Chapter 4

Overview of the Optimization Process

4.1 Statement of the Optimization Problem

Once the various modules have been constructed it is essential to facilitate flow of data amongst them and to link them with the geometry manipulation scheme and the optimization tool. Figure 4.1 is a flowchart of the design process.

The design variables for the problem are:

1. Loa: Length, overall in meters
2. B: Beam, maximum in meters
3. D: Depth, at side in meters
4. T: Draft, design in meters
5. Vk: Speed in knots
6. C: Blending coefficient

The number alongside the symbol for the blending coefficient indicates the hull form to which it is applied. In this work, we use three basis hull forms. Therefore,

C14: Blending coefficient applied to basis hull 14.
C11: Blending coefficient applied to basis hull 11.
C22: Blending coefficient applied to basis hull 22.

The constraints on the design are as stated in section 2.4. They are stated here for convenient reference.

1. Weight = Displacement
2. $\frac{\text{Loa}}{\text{D}} \geq 8.3$
3. $\text{GM} \geq \frac{\text{P} \times \text{LA} \times \text{H}}{\text{Disp} \times \tan(\text{è})}$
4. $\text{Freeboard} \geq \text{Freeboard\_min}$
5. $\text{Troll} \quad \geq \text{Troll\_min}$

The input file allows the user to initiate the design process by identifying an initial vector of design variables subject to lower and upper bounds. The bounds check the feasibility of the design space by considering practical limitations on the design variables. For example, the bounds on the length, beam and depth are subject to:

1. Length of wharf
2. Breadth of channel
3. Depth of harbor

respectively. The upper bound on the draft is such that it does not exceed the lower bound on the depth. An objective function is identified. The user can choose from minimizing the required freight rate (2.62) and maximizing the return on investment (2.63). Three methods provided in DOT are available for optimization (explained in section 4.3). On completion of the optimization process, the final results along with the convergence history and the sensitivity of the objective function to each of the design variables are available in different output files. A discussion regarding iteration history and sensitivity analysis follows in sections 4.4.1 and 4.4.2 respectively.

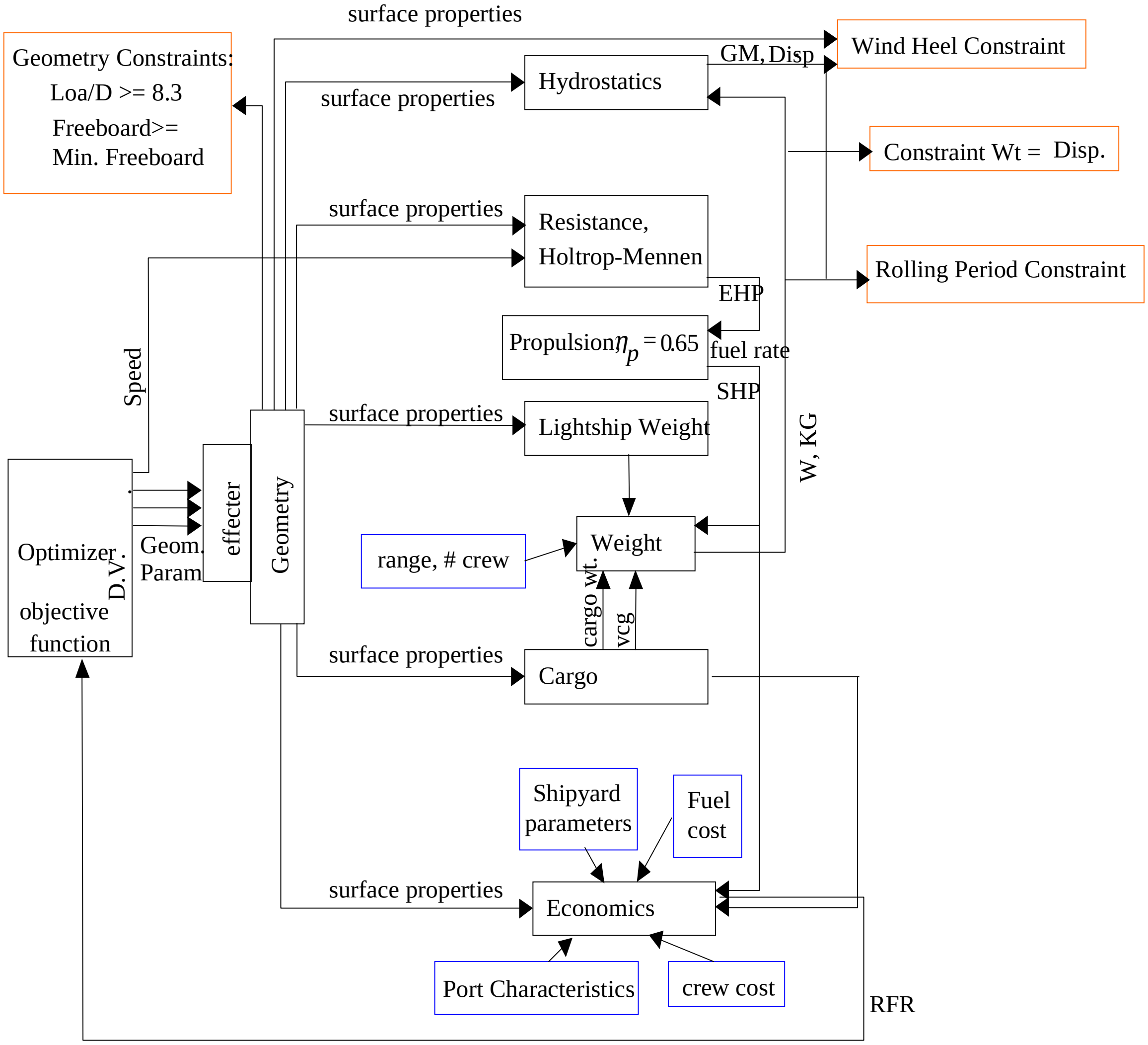

Figure 4.1 Flowchart for MDO

The input file also allows the user to define inputs pertaining to shipyard parameters and port characteristics that are fixed during the optimization process. A list of these input parameters is provided in the list of symbols. They are mentioned here again for convenient reference.

Symbol	Description	Value used in this work
Chargerate	Rate charged by the ship owner to ship cargo in dollars/ ton/ mile	0.0064
Chull:	Cost of hull engineering material in dollars/ ton	3500
Cof:	Cost of outfit material in dollars / ton	1500
Csh:	Cost of hull steel material in dollars / ton	400
DBH:	Double bottom height in meters	1.83
Dst:	Range in nautical miles	7000
Fcost:	Cost of Bunker C fuel oil in dollars / ton	80
HCH:	Hatch coaming height in meters	1.83
Ir:	Interest rate in percentage	0.08
Lc:	Labor cost in dollars / hour. For simplicity we assume the same labor cost for hull steel, hull engineering, outfit and machinery	20
Ot:	On-hire per year time in days	350
Pwt:	Port waiting time in days	2
SFC:	Specific fuel consumption in gms/ shp/ hour	120
Sl:	Ship life in years	20
TSLU:	TEUs loaded/ Unloaded per day per crane	1440
Wpc:	Weight per TEU in metric tons	12

The geometry module is responsible for the hull geometry manipulation. A detailed explanation follows in section 4.2. The hydrostatic properties of the resultant hull forms are calculated by the hydrostatics module [15]. Once the objective and constraint functions are evaluated, the data is passed on to the optimizer, which evaluates their gradients and moves in a direction that minimizes the objective function while satisfying the constraints. An explanation of the algorithms provided in DOT for optimization is given in section 4.3.

It is found that dividing the objective function by the range, which is a fixed parameter defined by the user, induces inconsistencies in convergence of the optimization results. This is shown in the results as obtained in cases 4.1.1 and 4.1.2 in Table 4.1. Normalizing the objective function results in better convergence. This can be seen from cases 4.1.3 and 4.1.4 in Table 4.1. The cause could be that normalization of the objective function causes the Hessian and hence the search direction to be normalized thus resulting in better convergence.

Table 4.1 Importance of Objective function scaling

Sub-case	Loa	B	D	T	Vk	C11	C22	Obj: RFR	IT	TC	AC
Case 4.1.1: Objective function: RFR in dollars/ metric ton											
1	300	40.5	14.1	9.4	16.97	1.00	0.00	7.63537	8	C	Troll,Fb
2	300	40.8	14.3	9.5	17.71	1.00	0.00	7.63397	6	C	Troll,Fb
3	300	42.9	15.4	9.9	17.7	0.98	0.02	7.6241	7	A,C	Troll
4	**300**	**43**	**15.4**	**9.9**	**17.72**	**1.00**	**0.00**	**7.6229**	**3**	**A,C**	**Troll**
Case 4.1.2: Objective function: RFR in dollars/ metric ton/ nautical mile											
1	140.2	32.1	12.0	8.1	4.02	0.00	1.00	.003807	3	C	-
2	300	41.8	15.9	10.8	28.85	1.00	0.00	.001647	5	C	-
3	202.1	39.4	14.3	9.4	19.51	0.48	0.52	.001377	3	C	Troll
4	**202.1**	**39.4**	**14.3**	**9.4**	**19.48**	**0.48**	**0.52**	**.001376**	**2**	**C**	**Troll**
Case 4.1.3: Normalized Objective function: RFR in dollars/ metric ton											
1	300	38.3	13.9	9.2	16.8	0.18	0.82	7.7521	13	C	Troll,Fb
2	300	43	15.4	9.9	17.58	1.00	0.00	7.6228	9	A,C	Troll
3	300	40.8	14.6	9.5	17.18	0.62	0.38	7.6578	10	A,C	Troll
4	300	43	15.4	9.9	17.64	1.00	0.00	7.6243	2	A,B,C	Troll
5	**300**	**43**	**15.2**	**9.8**	**17.58**	**1.00**	**0.00**	**7.6229**	**6**	**C**	**Troll**
Case 4.1.4: Normalized Objective function: RFR in dollars/ metric ton / nautical mile											
1	300	43	15.3	10.0	15.23	0.31	0.69	.001053	10	C	Troll
2	300	43	14.7	10.0	17.62	1.00	0.00	.001029	7	C	Troll,Fb
3	300	43	15.0	10.0	17.28	0.60	0.40	.001033	7	C	Troll
4	**300**	**43**	**14.7**	**10.0**	**17.62**	**1.00**	**0.00**	**.001029**	**7**	**C**	**Troll,Fb**

4.2 Overview of Hull Geometry Manipulation

The hull geometry manipulation is carried out as a weighted average of user defined basis hull forms. The design variables controlling the geometry are the weighting factors or the blending coefficients of the basis hull forms, length, beam and depth. When using two basis hulls [15], the vector of Fastship net points, H, defining the hull is given by

$$H = c_1 H_1 + c_2 H_2$$

Such that

$$c_1 + c_2 = 1$$

where

c_1 and c_2 are the weighting factors of the basis hull forms and

H_1 and H_2 are the net point vectors defining the basis hulls.

We have only one design variable for the weighting factors from the fact that they should sum to one. Once we have the resulting hull form, its hydrostatic properties are calculated by integration.

4.3 The Optimization Process and DOT [16]

Numerical optimization solves the nonlinear constrained problem; to find a set of design variables, X_i , i = 1, …, N ; constrained in vector X such as to

Minimize F(X)
Subject to

inequality constraints	$g_j (X) <= 0$	j = 1,…,M
equality constraints	$h_k (X)= 0$	k = 1,…,L
bounds on the design variables	$X_i^L <= X_i <= X_i^U$	i = 1,…,N

The DOT program does not directly deal with the equality constraint in the form shown above. It breaks the equality constraint down into two inequalities of the form:

$$g_j(X) <= 0$$
$$-g_j(X) <= 0$$

These two equal and opposite inequality constraints will force the function to become an equality.

When it is desired to maximize the objective function, we express the problem as seeking to minimize the negative of that function. To find the minimum of a function DOT calculates the gradients of that function by finite differences. The gradient is a vector of partial derivatives of the function with respect to each design variable independently. This slope gives us the search direction, say S. The number of steps we take in that direction is a scalar parameter. Let us call it α. Given a starting point X^0 where X is a vector of design variables, we move in a direction given by S until we encounter a minimum or a constraint in that direction. We can now update the description of our location by the mathematical expression:

$$X^1 = X^0 + \alpha S^1$$

where superscript 1 denotes the iteration number. This completes one iteration in the design process. In order to make any further improvement in the design we must find a new search direction S^2 that will continue to minimize our function while not violating any constraints. We continue searching and correcting in this direction until we can make no further improvement. The sub-problem of finding a new search direction is repeated, followed by a continued search until no search direction can be found that will improve the design without violating one or more constraints. We call this point the “optimum”, but it is not necessarily a global optimum. With this understanding of the optimization process, we proceed to discuss briefly the three methods provided in DOT for constrained minimization. For a more detailed discussion of these methods, the reader is referred to [13].

4.3.1 Modified Method of Feasible Directions (MMFD)

We seek to find the set of design variables, X_i , i=1,N, contained in the vector X that will

Minimize F(X)

Subject to
$g_j(X) <= 0$ j = 1,…,M
$X_i^L <= X_i <= X_i^U$ i = 1,…,N

We give an initial X- vector, X^0, and update the design according to the following equation:
$X^q = X^{q-1} + \alpha S^q$

The overall optimization process now proceeds in the following steps:

1. Start, q=0, $X = X^0$
2. q = q+1
3. Evaluate $F(X^{q-1})$ and $g(X^{q-1})$, j = 1,M
4. Identify the set of critical constraints, J
5. Calculate $\nabla F(X^{q-1})$ and $\nabla g(X^{q-1})\ j\in J$
6. Determine a search direction, S^q
7. Perform a one-dimensional search to find α
8. Set $X^q = X^{q-1} + \alpha S^q$
9. Check for convergence to the optimum. If satisfied, exit. Otherwise go to step 2.

The critical parts of the optimization task consists of:

1. Finding a search direction, S^q that will minimize the objective function while not violating any constraints.
2. Finding the scalar parameter α that will minimize $F(X^{q-1} + \alpha S^q)$ subject to the constraints.
3. Test for convergence to the optimum and terminate if convergence is achieved.

4.3.2 Sequential linear Programming (SLP)

We create a Taylor series approximation of the objective and constraint functions. Then we use this approximation for the optimization, instead of the original nonlinear function. By doing this, the value of the objective function and the constraints are inexpensively calculated from the linear approximation. Also, since the approximation is linear, the gradients of the objective and constraint functions are available directly from the Taylor series expansion. The Modified Method of Feasible Directions is used to solve the linear problem.

The steps involved in the general algorithm are:

1. For the current value of the design variables, retain the most critical constraints during the cycle. Typically we retain (5 × Number of design variables) constraints.
2. Create a first order Taylor series expansion of the objective function and the retained constraints with respect to the design variables.
3. Define move limits on the design variables. Typically, during one cycle, the design variables are allowed to change 20 – 40 %.
4. Solve the linearized approximate problem.
5. Check for convergence. If not satisfied, repeat from step 1.

The Taylor series expansion created by DOT as mentioned in step 1 is of the form:

$$F(X) = F(X^{q-1}) + \nabla F(X^{q-1})^T (X^q - X^{q-1}) \qquad (4.1)$$

and

$$g_j(X) = g_j(X^{q-1}) + \nabla g_j (X^{q-1}) (X^q - X^{q-1}) \qquad j \in J \qquad (4.2)$$

where
J is the set of retained constraints.

From the fact that everything in (4.1) and (4.2) is constant except the value of the design variables, X^q we can rewrite these equations as:

$$F(X) = F^0 + \nabla F(X^{q-1})^T X \qquad (4.3)$$

$$g_j(X) = g_j^0 + \nabla g_j (X^{q-1}) X \qquad j \in J \qquad (4.4)$$

where

$$F^0 = F(X^{q-1}) - \nabla F(X^{q-1})^T X^{q-1} \qquad (4.5)$$

and

$$g_j^0 = g_j(X^{q-1}) - \nabla g_j (X^{q-1}) X^{q-1} \qquad j \in J \qquad (4.6)$$

We now solve the linear approximate problem;

Minimize $F(X^q)$
Subject to

$g_j(X^q) <= 0 \qquad j \in J$

$X_i^L <= X_i <= X_i^U \quad i = 1,\ldots,N$

where
$$X_i^L = X_i^q - D|X_i q| \qquad (4.7)$$
and
$$X_i^U = X_i^q + D|X_i q| \qquad (4.8)$$

The multiplier D in (4.7) and (4.8) are set by the parameter RMVLMZ in DOT. Depending on the progress of the optimization, this parameter is sequentially reduced.

The convergence criteria for the SLP method are the same as those for the other methods. If there are constraint violations, we continue until the constraints are satisfied or the maximum number of iterations is reached.

4.3.3 Sequential Quadratic Programming

We create a Taylor series approximation of the objective and constraint functions. But unlike SLP where we minimize the linearized objective, we create a quadratic approximation of the objective function. We use the linearized constraints with this to create a direction finding sub-problem of the form:

Minimize $Q(S) = F^0 + \nabla F^T S + \frac{1}{2} S^T B S$
Subject to

$(\nabla g_j)^T S + g_j^0 \quad <= \quad 0 \qquad j = 1,\ldots,M$

The sub-problem is solved using the Modified Method of Feasible Directions. The matrix B is a positive definite matrix that is initially the identity matrix. On subsequent iterations, B is updated to approach the Hessian of the Lagrangian function. Once the sub-problem is solved, we calculate the lagrange multipliers λ_j $j = 1,\ldots,M$.
We now search in direction S using the approximate Lagrangian function. That is, we find α to

$$\text{Minimize} \quad \Phi = F(X) + \sum_{j=1}^{M} u_j \max\left[0, g_j(X)\right]$$

where

$X = X^{q-1} + \text{á}\, S$

$U_j = \left|\lambda_j\right|$ $\qquad$ $j = 1,\ldots,M$ first iteration

$U_j = \max\left[\left|\text{ë}_j\right|, \frac{1}{2}\left(u_j^{1} + \left|\text{ë}_j\right|\right)\right]$ $\qquad$ $j = 1,\ldots,M$ subsequent iterations

And $u_j^{1} = u_j$ from the previous iteration.

After the one-dimensional search is complete, the B matrix is updated using the BFGS formula. The next iteration begins.

The termination criteria used in DOT for the three methods described above are:
A: Relative convergence criteria (0.001) met for 2 consecutive iterations
B: Absolute convergence criteria (0.0001) met for 2 consecutive iterations
C: Search vector is less than the specified tolerance (0.0001)