

Addition of Features to an Existing MDO Model for Containerships

by

Amlan Dasgupta

Virginia Polytechnic Institute and State University

Thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

In

Ocean Engineering

Dr. Wayne Neu, Chairman

Dr. William H. Mason

Dr. Alan Brown

May 2001

Blacksburg, Virginia.

Keywords: MDO, Containership, Optimization

Abstract

Traditionally, the “Design Spiral” is used for the design of ships. The design spiral endorses the concept that the design process is sequential and iterative. Though this procedure was very effective over the years, the current trend of engineering demands that more stress be put on the exploration of optimum design. With the advancement of computing technologies, the onus has shifted from finding better calculation schemes to formulating an economically viable design scheme. One of the objects of the FIRST project funded by MARITECH was to develop a computer tool to give the best ship design using optimization techniques.

To achieve that goal, the team developed a package that consists of three modules: optimization, geometric and a performance evaluation module. A prototype MDO tool is developed based on Microsoft’s COM framework using ATL. With this design, the modules can be modified with minimum programming effort. A geometric shape manipulation scheme is developed in which the hull form was generated by blending two hull forms. This MDO tool is used to design a container ship with the required freight rate as the objective to be minimized.

A B/D constraint is applied to the design to get a better structural design. The number of tiers on deck (NT_d) is made a design variable to enable the optimizer to have the flexibility of manipulating the cargo carrying capacity. Ballast has also been added as a design variable to reduce the center of gravity of the ship increasing the GM of the ship. This feature enables the optimizer to carry greater cargo on deck improving the objective function.

An effort is made to analyze the efficacy of the MDO tool by varying various parameters that affect the design. Technology factors have been introduced which give an insight on effect of key parameters. The effect of another structural constraint L/D was also investigated. This constraint tends to bring down the overall length and is inconclusive in its results. Further analysis of this constraint is needed to draw usable conclusions. The linear response surface approximation was eliminated and the original stepwise discontinuous TEU capacity function is employed in the later examples.

Contents

	Nomenclature	3
	List of Figures	7
	List of Tables	7
1	Introduction	8
	1.1 Literature Review and Past Work	9
	1.2 Introduction to Current Work	10
2	Organization of the Project	12
	2.1 Design Overview	12
	2.2 Component Model	12
	2.3 Software Overview	14
3	Problem Formulation	18
	3.1 Objective Function	18
	3.2 Design Variables	19
	3.3 Constraints	20
	3.4 Constraint Tolerance	22
	3.5 Scaling	23
	3.6 Normalization of Constraint functions	23
	3.7 Hull Geometry Manipulation	24
4	A New Constraint	26
	4.1 Background	26
	4.2 Formulation of a new constraint	26
	4.3 Evaluation of Constraint	27
5	Number of Tiers on Deck	31
	5.1 Number of Containers	31
	5.2 Effect of B/D Constraint	33

5.3	NT _d as a Design Variable	37
5.4	Effect of NT _d as Design Variable	38
6	Ballast	43
6.1	Need for Ballast	43
6.2	Another Viable Alternative	44
6.3	Effect on Center of Gravity	45
6.4	Ballast as a Design Variable	46
6.5	Effect of ballast on the Design	47
6.6	Ballast and NT _d	49
7	Tools for evaluating the results of Optimization	54
7.1	Introduction	54
7.2	Alpha Plots	54
7.3	Sensitivity Analysis	56
7.4	Restart Option	59
8	Technology Factors	60
8.1	Introduction	60
8.2	Applications of Technology factors that affect fixed costs	61
8.3	Application of Technology Factors Affecting Operation Costs	65
9	Examples of Results of Optimization	70
9.1	Examples on variations in key parameters	71
9.2	Examples with L/D constraint	72
9.3	Secondary Optimization	73
10	Conclusions and Future Work	77
	References	79

Nomenclature

Symbol	Description	Units
A_{ac}	Annual Average Cost	Dollars
A_{bc}	Annual building Cost	Dollars
A_{BT}	Transverse Area of bulbous bow at the position where the still water surface intersects the stem	Meter ²
A_{fc}	Annual fuel cost	Dollars
A_{oc}	Annual operating Costs	Dollars
B	Beam, maximum	Meters
Ballast	Ballast	Metric tons
BHP	Brake Horse Power	
C_b	Block Coefficient	
C_m	Midship Section Coefficient	
C_p	Prismatic coefficient	
C_{sh}	Cost of hull steel material	Dollars / metric ton
C_{wp}	Water plane Coefficient	
C_1 and C_2	Blending Coefficients.	
D	Depth, at side	Meters
DBH	Double Bottom Height	Meters
Disp	Displacement, volume	Meter ³
Displ	Displacement, weight	Metric tons
Dst	Range	Nautical miles
EHP	Effective Horse Power	
F_{cost}	Cost of bunker C fuel oil	Dollars / metric ton
Freeboard	Freeboard	Meters
FreeBoard _{min}	Minimum freeboard required	Meters
GM	Metacentric height	Meters
GM _{min}	Minimum metacentric height	Meters

Symbol	Description	Units
KG	Vertical center of gravity, total	Meters
KG _f	Vertical center of gravity, fuel	Meters
KG _m	Vertical center of gravity, machinery	Meters
L _{bp}	Length between perpendiculars; in this work, it is approximated as L _{oa} / 1.05	Meters
L _{cb}	Longitudinal center of buoyancy forward of 0.5 L expressed as percentage of length	
L _{oa}	Length, overall	Meters
L _{wl}	Length, waterline	Meters
N _{cranes}	Number of cranes	
N _{cranes_i}	Number of cranes, integer	
N _{cranes_f}	Number of cranes, continuous	
N _{crew}	Number of crew	
N _t	Number of round trips per year	
NT _b	Number of tiers below deck	
NT _d	Number of tiers above deck	
P _{wt}	Port Waiting Time	Days
RFR	Required Freight Rate	Dollars / nautical mile / metric ton
ROI	Return on Investment expressed as percentage	
R _t	Resistance	Newtons
SFC	Specific fuel consumption	Gms/SHP/hr
SHP	Maximum continuous shaft horse power	
T	Draft, design	Meters
TEU	Twenty Foot Equivalent Units	
TEU _f	Total Number of TEUs, continuous	
TEU _i	Total Number of TEUs, integer	
TEU _{bd_f}	Number of TEUs below deck, continuous	
TEU _{bd_i}	Number of TEUs below deck, integer	

Symbol	Description	Units
TEU_{d_f}	Number of TEUs on deck, continuous	
TEU_{d_i}	Number of TEUs on deck, integer	
T_{roll}	Rolling Period	Seconds
T_{roll_min}	Minimum rolling period	Seconds
V_k	Speed	Knots
W_{cargo}	Weight, cargo	Metric tons
W_f	Weight, fuel	Metric tons
W_m	Weight, machinery	Metric tons
W_s	Weight, hull steel	Metric tons

User Inputs

Symbol	Description	Default
Chargerate	Rate charged by the ship owner to ship cargo in dollars / ton / nautical mile	0.0064
C_{hull}	Cost of hull engineering materials in dollars / ton	3500
C_{of}	Cost of outfit material in dollars / ton	1500
C_{sh}	Cost of steel hull in dollars / ton	400
DBH	Double bottom height in meters	1.83
D_{st}	Range in nautical miles	7000
F_{cost}	Cost of Bunker C fuel oil in dollars / ton	80
HCH	Hatch coaming height in meters	1.83
I_r	Interest rate in percentage	0.08
L_c	Labor cost in dollars / hour. For simplicity, the same labor cost is assumed for hull steel, hull engineering, outfit and machinery	20
O_t	On-hire time per year in days	350
P_{wt}	Port waiting time in days	2
SFC	Specific fuel consumption in gms / shp / hour	120
S_l	Ship life in years	20
TSLU	TEUs loaded / Unloaded per day per crane	1440
W_{pc}	Weight per TEU in metric tons	12

List of Figures

- Figure 2.1** Flow Chart of the MDO Project
- Figure 2.2** Organization of the Analysis modules.
- Figure 2.3** General Construction of the Software system.
- Figure 4.1** B/D ratio from different ships and from results of MDO.
- Figure 4.2** L/D ratio from different ships and from results of MDO.
- Figure 4.3** T/D ratio from different ships and from results of MDO.
- Figure 4.4** L/B ratio from different ships and from results of MDO.
- Figure 5.1** Illustration of container capacity function and linear response surface
- Figure 5.2** Variation of Normalized parameters with NT_d
- Figure 5.3** Variation of Design parameters with respect to NT_d .
- Figure 5.4** Variation of NT_d with beam
- Figure 6.1** Ballast / displacement from several ship data
- Figure 6.2** Variation of normalized parameters with respect to Ballast
- Figure 6.3** Variation of normalized parameters with respect to ballast.
- Figure 6.4** Ratio of ballast to displacement
- Figure 7.1** Alpha Plot of constraint g_3 (GM constraint) indicating discontinuity.
- Figure 7.2 a-h** Sensitivity of RFR to Design Variables
- Figure 8.1** Sensitivity of RFR to dimensions of the ship. A value of 1 for the normalized design values indicates the optimum point
- Figure 8.2** Sensitivity of RFR to C_{sh}
- Figure 8.3** Sensitivity of RFR with W_s Technology factor
- Figure 8.4** Sensitivity of RFR to speed
- Figure 8.5** Sensitivity of RFR to η .
- Figure 8.6** Variation of RFR with N_{crane} .
- Figure 9.1** Cube illustrating the process of secondary optimization

List of Tables

Table 3.1	List of design variables
Table 3.2	Container Ship Constraints
Table 3.3	Scaling Factor for Design Variables used in DOT
Table 3.4	Normalization of constraints for DOT
Table 4.1	Evaluation of B/D Constraint
Table 5.1	Design parameters for various NT_d
Table 5.2	Variation of Design Parameters with respect to NT_d
Table 5.3	Variation of NT_d with beam
Table 6.1	Effect of ballast on the length and RFR with constant NT_d
Table 6.2	Variation of parameters with respect to ballast.
Table 7.1	Change in Design variables and constraints between two design points.
Table 7.2	Optimized values of Design variables used in Figure 7.2
Table 7.4	Example of restart option.
Table 8.1	Variation of RFR with C_{sh} .
Table 8.2	Sensitivity of parameters to Technology Factor for W_s
Table 8.3	Variation of parameters with η
Table 8.4	Results of optimization of varying N_{crane}
Table 9.1	Initial Design Variables and Bounds.
Table 9.2	Design variable values and various ship characteristics at the optimum point for the first set of examples.
Table 9.3	Results of optimization with L/D constraint

Chapter 1: Introduction

Traditionally, the “Design Spiral” is used for the design of ships. The design spiral endorses the concept that the design process is sequential and iterative. Though this procedure was very effective over the years, the current trend of engineering demands that more stress be put on the exploration of optimum design. With the advancement of computing technologies, the onus has shifted from finding better calculation schemes to formulating an economically viable design scheme.

It is not right to say that the design spiral is obsolete. In fact, with the addition of an optimization tool to design, the word “spiral” is replaced by “loop”. We still go through the process of old fashioned design but now we optimize between the resources to get a ship, which satisfies everybody and without too many compromises. Somehow, this sounds as if it is the most natural thing to do. But this process is fraught with its own unique set of considerations. A ship is practically a floating city. So, the list of design parameters that can be worked upon for an optimum point –the golden ship - can go on and on.

One of the objects of the FIRST project funded by MARITECH was to develop a computer tool to give the best ship design using optimization techniques. This was entrusted to the Department of Aerospace and Ocean Engineering at Virginia Polytechnic Institute and State University in Blacksburg, Virginia. A container ship was chosen as the test case. Typically bulk carriers, tankers and containerships consist of 90% of the world’s commercial fleet. The problem was tackled from an owner’s point of view. Hence, the required freight rate was chosen as the objective.

To achieve that goal, the team developed a package that consists of three modules: optimization, geometric and a performance evaluation module. Though these modules are essentially independent, the user has control over an overall manager. He can change the initial value of design parameters, set bounds and vary constraint bounds as per his needs. Though he does not know what goes on behind the user interface, he still feels secure with the design process because he has overall control. This sense of security breaks down when he has access to limited variables and constraints.

The team is constantly faced with the dilemma of describing the amount of information the user needs to control. A huge list of design variables and constraints

makes it cumbersome for the user and the computer. A smaller list will make him uncomfortable. So, the team is constantly experimenting on the optimum set of design variables, constraints and design parameters. What follows is a logical course that was undertaken during the course of this work to add a constraint and two design variables to the existing MDO code.

1.1 Literature Review and Past Work

Ray et al. [1] investigate a multidisciplinary optimization approach to container ship design treating the number of containers and speed as fixed parameters in the design process. A characteristic of this approach was the unit method formulation. In the unit approach, each formulation could be considered as one unit independent of the other. An output of one unit could be used as the input of another. A set of design variables was identified using this approach. The objective function was defined as the weighted average of the building cost, power and steel weight of the ship. Three types of constraints have been specified: constraints specified by the owner, performance constraints and constraint related to the dimensions of the ship. The results obtained by this formulation and using a non-linear multi variable optimization tool could only arrive at a local optimum.

Sen [2] dealt with the aspect of multiple criteria decision making process and demonstrated its use in modeling design synthesis and design selection situations. As ship design involves multi-criteria and both “hard” and “soft” constraints, classical optimization techniques cannot be used because they deal effectively with “hard” constraints only. An MCDM approach uses three basic methods to achieve a solution: weighting methods, prioritizing and efficient solution methods. Deviation variables are used as they guarantee using design variables and constraints in the same form. The objective function should be normalized for numerical stability. A Ro-Ro vessel was used as an example. Sequential Linear Programming, Sequential Quadratic Programming and Powell’s direct search method were used as the basic non-linear optimization tools. Eleven design variables were used and the constraints either set limits on dimensional ratios, stability constraints and other technical requirements. The goal constraints were obtaining the ship capacity, minimizing the required freight rate and minimizing the

weight of ballast. The problem was solved by using generalized goal programming method including several linear and non-linear MCDM models.

Keane et al. [3] considered the use of optimization techniques and an integrated computational approach to ship concept design. The method made use of accepted naval architecture principles, a sophisticated data base handler and several optimization procedures. The system comprised of a design control module, an optimizer, some design modules, a data base handler and auxiliary modules. However, some key aspects including cost, sea keeping, structures etc., were not represented.

These references formed the basis for the development of the MDO project. Recent trends in the ship design process indicate a strong inclination towards the use of optimization routines. The problem was divided into units to identify design variables and analysis routines for the different aspects of ship design. Constraints have been identified that relate to geometric dimensions, stability criteria, ship capacity and powering requirements. Basic optimization routines have been used to design the ship.

Vikram Ganesan [4], a former member of the research group explored the aspect of having the number of containers and speed as a variable parameter. Varying the number of containers with the dimensions of the ship and setting the speed as a design variable have accomplished this. The thesis provides a comprehensive analysis on building and operation costs for a container ship. These principles have been implemented as analysis tools to develop the software model to design container ships.

Ying Chen [5], another member of the team has investigated aspects concerning weight formulation, resistance, stability and cost estimates. These have been put into practice as analysis modules that form the basis on which the tool was. These modules have been developed by using empirical and experimental results that conform to existing ship data.

1.2 Introduction to Current Work

With this thesis, further advances have been made to the software tool that was developed by the FIRST team. An additional constraint is imposed that helps the user to ensure structural stability. Two new design variables have been introduced that allow the optimizer to have the flexibility to determine the cargo to be carried. Tools have been

developed which help to assess the results of optimization. Investigations on factors that affect the results have been provided.

Chapter 2 discusses the overall organization of the project. The chapter provides an insight into the structure of the analysis modules and the user interface. A flowchart discusses the solution methodology and a wiring diagram indicates the data flow in the analysis modules. A diagram illustrates the process flow with respect to the modules.

Chapter 3 deals with the problem formulation of this software tool. The design variables and constraints are discussed in this chapter. A new structural constraint is discussed in chapter 4. With the addition of this constraint, the ships obtained from the results of optimization are more consistent with stable ships. Figures also illustrate the need for such a constraint and the results of adding this constraint.

Chapters 5 and 6 deal with the inclusion of two new design variables. Number of tiers on deck and ballast have been added as design variables. Examples show the effect of each variable on the overall design independently and together. Three tools have been included into the design process that help evaluate the results of optimization. Insight into them is provided in Chapter 7.

Chapter 8 discusses the effect of technology factors on the overall design. Certain parameters can be varied to improve the design based on the advances in technology. Examples of such factors and their effect on the design are shown.

Three different sets of examples are discussed in chapter 9. These examples illustrate the flexibility of the design process. The sets of examples include the effect of key parameters on the design, the investigation of another structural constraint and the effect of using a stepwise discontinuous TEU capacity function in place of a linear response function.

Chapter 10 concludes by summarizing the salient points of this thesis and also provides direction and scope for further advances in the software tool.

Chapter 2: Organization of the Project

2.1 Design Overview:

This project involves the development of a software tool for the multi-disciplinary optimization (MDO) of a container ship. This design process is multi-disciplinary in nature because various modules like economics, geometry, hydrostatics etc. are taken into account in the design. As this is a software tool, special attention is given to the user interface. This interface is provided to the user to change the list of initial values and bounds for the design variables, toggle constraints and specify their limiting values. The interface can also be used to change the optimization and performance parameters. There is flexibility to change various design parameters that are being used in empirical relations in the performance evaluation. The interface also has buttons that can help in viewing the results of the output. The outputs include the final results of optimization and the iteration histories of the design variables and various design parameters.

2.2 Component Model:

This software tool is basically divided into three types of components: the optimization components, the geometric components and the performance evaluation components. The idea behind the component models is that they can be independent of each other if necessary. Although the software tool is currently developed explicitly for the design of a containership, it is not very cumbersome to graduate to the design of any other ship. The function of the optimization component is to return the minimum value of the objective function given the constraints and design variables. Two popular classes of algorithms for the optimization process are gradient-based and genetic algorithms. In genetic algorithms, an initial set of design variables is ordered by their objective function values. A succeeding generation is obtained by random variations of the better designs of the previous generation. This process is continued until not much change is noticed in the objective function between succeeding generations. In gradient-based algorithms, changes in the objective function are evaluated with respect to small changes in the design variables and the new vector of design variables is obtained in the direction of lower objective value. Genetic algorithms are practically restricted to parametric analysis

methods by the computational time required. In the current formulation, first principle analysis codes are to be employed and hence gradient-based algorithms are used. Gradient-based algorithms cannot deal with discrete variables and discontinuous design space. This component is provided with the values of the design variables and the values of the constraint functions. The respective search directions of the design variables and the constraint functions are then calculated to give the new design variables.

$$\mathbf{X}^{i+1} = \mathbf{X}^i + \lambda \mathbf{S}^i \quad (2.1)$$

Where i is the iteration number, \mathbf{X} represents the vector of design variables, λ is the step length and \mathbf{S} is the new search direction. This process starts from \mathbf{X}^0 , the initial set of design variables as determined by the user. The search direction determines the direction of the next design point and the step length is a measure of the distance to be moved in that direction. The search direction \mathbf{S} should be both usable and feasible. Usability implies that the direction should be such that it reduces the objective function and feasibility means that no constraints are violated in the search direction.

The geometric modules generate a smooth hull form based on a NURBS (Non-Uniform Rational B-Spline) surface expression. Then, a mesh of offsets is calculated based on the NURBS net. The hydrostatic particulars of the ship are also calculated. The performance evaluation components calculate the technical and economical performance of the containership under question. The technical aspects of these components include resistance, propulsive efficiency, powering requirements, stability, container capacity and ship weight. The economic aspects include the building cost, operating cost and the required freight rate of the ship. All these evaluations require parameters that can be input by the user at the start of each iteration.

Figure (2.1) gives a flowchart of the MDO project. This flowchart indicates how the problem is organized. The user is required to input the initial design variables. Then the geometry module develops the hull shape based on the values of design variables followed by the calculation of the hydrostatics. The performance of the ship generated is then evaluated. Once the ship design is available, the design is checked against constraints and it is determined if the design is within the constraint boundaries. If the constraints are satisfied, it is checked to see if the minimum possible objective function is

reached. This is checked by comparing the objective function of two successive iterations. If the objective function values are similar, then it is assumed that the minimum value is reached. The optimizer also checks the gradients of the constraint functions and terminates the iterative process if the gradient values are too small (degree of small is user defined). This process is repeated until a minimum objective value is achieved or a maximum number of iterations (specified by the user) have been reached. The results of the optimization process are then output for further analyses.

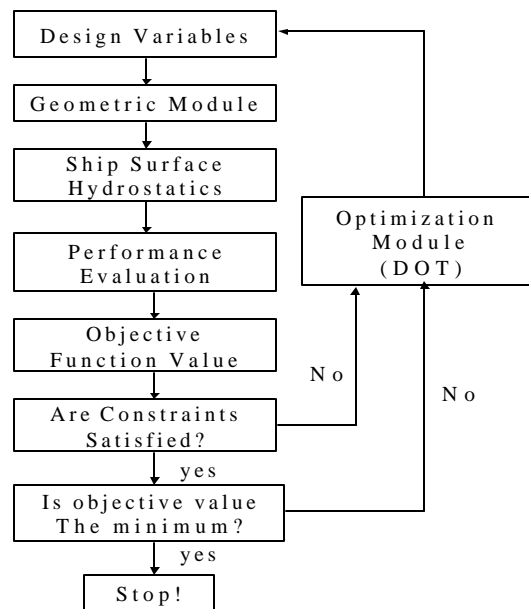


Figure 2.1 Flow Chart of the MDO Project.

2.3 Software Overview

The computational code consists of a number of analysis modules, an optimizer and a user interface. The code is written in a Microsoft Windows NT environment and Microsoft's Component Object Model (COM) framework for the analysis modules. This was done to make the MDO tool compatible with a larger FIRST software code, to which it will be integrated. FIRST stands for - First Principles Approach for Shipbuilding Integrated Process and Product Development. The goal of the FIRST project was to develop an integrated computational design environment that would enable the ship designer to reduce the time needed to produce and analyze alternate ship designs at a high level of detail. Active Template Libraries (ATL) have been used to generate the COM

modules. Each of these modules acts as a server independently and exposes functionalities to clients through interfaces. COM concepts insist that interfaces should be immutable. Hence, once the interfaces are fixed, the implementation can change in one module without really affecting code in the rest of the modules. This kind of framework is important for both MDO and FIRST as they were being developed simultaneously. Hence, the tools that were to be developed by the FIRST team and to be used in MDO were not yet available. So, surrogate modules were written with the same interfaces as the high fidelity ones. Simple, yet proven and published parameterizations were used for the surrogate modules used in MDO. However, many of the parameters and coefficients used in MDO had to be updated to reflect current trends in costs and construction practices.

ATL COM modules have been written in C++. A main program called the optimization manager manages these analysis modules. The optimization manager receives information from a user interface and directs them to appropriate modules in the right order. Both the optimization manager and the user interface have been written in Visual Basic. The user interface presents a standard set of MS Windows menus. The user can browse through the set of available input files and select one for analysis and can also save an input file. The text input in the input file is presented in various forms for easy readability. The user can alter the initial design variables or can reset the bounds. The user can toggle the constraints and set their minimum or maximum values. The user is also given the option of changing the parameters used in the analysis or adjusting the parameters controlling the operation of the optimizer. Once the process of optimization starts, the user can graphically see the change of the objective function through the iterations. The form containing the design variables also houses the current vector of design variables and the current value of the objective function as the optimization process proceeds. This tool gives an idea of how the optimization is proceeding. The user is also provided with the option of stopping the optimization, changing the design vector and starting from that position. Once the process of optimization stops, the user can verify the design by restarting from the current set of design variables. The current vector of design variables then becomes the initial values for the new iterations. After the user is satisfied with the design, a menu option allows him/her to see the final result in text format or view the iteration history of some of the important parameters in an Excel

sheet. Figure (2.2) illustrates the data flow in the analysis modules, which are COM objects. The arrow indicates the direction of data flow between each module.

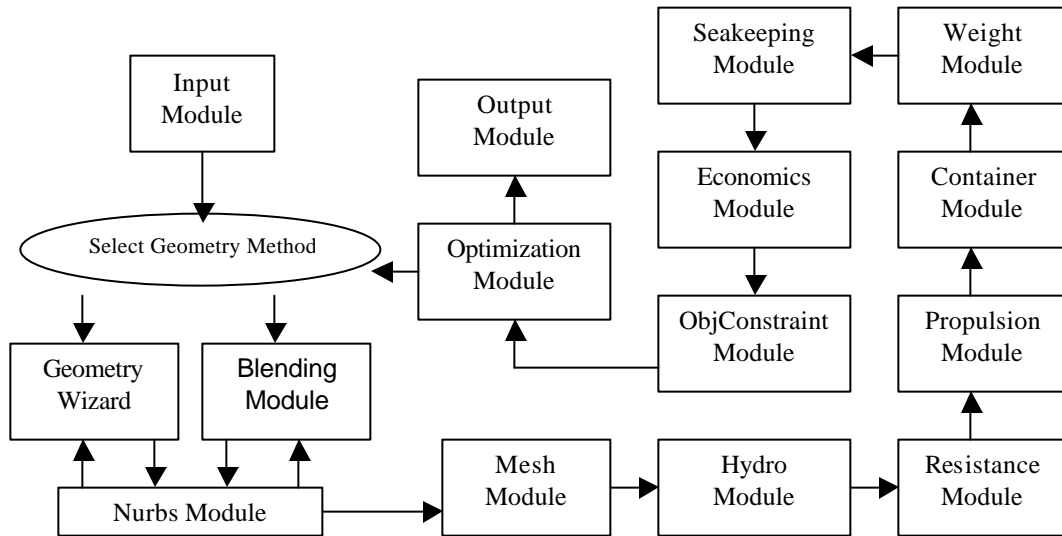


Figure 2.2: Organization of the Analysis modules.

The optimizer was chosen to be the Design Optimization Tools (DOT) package from Vanderplaats Research and Development. Dot gives the choice of three gradient-based optimization algorithms: Modified Method of Feasible Directions (MMFD), Sequential Linear Programming and Sequential Quadratic Programming. It is ascertained that for this application, the fastest algorithm, which also gives the most reasonable results, is the Sequential Linear Programming. The DOT Fortran subroutines have been made into a COM object by writing a C++ wrapper code around the DOT Fortran source code.

Figure (2.3) is a wiring diagram, which gives an overview of the entire system. This diagram illustrates the position of parameters in the whole picture. The location of constraints and the flow of data are also visible. But, only a few key parameters used are shown. Here, it can be seen that most of the modules are general in nature and only the weight and the container modules are specific to a particular ship. In Figure (2.3) constraints are in heavy boxes while the parameters are in dashed boxes.

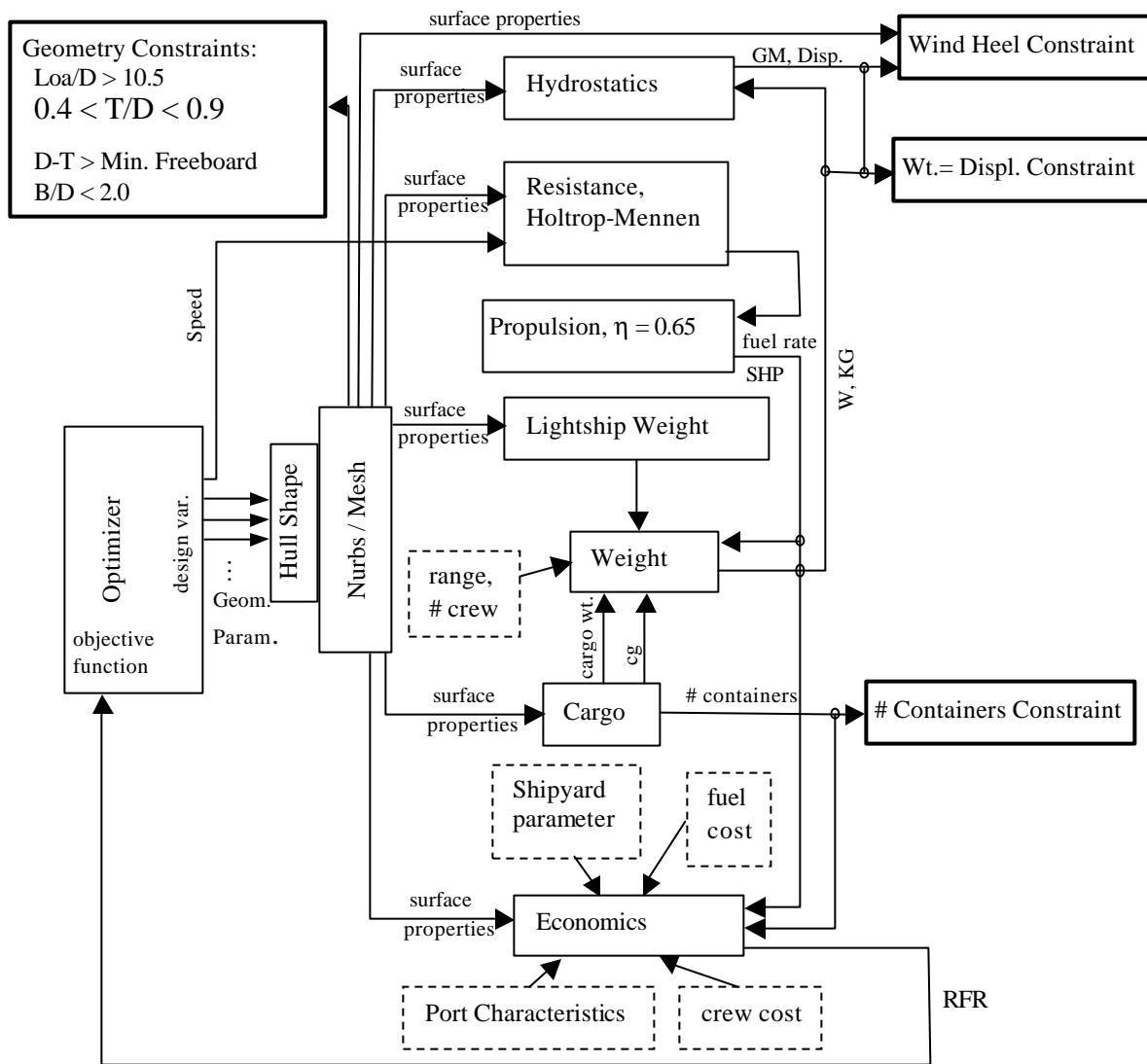


Figure 2.3: General Construction of the Software system.

Chapter 3: Problem Formulation

3.1 Objective Function:

The objective of the optimization process is to design a container ship. The main emphasis of the problem is to design a ship with maximum economic returns keeping in mind the stability and structural requirements of the ship. Bearing this in mind, the required freight rate was decided upon as the objective function. The freight rate is the money charged by the shipping company per unit weight of cargo carried over a unit distance. In this context it is useful to define a required freight rate that is the minimum freight rate that must be charged to break even over the life of the ship. Optimization is used to minimize the required freight rate (RFR) that is calculated as dollars/metric ton/nautical mile. Here the metric ton is a unit of cargo and the nautical mile is the unit of distance traveled by the ship. By intuition, one tends to maximize the freight rate. Practically, any freight rate above the required freight rate would lead to profits.

Before the start of the optimization process, the optimizer calculates the initial value of the objective function. This initial value of the RFR is calculated from the initial values of the design variables that are entered by the user. The objective function is normalized by the initial value of RFR. This is done because the magnitudes of an unscaled RFR can vary a lot. This results from the variation in magnitude of the design variables. The optimizer calculates a search direction that is the direction in which the optimization should proceed in the next iteration. Usually, this direction is a direction of minimizing the objective function subject to the constraints gradient. If the constraint gradients vary a lot in magnitude, then the optimizer will not be able to judge the direction. The constraint with the largest magnitude will define the search direction. The optimization process stops when the largest component of the search vector is small in magnitude. Here the magnitude of the objective function has to be similar to the magnitudes of the constraints and the design variables. If it is not normalized, then the optimizer may give different solutions for different initial values of the required freight rate. This is because the search direction might be different for various scales of the initial values. The results are then inconsistent. Among the many ways in which the RFR

This list of variables forms a subset of the total list of variables needed to form a ship. This list consists of the main geometric parameters and will be present when designing any ship. Using these variables and some other parameters, the initial design of any ship can be completed. Typically, the hull, light ship weight and stability characteristics can be determined by formulas based on experiments and actual ships. But this list lacks the variables that are necessary for the design of container ships in particular. To address these parameters, new design variables have to be incorporated in the analysis to determine the specifications of a container ship. Once a ship's purpose is defined, the other specifications like weight and economics can be determined.

3.3 Constraints

(a) Equality

$$\text{Weight} = \text{Displacement} \quad (3.2)$$

This is the basic constraint in every ship. In the current formulation, this equality constraint is enforced as two inequality constraints expressed below.

$$\text{i) } \frac{\text{Weight}}{\text{Displacement}} - 1.0 < 0.0 \quad (3.3)$$

$$\text{ii) } 1.0 - \frac{\text{Weight}}{\text{Displacement}} < 0.0 \quad (3.4)$$

Due to the above formulation, there is a disparity between the values of weight and displacement. But this value is negligible compared to the actual values. The error percentage is less than 1%. The tolerance for this constraint can be reduced even further by changing the tolerance factor if need be.

(b) Stability

$$1. \quad \text{GM} \geq \text{GM}_{\min} \quad [14] \quad (3.5)$$

GM_{\min} is calculated by the wind heel criterion set by the US Coast Guard. This stability constraint is formulated as the following:

$$g = 1.0 - \frac{\text{GM}}{\text{GM}_{\min}} . \text{ When } \text{GM} \ll 0, \text{ the value of } \partial g / \partial B, \text{ the derivative of } g \text{ with respect}$$

to beam can become positive due to the dependence of GM_{\min} on the beam. This implies

that to achieve stability, the beam has to be decreased. But to have better transverse stability, the beam has to increase. So, this is not the way the optimization has to proceed. To overcome this situation, GM_{\min} is held at its initial value as long as $GM < 0$. Thus the dependence of GM_{\min} on the beam is overcome until $GM \geq 0$. Once this condition is satisfied ($GM > 0$), GM_{\min} is calculated with respect to the current design variables.

$$2. \quad \text{Freeboard} \geq \text{Freeboard}_{\min} \quad [14] \quad (3.6)$$

Freeboard_{\min} is also calculated by standards set by the US Coast Guard. Freeboard is the difference in depth and the draft. A minimum freeboard is needed to prevent flooding of the deck in most cases.

(c) Geometric Constraints

$$1. \quad \text{Draft/Depth} > (\text{Draft/Depth})_{\min} \quad (3.7)$$

This constraint ensures that the hull maintains some minimum reasonable immersion. This minimum draft is required for the proper propeller immersion. The minimum value of the ratio can be changed by the user and this constraint can be toggled in the interface.

$$2. \quad \text{Draft/Depth} < (\text{Draft/Depth})_{\max} \quad (3.8)$$

This makes sure that draft is less than the depth. Even though there is a minimum freeboard constraint in use, this constraint ensures that draft is never more than a certain percentage of the depth. Different ships might have various draft requirements based on the use and on the waters through which it traverses. $(\text{Draft/Depth})_{\max}$ has maximum value of one to ensure that draft never goes above the depth.

$$3. \quad \text{Loa/Depth} > (\text{Loa/Depth})_{\min} \quad (3.9)$$

This condition is required by the formula to calculate the hull steel weight [11]. Having this constraint ensures that the hull steel weight is calculated correctly.

(d) Transverse Rolling Period (TRP)

1. $\text{Transverse Rolling Period} > (\text{TRP})_{\min}$ **(3.10)**

The transverse rolling period can also be used as a constraint to keep GM from getting too big. The TRP is also directly proportional to the GM of the ship and is hence undefined for $\text{GM} < 0$. So, when $\text{GM} < 0$, the TRP is calculated as an even function of GM. The minimum value of TRP can be input by the user.

Table 3.2 gives a further list of optional container ship constraints.

Table 3.2: Container Ship Constraints

Parameter	Default Value
Maximum Displacement (ton)	70000
Maximum Shaft Horsepower (HP)	60000
Minimum Container Number (TEU)	2000
Maximum Container Number (TEU)	20000

3.4 Constraint Tolerance

The constraint tolerances have to be defined because all calculations are performed numerically and not analytically. Hence, the optimizer has to be given a corridor; if the constraint values lie within that corridor, it can be decided whether the constraint is active or violated. A constraint is defined to be violated if the constraint inequality is not satisfied i.e., if the value of the constraint function is greater than the value of some tolerance. The constraint is said to be active if the absolute value of the constraint is close to zero and has a decisive effect on the design of the system. DOT defines two parameters to define constraint tolerances – CT and CTMIN with default values of -0.03 and 0.003 . These values can be changed by the user as required for the analysis. Since DOT standards for normalization are also wired into the optimization manager for design variables, objective function and constraint functions; using the default values for constraint tolerances gives acceptable results of optimization.

Constraint is active if $\text{CT} < g \leq \text{CTMIN}$

Constraint is violated if $g > \text{CTMIN}$.

By default, the optimization proceeds until all the constraints are within their tolerances and the objective value is within tolerance limits for two successive iterations. This termination feature can also be manipulated by the user at the start of the optimization process in the user interface. But, typically convergence for two successive iterations is both necessary and is for most of the analysis a sufficient condition for the completion of the optimization process.

3.5 Scaling

The design variables have been scaled in the formulation to facilitate the compatibility of all the parameters involved. The magnitudes of the design variables may differ a lot. As an example, length may vary in the order of 10^2 while the order of draft might be 10 and that of blending coefficients are 1.0. The scaling factor used is the one defined by DOT. The scaling factor is applied to the design variables before they are supplied to the optimizer. Table (3.3) gives the scaling factor and value after scaling for the design variables. As indicated, there is no definition for scaling factor for design variables that are very small in value. In the current list of design variables, only the blending coefficient can take realistic values that are small. Hence there is no scaling that is applied to the blending coefficient.

Table 3.3: Scaling Factor for Design Variables used in DOT

Original Value	Scaling Factor	Value after scaling
Largest (in order of 10^n)	10^{-n}	1.0 ~ 10.0
Larger than 1.0	Reciprocal of the original value	1.0
Less than 1.0	1.0	Same as the original value

3.6 Normalization of Constraint functions

As a general rule of any optimization process, the constraint functions have to be formulated in such a way that the value of the functions does not vary a lot from a specified constant, typically zero. As defined before, DOT has to determine whether a constraint is active or violated. The optimizer cannot estimate this if the values of the functions vary in magnitude. They are also normalized so that tolerance bounds need not

be defined for each constraint. DOT has a single set of tolerance bounds for all the constraint functions put together. Once the functions are standardized, they can be compared to one another. This is essential in determining a stable search direction. This is achieved by pulling active and violated constraints from the lot of functions and developing a direction to appropriately satisfy the violated constraints first and then addressing the active constraints. Care is taken to see that active constraints are not violated. A watch is also kept on the non-active constraints during optimization. If the constraints are not normalized, the constraint with the largest absolute value might determine the direction of the new iteration. However, this might not be right because a constraint value might be small in absolute terms but it might still be active or violated on a relative scale to itself i.e., for example; if the value of a constraint in the current iteration is 5 and in the succeeding iteration, it is 10, then the value is not high but it is a 100% increase compared to the value in the previous iteration. Hence, the constraints have to be normalized to ensure a constant scale of comparison and evaluation for active or violated behavior. Table (3.4) gives the formulation of constraints after normalization for DOT. Here, it is assumed that $A > 0$.

Table 3.4: Normalization of constraints for DOT

Constraint	Formulation for optimization	Normalization for DOT
$A \leq B$	$A - B \leq 0$	$1 - \frac{B}{A} \leq 0$

3.7 Hull Geometry Manipulation

Blending is a process by which blending coefficients (weighting factors) are applied to basis hulls to generate the hull for the current problem. The hull is represented as a barycentric blend of basis hulls. This is done to control the hull shape that is being generated. The blending is applied to the NURBS net points on the basis hulls. These points are then used to generate the mesh for the entire hull. So, the process of blending produces a hull that is a mixture of hulls that are already defined. But this blended hull has to be scaled to conform to the current set of design variables. This is done before the net points are processed for the mesh points. Scaling is achieved by dividing the

respective new points from the new hull with the maximum value of the net-coordinates (which gives the value of length/beam/depth). These values are then multiplied by the respective values of the design variables from the current design. For example, to calculate the scaled x-net point:

$$x - \text{net point} = \frac{x - \text{net point (from blending)} * \text{Length (Basis Ship)}}{\text{Length (Current Design Variable)}}$$

The Length of the basis ship is found by parsing through the x-net points of the basis ship and finding the maximum value. To scale the beam, this process is applied to y-net point and to scale the depth it is applied to the z-net point.

During the process of blending, it is ensured that the sum of the weighting factors is equal to one. The process of blending can be represented in the following manner:

$$\text{Resultant ship hull} = \sum C_n \text{Basis Hull}_n$$

Where,

$$\sum C_n = 1.0$$

and,

$$0 \leq C_n \leq 1, \quad n = 1, 2, \dots, N.$$

where, N represents the number of basis hulls being used to generate the new hull. This kind of formulation ensures that the hull resulting from the blending process is restricted to the hull shape space that is bounded by the set of basis hulls. This means that the hull form generated after blending is not an extrapolated form of the existing hulls.

The Blending coefficients C_n are the design variables that manipulate the hull shape. In the current formulation, the hull shape is generated by two basis hulls. This would indicate that a minimum of two blending coefficients is required. Due to the constraint $\sum C_n = 1.0$, the number of design variables needed is one.

Chapter 4: A New Constraint

4.1 Background

The more important geometric ratios that define a ship's particulars are the ones that involve various combinations of the principal dimensions. Experience shows that successful ships have certain bounds on these geometric ratios that make it structurally sound [6]. In the previous formulation, there is no constraint that addresses the structural aspect of the design. The important dimension ratios are length/beam, draft/depth, length/depth and beam/depth. The optimization results always tends towards the longest and the broadest ship. That is so because a longer and broader ship would mean a greater cargo carrying capacity of the ship and will be stable. But the same does not apply to the depth of the ship. The optimizer prefers to carry cargo on the deck rather than below the deck because of added steel weight as a result of increase of depth. Figures 4.1-4.4 show results obtained from the optimizer and ratios from actual ships. The optimized result shows a large variation from the actual ships with respect to the beam/depth and length/depth ratio. All the other ratios are comparable to the ratios calculated from actual container ship data. This is because the depths of ships obtained from the results of optimization are less than actual ships with comparable lengths and beams. The beam/depth ratio for good structural integrity lies between 1.5 and 2.0. Whereas, the ratios obtained from the MDO formulation discussed above was in excess of 3.0. Similarly, from Figure 4.2, it is evident that for better structural strength, (L/D) ratio should lie between 10 and 15 whereas, from the current MDO problem formulation, this ratio is in excess of 30. As the depth of the ship is increased, the length/depth ratio would also improve. Hence there was a need to add a new structural constraint.

4.2 Formulation of a new constraint

A new structural constraint on the geometry was formulated in the following way:

$$\frac{\text{Beam}}{\text{Depth}} < \left(\frac{\text{Beam}}{\text{Depth}}\right)_{\max} \quad (4.1)$$

Or

$$1 - \left(\frac{B}{D}\right) / \left(\frac{B}{D}\right)_{\max} < 0.0 \quad (4.2)$$

It was not necessary to define a lower bound for this particular ratio because of the nature in which the optimization proceeds. The optimizer tends to increase the value of the beam and decrease the depth. Hence the upper bound of this ratio has to be constrained. The user can interactively input the value of $(B/D)_{\max}$ at the start of the optimization process. This gives him the necessary freedom to control the extent to which the depth might increase. With the enforcement of the B/D constraint, the depth would significantly increase. This would mean that the L/D ratio would also diminish and assume acceptable proportions. Examples with the L/D constraint are dealt with in a later chapter.

4.3 Evaluation of Constraint

On the addition of this constraint, the results generally obtained followed a definite pattern. They were usually “short ships”. Here the term “short” refers to the length between perpendiculars of the ship and is relative to the results obtained earlier using MDO. The length of the optimized ship was reduced by almost 30-40 %. Due to this vast reduction in length, the cargo carrying capacity was reduced and the displacement of the ship reduced. This would mean an increase in KM (distance between keel and metacenter) and hence an increase in GM (distance between center of gravity and metacenter). The optimizer chose to reduce the length of the ship to make it more stable. The addition of the B/D constraint prompted an increase in the depth of the ship. Hence the containers carried below deck have also increased. The number of tiers below deck is formulated as a monotonically increasing variable with respect to the depth. Similarly, the number of tiers above deck increases with respect to the beam of the ship (Equation 4.1). As the steel and cargo weight increased below deck, there was a step up of the center of gravity of the ship. To compensate for this increase in CG, the optimizer had to decrease the cargo being carried on deck. Reducing the number of cargo carrying tiers on deck can do this. At this point, the optimizer has no means to decrease the number of tiers on deck. A decrease in the length of the ship will prompt a decrease in the overall cargo being carried, as the beam does not change appreciably. As the other economic parameters were not changed, the required freight rates obtained were much larger than previous results (Table 4.1). This was because of added steel weight and lesser cargo being carried. The lesser cargo is due to the decrease in length of the ship.

The stability constraints were still active but now the optimized ships were more structurally sound but not necessarily economically viable. As the design is for a container ship, the economic viability is an important criterion in determining the various design parameters.

On observation of the results obtained (Table 4.1), it was found that the length of the ship reduced and there was an increase in the velocity. There are two ways in which the optimizer can ensure the stability of the ship. First, it can increase the KM (I_t/∇). Second, it can reduce the CG of the ship. The optimizer attempts to do both. As discussed earlier, decreasing the length of the ship decreases displacement and hence increases KM. The other option is to increase the speed of the ship. The CG of the engine plant is lower than the CG of the ship. So, there is also an increase in the engine plant mass that is brought about by increasing the capacity of the engine. The increase in speed prompts an increase in the engine plant mass, which is good for the stability of the ship but an increase in speed increases the required SHP. This increases both the machinery building costs and the annual operating costs (increasing fuel costs). The final design achieved by this design though feasible is not practical. Hence, it was decided to make the load carrying capacity of the ship a variable.

Table 4.1: Evaluation of B/D Constraint

Design Variables	Lower Bound	Upper Bound	Initial Value	No B/D Constraint	B/D < 2.0	B/D < 2.5
Loa (m)	130	300	200	299.979	172.474	300.0
Beam (m)	18	40	30	39.992	40.0	40.0
Depth (m)	15	35	20	15.002	20.0	16.0
Draft (m)	4	20	15	10.327	13.238	10.634
Speed (knots)	4	25	14	17.457	22.985	17.835
C1	0	1	0.25	0.705	0.0	0.638
Objective Value			0.00159	0.001116	0.002896	0.00113

The number of tiers on deck is a function of beam

$$\begin{aligned}
 NT_d &= 4.0 && \text{for } B \leq 32.2 \text{ m} && \text{(equation 4.1)} \\
 &= 4.0 + \frac{B - 32.2}{43.0 - 40.0} && \text{for } 32.2 < B \leq 40.0 \\
 &= 5.0 + \frac{B - 40.0}{43.0 - 40.0} && \text{for } 40.0 < B \leq 43.0 \\
 &= 6.0 && \text{for } B > 43.0
 \end{aligned}$$

This relation was developed by Ying Chen [5].

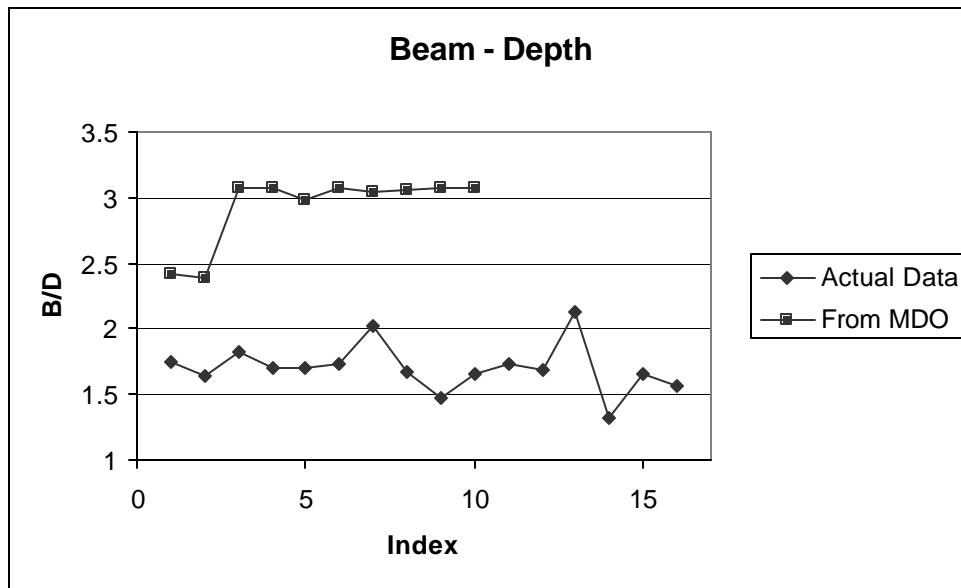


Figure 4.1: B/D ratio from different ships and from results of MDO.

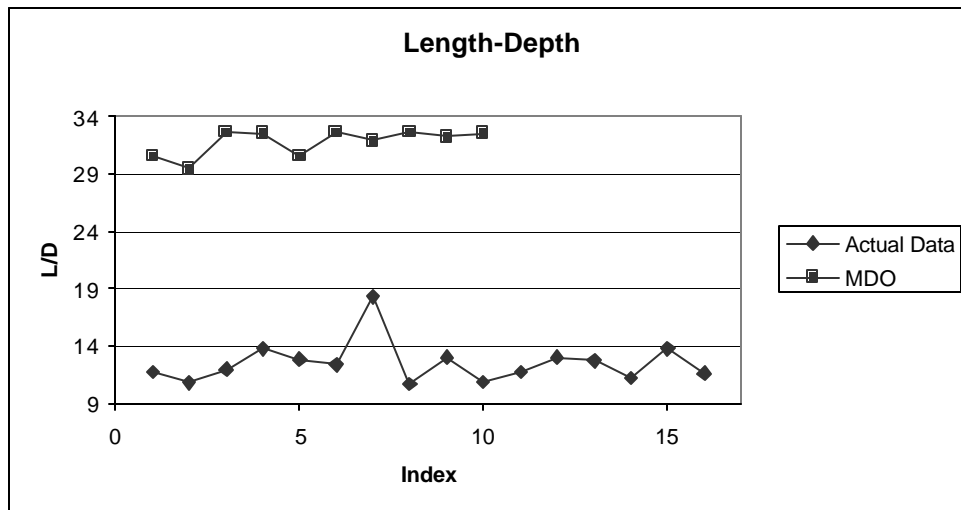


Figure 4.2: L/D ratio from different ships and from results of MDO.

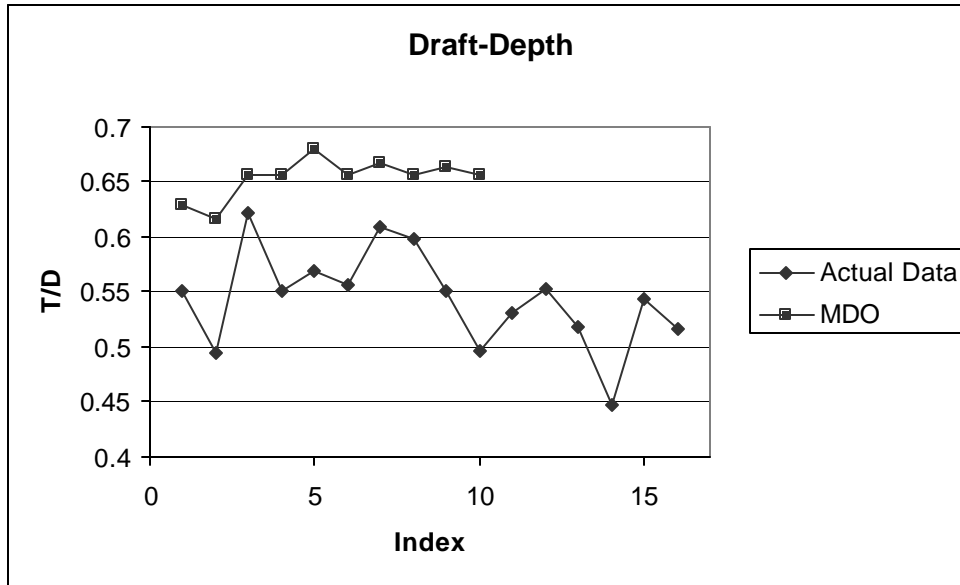


Figure 4.3: T/D ratio from different ships and from results of MDO.

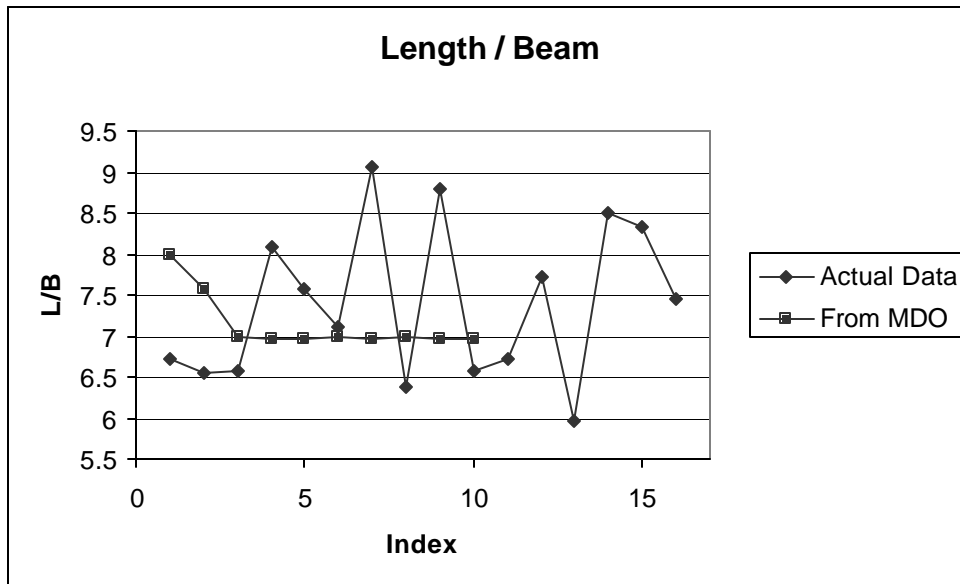


Figure 4.4: L/B ratio from different ships and from results of MDO.

Chapter 5: Number of Tiers on Deck

5.1 Number of Containers

In a container ship, the containers are stowed both above and below the deck. The number of containers being stowed below the deck is dependent on the geometric shape (as represented by C_b) and the principle dimensions of the ship (length, beam and depth). The measure of containers above the deck is also dependent on the number of tiers (height to which the containers are stacked) above the deck. The number of tiers below deck is largely dependent on the depth of the ship. The expression for the total number of containers is:

$$TEU_t = TEU_b + TEU_d \quad (5.1)$$

Where,

$$TEU_b = S_b * L_b * B_b * D_b$$

And

$$TEU_d = S_d * L_d * B_d * NT_d$$

In the above formulation, TEU is a representation of a container and stands for Twenty-foot Equivalent Unit. TEU_t is the total number of containers. The indices b and d represent parameters above and below the deck. TEU_b and TEU_d are the number of containers above and below deck respectively. S_b and S_d are the stowage factors for the containers. The stowage factor below the deck depends on the shape of the hull, which is given by C_b . Above the deck; the stowage factor is assumed to be a constant and is arrived at after referring to existing ships. L_b , B_b and D_b are the number of containers below deck in the length, beam and depth directions respectively. Similarly, L_d and B_d are the container numbers in the length and beam directions above the deck. NT_d is the number of tiers above deck. It is important to note that L_b , B_b , D_b , L_d and B_d are integer values. This is because a container cannot be stored in fractions. The number of tiers on deck, however, can take floating values because not all stacks of containers need to be of the same height. It is the average height of the stacks on deck. The number of tiers below deck depends on the depth of the ship.

Regarding the integral effect of the above variables, the container numbers have been formulated in the following manner:

$$L_b = L_d = \text{int}\left(\frac{L_{\text{con}}}{L_0}\right) \quad (5.2)$$

$$B_b = \text{int}\left(\frac{B - 2 * WB}{B_0}\right) \quad (5.3)$$

$$D_b = \text{int}\left(\frac{D - DBH}{D_0}\right) \quad (5.4)$$

$$B_d = \text{int}\left(\frac{B}{B_0}\right) \quad (5.5)$$

Where L_{con} is the effective length of the ship for carrying containers. L_{con} is assumed as a fraction of the overall length of the ship, L_{oa} . WB is the breadth of the wing tank; D the depth of the ship and DBH is the height of the double bottom. L_0 , B_0 and D_0 are the length, breadth and depth of the container respectively. Please refer to Equation (4.1) for the approximation on the number of tiers on deck, NT_d .

The formulation for NT_d is not appropriate for ships with small beam values. In the initial formulation of the design, the user can specify the NT_d through the user interface before the start of the optimization process. This was done to give the user the option of not using the Equation (4.1) to find the NT_d based on the beam. After optimization (discussed later in this section) it was noticed that the resulting ship design would tend towards the upper bound of the beam. This indicates that though the formulation is not appropriate for ships with smaller beams, if the user chooses to limit the beam of the ship, he can specify the NT_d too to suit the purpose of the design.

The number of containers must be integer values. Due to this, the number of containers being carried in the ship looks like a discontinuous series of steps as the dimensions of the ship vary. This has a big influence on the process of optimization. As the number of containers ceases to be a continuous variable due to its step nature, the

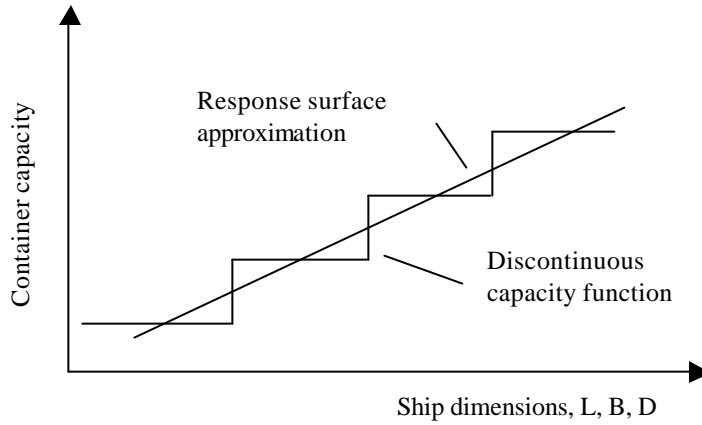


Figure 5.1: Illustration of container capacity function and linear response surface

objective and constraint functions cease to be continuous. The optimizer is unable to calculate the gradients of the objective and the constraint functions effectively. The objective function becomes a saw-toothed function with local minima in small intervals of the design space. To curb this problem, it was necessary to make the number of containers a continuous variable. To do this, a least square fitting method was used to pass a smooth function for the discontinuous function. The values of WB and DBH were assumed to a constant at 1.83 meters. The function that was derived was:

$$TEU_{b_float} = S_b * (0.0196 * Loa * B * D - 148.6129) \quad (5.6)$$

$$TEU_{d_float} = 0.050117 * Loa * B * NT_d - 82.6702 \quad (5.7)$$

Where, Loa is the overall length of the ship. Both the floating-point values and the integer values of the number of containers are presented in the output of the optimization process. The difference between the floating-point and the integer numbers of containers is usually less than one percent. Figure (5.1) illustrates the effect of the discontinuous function and the continuous response surface in two dimensions

5.2 Effect of B/D Constraint

The B/D constraint was introduced to give structural stability to the ship. This constraint ensured that the value of the depth was in a right proportion to the beam of the ship. Compared to the previous analysis, the addition of this constraint resulted in an

increase in the depth of the ship. The value of the beam still tended towards the upper bound as input by the user. All bounds remaining constant, an increase in the depth meant an increased number of tiers below deck. The number of tiers above the deck was similar to the previous results and this value depends on the beam of the ship (refer Equation (4.1)). Due to the increase in the number of tiers below deck, the center of gravity of the ship went up. This meant that the metacentric height (GM) of the ship came down affecting the transverse stability of the ship. The only recourse with the optimizer was to reduce the dimensions of the ship accordingly. With the reduction in dimensions of the ship, the displacement of the ship decreased resulting in an increase in BM, which is the height of metacenter above the base. This was done to improve the GM of the ship.

The cargo being carried by the ship is directly proportional to the number of tiers on the ship, both above and below the deck. Since, the number of tiers below the deck depends on the depth of the ship, the optimizer can vary the depth to vary the number of tiers. The optimizer would try to reduce the depth to decrease steel weight and also to reduce resistance, which comes about by reducing draft. Draft is varied to ensure that weight equals displacement of the ship. Sometimes, as the depth and draft are related by a constraint, a decrease in depth will also prompt a decrease in the draft of the ship. But the addition of the B/D constraint ensures that the depth does not fall too much. Since the depth cannot be reduced beyond what the constraint permits, it is advisable to reduce the number of tiers above the deck. This not only reduces cargo weight but also reduces it at a higher center of gravity thus lowering the overall center of gravity. This improves the transverse stability of the ship by increasing the GM (primarily because G goes down). The number of tiers above deck is formulated to depend on beam. The optimizer tends to maximize the beam for greater cargo capacity and stability. Due to decrease in the number of containers, the objective value decreases. Please refer to the Table (5.1) for results with varying NT_d . As mentioned, the beam tends towards its upper bound of 43 m. This implies that NT_d takes a value of 5.5 (refer Equation 4.1). But the user is given an option of changing the NT_d in the user interface. This value remains constant during the entire course of the analysis. The design was evaluated for different values of NT_d and the results are shown in Table (5.1).

Table 5.1: Design parameters for various NT_d

NT_d	4	4.5	5	5.25	5.5
Loa(m)	300	299.8	300	229.3	227.3
Beam(m)	43	42.4	42.7	43	43
Depth(m)	21.6	21.2	21.3	21.5	21.5
Draft(m)	10.6	10.9	11.3	11.6	12.4
Speed(kts)	18.6	18.4	18.7	20.7	25.6
Displacement (mt)	94280	95360	100140	79830	85420
#TEU's	5189	5363	5723	4494	4596
Shaft Horsepower	28292	26724	28855	40984	115868
GM (m)	2.03	1.52	0.82	0.25	0.25
RFR (\$/mt/nmi)	0.0013	0.00124	0.00118	0.00134	0.00173

Figure (5.2) illustrates the effect of increasing the number of tiers on deck on the primary design parameters. All the values that have been shown in the Figure (5.2) have been normalized to fit in the same graph. They have been fit into the same graph so that they can be compared to one another with the change in the number of tiers on deck. In all of the above examples, the B/D constraint is active. This means that the active constraint has forced the depth to go above the value that the depth would take if it were unconstrained and unbounded. As the value of NT_d increases, the GM constraint is not satisfied anymore. The optimizer resorts to decreasing the length of the ship to satisfy this constraint because this would mean an overall reduction in the displacement. The optimizer cannot decrease beam because reducing beam also reduces the GM of the ship. The beam as expected has tended towards the upper bound of 43 meters. The depth, due to the B/D constraint (B/D_{max} is set at 2.0) is approximately half of the beam. Hence this constraint is active in all the examples. The speed is higher when the NT_d is increased. This is because as the GM decreases with increasing NT_d , the optimizer has to resort to ways to decrease the CG of the ship. The optimizer increases the speed of the ship. This results in an increase of the fuel mass and the engine plant mass. As the engine room and the fuel tanks are located in a lower position, the CG of the overall ship comes down. This would imply an increase in the GM of the ship, which would improve the stability of the ship. Thus there is an increase in speed as well as an increase in the shaft horsepower.

The shaft horsepower is directly proportional to the speed of the ship. The displacement is approximately indicative of the cargo being carried by the ship. Displacement is equal to weight and as the cargo increases, the weight of the ship also increases. The increase in displacement is indicated by the increase in the draft of the ship. From the examples, it can be verified that the increase in cargo mass above the decks reduces the GM. In fact, the GM values of 0.25m and 0.24m make the GM constraint active. An increase in cargo mass above the deck would increase the overall center of gravity reducing the GM. The required freight rate is high for lower lengths but also becomes lower for increasing NT_d at similar lengths ($NT_d = 4.0; 4.5; 5.0$). Hence, a greater length and higher NT_d indicates a better RFR. Therefore, it would be advisable to try to increase both the principal dimensions of the ship (length and beam) and also increase the NT_d . Hence, there are conflicting results to increasing NT_d . One is that the RFR will decrease; and the other is that an increase in NT_d affects stability resulting in decreasing the length and increasing the speed. An increase in speed results in the increase of the shaft horsepower. A decrease in length increases RFR. Hence, in effect, the optimizer finds an optimum point where a further decrease in length does not compensate for the increase in NT_d . This also increases the annual operating costs of the ship. Therefore, it can be useful to vary NT_d as a design parameter during the process of optimization rather than maintaining it at a constant value during the process.

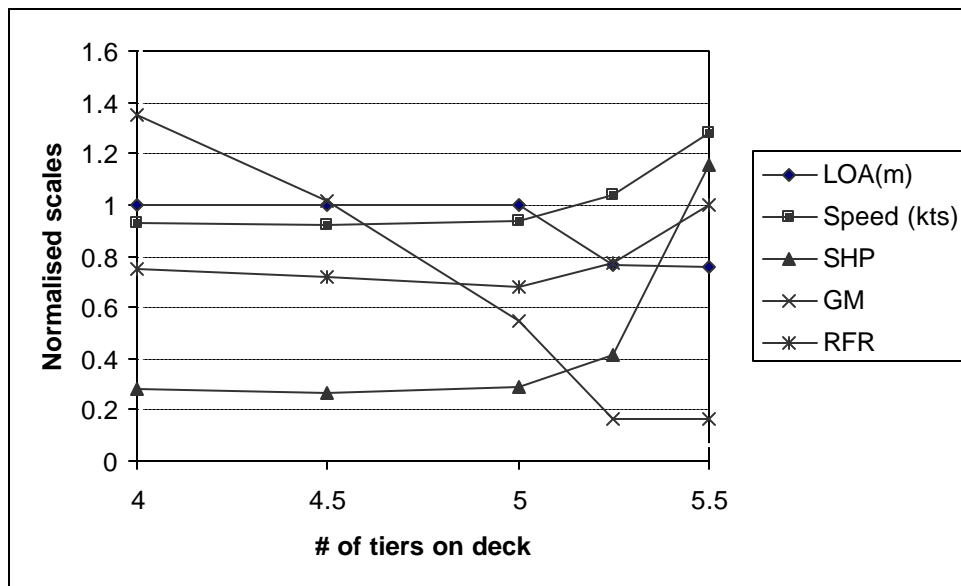


Figure 5.2: Variation of Normalized parameters with NT_d

5.3 NT_d as a Design Variable

The number of tiers on deck is an important factor to determine the cargo being carried by the ship. A greater cargo indicates higher revenue. Of course, the greater the cargo, the greater the displacement. A higher displacement indicates higher draft and hence higher resistance. But the revenue increase due to addition of cargo is more than the loss due to greater SHP. Hence the optimizer prefers to increase cargo. The compromise can be in the form of structural and transverse stability. The introduction of the B/D constraint ensures a certain degree of structural stability. The GM constraint is necessary for transverse stability. So, now the onus is on the optimizer to maximize the cargo and maintain the stability. There are two ways in which the cargo can be increased. One is to increase the cargo carried below deck and the other is through the cargo being carried above the deck. The number of tiers below the deck is not an independent variable and has to depend on the depth of the ship. The optimizer prefers not to increase the depth because of added steel mass of the hull. The number of tiers on deck on the other hand is independent of depth. Equation (4.1) was an empirical formula where NT_d varies with beam, which was developed to give the optimizer an idea of the number of tiers on deck. The user was also given an alternative of setting the NT_d before the start of the optimization. But the value of NT_d was constant during the process of optimization. Hence, it was decided to make the NT_d a design variable that could be varied by the optimizer at run time.

The number of tiers on deck is added to the MDO user interface as a design variable. All the other ways in which NT_d was calculated or was input (as a constant parameter in the user interface) were made null. In most cases the order of NT_d is one. Hence, the process of normalization returns the same value as before. NT_d can assume a continuous floating-point value. This indicates that the optimizer can calculate the gradients with respect to NT_d during the iterative design process at all the possible points. Moreover as the depth was already a design variable, it is not practical to make NT_b a design variable. With the addition of NT_d as a design variable, MDO has the capability of analyzing the system with low as well as high values of cargo above the deck.

5.4 Effect of NT_d as Design Variable

Since NT_d is made a design variable, it is imperative to verify the results of MDO at low and high values of NT_d . Though economically unviable, NT_d can be as low as one. It is assumed that NT_d will not be able to take values beyond 5.5 because the GM constraint is violated. Table (5.2) indicates how the other parameters perform with respect to the change in NT_d . Table (5.2) shows the variation of the other design parameters when the upper bound of the number of tiers on deck is increased. Figure (5.3) shows the graph of the same variation, but with normalized parameters. The normalization is achieved by dividing the value of the parameter by one of the largest values in its own list. This is done to scale all the parameters to order (1) and portray their variation with respect to each other on the same graph. Some conclusions can be drawn on inspection. The SHP varies in the same way as the speed does. As the speed increases or decreases, so does the SHP. This is because SHP is dependent on the speed of the ship.

Table 5.2: Variation of Design Parameters with respect to NT_d

NT_d upper Bound	1	2	4	5	5	6
Loa (m)	299.9*	300*	299.9*	300*	300*	300*
Beam (m)	40*	40*	40*	40*	43.9	45*
Depth (m)	24.7	23	20**	20**	22**	22.5**
Draft (m)	10	10.2	10.6	10.8	11.9	12.2
Speed (m)	18.8	18.6	18.6	17.3	17	17.2
NT_d	1*	2*	4*	4.53	5*	5.2
c1	0.57	0.55	0.4	0	0.04	0
Displacement (T)	65540.9	69039.1	76124.2	81219	97387.1	102976.3
SHP	26262.3	25675.1	26493	25096.8	25418.6	27866.2
GM (m)	0.34	0.3**	0.31**	0.33**	0.33**	0.33**
RFR (\$/MT/nmi)	0.002175	0.001856	0.001413	0.001366	0.001286	0.001264

* indicates that upper bound is reached.

** indicates that constraint is active

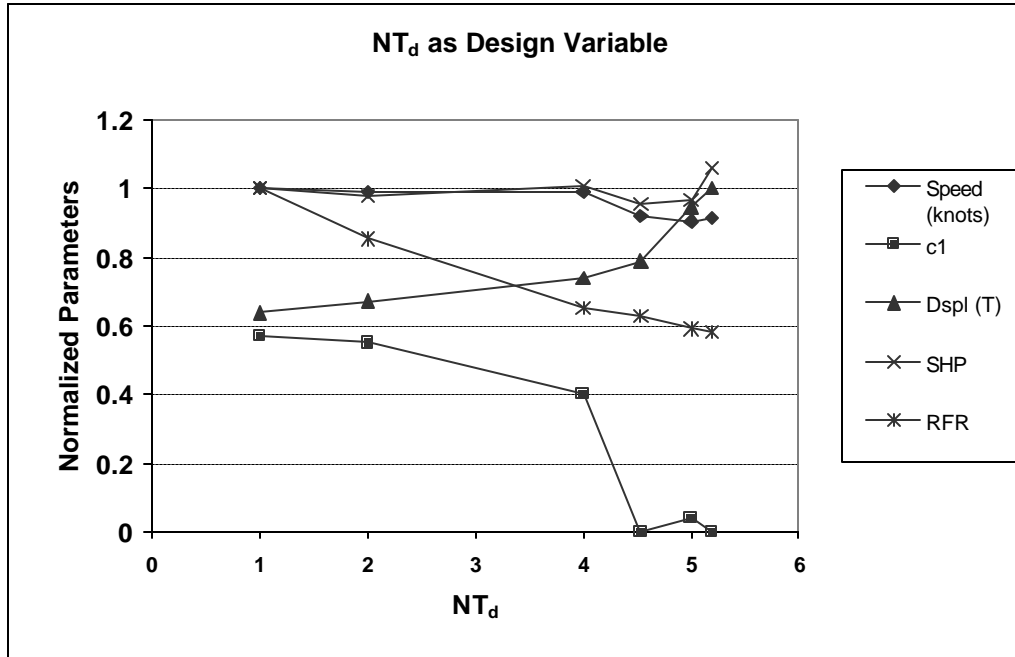


Figure 5.3: Variation of Design parameters with respect to NT_d .

When there is lesser cargo on deck ($NT_d = 1,2,4$), the value of $c_1 \cong 0.5$. A value of c_1 close to 1.0 indicates a fine ship and 0.0 means a full ship. This implies that since the cargo carried is lesser and the speed is higher, the optimizer prefers to reduce the resistance of the ship by increasing the value of c_1 . Hence, the speeds are similar and so is the SHP for low values of cargo carried. For the lower values of NT_d , the depth is higher than what is imposed by the B/D constraint. This is because, as there are less NT_d , the cargo being carried on deck is lesser. But the optimizer has to increase cargo if it has to reduce the RFR. This can be done by increasing the amount of cargo being carried below the deck. The optimizer does this by increasing the depth of the ship. But as the increase in depth also triggers an increase in the steel mass, the optimizer finds an optimum point where the increase in steel mass compensates for the increase in cargo capacity.

It is interesting to note that the GM constraint is active in all the results for different NT_d . Generally a lesser NT_d would imply an increase in the GM which would not make the GM constraint active. But the optimizer finds an optimum point by either increasing depth or by increasing the NT_d . When it is limited by the upper bound of NT_d , the optimizer increases depth. This happens until the increase in steel mass and the cargo

below deck just satisfies the GM constraint. Thus, it is useful that NT_d is a design variable because now the optimizer can minimize RFR with greater tools to control the cargo. This is very important for future analysis because the dependence of cargo on NT_d might become low as compared to the depth (i.e.), it might become cheaper to increase the cargo below deck than to add cargo on deck. In such a case, it might seem plausible for the optimizer to reduce NT_d optimally and increase depth to produce the best results with respect to the RFR.

It can also be noted that RFR decreases with increase in NT_d . But at each stage, the GM constraint prevents the design process from further reducing the RFR. The displacement increases with the increase in NT_d . This is expected because an increase in cargo means an increase in the weight of the ship. Increase in weight increases the displacement. This is reflected in the values of the draft. The draft increases with displacement. This is because the length and the beam are fairly constant at the optimized point for all the results of varying the NT_d . It is also interesting to note that when the upper bound for NT_d is 5, the optimizer arrives at a value of 4.5 for NT_d when the beam is 40 meters. When the upper bound of the beam is increased, NT_d takes the value of 5, which is the upper bound. This happens because an increase in beam increases the GM. This means that more cargo can be accommodated until the GM constraint is satisfied. Thus NT_d can be varied as a function of design variables like beam that affects the GM constraint. Table (5.3) shows the variation of NT_d with respect to the beam and other design variables and Figure (5.4) indicates this phenomenon graphically. This means that if the beam were to be constrained below what is being used in these analyses, then the NT_d will assume values that will satisfy the GM constraint and give the minimum value for the RFR. This was not possible if the NT_d were to be input as a constant value at the beginning of the iteration. If the user were to manually change NT_d with respect to an upper bound on the beam, the user will have no idea about the variation of NT_d with beam. He will not be in a position to input an appropriate value of NT_d . The variation with other design variables were also noted but were not of much significance. Length tended towards its upper bound. Depth varies as constrained by the B/D constraint. That is so because no other design variable affects the GM constraint as the beam. Draft changes with increasing cargo and hence increasing displacement. Changes in other

variables are reflected on the NT_d only if they affect the beam directly or indirectly. For example, a decrease in length goes on to decrease the depth because of the constraint $Loa/Depth > (Loa/Depth)_{min}$. A decrease in depth implies a decrease in the beam because of the B/D constraint and hence a decrease in NT_d .

Table 5.3: Variation of NT_d with beam

Loa (m)	300*	300*	300*	300*	300*	300*
Beam (m)	20*	25*	30*	35*	40*	45*
Depth (m)	10**	12.5**	15**	17.5**	20**	22.5**
Draft (m)	5.3	6.3	8	9.4	10.8	12.2
Speed (m)	16.5	17	17	17.3	17.3	17.2
NTd	0.94	2.5	3.18	3.86	4.53	5.2
C1	0.11	0.45	0.01	0.01	0	0
Displacement (T)	19563.4	26463.9	45173.3	61897.6	81219	102976.3
GM (m)	0.28**	0.29**	0.31**	0.32**	0.33**	0.33**
RFR (\$/MT/nmi)	0.003214	0.002487	0.001756	0.001514	0.001366	0.001264

* indicates that upper bound is reached.

** indicates that constraint is active

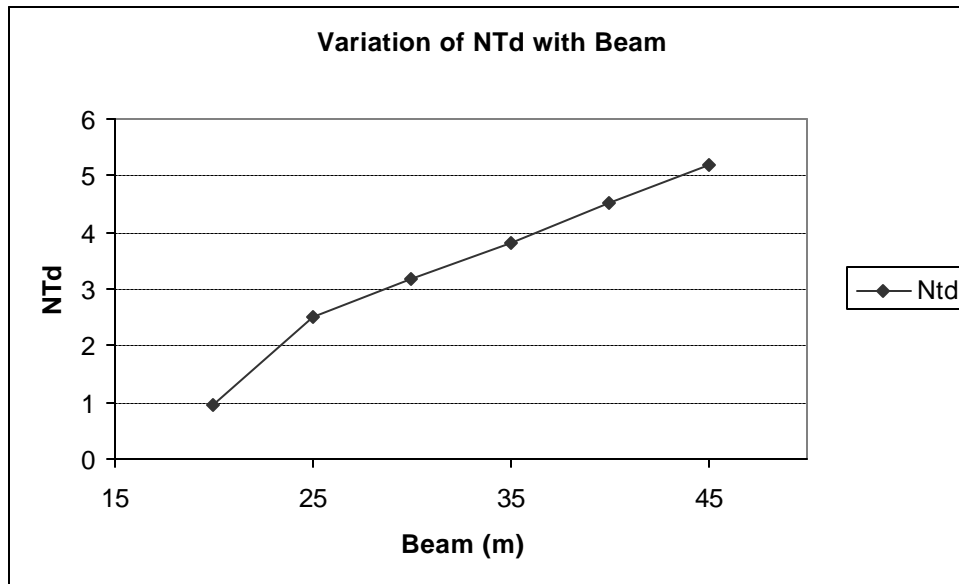


Figure 5.4: Variation of NT_d with beam

Thus, it is useful to have the number of tiers on deck as a design variable. This gives flexibility to the optimizer to manage cargo either below or above the deck. The amount of cargo being carried is a very important consideration for economics of a container ship. Hence the optimizer must be given a tool to increase or decrease the cargo directly rather than through the geometric design variables. Moreover, any empirical formulation for the calculation of NT_d might change with time. Having NT_d as a design variable implies that the user need not be aware of such changes that affect NT_d . As is seen from Figure (5.3), RFR decreases with increasing NT_d . But there is a limit on the NT_d because of stability constraints. Hence, a balance has to be found to increase the NT_d of the ship in order to reduce RFR. This has to be achieved without effecting the stability of the ship.

Chapter 6: Ballast

6.1 Need for Ballast

The introduction of the number of tiers on deck as a design variable gave the design process the freedom to change the amount of cargo being carried above deck during runtime. This implied that maximum cargo could be carried on deck and the minimum stability requirement could also be met. The mass that is being added due to the excess cargo is added above the deck. Hence, there is a shift in the center of gravity of the ship upwards. Due to this, the design is actually limited on the maximum cargo that can be carried given a certain set of geometric dimensions. The optimizer tends to maximize length and beam and move it towards the upper bound. This enables the optimizer to carry the maximum cargo possible.

It was decided that the optimizer should be given another tool to manipulate the cargo without having to increase the values of the geometric dimensions. Most of the ships are limited to certain maximum geometric dimensions because of the capacity of the shipyard where the ship being built. The cargo cannot be increased below deck because the number of tiers below deck is again a factor of the depth. So, the only recourse is to increase the number of tiers on deck. The primary reason why the optimizer cannot do that is because increasing NT_d reduces GM and hence adversely affects transverse stability. An alternative to increasing GM is to reduce the overall CG of the ship. This can be done by increasing the engine plant mass. The CG of the engine plant is well below center of gravity of the ship. The optimizer can increase the engine plant mass by increasing the speed of the ship. This is because the mass of the engine plant is proportional to the speed of the ship in the following manner.

$$W_m = C_m * \left(\frac{BHP}{1000}\right)^{0.72} * C_f \quad (6.1)$$

where,

W_m = Propulsion Machinery Weight

C_m = Machinery weight coefficient = 215

SHP = Shaft Horsepower

C_f = Coefficient to convert long tons to metric tons = 0.9843

And
$$SHP = \frac{R_{total} * V}{746 * \eta} \quad (6.2)$$

where,

- R_{total} = total resistance
- V = Speed of the ship
- η = Propulsive efficiency of the ship.

Hence, an increase in the propulsive machinery weight can be brought about only by increasing the speed of the ship. This would mean an increase in the total SHP. This means an increase in the fuel and engine plant mass. Increasing the engine plant mass and fuel mass would increase both the initial costs and the operating costs and hence increase RFR. In the current formulation, it is possible to increase the capacity of the engine to any desired value but the corresponding cost and weight of the engine also increase. So, increasing engine plant mass is not a desired alternative keeping in mind a long term objective of increasing the cargo capacity of the container ship.

6.2 Another Viable Alternative

Ballast is an alternative to increasing the number of tiers on deck and satisfying the GM constraint. Ballast is mass added to the ship to increase the weight of the ship. Ballast is usually added to the ship during the lightly loaded conditions to ensure a minimum draft. This draft is required for minimum propeller immersion. Usually, there are designated areas in the ship like wing tanks and double bottom tanks where ballast can be added. The location of ballast in the ship is usually low to lower the CG of the ship. The material used for ballasting a ship can also vary from water to iron filings.

In a container ship, ballast forms an essential part of the displacement. Sometimes the weight of the ballast can be as large as twenty five percent of the total displacement in the case of a fully loaded ship. Figure (6.1) shows the data collected from several container ships and indicates the ratio of ballast and displacement in a fully loaded condition.



Figure 6.1: Ballast / displacement from several ship data

Figure (6.1) shows that ballast is an important part of a container ship’s weight. Typically the value of ballast is between 20-25 % of the total displacement. Table (6.1) gives an indication about the displacement of the ship achieved after the process of optimization. For example, for a ship of displacement 80,000 MT, adding ballast of 20,000 MT would not be unreasonable. The addition of this mass by itself would not serve any purpose. In the case of the formulation being used for MDO, a ship that is fully loaded is considered. So, it would seem futile to add any mass that is not cargo to the already “heavy” ship. It is more relevant to consider where this mass has to be added. This would mean that the effect of adding the ballast to the center of gravity has to be considered.

6.3 Effect on Center of Gravity

The weight of the ballast being considered is substantial when compared to the displacement of the ship. Hence, the position of the ballast will affect the overall center of gravity of the ship. The position of the cargo is such that it raises the CG of the ship. So, if ballast has to be added, it has to reduce the CG of the ship. Hence, it was decided that the ballast is to be added in the double bottom space of the ship. This space is the lowermost compared to any other tank on the ship. The center of gravity of ballast placed in these tanks is assumed to be equal to the height of the geometric center of the tank

when viewed in the transverse direction. This height is equal to half the double bottom height (DBH). The DBH is assumed to be 1.83 meters but the user is given the option of changing this value in the user interface. As the material used for the ballast is uncertain, only the mass of the ballast is evaluated. If the material were known, the volume of the double bottom tank would have to be calculated. The maximum mass of ballast could be calculated if the specific gravity of the material was known.

6.4 Ballast as a Design Variable

It cannot be assumed that ballast will be a constant value. This is so because the displacement varies with each iteration. The maximum ballast that can be safely added is about twenty percent of the displacement of the ship. Since the displacement changes, the ballast can also be changed with the displacement keeping the ratio constant during the entire process. But this would mean that even if there were not much cargo to be carried (the designer might limit NT_d), the weight of ballast would still bear a constant ratio whereas the ballast required for optimum design is much lesser. The user cannot be expected to enter constant ballast for the entire process at the start of the optimization. This would mean that the optimizer is not given the necessary flexibility to manipulate the cargo.

Adding too much ballast means increasing the weight of the ship with mass that does not have economic value. But adding ballast at the double bottom tanks will reduce the CG of the ship. This will facilitate the addition of cargo above the deck, which is at a higher CG. These conflicting notions of adding ballast make it ideal to make ballast a design variable. Ballast can be a floating point variable and is thus continuous. This makes it possible for the optimizer to find the gradients for all possible values of ballast.

Therefore, ballast is added as a design variable. In the weight module, once the entire weight is calculated, ballast is added to the weight. This would also force the DOT module to adjust the design variables to increase the displacement of the ship. Weight has to be equal to displacement. This is brought into effect by the equality constraint. The addition of ballast will also change the center of gravity of the ship accordingly. This change also has to be taken into account.

The CG of the ship is now calculated as follows:

$$CG_{\text{ship}} = \frac{CG_{\text{orig}} * W + CG_{\text{ballast}} * W_{\text{ballast}}}{W + W_{\text{ballast}}} \quad (6.3)$$

where, CG_{ship} = center of gravity of ship.
 CG_{orig} = CG without considering ballast weight
 W = Weight of ship without ballast
 W_{ballast} = Weight of ballast
 CG_{ballast} = CG of the ballast = half of DBH (= 0.915 m by default)

The lower bound of ballast is zero. It is advisable to start the iteration with an initial value of zero because addition of ballast by itself increases RFR. The units of ballast are metric tons. As the value of ballast is much higher ($\sim 10^4$) than the other design variables, it has to be appropriately scaled before the process of optimization. The other design variables are scaled down with respect to their initial values. In the case of ballast, the initial value is usually zero. So, it was thought to change the units of ballast from tons to 1000 tons. This brings down the order of ballast to 1~10. Hence, the gradients calculated for the ballast by the optimizer are in the same range as the other design variables. Though the user inputs the initial value and bounds for ballast in 1000's of tons, the calculations have been performed after considering the value in the same dimensions as the weight of the ship.

6.5 Effect of ballast on the Design

Before ballast was added as a design variable, the optimizer could not add many tiers on deck. Since, in the earlier formulations before NT_d was considered as a design variable, NT_d was dependent on the beam, for a value of 43.0 meters for the beam, NT_d was constant at 5.5. At this value of NT_d , with the addition of the B/D constraint, the optimizer would choose to decrease the length of the ship to achieve transverse stability. A decrease in length of the ship prompts a decrease in the overall displacement. This means that the GM of the ship increases. At the optimized point, the parameters are such that the GM constraint is active (i.e.) the GM of the ship is equal to the minimum GM required. Once ballast is added, the CG of the ship decreases and the optimizer can

increase the length of the ship to reduce the RFR. Table (6.1) illustrates this effect. During the calculations, NT_d remained constant at 5.5.

Table 6.1: Effect of ballast on the length and RFR with constant NT_d

Results	1	2	3	4	5
Loa (m)	205.123	230.855	260.348	281.551	299.92
Beam (m)	43.0	42.983	42.968	43.0	43.0
Depth (m)	21.495	21.492	21.486	21.508	21.5
Draft (m)	12.442	12.13	12.164	12.204	12.33
Speed (kn)	25.532	21.354	19.706	18.529	17.24
Ballast (Tons)	0.269	999.94	3486.35	4655.346	5100.23
RFR (\$/MT/nmi)	0.001722	0.00147	0.00121	0.00116	0.00109

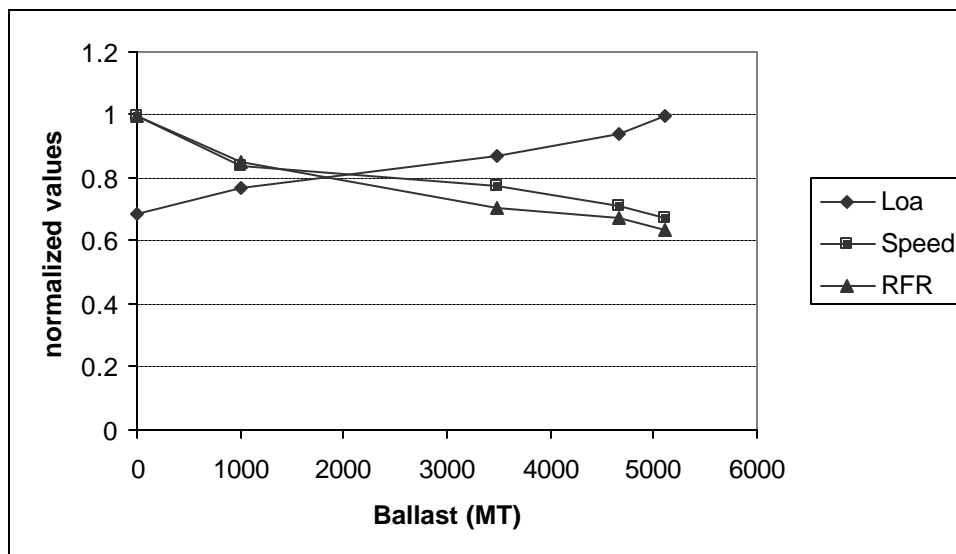


Figure 6.2: Variation of normalized parameters with respect to Ballast

As is evident from Figure (6.2), an increase in ballast for a constant NT_d (=5.5) progressively increases LOA, which in turn decreases RFR. This is so because increasing Loa would mean an increase in the cargo of the ship. Increasing cargo keeping all other factors constant reduces the RFR of the ship. Even though ballast is not a commodity that has any economic significance, it enables the ship to carry more cargo. This is done by

increasing the length of the ship. The optimizer could also have accomplished this by increasing the NT_d . The effect of ballast on NT_d is discussed in the next section. The process of increasing the cargo is carried on until it can reach an optimized solution where the addition of further ballast is detrimental to the RFR. In this case, the addition of further cargo is not possible without a mammoth increase in the value of ballast. All the succeeding results were achieved after starting from a current value of ballast. This means that the second solution was got after starting from the first optimized set of design variables. The optimizer was prompted into arriving at a better solution each time. This was done by increasing the upper bound of the ballast each time enabling the ship to carry more cargo. Even though each solution was the best for the conditions given, the RFR continued to decrease with increase in length. The best results were obtained for the maximum length possible (= 300 m for the examples in Table (6.1)). It is interesting to note the reduction in the speed of the ship. The addition of ballast reduces the requirement to increase power plant and fuel mass to meet the stability requirement. Consequently, the speed of the ship is reduced. The optimizer is always trying to increase cargo and reduce the SHP of the ship.

6.6 Ballast and NT_d

Once the effect of ballast is known on the existing design variables, it is important to consider the effect of ballast on NT_d . To test the effect of ballast on NT_d ; length, beam and depth will have to be kept more or less a constant with the variation of ballast. Draft and c_1 will be allowed to change because an increase in ballast implies an increase in the weight and this will lead to an increase in the displacement. If length and beam are kept constant, the displacement can be varied only by the C_b and the draft of the ship. C_b is largely controlled by the hull shape, which varies with c_1 .

Ballast was considered as a design variable because of its effect in reducing the CG of the ship when it is placed in the double bottom space. This would imply that a larger cargo could be added above the deck. This cargo varies directly with the NT_d . So, addition of ballast would mean the possibility of increasing NT_d without compromising on stability. The addition of ballast would increase the weight and hence the displacement. This would indicate an increase in the draft prompting an increase in the

resistance of the ship. Hence the positive effects of ballast will hold true until increase in resistance becomes more significant. The amount of ballast that can be added is also limited. It is mentioned earlier that the maximum amount of ballast that can be added to a container ship is about 20-25% of the weight or displacement of the ship. Care should be taken to make sure that ballast does not go way beyond this ratio even though the addition of further ballast is economically viable. It might not be physically possible to add ballast beyond a certain weight because of space constraints in the double bottom tanks. In the current formulation, ballast is considered as weight rather than volume. This is possible as long as the weight of ballast is below acceptable ballast weights in container ships. So the process of checking the ballast to displacement ratio is the following. First set an arbitrary upper bound for the ballast and run the optimization process. On completion, check the optimized ballast to displacement ratio. If this is within acceptable norms, the upper bound on the ballast can be increased accordingly if it is economically viable. If the ratio is higher, then the upper bound on the ballast has to be reduced and the process must be repeated again for stable results. The process of checking on ballast can also be accomplished by having a constraint on ballast and displacement. This constraint should be added in the future. Table (6.2) shows the variation of important parameters with an increase in the upper bound of the ballast.

From Table (6.2), it can be inferred that length and beam have reached their upper bounds. Depth is also fixed with respect to the beam due to the B/D constraint. The variation of speed with respect to ballast is not much. This is also illustrated with the variance of SHP, which depends on speed. The value of GM is such that the GM constraint is always active. The GM constraint ensures that the GM value never goes below the minimum. The variation of the other parameters with respect to ballast is illustrated in Figure (6.3).

Table 6.2: variation of parameters with respect to ballast.

Loa(m)	299.7	300*	300*	300*	300*	299.8	300*	300*
Beam(m)	40*	40*	40*	40*	40*	40*	40*	40*
Depth(m)	20***	20***	20***	20***	20***	20***	21	22.5
Draft(m)	10.8	11.9	12.4	12.9	14.1	15.3	16***	17.1
Speed(m)	16.8	17.5	17.6	17.6	17.6	17.1	16.6	17.1
Ballast (T)	0	5000	7500	10000	15000	19600	25000**	30000**
NTd	4.52	5.13	5.45	5.75	6.4	7.13	7.33	7.19
Displacement (T)	80897.2	91202.4	96535	101370.3	111130.3	120844.7	131399.5	138464.4
SHP	22683.4	26524.6	27569	27471.4	27173	24748.2	25240.6	27734.8
GM(m)	0.33***	0.32***	0.32***	0.32***	0.33***	0.37***	0.34***	0.33***
RFR (\$/MT/nmi)	0.001367	0.00127	0.001228	0.001188	0.001111	0.001044	0.001057	0.001082
TEUb	4628	5039	5263	5457	5856	6297	6620	6667

- * Indicates an upper bound is reached.
- ** Indicates a lower bound value
- *** Indicates that a constraint is active.

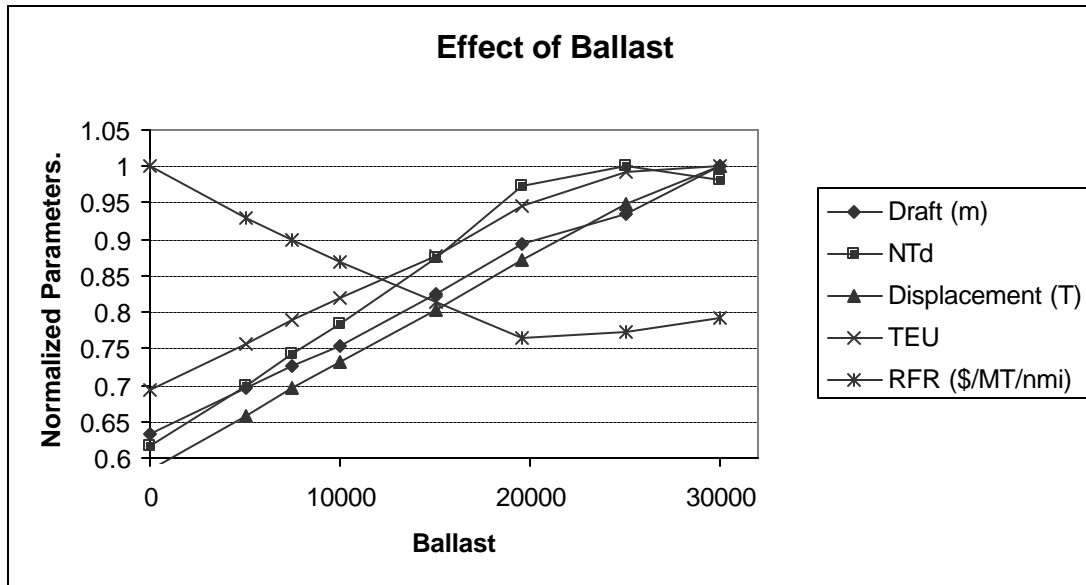


Figure 6.3: variation of normalized parameters with respect to ballast.

The above figure illustrates the variance of normalized parameters with respect to the increase in the upper bound of ballast. In most of the examples except for ballast = 19600 Tons, the ballast has reached its upper bound. In the last two cases, the lower bounds were 25000 and 30000 Tons and it came in to the lower bound. With the increase

in the ballast values, the displacement evidently increases. This is because the weight of the ship increases and hence displacement also increases. The increase in displacement translates to an increase in draft. The length and beam are constant at their upper bounds. C_1 stays at 0.0 because the optimizer prefers a full ship to carry more cargo. Hence, the displacement of the ship can be increased by increasing the draft. This is also reflected in the above figure. RFR decreases initially with increasing ballast but after a certain value ($\cong 20000$ T), increasing the ballast only increases the RFR. There is a marked increase in SHP due to increase in resistance.

NT_d increases with the increase in ballast. This is because the CG of the ship is lowered with an increase in ballast. Hence the cargo can be added above the deck until the metacentric height is just above the minimum value required for transverse stability. This is a desirable effect because with the increase in NT_d , the TEU's also increase. An increase in cargo decreases the RFR of the ship. An increase in ballast does not change the length and beam of the ship. Hence it is a linear increment to the weight and hence to the displacement. CG also decreases linearly which facilitates a linear increase in NT_d . RFR decreases with increasing ballast. Hence it is advisable to have ballast as a design variable. Figure (6.4) shows the ratio of ballast and displacement as a percentage.

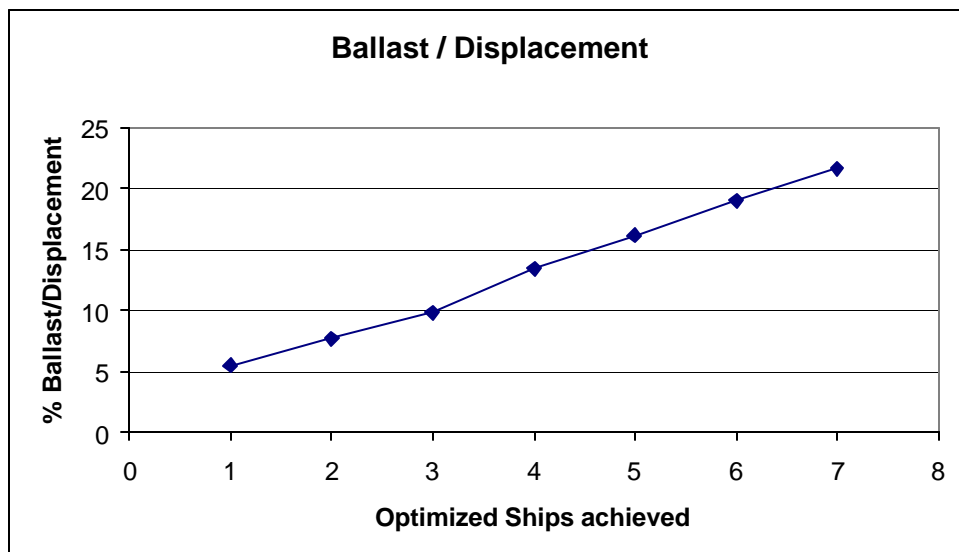


Figure 6.4: Ratio of ballast to displacement

The above figure indicates that though the ratio between ballast and displacement is increasing the maximum reached is about 22%, which is below the maximum ratio of 25%. Hence these are valid example problems. Thus it is decided to have ballast as a design variable to supplement NT_d which is also a design variable. With the introduction of ballast, the optimizer can increase NT_d and the cargo beyond what it could achieve without ballast.

Chapter 7: Tools For Evaluating Results of Optimization

7.1 Introduction

The MDO design involves many design variables and parameters, which collectively affect the result of optimization. In this environment, it is very difficult to come to a conclusion as to the effect of each individual parameter. It is not possible to visualize the dependence of the objective value on the various parameters. Due to this, tracking an error in the formulation or in the value of a parameter becomes too cumbersome. Sometimes, the errors are not visible during a normal course of optimization. It might become relevant only when some constraint is imposed on the problem, which stresses more on the faulty parameter. Since, this software tool will be employed for preliminary design, there might not be standards available for comparison of results. Hence, tools have to be employed, which can systematically evaluate the results of optimization. These tools must be able to individually examine the effect of parameters and design variables and also verify the veracity of the final result. Results must be checked for at least a local optimum and whether all the resources available to MDO have been optimally utilized. Three diagnostic tools have been used to evaluate the results of optimization.

7.2 Alpha Plots

The alpha plot is a diagnostic tool used to identify barriers in the process of optimization. The response of the system is plotted along a straight line through the design space. The line is obtained by joining two points (X_1 and X_2) identified in the design space.

$$X = \alpha X_1 + (1 - \alpha) X_2 \quad (7.1)$$

where,

X is the vector of design variables.

X_1 and X_2 are design points that can be selected in a variety of ways including the following two:

1. They can be two optimum points that have been arrived at with the same initial design variables. In this case, it should be verified as to why, the optimizer arrived at both the designs.

2. They can be the initial point and the final design point. The approximate values of design variables can be found out for feasible regions.

Alpha plots can also be used to detect discontinuities in the system. Table (7.1) is a table of design variables and parameters between two points on the design space that show discontinuity in one of constraint function values.

Table 7.1: Change in Design variables and constraints between two design points.

Loa	B	D	T	V	Ballast	NT _d	g ₁	g ₂	g ₃
247.9215*	42.70	21.35	13.48	17.2425	5776.9	6.037	0.0007	-0.0007	0.6682
248.9631	42.70	21.35	13.48	17.24147	5812.6	6.037	0.0007	-0.0007	0.6794
250.0046	42.69	21.34	13.48	17.24044	5848.4	6.038	0.0007	-0.0007	-0.6737
251.0462	42.69	21.34	13.48	17.23941	5884.1	6.039	0.0007	-0.0007	-0.6581
252.0878	42.68	21.34	13.48	17.23838	5919.8	6.039	0.0007	-0.0007	-0.6426
253.1294	42.68	21.34	13.48	17.23735	5955.5	6.040	0.0007	-0.0007	-0.6271
254.1709	42.67	21.33	13.48	17.23632	5991.2	6.041	0.0007	-0.0007	-0.6116
255.2125	42.66	21.33	13.48	17.23529	6026.9	6.042	0.0007	-0.0007	-0.5915
256.2541	42.66	21.33	13.48	17.23426	6062.6	6.042	0.0006	-0.0006	-0.5762
257.2956	42.65	21.32	13.48	17.23323	6098.3	6.043	0.0006	-0.0006	-0.5658
258.3372	42.65	21.32	13.48	17.2322	6134.0	6.044	0.0006	-0.0006	-0.5506
259.3788	42.64	21.32	13.48	17.23117	6169.7	6.044	0.0006	-0.0006	-0.5355
260.4203**	42.64	21.32	13.48	17.23014	6205.4	6.045	0.0006	-0.0006	-0.5205

* indicates starting point

** indicates end point

In Table (7.1), g_1 and g_2 are the enforcement of the weight = displacement constraint. As is indicated by Table (7.1), g_1 and g_2 are equal and opposite of each other. Please refer chapter 3 for constraint equations. The GM constraint is shown by g_3 . Figure (7.1) is a plot of g_3 versus α . This plot shows the discontinuity in g_3 constraint. It is evident that there is a discontinuity in the formulation that has produced this jump in the GM constraint. Various other parameters were plotted with the change in α and it was noticed that the discontinuity arose because number of tiers below deck (NT_b) is considered with its integer value. Thus, as depth varies, NT_b assumes discrete values. The number of tiers on deck (NT_d) can take up floating point values because it is the average of the number of tiers that stack up the TEU's on deck. NT_b requires that all the tiers below deck are stacked equally to ensure stability. This problem was corrected by

introducing a continuous function for NT_b that takes floating-point values for the design process. The output of MDO presents NT_b as an integer value.

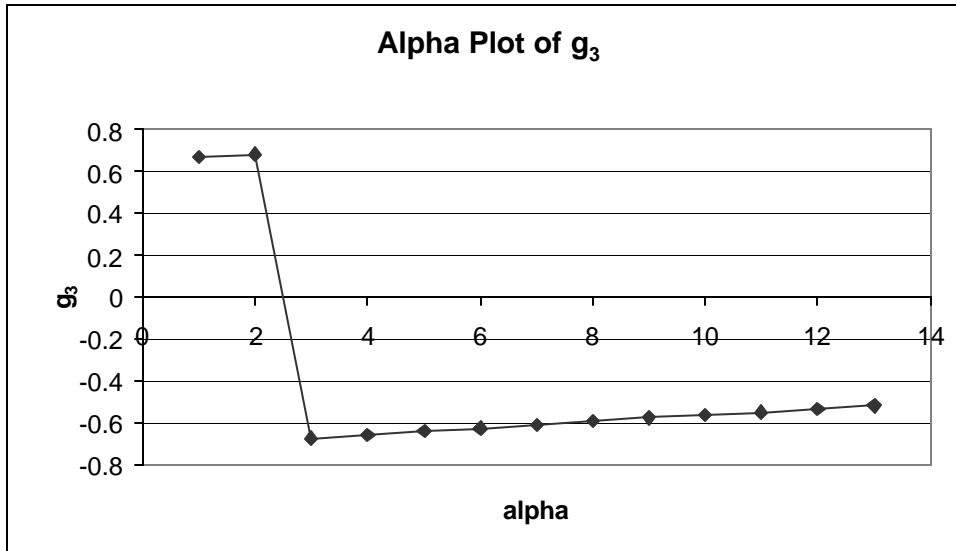


Figure 7.1: Alpha Plot of constraint g_3 (GM constraint) indicating discontinuity.

7.3 Sensitivity Analysis

The object of this type of analysis is to investigate the sensitivity of the objective function with respect to the design variables at the optimized point. At the optimized point, the design variables are varied around the optimized point and the objective values are calculated. During this process, all the other design variables are held constant at their optimized value. This process indicates to the user about design trends around the optimum point with reference to the design variables and their bounds. The sensitivity analysis also indicates the relative dependence of the objective function on the individual design variables.

Figure (7.2) is a collection of plots of sensitivity analysis of objective function to all the design variables with the following optimized values.

Table 7.2: Optimized values of Design variables used in Figure 7.2

Length (m)	Beam (m)	Depth (m)	Draft (m)	Speed (knots)	Ballast (10^3 MT)	NT_d	C_1
300	43	21.5	12.25	17.52	1.54	5.17	0.0042

The objective value at the optimized point is 0.001264 \$/MT/nmi.

■ indicates the optimum point in Figures 2 (a-h).

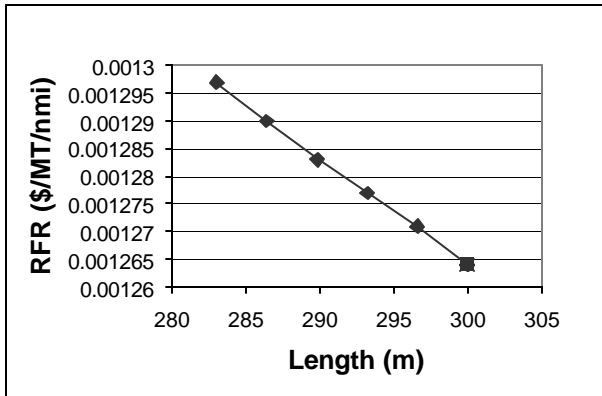


Figure 7.2 (a): Sensitivity of RFR to Length

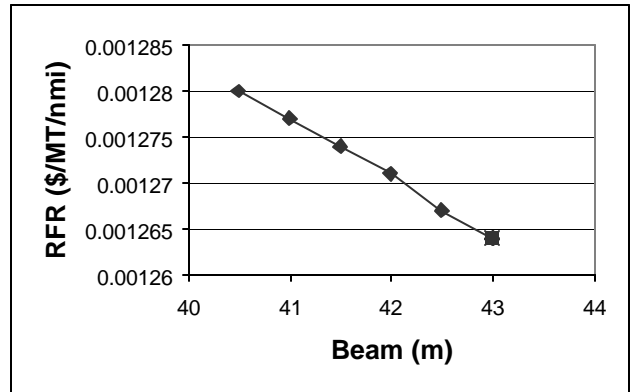


Figure 7.2 (b): Sensitivity of RFR to Beam

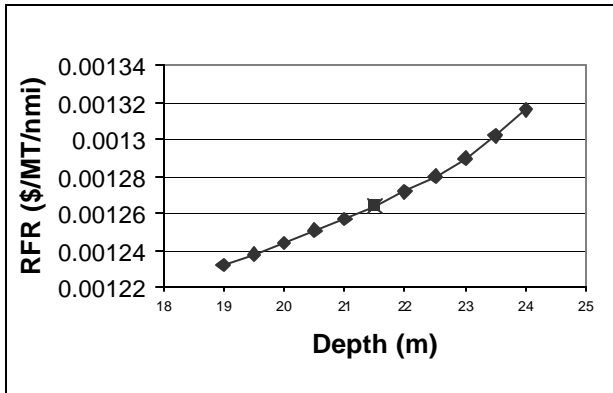


Figure 7.2 (c): Sensitivity of RFR to Depth

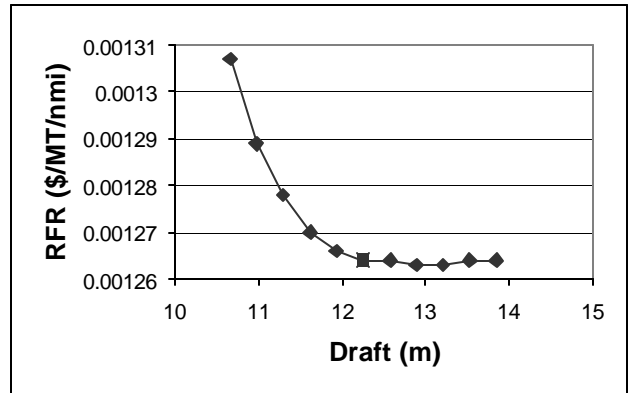


Figure 7.2 (d): Sensitivity of RFR to Draft

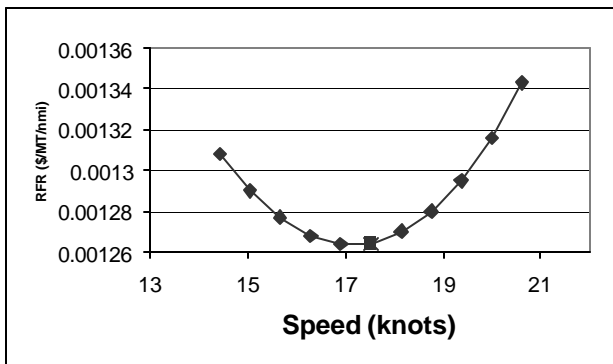


Figure 7.2 (e): Sensitivity of RFR to Speed

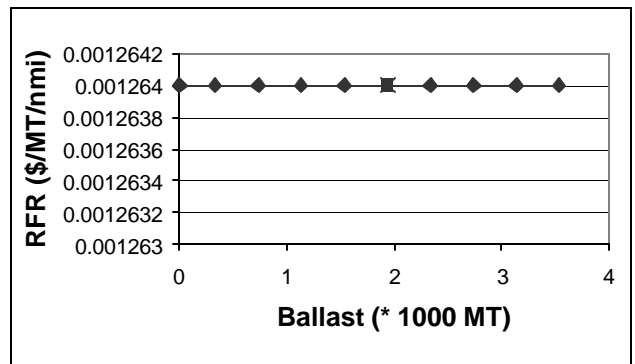


Figure 7.2 (f): Sensitivity of RFR to Ballast

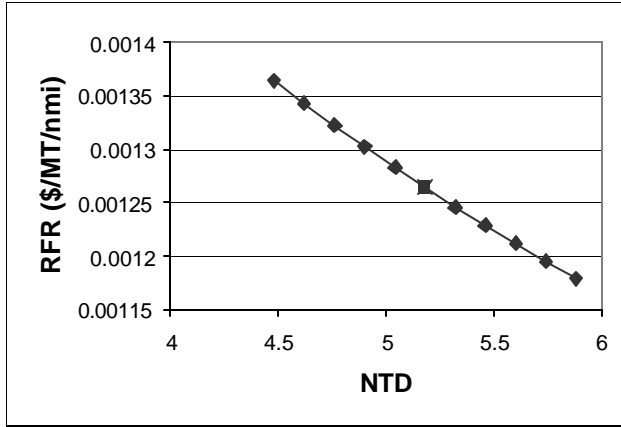


Figure 7.2 (g): Sensitivity of RFR to NT_d

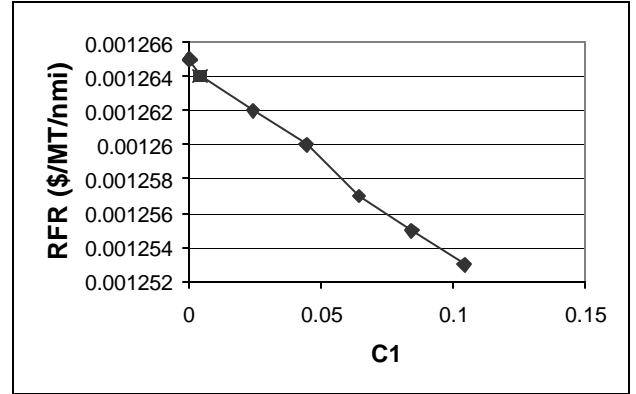


Figure 7.2 (h): Sensitivity of RFR to C_1

Figure 7.2(a) and (b) are a plots of the sensitivity of RFR to Length and beam respectively. They indicate that RFR reduces with increasing Loa and beam in the vicinity of the optimum point. The maximum value for the analysis is curtailed at 300 and 43 m respectively because they are the upper bounds and Loa and beam have reached their upper bound during the optimization process. Figure 7.2(c) indicates that RFR increases with depth. An increase in depth increases the hull steel weight of the ship resulting in greater building costs. This further increases the RFR. The optimizer to forced to reach the point because of the enforcement of the B/D constraint. Figure 7.2(d) shows that RFR is fairly constant with respect to draft after the optimized point. The value of draft is arrived at by the weight = displacement constraint. An increase in draft increases the displacement forcing the ship to carry greater cargo. Hence, the increase in resistance is offset by the increase in cargo. Figure 7.2(e) is a plot of RFR to variations in speed. The optimum point is in the lowermost point in the curve. Any increase in speed would increase resistance and hence increase operating costs. A decrease in speed would mean a decrease in the number of round trips made per annum. This sensitivity to speed indicates that a value for speed is reached on optimizing the two conditions. Figure 7.2(f) varies RFR to ballast. Here, RFR does not change with ballast. RFR is not directly affected by ballast and hence the effect of change of ballast is not realized on the RFR. Figure 7.2(g) shows a favorable trend to increasing NT_d . The optimizer cannot take

advantage of this trend because beyond the optimum point, the GM constraint is violated. Figure 7.2(h) also indicates that RFR decreases with increase in C_1 .

7.4 Restart Option

The restart option is a tool provided to the user to restart the optimization process from any point in the optimization history. The user is provided with a convenient tool to stop the optimization process anywhere he pleases and start again from the same point with or without changing any of the other conditions. The user is permitted to change the bounds of the design variables and/or change the constraint limits. The user can start re-optimization if he is not satisfied with the results of optimization. A newly converged design point in the vicinity of the previous optimum would indicate that for the same set of bounds on the design variables and constraint values, an optimum point is reached. Table (7.4) shows results from starting the optimization process from points of optimum without changing any of the bounds or conditions.

Table 7.4: Example of restart option.

Design Variables	Lower Bound	Upper Bound	Initial Value	First Convergence	Second Convergence	Third Convergence
Length (m)	130	300	200	300	300	300
Beam (m)	15	43	30	43	43	43
Depth (m)	10	35	15	21.5	21.5	21.5
Draft (m)	4	20	10	16.4	16.393	16.393
Speed (m)	4	35	12	18.3	18.287	18.287
NT_d	0	10	0	7.46	7.464	7.464
Ballast (T)	0	20000	0	19998	20000	20000
C₁	0	1.0	0.5	0.73	0.73	0.73
RFR (\$/MT/nmi)			0.001953	0.001016	0.0010155	0.0010155

Chapter 8: Technology Factors

8.1 Introduction

As mentioned in chapter 2, the modules for the optimization process make use of proven, published parameterizations. However, only simple attempts have been made to update the various parameters and coefficients to reflect current costs and construction practices. This was done because the modules that were developed were to perform in the same general manner as high fidelity modules. Many of the coefficients and parameters have been developed after referring to current container ship designs. Regression analysis was performed on some of the equations like the weight equation to arrive at a solution, which was acceptable.

Since, the MDO project is applied to design a containership, it is important to consider the economic aspects of the design. An operational analysis is required to judge whether the design that is achieved is indeed the best. In the process of optimization, MDO arrives at a design which has the least objective value (RFR) and which has satisfied all the constraints. First, the optimizer decides on a potential hull for the container ship. The optimizer then estimates the transport potential (TEU), construction costs and operating costs for each hull and then improves the design. The weight of lightship is also an important consideration. This design process may suggest that a lot of detail is involved. But in the preliminary design process, it is worthwhile to identify key players and parameters in the design process. These parameters affect the design more than others.

Once these parameters are identified, it is pertinent to evaluate how the overall design might vary with the relative change in these parameters. If the change is too drastic, then appropriate steps can be taken to concentrate on local design. For example, suppose a 10% increase in parameter 'A' increases the RFR by more than 10%, all other parameters remaining the same, then it can be said that RFR is over-sensitive to parameter A. Now, suppose, parameter 'A' is not under control of the owner of the ship, it is advisable to make the design less dependent on 'A'. On the other hand, there might be some advancement in technology that might improve the RFR. Hence, it is useful to identify key parameters that affect the design.

In the current project, efforts have been made to identify these parameters, which have been named technology factors. When these factors are identified, their relevance to the design has to be evaluated. Some of the factors used in the design reflect current market values and construction practices. But they are subject to future trends. These trends might have adverse effects on future ship design if the design is over-sensitive to these factors. When technology factors have to be considered, it has to be relevant to the economics of the design. This is because, it is assumed that all other constraints have been satisfied when technology factors are varied.

The technology factors can be applied to parameters that affect the total cost or performance of the ship. Once the specifications of the ship have been determined, it is unlikely that the dimensions of the ship will change (especially length, beam and depth) over the lifetime of the ship. So, these can be deemed as fixed costs. In this context, it is worthwhile to identify technology factors that will affect fixed costs like construction costs and those that will affect operating costs of the ship. In the following analyses, two technology factors have been identified that will affect the operating and fixed costs of the container ship.

8.2 Applications of Technology factors that affect fixed costs

(1) Cost of Steel Hull, C_{sh}

Since fixed costs have been targeted, the sensitivity of RFR to the depth was plotted. Figure (8.1) indicates this plot. This plot shows that once the optimum point is reached, the RFR increases rapidly with the increase in the depth of the ship. This is so because, the increase in depth increases the steel hull mass and adds to the weight of the ship. There is also an addition of cargo below the deck with the increase in depth but this increase does not compensate for the added costs of the increased steel mass. But length and beam have opposite effects as compared to depth. Increasing length and beam beyond the optimized point decreases the RFR because of a substantial increase in the cargo being shipped. As the steel hull weight has this effect on RFR, it was decided to find the sensitivity of RFR to cost of steel hull that can be manipulated using a technology factor.

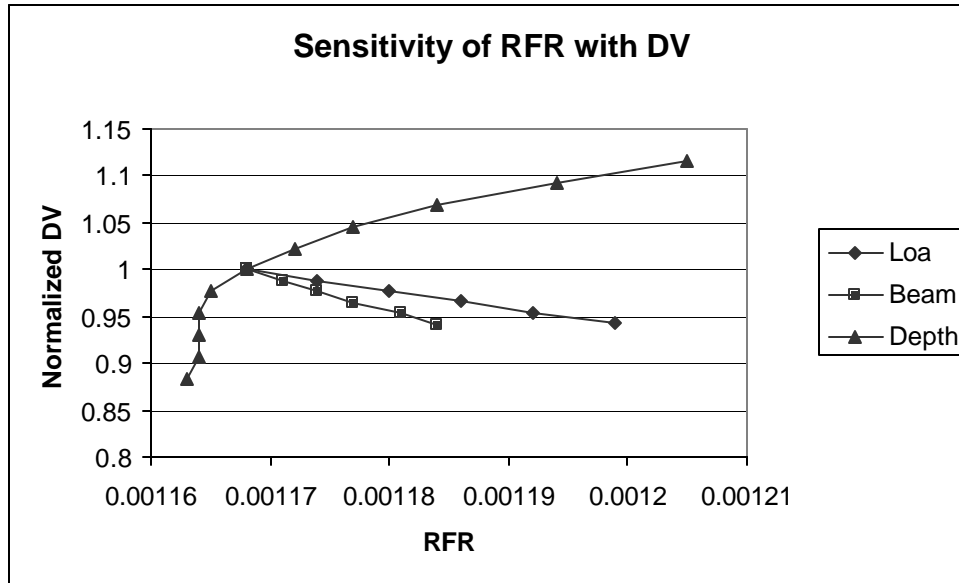


Figure 8.1: Sensitivity of RFR to dimensions of the ship. A value of 1 for the normalized design values indicates the optimum point

The material cost of hull steel is formulated as the following:

$$\text{Mats} = C_{sh} * W_s \quad (8.1)$$

Where Mats = material cost

C_{sh} = cost of steel hull

W_s = Weight of steel hull.

In the current analysis, C_{sh} has a baseline value of \$400 / MT.

Table (8.1) indicates the variation of parameters with C_{sh} . In this table, there is no alteration of any other parameter of design including design variables. As the technology factor varies from 0.5 to 1.25, the C_{sh} varies from \$200/MT to \$500/MT.

From Table (8.1) it is evident that the other design variables do not vary much from the original optimum point. The upper bound on the ballast is fixed at 2000 tons. Figure (8.1) shows the graphical variation of RFR with C_{sh} . This plot indicates an increase of RFR with increase in C_{sh} . This is because, with the increase in C_{sh} , there is also an increase in Annual Building costs and thereby

increasing the freight rate that has to be charged. The figure also indicates that the variation is almost linear with a little steepness towards the end of C_{sh} . The cost of steel is dependent on the market at the time of building the ship. But a near linear sensitivity of RFR to C_{sh} indicates that the ship builder should complete production faster for lesser RFR or develop technology that will reduce the cost of steel.

Table 8.1: Variation of RFR with C_{sh} .

Csh (\$/MT)	RFR (\$/MT/nmi)	Annual Building Costs	Loa (m)	Beam (m)	Depth (m)	Draft (m)	Speed (knots)	Ballast (T)	NT _d
300	0.001243	\$ 6242364	299.9*	43*	21.49	12.35	17.63	2000*	5.23
350	0.001252	\$ 6403911	300*	43*	21.49	12.27	17.89	2000*	5.06
400	0.001264	\$ 6528432	299.9*	43*	21.49	12.34	17.51	2000*	5.23
450	0.001281	\$ 6668329	299.9*	43*	21.49	12.41	18.43	2000*	5.26
500	0.001297	\$ 6811488	300*	43*	21.49	12.34	17.75	2000*	5.23

* indicates that an upper bound is reached.

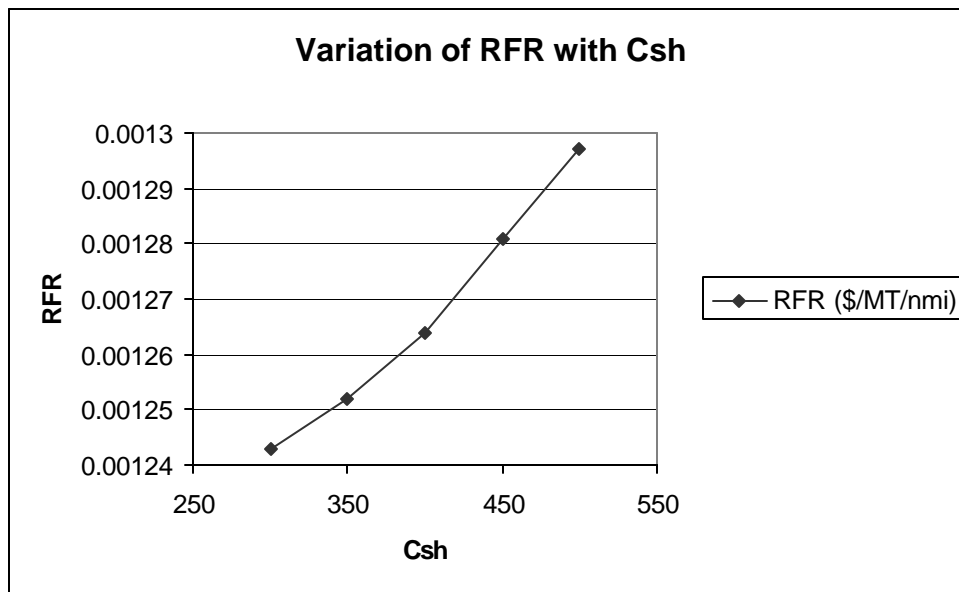


Figure 8.2: Sensitivity of RFR to C_{sh}

(2) Weight of Hull, W_s

The material cost of steel in the hull depends on two parameters – C_{sh} and W_s . It would be expected that RFR would increase with increasing W_s and the C_{sh} remaining the same because material cost has a linear relationship with W_s . Technology could be applied to develop a lighter hull and this would reduce the

material cost. Table (8.2) is a compilation of results with varying W_s . Here the technology factor is applied to W_s after it is evaluated by empirical formulas. A technology factor of 1.0 indicates that W_s is as calculated by the optimizer. A technology factor of 0.9 indicates that W_s is reduced by 10%. W_s have been multiplied by the technology factor to get the new steel weight. Results have been shown for W_s reduced by at most 10% to an increase by 10%.

Table 8.2: Sensitivity of parameters to Technology Factor for W_s

Technology Factor	RFR (\$/MT/nmi)	Lightship weight (T)	NTd	TEU	Loa (m)	Beam (m)	Depth (m)	Draft (m)	Speed (knots)	Ballast (T)
0.9	0.001268	27153	5.155	5576	299.79*	42.99*	21.49	12.03	17.48	2000*
0.95	0.001263	28231	5.17	5589	299.99*	42.97*	21.48	12.16	17.30	2000*
1	0.001257	29452	5.237	5645	299.87*	42.99*	21.49	12.34	17.51	2000*
1.05	0.001251	30601	5.279	5683	299.96*	42.99*	21.49	12.49	17.44	2000*
1.1	0.001245	31758	5.327	5721	299.99*	42.98*	21.49	12.65	17.53	2000*

* indicates that an upper bound is reached.

Table (8.2) indicates that lightship weight increases with W_s (increasing technology factor). All the design variables with the exception of draft do not vary too much. The increase in draft is to account for the increase in displacement. This is expected but RFR decreases with increasing W_s . Figure (8.3) shows a plot of RFR with changing technology factor.

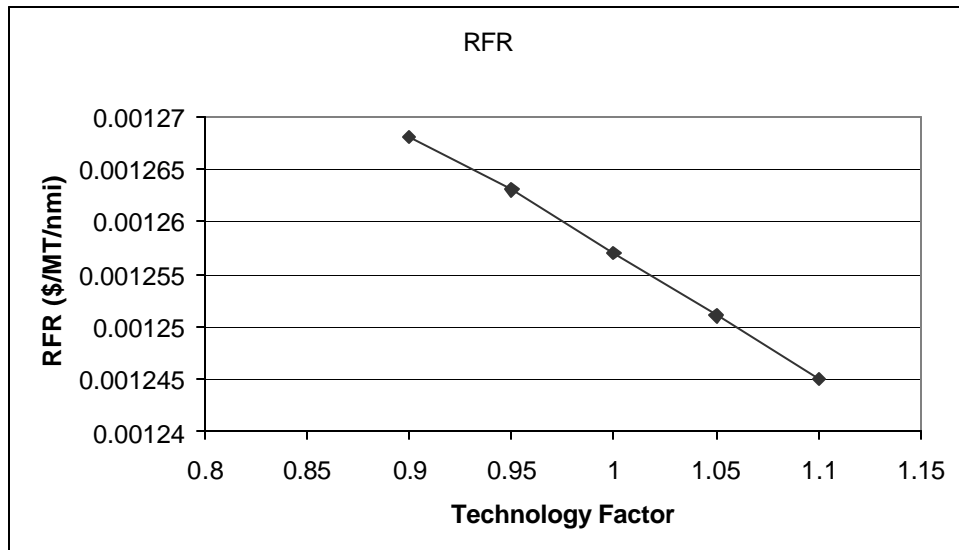


Figure 8.3: Sensitivity of RFR with W_s Technology factor

These results are counter-intuitive. The lighter hull actually increases RFR. The reason was that the heavier hull lowered the center of gravity of the ship allowing more containers to be stacked. This is indicated by the increase in TEU. A larger container capacity implies greater revenue and hence lesser RFR. The actual change in RFR is quite small but the direction of change was not expected. An investment in technology to reduce the weight of the ship is not advisable. These results on technology factors give a general direction for spending resources on technology advancement. While it is worthwhile to reduce C_{sh} , it is better to increase the weight of the ship to reduce RFR.

8.3 Application of Technology Factors Affecting Operation Costs

(1) Propulsive Efficiency, η

While considering annual operating costs, it is pertinent to target factors, which are consumed during the operation. An example of these factors is fuel costs. Speed is a design variable that plays an important role in evaluating operating costs. Figure (8.4) shows the sensitivity of RFR to speed. This plot indicates that after the optimized point (which is the lowermost point on the graph), RFR increases with increasing speed. So, it worthwhile to apply a technology factor which will subdue the effect of speed on RFR.

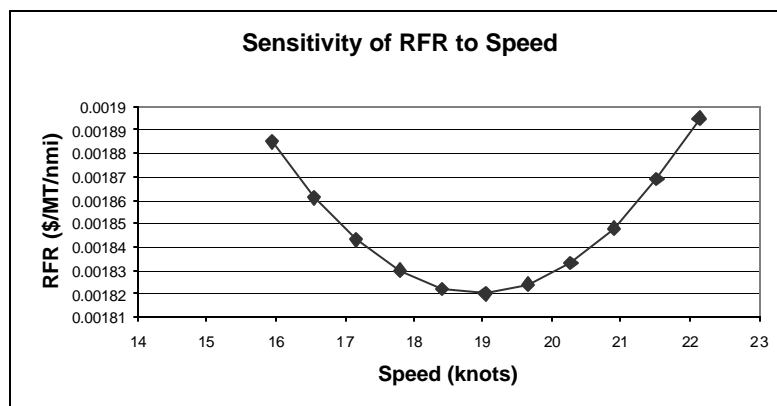


Figure 8.4: Sensitivity of RFR to speed

Speed directly affects the Shaft horsepower (SHP) in the following manner.

$$\text{EHP} = R_t * V \quad (8.2)$$

And

$$\text{SHP} = \frac{\text{EHP}}{\eta} \quad (8.3)$$

Where, EHP = Effective horsepower

R_t = Resistance of the hull

V = Speed of the ship.

SHP = Shaft horsepower

η = Propulsive efficiency

$$A_{fc} = W_f * F_{cost} * NT \quad (8.4)$$

$$W_f = \frac{\text{SHP} \times \text{Dst} \times 1.1 \times \text{SFC}}{V \times 10^6} \quad (8.5)$$

Where, W_f = Weight of fuel

F_{cost} = Cost of fuel oil

NT = Number of trips/year

Dst = Range in nautical miles / year

SFC = Specific fuel consumption

SHP has direct impact on various other operating costs like fuel costs (Equation 8.4). Since, these costs are directly proportional to SHP, it is worthwhile to reduce SHP. This can be done by decreasing R_t or V or increasing the propulsive efficiency, η . Figure (8.4) shows that reducing velocity below the optimized point also increases RFR. This happens because reducing V means making less round trips over the year. So, the optimizer chooses between making more round trips and reducing SHP. Reducing R_t is again fixed cost problem. Once the hull is fixed, R_t is more or less constant for the lifetime of the ship. That leaves the option of increasing η . This is viable because changes can be made during ship's operation to the propulsion system or machinery, which might improve η . It is also quite possible, that the systems might not perform at design efficiency after some period of operation of the ship. So, technology factor can be applied to the propulsive efficiency, which will affect RFR.

Table (8.3) shows the results of varying η and its impact on some of the factors. Figure (8.5) shows the variance of η on RFR. As the technology factor

varies from 0.85 to 1.15, η varies from 0.55 to 0.75. The technology factor is 1.0 for $\eta = 0.65$. The dimensions were the same for all the values of η . As expected, the required freight rate decreased with increasing η . This could have been possible either by greater economy (increasing η) at the same cruising speed or by increasing the speed and thereby increasing the total cargo being shipped per year. It cannot be determined in advance which method the optimizer would use to decrease in RFR. As it turned out, \dot{i} is a blend of both the effects. The speed increased with increasing η . This would have meant greater fuel weight, W_f and BHP but they too decrease because of increasing η . All the other design variables are fairly similar to the optimum values.

Table 8.3: Variation of parameters with η

n	Speed (knots)	RFR (\$/ton/mile)	W_f (T)	BHP	Loa (m)	Beam (m)	Depth (m)	Draft (m)	Ballast (T)	NTd
0.55	16.911	0.001305	1626	29758	299.74	42.99	21.49	12.39	2000	5.25
0.6	17.04	0.001296	1517	29177	299.78	42.98	21.49	12.34	2000	5.16
0.65	17.52	0.001264	1511	28649	299.87	42.99	21.49	12.34	2000	5.23
0.7	17.5	0.001239	1502	27810	299.99	42.99	21.49	12.33	2000	5.23
0.75	18.36	0.001222	1498	27179	299.86	42.99	21.49	12.31	2000	5.20

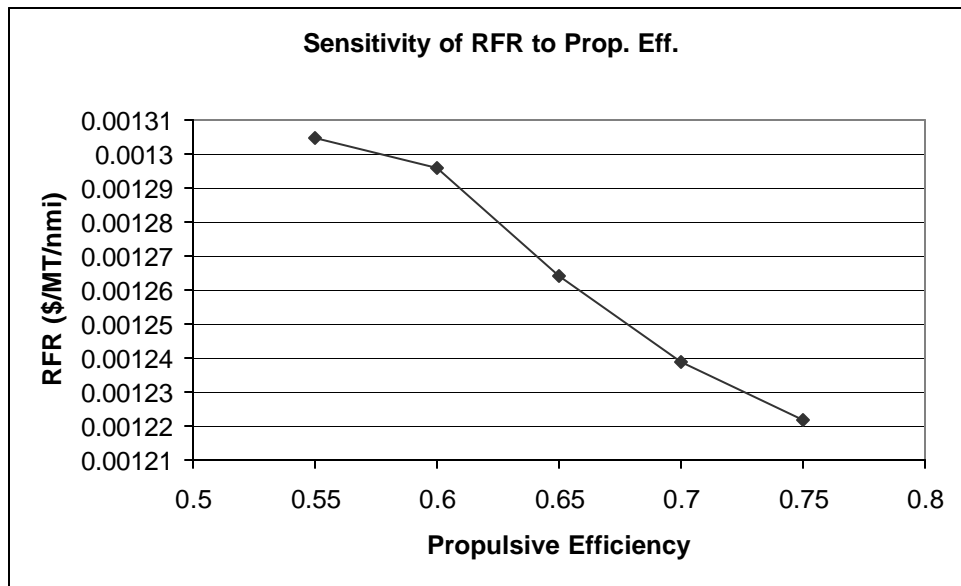


Figure 8.5: Sensitivity of RFR to η .

The base value of propulsive efficiency is 0.65. Figure (8.5) indicates that there are proportionately higher gains that can be achieved with increasing the propulsive efficiency. Efforts should be made to constantly improve the the technology to increase η .

(3) Number of Cranes (N_{crane})

The number of cranes (N_{crane}) is a factor that affects the loading and unloading time of containers in a port (Equation 8.6). This time further affects the number of round trips in a year, which determines the amount of cargo being shipped in a year. Thus, N_{crane} indirectly affects the RFR. In the current formulation, N_{crane} is calculated by assuming that there will be one crane for every 135 feet over 75% of the ship's length (Equation 8.7).

$$L_{ut} = 4 \times \frac{\text{TEU}}{\text{TSLU} \times N_{\text{crane}}} \quad (8.6)$$

$$N_{\text{crane}} = \text{int} \left(\frac{0.75 \times \text{Loa}}{41.175} + 1 \right) \quad (8.7)$$

where, L_{ut} = Loading and Unloading time per round trip

TEU = Total number of containers

TSLU = loading/unloading speed of the crane

Equation 8.7 calculates the number of cranes for the ship if the user does not specify it. The number of cranes required by a ship of $\text{Loa} = 300$ m is 6. But this is quite a large number of cranes and it is not always possible to have access to docks that can provide this facility. As N_{crane} is a factor, which affects the annual operating costs, a technology factor can be applied to it. Table (8.4) shows the results of varying N_{crane} and its impact on other parameters. As the technology factor varies from 0.5 to 1.5, N_{crane} varies from 2 to 6.

With the increase in N_{crane} , there is a marked increase in number of round trips in the year, NT. This also prompts a decrease in RFR. All other design variables are fairly constant. Figure (8.6) shows this variation of RFR with N_{crane} .

All the other design variables were the same during this analysis. Since, the number of cranes makes a substantial difference to the RFR, it is worthwhile to increase the technology factor to increase N_{crane} . There is no negative effect of increasing N_{crane} in the current formulation. But a more optimized solution can be obtained by specifying the costs of increasing the number of cranes for loading and unloading.

Table 8.4: Results of optimization of varying N_{crane}

N_{crane}	RFR (\$/MT/nmi)	NT	Loa (m)	Beam (m)	Depth (m)	Draft (m)	Speed (knots)	Ballast (T)	NT_d	C_1
2	0.001528	13.15	299.8	42.9	21.5	12	17.2	2000	5.1	0.3
4	0.001331	15.44	299.8	43	21.5	12	17.4	2000	5.15	0
5	0.001288	15.95	299.8	43	21.5	12	17.3	2000	5.15	0
6	0.001268	16.41	299.8	43	21.5	12	17.4	2000	5.15	0

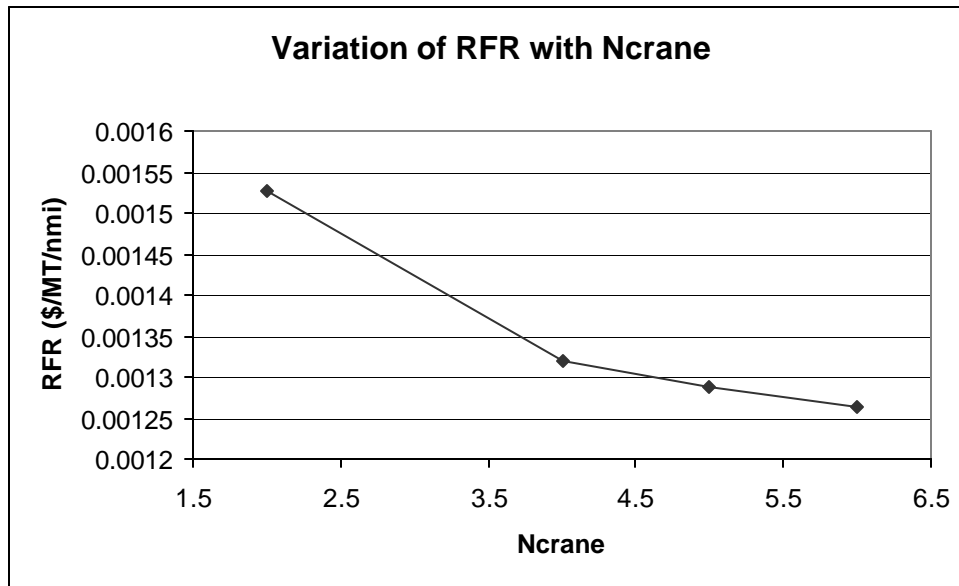


Figure 8.6: Variation of RFR with N_{crane} .

Chapter 9: Examples of Results of Optimization

This chapter features certain examples that indicate the efficacy of the MDO design process. The list of examples is divided into three sets. The first set explores the effect of certain key parameters on the design with the B/D constraint activated. A B/D value of 2.0 (as mentioned earlier) ensures ship dimensions that would allow the ship designer to satisfy structural requirements. The second set of examples pertains to the addition of another structural constraint and its effect on the overall design. The B/D constraint is deactivated in this set. The third set of examples, an effort is made to remove the approximation of the container capacity. The original step-wise discontinuous function is used to calculate the containers that was calculated earlier by the linear surface fit.

In each example, the hull shape is adjusted by blending two basis hulls. There is a need for only one shape design variable: c_1 (as $c_2 = 1 - c_1$). Basis hull 1 is finer than basis hull 2. The important parameters used in the design process are: ship life = 20 years, interest rate = 8%, weight per TEU = 12 MT and specific fuel consumption = 120 g/BHP/hr. Other than in example 4, the cost of fuel is \$80/T. Table 9.1 is a list of initial design variables and bounds that have been in all the examples. This list is just indicative of a default set of initial conditions used in the examples. Any changes to the list have been indicated in the description of each example.

Table 9.1: Initial Design Variables and Bounds.

	Loa (m)	Beam (m)	Depth (m)	Draft (m)	Speed (knots)	Ballast (T)	NT_d	C₁
Lower Bound	130	15	4	4	4	0	0	0
Initial Value	200	35	15	10	10	0	4	0.5
Upper Bound	300	43	35	25	35	10000	10	1.0

9.1 Examples on variations in key parameters

Table 9.2 lists the results of optimization for variations in key parameters.

Table 9.2: Design variable values and various ship characteristics at the optimum point for the first set of examples.

Parameters	Example 1	Example 2	Example 3	Example 4
Loa (m)	299.8	300	300	299.9
Beam (m)	43	43	43	43
Depth (m)	21.5	21.5	21.5	21.5
Draft (m)	13.6	15	13.5	13.5
Speed (knots)	17.5	25	17.6	16.2
Ballast (T)	10000	10000	10000	10000
NT_d	6.1	5.89	5.91	6.05
c₁	0	1	0.13	0
RFR (\$/MT/nmi)	0.001156	0.001265	0.001383	0.001253
# of TEU's	6266	5769	6111	6233
Displacement (T)	114476.9	110730.6	111683.8	113533.3
SHP	29538.6	78923	28550.1	22332.8
GM	0.33	0.32	0.32	0.33

Example 1: The initial values are the same with the B/D constraint activated. The maximum value of this ratio is 2.0. Here the ballast and length values go to their maximum to satisfy the GM constraint while allowing a large NT_d. C₁ is zero indicating a full ship.

Example 2: The initial values are the same as example 1 except that the minimum value for speed is set at 25 knots. The resultant speed is at its lower bound. There is an increase of 9.5% in the RFR, which is due to the increase in SHP by almost 167%. The increase in speed from the previous example increases the SHP that in turn increases the annual fuel and machinery costs. This increase is reflected in the increase in RFR. C₁ is 1.0 indicating that a finer ship is achieved. This is done to decrease the resistance of the ship at higher speed.

Example 3: All the input variables are the same as example 1 except that the range is decreased from 7000 nmi to 3500 nmi. All the parameters and design variables are comparable to example 1 except the RFR. The RFR is higher by almost 19.6%. As the range is halved, the number of round trips in a year has increased but this increase is not

directly proportional with the decrease in range. As the cargo being carried is the same, the port waiting time for both the ships is the same but total time spent in the port annually is greater. The number of round trips is 26.67 as compared to 16.21 for example 1. Hence, cargo is carried for lesser distance as compared to example 1. The increase in RFR is indicated by this difference.

Example 4: All input variables are the same as example 1 except that the cost of fuel is increased from \$80/T to \$120/T to reflect current bunker fuel costs. In a bid to reduce annual fuel costs, the optimizer reduces the speed of the ship by 7.5% and hence reduces the RFR. The number of trips made in a year are now lesser than example 1 and hence the there is an increase in RFR by 8.5%.

9.3 Examples with L/D constraint

Table 9.3 is a compilation of the second set of examples: results with L/D constraint.

Table 9.3: Results of optimization with L/D constraint

Parameters	Example 5	Example 6	Example 7	Example 8
Loa (m)	244.3	300	300	215.2
Beam (m)	43	43	43	43
Depth (m)	16.3	20	20.1	16.6
Draft (m)	12.4	11.8	15.3	12.5
Speed (knots)	16.3	18.1	16.7	16
Ballast (T)	0	0	17800	300
NTd	8.15	5.6	7.94	8.09
c1	0.24	0.17	0.2	0.29
RFR (\$/MT/nmi)	0.000949	0.001177	0.000974	0.000996
Displacement (T)	87914.7	95851.4	134237.6	77597
SHp	19941.6	29617.2	26101.4	17750.3
GM	0.48	0.33	0.41	0.42
TEU	5724	5693	7391	5007

Example 5: In this example the maximum value allowed for L/D is 15. There is a decrease of 18% in the RFR in comparison with example 1. Even though the length is lower in this example, the cost of increasing depth is significant. The value of NT_d is also

higher in this example. Contrary to earlier results, better RFR can be achieved with lesser length without compromising transverse and structural stability.

Example 6 The input variables are the same as example 5 except that length of the ship is constrained at its upper bound of 300 m. It was thought that a larger ship, which will be carrying a larger amount of cargo, would result in a better RFR. Contrary to expectations, the optimizer chose to decrease NT_d thereby increasing the RFR by 24%. The ship is not able to carry more cargo on deck due to the GM constraint. The optimizer is not able to recognize that ballast can be added to reduce the CG of the ship and hence increasing the GM. Figure (7.2f) shows the insensitivity of RFR to ballast. As the L/D constraint is more prominent than the GM constraint, the optimizer chooses to decrease NT_d (to reduce GM) rather than to increase ballast.

Example 7: The input conditions are similar to example 6 except that there is an initial ballast of 20000 T. In this case, the optimizer chose to increase NT_d until the GM constraint is satisfied. The RFR has also decreased and is comparable to example 5.

Example 8: The input conditions are the same as example 5 with the exception that the maximum value of L/D is set at 13. This is a more appropriate ratio in accordance with Figure (4.2). This example proves that depth is a significant cost parameter to the overall design. Figure (7.2a) indicates the sensitivity of RFR to length and indicates that RFR decrease with increasing length. In spite of this factor, the optimizer prefers to decrease length. The maximum value of L/D turns to be an important design constraint. The RFR is similar with example 5 and 7.

9.3 Secondary Optimization

As discussed in chapter (5) the cargo capacity is a stepwise discontinuous function. Equations (5.1 - 5.4) indicate that though the number of TEU's varies with length, beam and draft (or NT_d), it is not a continuous variable. Due to this, the optimizer ends getting trapped within each step of the discontinuous step. As a result, the process of optimization leads to many local minima. This prevents the optimizer from effectively calculating the gradients of the objective and the constraint functions. To counter this effect, steps have been taken to make the cargo capacity function a smooth variation to the design variables. Equations (5.5-5.6) indicate this capacity response surface to the

design variables. Figure (5.1) illustrates the effect of the discontinuous function and the continuous response surface in two dimensions.

The third set of examples deals with the effect of using the original stepwise discontinuous function in lieu of the smooth response fit. This process is termed secondary optimization and is provided as an option to the user once the primary optimization is completed.

The process of secondary optimization can be interpreted as moving along the planes of a cube with the optimum point as the center of the cube. Figure (9.1) illustrates such a cube.

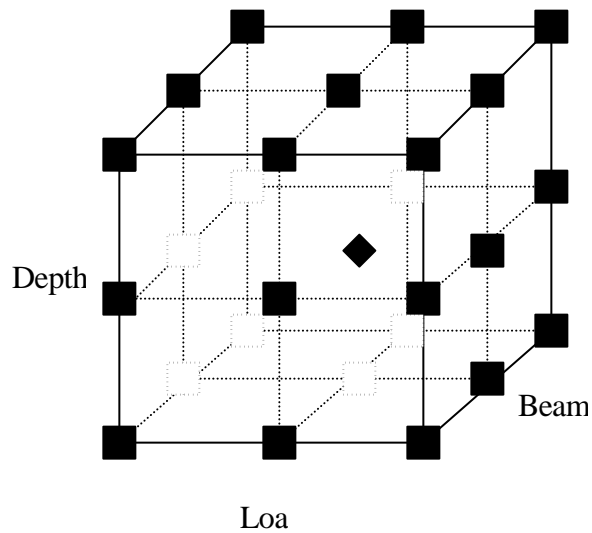


Figure 9.1: Cube illustrating the process of secondary optimization

Key: ◆ indicates the optimum point (center of cube)
 ■ □ indicates a point to be searched for optimum.

Once the optimum point is found using the linear response fit, the capacity function is replaced with the stepwise discontinuous capacity function. The axes of the cube are the dimensions of the ship – length, beam and depth. The dimensions of the cube are determined by the bounds on these variables. It is made sure that the optimizer does not go to points that are beyond the initial bounds set by the user. The bounds are also determined by step sizes as indicated here. Each step is determined by the dimensions of the container that is being used to carry the cargo. The containers are measured in TEU's

(Twenty Foot Equivalent Unit, 6.1m x 2.44m x 2.44 m). Hence, the step in either dimension is never more than the dimensions of the container in each direction. Though, the points are present for convenience, the optimizer is free to move in any direction in a single plane. The number of TEU's actually does not remain constant even though the dimensions of the ship are. The number of TEU's also depends on the block coefficient, which is free to vary. These points represent possible plateaus where the optimizer has to perform the process of optimization within the given bounds. It is convenient to just check these plateaus because, typically, the optimum will be near the lower limit of length and beam for any TEU capacity plateau since this will be the least expensive ship to build and operate for its cargo capacity. If a constraint is violated along the process of reaching any of the plateau points, the optimizer terminates the optimization along the plane. The process is then carried out in other possible plateaus with the new side bound. If a point is found that has lower RFR than the initial optimum, then secondary optimization is performed around this new point. A new cube is formed around this point with new side bounds. All plateaus that might represent a step up or down in the three dimensions have to be checked. The formulation of the process of secondary optimization also makes sure that redundant points are not checked. Points are also checked for constraint violations and initial bounds set for the user. The process is continued until a feasible point with the lowest RFR is found and verified with its neighboring domain.

Table (9.4) is an example of the process of secondary optimization. The converged point is listed in the first column. The optimizer has determined the next optimum (lower RFR) by stepping down in the length and beam directions. Now a cube is formed around this point that has a lower RFR than the initial convergence point.

In Table (9.4), though there are 26 points around the secondary optimum (sec 1), 17 of them have been neglected because of violated constraints and side bounds. The lowest RFR is obtained in the case of sec 1 and is presented to the user as the final result.

Table 9.4: Results of Secondary optimization.

Parameters	First	Sec 1	Sec 2	Sec 3	Sec 4	Sec 5	Sec 6	Sec 7	Sec 8	Sec 9
Loa (m)	299.9	294.7	294.6	288.4	294.6	294.7	288.4	288.3	294.7	288.3
Beam (m)	43	42.7	43	42.7	42.6	42.7	42.7	42.7	41.5	41.5
Depth (m)	21.5	21.4	23.8	21.4	21.4	21.3	21.4	21.3	20.7	20.7
Draft (m)	11.4	11.7	11.8	11.8	11.4	11.4	11.4	11.4	11.1	11.1
Speed (knots)	19.2	19	18.7	19.1	17.9	17.5	15.9	18.1	18.5	18.6
Ballast (T)	720	730	740	760	930	880	1003	730	1250	740
NTd	5.06	5.21	3.84	5.23	5.1	5.43	5.12	5.42	5.09	5.11
c1	0	0	0.1	0	0.01	0	0	0	0	0
RFR (\$/nmi/MT)	0.001172	0.001124	0.00128	0.001133	0.001164	0.001165	0.001198	0.001172	0.00118	0.00119
Displacement (T)	101832.8	102946	100092.7	101004.5	99581.3	99773.5	97261.9	97753.7	94269.6	92445.7
# TEU's	5821	5991	5518	5874	5760	5771	5645	5635	5407	5300
SHP	32031.7	30431.6	28823.2	31005.7	24759.3	23236.3	16891.2	25760.3	26697.1	27203.3
Cb	0.6972	0.6939	0.6878	0.6943	0.6948	0.6966	0.6968	0.6965	0.696	0.6964
L Lower Bound	130	294.6	294.6	288.2	294.6	294.6	288.2	288.2	294.6	288.2
L Upper Bound	300	300	300	294.6	300	300	294.6	294.6	300	294.6
B Lower Bound	15	42.7	42.7	42.7	41.5	42.7	41.5	42.7	41.5	41.5
B Upper Bound	43	43	43	43	42.7	43	42.7	43	42.7	42.7
D Lower Bound	4	21.4	23.8	21.4	21.4	18.9	21.4	18.9	18.9	18.9
D Upper Bound	35	23.8	26.2	23.8	23.8	21.3	23.8	21.3	21.3	21.3

Chapter 10: Conclusions and Future Work

10.1 Conclusions:

A prototype MDO tool is developed based on Microsoft's COM framework using ATL. With this design, the modules can be modified with minimum programming effort. The user interface gives the user flexibility to manipulate relevant parameters that affect the design. A geometric shape manipulation scheme is developed in which the hull form was generated by blending two hull forms. This MDO tool is used to design a container ship with the required freight rate as the objective to be minimized. It is noticed that without a structural constraint, the design tends towards one with maximum length and beam. This led to unreasonably large ratios of B/D and L/D.

A B/D constraint is applied to the design to get a better structural design. Results with this constraint enabled have pointed in the direction of adding two other design variables. This constraint increases the depth of the ship. With the increase in depth, the center of gravity of the ship also rises decreasing the GM of the ship. This lowering of GM adversely affects the GM constraint. The number of tiers on deck (NT_d) is made a design variable to enable the optimizer to have the flexibility of manipulating the cargo carrying capacity. It was noticed that the ship is unable to have a high NT_d because of the violation of the GM constraint. Hence, ballast has also been added as a design variable to reduce the center of gravity of the ship increasing the GM of the ship. This feature enables the optimizer to carry greater cargo on deck improving the objective function.

An effort is made to analyze the efficacy of the MDO tool by varying various parameters that affect the design. Technology factors have been introduced which give an insight on effect of key parameters. They also reflect on future design trends. Three evaluation tools: sensitivity analysis, alpha plots and restart option have been incorporated in the design process to gauge the results of optimization.

The effect of another structural constraint L/D was also investigated. This constraint tends to bring down the overall length and is inconclusive in its results. Further analysis of this constraint is needed to draw usable conclusions. The linear response surface approximation was eliminated and the original stepwise discontinuous TEU capacity function is employed in the later examples.

It was found that the minimum of the required freight rate occurred at the lower limits of length and beam on each TEU capacity platform. A systematic search of TEU plateaus in the vicinity of the primary optimum was necessary to define the secondary optimum.

10.2 Future Work:

1. The new structural constraint L/D has to be investigated further.
2. A new constraint has to be added to ensure that the value of ballast is not above a certain percentage of displacement.
3. Investigations of Examples should be done with other basis ships.
4. It was noticed that the RFR has very high sensitivity to the depth of the ship. A small increase in depth increases the RFR disproportionately. This condition is attributed to the weight module. A different weight analysis has to be developed to improve the sensitivity.

References

- [1] **“A global optimization model for ship design”**, T.Ray, R.P. Gokarn and O.P.Sha, *Computers in Industry*, 26(1995), 175-192
- [2] **“Marine Design: the Multiple Criteria Approach”**, P.Sen, *Transactions of the Royal Institute of Naval Architecture*, Vol. 134, Part B, 1992
- [3] **“Optimization Techniques in Ship Concept Design”**, A.J. Keane, W.G. Price and R.D Schachter, *Transactions of the Royal Institute of Naval Architecture*, A.133, 1991.
- [4] **“A Model for Multidisciplinary design Optimization of Container Ships”**, Vikram Ganesan, Masters thesis, Virginia Tech, July 1999.
- [5] **“Formulation of a Multi-disciplinary Design Optimization of Container Ships”**, Ying Chen, Masters thesis, Virginia Tech, September 1999.
- [6] **“Practical ship Design”**, Watson, David G.M. First Edition.
- [7] **“Principles of Naval Architecture”**, second revision, Edward V. Lewis, editor, *Society of Naval Architecture and Marine Engineering*, 1988.
- [8] **“The Practical Approach of Economics of Merchant Ship Design”**, Benford, Harry, Marine Technology, *The Society of Naval Architecture and Marine Engineering*, January 1967.
- [9] **“Ship Design and Construction”**, Taggart, Robert, editor, *The Society of Naval Architecture and Marine Engineering*, 1980.
- [10] **“DOT Users Manual”**, Version 4.20, *Vanderplaats Research and Development*, Inc.
- [11] **“General Cargo ship Economics and Design”**, Benford, Harry, *College of Engineering, University of Michigan*, 1968.
- [12] **“A statistical re-analysis of resistance and propulsion data”**, J. Holtrop and G.G.J Mennen, *International Ship Building Progress*, Vol. 31, November 1984.
- [13] **“Transport Capacity and economics of container ships from a production theory point of View”**, Kuvas, Jomar, *Royal Institute of Naval Architects*, 1974
- [14] **Coast Guard**, *Department of Transportation*, 46 CFR Ch.1(10-1-97) edition

VITA

Name	Amlan Dasgupta
DOB	07/23/1977
Email Address	amlan@vt.edu
Address	1207 Snyder Lane, #1200F, Blacksburg, VA 24060
Major	Ocean Engineering
Degree	Master of Science