlation function describes the smoothness of the model and controls the influence that nearby points have on each other.

One of the most common choices for the spatial correlation function is the Gaussian spatial correlation function (used in this work):

$$R(\mathbf{x},\mathbf{w})=\prod_{i=1}^{k}\exp(-\theta_i|x_i-w_i|^2) \tag{5.8}$$

where x_i and w_i indicate the i^{th} elements of the vectors $\mathbf{x}$ and $\mathbf{w}$, respectively. The parameter $\boldsymbol{\theta}$ affects the influence points have on other nearby points.

The next step in creating a Kriging model is to train the model using sample data. Let $\mathbf{x}^{(i)}$, $i = 1, \ldots, n$, be the input for the sample points, and let $\mathbf{y}$ be the vector of length n that contains the true values of $f(\mathbf{x}^{(i)})$ for the sample points. Define the $n\times m$ matrix $\mathbf{F}$ as the regression functions $\mathbf{b}$ evaluated at the sample points:

$$\mathbf{F}=\begin{pmatrix}\mathbf{b}(\mathbf{x}_1)^{\mathrm{T}}\\ \vdots\\ \mathbf{b}(\mathbf{x}_n)^{\mathrm{T}}\end{pmatrix} \tag{5.9}$$

Define $\mathbf{R}$ as the $n\times n$ matrix of correlations for the sample points, $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

Next, the parameters for the model can be determined. The generalized least-squares estimate of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{F}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{F}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{y}\right) \tag{5.10}$$

and the maximum likelihood estimate of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n} \left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}\right) \tag{5.11}$$

This work uses ordinary Kriging, a common special case of Kriging, where the regression functions $\mathbf{b}(\mathbf{x})$ are taken to be a constant term. Then only one value of θ is necessary to fit the model, and the maximum likelihood estimate of θ is given by the maximization of the following quantity with respect to θ (Sasena 2002):

$$-\frac{1}{2}\left(n \ln\left(\hat{\sigma}^2\right) + \ln\left(\det \mathbf{R}\right)\right) \tag{5.12}$$

The final step is to use the Kriging model to predict the response at a new point, $\mathbf{x}$. Define the vector $\mathbf{r}(\mathbf{x})$ of length n as the correlation between the new point and the sample points:

$$\mathbf{r}(\mathbf{x}) = [R(\mathbf{x}^{(1)}, \mathbf{x}) \ldots R(\mathbf{x}^{(n)}, \mathbf{x})]^{\mathrm{T}} \tag{5.13}$$

The best linear unbiased predictor (BLUP) for the performance at the new point $\mathbf{x}$ is given by

$$\hat{f}(\mathbf{x}) = \mathbf{b}^{\mathrm{T}}(\mathbf{x})\hat{\boldsymbol{\beta}} + \mathbf{r}^{\mathrm{T}}(\mathbf{x})\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}\right) \tag{5.14}$$

5.2 Surrogate Models in Optimization

Surrogate models can be a very powerful tool in optimization. Forrester, Sóbester, and Keane (2008) provide a readable, friendly introduction to the important concepts in surrogate-based optimization. In particular, the authors point out that surrogate models are useful not only for approximating the objective function in an optimization problem, but also for constraints. Queipo et al. (2005) highlight the straightforward nature of this approach: once surrogate models are constructed, the optimization problem can be solved with any popular optimization technique by simply replacing the functions with their surrogates.

Wang and Shan (2007) discuss three different strategies for the use of surrogate models in optimization: sequential, adaptive, and direct sampling. In the sequential approach, sample points are selected and the surrogate model is constructed before beginning the optimization. In the adaptive approach, validation (and possibly some optimization) is included during construction of the surrogate model; the validation and/or optimization are used to determine new sample points which update the surrogate model iteratively. In the direct sampling approach, the surrogate model is used only to determine new sample points while moving toward the optimum (optimization is based on the sample point evaluations not the surrogate model predictions). In this work, the traditional sequential approach is used.

Three very thorough review articles (Forrester and Keane 2009; Jones 2001; Queipo et al. 2005) provide further technical information on sampling schemes, popular surrogate modeling techniques, and evaluation of surrogate model accuracy for the support of surrogate-based optimization.

Techniques for the use of surrogate models in design optimization are also applicable in MDO. Surrogate models are valuable in MDO because MDO problems include more functions to evaluate, giving a greater potential for computational cost and time savings. Wang and Shan (2007) highlight the value of surrogate models for MDO, because MDO typically uses a large number of design variables, expensive function evaluations, and coupling between disciplines.

Many specific technical examples of the application of surrogate models in optimization are available in the literature, which will be discussed in this section. The use of surrogate-based optimization in a wide range of fields demonstrates the versatility of the approach.

Wan (2004) applies surrogate-based optimization to supply chain management (in particular, management of inventory); Wan uses surrogate models that are updated iteratively during the optimization process to provide better local estimation. Glaz, Friedmann, and Liu (2008) use polynomial regression, radial basis functions, and Kriging surrogate models to replace an aeroelastic analysis for evaluating vibrations in the optimization of a helicopter blade. Song et al. (2010) use response surfaces models and Kriging surrogate models to replace high-fidelity finite element methods analyses for the optimization of an automotive suspension component.

Surrogate-based optimization has also been used in multidisciplinary design optimization. Jouhaud et al. (2007) use the Kriging method for a multidisciplinary optimization of a foil geometry with aeroelastic analysis and acoustic analysis. Queipo et al. (2005) study the multiobjective optimization of a liquid rocket injector using polynomial regression models, with the objectives of optimizing performance and life.

5.3 Ship Hull Form Optimization

The multidisciplinary design optimization of a ship hull form, with the use of surrogate models, is presented in this section. The goal of this study is to demonstrate that the MDO framework presented in Chapter 3 can effectively incorporate surrogate models, which can produce accurate predictions and successfully optimize a design.

A container ship hull form was selected for this analysis. This hull form was selected because a full model of the hull geometry was readily available (from the website for the commercial naval architecture software MaxSurf, www.formsys.com/maxsurf). The hull geometry model greatly facilitated the computational fluid dynamics analysis. The parent hull form had the following principal dimensions: length L = 111.19 m, beam B = 19.5 m, draft T = 7.24 m, and block coefficient C_b = 0.71.

5.3.1 Problem Definition

The hull form is described by three design variables: molded ship length L, length-to-beam ratio L/B, and beam-to-draft ratio B/T. The lower and upper bounds of the design variables are summarized in Table 5.2. The selection of L, L/B, and B/T as design variables offers a practical advantage over the use of the dimensions L, B, and T as design variables: the upper and lower bounds on L/B and B/T enforce the constraints on these defined by Watson (1998).

Table 5.2. Lower and upper bounds of the design variables

Design Variable	Lower Bound	Upper Bound
L (m)	151.957	205.589
L/B	5.00	8.00
B/T	2.25	3.75

All disciplines included two constraints (upper and lower bound) on the block coefficient C_b to ensure that the geometry remained reasonable. The displacement Δ of the ship was set constant and equal to 38,500 tonnes; then the block coefficient was computed from the constant displacement requirement:

$$C_b = \frac{\Delta}{\gamma LBT} \tag{5.15}$$

where γ is the weight density of sea water.

The average (mean) value of the block coefficient C_{bm} can be estimated based on the operational speed requirement from Watson (1998):

$$C_{bm} = 0.70 + 0.125 \tan^{-1}\left(\frac{23 - 100\,Fn}{4}\right) \tag{5.16}$$

where Fn is the Froude number, defined as

$$Fn = \frac{V}{\sqrt{gL}} \tag{5.17}$$

V is the ship speed (in m/s); in this analysis, the ship speed was held constant at 19 knots.

Watson (1998) recommends an allowable variation margin of 0.025 from the mean for the block coefficient. To ensure compatibility for the block coefficient values in Equation (5.16), two constraints were specified in all of the optimization statements:

$$C_{bm} - 0.025 \le C_b \le C_{bm} + 0.025 \tag{5.18}$$

A very simple cost regression model from Parsons (2003) was employed to define the top level objective function. The cost model considered the effects of the main principal dimensions on capital cost, and is given by

$$\text{cost} = 100\left(1.03\frac{L}{L_0} + 0.78\frac{B}{B_0} + 0.18\frac{C_b}{C_{b0}}\right) \tag{5.19}$$

where L_0, B_0, and C_{b0} are the initial values for the length, beam, and block coefficient.

Three disciplines were studied in the multidisciplinary design optimization: seakeeping, maneuvering, and resistance.

5.3.2 Seakeeping Discipline Analysis

The seakeeping analysis utilizes the Seakeeping Prediction Program (SPP) (Parsons, Li, and Singer 1998), which is an implementation of the SCORES Program (Raff 1972). This section presents an overview of the analysis performed by SPP.

In SPP, strip theory is used to approximate the hull's inertia properties and the loading from waves. The root-mean-square (RMS) responses of the ship due to the prescribed wave conditions are evaluated; this analysis focuses on the RMS response in heave and pitch.

The equations of motion for heave z and pitch θ are given by

$$\begin{aligned} m\ddot{z} &= \int_{x_s}^{x_b} \frac{dZ}{dx}\,dx + Z_w \\ I_Y\ddot{\theta} &= -\int_{x_s}^{x_b} \frac{dZ}{dx}\,x\,dx + M_w \end{aligned} \tag{5.20}$$

where m is the mass of the ship, I_Y is the mass moment of inertia of the ship about the y axis, dZ/dx is the local sectional vertical force on the ship, and Z_w and M_w are the wave excitation force and moment on the ship. The limits x_s and x_b indicate integration from the stern to the bow.

The wave excitation is given by the integrals

$$
\begin{aligned}
Z_w &= \int_{x_s}^{x_b} \frac{dZ_w}{dz} dx \\
M_w &= -\int_{x_s}^{x_b} \frac{dZ_w}{dz} x\, dx
\end{aligned}
\tag{5.21}
$$

where

$$
\frac{dZ_w}{dx} = -\left[\rho\, g\, B^* \eta + \left(N_z{}' - V \frac{dA_{33}{}'}{dx} \right) \dot{\eta} + A_{33} \ddot{\eta} \right] e^{-k\bar{h}} \tag{5.22}
$$

B^* is the waterline beam, η is the surface wave elevation, $N_z{}'$ is the local sectional vertical damping force coefficient, V is the ship speed, $A_{33}{}'$ is the local sectional vertical added mass, k is the wave number, and $\bar{h}$ is the mean section draft.

The solution of the differential equations for heave and pitch (Equation 5.20) yield solutions of the following form:

$$
\begin{aligned}
z &= z_0 \sin(\omega_e t + \delta) \\
\theta &= \theta_0 \sin(\omega_e t + \delta)
\end{aligned}
\tag{5.23}
$$

where ω_e is the encounter frequency. The RMS responses for heave and pitch are determined from Equation 5.23, and SPP evaluates and returns these values.

The objective of the seakeeping discipline was to minimize the maximum vertical movement of the bow when the pitch and heave motions are synchronized. The maximum bow movement $M_{\max}$ is defined as

$$
M_{\max} = \max(\text{RMS heave}) + \frac{L}{2} \max(\text{RMS pitch}) \tag{5.24}
$$

SPP was used to evaluate the pitch and heave response of the ship in Sea State 4 using the ISSC spectrum. The heading angle was varied from 0º to 180º to estimate the heading where the worst seakeeping performance was encountered.

Two additional constraints were introduced in the seakeeping discipline: the RMS heave amplitude must be less than 0.56 m and the RMS pitch amplitude must be less than 0.55º.

5.3.3 Maneuvering Discipline Analysis

The maneuvering analysis utilizes the Maneuvering Prediction Program (MPP) (Parsons, Li, and Singer 1998); calculations in MPP are based on the formulation from Clarke, Gedling, and Hine (1982). This section presents mathematical background information on the maneuvering calculations performed by MPP.

The maneuvering analysis treats the ship as a rigid body with only three degrees of freedom: surge, sway, and yaw. Then the equations of motion can be written

$$\begin{aligned} X &= m\left(\dot{u} - rv - x_G r^2\right) \\ Y &= m\left(\dot{v} + ur + x_G \dot{r}\right) \\ N &= I_Z \dot{r} + m x_G\left(\dot{v} + ru\right) \end{aligned} \tag{5.25}$$

where u and v are the longitudinal and lateral velocities, respectively, and r is the yaw rate. m is the mass of the ship, I_z is the moment of inertia about midship, and x_G is the location of the center of gravity (measured from midship). X, Y, and N are the hydrodynamic forces and moments that act on the ship.

For a linear analysis of Equation 5.25, only the terms linear in u, v, r and the corresponding derivatives are considered. Then the hydrodynamic forces can be written

$$\begin{aligned} X &= X_{\dot{u}} \dot{u} + X_u \Delta u \\ Y &= Y_{\dot{v}} \dot{v} + Y_v v + Y_{\dot{r}} \dot{r} + Y_r r \\ N &= N_{\dot{v}} \dot{v} + N_v v + N_{\dot{r}} \dot{r} + N_r r \end{aligned} \tag{5.26}$$

where Δu is the perturbation velocity about a steady speed u_0, and using the notation

$$Y_v = \frac{\partial Y}{\partial v} \tag{5.27}$$

and similarly for the other variables. Then the linear equations of motion become

$$\begin{gathered}(X_{\dot{u}} - m)\dot{u} + X_u \Delta u = 0 \\ (Y_{\dot{v}} - m)\dot{v} + Y_v v + (Y_{\dot{r}} - m x_G)\dot{r} + (Y_r - m u_0) r = 0 \\ (N_{\dot{v}} - m x_G)\dot{v} + N_v v + (N_{\dot{r}} - I_z)\dot{r} + (N_r - m x_G u_0) r = 0\end{gathered} \tag{5.28}$$

where u_0 denotes a steady forward speed and Δu is the perturbation about the steady speed; that is, $u = u_0 + \Delta u$.

The first equation in surge is decoupled from the equations in sway and yaw; however, the sway and yaw equations are coupled. The sway and yaw equations can be decoupled by writing them in the following form:

$$\begin{gathered}T_1{}'T_2{}'\ddot{r}{}' + (T_1{}' + T_2{}')\dot{r}{}' + r' = K'\delta + K'T_3{}'\dot{\delta}{}' \\ T_1{}'T_2{}'\ddot{v}{}' + (T_1{}' + T_2{}')\dot{v}{}' + v' = K_v{}'\delta + K_v{}'T_4{}'\dot{\delta}{}'\end{gathered} \tag{5.29}$$

Equation 5.29 introduces the rudder angle δ; the effect of the rudder is to apply a side force and moment proportional to rudder angle. The prime superscripts indicate that the variables are written in nondimensional form, and values for the variables can be estimated using regression equations given by Clarke, Gedling, and Hine (1982).

Finally, the solution to the decoupled equations of motion becomes:

$$\begin{bmatrix} v' \\ r' \end{bmatrix} = \begin{bmatrix} v_1 \\ r_1 \end{bmatrix} \exp\left(-\frac{t'}{T_1{}'}\right) + \begin{bmatrix} v_2 \\ r_2 \end{bmatrix} \exp\left(-\frac{t'}{T_2{}'}\right) \tag{5.30}$$

where the coefficients v_1, v_2, r_1, and r_2 are constants determined by the initial conditions. The solution of the equations of motion can be used to evaluate the steady turning diameter, advance, and tactical diameter for the ship.

The ship’s maneuverability is also assessed using Clarke's turnability index. The following equation describes the motion due to a rudder angle δ:

$$\frac{\psi(t)}{\delta}=K'\left[1-(T_1'+T_2'-T_3')+\frac{(T_1'-T_3')}{(T_1'-T_2')}T_1'e^{\left(-\frac{1}{T_1'}\right)}-\frac{(T_2'-T_3')}{(T_1'-T_2')}T_2'e^{\left(-\frac{1}{T_2'}\right)}\right] \quad (5.31)$$

where ψ describes the change in heading. Clarke's turnability index P_C is an approximation for $\psi(t)/\delta$ that can be used to evaluate the turning ability of the ship.

Dynamic stability is assessed using the stability criterion C:

$$C = Y_v'(N_r' - m'x_G') - N_v'(Y_r' - m) \quad (5.32)$$

The stability criterion must be positive for the linear system to be stable.

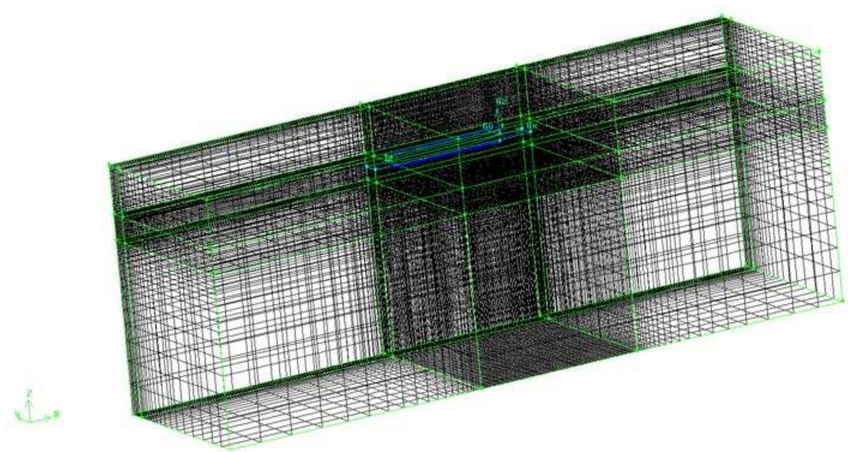
The objective of the maneuvering discipline was to minimize the steady turning diameter of the ship, subject to constraints on advance, tactical diameter, Clarke's turnability index, and the stability criterion. The advance was required to be less than or equal to 4.5 times the ship length. The tactical diameter was required to be less than five ship lengths. Clarke's turnability index must satisfy $P_C \geq 0.3$, and the stability criterion must be positive ($C > 0$).

5.3.4 Resistance Discipline Analysis

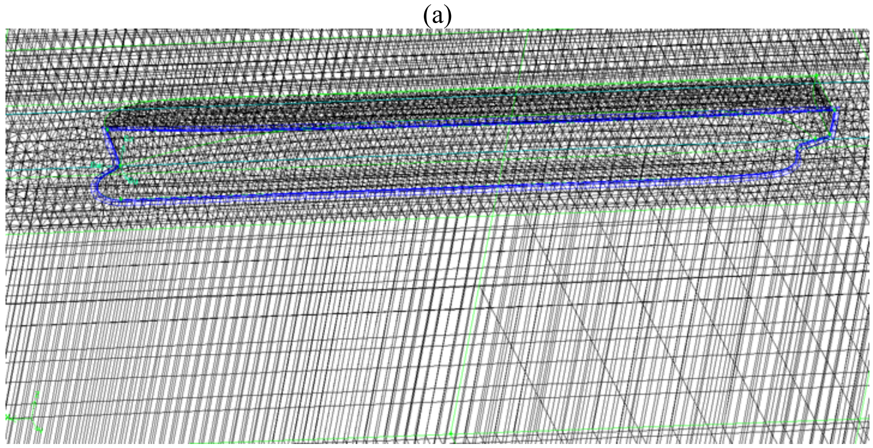
The resistance computations were performed with computational fluid dynamics (CFD) using FLUENT software. The volume of fluid model in FLUENT was selected to solve the multiphase CFD problem, because the CFD model consists of two different phases: water and air. The *k*-omega shear-stress transport model was used to model viscosity.

The CFD mesh was generated by using GAMBIT, the pre-processing software for FLUENT. To improve the numerical stability of the CFD computation, structured hexahedral elements were used for the far field domain, while tetrahedral elements were used for the domain around the hull surface. Boundary layer elements on the hull surface were also modeled in order to account for the viscosity effects. A typical CFD model used in this work had 235,538 nodes and 428,629 elements. Figure 5.1 shows the CFD mesh of the computation domain.

More than 12 hours were required for completing the CFD computation for a single hull form configuration using a Windows machine with a Pentium-4 3.40 GHz CPU and 2 GB 3.39 GHz RAM. This high computational cost illustrates the potential benefit of using surrogate models instead of the CFD solver within the optimization.



(a)



(b)

Fig. 5.1. CFD mesh of the computation domain for resistance calculation;
(a) CFD mesh of full domain, (b) CFD mesh near the hull surface.

The objective of the resistance discipline was to minimize the resistance of the ship, where the resistance included both the wave-making resistance and the frictional resistance. No additional constraints were introduced in the resistance discipline.

The entire ship MDO problem is summarized in Figure 5.2.

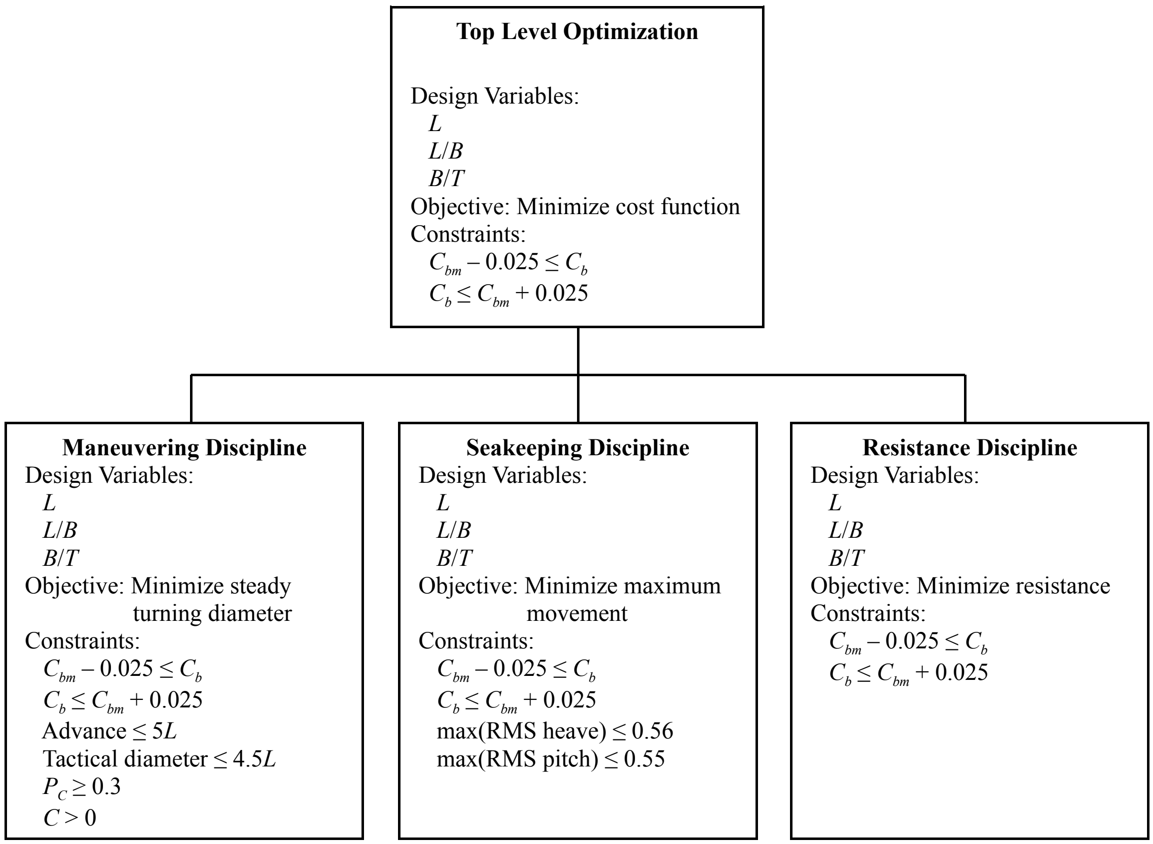


Fig. 5.2. Summary chart of MDO process for the ship hull form optimization.

5.3.5 Construction of Surrogate Models

Surrogate models were used instead of the actual solvers to evaluate objective functions and constraints in all three discipline analyses. A surrogate model was used for the resistance discipline because of the prohibitively long computation time required for the CFD analysis. Surrogate models must be used for the seakeeping and maneuvering disciplines because of the practical limitation that MPP and SPP lack a batch mode; for these programs the data for each sample point must be input manually via the GUI.

For the maneuvering discipline, surrogate models were developed to evaluate the steady turning diameter, advance, tactical diameter, turnability index, and stability criterion. For the seakeeping discipline, surrogate models were developed to evaluate the maximum RMS heave and maximum RMS pitch. For the resistance discipline, a surrogate model was developed to evaluate the resistance. The input parameters to the surrogate models were the three design variables: length L, length to beam ratio L/B, and beam to draft ratio B/T.

The first step in the development of the surrogate models was to evaluate data at a set of sample points. An optimal symmetric Latin hypercube algorithm was used to select the sample points. The seakeeping and maneuvering discipline surrogate models used 34 sample points and the resistance discipline surrogate model used 31 sample points. The number of sample points (34) was selected based on practical time limitations for running the solvers; three of the CFD simulations did not converge in the resistance analysis so only 31 points were used to build the surrogate model for resistance. At each sample point, the actual solvers were used to evaluate the quantities of interest, and this set of data was used to build the surrogate models using the Kriging method.

The accuracy of the surrogate models was tested before using the surrogate models in the optimization analysis. Two additional points were selected randomly; the actual solvers were used to evaluate the performance at these points and the actual performance was compared to the surrogate model predictions. Table 5.3 compares the actual solver results and the surrogate model predictions for the two test points A and B, defined in the top half of the table. The data show that the surrogate models provide good predictions for the responses of interest. For test point A, only two responses have a percentage error with a magnitude greater than 5% (the stability criterion at 12.0%, and the maximum RMS pitch at 6.5%). For test point B, only one of the responses has a percentage error with a magnitude greater than 5% (the resistance at 5.9%). This indicates an acceptable level of accuracy in the surrogate models to proceed with the optimization.

Table 5.3. Comparison of surrogate model predictions and actual solver results for two test points.

	Test Point A		Test Point B	
L (m)	199.67		180.23	
L/*B*	6.766		6.038	
B/*T*	3.369		2.934	
	Predicted: A	Actual: A	Predicted: B	Actual: B
Advance (m)	762.18	759.79	705.47	703.41
Tactical diameter (m)	967.31	962.31	906.52	902.52
Turnability index P_C	0.3168	0.3182	0.3380	0.3391
Stability criterion C	1.59×10^{-5}	1.42×10^{-5}	3.06×10^{-5}	2.96×10^{-5}
Steady turning diameter (m)	849.67	843.91	800.18	795.75
Maximum RMS heave (m)	0.487	0.487	0.528	0.518
Maximum RMS pitch (deg)	0.494	0.460	0.539	0.529
Maximum movement (m)	1.348	1.288	1.376	1.350
Resistance (kN)	1813.4	1755.0	2279.2	2152.6

5.3.6 Hull Form Optimization Results

The optimization was run using the MDO algorithm presented in Chapter 3. Throughout the optimization, the expensive objective functions and constraints were replaced with their surrogate models defined previously in this section. The optimization was started from an initial point that represented the original parent hull form; this was selected as the initial point because it was known to be feasible and had reasonable performance characteristics. The values for the design variables, objective functions, and constraints are shown in Table 5.4 for the initial point and optimal point. The performance at the optimal point was also evaluated using the actual solvers; the actual solver results are included in Table 5.4 for comparison. (Note that the data for the initial point are from the actual solvers.)

Table 5.4. Comparison of initial point and MDO optimal point.

	Initial Point	Optimal Point	
	Actual Solver	Surrogate Model	Actual Solver
Design Variables			
L (m)	178.77	155.42	
L/B	6.000	5.161	
B/T	3.000	2.566	
Beam B (m)	29.80	30.12	
Draft T (m)	9.932	11.74	
Objective Functions			
Cost function	100.0	93.33	
Steady turning diameter (m)	759.04	681.69	672.55
Maximum displacement (m)	1.335	1.279	1.247
Resistance (kN)	2153.2	2029.9	2078.9
Constraints			
C_b	0.7100	0.6836	
Upper bound on C_b	0.7332	0.6834	
Lower bound on C_b	0.6832	0.6334	
Advance (m)	683.55	609.44	602.17
Advance constraint: 4.5L (m)	804.49	699.41	
Tactical diameter (m)	867.91	779.80	769.77
Tactical diameter constraint: 5L (m)	893.87	777.12	
Turnability index P_C	0.3351	0.3791	0.3862
Stability criterion C	2.7×10^{-5}	5.0×10^{-5}	5.5×10^{-5}
Maximum RMS heave (m)	0.510	0.551	0.552
Maximum RMS pitch (deg)	0.529	0.537	0.513

Comparing the results between the starting point and the optimal point, the system level cost function and the three discipline objective functions have all been improved at the MDO optimum, while all of the constraints are satisfied. To confirm that the surrogate models performed well in this application, the actual solvers were used to re-evaluate the performance at the MDO optimal point. The results show that improvement in the objective functions was achieved in the actual solver results, and that all the constraints are satisfied.

Compared to the initial design, the length of the MDO optimal hull is reduced, while the beam and draft are increased. The cost function is highly dependent on the length of the hull and, in general, the maneuverability increases as the length of the hull decreases. For the selected seakeeping metrics, the seakeeping performance typically improves as the beam and draft increase. Half-hull models for the initial and the optimal designs are shown in Figure 5.3 to illustrate the change in dimensions.

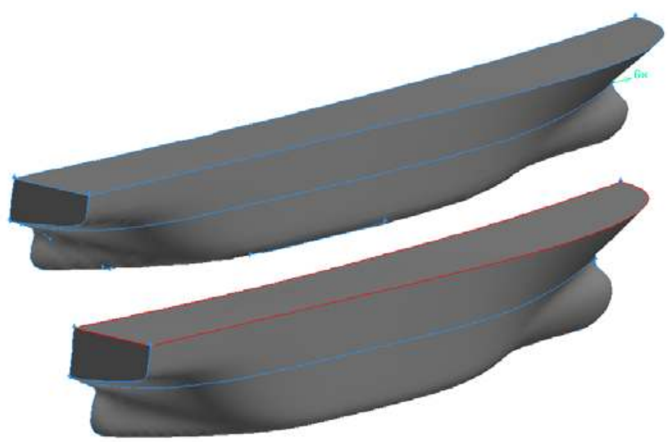


Fig. 5.3. Half hull models for the initial design (upper) and MDO optimal design (lower).

The purpose of utilizing surrogate models in the optimization was to achieve time savings during the optimization. In the optimization, the resistance objective function was called 167 times (using the pre-computed surrogate model); a single run of the CFD analysis took over 12 hours, while the entire optimization was completed in about 15 seconds. Furthermore, the maneuvering and seakeeping solvers cannot be called automatically, making their direct use in optimization impossible; manual operation of these solvers within the optimization would have made the optimization extremely slow.

5.4 Chapter Summary

In this chapter, several popular surrogate modeling techniques were introduced, including the Kriging method. Surrogate models are useful in optimization and MDO because expensive functions can be replaced with surrogate models that can be evaluated quickly. An MDO problem relevant to ship design was defined with the following disciplines: cost, resistance, maneuvering, and seakeeping. Surrogate models were developed to replace the expensive objective functions and constraints in the resistance, maneuvering, and seakeeping disciplines, and the MDO problem was solved using surrogate models with the MDO algorithm presented in Chapter 3. The results of the optimization yielded an improved design which satisfied all of the constraints when evaluated with the surrogate models. The performance at the optimum was reevaluated using the actual solvers; the true values of the objective functions demonstrated the predicted improvement and the constraints were satisfied. These results demonstrated that surrogate models (in particular, Kriging models) are valuable in reducing the computational time when solving MDO problem relevant to ship design.

CHAPTER 6

Set-Based Design and Multidisciplinary Design Optimization

The MDO methods discussed thus far pursue a single point design, and once the optimal configuration has been determined, the design is fixed. The design of naval vessels is a process which evolves over an extended period of time. During the design process, performance and cost requirements can change due to changes in the geopolitical and economic environments. A single point design that is fixed from the early stages of the design timeline cannot adapt easily to meet the evolving performance requirements. Alternatively, set-based design offers flexibility in the design process by pursuing a gradual reduction of the design space. In set-based design, designers begin with a broad set of values for the design variables, then gradually narrow the sets as more information becomes available.

In this chapter, background and theory of set-based design are presented. The techniques and advantages of set-based design which are used to develop an MDO algorithm along with the mathematical formulation of the new MDO algorithm are described in this chapter. Finally, the set-based MDO algorithm is applied to a simple ship design analysis, and the results are presented to demonstrate the effectiveness of the algorithm.

6.1 Set-Based Design Background and Review

The conventional design approach is point-based and iterative. Designers begin by selecting a variety of possible solutions, then choosing one to investigate further. The single design is evaluated and modified as necessary, then re-evaluated and changed again, in an iterative process which continues until a satisfactory design has been found (Liker et al. 1996). This process can be inefficient because new designs must continually be evaluated. The problem is compounded in complex multidisciplinary problems, with multiple likely-conflicting discipline analyses, and where different disciplines may be the responsibility of different design teams.

Set-based design is a design methodology that seeks to offer improvements over the traditional point-based design approach. Set-based design was popularized in a series of articles (Ward et al. 1995; Liker et al. 1996; Sobek et al. 1999) discussing Toyota's success with set-based design practices. The fundamental idea behind set-based design is to utilize sets of values for the design variables so that engineers communicate about the design in terms of those sets, instead of points. Singer, Doerry, and Buckley (2009) outline the following four features of set-based design:

1. Initially define a broad set of values for the design parameters.
2. Delay narrowing the sets to increase the amount of information available when making decisions.
3. Narrow the sets gradually as the design is improved.
4. Increase the design fidelity as the sets are narrowed.

In set-based design, engineers from different areas of the design (or disciplines) determine sets of feasible values for their own analysis, possibly also including preference information. Engineers share the set information to determine areas of feasible overlap, then gradually reduce the sets to focus on the feasible region.

The advantages of set-based design may not be immediately clear; it may seem inefficient to examine multiple designs simultaneously and to delay decisions about the design, hence Ward et al.'s (1995) term “the second Toyota paradox,” when describing the effectiveness of set-based design. In this work, four broad advantages of set-based design are considered: (1) communicating with sets leads to less rework than point-based design, (2) delaying decisions means that decisions are better-informed, (3) working with sets and delaying decisions allows for better to handling of uncertainty during the design process, and (4) the set-based design process inherently includes some degree of optimization.

The first advantage of set-based design is that it can reduce the amount of rework required in the design process. In point-based design, changes are made to the (single) design of interest when moving from one iteration to the next. It is unlikely that properties of the new design are the same as the previous iterations (otherwise there is no reason to move to the new design); this requires that the analyses for the new design be performed again with the new design characteristics. Therefore, a clear disadvantage of point-based design is that iterations of the design require constant rework and re-analysis (Liker et al. 1996).

In contrast, in set-based design, engineers work with sets of values for the design variables, and the sets are gradually narrowed as work progresses. Because the sets are being narrowed, no new designs are added to the design space; not allowing the sets to expand once they have been reduced is a critical aspect of Toyota's implementation of set-based design (Ward et al. 1995). Therefore, the previous analysis performed on the larger set is still valuable for the reduced set and rework is not necessarily required. While it may not seem possible to maintain analysis information on an entire set of points, Bernstein (1998) points out that the analysis does not need to be highly detailed. During the set-based design process, the purpose of the analysis is to determine the boundaries for the sets and also possibly assess preferences, not perform high-fidelity evaluations.

The second advantage of set-based design is that design decisions are more informed because the decisions are delayed. The amount of information about a design increases with time, as more analyses are performed and requirements become more clear (regardless of the choice of design method). An illustration of knowledge over time is shown in Figure 6.1. The figure shows that initially, there is very little knowledge about the design, then the amount of knowledge increases rapidly in the first portion of the design time (Bernstein 1998).

Therefore, when design decisions are made early in the design process, there is less information available on which to base the decisions. This may lead to increased iterations with point-based design because it is highly unlikely that the first few iterations will be based on accurate information. One of the principles of set-based design is to delay decisions until more information is available, because designers are much more likely to make a good decision when they have more information (Sobek, Ward, and Liker 1999). Delaying decisions about the design becomes possible because the design is described using sets, because engineers can continue to study the range of designs available without requiring a specific design selection.

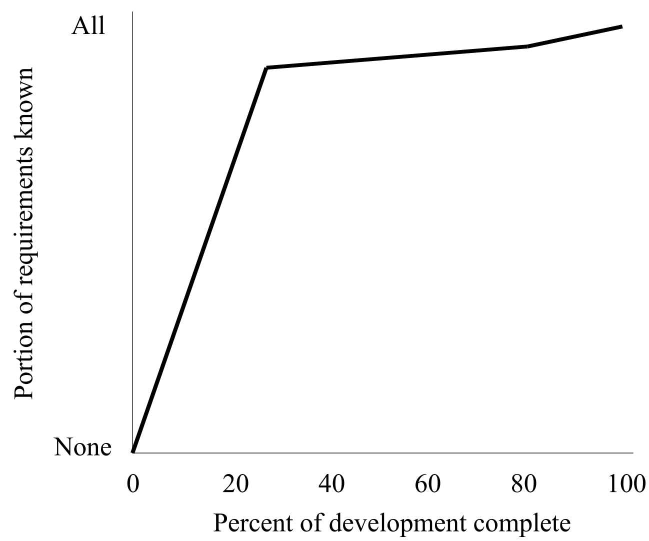


Fig. 6.1. Illustration of designers' knowledge about a design over time. Adapted from Bernstein (1998).

Not only does utilizing sets and delaying decisions improve decision-making, but it also improves the process' response to uncertainty. By using sets of values for the design variables, the design maintains flexibility in an uncertain environment (Sobek, Ward, and Liker 1999). The advantage over the point-based method is expressed concisely by Lee (1996): "... designers can represent sets of design possibilities instead of guessing one design if there are uncertainties." With set-based design, small changes due to uncertainty do not necessarily push the design into an unfeasible region or require rework.

Finally, the set-based design process inherently includes a degree of optimization. Singer, Doerry, and Buckley (2009) point out that set-based design requires the coordination of feasible sets between different disciplines. Engineers can locate and focus on the feasible intersection between the sets and integrate discipline information to further narrow the design space in the direction of an optimum (Bernstein 1998).

As the studies at Toyota by Liker, Sobek, Ward, and Cristiano (1995, 1996) show, set-based design practices are already in use in industry. However, there is a distinctive difference between implementing principles in an industrial setting and specifying a mathematical algorithm to solve design problems analytically. Recently, analytical formulations for set-based design have been presented in the literature.

Wang and Terpenny (2003) developed an evolutionary design synthesis procedure that handles populations of designs with the principles of genetic algorithms. The method utilizes fuzzy set theory to handle modeling inaccuracies, and principles of set-based design are included in the evolutionary procedure for generating and selecting designs. The method was applied to an automotive speedometer design problem.

Nahm and Ishikawa (2006) developed a design method inspired by set-based design. The method begins by developing a mapping from the design space to the performance space using interval arithmetic. The authors created an aggregated preference and robustness index that measures the designer's preference, and that includes robustness for handling uncertainty in the preference metrics. The aggregated preference and robustness index is used to narrow the design space. The authors applied this method to the preliminary design of a vehicle side impact beam.

Shahan and Seepersad (2009) developed a set-based design technique where a Bayesian network (essentially a joint probability distribution) is constructed for each discipline. The Bayesian networks indicate the regions of interest in the design space for each discipline, and the networks are shared to communicate information about the design. This method was then applied to the design of an unmanned aerial vehicle with aerodynamics, structures, and performance analyses.

Madhavan (2008) et al. developed a set-based design method where a compromise decision support problem (a mathematical model for multiobjective decision-making) is developed for each discipline. The disciplines calculate and share target values for coupled parameters, then the disciplines use the target values to generate Pareto optimal solutions; the final solution is selected from the Pareto optimal designs. This strategy was used in an industrial application in the design of a downhole module for oil and gas drilling applications.

Malak, Aughenbaugh, and Paredis (2009) developed an approach for conceptual design that handles imprecision, or the lack of specific knowledge for the design. The authors implemented features from multi-attribute utility theory (scalar preference functions for design attributes) and set-based design. The authors utilized set-based design because sets are useful for describing imprecision in the design. The design strategy was applied to the conceptual design of a vehicle transmission.

Finally, Avigad and Moshaiov (2010) developed a computational approach for multiobjective problems to incorporate the set-based design concept of intentionally delaying decisions about the design. The authors utilized a tree representation of the design space, where different branches represent different possible decisions, and the trees are pruned as decisions are made. Robustness to the uncertainty delayed decision-making was included by using worst-case considerations. The method was applied to an example problem in structural mechanics.

6.2 Multidisciplinary Design Optimization Algorithm

In this section, the new MDO algorithm inspired by the principles of set-based design is presented. Incorporating features of set-based design and allows for greater flexibility when dealing with evolving requirements compared to a single-point optimization. The principles of set-based design which are included in the MDO algorithm are identified in the following sections (indicated with italics) and then transformed into mathematical statements for the optimization algorithm.

This section is discusses the details of the new MDO algorithm. Information from five areas of the algorithm is presented: system design variables, objective function scaling, flexibility in constraints, the system optimization statement, and the discipline level optimization statement.

6.2.1 System Design Variable Definition

The MDO algorithm assumes the formulation of the MDO problem as in Equation (2.36), which is repeated below:

$$\min_{\mathbf{x}\in\chi} \ \mathbf{f}(\mathbf{x}) \tag{6.1}$$

$$\text{subject to } \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0} \quad i = 1, \ldots, p$$

In the MDO problem, there are n design variables contained in the vector $\mathbf{x}$, and p disciplines with p corresponding objective functions $f_i(\mathbf{x})$ which are contained in the vector $\mathbf{f}(\mathbf{x})$. The vector of constraints $\mathbf{g}_i(\mathbf{x})$ applies to discipline i. The allowable ranges for the design variables are denoted χ; the ranges for the design variables can be defined with lower and upper bounds:

$$x_i \in [x_{i,LB},\ x_{i,UB}] \quad i = 1, \ldots, n \tag{6.2}$$

The range of allowable values for x_i can be viewed as the set of values between the lower bound $x_{i,LB}$ and the upper bound $x_{i,UB}$.

As described in Section 6.1, one of the principles of set-based design is to *describe the design variables by sets which change while the design progresses*. The bounds on the design variables for the original MDO problem are defined in Equation (6.2). Because the bounds on the design variables change during the optimization, the bounds on the design variables at any other state are defined as

$$x_i \in [x_{i,min}, \; x_{i,min} + \Delta x_i] \quad i = 1, \ldots, n \tag{6.3}$$

This expression indicates that the new lower bound on x_i is $x_{i,min}$ and the new upper bound on x_i is $x_{i,min} + \Delta x_i$, where Δx_i is the width of the interval for x_i. While it may seem simpler to define a maximum for x_i, instead of the sum $x_{i,min} + \Delta x_{i,}$ the purpose of this formulation is to easily track the width of the interval.

Figure 6.2 illustrates the change in the bounds on the design variables for a simple case with only two design variables. The figure shows the original design space as the large outer rectangle, and the new design space is the smaller, lightly shaded region. The new design space is defined by the dashed lines which indicate the new ranges on the design variables.

The purpose of the new MDO algorithm is to use the principles of set-based design to motivate the design optimization. This means that the design space is changed through the optimization process; the design space is defined using the sets for the design variables defined in Equation (6.3). Therefore, the choice for the design variables in the system optimization statement are not the design variables **x**, but the variables which define the sets: $x_{i,min}$ and Δx_i.

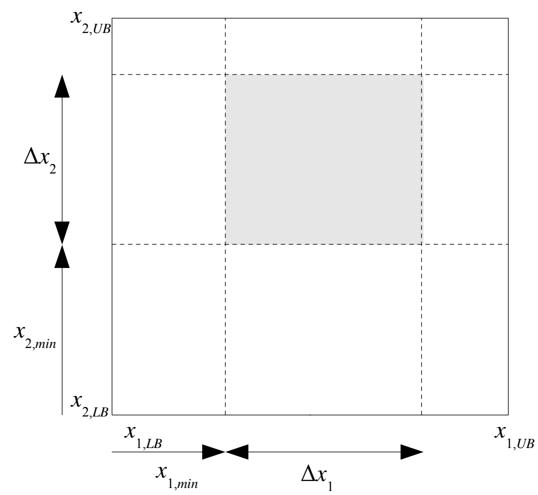


Fig. 6.2. Illustration of the reduction in size of the design space by changing the ranges of the design variables.

Additionally, before beginning the optimization, scaling of the design variables is important for accurate performance of the optimizer. The system design variables are scaled so that they take values between zero and one:

$$z_{i,min} = \frac{x_{i,min} - x_{i,LB}}{x_{i,UB} - x_{i,LB}} \tag{6.4}$$

$$\Delta z_i = \frac{\Delta x_i}{x_{i,UB} - x_{i,LB}} \tag{6.5}$$

In summary, the system level optimization includes $2n$ design variables: $z_{i,min}$ and Δz_i (i = 1, …, n), or as vectors $\mathbf{z}_{min}$ and $\Delta\mathbf{z}$; these design variables describe the size and location of the new design space. Furthermore, *the system level design variables describe the design space in terms of sets for the design variables*, which is a fundamental requirement for set-based design.

6.2.2 Objective Function Scaling

The MDO problem includes *p* objective functions which may have different units of measurement and vastly different magnitudes. Additionally, some objective functions may vary greatly throughout the design space, while others exhibit less variation. Therefore, in order to include any evaluation and comparison of the different objective functions, they must be scaled, or normalized. The objective functions are normalized as in Marler and Arora (2004):

$$F_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^*}{f_i^m - f_i^*} \tag{6.6}$$

where F_i is the normalized form of objective function f_i. f_i^* is the minimum of f_i when considering only discipline i, or

$$f_i^* = f_i(\mathbf{x}_i^*) = \min_{\mathbf{x}} \ f_i(\mathbf{x}) \tag{6.7}$$

$$\text{subject to} \quad \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}$$

f_i^m indicates the maximum value of f_i, and f_i^m can be approximated according to (Marler and Arora 2004)

$$f_i^m = \max_j f_i(\mathbf{x}_j^*) \qquad i \neq j \tag{6.8}$$

Typically, the objective functions are conflicting (otherwise the problem is not a true multidisciplinary problem); then f_i^m is close to the worst performance that may be encountered in objective i.

Equation (6.6) yields values for the normalized objective function F_i in the range between 0 and approximately 1. The discipline optimum $\mathbf{x}_i^*$ gives the best possible performance for f_i, and at that point F_i is 0. The value of F_i can be interpreted as an amount of compromise in objective f_i; small values of F_i indicate little compromise, while values close to 1 indicate that the current point is far from f_i^*.

6.2.3 Flexibility in Constraints

When an engineer configures a real design, it may not be preferable (or possible) to select a design that exactly satisfies all of the constraints. Instead, it may be beneficial to relax a constraint if it allows for a significant improvement of the objective function. For example, consider a single-objective optimization problem where the objective is to maximize the speed of a vehicle, subject to a constraint for the maximum cost of the vehicle. For a real design problem, the designer may be willing to compromise on the cost constraint; if a small violation in the cost constraint (say, 1%) allows for a 20% increase in speed, the designer could choose to allow the slightly higher cost. One way to handle compromise in the constraints would be to determine the maximum allowable compromise, and simply shift the constraint by that amount. However, this does not ensure that the compromise in the constraint will correspond to a *significant* improvement in the objective function.

A method to handle the compromise in the constraints and the corresponding changes in the objective functions is included in the system level optimization of the new MDO algorithm. A new design variable denoted $\boldsymbol{\varepsilon}$ is introduced where each element in $\boldsymbol{\varepsilon}$ corresponds to a constraint in the MDO problem:

$$g_j(\mathbf{x}) \le \varepsilon_j \tag{6.9}$$

This expression indicates that the original constraint g_j can be violated by as much as ε_j. However, because ε_j is a design variable, not a constant, the amount by which the constraint can be violated will vary during the optimization; the size of ε_j is controlled by the system level optimization statement.

6.2.4 System Level Optimization Statement

This section describes the system level optimization statement, where the purpose of the system level optimization is to coordinate the discipline optimizations. This section begins with reviewing the design variables used in the system level optimization statement. Then the system level optimization statement is presented, followed by an explanation of each term included in the optimization statement.

Before introducing the system level optimization, it is important to review the design variables used in the system optimization. The system optimization statement uses the design variables $\mathbf{z}_{min}$, $\Delta\mathbf{z}$, and $\boldsymbol{\varepsilon}$ which were defined in the previous sections. The design variables $\mathbf{z}_{min}$ and $\Delta\mathbf{z}$ describe the location and size of the modified design space, and because they are scaled, both take values in the range 0 to 1. The design variable ε_j takes values between 0 and a predefined value $\varepsilon_{j,\max}$, where $\varepsilon_{j,\max}$ is the absolute maximum amount the designer is willing to compromise on constraint j (this amount will vary depending on the problem).

Therefore, the system optimization problem has $2n + m$ design variables (when n is the number of design variables for the original problem and m is the total number of constraints with flexibility). *The design variables for the system level optimization are no longer the design variables* $\mathbf{x}$ for the original MDO problem, because in set-based design logic, point designs are no longer the focus of the optimization.

The system level optimization statement is:

$$\min_{\mathbf{z}_{min},\,\Delta\mathbf{z},\,\boldsymbol{\varepsilon}} \quad \exp\left(\Delta z_1 \cdot \ldots \cdot \Delta z_n\right) + \sum_j \exp\left(\frac{\varepsilon_j}{\varepsilon_{j,\max}}\right) + \sum_{i=1}^{p} \exp\left(F_i\left(\mathbf{x}_i^{\text{new}}\right)\right) \qquad (6.10)$$

$$\text{subject to} \quad g_j(\mathbf{x}_i^{\text{new}}) \leq \varepsilon_j \quad \text{for all } j \text{ and for all } i$$

The first term in the system objective function describes the size of the design space (the hypervolume $\Delta z_1 \cdot \ldots \cdot \Delta z_n$). The first term is used reduce the size of the design space as much as possible, because minimization of this term reduces the size of the design space. Therefore, this term implements the principle of set-based design to *gradually reduce the size of the design space*.

The second term in the system objective includes the compromise values $\boldsymbol{\varepsilon}$. During the optimization, the compromise values allow the constraint j to be violated by as much as ε_j. However, the amount of compromise should be as little as possible, thus the system objective function minimizes the compromise values. The summation over index j represents the summation over all constraints which allow for compromise; this allows for the possibility that some constraints cannot be relaxed.

The third term includes the effects of the discipline objective functions. The term includes the function F_i evaluated at the point $\mathbf{x}_i^{\text{new}}$, where $\mathbf{x}_i^{\text{new}}$ denotes the current value for the design variables returned by the discipline i optimization (details of the discipline level optimization are given in the following section). The purpose of the third term is to evaluate the current performance of the (normalized) objective functions. Because of the normalization, F_i takes values close to zero when $\mathbf{x}_i^{\text{new}}$ is close to $\mathbf{x}_i^*$; therefore, minimization of the sum of F_i seeks to minimize the discipline objective functions.

The system objective function is composed of a sum of functions (terms) that essentially represent a multiobjective problem (the first term describes the size of the design space, the second term describes the constraint flexibility, and the third term describes the improvement in the disciplines' objective functions). Objective functions which are stated as a sum of other functions sometimes are not well-behaved, because the Pareto front for the corresponding multiobjective problem is not necessarily convex. In Equation (6.10), the system objective function evaluates the exponential of each term. The purpose of this choice is to make the Pareto front convex (Athan and Papalambros 1996), which can improve the performance of the optimization.

Finally, the system optimization statement includes the constraints g_j from all disciplines (including the tolerance of ε_j) and the constraints are evaluated at each of the new discipline optima i; that is, every constraint is checked at each discipline optimum $\mathbf{x}_i^{\text{new}}$. The constraints ensure that all of the new discipline optima are feasible for all disciplines, or that all (new) discipline optima lie within the common feasible space. This enforces the concept of set-based design that *designs must be found in the intersection of the sets*.

To reiterate the earlier statements regarding the system design variables, the solution of the system optimization returns the optimal values $\mathbf{z}_{min}^*$ and $\Delta\mathbf{z}^*$. The values describe the reduced design space, not a specific choice for the original design variables $\mathbf{x}$. This is in agreement with the set-based design perspective of *viewing the design space in terms of sets instead of point designs*.

6.2.5 Discipline Level Optimization Statement

The purpose of the discipline level optimization statement is to call the discipline analyses and calculate the new design point $\mathbf{x}_i^{\text{new}}$ for each discipline i. At each iteration in the optimization, discipline optimizations are performed, but with the bounds on the design variables determined by the current values of $\mathbf{z}_{min}$ and $\Delta\mathbf{z}$. The discipline optimization statement is

$$\min_{\mathbf{x} \in \chi(\mathbf{z}_{\min}, \Delta \mathbf{z})} f_i(\mathbf{x}) \tag{6.11}$$

$$\text{subject to} \quad g_j(\mathbf{x}) \leq \varepsilon_j \quad \text{for all } j \text{ in } \mathbf{g}_i$$

The primary difference between this discipline optimization statement and the independent discipline optimization in Equation (6.7) is the allowable ranges of the design variables, χ. For the discipline level optimization statements, χ is a function of the current values of the system design variables $\mathbf{z}_{min}$ and $\Delta\mathbf{z}$. Therefore, the discipline optimization is performed over the reduced design space described by $\mathbf{z}_{min}$ and $\Delta\mathbf{z}$, so the results for the discipline level optimization will be different from the discipline optimization of Equation (6.7).

The discipline level optimization also includes the flexibility in the constraints, $\boldsymbol{\varepsilon}$. The optimization statement of Equation (6.11) requires that the constraints for discipline i, $\mathbf{g}_i$, are satisfied but with the appropriate level of flexibility given by the corresponding values of ε_j. The values for $\boldsymbol{\varepsilon}$ are the current values from the system level optimization and they will change during the system optimization process.

The discipline level optimization with the new variable bounds $\chi(\mathbf{z}_{min}, \Delta\mathbf{z})$ returns the optimum values for the design variable values $\mathbf{x}_i^{\text{new}}$ and objective function value $f_i(\mathbf{x}_i^{\text{new}})$. The purpose of the discipline optimization is to locate the the discipline optimum within the current design space, defined by the sets at the system level; this ensures that *the algorithm takes into account the discipline optima (or viewed as discipline* *preference*), which is part of the set-based design approach.

6.3 Simple Example Application

To demonstrate the new MDO algorithm, a simple example problem relevant to ship design was tested. The simple example problem was defined with only two design variables so that the results of the algorithm can easily be visualized in a two-dimensional plot. The two design variables are the length of the ship (in meters) $145 \leq L \leq 170$ and the beam of the ship (in meters) $25 \leq B \leq 28$. There are three disciplines: cost, speed, and stability, and two constraints. Calculations for the model are based on simple regression equations from Parsons (2003); details for the model are provided in Table 6.1.

Table 6.1. Simple ship model definition.

Depth D (m)	$D=\dfrac{B}{1.70}$
Draft T (m)	$\dfrac{B}{T}=5.93-3.33\,C_M$
Midship Coefficient C_M	$C_M = 0.977 + 0.085(C_B - 0.60)$
Block Coefficient C_B	$C_B=\dfrac{\Delta}{LBT\,1.025\cdot 1.005}$
Cost C	$C=1.03\dfrac{L}{L_0}+0.78\dfrac{B}{B_0}+0.18\,C_B$
Speed V_k (knots)	$V_k=\dfrac{(1.18-C_B)}{0.69}\sqrt{3.2808\,L}$
KG (m)	$\mathrm{KG} = 0.68D$
Waterplane Coefficient C_{WP}	$C_{WP}=\dfrac{1+2\,C_B}{3}$
Vertical Prismatic Coeff. C_{VP}	$C_{VP}=\dfrac{C_B}{C_{WP}}$
KB (m)	$\mathrm{KB}=\dfrac{T}{3}(2.5-C_{VP})$
Waterplane Inertia Coeff. C_I	$C_I = 0.1216C_{WP} - 0.041$
BM (m)	$\mathrm{BM}=\dfrac{C_I\,L\,B^3}{\nabla}$
GM (m)	$\mathrm{GM} = \mathrm{KG} + \mathrm{BM} - \mathrm{KB}$

The first discipline is the cost estimate for the ship, and the goal of the discipline is to minimize the (nondimensional) cost; that is, $f_1(\mathbf{x}) = C$, where C is the cost evaluated according to the equations in Table 6.1. The second discipline is the speed of the ship, and the goal of the discipline is to maximize the speed (in knots); that is, $f_2(\mathbf{x}) = -V_k$, where V_k is evaluated according to Table 6.1. The third discipline is the stability, which is measured by the GM (transverse metacentric height, in meters). The goal of the discipline is to maximize the GM; that is, $f_3(\mathbf{x}) = -\mathrm{GM}$, where GM is evaluated according to Table 6.1.

The design problem includes two constraints which are upper and lower bounds on the block coefficient, C_B. The Watson and Gilfillan mean line recommendation (Watson 1998) for the mean value of C_B is $C_{B,\mathrm{mean}}$:

$$C_{B,\,\mathrm{mean}} = 0.70 + 0.125 \tan^{-1}\left(\frac{23 - 100\,Fn}{4}\right) \tag{6.12}$$

The recommended variation from $C_{B,\mathrm{mean}}$ is 0.025 (Watson 1998),which defines the constraints:

$$C_{B,\mathrm{mean}} - 0.025 \le C_B \le C_{B,\mathrm{mean}} + 0.025 \tag{6.13}$$

The constraints on C_B are contained in the vector $\mathbf{g}(\mathbf{x})$, rewritten in negative null form. With the definitions for the three disciplines (cost, speed, and stability), and the two constraints, the example MDO problem can be defined as:

$$\begin{array}{lcccc} \min\ f_1(\mathbf{x}) & \text{and} & \min\ f_2(\mathbf{x}) & \text{and} & \min\ f_3(\mathbf{x}) \\ \text{s.t.}\ \ \mathbf{g}(\mathbf{x}) \le \mathbf{0} & & \text{s.t.}\ \ \mathbf{g}(\mathbf{x}) \le \mathbf{0} & & \text{s.t.}\ \ \mathbf{g}(\mathbf{x}) \le \mathbf{0} \end{array} \tag{6.14}$$

The design space for the example problem is shown in Figure 6.3. The figure includes the individual discipline optima and the constraints on the block coefficient. The figure shows that the individual discipline optima are in different corners of the design space, indicating that compromise is necessary in order to find a multidisciplinary solution.

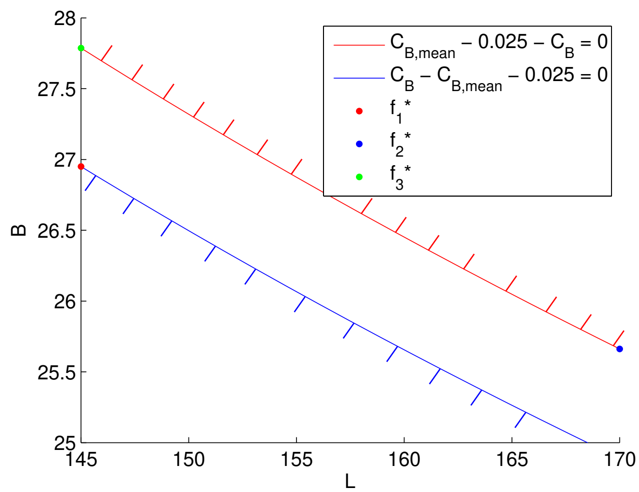


Fig. 6.3. Initial design space for the simple example.

The set-based MDO algorithm was applied to the simple ship example problem, and the results are shown in Table 6.2. For the top portion of the table, the "Initial" column describes the initial design space, where the design space is the original size and the constraints are unmodified. The "Optimal" column describes the design space determined by the optimization algorithm. For the bottom portion of the table, the lower and upper bounds for the two design variables are given for the initial conditions and the results given by the optimization.

The results in Table 6.2 are plotted in Figure 6.4. The new boundaries for the design space are shown as dashed lines, and the discipline optima within the new design space are shown as stars. The design space has been reduced to the rectangle in the upper left corner; the vast majority of the multidisciplinary compromise is in f_2, which has been moved from its original location in the lower left corner. This result makes sense because the discipline optima for f_1 and f_3 are relatively close together, so that they pull the final design space in their direction.

Table 6.2. New design space for the simple ship design problem.

	Individual	System
Discipline 1 Optimum f_1^*	1.905	1.906
Discipline 2 Optimum f_2^*	-11.025	-10.492
Discipline 3 Optimum f_3^*	-11.388	-11.386
	Initial	**Optimal**
L min	145.00	145.01
L max	170.00	153.95
B min	25.00	26.97
B max	28.00	27.78

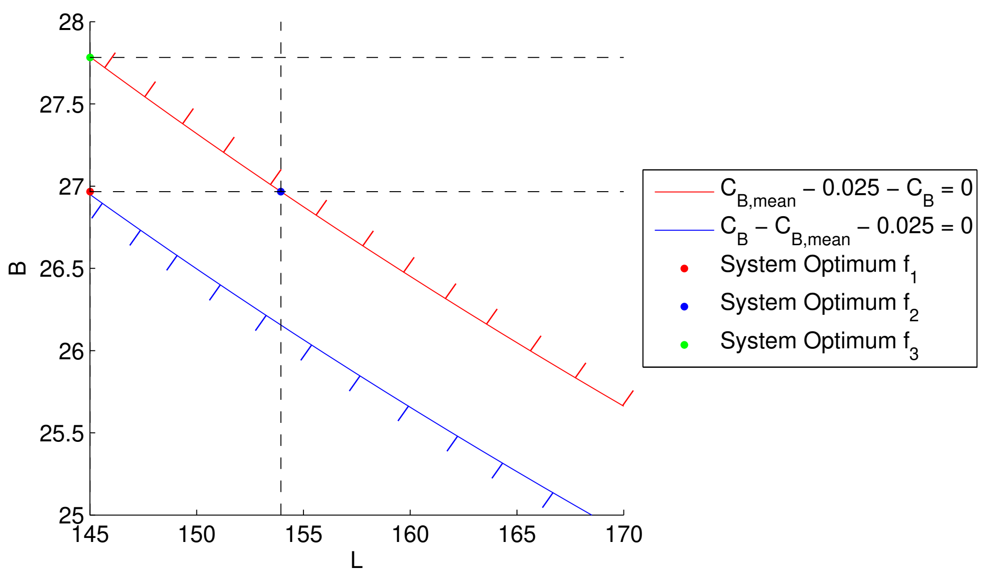


Fig. 6.4. Final design space determined by the MDO algorithm for the simple example problem.

For this example problem, the algorithm did not result in compromise in the constraints (both ε_j were approximately zero). This makes sense when looking at the location of the constraints in Figure 6.4; in particular, it can be seen that relaxing the constraints would not yield an improvement in f_2, and improvements in f_1 and f_3 would be minimal because these solutions are already so close to their individual discipline optima.

6.5 Chapter Summary

This chapter introduced the theory and advantages of set-based design. A new MDO algorithm was developed using the principles of set-based design, and the mathematical formulation was presented. The performance of the algorithm was demonstrated through a simple ship design example.

CHAPTER 7

Set-Based Design and Multidisciplinary Design Optimization Application to Ship Design

The final element of this dissertation is a comprehensive application of the set-based design MDO algorithm presented in Chapter 6 to a ship design problem. Fortran codes were used for conducting resistance, maneuvering, and seakeeping computations based on mathematical models from the literature. The results from the set-based design MDO algorithm are compared with results from a point-based multiobjective optimization.

7.1 Ship MDO Problem Definition

A tanker hull that was readily available from the website of the commercial naval architecture software MaxSurf was selected for this study. The MaxSurf model provided a useful starting point because the hydrostatic properties of the hull could be evaluated directly and accurately within MaxSurf. Several dimensions of the original hull form from MaxSurf were outside the range of applicability for the software used in the discipline analyses, so the hull form was adjusted using MaxSurf's parametric transformation. The modified hull form is referred to as the parent hull for this analysis, and a profile view of the hull form is shown in Figure 7.1.

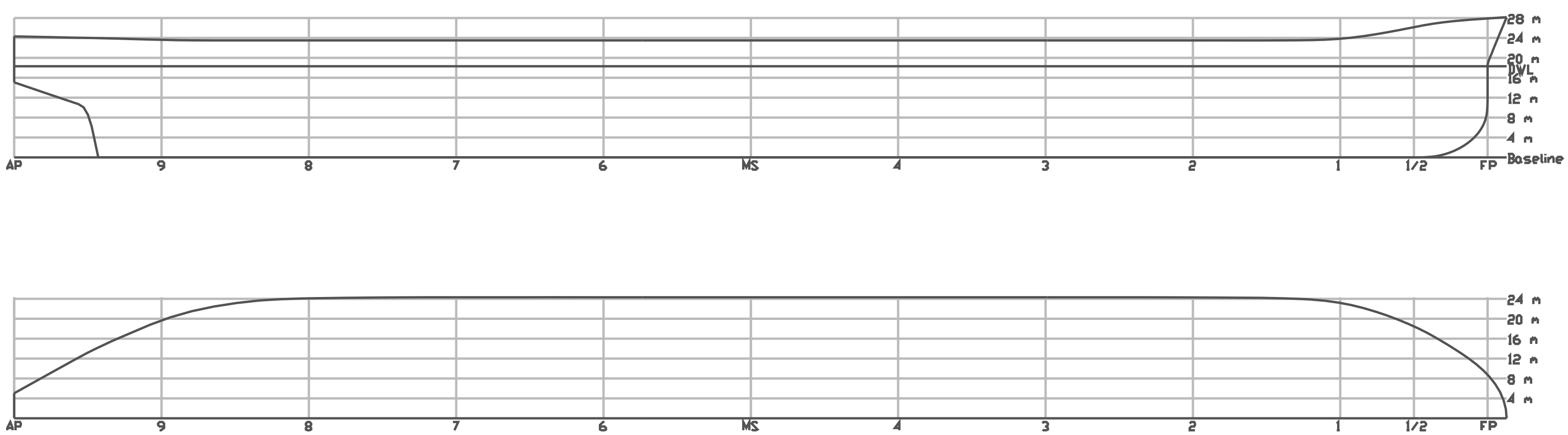


Fig. 7.1. Profile and plan views of the parent hull form from MaxSurf.

The next step in defining the MDO problem is to select design variables that describe the hull form parametrically; the design variables for ship model are length L, length-to-beam ratio L/B, and beam-to-draft ratio B/T. Ratios are selected as design variables instead of the dimensions (such as beam) because the programs used for the discipline analyses have requirements on the ratios; the lower and upper bounds on the design variables ensure that the values for the length, length-to-beam ratio, and beam-to-draft ratio meet the requirements for the discipline analyses. The allowable ranges for the design variables are given in Table 7.1; the table also includes the values for the parent hull for comparison.

Table 7.1. Ranges for the design variables.

	Lower Bound	Upper Bound	Parent Hull
L (m)	250.0	320.0	315.0
L/B	5.60	8.00	6.20
B/T	2.25	3.75	3.00

While it is possible to use MaxSurf's parametric transformation capabilities to generate new hull forms based on the values of the design variables, including MaxSurf within the optimization loop was considered infeasible since manual operations are required for using the program. Instead, parametric models were developed to evaluate all of the information necessary to run the discipline analysis codes; the following section presents the parametric modeling information. The remainder of the section defines the MDO problem, which includes three disciplines: resistance, maneuvering, and seakeeping.

7.1.1 Hull Modeling

The codes for the discipline analyses require a variety of input information that describes the hull form. In order to run the discipline analysis, parametric models for the inputs were developed so that the necessary inputs could be calculated from the three design variables.

Some properties of the hull form were fixed, for example the displaced volume and the speed of the ship. Other characteristics were assumed to be approximately constant, such as the midship coefficient; the midship coefficient for the parent hull was 0.996 (essentially a box barge midsection), and scaling the geometry was assumed to have a negligible effect on the midship section. All design properties that were either held fixed or assumed to be constant are listed in Table 7.2.

Table 7.2. Ship modeling properties fixed or assumed constant.

displaced volume (m^3)	∇	216,971
transverse bulb area (m^2)	A_{BT}	0
transom area (m^2)	A_{TR}	0
nondimensional bow profile	A_{BOW}	-0.0055
midship coefficient	C_M	0.996
VCG (as proportion of draft)	VCG	1.076
ship speed (knots)	V_k	18

The waterplane coefficient C_{WP} is estimated with a simple parametric model using the block coefficient C_B (Parsons 2003)

$$C_{WP} = \frac{C_B}{0.471 + 0.551\, C_B} \tag{7.1}$$

The seakeeping analysis requires the most extensive hull information, including the draft, beam, sectional area, and weight distribution at each station. A simple approach was used for assessing the draft at each station because, as shown in Figure 7.1, the long parallel midbody gives a hull form where only stations 0 and 10 do not have a sectional draft equal to the draft at midship. Therefore, the draft at each station was assumed to be proportional to the draft of the parent hull, simply scaled by the new draft.

The sectional beam was assumed to be more sensitive to the changes in the hull form. While the large area of parallel midbody was assumed to have constant beam, the bow and the stern areas were adjusted from their parent hull values. The sectional beam in the bow and stern areas was scaled by the ratio of the waterplane coefficient of the new hull to the waterplane coefficient of the parent hull. The waterplane coefficient was selected for the scaling because it is defined using the waterline beam along the length of the hull.

The sectional area curve was modeled using the technique of Taylor (1915), who described the sectional area curve using two fourth-order polynomials (one for the forward half of the ship and one for the aft half). Parameters were fit to the fourth-order model using the parent hull; then the sectional area curve can be generated for a new hull given the prismatic coefficient. Additionally, the sectional area curve can be used to evaluate the LCB (the centroid of the sectional area curve).

A very simple weight distribution was developed for the parent hull, where a portion of the displacement was assigned to each station. To determine a weight distribution for a new hull, the relative weight of the stations is adjusted until the LCG matches the LCB.

Several of the discipline analyses have requirements on the value for the block coefficient. All disciplines include the block coefficient constraints:

$$0.56 \le C_B \le 0.87 \tag{7.2}$$

7.1.2 Resistance Discipline

The first discipline considered in the optimization analysis is the resistance. The objective function is the hull resistance, which is evaluated using the powering prediction method of Holtrop and Mennen (1982) and Holtrop (1984). The resistance discipline includes the two constraints on the block coefficient from Equation (7.2). Then the resistance discipline optimization problem is

$$\min_{\mathbf{x}} \; R_{TOTAL} \tag{7.3}$$

$$\text{s. t.} \quad \begin{array}{l} 0.56 - C_B \le 0 \\ C_B - 0.87 \le 0 \end{array}$$

where R_{TOTAL} denotes the total resistance of the hull, and **x** is the vector containing the three design variables L, L/B, and B/T.

7.1.3 Maneuvering Discipline

The second discipline is the maneuvering performance, which is evaluated using the Maneuvering Prediction Program (MPP) that was introduced earlier in Chapter 5, Section 5.3.3. The objective function of the maneuvering discipline is the tactical diameter, which is to be minimized. The maneuvering discipline includes the constraints on the block coefficient from Equation (7.2). The maneuvering discipline also includes a constraint on the stability criterion C, which is required to be positive; the stability criterion was originally defined in Equation (5.32). Then the maneuvering discipline optimization problem is

$$\min_{\mathbf{x}} \; D_T \tag{7.4}$$

$$\text{s. t.} \quad \begin{aligned} 0.56 - C_B &\le 0 \\ C_B - 0.87 &\le 0 \\ -C &\le 0 \end{aligned}$$

where D_T is the tactical diameter of the ship.

7.1.4 Seakeeping Discipline

The third discipline is the seakeeping performance, which is evaluated using the Seakeeping Prediction Program (SPP) that was introduced earlier in Chapter 5, Section 5.3.2. The objective function of the seakeeping discipline is a metric which represents the maximum combined motion:

$$M_{\max} = \max(\text{RMS heave}) + \frac{L}{2}\max(\text{RMS pitch}) + \frac{B}{2}\max(\text{RMS roll}) \tag{7.5}$$

where roll is evaluated at a 30 degree heading, heave is evaluated in beam seas, and pitch is evaluated at a 120 degree heading; the sum in Equation (7.5) does not represent the actual motion of any part of the ship, but a multiobjective metric for the ship motion. The seakeeping discipline includes the constraints on the block coefficient from Equation (7.2). The seakeeping discipline also includes a stability constraint on the GM, where an approximation for the GM (as calculated by SPP) is required to be greater than the US Coast Guard GM requirement. Then the seakeeping discipline optimization problem is

$$\min_{\mathbf{x}} \; M_{\max} \tag{7.6}$$

$$\text{s. t.} \quad \begin{aligned} 0.56 - C_B &\le 0 \\ C_B - 0.87 &\le 0 \\ \text{GM}_{\text{USCG}} - \text{GM} &\le 0 \end{aligned}$$

7.1.5 Cost Estimate

An additional constraint was introduced on the cost of the ship. A simple linear cost model based on Parsons (2003) is used to evaluate the cost of the ship, where the subscript "0" indicates properties of the parent hull:

$$\text{cost}(\mathbf{x})=\left[1+1.01\frac{(L-L_0)}{L_0}+0.58\frac{(B-B_0)}{B_0}+0.40\frac{(T-T_0)}{T_0}+0.22\frac{(C_B-C_{B0})}{C_{B0}}\right] \quad (7.7)$$

Based on this cost formulation, the cost of the parent hull is 1 and variations can be viewed as percentage increases or decreases in cost from the parent hull (for example, the cost of another hull may be 1.02, which indicates a 2% increase in cost). A cost constraint is defined so that the cost of the new hull must be less than or equal to the cost of the parent hull:

$$g_{\text{cost}}(\mathbf{x}) = \text{cost}(\mathbf{x}) - 1 \leq 0 \quad (7.8)$$

Flexibility was only considered in the cost constraint; no flexibility was included in the other four constraints because the they help ensure that the discipline analyses run properly (for example, the seakeeping analysis code will error if the ship has a negative GM due to flexibility introduced in the corresponding constraint).

7.1.6 MDO Problem Summary

The disciplines can be optimized individually by solving the independent optimization statements in Equations (7.3), (7.4), and (7.6). Results for the independent discipline optimizations are given in Table 7.3; the results show that the optima occur at different points for the different disciplines, which is expected for a multidisciplinary problem.

Table 7.3. Independent discipline optimization results.

	Design Variables $\mathbf{x}^*$			Objective Function f_i^*
	L (m)	L/B	B/T	
Discipline 1	320.00	6.8889	2.2500	4.2301×10^9 N
Discipline 2	303.18	5.6000	3.5633	869.78 m
Discipline 3	314.22	5.7716	3.7291	22.178 m

Additionally, the constraints at the three discipline optima were evaluated and the results are summarized in Table 7.4. The Discipline 1 optimization statement in Equation (7.3) includes only the constraints on the block coefficient, so it is not surprising that the optimum violates the constraint for GM (this makes sense because the hull form with the least resistance is long and thin, which will also have the least transverse stability). The conflicts between the different disciplines illustrate that this ship design problem is a good example MDO problem because compromise must be achieved to ensure that all discipline optima are feasible.

Table 7.4. Independent discipline optimization constraint values.

	Discipline 1 Optimum	Discipline 2 Optimum	Discipline 3 Optimum
$g_1 = 0.56 - C_B$	-1.47×10^{-1}	-3.10×10^{-1}	-3.09×10^{-1}
$g_2 = C_B - 0.87$	-1.63×10^{-1}	-1.96×10^{-11}	-1.25×10^{-3}
$g_3 = -C$	-2.33×10^{-5}	-3.00×10^{-7}	-2.00×10^{-6}
$g_4 = \mathrm{GM}_{\mathrm{USCG}} - \mathrm{GM}$	1.61	-6.66	-7.83
$g_{\mathrm{cost}} = \mathrm{cost} - 1$	0.075	0.034	0.062

Table 7.4 also includes the cost constraint evaluated at the discipline optima; the cost constraint was not enforced during the discipline optimizations and all three discipline optima violate the cost constraint. Therefore, significant compromise is expected during the multidisciplinary analysis for either the cost constraint (utilizing the flexibility) or the location of the final design space. Figure 7.2 shows a plot of horizontal

contours of the cost constraint (which is a surface); the constraint is satisfied to the left of the contours, showing that all three discipline optima are fairly far from the constraint boundary.

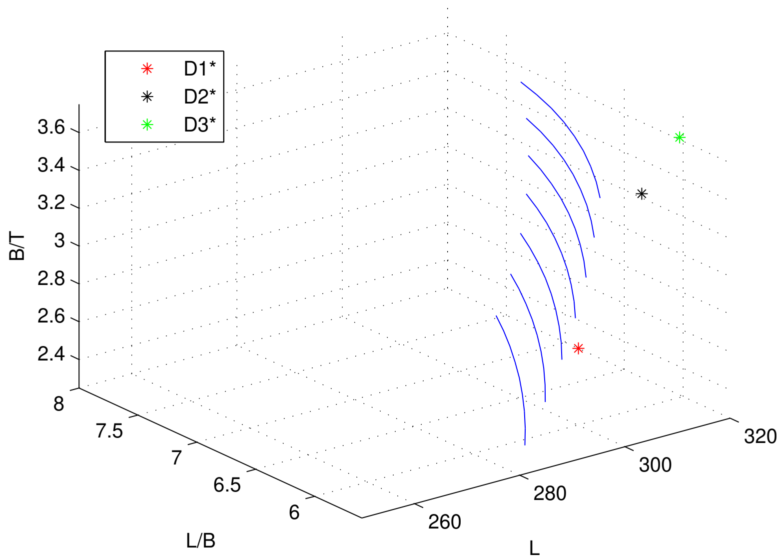


Fig. 7.2. Discipline optima and contours of the cost constraint.

To summarize the overall set-based design MDO statement, the system design variables are listed in Table 7.5. The table relates the system design variables to the discipline design space. Figure 7.3 reviews the structure of the set-based design MDO problem.

Table 7.5. System level design variables for the ship design problem.

System Design Variable	Discipline Equivalent
$z_{1,\min}$	Lower bound on *L*
Δz_1	Width of interval in *L*
$z_{2,\min}$	Lower bound on *L/B*
Δz_2	Width of interval in *L/B*
$z_{3,\min}$	Lower bound of *B/T*
Δz_3	Width of interval in *B/T*
ε	Relaxation in the cost constraint

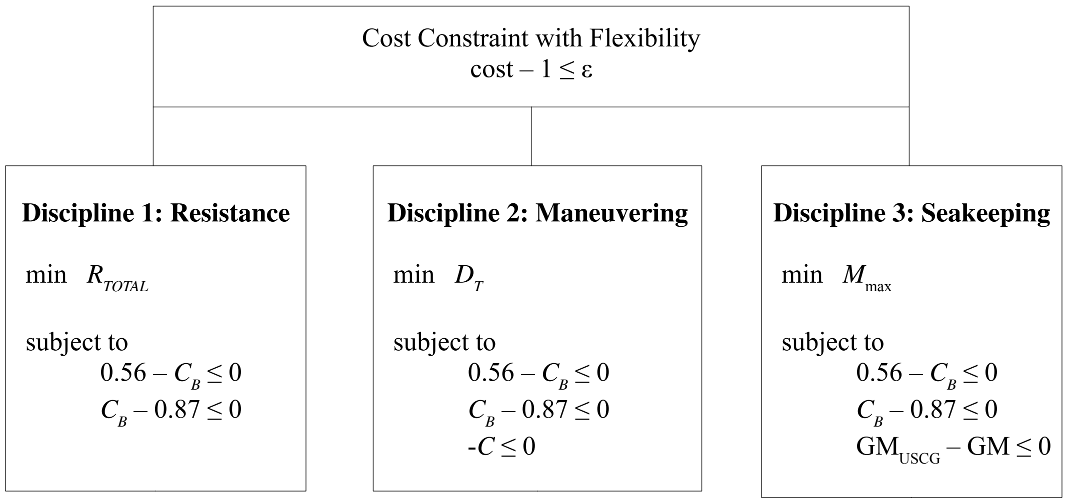


Fig. 7.3. Diagram of the ship design MDO problem.

7.2 Application of the Multidisciplinary Design Optimization Algorithm

The set-based design MDO algorithm is applied for analyzing the ship design problem. In order to demonstrate the benefits of the new algorithm, a single-point multi-objective optimization was conducted by considering the same design disciplines. the ship MDO problem was approached in two ways.

7.2.1 Results from the Set-Based Design MDO Analysis

The cost constraint from Equation (7.8) was included at the system level. The system optimization optimization statement for the ship design problem is

$$\min_{\mathbf{z}_{min},\, \Delta\mathbf{z},\, \varepsilon} \quad \exp\left(\Delta z_1 \cdot \Delta z_2 \cdot \Delta z_3\right) + \exp\left(\frac{\varepsilon}{\varepsilon_{\max}}\right) + \sum_{i=1}^{3} \exp\left(F_i\left(\mathbf{x}_i^{\text{new}}\right)\right) \tag{7.9}$$

$$\text{s. t.} \quad g_j(\mathbf{x}_i^{\text{new}}) \leq 0 \quad \text{for } j = 1, 2, 3, 4 \text{ and for } i = 1, 2, 3$$

$$g_{\text{cost}}(\mathbf{x}_i^{\text{new}}) \leq \varepsilon \quad \text{for } i = 1, 2, 3$$

The maximum relaxation of the constraint ε_{max} was set to 0.1.

The results for the optimization are tabulated in Table 7.6. The upper half of the table shows the bounds on the new, reduced design space. The lower half of the table shows the properties for the discipline optimizations when performed in the reduced design space. The resulting value for the relaxation of the cost constraint is $\varepsilon = 0.046$.

The results are also shown in a series of plots. Figure 7.4 shows the original design space with the reduced design space indicated by dashed lines. The individual discipline optima are marked with small dots and the discipline optima within the reduced space are marked with stars. Figures 7.5-7.7 show the same plot but with 2D projections so that it is easier to see the new design space boundaries.

Table 7.6. Results from the set-based design inspired MDO algorithm.

Reduced design space results			
	Initial	Optimal	
L min	250.00	302.12	
L max	320.00	314.86	
L/B min	5.60	6.66	
L/B max	8.00	7.02	
B/T min	2.25	2.69	
B/T max	3.75	3.08	
Properties of the discipline optima in the new design space			
	Discipline 1	Discipline 2	Discipline 3
L^*	314.86	309.99	310.23
L/B^*	6.66	6.66	6.66
B/T^*	2.69	2.69	2.70
f_i^*	8.1882×10^9	995.30	25.566

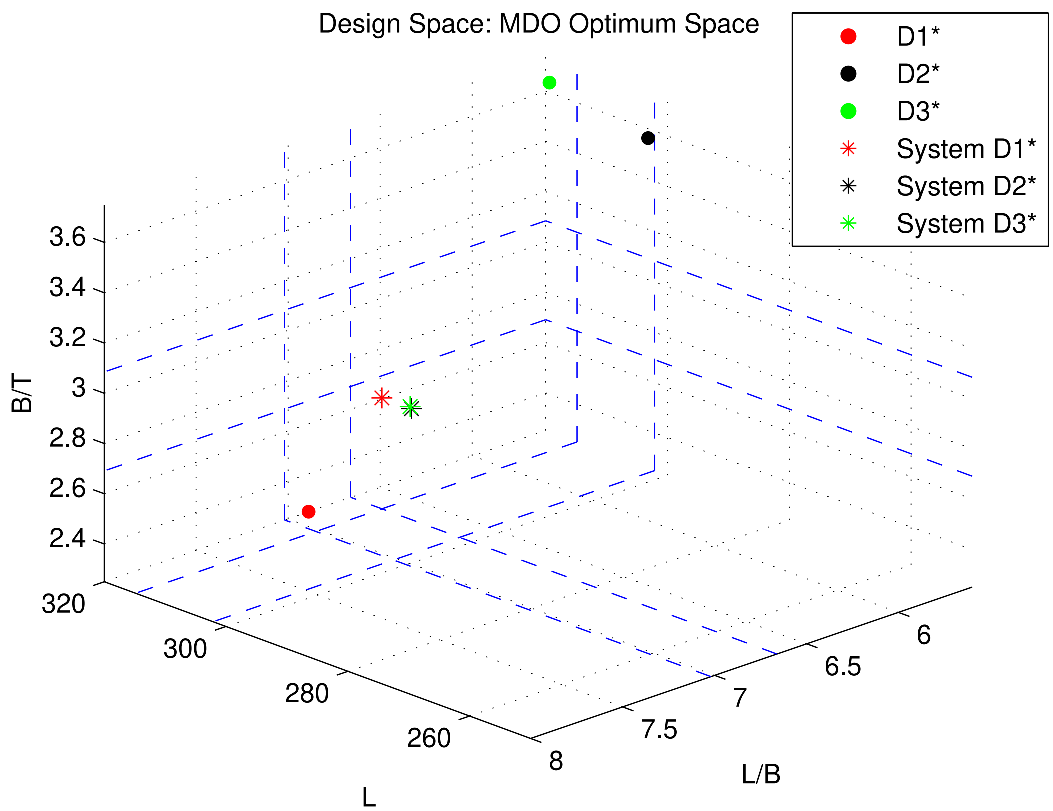


Fig. 7.4. Reduced design space returned by the set-based design MDO algorithm.

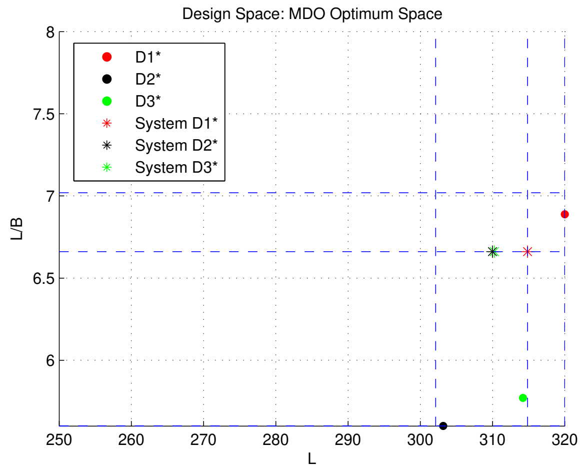


Fig 7.5. New design space viewed in the *L-L/B* plane.

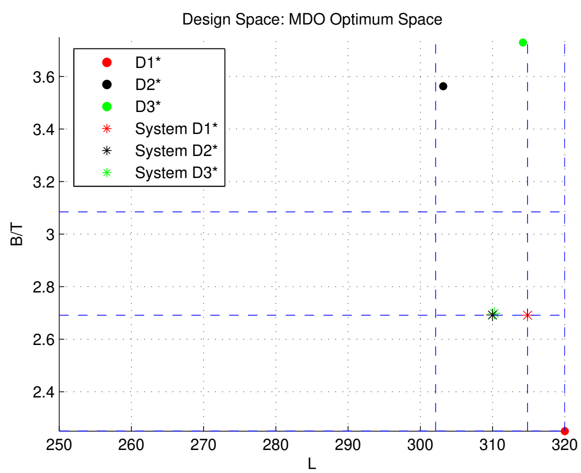


Fig 7.6. New design space viewed in the *L-B/T* plane.

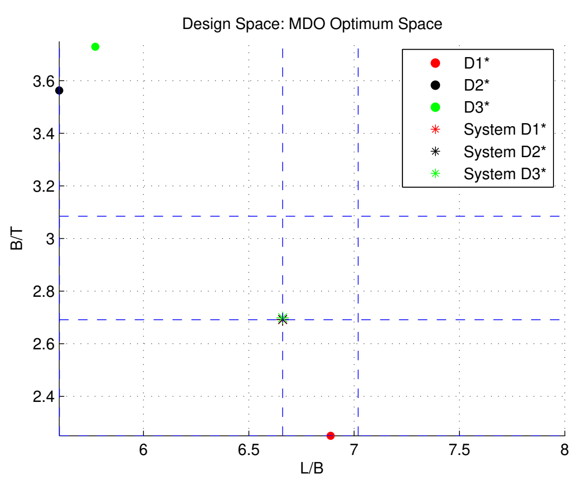


Fig 7.7. New design space viewed in the *L/B-B/T* plane.

The results for the application of the MDO algorithm to the ship design problem yielded a reduced design space, which is the goal of the algorithm. As expected, the discipline optima within the reduced design space (shown in the lower half of Table 7.6) are inferior to the individual discipline optima, due to the compromise between disciplines.

The relaxation in the constraint returned by the optimization was $\varepsilon = 0.046$. Relaxation of the cost constraint was expected because as shown in the single discipline optimizations (Table 7.4), much of the original design space did not satisfy the cost constraint, including the individual discipline optima. The MDO results are valuable because the algorithm seeks to allow only the minimum relaxation necessary. Even though ε_{max} was 0.1, the MDO algorithm was able to find a suitable solution without having to relax the constraint the maximum amount.

The set-based design MDO algorithm returns a reduced portion which represents preferred region of the original design space. The reduced space can be used as the design space for conducting a point-based optimization at later stages of the design process when the requirements are better defined.

7.2.2 Comparison to Point-Based Design

The results of the new set-based design MDO algorithm are compared to results from a single point design multiobjective optimization. The purpose of the comparison was to illustrate the advantages of the set-based design MDO algorithm by modeling flexibility in the constraints and by handling design requirements which may change during the design process. For the comparison, three different point-based optimizations were performed:

1. Point-based optimization without the cost constraint in the original design space.
2. Point-based optimization with the relaxed cost constraint in the original design space.
3. Point-based optimization without the cost constraint in the reduced design space.

Case 1: The ship MDO problem was studied using a point-based optimization approach (a weighted sum of the objective functions) in the original design space. The point-based optimization included all of the disciplines' constraints but did not include the cost constraint. The optimization statement is:

$$\min_{\mathbf{x}\in\chi} \sum_{i=1}^{3} F_i(\mathbf{x}) \tag{7.10}$$

$$\text{s. t.} \quad \mathbf{g}_{\text{MO}}(\mathbf{x}) \leq \mathbf{0}$$

where χ is the original design space defined in Table 7.1. F_i are the normalized objective functions, where the three disciplines objective functions (R_{TOTAL}, D_T, and $M_{\max}$) are scaled according to Equation (6.6). The vector $\mathbf{g}_{\text{MO}}$ contains the constraints from all three disciplines (a total of four constraints), but not the cost constraint. Omitting the cost constraint represents a situation where the design is initially driven by technical requirements and cost enters the decision-making process at a later stage.

The results of the multiobjective optimization are shown in Table 7.7 in the row labeled "Case 1," and the multiobjective optimization results are plotted with comparison to the individual discipline optima in Figure 7.8 (note that the plot is rotated to the same angle as Figure 7.4 for comparison).

Table 7.7. Comparison of the results for multiple point-based optimizations.

	L^*	L/B^*	B/T^*	f_1^*	f_2^*	f_3^*	$g_{\text{cost}}(\mathbf{x}^*)$
Case 2	314.91	5.6000	3.7473	11.963×10^9	989.2	23.296	0.081
Case 3	314.89	5.6000	3.7471	11.972×10^9	989.0	23.295	0.081
Case 4	314.00	6.6609	2.7000	8.999×10^9	1,057	26.917	0.029

The weighted sum approach is successful at locating one possible solution to the multiobjective problem. The multiobjective solution satisfies the four constraints from the disciplines (upper and lower bound on C_B, maneuvering stability, and GM). The performance of the three discipline objective functions at the multiobjective solution is inferior to the individual optima, as expected. However, upon visual inspection (Figure 7.8) the multiobjective solution may not be a good candidate for a solution because it is located in a corner of the design space; this does not illustrate much compromise between all three disciplines, and more importantly, leaves almost no room to adjust the design should any changes in requirements occur (for example, inclusion of a cost consideration).

Additionally, the cost constraint was not considered in the multiobjective optimization. Without accounting for cost, neither the discipline optima nor the multiobjective optimum satisfy the cost constraint; the value of the cost constraint is 0.081 at the multiobjective optimum (a value greater than zero indicates that the constraint is violated). Starting from this specific design point, it would be difficult to approach the problem with the cost constraint because none of the current solutions are feasible.

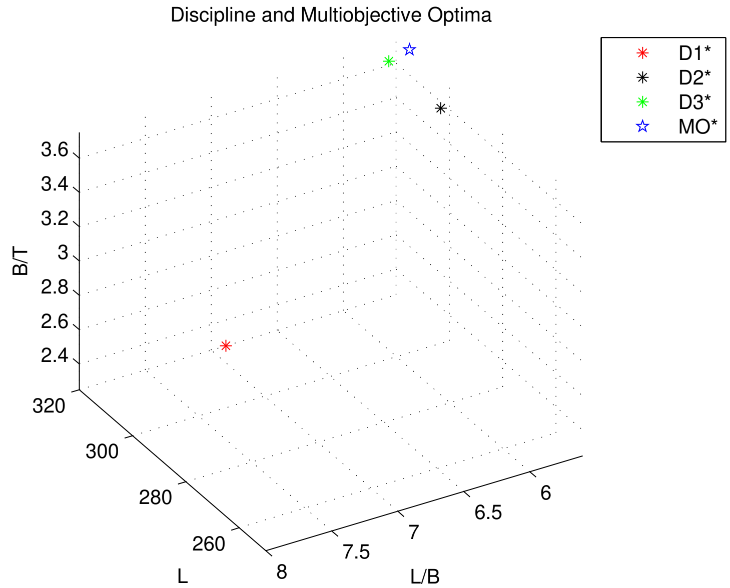


Fig. 7.8. Multiobjective optimization results.

Case 2: The ship MDO problem was next solved using a point-based optimization including the cost constraint in the original design space. The cost constraint was relaxed according to ε_{max} to model the situation that the designer has decided that the constraint may be violated by as much as ε_{max}; this is the same statement made for the set-based design MDO, except that the set-based design MDO algorithm can reduce the constraint relaxation during the optimization. The optimization statement is:

$$\begin{aligned} \min_{\mathbf{x}\in\chi} \quad & \sum_{i=1}^{3} F_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_{\text{MO}}(\mathbf{x}) \leq \mathbf{0} \\ & g_{\text{cost}}(\mathbf{x}) \leq \varepsilon_{\text{max}} \end{aligned}$$

where $\mathbf{g}_{\text{MO}}$ contains all of the constraints from the disciplines and χ indicates the original design space. This optimization problem represents the situation where the designer has encountered a new, strict constraint during the point-based design process and the constraint has been relaxed in order to locate feasible solutions.

The results for this point-based optimization are shown in Table 7.7, in the row labeled "Case 2." The solution is almost identical to the solution for Case 1; this result is expected because the solution to Case 1 yielded a cost constraint violation of 0.081, which is less than $\varepsilon_{max} = 0.1$. Addition of the cost constraint with the maximum relaxation of 0.1 has no effect on the solution because it is not active, and therefore the design from Case 2 has not incorporated any changes due to the cost requirements when compared to Case 1.

Case 3: To demonstrate the benefits of the set-based design MDO algorithm, a multiobjective optimization was conducted using the reduced design space determined by the set-based design MDO algorithm. The multiobjective optimization statement is

$$\min_{\mathbf{x} \in \chi_S} \sum_{i=1}^{3} F_i(\mathbf{x}) \tag{7.11}$$

$$\text{s. t.} \quad \mathbf{g}_{\mathrm{MO}}(\mathbf{x}) \leq \mathbf{0}$$

where χ_S is the reduced design space located by the set-based design MDO algorithm (listed in Table 7.6). The cost constraint was not included in the optimization statement. The results of the multiobjective optimization are summarized in Table 7.7 in the row labeled "Case 3;" the results are clearly very different compared to the multiobjective optimization in Table 7.7. The results are plotted in Figure 7.9 along with the contours of the cost constraint.

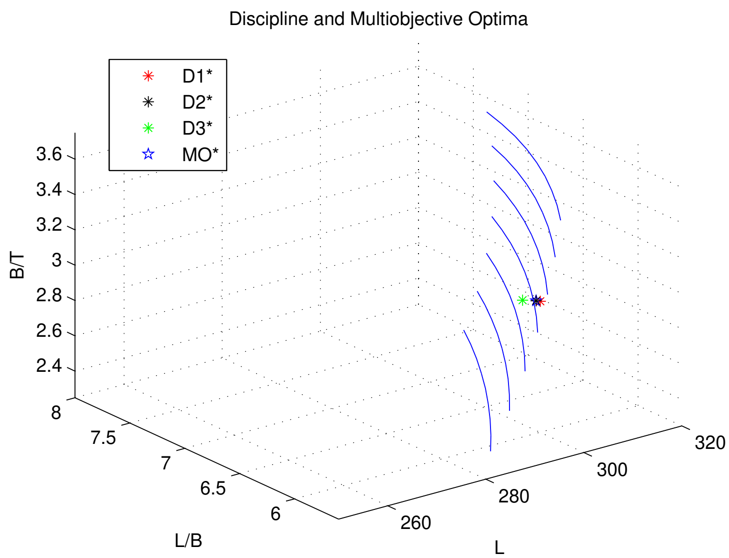


Fig. 7.9. Multiobjective optimization results for the new design space with contours of the cost constraint. The constraint is satisfied on the left side of the contours.

As Figure 7.9 shows, the discipline optima and the multiobjective optimum are closely grouped within the new design space; they are also located very close to the cost constraint boundary. Even though the cost constraint was not included in the multiobjective optimization, the value of the cost constraint at the multiobjective optimum is 0.029; not only is this a smaller constraint violation compared to the previous multiobjective optimization, but 0.029 is also less than the relaxation of the constraint from the MDO solution (ε = 0.046). Thus the single-point multiobjective optimization solution achieved by operating within the reduced design space satisfies the relaxed cost constraint even though the cost constraint was not included in the multiobjective optimization. This is accomplished because the cost constraint was accounted for when determining the reduced design space by the set-based design MDO algorithm.

7.3 Chapter Summary

This chapter presented an application of the set-based design MDO algorithm to a ship design problem. Resistance, maneuvering, and seakeeping design disciplines are considered along with a simple cost estimate. A reduced design space is identified by the set-based design MDO algorithm. The MDO solution also resulted in introducing relaxation to the cost constraint.

A multiobjective optimization is performed within the reduced design space located by the MDO algorithm, and the results are preferable to the results of a similar multiobjective optimization performed within the original design space. This ship design application illustrates that is it is valuable to first utilize a space-reducing technique (using sets to describe the design variables) before approaching a problem with a single point-based optimization. Furthermore, incorporating flexibility in the constraints of the set-based design MDO allows the optimization to handle a problem with very strict constraints in a rational manner and minimize the relaxation introduced in the constraints.

CHAPTER 8

Conclusion, Thesis Contributions, and Future Research

This dissertation discussed the topic of multidisciplinary design optimization (MDO), the optimization and systematic coordination of the exchange of information of a design problem with multiple discipline analysis.

The first contribution of this dissertation is a new multi-level MDO algorithm which was presented in Chapter 3. The MDO algorithm has two levels: a system level optimization and a discipline level optimization. The system level optimization is used to coordinate the discipline optimizations using target values for the design variables.

Next, this dissertation discussed two popular areas of MDO: optimization under uncertainty (Chapter 4) and surrogate-based optimization (Chapter 5). Optimization under uncertainty describes the process of optimization while accounting for variables and parameters that cannot be exactly predicted. Surrogate models are mathematical interpolation models that approximate the behavior of expensive functions with acceptable accuracy at a reduced computational cost.

Due to the variations in design variables and parameters with uncertainty, the response of the optimal design will differ from the deterministic expectation, which can lead to violation of constraints on the design and/or deterioration of the expected optimal performance. The research presented in this thesis implemented two techniques for addressing uncertainty (robustness and reliability) in the MDO algorithm using target values. The MDO algorithm using target values was applied to a conceptual ship design problem with uncertainty and it was demonstrated that the variability in the objective function was reduced and that the prescribed probability of not violating constraints at the optimum was met, while improving the discipline objective functions.

In surrogate-based optimization, surrogate models replace expensive functions during the optimization to reduce the computational time required for optimization. One method for surrogate modeling is Kriging, which has been shown to have good performance and flexibility. Kriging models were developed for three disciplines in a ship design problem: resistance, maneuvering, and seakeeping. This thesis demonstrated that the MDO algorithm using target values is capable of performing surrogate-based optimization using Kriging models. The optimization with Kriging models achieved a computational time savings with accurate predictions when compared to the true solvers.

A discussion and review of set-based design was presented in Chapter 6. Set-based design is a design methodology that seeks to offer improvements over the traditional point-based design approach. In set-based design, engineers from different disciplines determine sets of feasible values for their own analysis. Engineers share the set information to determine areas of feasible overlap, then gradually reduce the sets to focus on the feasible region.

The final contribution of this thesis is the development of a set-based design MDO algorithm. Greater flexibility is achieved by the set-based design MDO algorithm compared to single-point optimization. The new algorithm was defined in Chapter 6 and then applied to a ship design problem in Chapter 7. The ship design analysis included resistance, maneuvering, and seakeeping performances, along with a cost estimate. The ship design problem was solved using the set-based design MDO algorithm, which yielded a reduced design space. A multiobjective optimization was performed within the reduced design space identified by the set-based design MDO algorithm, and the results were preferable when compared to the results of a multiobjective optimization performed within the original design space. The ship design application illustrates that is it is valuable to first utilize a space-reducing technique before approaching a problem with a point-based optimization. Furthermore, incorporating flexibility in the constraints allows the optimization to handle a problem with very strict constraints in a rational manner and minimize the necessary constraint relaxation.

This research has inspired several questions for future research. While the areas of optimization under uncertainty and surrogate-based optimization have received much attention in the literature, there are still relatively few analytical methods for optimization in set-based design. More interpretations of optimization with set-based design are needed to fully explore this topic; this dissertation presents only one “translation” of the principles of set-based design into mathematical statements.

While other formulations for optimization with set-based design exist, it is very difficult to compare approaches because authors have different understandings of the design problem and the underlying goals and assumptions are different. Because the algorithm developed in this work is an MDO algorithm, the focus is on coordinating the exchange of information between disciplines. Other authors have formulated set-based design algorithms from the perspective of fuzzy set theory or multi-attribute utility theory, for example, where the focus is on designer or discipline preference information.

Therefore, it is valuable to develop an approach for comparing set-based design algorithms, both in terms of formulation and performance. The study of a common benchmarking problem is perhaps the most direct approach for comparing the performance of different algorithms, however, comparing the formulation of different approaches is a challenging problem.

Furthermore, the set-based design MDO algorithm presented in this dissertation can only work with continuous design variables, and examines continuous intervals of those design variables. Real design problems often include discrete design variables, which would be a valuable element to add to the algorithm. The capability to examine sets of the design variables which are disjoint would be beneficial when designers are interested in different areas of the design space.

REFERENCES

Acar, E., & Solanki, K. (2009). System reliability based vehicle design for crashworthiness and effects of various uncertainty reduction measures. *Structural and Multidisciplinary Optimization*, *39*, 311-325.

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

Ahn, J., & Kwon, J. H. (2006). An efficient strategy for reliability-based multidisciplinary design optimization using BLISS. *Structural and Multidisciplinary Optimization*, *31*, 363-372.

Allen, M., & Maute, K. (2004). Reliability-based design optimization of aeroelastic structures. *Structural and Multidisciplinary Optimization*, *27*, 228-242.

Athan, T. W., & Papalambros, P. (1996). A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, *27*, 155-176.

Avigad, G., & Moshaiov, A. (2010). Set-based concept selection in multi-objective problems involving delayed decisions. *Journal of Engineering Design*, *21*(6), 619-646.

Balling, R. J., & Sobieszczanski-Sobieski, J. (1996). Optimization of coupled systems: Overview of approaches. *AIAA Journal*, *34*(1), 6-17.

Bernstein, J. I., (1998). Design methods in the aerospace industry: Looking for evidence of set-based practices. Master's thesis, Cambridge, MA: Massachusetts Institute of Technology.

Besnard, E., Schmitz, A., Hefazi, H., & Shinde, R. (2007). Constructive neural networks and their application to ship multidisciplinary design optimization. *Journal of Ship Research*, *51*(4), 297-312.

Beyer, H. G., & Sendhoff, B. (2007). Robust optimization – A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, *196*, 3190-3218.

Braun, R. D., & Kroo, I. M. (1995). Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In N. Alexandrov N., & M. Y. Hussaini (Eds.), *Multidisciplinary design optimization: State of the art* (pp. 98-116). Philadelphia, PA: SIAM.

Chen, V. C. P., Tsui, K. L., Barton, R. R., & Meckesheimer, M. (2006). A review on design, modeling and applications of computer experiments. *IIE Transactions*, *38*(4), 273-291.

Choi, J. H., Lee, W. H., Park, J. J., & Youn, B. D. (2008). A study on robust design optimization of layered plate bonding process considering uncertainties. *Structural and Multidisciplinary Optimization*, *35*, 531-540.

Clarke, D., Gedling, P., & Hine, G. (1982). The application of manoeuvring criteria in hull design using linear theory. *Transactions RINA*, *124*, 45-68.

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

de Wit, A. J., & van Keulen, F. (2010). Overview of methods for multi-level and/or multi-disciplinary optimization. *Proceedings of the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material Conference*, 12 – 15 April 2010, Orlando, FL. AIAA Paper 2010-2914.

Delpiano, J., & Sepúlveda, M. (2006). Combining iterative heuristic optimization and uncertainty analysis methods for robust parameter design. *Engineering Optimization*, *38*(7), 821-831.

Der Kiureghian, A. (2005). First- and second-order reliability methods. In E. Nikolaidis, E., D. M. Ghiocel, & S. Singhal (Eds.), *Engineering design reliability handbook* (pp. 14-1 – 14-24). Boca Raton, FL: CRC Press.

Diez, M., Peri, D., Fasano, G., & Campana, E. F. (2012). Hydroelastic optimization of a keel fin of a sailing boat: a multidisciplinary robust formulation for ship design. *Structural and Multidisciplinary Optimization*. doi 10.1007/s00158-012-0783-7.

Ferguson, S., Kasprzak, E., & Lewis, K. (2009). Designing a family of reconfigurable vehicles using multilevel multidisciplinary design optimization. *Structural and Multidisciplinary Optimization*, *39*, 171-186.

Forrester, A. I. J., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, *45*, 50-79.

Forrester, A., Sóbester, A., & Keane, A. (2008). *Engineering design via surrogate* *modelling: A practical guide*. Hoboken, NJ: John Wiley & Sons.

Gale, P. A. (2003). The ship design process. In T. Lamb (Ed.), *Ship Design and Construction* (pp. 5-1 – 5-40). Jersey City, NJ: Society of Naval Architects and Marine Engineers.

Geyer, P. (2009). Component-oriented decomposition for multidisciplinary design optimization in building design. *Advanced Engineering Informatics*, *23*, 12-31.

Glaz, B., Friedmann, P. P., & Liu, L. (2008). Surrogate based optimization of helicopter rotor blades for vibration reduction in forward flight. *Structural and Multidisciplinary Optimization*, *35*, 341-363.

Goh, C. K., Tan, K. C., Cheong, C. Y., & Ong, Y. S. (2010). An investigation on noise-induced features in robust evolutionary multi-objective optimization. *Expert Systems with Applications*, *37*, 5860-5980.

Grujicic, M., Arakere, G., Pandurangan, B., Sellappan, V., Vallejo, A., & Ozen, M. (2010). Multidisciplinary design optimization for glass-fiber epoxy-matrix composite 5 MW horizontal-axis wind-turbine blade. *Journal of Materials Engineering and Performance*, *19*(8), 1116-1127.

Haldar, A., & Mahadeval, S. (2000). *Probability, reliability, and statistical methods in engineering design*. New York, NY: John Wiley & Sons.

Hart, C., & Vlahopoulos, N. (2010). An integrated multidisciplinary particle swarm optimization approach to conceptual ship design. *Structural and Multidisciplinary Optimization*, *41*(3), 481-494.

Hefazi, H., Mizine, I., Schmitz, A., Klomparens, S., & Wiley, S. (2010). Multidisciplinary synthesis optimization process in multihull ship design. *Naval Engineers Journal*, *3*, 29-47.

Holtrop, J. (1984). A statistical re-analysis of resistance and propulsion. *International Shipbuilding Progress*, *31*(363), 272-276.

Holtrop, J., & Mennen, G. G. J. (1982). An approximate power prediction method. *International Shipbuilding Progress*, *29*(335), 166-170.

Hunt, E. C., & Butman, B. S. (1995). *Marine engineering economics and cost analysis*. Centreville, MD: Cornell Maritime Press.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*, 345-383.

Jouhaud, J. C., Sagaut, P., Montagnac, M., & Laurenceau, J. (2007). A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil. *Computers and Fluids*, *36*, 520-529.

Kalavalapally, R., Penmetsa, R., & Grandhi, R. (2006). Multidisciplinary optimization of a lightweight torpedo structure subjected to an underwater explosion. *Finite Elements in Analysis and Design*, *43*, 103-111.

Kim, H. M, Michelena, N. F., Papalambros, P. Y., & Jiang, T. (2003). Target cascading in optimal system design. *Transactions of the ASME*, *125*, 474-480.

Kleijnen, J. P. C. (1998). Experimental design for sensitivity analysis, optimization, and validation of simulation models. In J. Banks (Ed.), *Handbook of simulation* (pp. 173-233). New York, NY: Wiley.

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, *192*, 707-716.

Kovach, J., & Cho, B. R. (2008). Development of a multidisciplinary-multiresponse robust design optimization model. *Engineering Optimization*, *40*(9), 805-819.

Kuhn, H. W., & Tucker, A. (1951). Nonlinear programming. *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA: University of California Press.

Lee, J. (1996). *Set-based design systems for stampings and flexible fixture workspaces*. Doctoral Dissertation, Ann Arbor, MI: University of Michigan.

Leifsson, L., Ko, A., Mason, W. H., Schetz, J. A., Grossman, B., & Haftka, R. T. (2011). Multidisciplinary design optimization of blended-wing-body transport aircraft with distributed propulsion. *Aerospace Science and Technology*. doi 10.1016/j.ast.2011.12.004.

Liang, J., Mourelatos, Z. P., & Tu, J. (2008). A single-loop method for reliability-based design optimisation. *International Journal of Product Development*, *5*(1/2), 76-92.

Liker, J. K., Sobek, D. K. II, Ward, A. C., & Cristiano, J. J. (1996). Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering. *IEEE Transactions on Engineering Management*, *43*(2), 165-178.

Madhavan, K., Shahan, D., Seepersad, C., Hlavinka, D. A., & Benson, W. (2008). An industrial trial of a set-based approach to collaborative design. DETC2008/49953. *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 3-5, 2008, New York, NY.

Malak, R. J. Jr., Aughenbaugh, J. M., & Paredis, C. J. J. (2009) Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design*, *41*, 214-227.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*, 369-395.

Martin, J. D., & Simpson, T. W. (2005). Use of kriging models to approximate deterministic computer models. *AIAA Journal*, *43*(4), 853-863.

Maute, K., & Frangopol, D. M. (2003). Reliability-based design of MEMS mechanisms by topology optimization. *Computers and Structures*, *81*, 813-824.

Nahm, Y. E., & Ishikawa, H. (2006). Novel space-based design methodology for preliminary engineering design. *International Journal of Advanced Manufacturing Technology*, *28*, 1056-1070.

Noor, A. K. (2005). Perspectives on nondeterministic approaches. In E. Nikolaidis, E., D. M. Ghiocel, & S. Singhal (Eds.), *Engineering design reliability handbook* (pp. 2-1 – 2-19). Boca Raton, FL: CRC Press.

Onwubiko, C. (2000). *Introduction to engineering design optimization*. Upper Saddle River, NJ: Prentice Hall.

Papalambros, P. & Wilde, D. J. (2000). *Principles of optimal design: Modeling and computation* (2nd ed.). Cambridge, England: Cambridge University Press.

Park, S. H. (1996). *Robust design and analysis for quality engineering*. London: Chapman & Hall.

Park, E. J., da Luz, L. F., & Suleman, A. (2008). Multidisciplinary design optimization of an automotive magnetorheological brake design. *Computers and Structures*, *86*, 207-216.

Parsons, M. G. (2003). Parametric design. In T. Lamb (Ed.), *Ship design and construction* (pp. 11-1 – 11-48), Jersey City, NJ: Society of Naval Architecture and Marine Engineers,

Parsons, M. G., Li, J., & Singer, D. J. (1998). *Michigan conceptual ship design software environment – User's manual*, Report No. 338, Ann Arbor, MI: University of Michigan Department of Naval Architecture and Marine Engineering.

Parsons, M. G., & Scott, R. L. (2004). Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods. *Journal of Ship Research*, *48*(1), 61-76.

Quiepo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P. K.

(2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, *41*, 1-28.

Raff, A. I. (1972). *Program SCORES – Ship structural response in waves.* Report SSC-230. Washington, DC: Ship Structure Committee.

Rao, S. (2009). *Engineering optimization: Theory and practice* (4th ed.). Hoboken, NJ: Wiley.

Ray, T., & Smith, W. (2006). A surrogate assisted parallel multiobjective evolutionary algorithm for robust engineering design. *Engineering Optimization*, *38*(8), 997-1011.

Reale-Levis, F. D., Romero, V. J., & Swiler L. P. (2008). An efficient robust design strategy employing surrogate variance minimization and adaptive mean constraint satisfaction. AIAA Paper 2008-2293. *Proceedings of the 10th AIAA Non-Deterministic Approaches Conference*, April 7-10, 2008, Schaumberg, IL.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, *4*(4), 409-435.

Sasena, M. J. (2002). *Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations*. Doctoral Dissertation, Ann Arbor, MI: University of Michigan.

Scher, R. M., & Benford, H. (1980). *Some aspects of fuel economy in bulk carrier design and operation*, Report No. 228, Ann Arbor, MI: University of Michigan Department of Naval Architecture and Marine Engineering.

Schittkowski, K., & Yuan, Y. X. (2011). Sequential quadratic programming methods. In J. J. Cochran (Ed.), *Wiley encyclopedia of operations research and management science*. Hoboken, NJ: Wiley.

Schuëller, G. I., & Jensen, H. A. (2008). Computational methods in optimization considering uncertainties – An overview. *Computer Methods in Applied Mechanics and Engineering*, *198*(1), 2-13.

Shahan, D., & Seepersad, C. C. (2009). Bayesian networks for set-based collaborative design. DETC2009/85741. *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 30 – September 2, 2009, San Diego, CA.

Sen, P., & Yang, J. B. (1998). *Multiple criteria decision support in engineering design*. London: Springer.

Simpson, T. W., Mauery, T. M., Korte, J. J., & Mistree, F. (1998). Comparison of response surface and Kriging models for multidisciplinary design optimization. AIAA Paper 98-4755. *Collection of Technical Papers. Pt. 1*, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 2 – 4, 1998, St. Louis, MO.

Simpson, T. W., Peplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, *17*, 129-150.

Singer, D. J., Doerry, N., & Buckley, M. E. (2009). What is set-based design? *Naval Engineers Journal*, *121*(4), 31-43.

Sinha, K. (2007). Reliability-based multiobjective optimization for automotive crashworthiness and occupant safety. *Structural and Multidisciplinary Optimization, 33*, 255-268.

Sobek, D. K. II, Ward, A. C., & Liker, J. K. (1999). Toyota's principles of set-based concurrent engineering. *Sloan Management Review*, *40*(2), 67-83.

Sobieszczanski-Sobieski, J. (1988). Optimization by decomposition: A step from hierarchic to non-hierarchic systems. NASA-TM-101494. Hampton, VA: NASA Langley Research Center.

Sobieszczanski-Sobieski, J. (1989). Multidisciplinary optimization for engineering systems: Achievements and potential. In H. W. Bergmann (Ed.), *Optimization: Methods and applications, possibilities and limitations* (pp. 42-62). Berlin: Springer-Verlag.

Sobieszczanski-Sobieski, J., Agte, J. S., & Sandusky, R. R. Jr. (1998). Bi-level integrated system synthesis (BLISS), Report NASA/TM-1998-208715, Hampton, VA: NASA Langley Research Center.

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

Song, X. G., Jung, J. H., Son, H. J., Park, J. H., Lee, K. H., & Park, Y. C. (2010). Metamodel-based optimization of a control arm considering strength and durability performance. *Computers and Mathematics with Applications*, *60*, 976-980.

Steenackers, G., Guillaume, P., & Vanlanduit, S. (2009). Robust optimization of an airplane component taking into account the uncertainty of the design parameters. *Quality and Reliability Engineering International*, *25*, 255-282.

Taguchi, G. (1986). *Introduction to quality engineering: Designing quality into products and processes*. Dearborn, MI: Asian Productivity Organization.

Taylor, D. W. (1915). Calculations for ships' forms and the light thrown by modern experiments upon resistance, propulsion and rolling of ships. *Transactions of the International Engineering Congress, 1915: Naval Architecture and Marine Engineering*, September 20-15, 1915, San Francisco, CA.

Tedford, N. P., & Martins, J. R. R. A. (2010). Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, *11*, 159-183.

Wan, X. (2004). *Simulation based optimization with surrogate models.* Doctoral dissertation, West Lafayette, IN: Purdue University.

Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Transactions of the ASME*, *129*, 370-380.

Wang, J., & Terpenny, J. (2003). Interactive evolutionary solution synthesis in fuzzy set-based preliminary engineering design. *Journal of Intelligent Manufacturing*, *14*, 153-167.

Wang, W. M., Peng, Y. H., Hu, J., & Cao, Z. M. (2009). Collaborative robust optimization under uncertainty based on generalized dynamic constraints network. *Structural and Multidisciplinary Optimization*, *38*, 159-170.

Ward, A., Liker, J. K., Cristiano, J. J., & Sobek, D. K. II. (1995). The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, *36*(3), 43-61.

Watson, D. G. M. (1998). *Practical ship design.* Amsterdam: Elsevier.

Xuebin, L. (2009). Multiobjective optimization and multiattribute decision making study of ship's principal parameters in conceptual design. *Journal of Ship Research*, *53*(2), 83-99.

Yang, R. J., Chuang, C., Guo, L., & Li, G. (2005). Experience with approximate reliability-based optimization methods II: An exhaust system problem. *Structural and Multidisciplinary Optimization*, *29*, 488-497.

Yang, Y. S., Park, C. K., Lee, K. H., & Suh, J. C. (2007). A study on the preliminary ship design method using deterministic approach and probabilistic approach including hull form. *Structural and Multidisciplinary Optimization*, *33*, 529-539.

Yao, W., Chen, X., Luo, W., van Tooren, M., & Guo, J. (2011). Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. *Progress in Aerospace Sciences*, *47*, 450-479.

Zhao, D., & Xue, D. (2010). A comparative study of metamodeling methods considering sample quality merits. *Structural and Multidisciplinary Optimization*, *42*, 923-938.

Zhao, M., & Cui, W. (2011). On the development of Bi-Level Integrated System Collaborative Optimization. *Structural and Multidisciplinary Optimization*, *43*, 73-94.